

삼성전자 ARTIK 플랫폼 교육

(기초과정)



Agenda

I. ARTIK 소개

1. ARTIK 소개
2. PS-ED500 소개
3. 개발환경 구축
4. ARTIK Firmware Update

II. ARTIK 기술 교육

1. GPIO 제어
2. PWM 제어
3. 아날로그 제어(1)
4. 아날로그 제어(2)
5. Wi-Fi
6. 소켓 통신
7. 블루투스
8. ARTIK 클라우드(1) – IoT System
9. ARTIK 클라우드(2) – Create Device
10. ARTIK 클라우드(3) – Send Message
11. ARTIK 클라우드(4) – Get Action
12. ARTIK 클라우드(5) – Scenes
13. ARTIK 클라우드(6) – Node-RED
14. Android App 연동
15. ARTIK Cloud Connector

III. ARTIK 유닛 프로젝트 사례

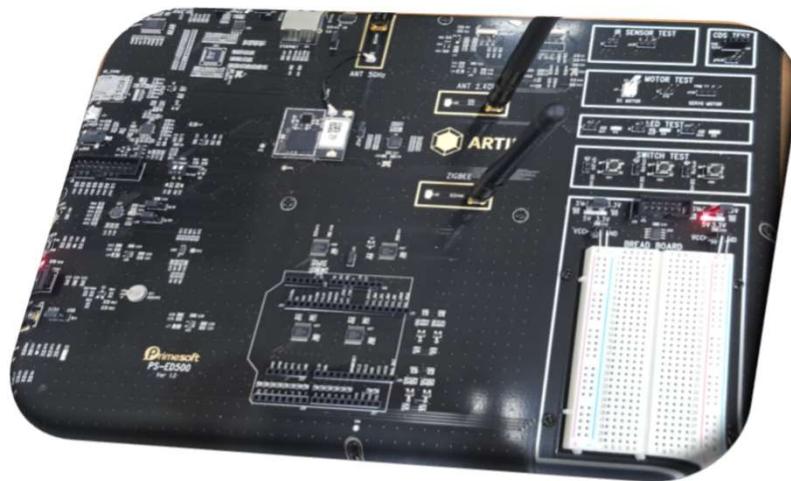
교육 일정

주차	수업내용	비고
1주차	ARTIK 소개, PS-ED500소개, 개발환경 구축	오리엔테이션, 프로젝트 팀 구성
2주차	GPIO 제어	
3주차	PWM 제어	
4주차	아날로그 제어	
5주차	통신(Wi-Fi, Bluetooth)	프로젝트 소개, 선정 및 제안서 작성
6주차	ARTIK Cloud	
7주차	프로젝트 진행	
8주차	중간고사	
9주차	프로젝트 중간발표	
10주차	프로젝트 진행	특허 관련 특강(변리사)
11주차	프로젝트 진행	
12주차	프로젝트 진행	
13주차	프로젝트 진행	
14주차	프로젝트 진행	
15주차	프로젝트 최종발표	
16주차	기말고사	

- ❖ 기본교육 (1주차~6주차) : PS-ED500 사용
- ❖ 프로젝트 (7주차~) : ARTIK 5 developer kit 사용

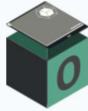
I. ARTIK 소개

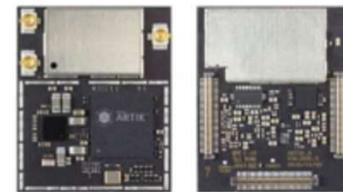
1. ARTIK 소개



Samsung ARTIK Modules

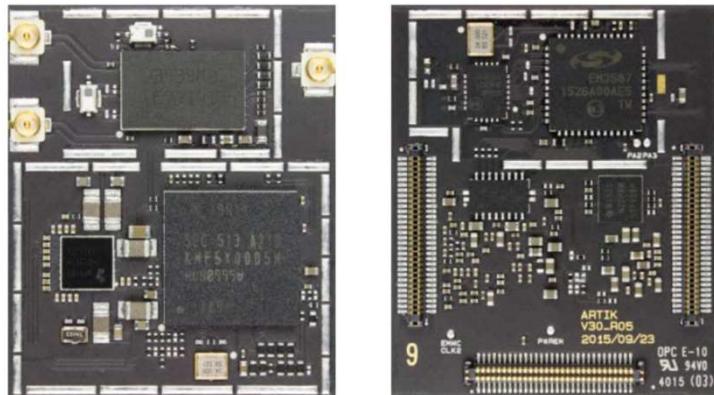
■ ARTIK 0 , ARTIK 5, ARTIK 7 and ARTIK 10

			
<p>ARTIK 0 Family Connected. Flexible. Economical. Tiny. Connected. Flexible. Economical. Tiny. Low power, lightweight, power- and cost-optimized to power single-function "things", like door locks, lights, and sensors. Pre-certified RF hardware saves you time getting your product to market. Choose ARTIK 020 for Bluetooth® applications and ARTIK 030 for Thread and ZigBee®.</p>	<p>ARTIK 5 Family Balanced power and performance. Ideal balance of power and performance for gateways or devices with modest video and processing requirements. Also great for very smart "things". Built for maximum compactness. Ready to integrate with ARTIK Cloud. Includes our hardware Secure Element for tight security and pre-certified RF hardware.</p>	<p>ARTIK 7 Family Powerful. Secure. Communicative. Great performance for high-end gateways with local processing and analytics to improve latency and responsiveness. Ready to integrate with ARTIK Cloud. Includes our hardware Secure Element for tight security and pre-certified RF hardware.</p>	<p>ARTIK 10 Family Maximum performance Maximum performance plus multimedia: use for graphics- and processing-intensive applications, such as autonomous vehicle navigation, intensive 3D graphics or large, immersive displays. Built with ePOP packaging for maximum compactness. Ready to integrate with ARTIK Cloud. Includes our hardware Secure Element for tight security and pre-certified RF hardware.</p>



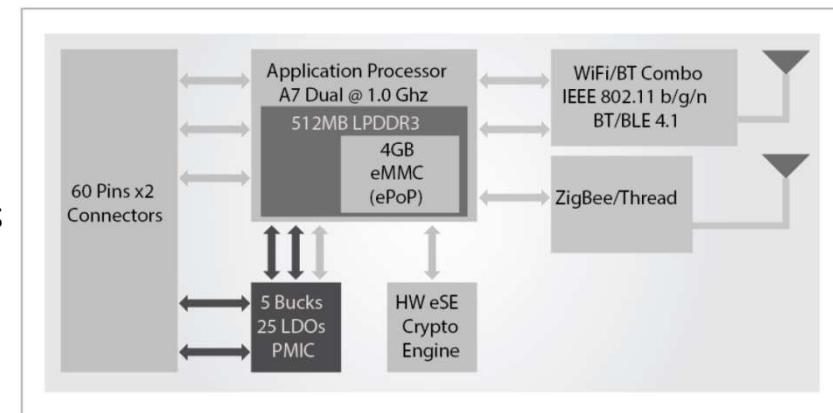
ARTIK 5 module

■ Top/bottom view of ARTIK 5



■ Features

- Highly-integrated System-in-Module
 - The Exynos architecture with dual core ARM Cortex-A7 Process packaged DRAM and Flash memories
 - Wide range of wireless communication options (802.11.a/b/g/n, Bluetooth 4.1, Bluetooth Low Energy(BLE), 802.15.4/ZigBee)



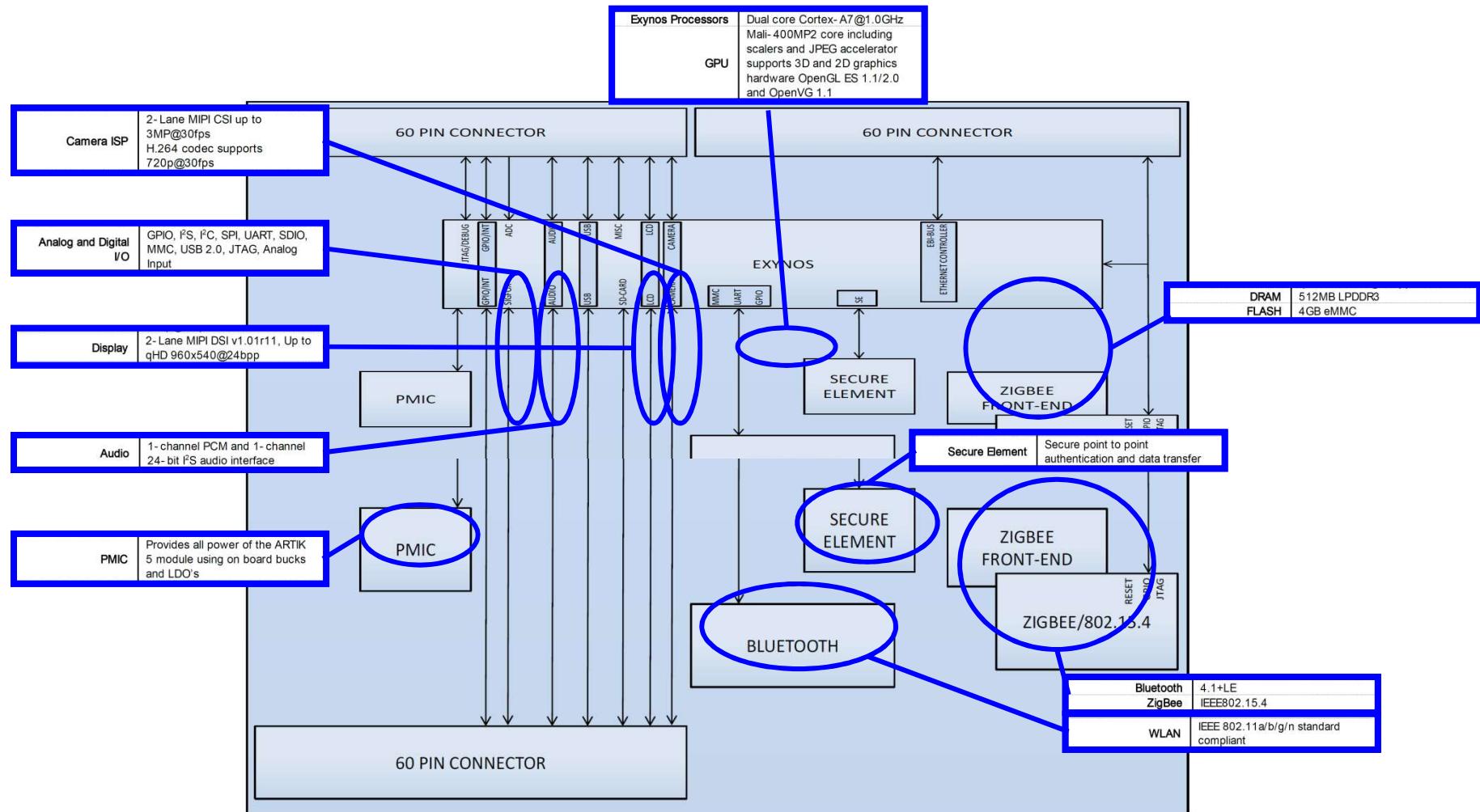
ARTIK 5 module

■ ARTIK 5 vs Atmega128

	 Atmega 128	 ARTIK 5
CPU	Atmega128 made by Atmel	Exynos with Dual core ARM Cortex-A7
Memory	128KB Flash Memory 4KB EEPROM 4KB embedded SRAM Possible to add up to 64kB external memory	4 GB eMMC Flash Memory 512 MB LPDDR3 DRAM
Interfaces	GPIO, ADC, JTAG , USART, PWM	GPIO, I2C, SPI, UART, SDIO USB 2.0, JTAG, Analog Input
Development Environment	AVR Studio program	MQTT(Node-RED, Mosquitto) Putty (Windows, Mac) Minicom (Linux) Arduino sketch program(Verification step)
Communication	USART Communication (No wireless communication without additional module)	WiFi , Bluetooth, Zigbee, WLAN Sigfox(Verification step), Z-Wave(Verification step)

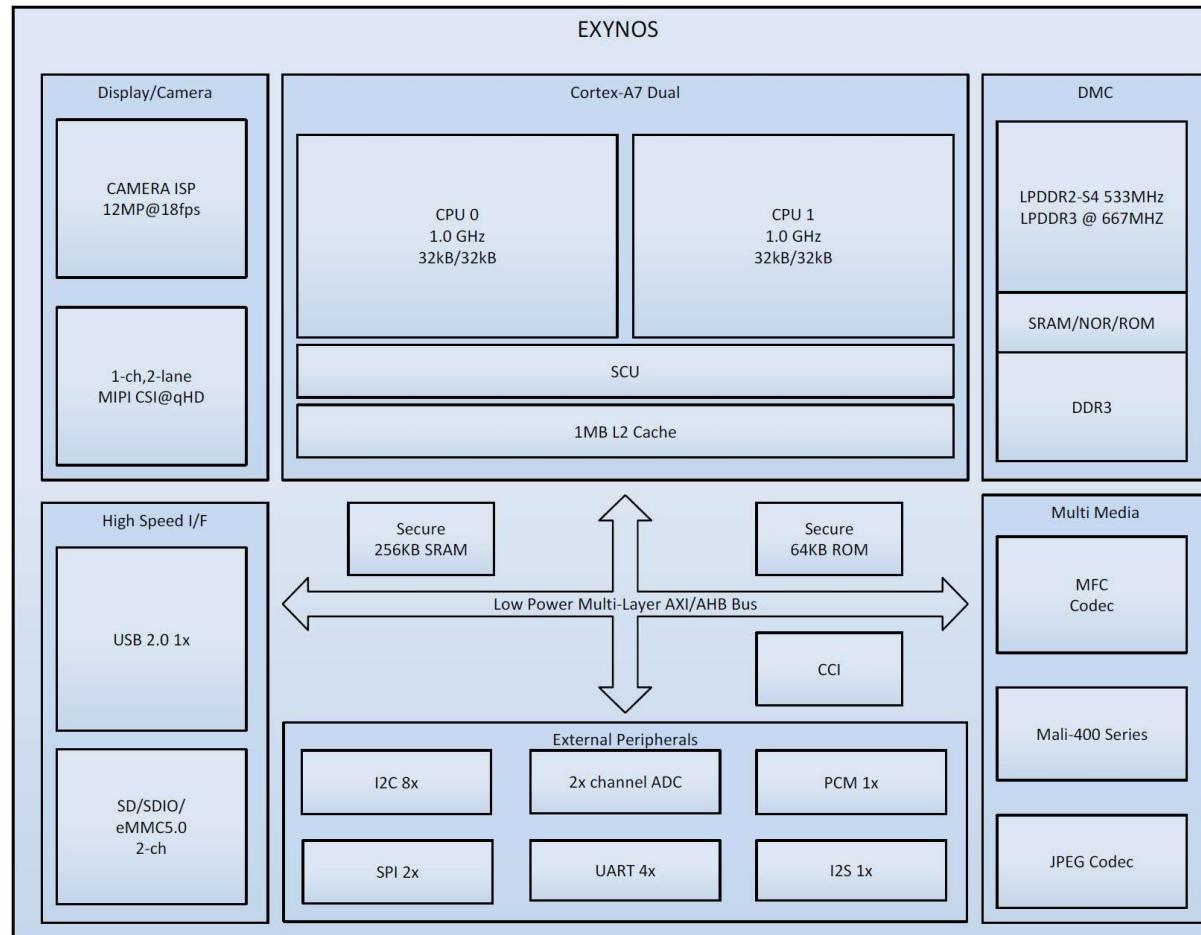
ARTIK 5 module

■ ARTIK 5 Module Block Diagram



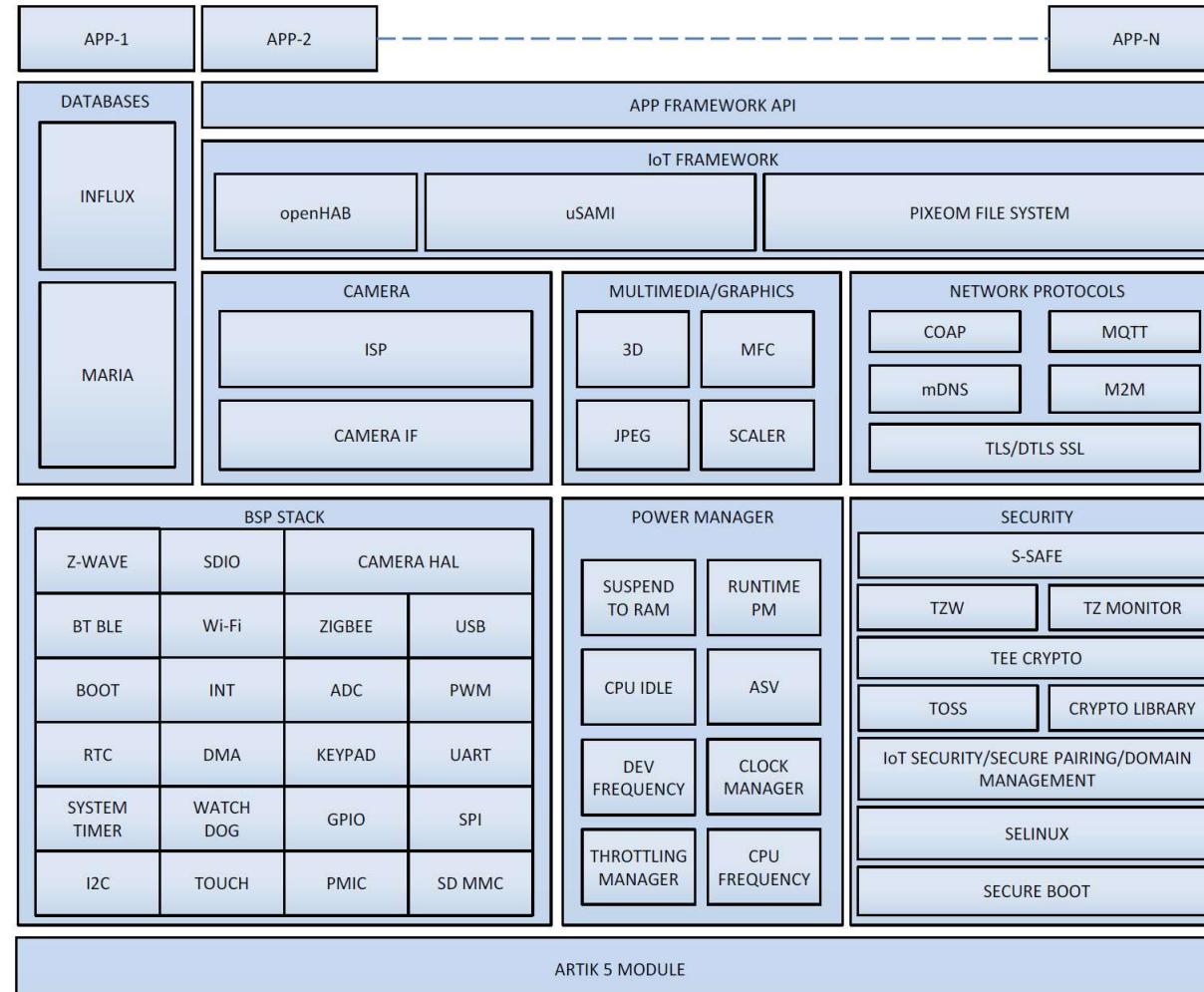
ARTIK 5 module

■ ARTIK 5 Module Exynos Sub System



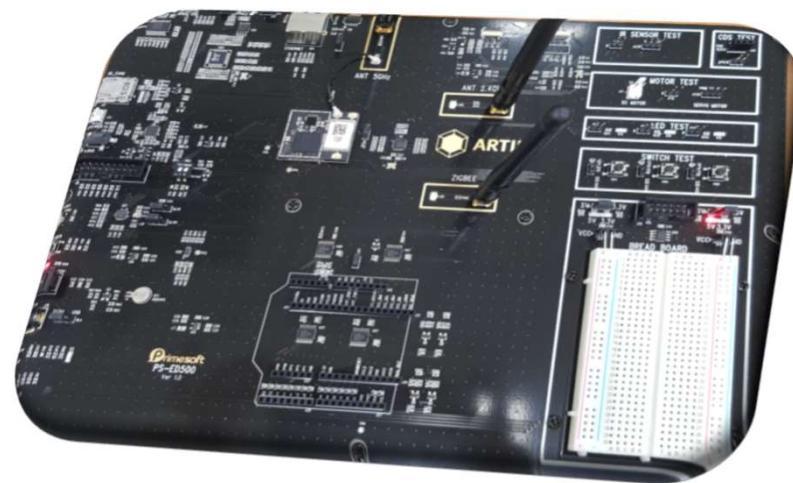
ARTIK 5 module

■ ARTIK 5 Module Software Stack



I. ARTIK 소개

2. ARTIK 교육 키트 소개



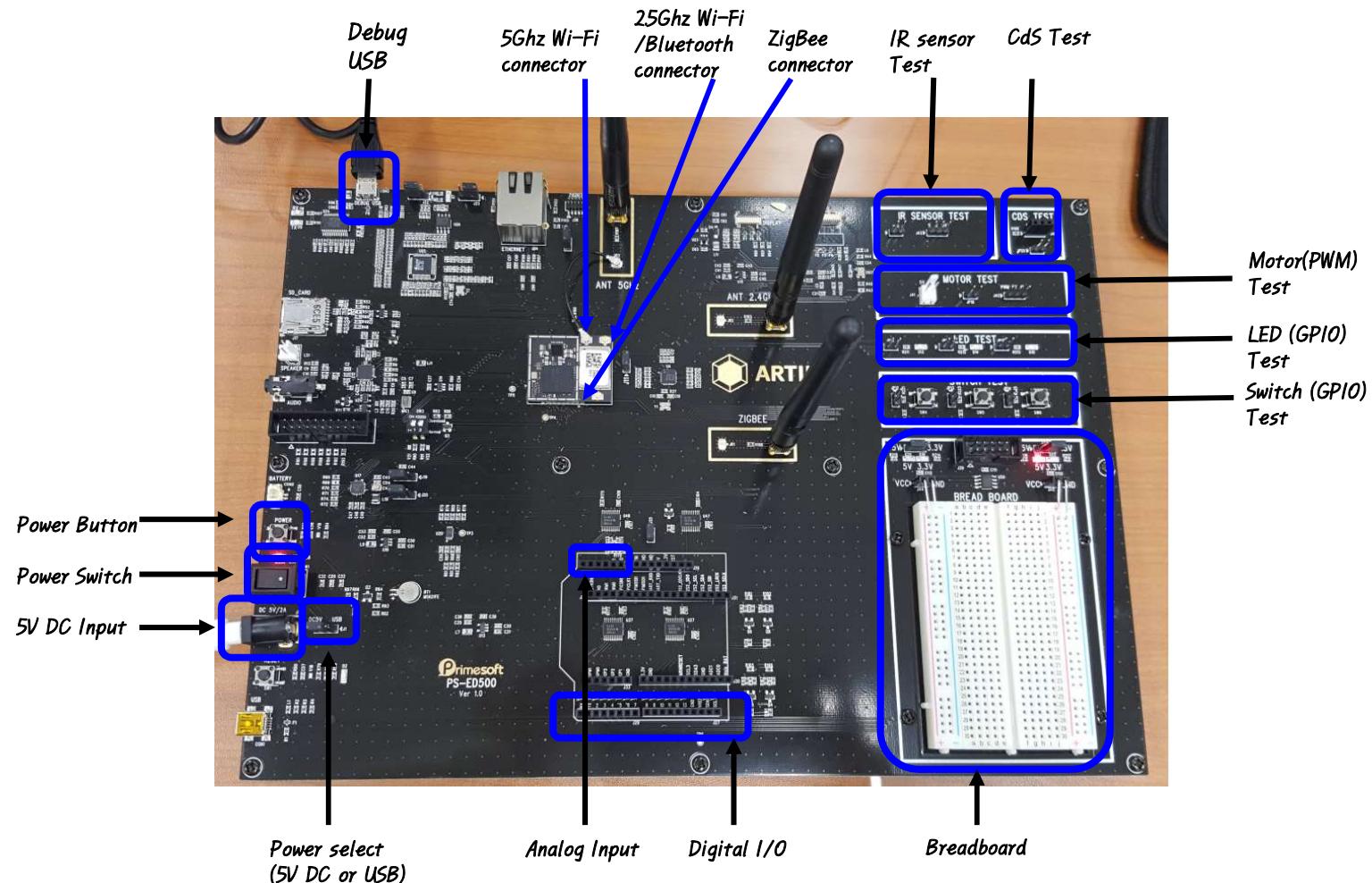
ARTIK Basic : PS-ED500

■ ARTIK Basic Kit



ARTIK Basic : PS-ED500

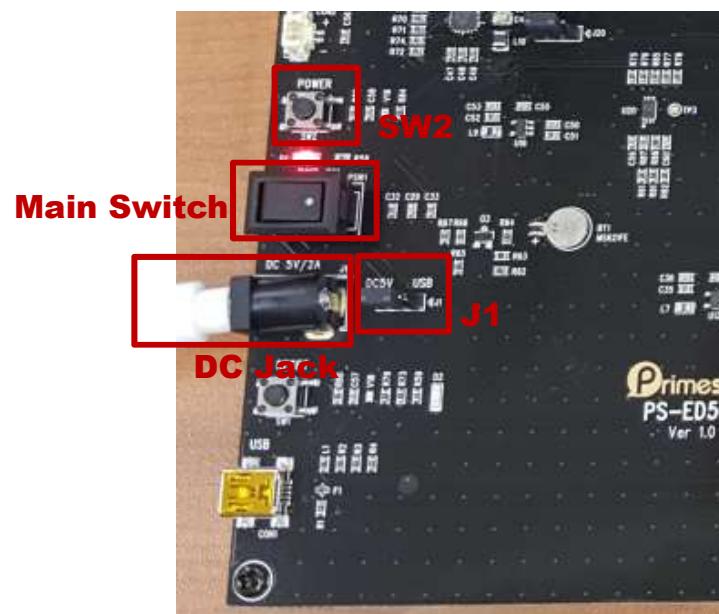
■ Configuration of PS-ED500 Board



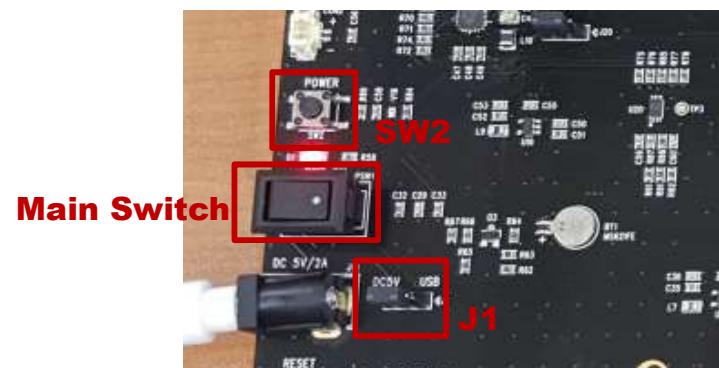
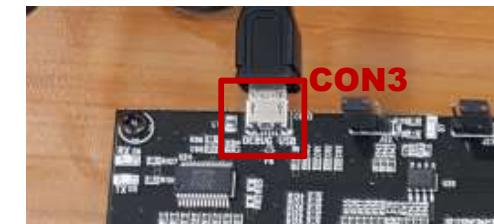
ARTIK Basic : PS-ED500

■ Power : DC 5V or USB

- DC 5V connection
 - ✓ J1 : DC5V selection
 - ✓ DC 5V plug
 - ✓ Main Switch ON
 - ✓ SW2 : power switch enable



- USB connection
 - ✓ J1 : USB selection
 - ✓ CON3 : USB Debug port or, CON2 : mini USB port
 - ✓ Main Switch ON
 - ✓ SW2 : power switch enable



ARTIK Basic : PS-ED500

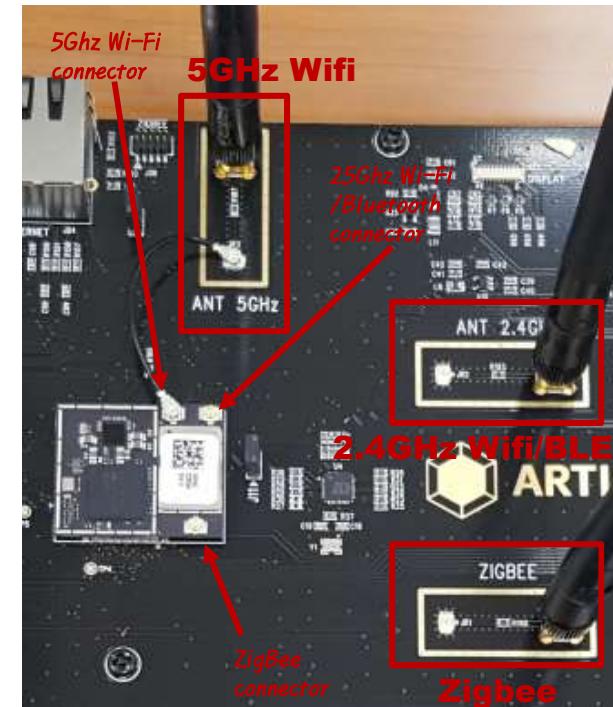
■ Debug USB

- Debug USB connection
 - ✓ CON3 : USB Debug port



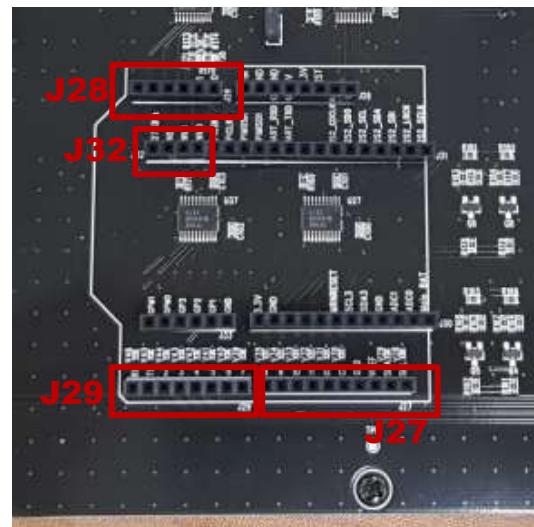
■ Antenna

- Wifi/BLE/Zigbee Antenna



■ I/O port

- Analog : J28
- PWM : J32
- GPIO : J29, J27



▪ Pin Map

Header J29 (GPIO)	
J29[Rx-0]	Rx
J29[Tx-1]	Rx
J29[2]	GPIO121
J29[3]	GPIO122
J29[4]	GPIO123
J29[5]	XPWM00
J29[6]	XPWM01
J29[7]	GPIO124

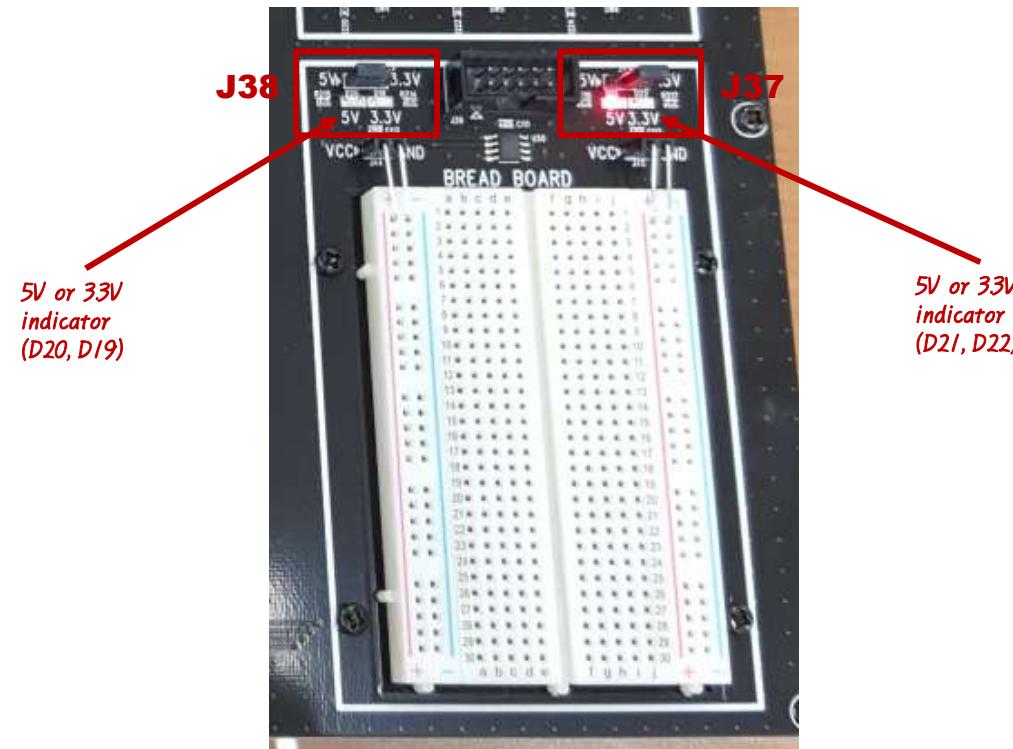
Header J28 (analog)	
J28[A0]	Analog 0
J28[A1]	Analog 1

Header J32 (PWM)	
J32[PWM0]	PWM 0
J32[PWM1]	PWM 1

Header J27 (GPIO)	
J27[8]	Rx
J27[9]	Rx
J27[10]	GPIO121
J27[11]	GPIO122
J27[12]	GPIO123
J27[13]	XPWM00
J27[GND]	XPWM01

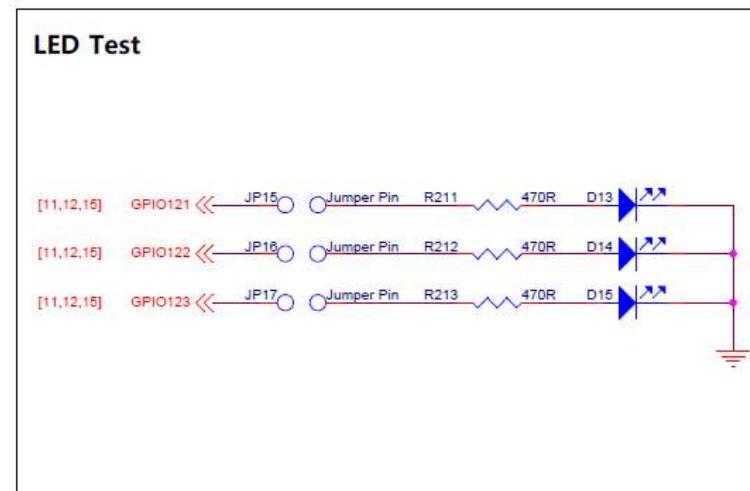
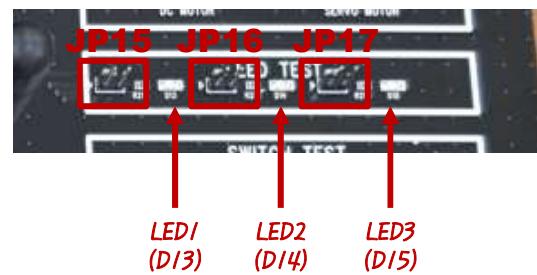
■ Breadboard & Power

- 5V or 3.3V selection
- Left side : J38 jumper selection
- Right side : J37 jumper selection
- Power indicator : D20(left 5V), D19(left 3.3V), D21(right 5V), D22(right 3.3V)



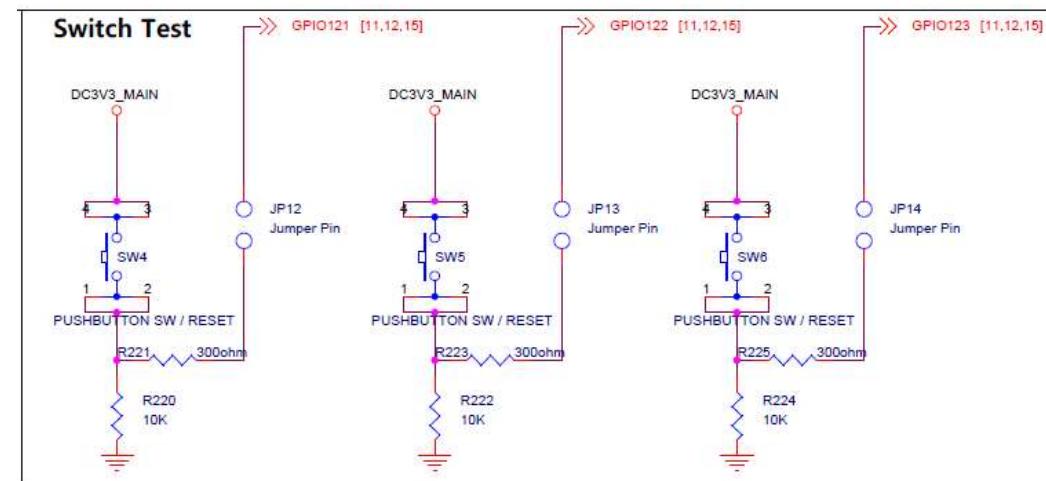
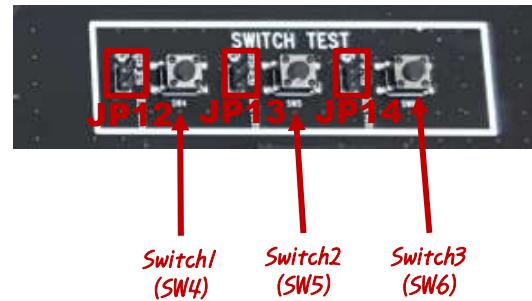
■ LED Test

- 3 LEDs : GPIO 121/122/123
- JP15 : LED1 enable (GPIO121)
- JP16 : LED2 enable (GPIO122)
- JP17 : LED3 enable (GPIO123)



■ Switch Test

- 3 Switches : GPIO 121/122/123
- JP12 : Switch1 enable (GPIO121)
- JP13 : Switch2 enable (GPIO122)
- JP14 : Switch3 enable (GPIO123)



■ Motor Test

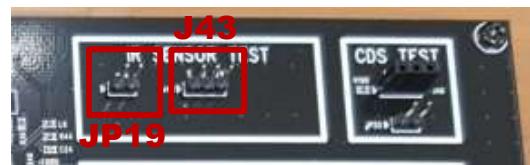
- DC/Servo Motor : PWM0
- JP18 : PWM0 enable
- J41 : Power(5V/GND)
- J42 : 3pin PWM connector (PWM0/5V/GND)



ARTIK Basic : PS-ED500

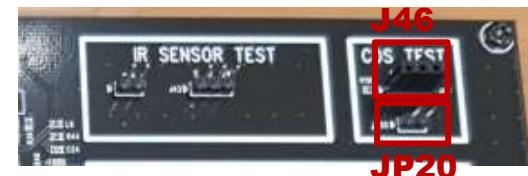
■ IR Sensor Test

- IR Sensor : Analog0
- JP19 : Analog0 enable
- J43 : 3pin Analog connector (A0,GND,5V)



■ CDS Test

- CDS Sensor : Analog0
- JP20 : Analog0 enable
- J46 : 3pin Analog connector (A0,not use,GND)



ARTIK Basic : PS-ED500

■ Jumper Pin 초기 위치

- J11, J19, J20, J25 Jumper Pin 제거 및 이동 금지



- J1 Jumper Pin : DC5V와 USB 중 선택
- SW3 : 아래로 내린 상태 고정



ARTIK 5 developer kit

■ Configuration of beta developer kit

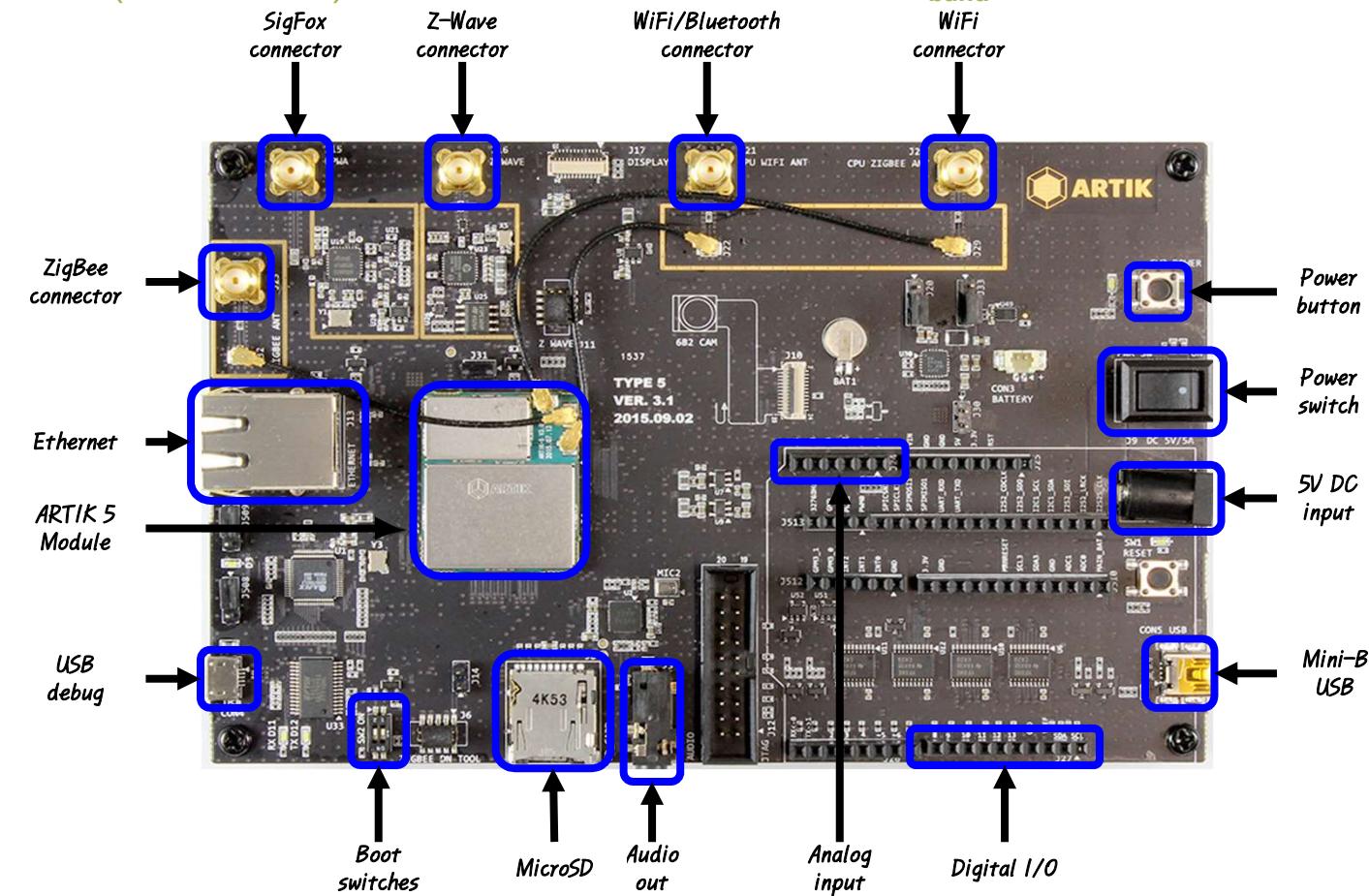
Similar with Cellular communication

Use 900MHz wide range ,

5 million connections

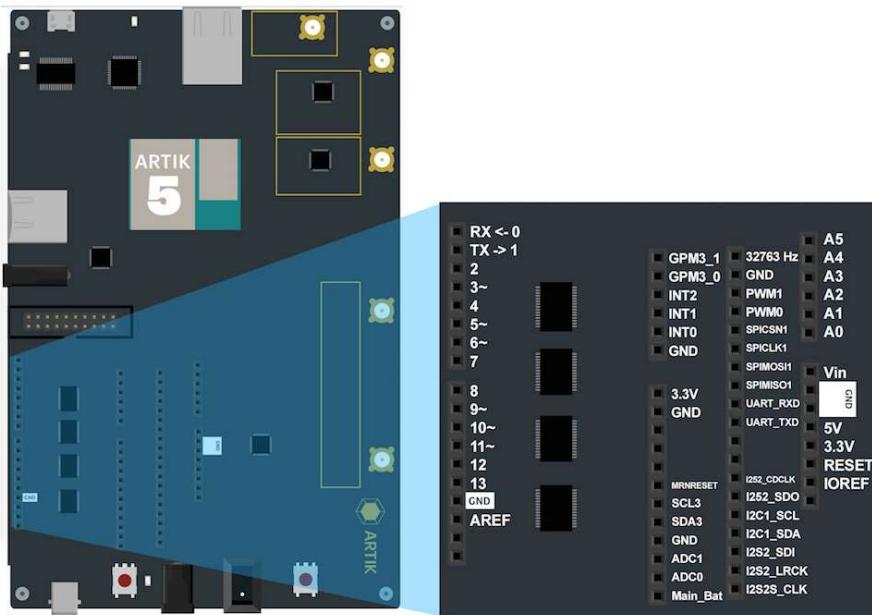
small data, narrow band

LPWAN(Low Power WAN)



ARTIK 5 developer kit

■ Pin mapping of beta developer kit



Header J24 (analog inputs)

PIN [SILKSCREEN]	MAPPING
J24[A0]	Analog input 0
J24[A1]	Analog input 1

Header J26

PIN [SILKSCREEN]	MAPPING
J26[Rx-0]	Rx
J26[Tx-1]	Tx
J26[2]	GPIO 121 (GPX0[0])
J26[3]	GPIO 122 (GPX0[1])
J26[4]	GPIO 123 (GPX0[2])
J26[5]	XPWM00
J26[6]	XPWM01
J26[7]	GPIO 124 (GPX0[3])

Header J27

PIN [SILKSCREEN]	MAPPING
J27[8]	GPIO 125 (GPX0[4])
J27[9]	GPIO 126 (GPX0[5])
J27[10]	GPIO 127 (GPX0[6])
J27[11]	GPIO 129 (GPX1[0])
J27[12]	GPIO 134 (GPX1[5])
J27[13]	GPIO 135 (GPX1[6])
J27[GND]	GND
J27[Vref]	VextIN
J27[SDA]	I2C7 SDA
J27[SCL]	I2C7 SCL



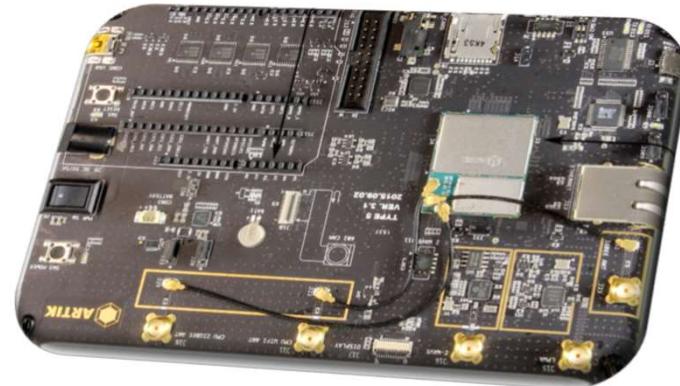
Note: The XPWM pins are not active yet.



Note: The I2C pins are not active yet.

I. ARTIK 소개

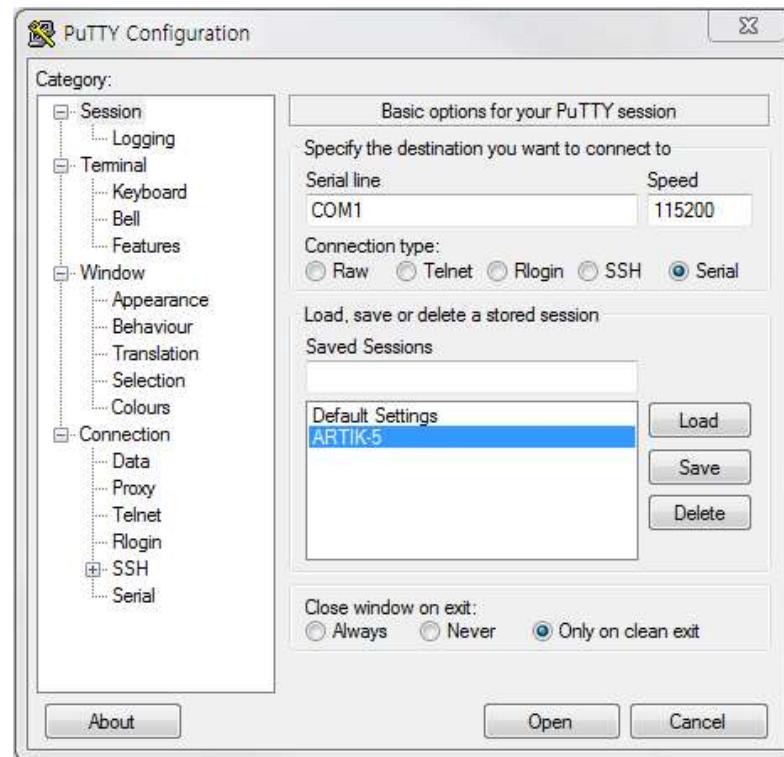
3. Development environment - Linux terminal



Setting up a development environment

■ Access to the ARTIK 5 beta developer kit with PuTTY

- Connection type → Set to 'Serial'
- COM port number check at Device Manager, write to 'Serial line'
- Set the speed to 115200
- Click 'Open'



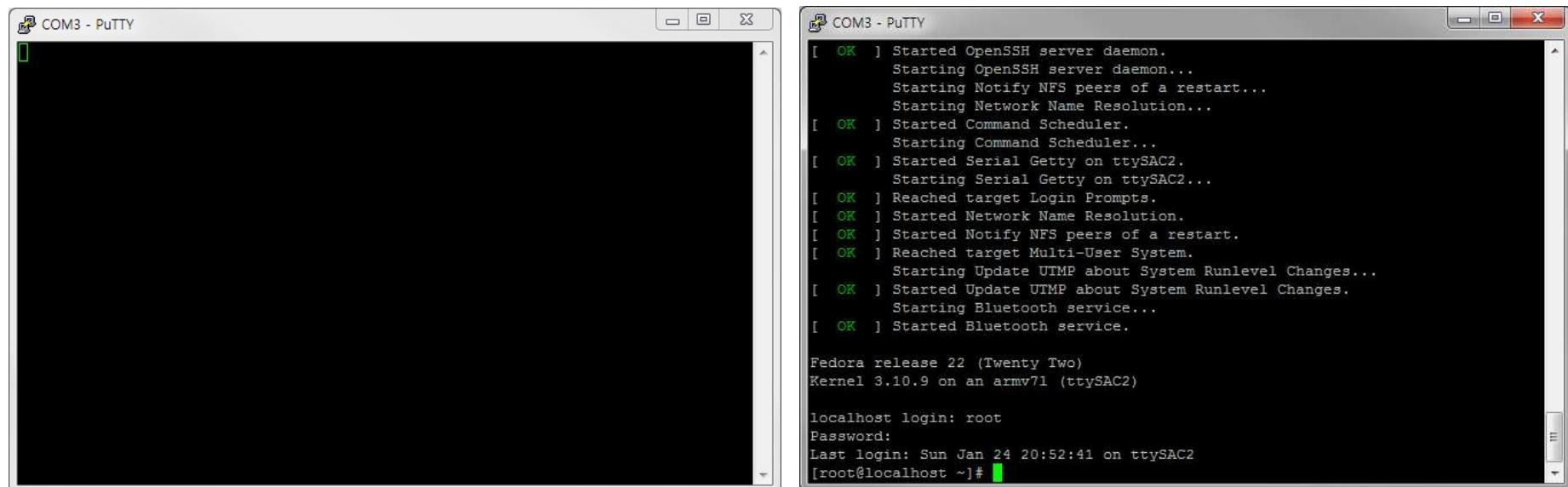
※ Putty

Free and open-source terminal emulator, serial console and network file transfer application.

Setting up a development environment

■ Access to the ARTIK 5 beta developer kit with PuTTY

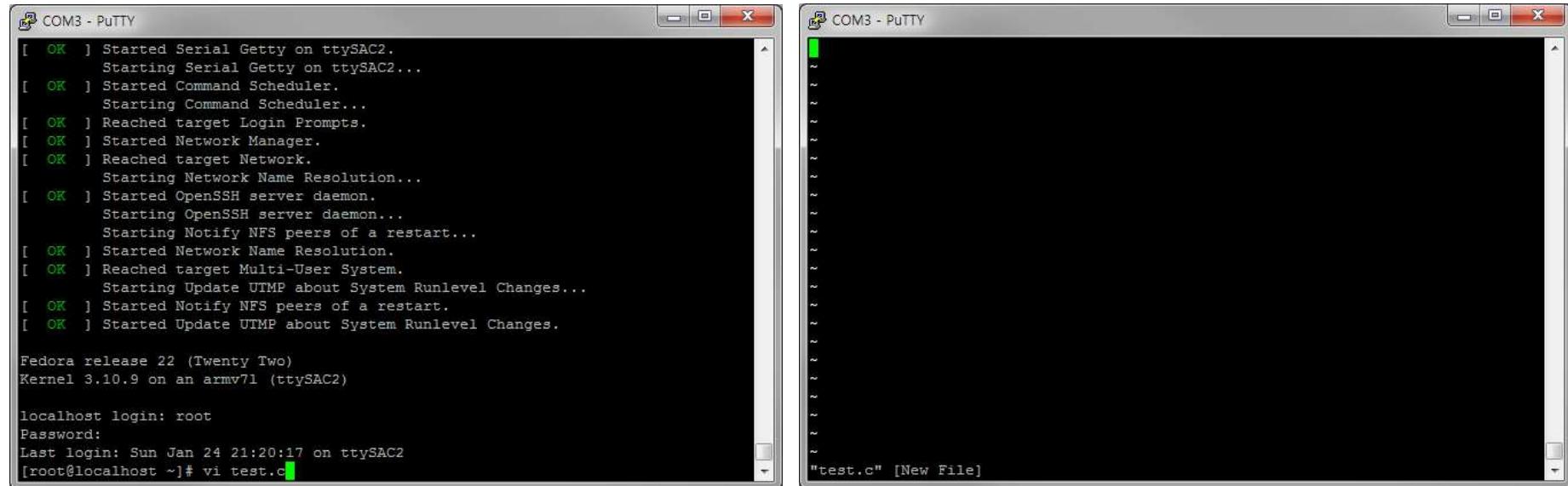
- Screen for initial connection with PuTTY(left picture)
- Press SW3 POWER button to boot(right picture)
 - Login : root
 - Password : root



Setting up a development environment

■ Access to the ARTIK 5 beta developer kit with PuTTY

- Create and edit files by using vi editor or nano editor (left picture)
- Write code(right picture)
- vi editor or nano editor usage referring to the appendix of Part 1.6 for usage.



Setting up a development environment

■ Access to the ARTIK 5 beta developer kit with PuTTY

- Save and exit code(Shift+Z+Z)
- Input `# gcc -o [filename] [source code name]`
 - Create an executable file by the gcc compiler
- Input `# ./[filename]` and execute the executable file
 - Pause: Ctrl + c

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>

#define HIGH 1
#define LOW 0
#define INPUT 1
#define OUTPUT 0

int outputPin = 135;

bool digitalPinMode(int pin, int dir){
    FILE * fd;
    char fName[128];

    // Exporting the pin to be used
    if(( fd = fopen("/sys/class/gpio/export", "w")) == NULL) {
        printf("Error: unable to export pin\n");
        return false;
    }
    fprintf(fd, "%d\n", pin);
    fclose(fd);
```

```
#include <stdbool.h>

#define HIGH 1
#define LOW 0
#define INPUT 1
#define OUTPUT 0

int outputPin = 135;

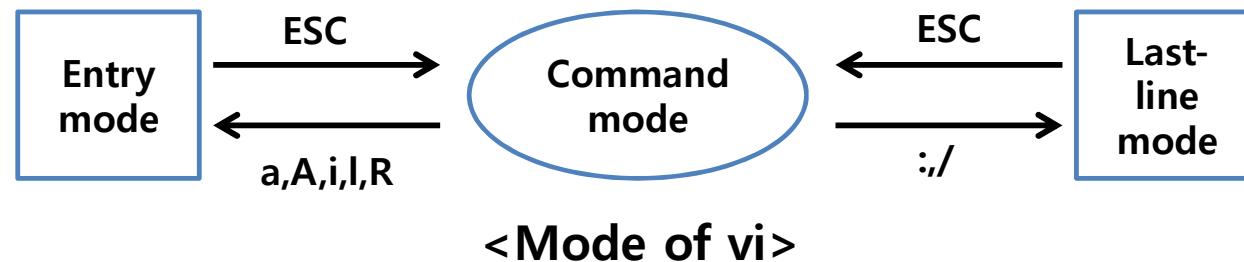
bool digitalPinMode(int pin, int dir){
    FILE * fd;
    char fName[128];

    // Exporting the pin to be used
    if(( fd = fopen("/sys/class/gpio/export", "w")) == NULL) {
        printf("Error: unable to export pin\n");
        return false;
    }
    fprintf(fd, "%d\n", pin);
    fclose(fd);

"test.c" 81L, 1517C written
[root@localhost ~]#
[root@localhost ~]# gcc -o test test.c
```

Text editor - vi

■ Shortcuts of Entry mode



	명령어	설명
Entry mode	i	insert before cursor
	a	append after cursor
	I	insert text at beginning of current line
	A	append text to end of current line
	R	replace characters, starting with current cursor position

Text editor - vi

■ Shortcuts of Command mode

	명령어	설명
Command mode	Esc	Change command mode
	x	delete character
	dd	delete line
	dw	delete word
	D	delete the remainder of the line
	yy	copy the current line into the buffer
	p	put the line after the current line
	u	Undo the last command
	Ctrl + b	page up
	Ctrl + f	page down
	ZZ	save changes and exit editor

Text editor - vi

■ Shortcuts of Last-line mode

	명령어	설명
Last-line mode	:w	writes changes to file
	:q, :q!	quits edit session
	:wq, :wq!	writes changes to current file and quits edit session
	:set nu	Line numbering
	:14	Move cursor line 14
	:4y	Copy line 4
	:5d	Delete line 5
	:1,12d	Delete line 1~12
	:.,\$d	Delete cursor ~ last line
	:/word => n	Find word => n : Find next word
	:%s/word1/word2/g	Change all word1 to word2 at all file
	:2,4s/word1/word2/g	Change all word1 to word2 at line 2~4
	:r file	Insert file to the current cursor

Text editor - nano

■ Shortcuts of nano editor

	단축키	설명
Shortcuts of Nano editor	ctrl + x	Quit Y : Quit after save N : Quit (No Save) C : Cancel
	ctrl + w	Find word
	nano -m test_name	Mouse Activation
	nano -i test_name	Auto indent
	nano -A test_name	Smart home key

“Hello world” Example

■ Access to the ARTIK 5 beta developer kit with PuTTY

- Input *# mkdir [directory name]*
 - Create directory
- Input *# cd [directory name]*
 - Change directory
- Input *# vi [filename]*
 - vi editor excution

```
[root@localhost ~]# mkdir test
[root@localhost ~]# cd test
[root@localhost test]# vi hello.c
```

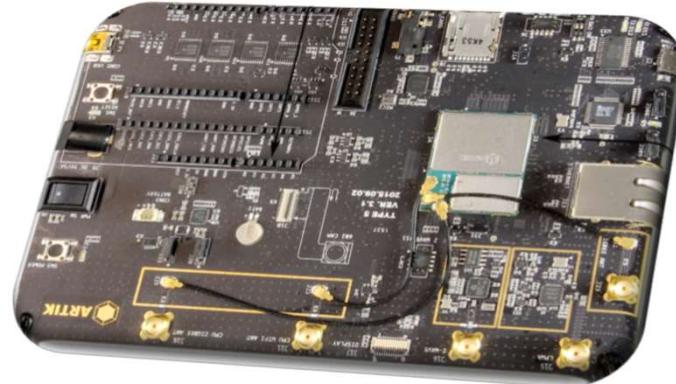
```
#include <stdio.h>

void main() {
    printf("hello world\n");□
}

"hello.c" [New] 5L, 62C written
[root@localhost test]# gcc -o hello hello.c
[root@localhost test]# ls
hello  hello.c
[root@localhost test]# ./hello
hello world
[root@localhost test]# □
```

I. ARTIK 소개

3. Development environment - Windows Network Drive



■ Introduction to SAMBA

- SAMBA is a free software which allows users to access a Linux or UNIX server from a PC running a Windows OS and share files and printers.
- When you are working through the terminal emulator(Putty), it is possible to just use vi editor to save code for compiling.
- If you use SAMBA, you probably need to have Wi-Fi to the Internet.

■ Initial Setup

- Step 1 : Connect ARTIK to your PC Wi-Fi
 - Attach the Wi-Fi antenna to ARTIK developer board.
 - List Wi-Fi access point available. `# wpa_cli scan_results or # iwlist wlan0 scan`
 - Configure the wpa_supplicant.conf file to write your Wi-Fi router SSID and password into the file, using the wpa_passphrase command.
`# cd /etc/wpa_supplicant`
`# wpa_passphrase "SSID" "PASSWORD" >> wpa_supplicant.conf`
 - Use VI editor to clean up the file contents.
`# vi wpa_supplicant.conf`
 - a. For security, delete your personal passphrase(the line that has # in front of it.)
 - b. Check whether the file contains multiple network login assignments.
 - C. Make sure that the first two lines of the file look like this.
`ctrl_interface=/var/run/wpa_supplicant`
`ctrl_interface_group=wheel`
 - Reboot and use the DHCP client services program to request a WLAN IP address.
`# dhclient wlan0`
 - Check whether IP address was assigned.
`# ifconfig wlan0`
 - Check if your device can connect to the Internet
`# ping XXX.XXX.XXX.XXX`

```
[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
              ether fe:0f:72:ff:fed5:188f  txqueuelen 1000  (Ethernet)
                  RX packets 51 bytes 5225 (5.1 KiB)
                  RX errors 0 dropped 10 overruns 0 frame 0
                  TX packets 30 bytes 3864 (3.7 KiB)
                  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

■ Initial Setup

- Step 2 : Install SAMBA

- *#dnf install samba*

```
[root@localhost /]# dnf install samba
RPM Fusion for Fedora 22 - Free - Updates      37 kB/s | 165 kB   00:04
Fedora 22 - armhf                  4.2 MB/s | 37 MB   00:08
Fedora 22 - armhf - Updates        4.6 MB/s | 20 MB   00:04
RPM Fusion for Fedora 22 - Free      55 kB/s | 389 kB   00:07
Last metadata expiration check performed 0:00:04 ago on Fri Oct 14 00:11:30 2016
.
Dependencies resolved.

=====
          Package           Arch         Version       Repository      Size
=====
Installing:
  libldb            armv7hl     1.1.24-1.fc22    updates       116 k
  libwbclient       armv7hl     2:4.2.12-1.fc22   updates       92 k
  python-talloc     armv7hl     2.1.5-2.fc22    updates       18 k
  samba            armv7hl     2:4.2.12-1.fc22   updates      583 k
  samba-client-libs armv7hl   2:4.2.12-1.fc22   updates      3.7 M
  samba-common      noarch     2:4.2.12-1.fc22   updates      162 k
  samba-common-libs armv7hl   2:4.2.12-1.fc22   updates      148 k
  samba-common-tools armv7hl  2:4.2.12-1.fc22   updates      517 k
  samba-libs        armv7hl   2:4.2.12-1.fc22   updates      237 k
  systemd-compat-libs armv7hl 219-27.fc22      updates      125 k

Transaction Summary
=====
Install 10 Packages
```

■ Initial Setup

- Step 3 : Create a new directory

- `# cd /home`
 - `# mkdir samba`

- Step 4 : Modify the configuration file

- `# vi /etc/samba/smb.conf`

- ① Set workgroup same as PC

컴퓨터 이름, 도메인 및 작업 그룹 설정

컴퓨터 이름:

전체 컴퓨터 이름:

컴퓨터 설명:

작업 그룹:

WORKGROUP

```
# hosts deny = the hosts not allowed to connect. This option  
# is per-share basis.  
  
# max protocol = used to define the supported protocol. This  
# can set it to SMB2 if you want experimental SMB2 support  
  
#  
workgroup = WORKGROUP  
server string = Samba Server Version %v
```

- ② Configure directory, which will be shared with PC

```
[user]  
path = /home/samba  
read only = no  
writable = yes  
valid users = samba  
browseable = yes  
printable = no  
create mask = 0765
```

■ Initial Setup

- Step 5 : Add user and set samba password

- `# useradd _user_name_`
 - `# smbpasswd -a _user_name_`

```
[root@localhost [REDACTED]]# useradd samba
useradd: warning: the home directory already exists.
Not copying any file from skel directory into it.
[root@localhost [REDACTED]]# smbpasswd -a samba
New SMB password:
Retype new SMB password:
Added user samba.
```

- Step 6 : Add the following lines to /etc/sysconfig/iptables file.

- `# vi /etc/sysconfig/iptables`
 - `-A INPUT -p udp -m udp --dport 137 -j ACCEPT`
 - `-A INPUT -p udp -m udp --dport 138 -j ACCEPT`
 - `-A INPUT -m state --state NEW -m tcp -p tcp --dport 139 -j ACCEPT`
 - `-A INPUT -m state --state NEW -m tcp -p tcp --dport 445 -j ACCEPT`

■ Initial Setup

```
# sample configuration for iptables service
# you can edit this manually or use system-config-firewall
# please do not ask us to add additional ports/services to this default configuration
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -p udp -m udp --dport 137 -j ACCEPT
-A INPUT -p udp -m udp --dport 138 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 139 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 445 -j ACCEPT

-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
~
"/etc/sysconfig/iptables" 21L, 775C
```

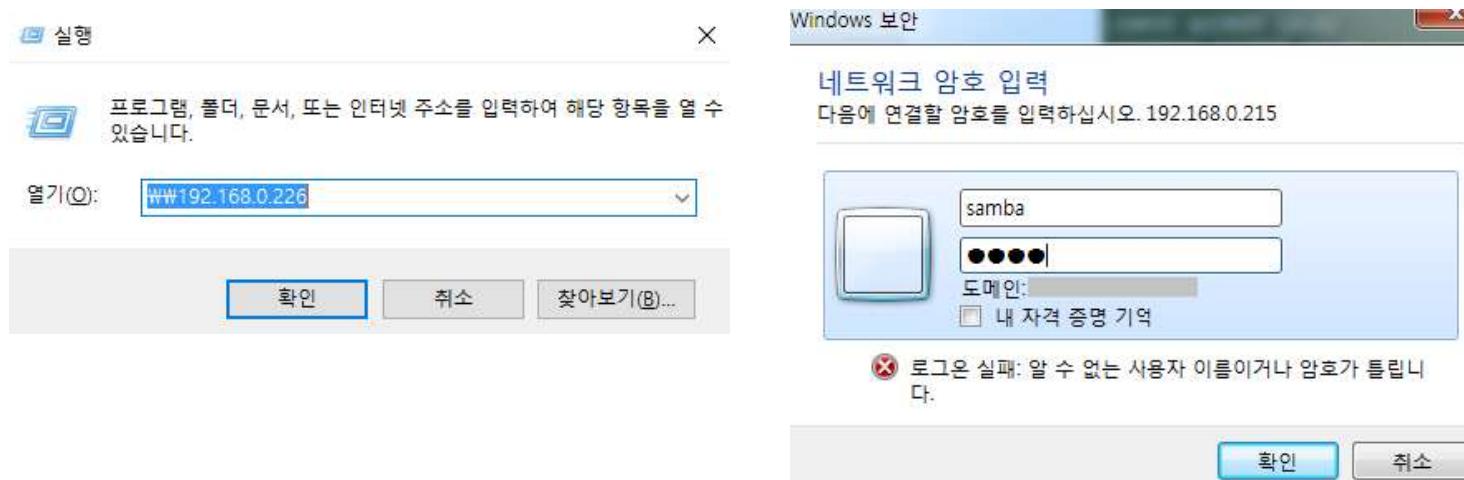
■ Run SAMBA

- Step 1 : Re-start iptables service and smb service

- *# service iptables restart*
 - *# service smb restart*

```
[root@localhost ~]# service iptables restart
Redirecting to /bin/systemctl restart iptables.service
[root@localhost ~]# service smb restart
Redirecting to /bin/systemctl restart smb.service
[root@localhost ~]#
```

- Step 2 : On your PC, access to *WWWxx.xx.xx.xx* (IP address of ARTIK)
 - Then, you should enter username and samba password.



SAMBA

■ Change the owner of the directory(or file)

- *# Chown [-R] [user][:group] directory1 [directory2...]*

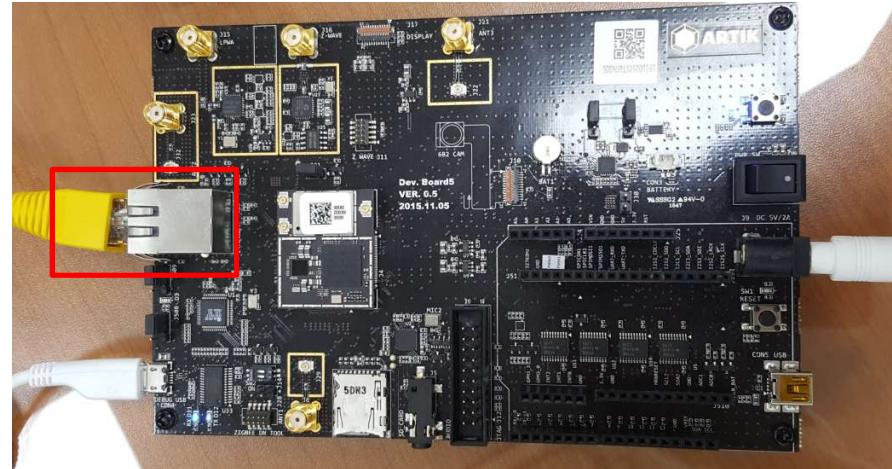
- Option -R : changes the owner of all directories and the files contained within them.

```
[root@localhost home]# ls -al
total 24
drwxr-xr-x  6 root      root      4096 Feb 17 02:27 .
drwxr-xr-x 21 root      root      4096 Jan  9 00:02 ..
drwxr-xr-x  2 root      root      4096 Feb 17 02:16 samba
drwxr-xr-x  2 root      root      4096 Feb 17 02:27 shared
drwx-----  2 skku     skku      4096 Feb 17 01:58 skku
```

```
[root@localhost home]# chown samba:samba samba
[root@localhost home]# ls -al
total 24
drwxr-xr-x  6 root      root      4096 Feb 17 02:27 .
drwxr-xr-x 21 root      root      4096 Jan  9 00:02 ..
drwxr-xr-x  2 samba    samba    4096 Feb 17 02:16 samba
drwxr-xr-x  2 root      root      4096 Feb 17 02:27 shared
drwx-----  2 skku     skku      4096 Feb 17 01:58 skku
```

■ Connect to PC with using crossover cable

- Step 1 : Connect ARTIK to PC, using crossover cable



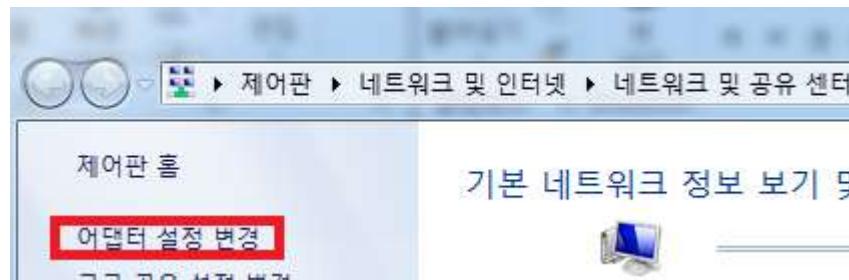
- Step 2 : Using the `ifconfig` command, change the IP address of ARTIK temporarily.

- `# ifconfig eth0 192.168.200.2`

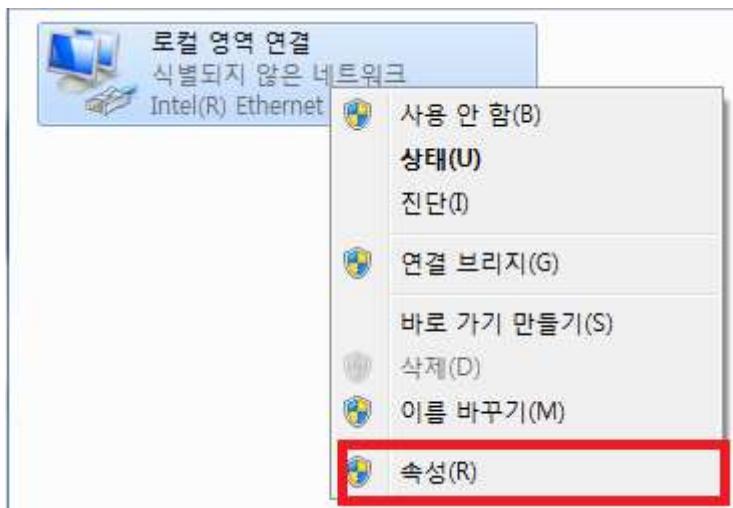
```
[root@localhost ~]# ifconfig eth0 192.168.200.2
[root@localhost ~]# ifconfig eth0
eth0: flags=-28605<UP,BROADCAST,RUNNING,MULTICAST,DYNAMIC> mtu 1500
        inet 192.168.200.2 netmask 255.255.255.0 broadcast 192.168.200.255
        inet6 fe80::80f3:70ff:fe1b:621e prefixlen 64 scopeid 0x20<link>
          ether 82:f3:70:1b:62:1e txqueuelen 1000 (Ethernet)
            RX packets 232 bytes 16793 (16.3 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 51 bytes 16891 (16.4 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
            device interrupt 32
```

■ Connect to PC with using crossover cable

- Step 3 : Go to Network and Sharing Center and click "Change adapter settings."

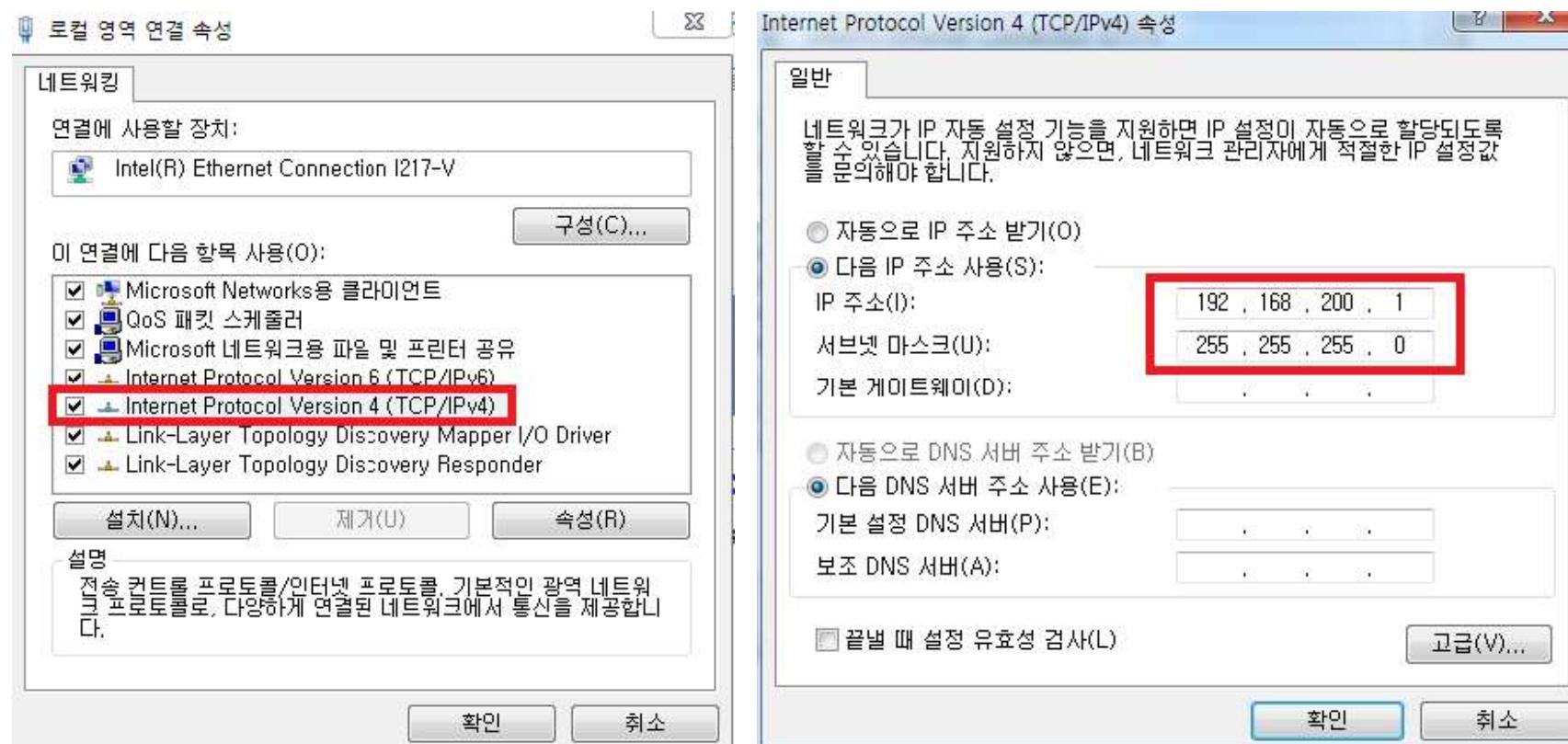


- Step 4 : Right-click on your local adapter and select Properties.



■ Connect to PC with using crossover cable

- Step 5 : Double-click Internet Protocol Version 4(TCP/IPv4). Then enter the following IP address and subnet mask.



■ Connect to PC with using crossover cable

- Step 6 : Start SAMBA and iptables, and access to `WW192.168.0`

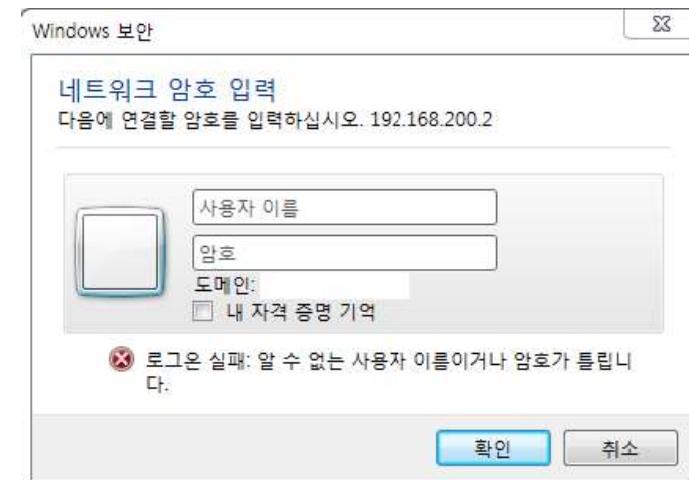
- `# service smb start`
 - `# service iptables start`

```
[root@localhost ~]# service smb start
Redirecting to /bin/systemctl start  smb.service
[root@localhost ~]# service iptables start
Redirecting to /bin/systemctl start  iptables.service
```

- Step 7 : On your PC, access to `WW192.168.200.2`

- If PC cannot access to Samba server, go to ARTIK and stop iptables.
 - `# systemctl stop iptables`

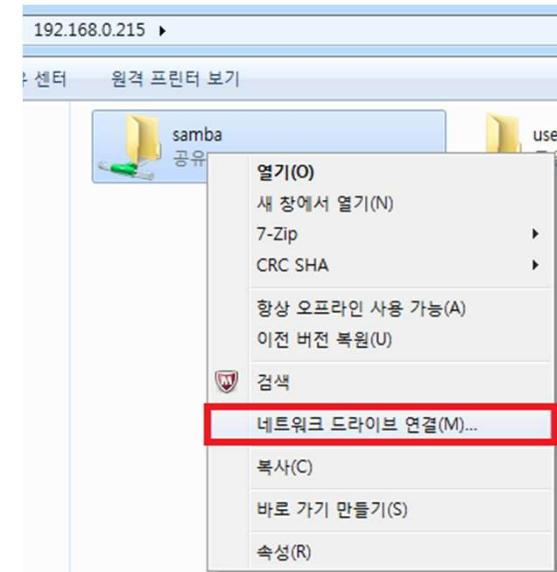
```
[root@localhost ~]# systemctl stop iptables
[root@localhost ~]#
```



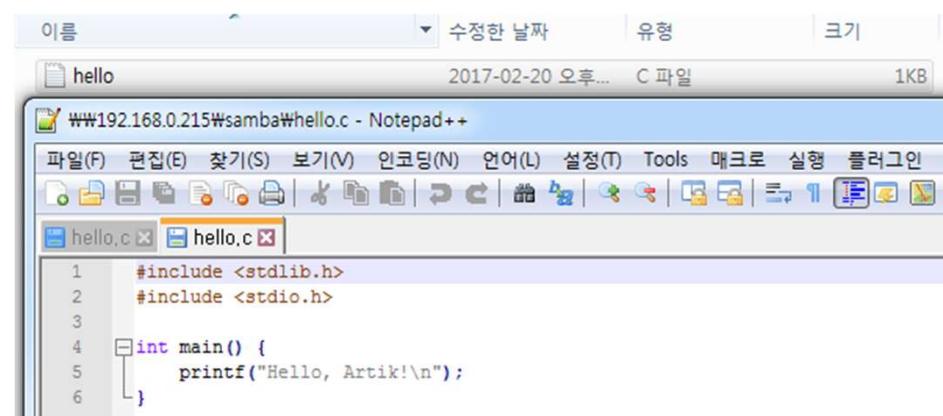
SAMBA – “Hello, World” Example

■ Tutorial

- Step 1 : Connect to a network drive



- Step 2 : Write code with PC and save to network drive



SAMBA – “Hello, World” Example

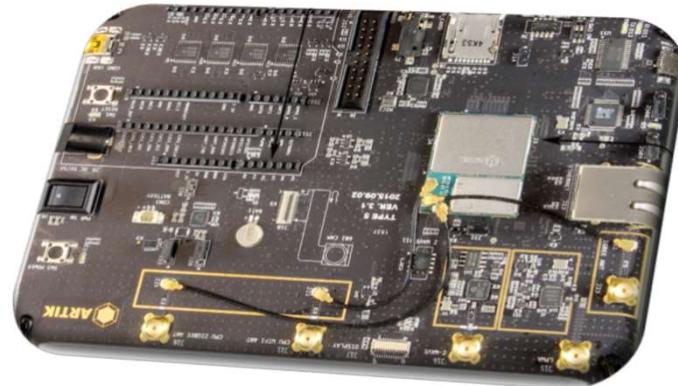
■ Execution result

- Input *# cd [directory name]*
 - Change directory
- Input *# gcc -o [filename] [source code name]*
 - Create an executable file by the gcc compiler
- Input *# ./[filename]* and execute the executable file
 - Pause: Ctrl + c

```
[root@localhost ~]# cd /home/samba
[root@localhost samba]# dir
hello.c
[root@localhost samba]# gcc -o hello hello.c
[root@localhost samba]# ./hello
Hello, Artik!
[root@localhost samba]#
```

I. ARTIK 소개

3. Development environment - ARTIK IDE



Set up the ARTIK IDE

■ Download ARTIK IDE

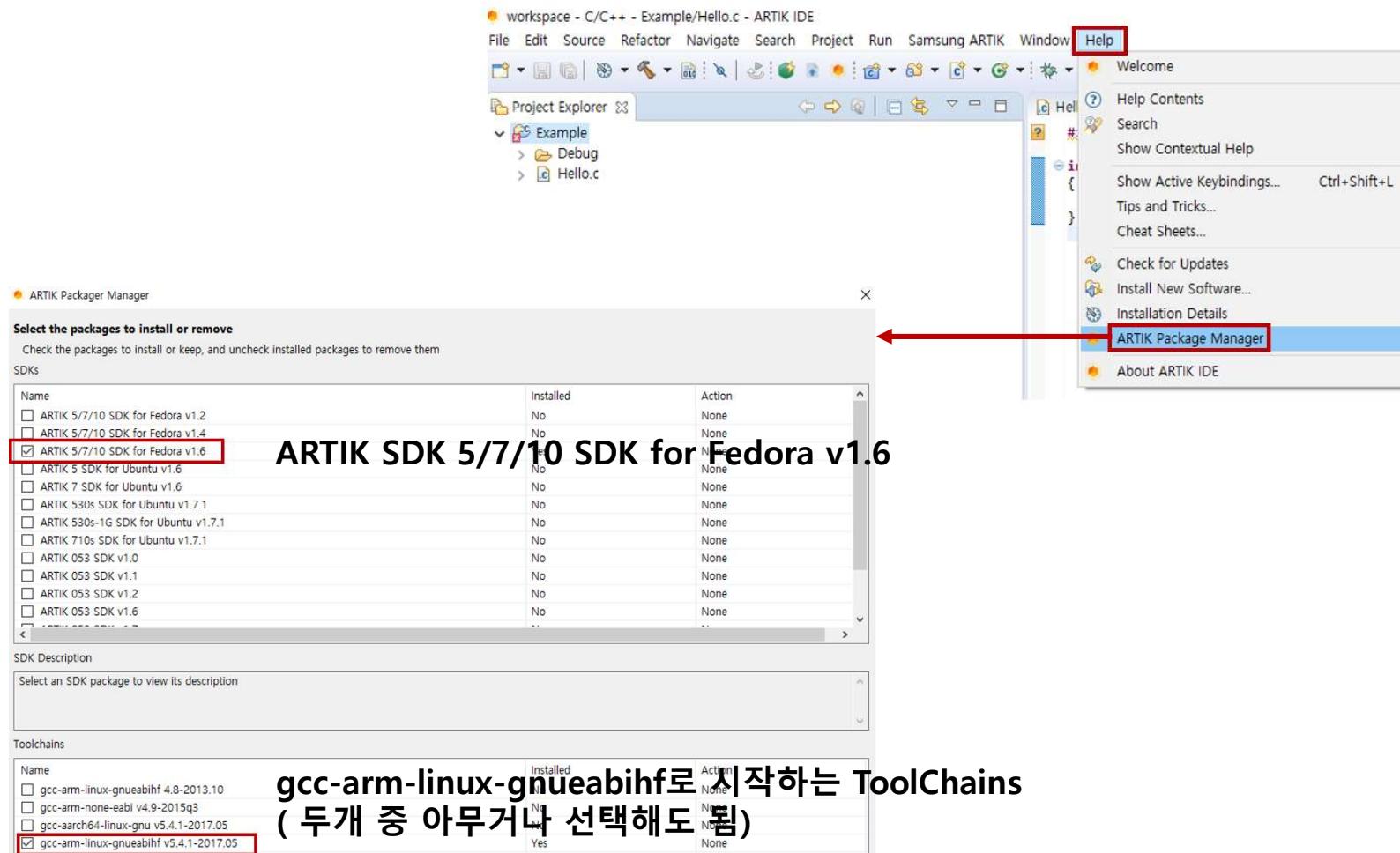
- <https://developer.artik.io/documentation/artik-05x/getting-started/prepare-ide.html>

Step 1. Click the tab for ARTIK IDE setup instructions for your OS.

The image shows two parts. On the left, a screenshot of a web page for downloading the ARTIK IDE. It has tabs for Windows, Linux, and Mac, with Windows selected. Below the tabs is a paragraph about installing the IDE on Windows 7, 8, and 10. A blue button labeled "Download Installer" is highlighted with a red rectangle. On the right, a screenshot of the ARTIK IDE software interface. The window title is "workspace - C/C++ - ARTIK IDE". The interface includes a toolbar, a menu bar with File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help, and a "No Launch Configurations" dropdown. Below the toolbar are tool buttons. The main workspace contains a "Project Explorer" view showing a folder icon, a "Problems" view showing 0 items, and a "Tasks" view showing 0 items. The status bar at the bottom left says "0 items selected".

Set up the ARTIK IDE

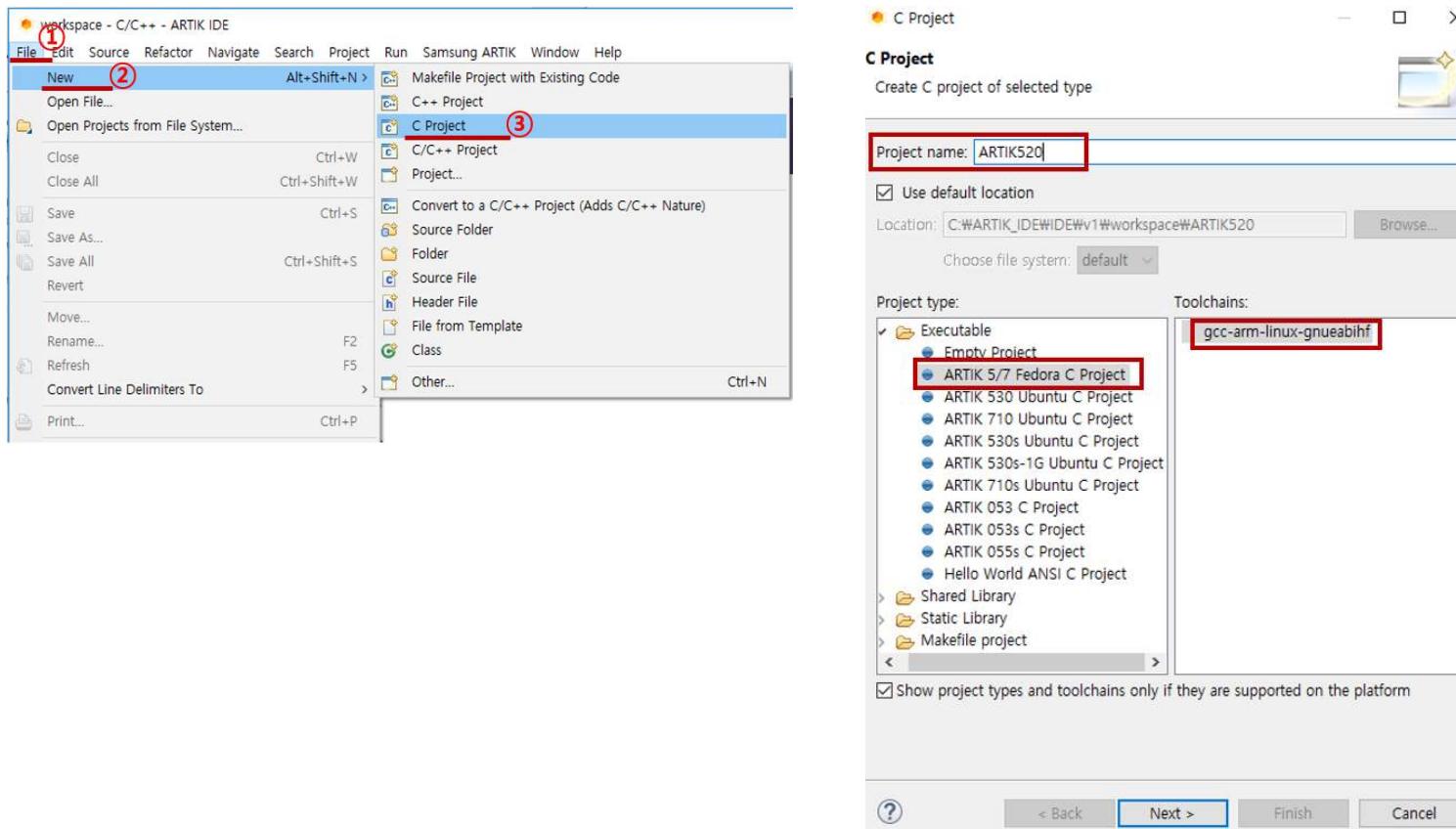
■ ARTIK Package Manager



ARTIK IDE Tutorial

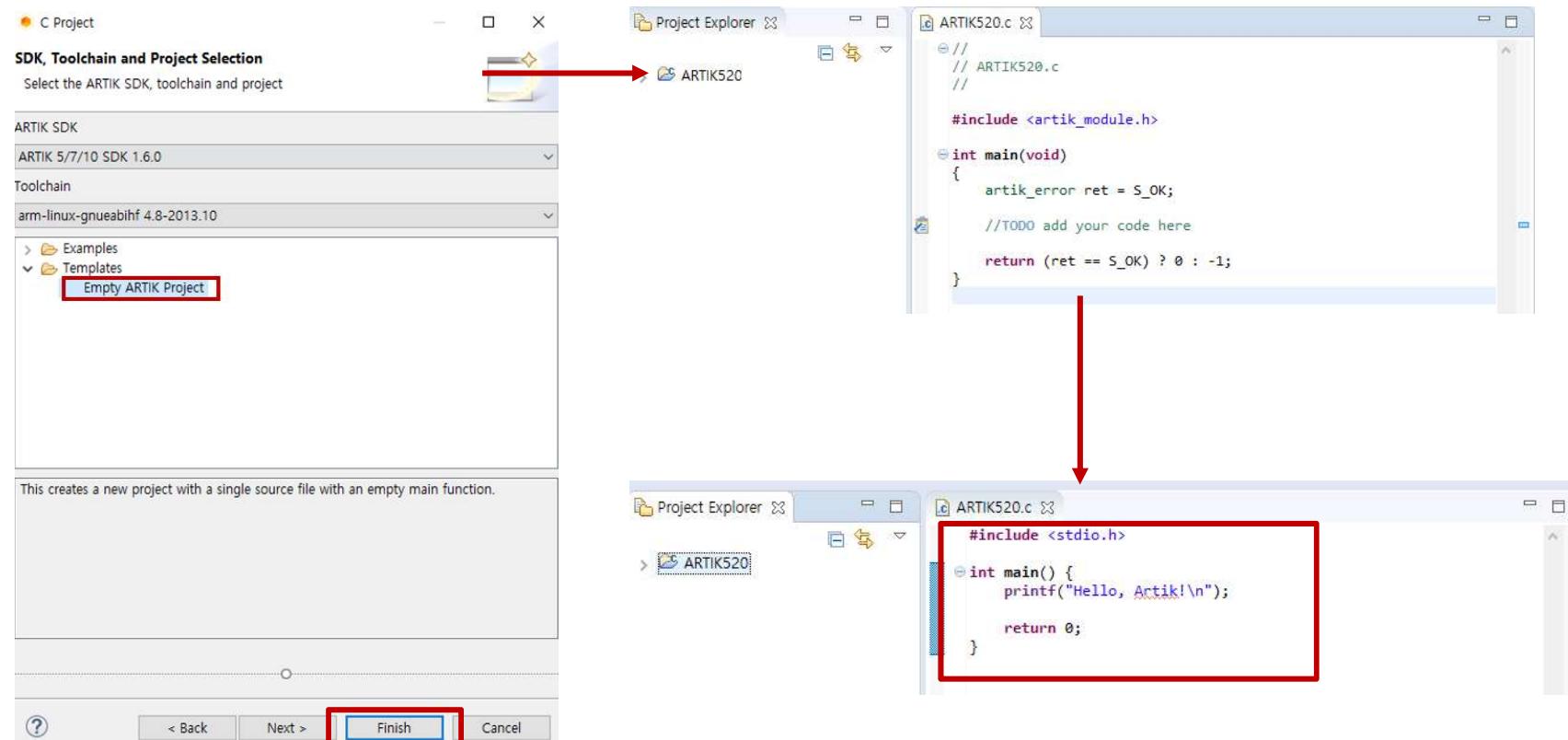
■ Create New C Project

- [File] – [New] – [C Project]
- Project Type: ARTIK 5/7 Fedora C Project
- ToolChains: gcc-arm-linux-gnueabihf



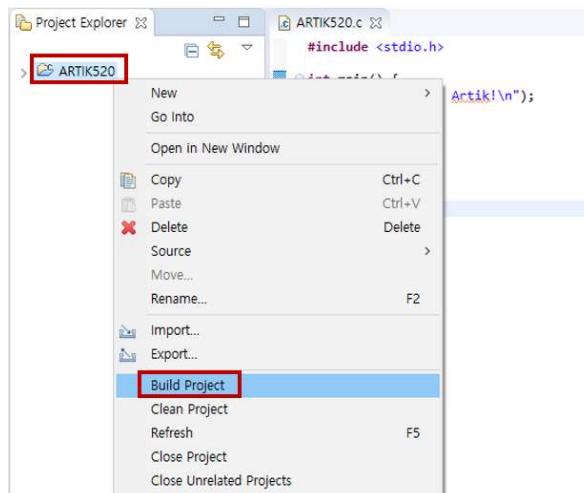
■ Create New C Project

- Empty ARTIK Project 선택 후 Finish
- 이외에도 [Examples]에서 C 예제 선택 가능

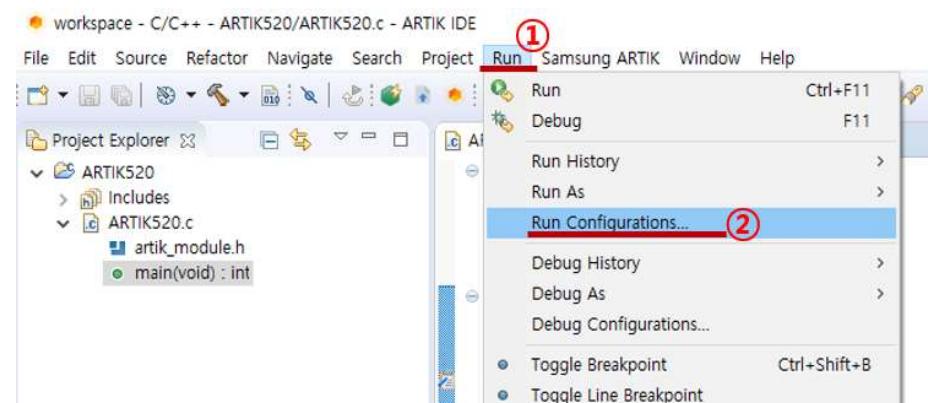


■ Build Project

- Run Configuration을 설정하기 위한 단계 (생략하면 안 됨)

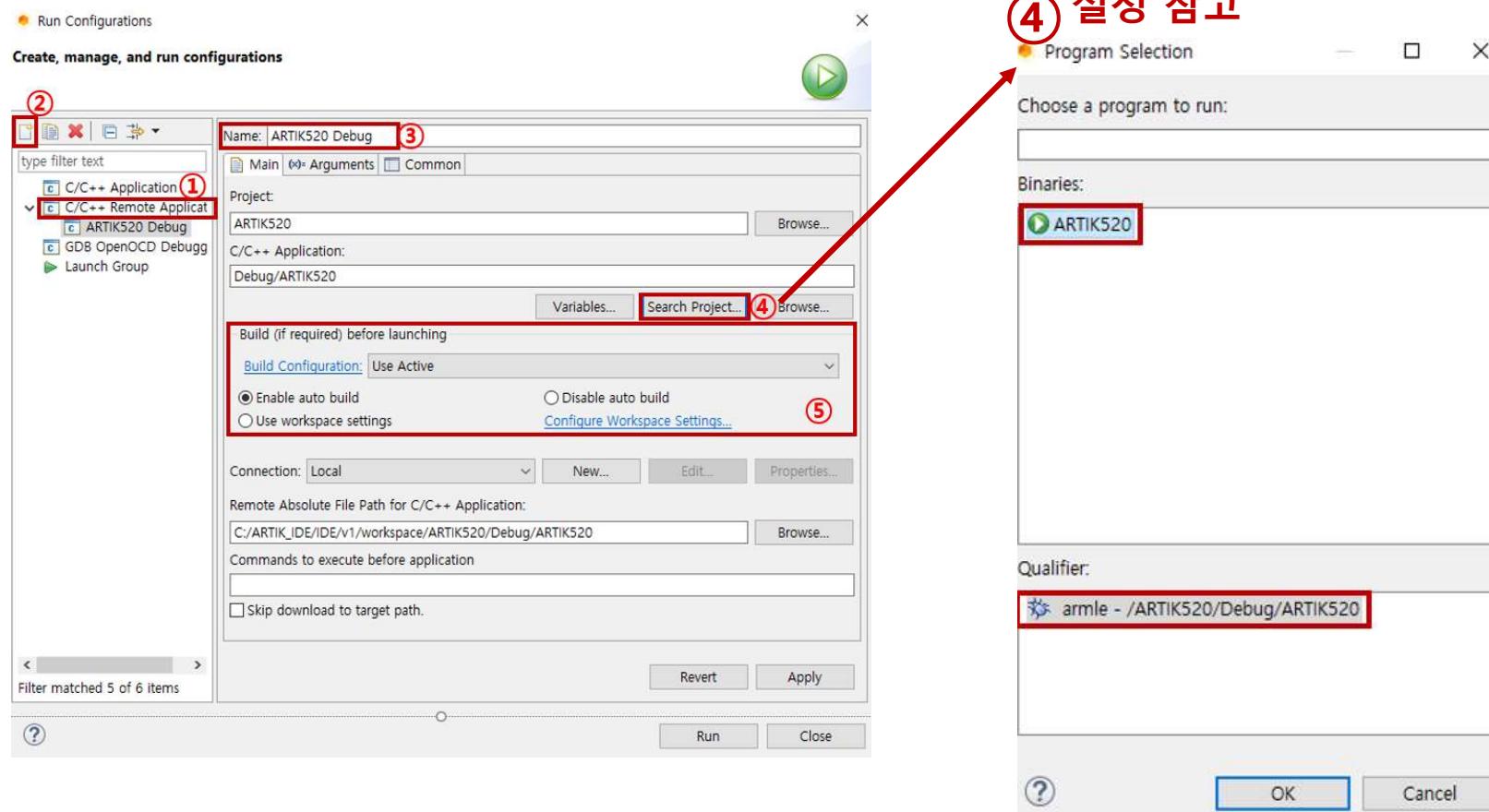


■ Run Configuration

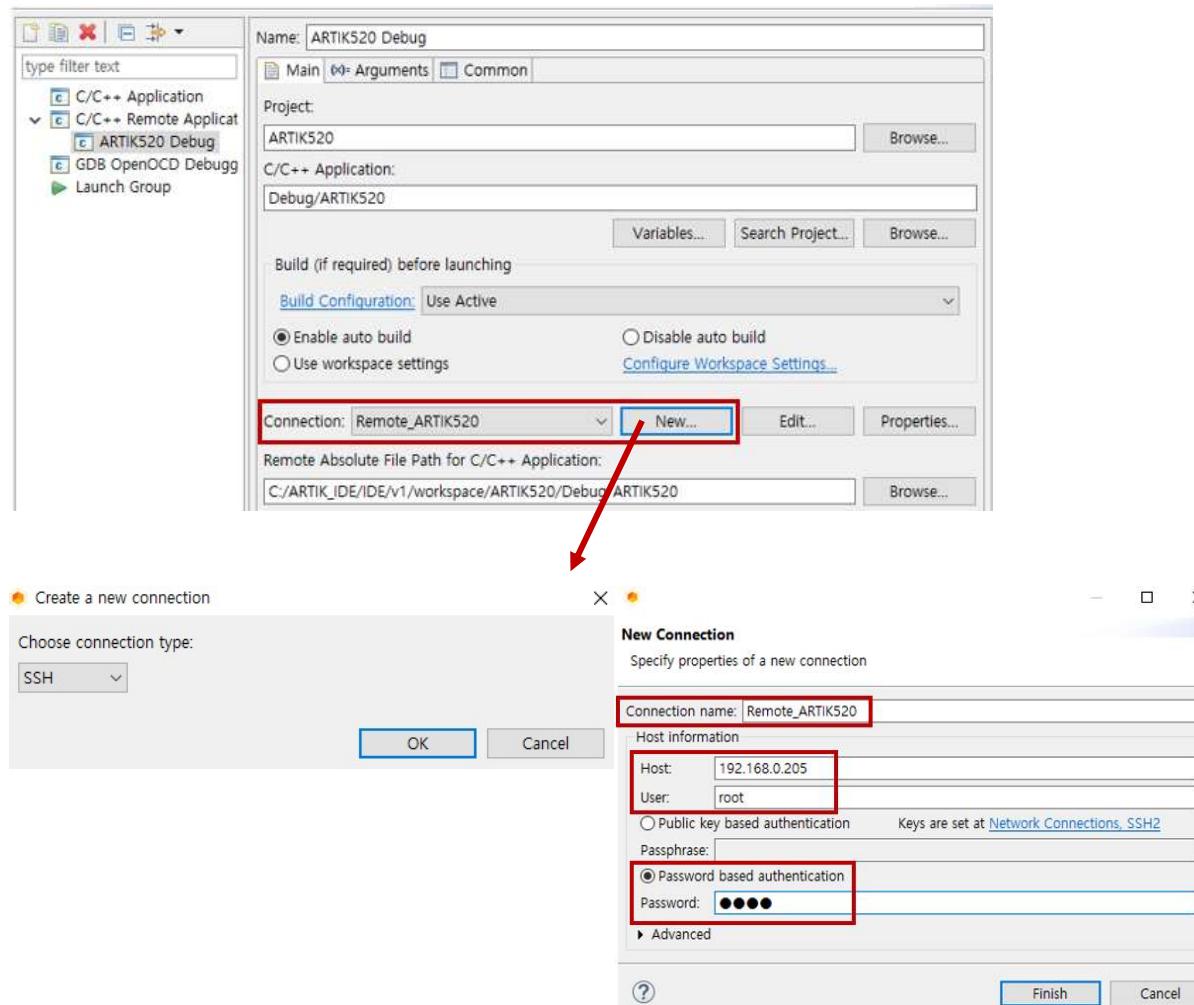


■ Run Configuration

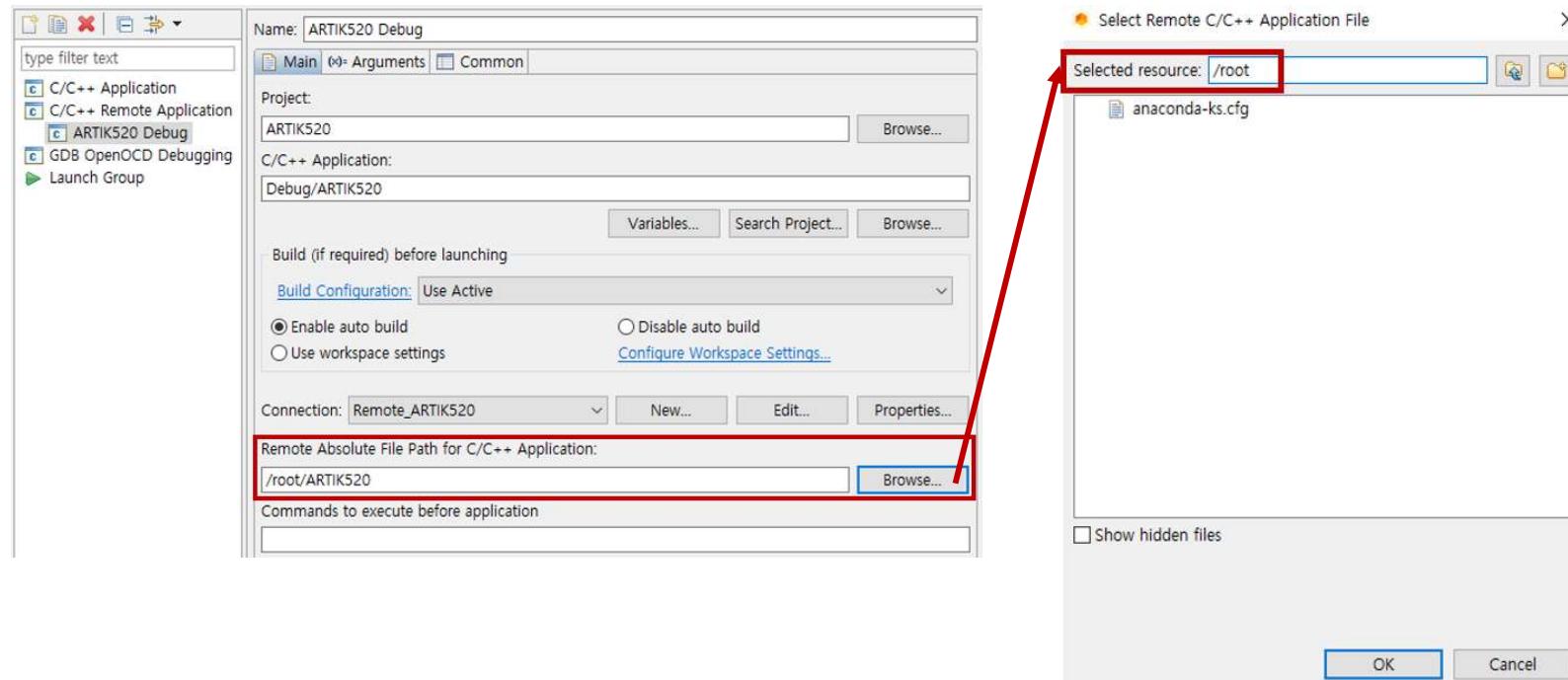
- Wi-Fi를 이용하여 실행 환경 설정
- ARTIK 모듈과 PC는 같은 네트워크(라우터) 상에 있어야 함



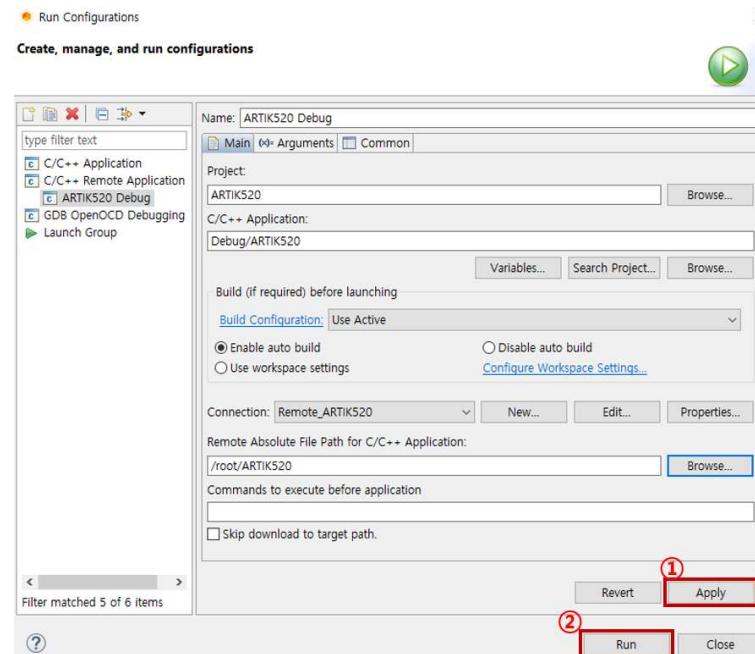
■ Run Configuration



■ Run Configuration



■ Run Configuration



■ View Console

The screenshot shows the 'Console' tab of the ARTIK IDE. The title bar includes 'Problems', 'Tasks', 'Console', 'Properties', and various icons. The console window displays the following terminal session:

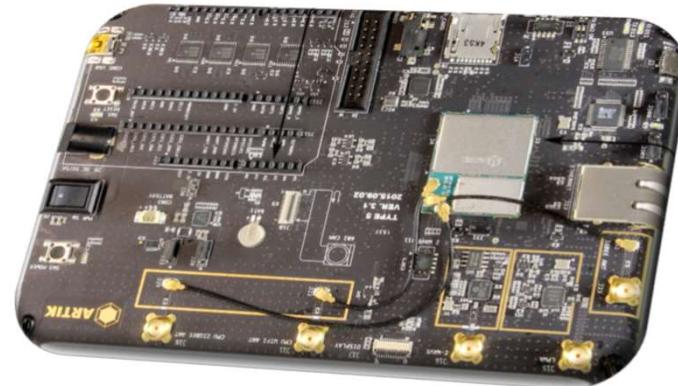
```
<terminated> ARTIK520 Debug [C/C++ Remote Application] C:\ARTIK_IDE\IDE\v1\workspace\ARTIK520\Debug\ARTIK520 (1
Last login: Mon Feb 26 02:17:24 2018 from

/root/ARTIK520;exit

[root@localhost ~]# /root/ARTIK520;exit
Hello, Artik!
logout
```

I. ARTIK 소개

3. Development environment - Arduino IDE



■ Introduction to Arduino IDE

- Arduino provides an integrated development environment with hundreds of downloadable libraries available for simplifying all sorts of tasks.
- With the Arduino IDE you need to have Wi-Fi to the Internet.
 - During initial setup, you run a curl command on ARTIK to copy the Arduino client side communications program from the internet.
 - During use, when transferring a compiled program from the host PC, Arduino typically uses Wi-Fi.
- Once you set up initially, you don't have to need Wi-Fi connection since it is possible to connect Arduino IDE to ARTIK over USB. But you must prepare mini type B USB connector.



■ Initial Setup

- Step 1 : Connect ARTIK to your PC Wi-Fi
 - Attach the Wi-Fi antenna to ARTIK developer board.
 - List Wi-Fi access point available. `# wpa_cli scan_results`
 - Configure the wpa_supplicant.conf file to write your Wi-Fi router SSID and password into the file, using the wpa_passphrase command.
`# cd /etc/wpa_supplicant`
`# wpa_passphrase "SSID" "PASSWORD" >> wpa_supplicant.conf`
 - Use VI editor to clean up the file contents.
`# vi wpa_supplicant.conf`
 - a. For security, delete your personal passphrase(the line that has # in front of it.)
 - b. Check whether the file contains multiple network login assignments.
 - C. Make sure that the first two lines of the file look like this.
`ctrl_interface=/var/run/wpa_supplicant`
`ctrl_interface_group=wheel`
 - Reboot and use the DHCP client services program to request a WLAN IP address.
`# dhclient wlan0`
 - Check whether IP address was assigned.
`# ifconfig wlan0`

```
[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
              ether ec:1f:72:d5:18:8f txqueuelen 1000 (Ethernet)
              RX packets 51 bytes 5225 (5.1 KiB)
              RX errors 0 dropped 10 overruns 0 frame 0
              TX packets 30 bytes 3864 (3.7 KiB)
              TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

■ Initial Setup

- Step 2 : Download and install the Arduino connectivity program
 - `# curl downloads.Arduino.cc/libArduino/install_artik_prereq.sh / sh`

```
# Completed on Thu Feb  9 01:33:36 2017

Done! You can now find your board in IDE, "Network ports" menu
If no board is listed, please follow the online guide on adding additional board
s
Add the following link and install "ARM Linux Boards" from board manager
http://downloads.arduino.cc/libArduino/package_arduino.cc_linux_index.json
Please use IDE version 1.6.6

[root@localhost ~]#
```

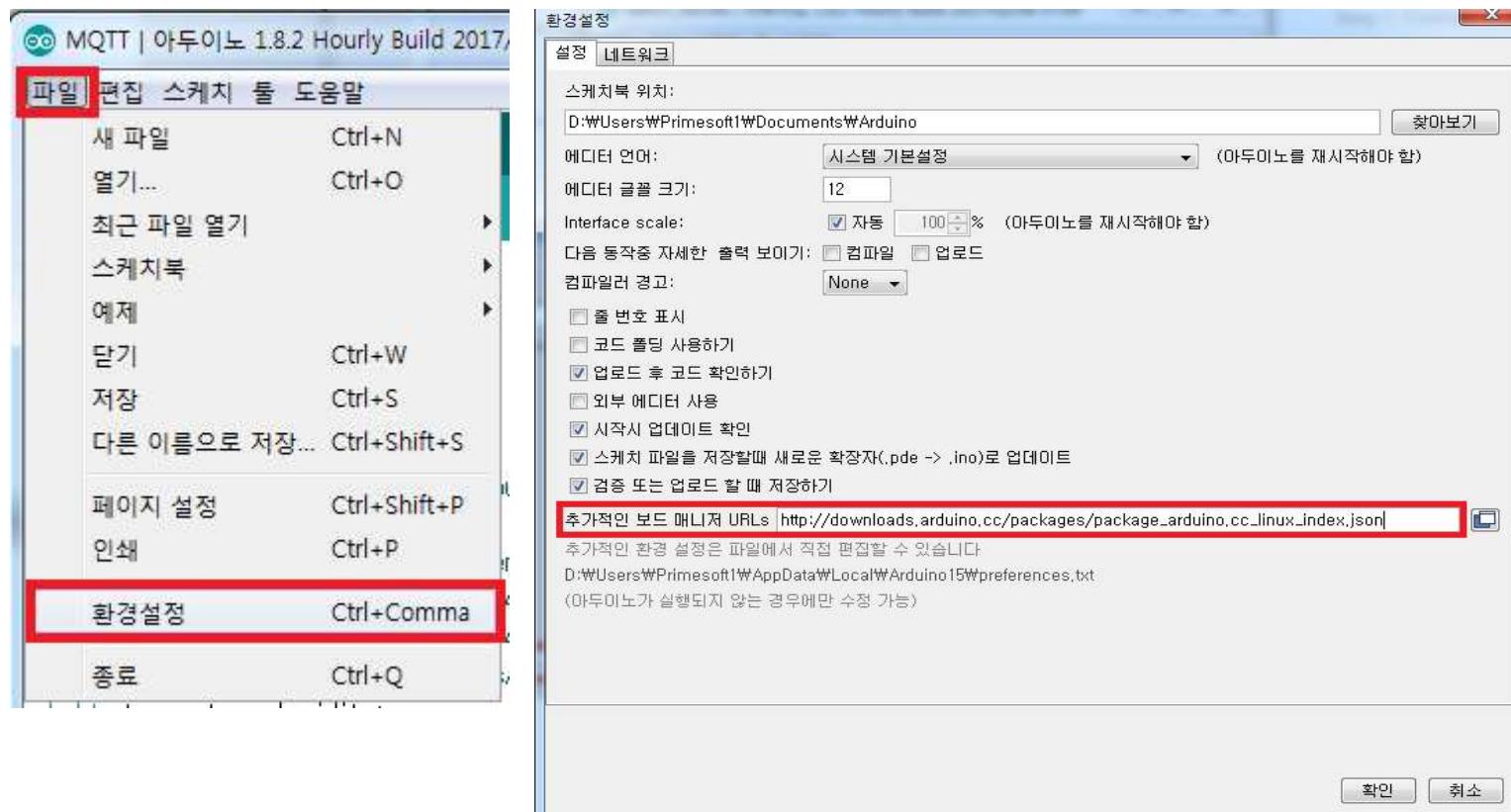
- Step 3 : Install Arduino IDE on your PC
 - Go to the Arduino Web site to download and install the IDE package.
 - <https://www.arduino.cc/en/Main/Software>



■ Initial Setup

▪ Step 4 : Update Boards Manager

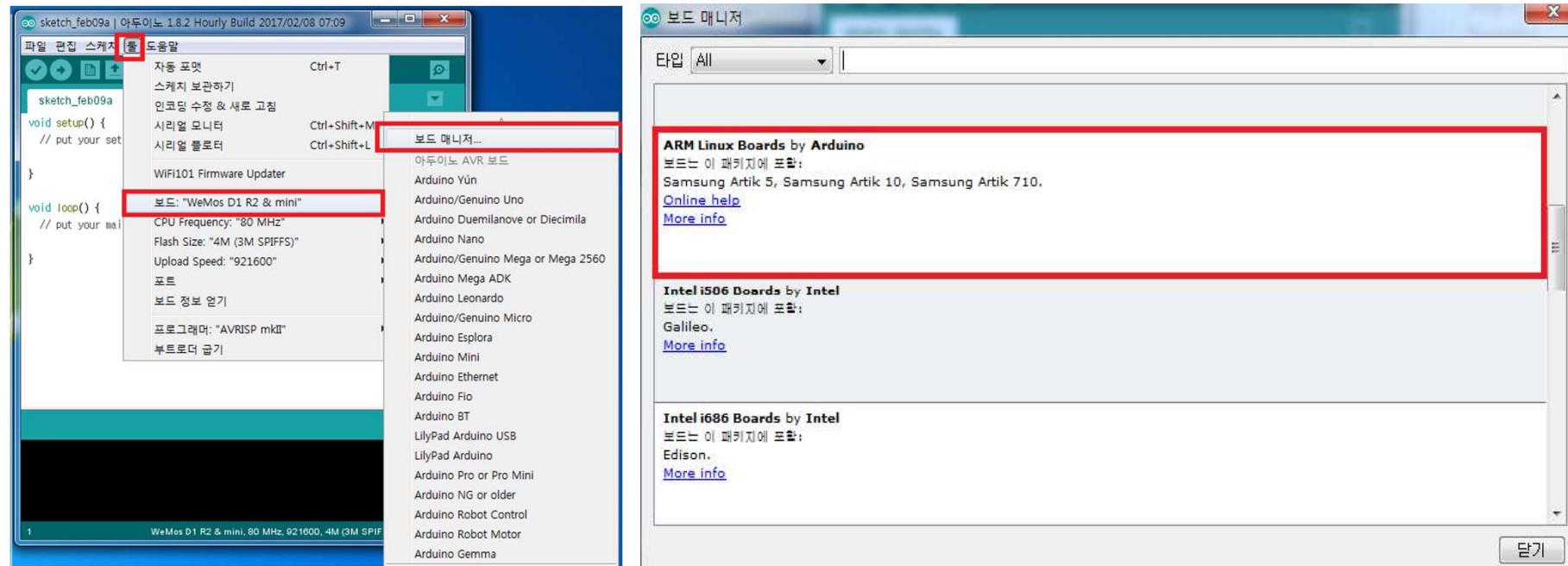
- Select [File]>[Preferences] to copy and paste this URL into the place indicated.
- http://downloads.arduino.cc/packages/package_arduino.cc_linux_index.json



■ Initial Setup

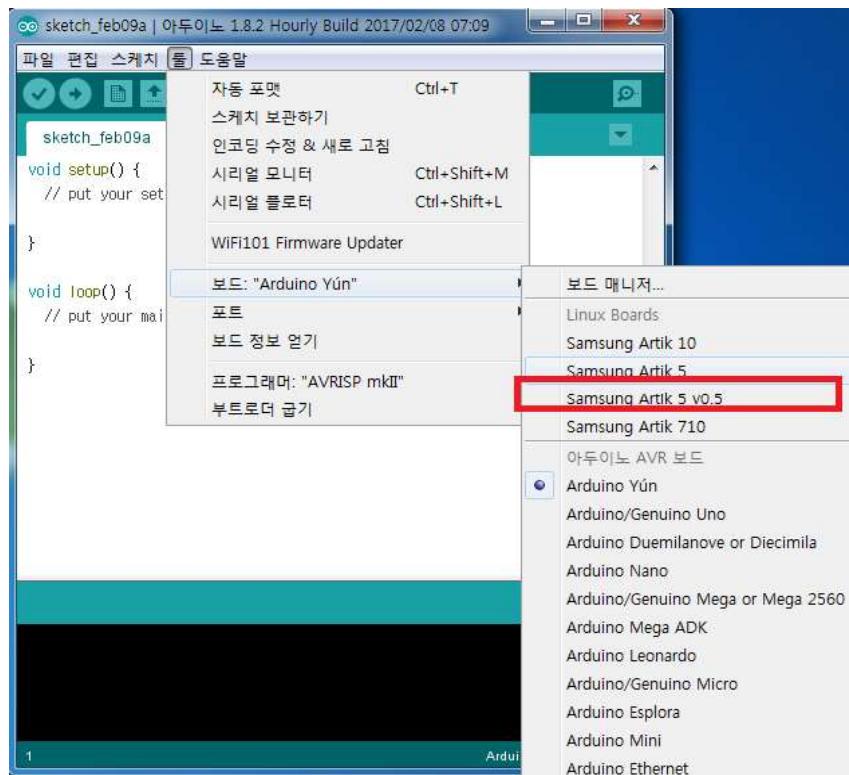
▪ Step 5 : Install ARM Linux Boards Package

- Select [Tools]>[Board]>[Boards Manager...]
- Install the latest ARM Linux Boards package which include Samsung ARTIK boards.



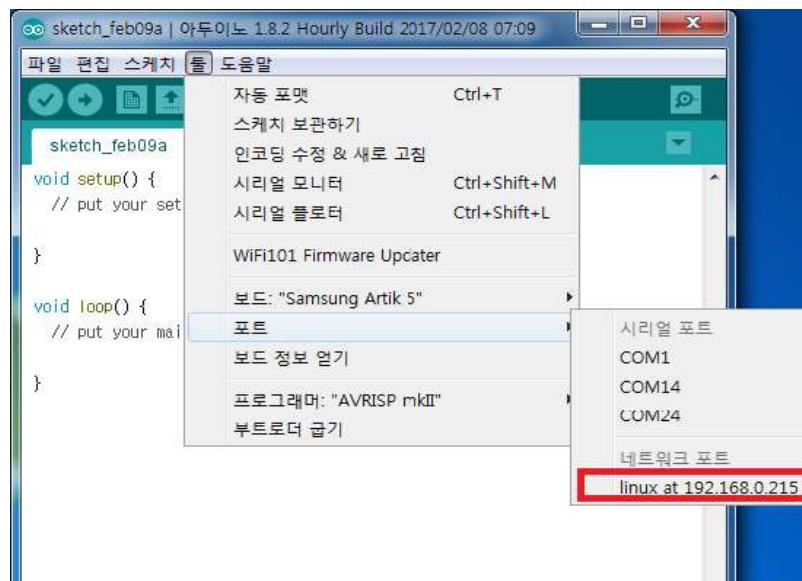
■ Initial Setup

- Step 6 : Select Boards in IDE
 - Under [Tools]>[Board] drop-down, find and select an "ARTIK 5 v0.5."



■ Connect Arduino IDE to ARTIK with Wi-Fi

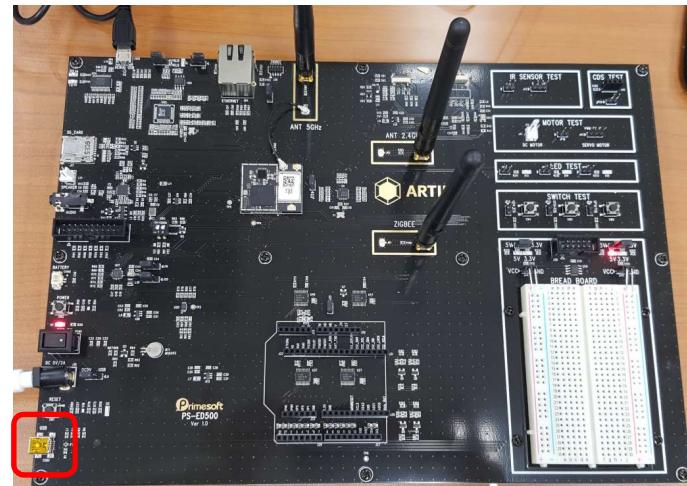
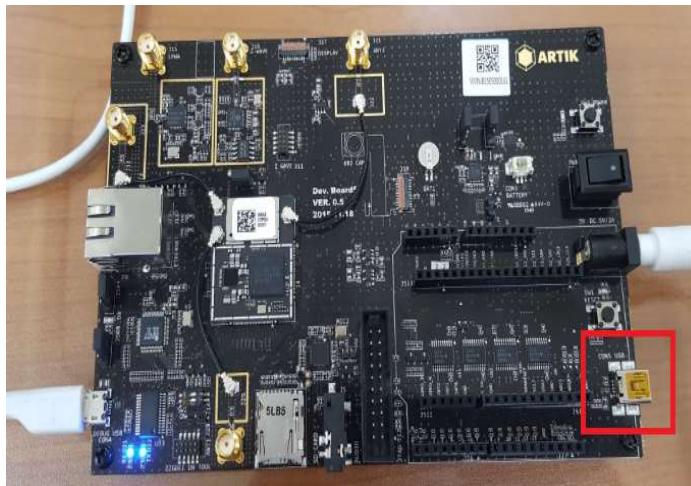
- Step 1 : Start ARTIK and connect to Wi-Fi.
 - # dhclient wlan0
 - # ifconfig wlan0
- Step 2 : Select port in Arduino IDE.
 - Under [Tools]>[Port], select "linux at xx.xx.xx.xx (Samsung ARTIK 5)"
 - When uploading your source code to ARTIK, you should enter password of ARTIK.



Arduino IDE

■ Connect Arduino IDE to ARTIK over USB

- Step 1 : Attach a USB cable to the integrated USB device port.
 - Not Debug USB port
 - Use the appropriate cable. For ARTIK 520, you need a mini type B USB cable.



- Step 2 : Run the script /root/watcher

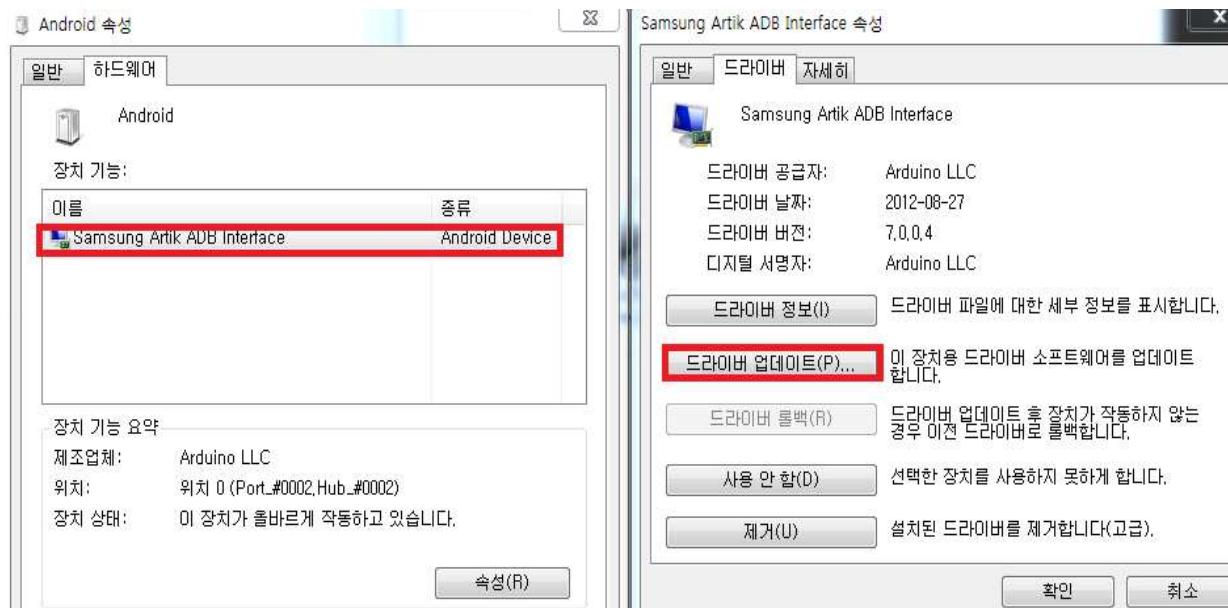
- # ./watcher

```
[root@localhost ~]# ./watcher
[    76.153320] android_usb: already disabled
--> Press Ctrl+C to force start, Ctrl+\ to exit.
```

■ Connect Arduino IDE to ARTIK over USB

▪ Step 3 : Run Device Manager on your PC

- Select "Android" device and you should see "[Samsung Artik ADB Interface](#)".
- If it says "[Samsung Artik Composite ADB Interface](#)", then it is the **wrong** device type to update.
- Then, you should reset the board, and run again ./watcher.
- Double-click on "Samsung Artik ADB Interface" and select "Update Driver Software..."



■ Connect Arduino IDE to ARTIK over USB

▪ Step 4 : Update Driver Software

- Click "Browse my computer for driver software".
- Click "Let me pick from a list of device drivers on my computer".



드라이버 소프트웨어는 어떻게 검색합니까?



▶ 업데이트된 드라이버 소프트웨어 자동으로 검색(S)
컴퓨터와 인터넷에서 장치에 대한 최신 드라이버 소프트웨어를 검색합니다. 이렇게 하지 않으면 장치 설치 설정에서 이 기능을 사용하지 않도록 설정해야 합니다.

▶ 컴퓨터에서 드라이버 소프트웨어 찾아보기(R)
수동으로 드라이버 소프트웨어를 찾아 설치하십시오.

▶ 컴퓨터의 장치 드라이버 목록에서 직접 선택(L)
이 목록에는 장치와 호환되는 설치 드라이버 소프트웨어 및 동일한 범주에서 모든 드라이버 소프트웨어를 표시합니다.

다음 위치에서 드라이버 소프트웨어 검색:

하위 폴더 포함(I)

■ Connect Arduino IDE to ARTIK over USB

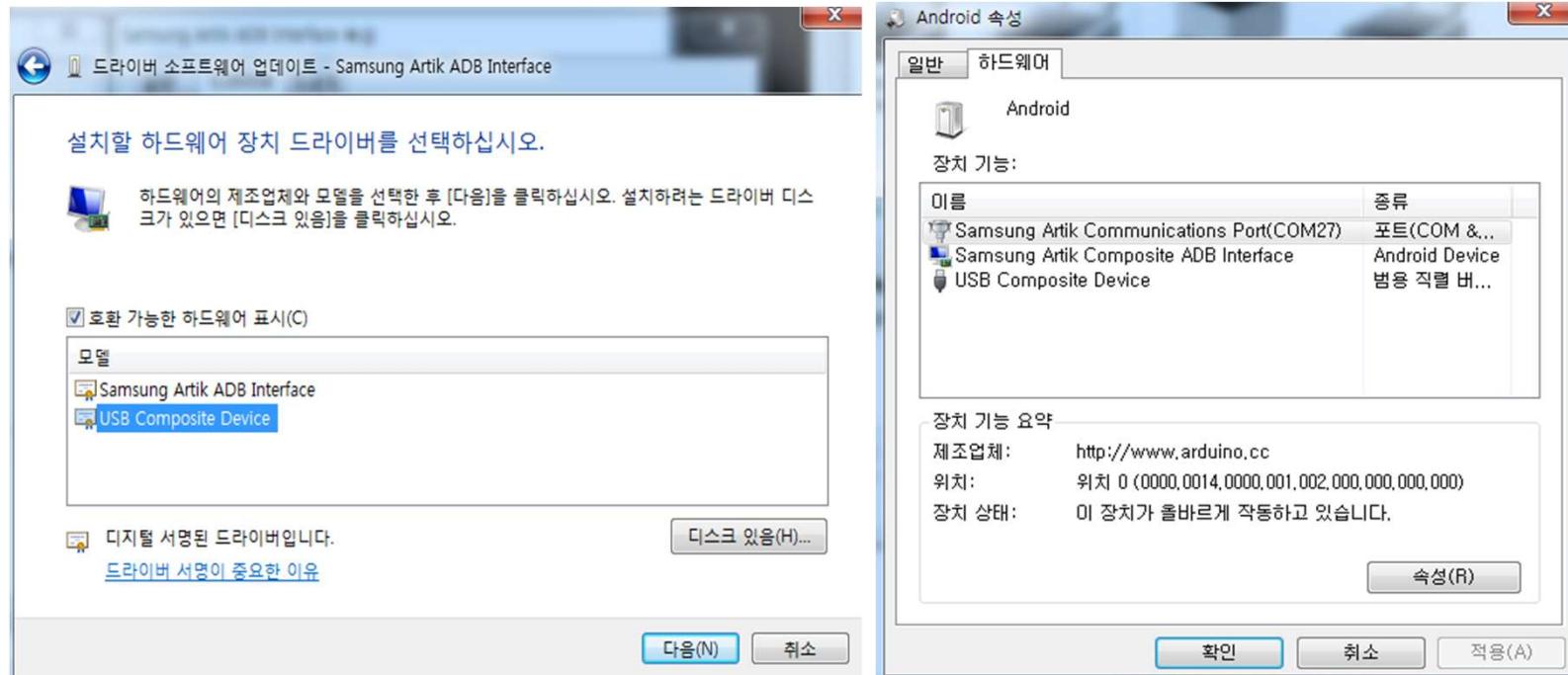
▪ Step 4 : Update Driver Software

- Choose "USB Composite Device" and click "Next" to let the installation complete.
 - Reset the ARTIK board and relaunch watcher. Now, you may see devices show :

USB Composite Device

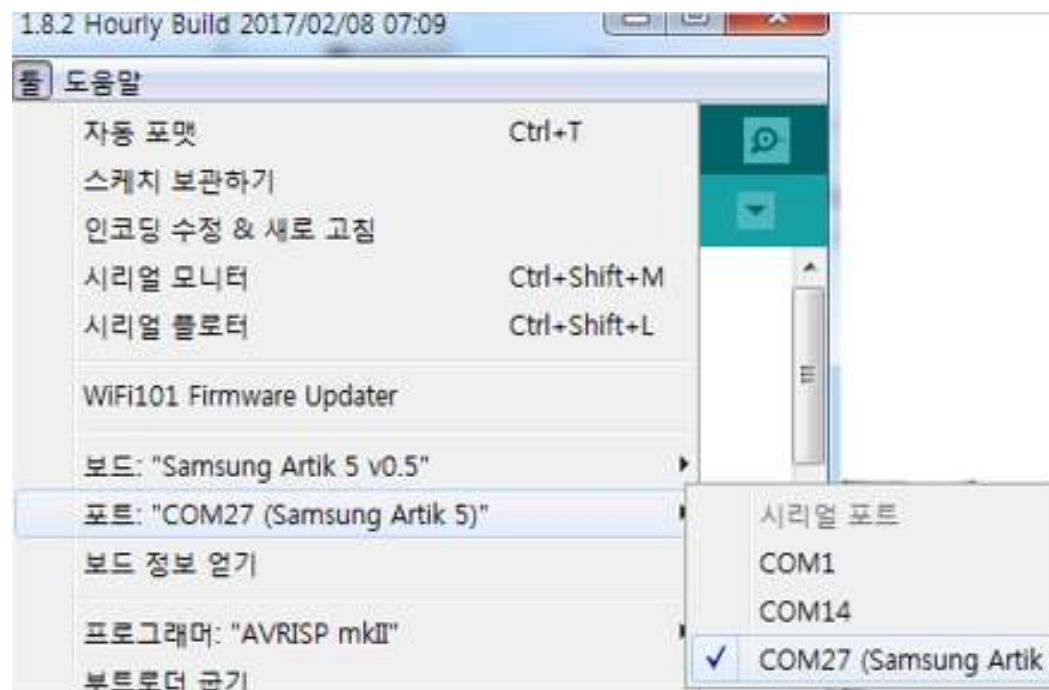
Samsung Artik Composite ADB Interface

Samsung Artik Communications Port(COMx)



■ Connect Arduino IDE to ARTIK over USB

- Step 5 : Select port in Arduino IDE
 - Under [Tools]>[Port], select "COMxx (Samsung Artik 5)".



Arduino IDE – “Hello Word” Example

■ Compile Arduino code and upload to ARTIK

- Step 1 : Write code in Arduino IDE as follows



The screenshot shows the Arduino IDE interface with the title bar "HelloArtik | 아두이노 1.8.2 Hourly Build 2017/02/08 07:09". The menu bar includes "파일", "편집", "스케치", "둘", and "도움말". Below the menu is a toolbar with icons for file operations. The main window displays the code for the "HelloArtik" sketch:

```
#include <DebugSerial.h>

void setup() {
    DebugSerial.begin(9600);
}

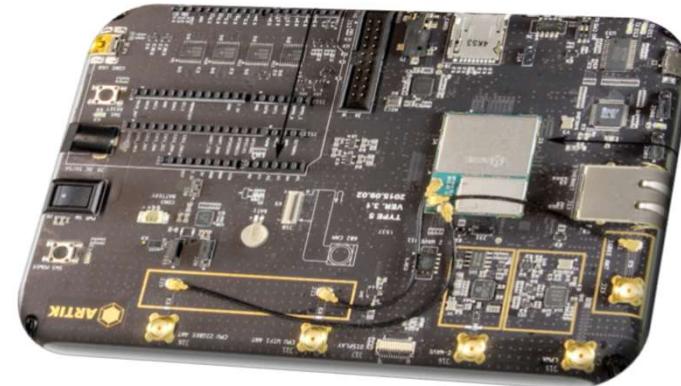
void loop() {
    DebugSerial.println("Hello, Artik!");
    exit(1);
}
```

- Step 2 : Click [Sketch]>[Upload] or use [Crtl]+[u]

```
[root@localhost ~]# ./watcher
[ 213.473827] android_usb: already disabled
--> Press Ctrl+C to force start, Ctrl+\ to exit.
.....change detected, Starting sketch execution
Hello, Artik!
.....Quit (core dumped)
```

I. ARTIK 소개

4. Firmware Update



ARTIK Firmware Update

■ Checking current firmware version

- `# cat /etc/artik-release`

■ Download the latest version

- webpage : <https://developer.artik.io/documentation/downloads.html>

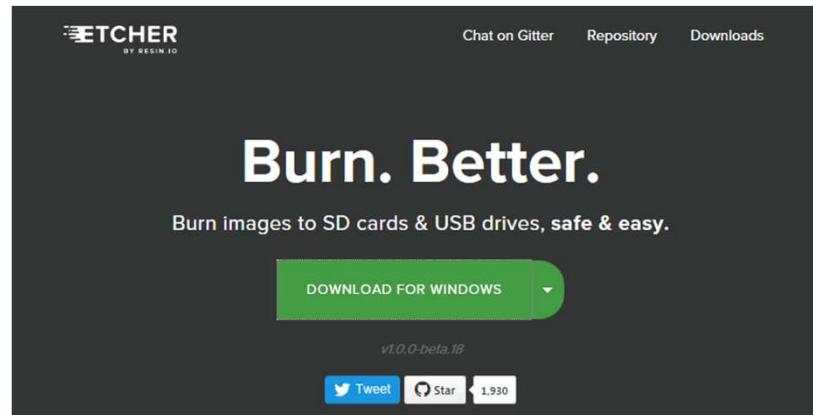
☞ Firmware (8)

DESCRIPTION	UPDATED	SIZE	NOTES
ARTIK 520 Firmware Image A520-OS-1.1.0	02/17/2017	643 MB	
ARTIK 530 Firmware Image A530-OS-1.0.0	02/17/2017	642 MB	
ARTIK ZigBee Tutorial Package 17-01-15	01/26/2017	612 KB	
NCP Firmware 17-01-15	01/26/2017	262 KB	
ARTIK 710 Firmware image A710-OS-1.0.0	10/25/2016	670 MB	
ARTIK 030 Tutorial 1 Package	10/24/2016	235 KB	
ARTIK 520 Firmware image A520-OS-1.0.0	06/21/2016	557 MB	
ARTIK 1020 Firmware image A1020-OS-1.1.0	06/21/2016	557 MB	

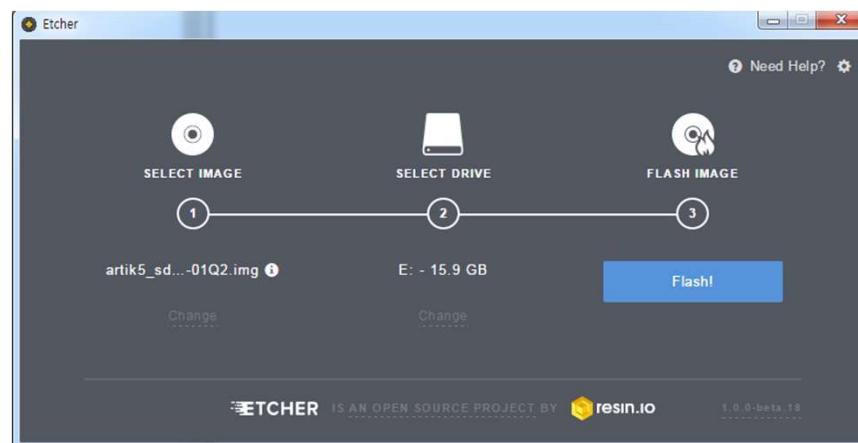
ARTIK Firmware Update

■ Preparing image writing tool

- etcher : <https://etcher.io>



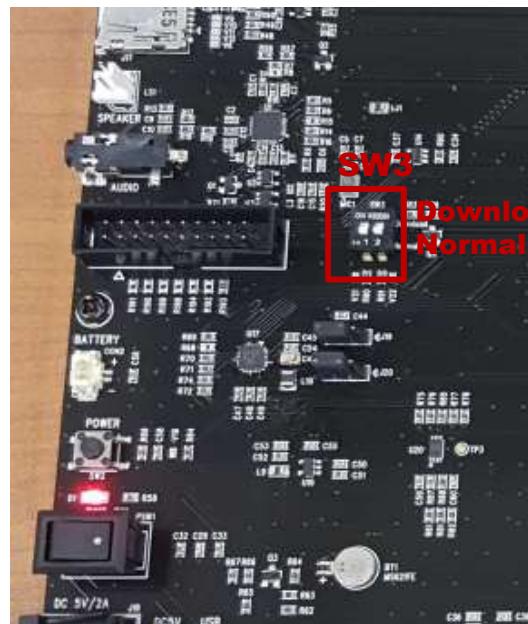
■ Writing the latest firmware image to microSD card



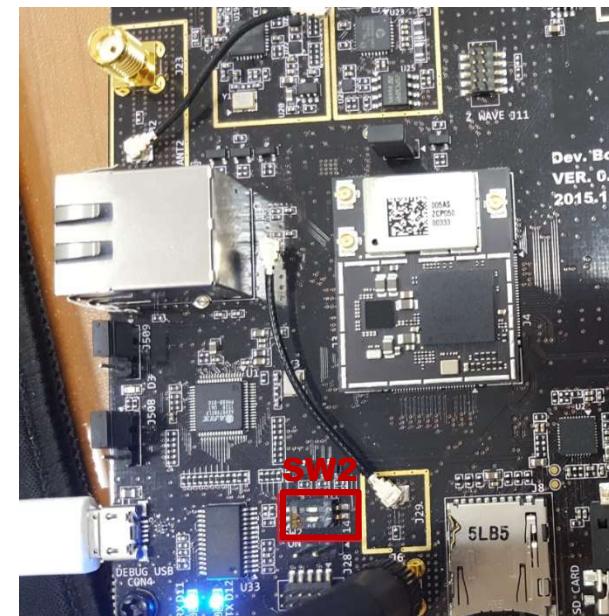
ARTIK Firmware Update

■ Booting ARTIK from the microSD card

- Set two pins of Boot-Switch(sw3) "Download" for PS-ED500
sw2 for ARTIK520 dev. Board.
- "Download" : booting from microSD card
- "Normal" : booting from eMMC (internal storage of ARTIK)



ARTIK Basic

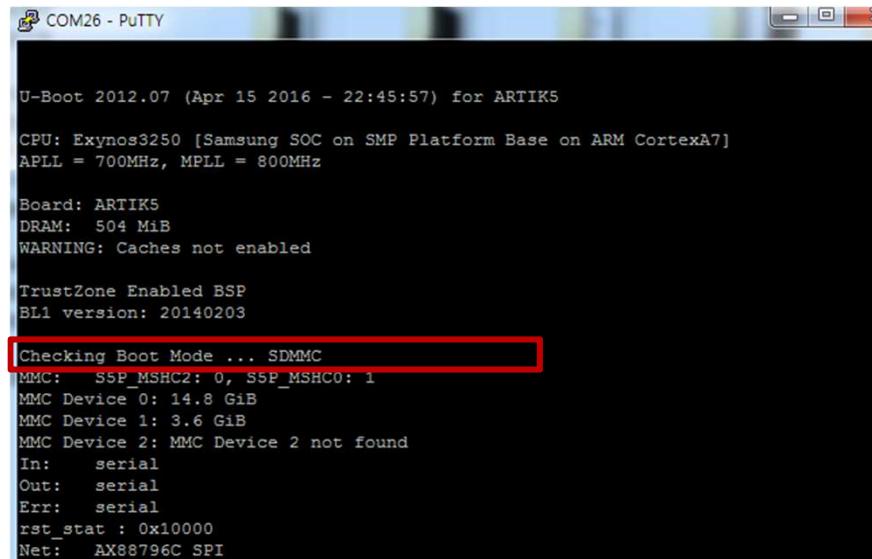


ARTIK520 dev. board

ARTIK Firmware Update

■ Power on the board

- Power on and press the power button for 1 second
- Checking Boot Mode ... SDMMC

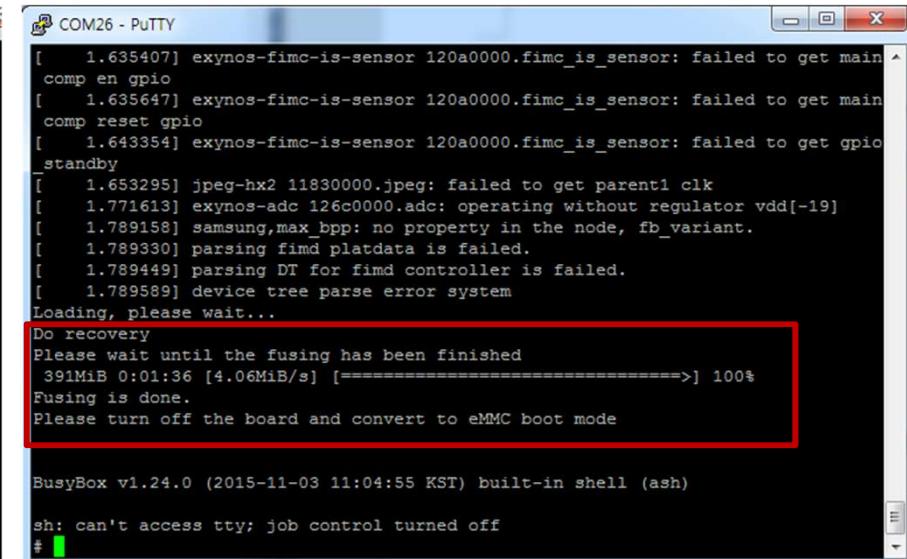


```
U-Boot 2012.07 (Apr 15 2016 - 22:45:57) for ARTIK5
CPU: Exynos3250 [Samsung SOC on SMP Platform Base on ARM CortexA7]
APLL = 700MHz, MPLL = 800MHz
Board: ARTIK5
DRAM: 504 MiB
WARNING: Caches not enabled
TrustZone Enabled BSP
BL1 version: 20140203

[ 1.635407] exynos-fimc-is-sensor 120a0000.fimc_is_sensor: failed to get main
comp en gpio
[ 1.635647] exynos-fimc-is-sensor 120a0000.fimc_is_sensor: failed to get main
comp reset gpio
[ 1.643354] exynos-fimc-is-sensor 120a0000.fimc_is_sensor: failed to get gpio
_standby
[ 1.653295] jpeg-hx2 11830000.jpeg: failed to get parent1 clk
[ 1.771613] exynos-adc 126c000.adc: operating without regulator vdd[-19]
[ 1.789158] samsung,max_bpp: no property in the node, fb_variant.
[ 1.789330] parsing fimd platdata is failed.
[ 1.789449] parsing DT for fimd controller is failed.
[ 1.789589] device tree parse error system
Loading, please wait...
Do recovery
Please wait until the fusing has been finished
391MiB 0:01:36 [4.06MiB/s] [=====>] 100%
Fusing is done.
Please turn off the board and convert to eMMC boot mode

BusyBox v1.24.0 (2015-11-03 11:04:55 KST) built-in shell (ash)
sh: can't access tty; job control turned off
#
```

Starting SDMMC boot mode



```
[ 1.635407] exynos-fimc-is-sensor 120a0000.fimc_is_sensor: failed to get main
comp en gpio
[ 1.635647] exynos-fimc-is-sensor 120a0000.fimc_is_sensor: failed to get main
comp reset gpio
[ 1.643354] exynos-fimc-is-sensor 120a0000.fimc_is_sensor: failed to get gpio
_standby
[ 1.653295] jpeg-hx2 11830000.jpeg: failed to get parent1 clk
[ 1.771613] exynos-adc 126c000.adc: operating without regulator vdd[-19]
[ 1.789158] samsung,max_bpp: no property in the node, fb_variant.
[ 1.789330] parsing fimd platdata is failed.
[ 1.789449] parsing DT for fimd controller is failed.
[ 1.789589] device tree parse error system
Loading, please wait...
Do recovery
Please wait until the fusing has been finished
391MiB 0:01:36 [4.06MiB/s] [=====>] 100%
Fusing is done.
Please turn off the board and convert to eMMC boot mode

BusyBox v1.24.0 (2015-11-03 11:04:55 KST) built-in shell (ash)
sh: can't access tty; job control turned off
#
```

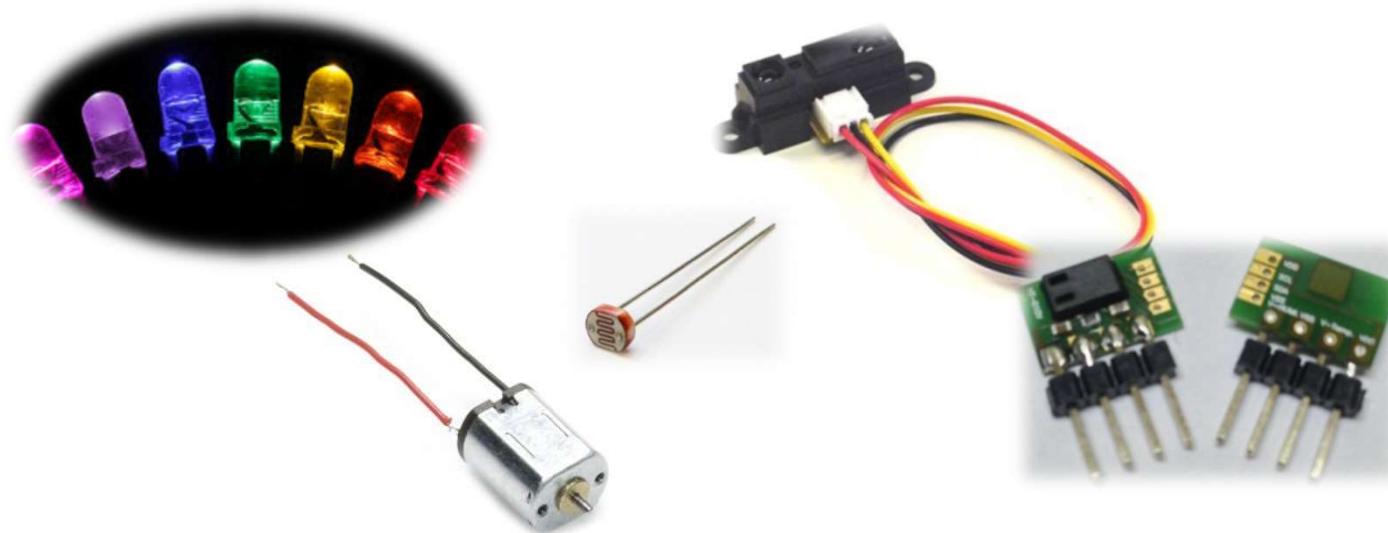
Finishing downloading ARTIK firmware

■ Reboot

- Set sw3 "Normal" for PS-ED500, or sw2 OFF for ARTIK520 dev. board
- Reboot for normal booting from eMMC mode

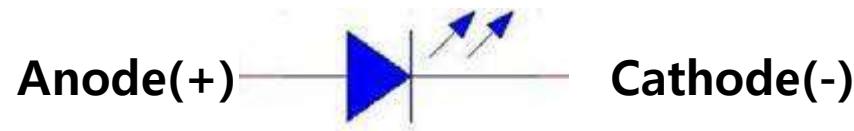
II. ARTIK 기술 교육

1. GPIO Control



LED(Light-Emitting Diode)

■ Symbol of LED



■ Type of LED

- DIP(Duel In-Line Package) Type
- SMD(Surface Mount Device/Diode) Type



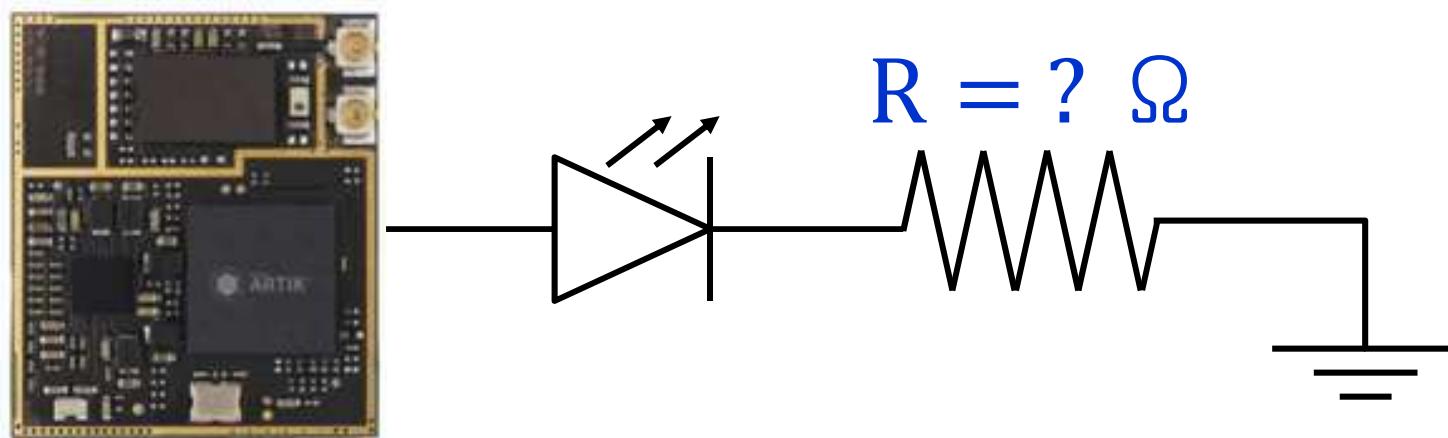
[DIP type]



[SMD type]

LED(Light-Emitting Diode)

■ How to calculate LED connection resistance



LED(1.7V, 10mA 기준)

$$R = \frac{V_{supply} - V_{LED}}{I_{LED}} = \frac{3.3V - 1.7V}{10mA} = 160\Omega$$

List of examples

■ Using Linux command line

- Control LED on/off

■ Using C compiler

- Control LED on/off
- Control LED with switch

■ Using the Arduino IDE

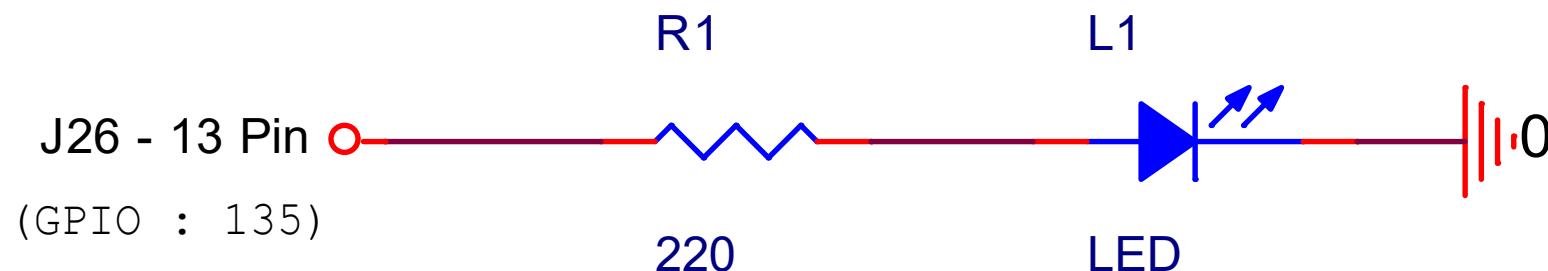
- Control LED on/off

Using Linux Command Line

■ Required Hardware

- ARTIK 5 beta developer kit
- DIP type LED
- 220 ohm resistor
- Breadboard
- Connector wires

■ Circuit Configuration



Using Linux Command Line

■ Command

- Request control of the desired GPIO pin.
 - `# echo 135 > /sys/class/gpio/export`
 - This command makes a GPIO-specific directory created.
- Configure the GPIO pin to be an output.
 - `# echo out > /sys/class/gpio/gpio135/direction`
- Set the output level by writing the value file contents to "0" or "1".
 - `# echo 1 > /sys/class/gpio/gpio135/value`
 - `# echo 0 > /sys/class/gpio/gpio135/value`
- Unexport GPIO pin
 - `# echo 135 > /sys/class/gpio/unexport`

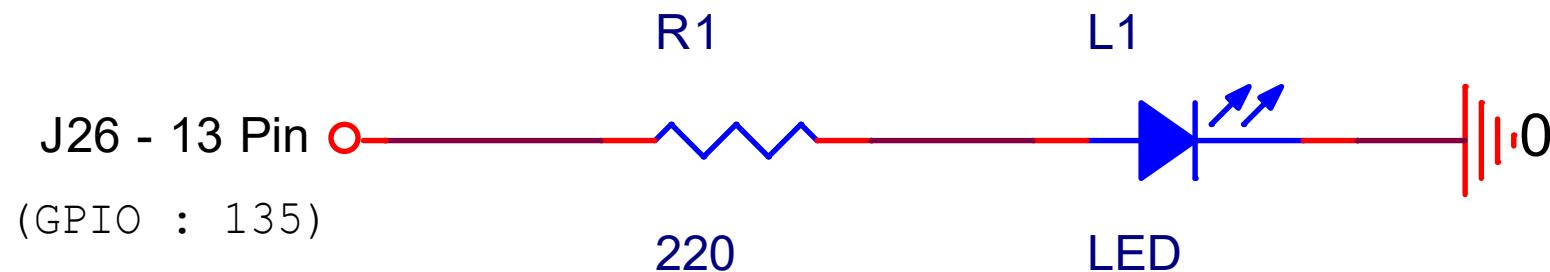
```
[root@localhost ~]# echo 135 > /sys/class/gpio/export
[root@localhost ~]# echo out > /sys/class/gpio/gpio135/direction
[root@localhost ~]# echo 1 > /sys/class/gpio/gpio135/value
[root@localhost ~]# echo 0 > /sys/class/gpio/gpio135/value
[root@localhost ~]# echo 135 > /sys/class/gpio/unexport
[root@localhost ~]# █
```

C Compiler Example (1)

■ Required Hardware

- ARTIK 5 beta developer kit
- DIP type LED
- 220 ohm resistor
- Breadboard
- Connector wires

■ Circuit Configuration



C Compiler Example (1)

■ Source Code

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <unistd.h>

#define HIGH 1
#define LOW 0
#define INPUT 1
#define OUTPUT 0

int outputPin = 135;

bool digitalPinMode(int pin, int dir)
{
    FILE * fd;
    char fName[128];

    // Exporting the pin to be used
    if(( fd = fopen("/sys/class/gpio/export", "w")) == NULL){
        printf("Error: unable to export pin\n");
        return false;
    }
    fprintf(fd, "%d\n", pin);
    fclose(fd);

    // Setting direction of the pin
    sprintf(fName, "/sys/class/gpio/gpio%d/direction", pin);
    if((fd = fopen(fName, "w")) == NULL){
        printf("Error: can't open pin direction\n");
        return false;
    }
    if(dir == OUTPUT) { fprintf(fd, "out\n"); }
    else {fprintf(fd, "in\n"); }

    fclose(fd);
    return true;
}
```

```
bool digitalWrite(int pin, int val)
{
    FILE * fd;
    char fName[128];

    // Open pin value file
    sprintf(fName, "/sys/class/gpio/gpio%d/value", pin);
    if((fd = fopen(fName, "w")) == NULL)
    {
        printf("Error: can't open pin value\n");
        return false;
    }
    if(val == HIGH) {fprintf(fd, "1\n"); }
    else {fprintf(fd, "0\n"); }

    fclose(fd);
    return true;
}

int setup()
{
    if (!digitalPinMode(outputPin, OUTPUT)) return -1;
    return 0;
}

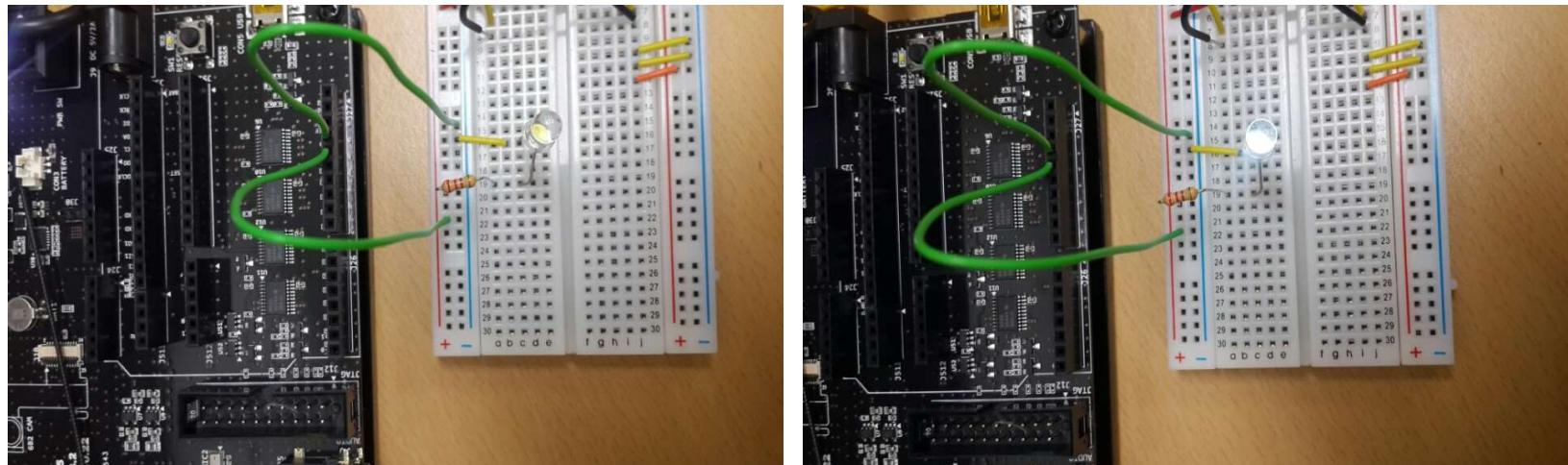
int main(void)
{
    if (setup() == -1) {exit(1);}

    while(1)
    {
        digitalWrite(outputPin, HIGH);
        sleep(1);
        digitalWrite(outputPin, LOW);
        sleep(1);
    }
    return 0;
}
```

C Compiler Example (1)

■ Execution result

- Before executing code : LED off
- After executing code : LED repeats on / off at 1 second intervals



ARTIK 520



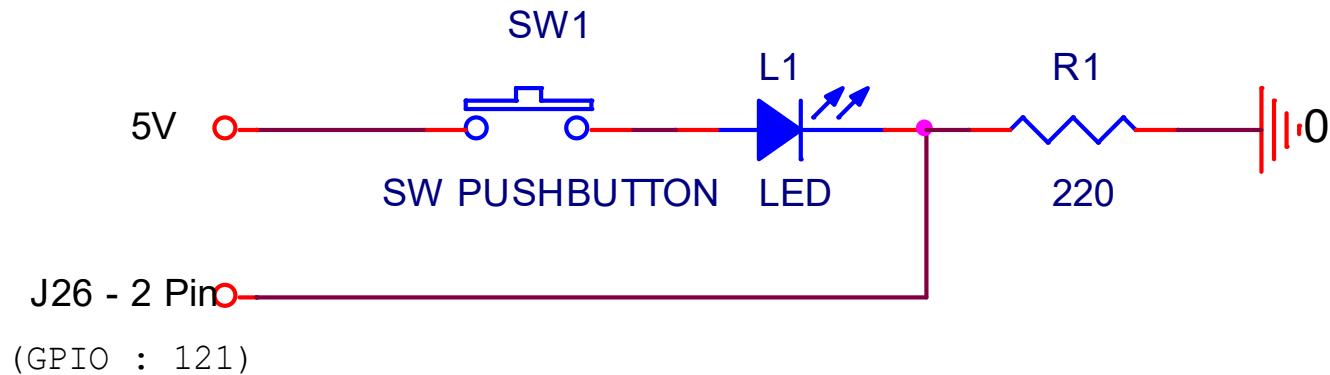
PS-ED500

C Compiler Example (2)

■ Required Hardware

- ARTIK 5 beta developer kit
- LED
- 220 ohm resistor
- Momentary button or Switch
- Breadboard
- Connector wires

■ Circuit Configuration



C Compiler Example (2)

■ PS-ED500 Switch – LED GPIO

- 같은 GPIO에 연결되어 있는 LED와 Switch를 함께 사용하지 않도록 주의
- 한 핀을 SW로 사용할 때는 LED의 점퍼 핀을,
LED로 사용할 때는 SW 점퍼 핀 제거
- C Compiler Example(3)의 회로도 참고

PS-ED500	D13, SW4	D14, SW5	D15, SW6
ARTIK 520	J26[2]	J26[3]	J26[4]
GPIO	121	122	123

C Compiler Example (2)

■ Source Code

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <unistd.h>

#define HIGH 1
#define LOW 0
#define INPUT 1
#define OUTPUT 0

int inputPin = 121;

bool digitalPinMode(int pin, int dir)
{
    FILE * fd;
    char fName[128];

    // Exporting the pin to be used
    if(( fd = fopen("/sys/class/gpio/export", "w")) == NULL){
        printf("Error: unable to export pin\n");
        return false;
    }
    fprintf(fd, "%d\n", pin);
    fclose(fd);

    // Setting direction of the pin
    sprintf(fName, "/sys/class/gpio/gpio%d/direction", pin);
    if((fd = fopen(fName, "w")) == NULL){
        printf("Error: can't open pin direction\n");
        return false;
    }
    if(dir == OUTPUT) {fprintf(fd, "out\n");}
    else {fprintf(fd, "in\n");}

    fclose(fd);
    return true;
}
```

```
int digitalRead(int pin)
{
    FILE * fd;
    char fName[128];
    char val[2];

    // Open pin value file
    sprintf(fName, "/sys/class/gpio/gpio%d/value", pin);
    if((fd = fopen(fName, "r")) == NULL)
    {
        printf("Error: can't open pin value\n");
        return false;
    }
    fgets(val, 2, fd);
    fclose(fd);

    return atoi(val);
}

int setup()
{
    if (!digitalPinMode(inputPin, INPUT)) return -1;

    return 0;
}

int main(void)
{
    if (setup() == -1) { exit(1); }

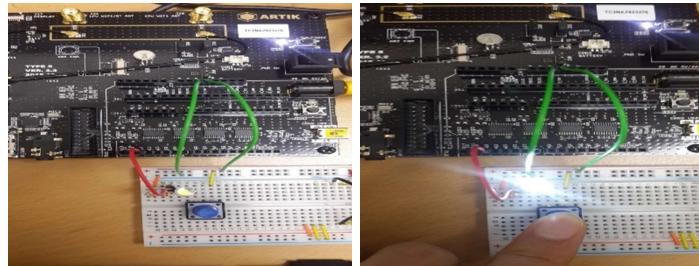
    while(1)
    {
        int sensorVal;
        sensorVal = digitalRead(inputPin);
        printf("sensorVal is %d\n", sensorVal);
        sleep(1);
    }

    return 0;
}
```

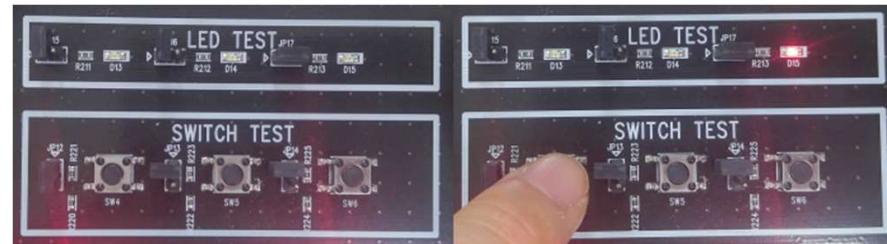
C Compiler Example (2)

■ Execution result

- Before executing code : LED off
- After executing code
 - Switch is pressed – LED on, sensorVal = 1
 - Switch is released– LED off, sensorVal = 0



ARTIK 520



PS-ED500

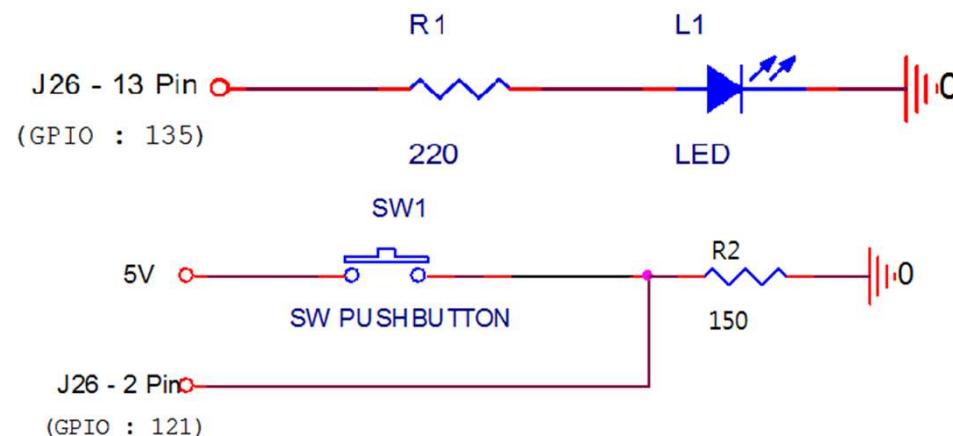
```
Last login: wed Jan 4 00:13:00 on ttys002
[root@localhost ~]# ./led2
sensorVal is 0
sensorVal is 0
sensorVal is 1
sensorVal is 1
sensorVal is 0
sensorVal is 0
sensorVal is 0
```

C Compiler Example (3)

■ Required Hardware

- ARTIK 5 beta developer kit
- LED
- 220 ohm, 150 ohm resistor
- Momentary button or Switch
- Breadboard
- Connector wires

■ Circuit Configuration



C Compiler Example (3)

■ Source Code -(1)

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <unistd.h>

#define HIGH 1
#define LOW 0
#define INPUT 1
#define OUTPUT 0

int inputPin = 121;
int outputPin = 135;

bool digitalPinMode(int pin, int dir) {
    FILE *fd;
    char fName[128];

    // Exporting the pin to be used
    fd = fopen("/sys/class/gpio/export", "w");
    if (fd == NULL) {
        printf("Error: unable to export pin\n");
        return false;
    }
    fprintf(fd, "%d\n", pin);
    fclose(fd);

    // Setting direction of the pin
    sprintf(fName, "/sys/class/gpio/gpio%d/direction", pin);
    fd = fopen(fName, "w");
    if (fd == NULL) {
        printf("Error: can't open pin direction\n");
        return false;
    }
    if (dir == OUTPUT) fprintf(fd, "out\n");
    else fprintf(fd, "in\n");

    fclose(fd);
    return true;
}
```

```
int digitalRead(int pin) {
    FILE *fd;
    char fName[128];
    char val[2];

    // Open pin value file
    sprintf(fName, "/sys/class/gpio/gpio%d/value", pin);
    fd = fopen(fName, "r");
    if (fd == NULL) {
        printf("Error: can't open pin value\n");
        return false;
    }
    fgets(val, 2, fd);
    fclose(fd);
    return atoi(val);
}

bool digitalWrite(int pin, int val) {
    FILE *fd;
    char fName[128];

    // Open pin value file
    sprintf(fName, "/sys/class/gpio/gpio%d/value", pin);
    fd = fopen(fName, "w");
    if (fd == NULL) {
        printf("Error: can't open pin value\n");
        return false;
    }
    if (val == HIGH) fprintf(fd, "1\n");
    else fprintf(fd, "0\n");

    fclose(fd);
    return true;
}
```

C Compiler Example (3)

■ Source Code -(2)

```
int setup() {
    if (!digitalPinMode(outputPin, OUTPUT) ||
        !digitalPinMode(inputPin, INPUT)) return -1;
    return 0;
}

int main(void) {
    int sensorVal;
    int ledState;

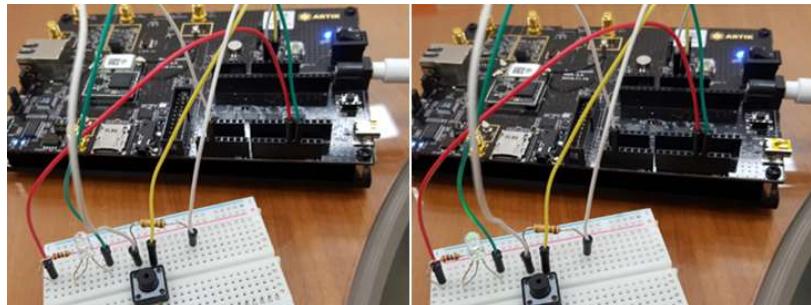
    if (setup() == -1) exit(1);

    // initialize
    ledState = LOW;
    digitalWrite(outputPin, ledState);
    while(1) {
        sensorVal = digitalRead(inputPin);
        if (sensorVal == 1) {
            if (ledState == LOW) ledState = HIGH;
            else ledState = LOW;
            digitalWrite(outputPin, ledState);
        }
        printf("sensorVal is %d\n", sensorVal);
        sleep(1);
    }
    return 0;
}
```

C Compiler Example (3)

■ Execution result

- Before executing code : LED off
- After executing code
 - Switch is pressed – LED state change



ARTIK 520



PS-ED500

Using the Arduino IDE

■ GPIO Pin Number

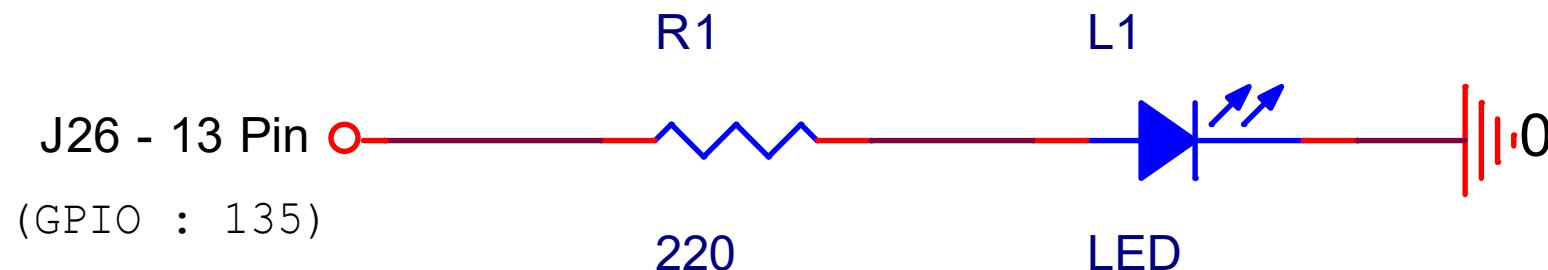
External Pin	A520-GPIO	Arduino Pin
J26[2]	121	2
J26[3]	122	3
J26[4]	123	4
J26[7]	124	7
J27[8]	125	8
J27[9]	126	9
J27[10]	127	10
J27[11]	129	11
J27[12]	134	12
J27[13]	135	13

Using the Arduino IDE

■ Required Hardware

- ARTIK 5 beta developer kit
- DIP type LED
- 220 ohm resistor
- Breadboard
- Connector wires

■ Circuit Configuration



Using the Arduino IDE

■ Arduino Source Code

```
void setup() {  
    pinMode(8, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(8, HIGH);  
    delay(1000);  
    digitalWrite(8, LOW);  
    delay(1000);  
}
```

■ Execute result

- Enter [Ctrl]+[u] or using menu, compile and upload your code to ARTIK.
- Then, you should see LED blinking.

II. ARTIK 기술 교육

2. PWM Control



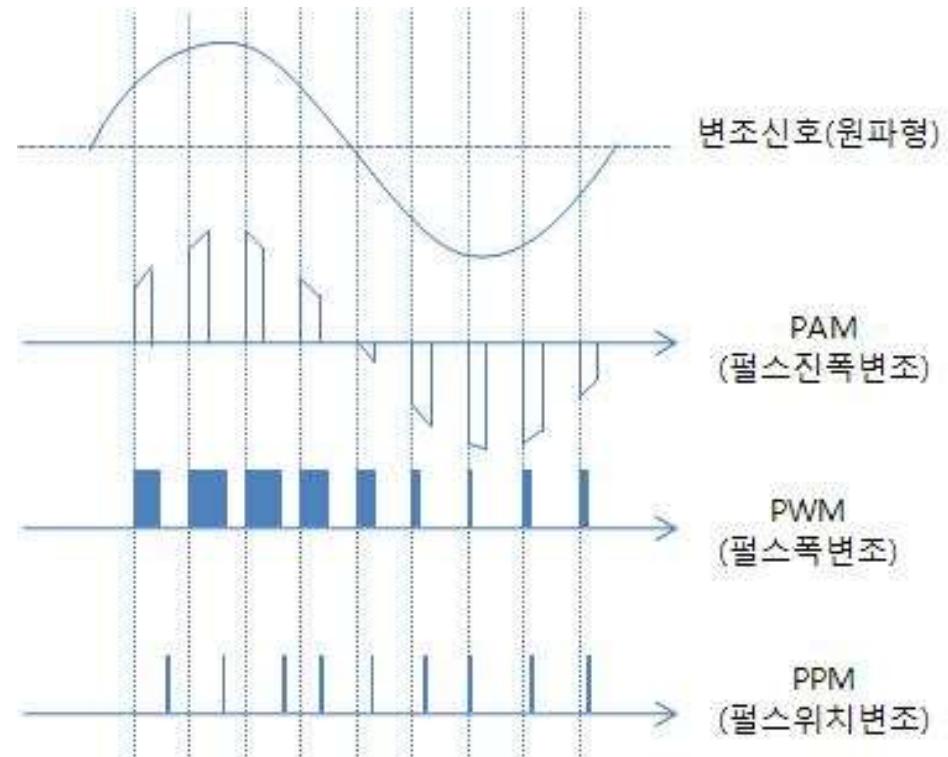
Pulse Modulation

■ PM(Pulse Modulation)

- A method of modulating a periodic pulse by a signal

■ Type of Pulse Modulation

- PAM(Pulse Amplitude Modulation)
- PWM(Pulse Width Modulation)
- PPM(Pulse Position Modulation)



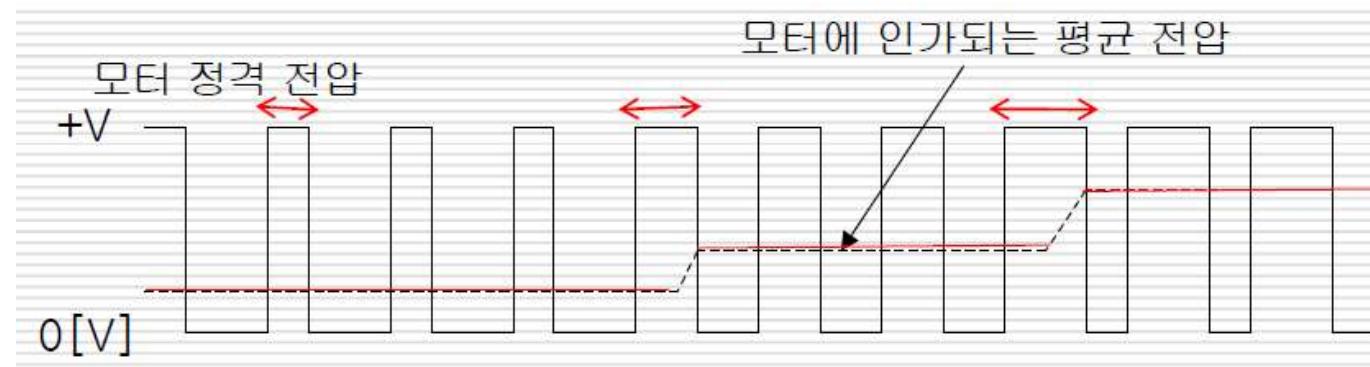
PWM(Pulse Width Modulation)

■ Pulse

- Signal level changes over a short period of time
- Pulse Width
- The width of one pulse

■ PWM

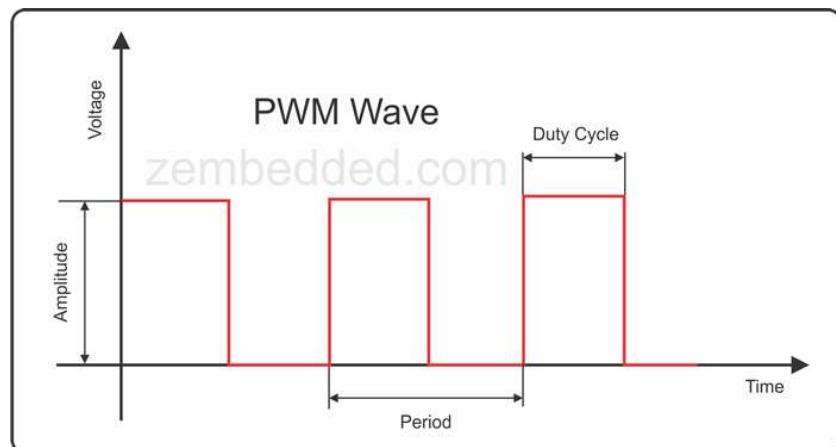
- Methods of adjusting 'ON' or 'OFF' time of pulse
- It is able to adjust pulse width and control motor because the average voltage applied to the motor can be changed and the effect of changing the magnitude of the voltage can be obtained, in the case of a DC motor.



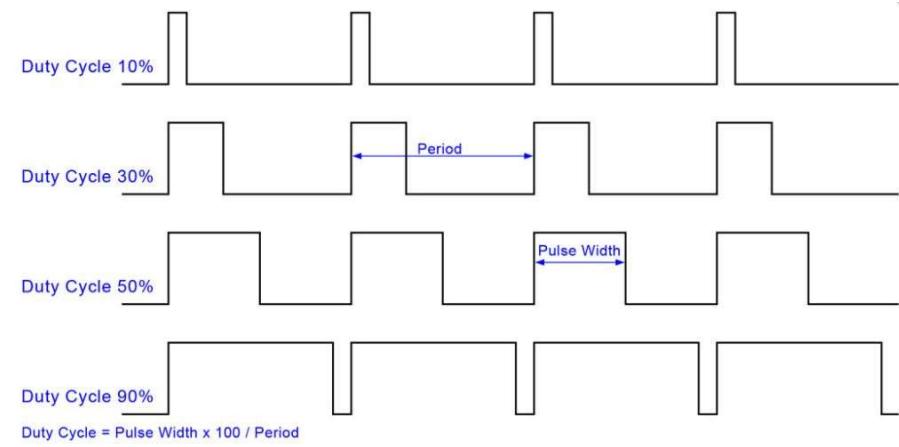
PWM(Pulse Width Modulation)

■ PWM control

- Period : Time at any regular interval
- Duty Cycle : The ratio of 'ON' time in a period



(1) PWM Wave

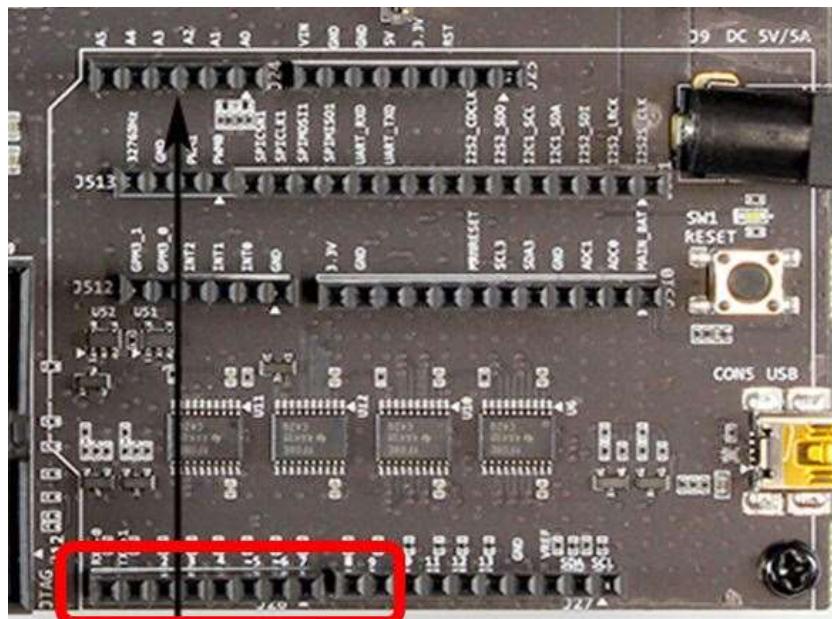


(2) Duty Cycle

PWMs in ARTIK

■ PWM pins

- J26[5] (XPWM01)
- J26[6] (XPWM00)



(1) ARTIK5 Board

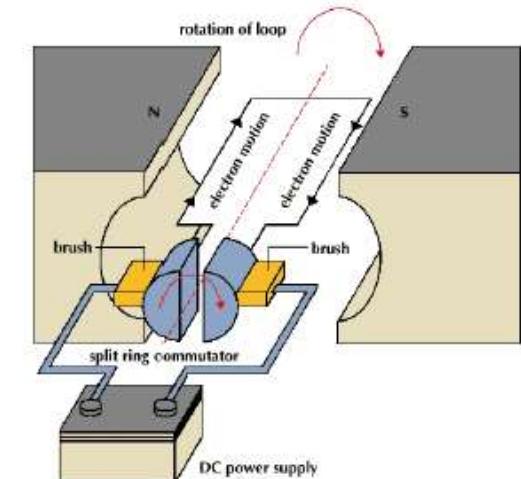
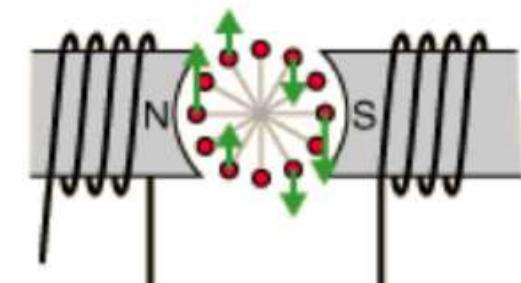
A-520 Header J26

EXTERNAL PIN	SYSFS MAPPING	NOTES
[Rx-0]	Rx	
[Tx-1]	Tx	
[2]	GPIO 121	(GPX0[0])
[3]	GPIO 122	(GPX0[1])
[4]	GPIO 123	(GPX0[2])
[5] [6]	(pwm functions)	These pins are reversed between production boards and early boards. Refer to this section .
[7]	GPIO 124 (GPX0[3])	

(2) ARTIK5 Pin mapping (J26)

■ DC Motor

- A motor, which uses DC power, rotates with a repulsive force between the magnetic field of the permanent magnet and the electromagnetic force generated by the current flowing in the conductor lying in the magnetic field.
- Electrical flame, rotation noise at contact point because of brush and shortening of life.
- Large driving torque
- Rotation characteristics linearly proportional to the applied voltage
- Output torque linearly proportional to input current
- Good output efficiency and low cost



DC Motor

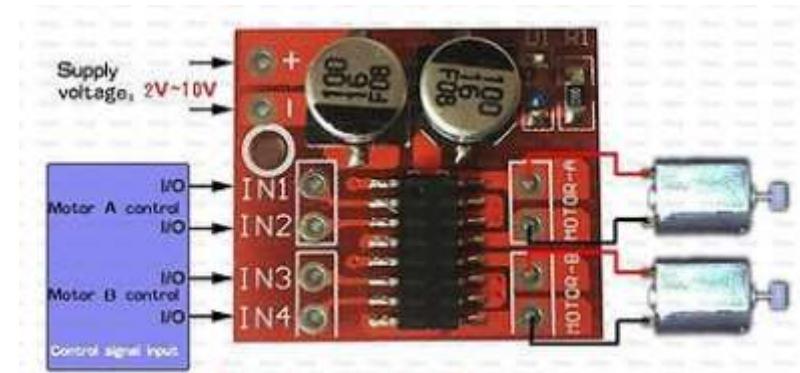
■ Motor Driver(MX1508 2-channel)

- Use to control motor rotation direction and driving speed
- Control by 2 signals which control other direction



Model	MX1508 2-channel
Motor control	2EA
Maximum current	2.5A
Driving voltage	2V~10V
Controller	MX1508
Control signal	PWM
Board size	82.47*21*5mm (LxWxH)

(1) Motor Driver Spec



(2) Motor Driver H/W

■ Motor Driver connection methods

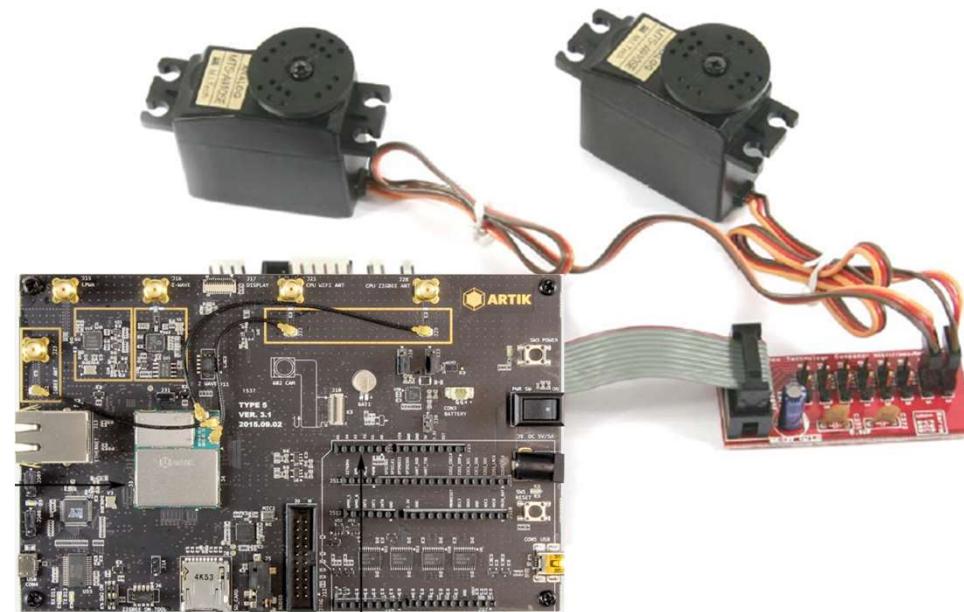
	회전방식	IN1	IN2	IN3	IN4
Motor-A	정방향	1/PWM	0		
	역방향	0	1/PWM		
	대기	0	0		
	정지	1	1		
Motor-B	정방향			1/PWM	0
	역방향			0	1/PWM
	대기			0	0
	정지			1	1

(3) Motor Driver Control Signals

Servo Motor

■ RC Servo Motor

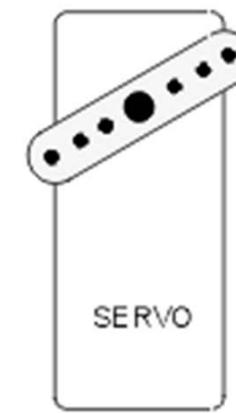
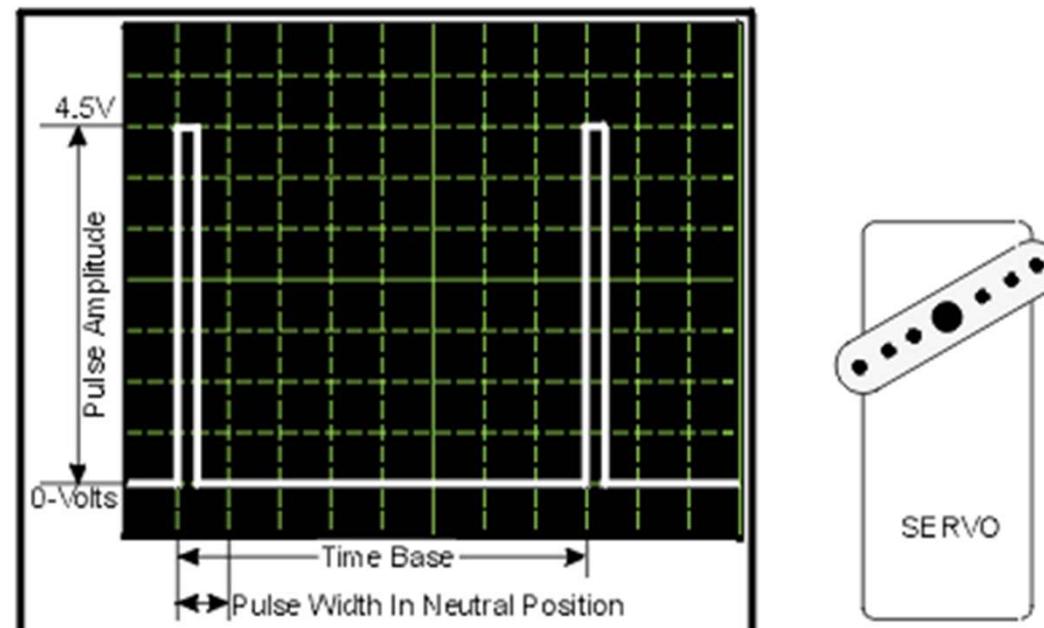
- Used for Radio Control's airplane, automobile, rear-wheel wireless models
- Open-loop control
- Able to produce large force with small size by using large gear ratio
- Mainly use in small size robots
- The operating range of the servo is $-60^\circ \sim +60^\circ$, $-90^\circ \sim +90^\circ$, torque and response speed are different, depending on servo characteristics and price.
- There are three signal lines(VCC, GND, control line)



Servo Motor

■ RC Servo Motor control

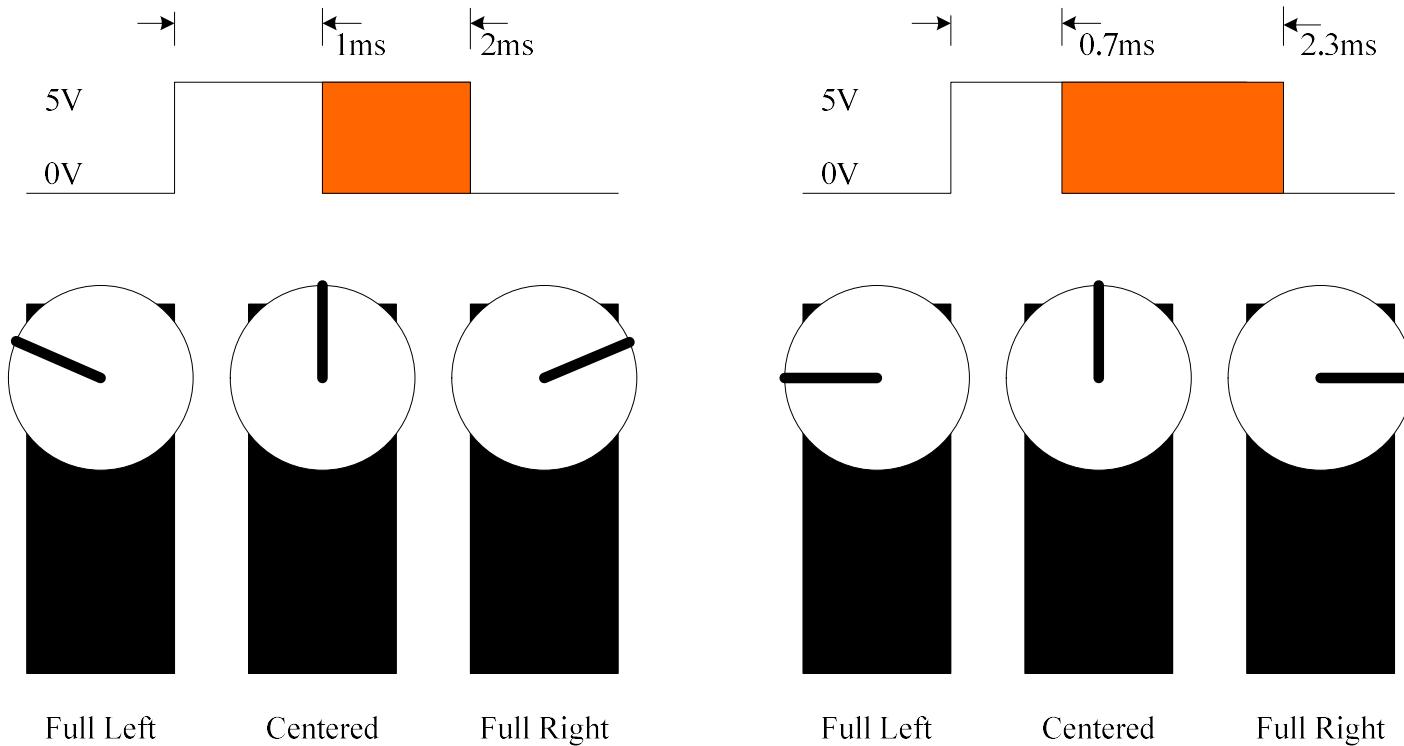
- Passing target angle through control line(specified by pulse width) is called by PCM (Pulse Code Modulation)
- Pulse period : 20msec
- Generally, 1.5 msec is a neutral point. If it is shorter than 1.5 msec, the motor moves toward 0 degree. If it is longer than 1.5 msec, the motor moves toward 180 degree.



Servo Motor

■ RC Servo Motor control

- less than 0.7ms → 0° (90° to the left)
- 1.5ms → 90° (neutral point)
- More than 2.3ms → 180°(90° to the right)



List of examples

■ Using Linux command line

- Control duty cycle and period

■ Using C compiler

- Control DC motor with kbhit() function
- Control Servo motor

■ Using the Arduino IDE

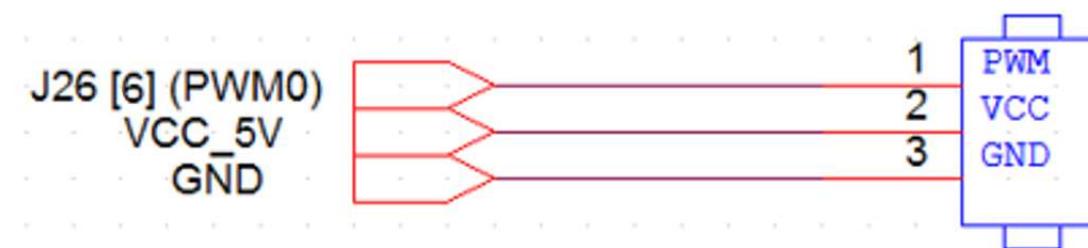
- Control PWM pin, making the LED pulse slowly on and off

Using Linux Command Line

■ Required Hardware

- ARTIK 5 beta developer kit
- Servo Motor (SG90)
- Breadboard
- Connector wires

■ Circuit Configuration



■ Command

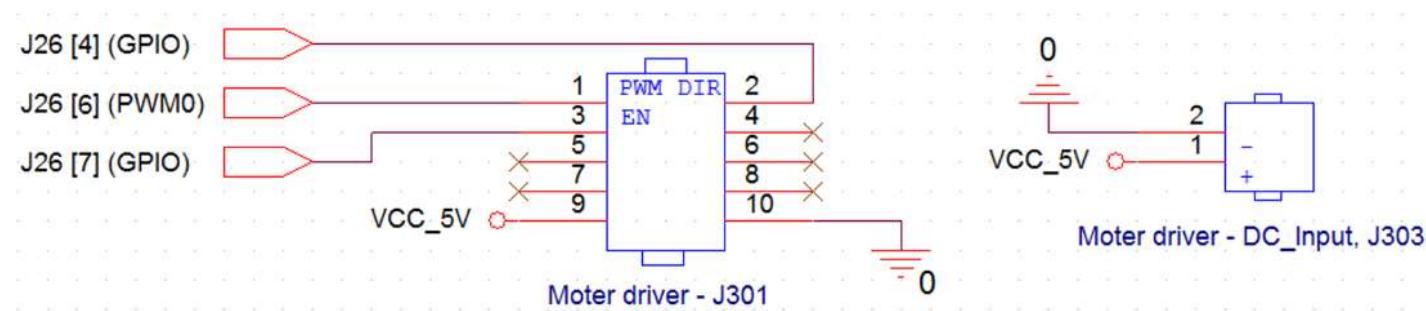
- Export PWM0
 - `# echo 0 > /sys/class/pwm/pwmchip0/export`
 - It will make a pwm0 subdirectory created.
- Set period to 20ms (Unit : ns)
 - `# echo 20000000 > /sys/class/pwm/pwmchip0/pwm0/period`
- Set duty cycle to 1.5ms
 - `# echo 1500000 > /sys/class/pwm/pwmchip0/pwm0/duty_cycle`
 - It will set the angle of rotation of servo motor to 0°.
 - For SE-A410, $\text{duty_cycle} = 1500000 + \text{angle} * 10000$
 - For SG90, $\text{duty_cycle} = 1500000 + \text{angle} * 8889$
- Enable PWM0
 - `# echo 1 > /sys/class/pwm/pwmchip0/pwm0/enable`
- Disable PWM0
 - `# echo 0 > /sys/class/pwm/pwmchip0/pwm0/disable`
- Unexport PWM0
 - `# echo 0 > /sys/class/pwm/pwmchip0/unexport`

C Example (1) - DC Motor

■ Required Hardware

- ARTIK 5 beta developer kit
- DC Motor(KR-250)
- Motor driver(AM-DC2-2D)
- Breadboard
- Connector wires

■ Circuit Configuration



C Example (1) - DC Motor

■ PWM function

- `bool pwmPin(int pin, int period, int duty_cycle, int val)`
 - int pin : PWM Pin setting(XPWM00, XPWM01)
 - int period : PWM period setting
 - int duty_cycle : PWM duty_cycle setting
 - int val : PWM enable, disable setting

■ PWM Code(1)

```
//period, duty_cycle -> nsec 기준
bool pwmPin(int pin, int period, int duty_cycle, int val){
    FILE *fd;
    char fName[128];

    if ((fd = fopen("/sys/class/pwm/pwmchip0/export", "w")) == NULL) {
        printf("Error: unable to export pin\n");
        return false;
    }
    fprintf(fd, "%d\n", pin);
    fclose(fd);
```

pin

```
sprintf(fName, "/sys/class/pwm/pwmchip0/pwm%d/period", pin);
if ((fd = fopen(fName, "w")) == NULL) {
    printf("Error: can't open pin direction\n");
    return false;
}
fprintf(fd, "%d\n", period);
fclose(fd);
```

period

C Example (1) - DC Motor

■ PWM Code(2)

```
sprintf(fName, "/sys/class/pwm/pwmchip0/pwm%d/duty_cycle", pin);
if((fd = fopen(fName, "w")) == NULL){
    printf("Error: can't open pin direction\n");
    return false;
}
fprintf(fd, "%d\n", duty_cycle);
fclose(fd);
```

duty_cycle

```
if(val == HIGH) {
    sprintf(fName, "/sys/class/pwm/pwmchip0/pwm%d/enable", pin);

    if((fd = fopen(fName, "w")) == NULL) {
        printf("Error: can't open pin value\n");
        return false;
    }
    fprintf(fd, "1\n");
} else {
    sprintf(fName, "/sys/class/pwm/pwmchip0/pwm%d/enable", pin);

    if((fd = fopen(fName, "w")) == NULL){
        printf("Error: can't open pin value\n");
        return false;
    }
    fprintf(fd, "0\n");

}

fclose(fd);

return true;
```

val

C Example (1) - DC Motor

■ Keyboard hit function

- int kbhit()
 - Method of inputing by keyboard
 - Visual studio : O, Linux : X
 - Mainly used by the user to create a header file

```
#include <stdio.h>
#include <termios.h>
#include <unistd.h>
#include <fcntl.h>

int kbhit(void) {
    struct termios oldt, newt;
    int ch;
    int oldf;

    tcgetattr(STDIN_FILENO, &oldt);
    newt = oldt;
    newt.c_lflag &= ~(ICANON | ECHO);
    tcsetattr(STDIN_FILENO, TCSANOW, &newt);
    oldf = fcntl(STDIN_FILENO, F_GETFL, 0);
    fcntl(STDIN_FILENO, F_SETFL, oldf | O_NONBLOCK);

    ch = getchar();

    tcsetattr(STDIN_FILENO, TCSANOW, &oldt);
    fcntl(STDIN_FILENO, F_SETFL, oldf);

    if(ch != EOF) {
        ungetc(ch, stdin);
        return 1;
    }

    return 0;
}
```

(1) kbhit function

C Example (1) - DC Motor

■ Source Code(1)

```
#include "wiringARTIK.h"
#include "kbhit.h"

//===== pin number ======
#define PWM_PIN0          0
#define PWM_PIN1          1
//=====

#define PWM_ENABLE         1
#define PWM_DISABLE        0

#define INC_DEC 100000
#define PERIOD 1000000

static int index=0;
```

```
void print_monitor(int duty_value) {
    int duty_ratio = 0;

    duty_ratio = (duty_value * 100) / PERIOD;

    //printf("duty_value : %d\n", duty_value);
    //printf("duty_ratio : %d\n", duty_ratio);

    printf("===== PWM Test =====\n");
    printf("Please keyboard hit!!!!\n");

    if((index == 0) || (index == 1)) {
        printf("F : Forward Direction\n");
        printf("R : Reverse Direction\n");
        printf("Q : Exit\n");
    }

    else {
        printf("E : PWM Enable & init\n");
        printf("Q : PWM Disable & exit\n");
        printf("I : PWM Increase\n");
        printf("D : PWM Decrease\n\n");
        printf("Duty Ratio (now) : %d%\n", duty_ratio);
    }

    printf("=====*\n");
}
```

C Example (1) - DC Motor

■ Source Code(2)

```

int main(void) {
    Int loop = 1;
    int ch = 0;
    int pin_dir = 0;
    int duty_cycle = 500000;
    int direction = 2;

    pwmPin(PWM_PIN0, PERIOD, 0, PWM_DISABLE);
    pwmPin(PWM_PIN1, PERIOD, 0, PWM_DISABLE);
    print_monitor(0);

    while(loop) {
        if(index == 0) {
            if(kbhit()) {
                pin_dir = getchar();
                index = 1;
            }
        }

        if(index == 1) {
            switch (pin_dir) {
                case 'f' : // f : forward direction
                    direction = PWM_PIN0;
                    index = 2;
                    print_monitor(0);
                    break;

                case 'r' : // r : reverse direction
                    direction = PWM_PIN1;
                    index = 2;
                    print_monitor(0);
                    break;
            }
        }

        case 'q' : // q : exit
        loop = 0;
        break;
    }
    if(!pin_dir)  print_monitor(0);
}

if(index == 2) {
    if(kbhit())    ch = getchar();

    switch (ch) {
        case 'e' : // e : enable & init
            duty_cycle = 500000;
            pwmPin(direction, PERIOD, duty_cycle, PWM_ENABLE);
            printf("Pressed '%c'!\\n", ch);
            printf("PWM Enable\\n");

            index = 2;
            print_monitor(duty_cycle);
            break;

        case 'i' : // i : inc
            duty_cycle += INC_DEC; // Max = 1000000
            if(duty_cycle > PERIOD) duty_cycle = PERIOD;
            pwmPin(direction, PERIOD, duty_cycle, PWM_ENABLE);
            printf("Pressed '%c'!\\n", ch);
            printf("PWM Increase\\n");

            index = 2;
            print_monitor(duty_cycle);
            break;
    }
}

```

C Example (1) - DC Motor

■ Source Code(3)

```
case 'd' : // d : dec
    duty_cycle -= INC_DEC; // Min = 200000
    if(duty_cycle < (PERIOD / 5)) duty_cycle = (PERIOD / 5);

    pwmPin(direction, PERIOD, duty_cycle, PWM_ENABLE);
    printf("Pressed '%c'!\\n", ch);
    printf("PWM Decrease\\n");

    index = 2;
    print_monitor(duty_cycle);
    break;

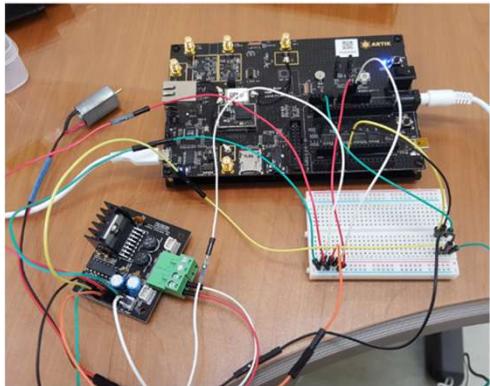
case 'q' : // q : disable & exit
    pwmPin(direction, PERIOD, 0, PWM_DISABLE);
    printf("Pressed '%c'!\\n", ch);
    printf("PWM Disable & Exit\\n");

    index = 0;
    print_monitor(0);
    break;
}
ch = 0;
}
}

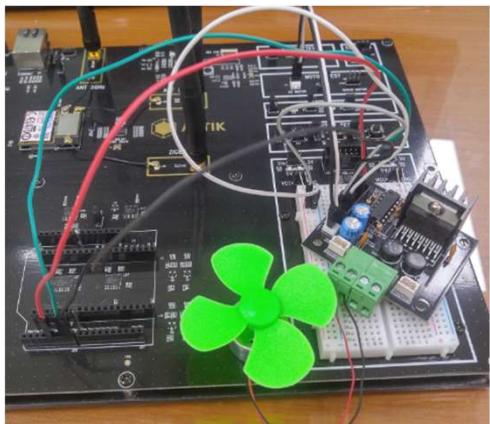
return 0;
}
```

C Example (1) - DC Motor

■ Execution result



ARTIK 520



PS-ED500

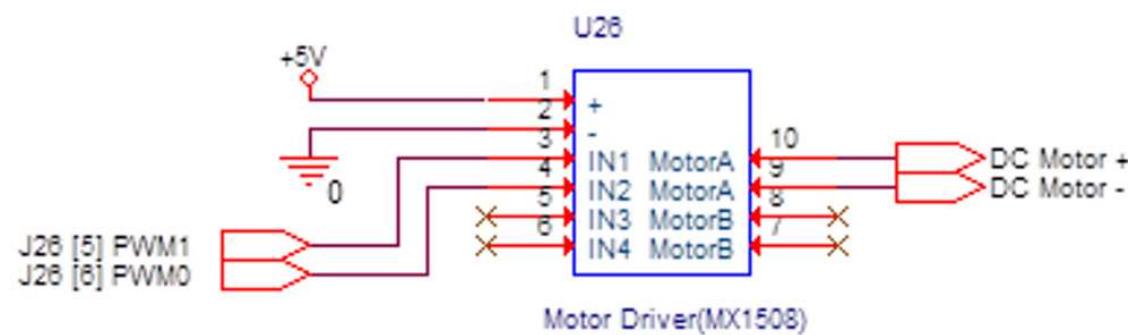
```
[root@localhost motor]# ./motor
duty_value : 0
duty_ratio : 0
===== PWM Test =====
Please keyboard hit!!!!
F : Forward Direction
R : Reverse Direction
Q : Exit
=====
duty_value : 0
duty_ratio : 0
===== PWM Test =====
Please keyboard hit!!!!
E : PWM Enable & init
Q : PWM Disable & exit
I : PWM Increase
D : PWM Decrease
Duty Ratio (now) : 0
=====
Pressed 'e'!
PWM Enable
duty_value : 500000
duty_ratio : 50
===== PWM Test =====
Please keyboard hit!!!!
E : PWM Enable & init
Q : PWM Disable & exit
I : PWM Increase
D : PWM Decrease
Duty Ratio (now) : 50
=====
Pressed 'i'!
PWM Increase
duty_value : 600000
duty_ratio : 60
===== PWM Test =====
Please keyboard hit!!!!
E : PWM Enable & init
Q : PWM Disable & exit
I : PWM Increase
D : PWM Decrease
Duty Ratio (now) : 60
=====
Pressed 'i'!
PWM Increase
duty_value : 700000
duty_ratio : 70
===== PWM Test =====
Please keyboard hit!!!!
E : PWM Enable & init
Q : PWM Disable & exit
I : PWM Increase
D : PWM Decrease
Duty Ratio (now) : 70
=====
Pressed 'i'!
PWM Increase
duty_value : 800000
duty_ratio : 80
===== PWM Test =====
Please keyboard hit!!!!
```

C Example (2) - DC Motor

■ Required Hardware

- ARTIK 5 beta developer kit
- DC Motor(KR-250)
- Motor driver(MX1508-2-Channel)
- Breadboard
- Connector wires

■ Circuit Configuration



C Example (2) - DC Motor

■ Source Code(1)

```
#include "wiringARTIK.h"
#include "kbhit.h"

//===== pin number ======
#define PWM_PIN0          0
#define PWM_PIN1          1
//=====

#define PWM_ENABLE         1
#define PWM_DISABLE        0

#define INC_DEC 100000
#define PERIOD 1000000

static int index=0;
```

```
void print_monitor(int duty_value) {
    int duty_ratio = 0;

    duty_ratio = (duty_value * 100) / PERIOD;

    //printf("duty_value : %d\n", duty_value);
    //printf("duty_ratio : %d\n", duty_ratio);

    printf("===== PWM Test =====\n");
    printf("Please keyboard hit!!!!\n");

    if((index == 0) || (index == 1)) {
        printf("F : Forward Direction\n");
        printf("R : Reverse Direction\n");
        printf("Q : Exit\n");
    }

    else {
        printf("E : PWM Enable & init\n");
        printf("Q : PWM Disable & exit\n");
        printf("I : PWM Increase\n");
        printf("D : PWM Decrease\n\n");
        printf("Duty Ratio (now) : %d%\n", duty_ratio);
    }

    printf("=====*\n");
}
```

C Example (2) - DC Motor

■ Source Code(2)

```

int main(void) {
    Int loop = 1;
    int ch = 0;
    int pin_dir = 0;
    int duty_cycle = 500000;
    int direction = 2;

    pwmPin(PWM_PIN0, PERIOD, 0, PWM_DISABLE);
    pwmPin(PWM_PIN1, PERIOD, 0, PWM_DISABLE);
    print_monitor(0);

    while(loop) {
        if(index == 0) {
            if(kbhit()) {
                pin_dir = getchar();
                index = 1;
            }
        }

        if(index == 1) {
            switch (pin_dir) {
                case 'f' : // f : forward direction
                    direction = PWM_PIN0;
                    index = 2;
                    print_monitor(0);
                    break;

                case 'r' : // r : reverse direction
                    direction = PWM_PIN1;
                    index = 2;
                    print_monitor(0);
                    break;
            }
        }

        case 'q' : // q : exit
        loop = 0;
        break;
    }
    if(!pin_dir)  print_monitor(0);
}

if(index == 2) {
    if(kbhit())    ch = getchar();

    switch (ch) {
        case 'e' : // e : enable & init
            duty_cycle = 500000;
            pwmPin(direction, PERIOD, duty_cycle, PWM_ENABLE);
            printf("Pressed '%c'!\\n", ch);
            printf("PWM Enable\\n");

            index = 2;
            print_monitor(duty_cycle);
            break;

        case 'i' : // i : inc
            duty_cycle += INC_DEC; // Max = 1000000
            if(duty_cycle > PERIOD) duty_cycle = PERIOD;
            pwmPin(direction, PERIOD, duty_cycle, PWM_ENABLE);
            printf("Pressed '%c'!\\n", ch);
            printf("PWM Increase\\n");

            index = 2;
            print_monitor(duty_cycle);
            break;
    }
}

```

C Example (2) - DC Motor

■ Source Code(3)

```
case 'd' : // d : dec
    duty_cycle -= INC_DEC; // Min = 200000
    if(duty_cycle < (PERIOD / 5)) duty_cycle = (PERIOD / 5);

    pwmPin(direction, PERIOD, duty_cycle, PWM_ENABLE);
    printf("Pressed '%c'!\\n", ch);
    printf("PWM Decrease\\n");

    index = 2;
    print_monitor(duty_cycle);
    break;

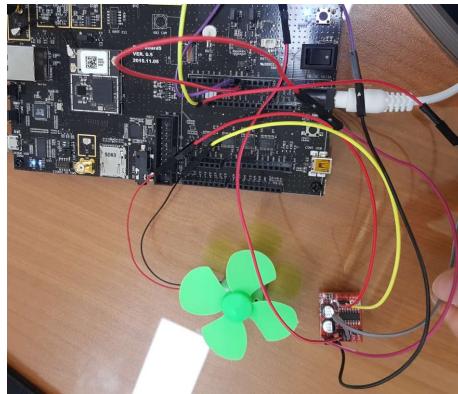
case 'q' : // q : disable & exit
    pwmPin(direction, PERIOD, 0, PWM_DISABLE);
    printf("Pressed '%c'!\\n", ch);
    printf("PWM Disable & Exit\\n");

    index = 0;
    print_monitor(0);
    break;
}
ch = 0;
}
}

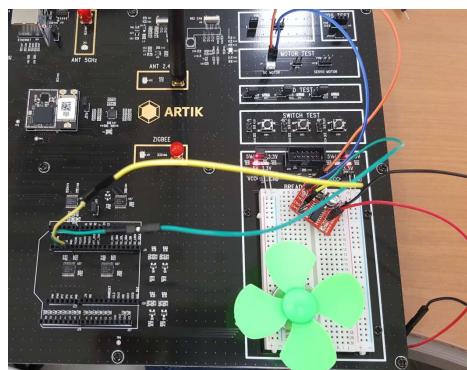
return 0;
}
```

C Example (2) - DC Motor

■ Execution result



ARTIK 520



PS-ED500

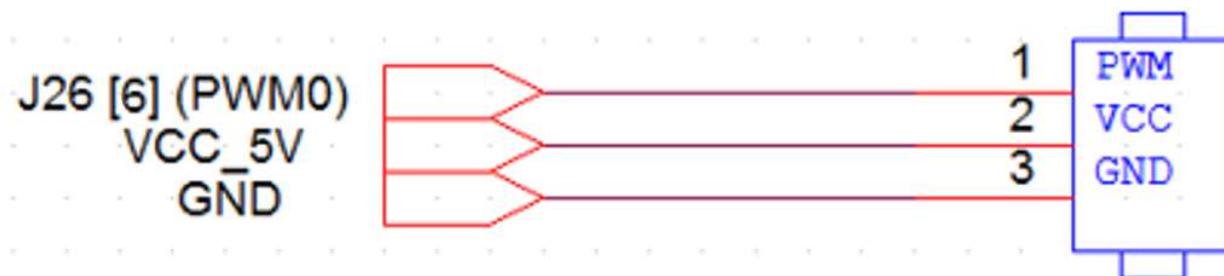
```
[root@localhost motor]# ./motor
duty_value : 0
duty_ratio : 0
===== PWM Test =====
Please keyboard hit!!!!
F : Forward Direction
R : Reverse Direction
Q : Exit
=====
duty_value : 0
duty_ratio : 0
===== PWM Test =====
Please keyboard hit!!!!
E : PWM Enable & init
Q : PWM Disable & exit
I : PWM Increase
D : PWM Decrease
Duty Ratio (now) : 0
=====
Pressed 'i'!
PWM Increase
duty_value : 600000
duty_ratio : 60
===== PWM Test =====
Please keyboard hit!!!!
E : PWM Enable & init
Q : PWM Disable & exit
I : PWM Increase
D : PWM Decrease
Duty Ratio (now) : 60
=====
Pressed 'i'!
PWM Increase
duty_value : 700000
duty_ratio : 70
===== PWM Test =====
Please keyboard hit!!!!
E : PWM Enable & init
Q : PWM Disable & exit
I : PWM Increase
D : PWM Decrease
Duty Ratio (now) : 70
=====
Pressed 'i'!
PWM Increase
duty_value : 800000
duty_ratio : 80
===== PWM Test =====
Please keyboard hit!!!!
```

C Example (3) - Servo Motor

■ Required Hardware

- ARTIK 5 beta developer kit
- Servo Motor (SG90)
- Breadboard
- Connector wires

■ Circuit Configuration



C Example (3) - Servo Motor

■ Source Code

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <unistd.h>
#include "wiringARTIK.h"

#define EN 1
#define DIS 0
#define PERIOD 20000000 //nsec

bool ServoAngle(int pwm_pin, int angle)
{
    int duty_cycle;
    duty_cycle = 1500000 + angle * 8889;
    pwmPin(pwm_pin, PERIOD, duty_cycle, EN);
    printf("duty cycle : %d, angle : %d, duty_cycle, angle);
    return true;
}
```

```
int main(void)
{
    while(1)
    {
        ServoAngle(0,90);
        sleep(1);

        ServoAngle(0,45);
        sleep(1);

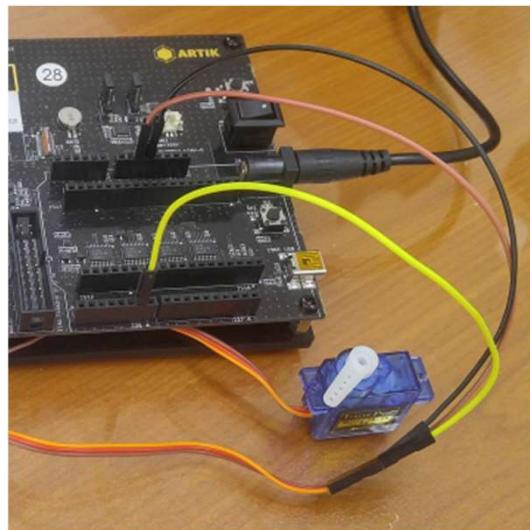
        ServoAngle(0,0);
        sleep(1);

        ServoAngle(0,-45);
        sleep(1);

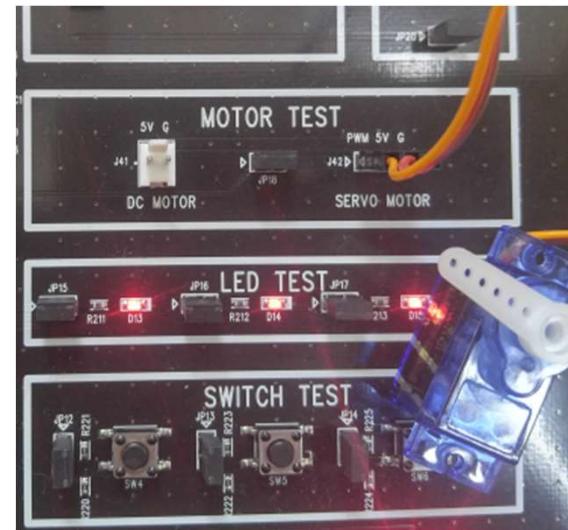
        ServoAngle(0,-90);
        sleep(1);
    }
    return 0;
}
```

C Example (3) - Servo Motor

■ Execution result



ARTIK 520



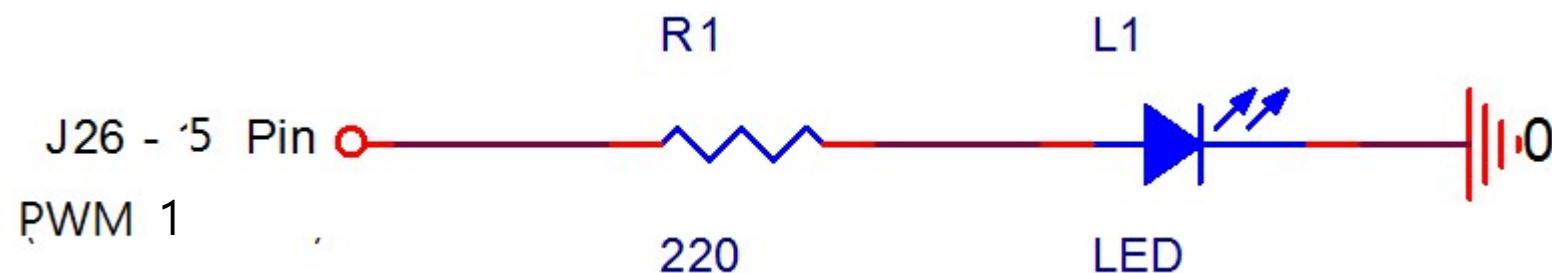
PS-ED500

Using the Arduino IDE

■ Required Hardware

- ARTIK 5 beta developer kit
- DIP type LED
- 220 ohm resistor
- Breadboard
- Connector wires

■ Circuit Configuration



Using the Arduino IDE

■ PWM Pin

Header Pin	Native Mapping ARTIK 520 v0.5	Arduino Pin
J26[5]	PWM1	5
J26[6]	PWM0	6

■ Arduino Source Code

```
LED_PWM
const int thisPin = 5;

void setup() {
}

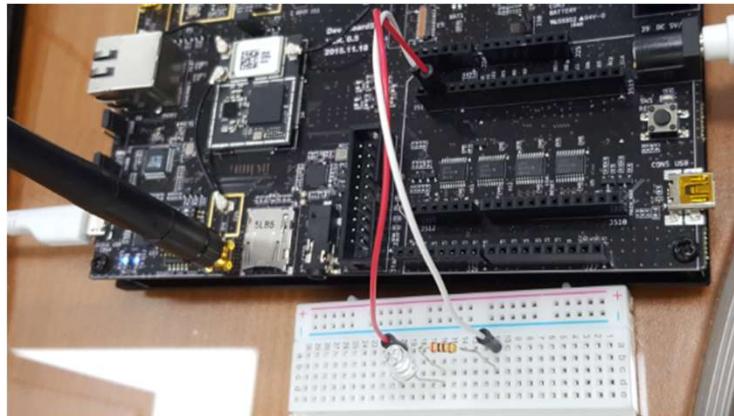
void loop() []
  for (int brightness = 0; brightness < 255; brightness++) {
    analogWrite(thisPin, brightness);
    delay(2);
  }
  for (int brightness = 255; brightness >=0; brightness--) {
    analogWrite(thisPin, brightness);
    delay(2);
  }

  delay(100);
}
```

Using the Arduino IDE

■ Execution result

- Enter [Ctrl]+[u] or use menu to upload the compiled program to ARTIK.
- After uploading, you should see LED slowly on and off.



ARTIK 520



PS-ED500

II. ARTIK 기술 교육

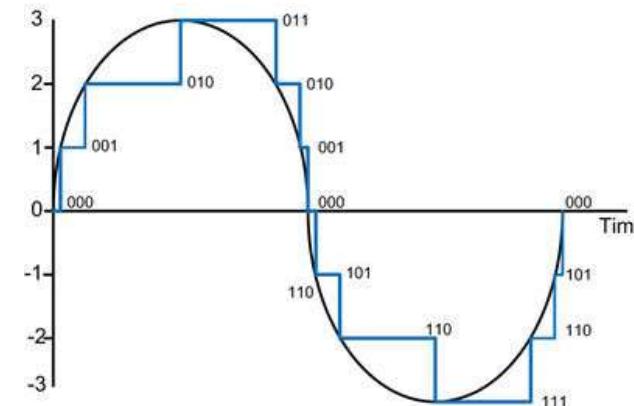
3. Analog Control(1)



Analog / Digital Converter

■ ADC(Analog/Digital Converter)

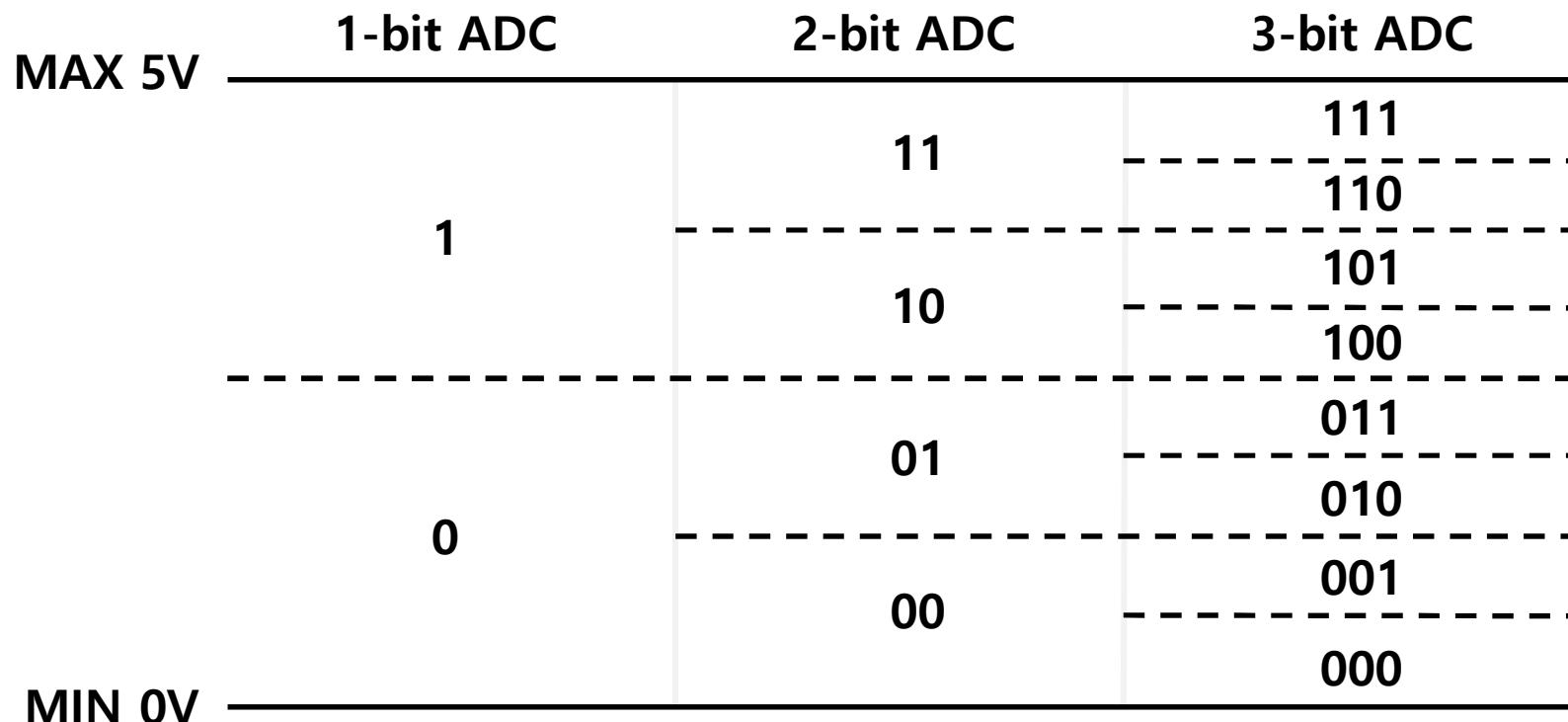
- An apparatus that converts an analog signal (sensor) to digital data (computer, MCU, etc.)
- Depending on the range of the physical quantity to be measured and the application purpose of the system, the one with the appropriate resolution and precision is used.
- Resolution
 - Minimum change of analog input to change digital output value by one class
 - Minimum analog amount that the ADC can represent
 - The minimum data range of the output is $1/2^n$, in case of n-bit ADCs
- Conversion time
 - Time required to perform A / D conversion
 - Expressed in terms of sampling rate per second



Analog / Digital Converter

■ ADC example

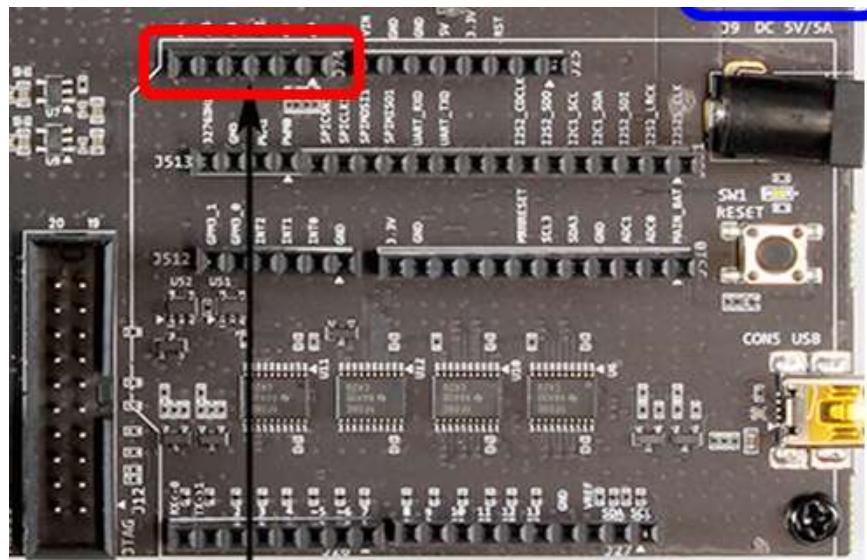
- The ADC interface of ARTIK 520 converts the analog input signal into 12-bit binary code.



ADCs in ARTIK

■ ADC pins

- J24[A0] (Analog input 0)
- J24[A1] (Analog input 1)



(1) ARTIK5 Board

Header J24 (analog inputs)

PIN [SILKSCREEN]	MAPPING
J24[A0]	Analog input 0
J24[A1]	Analog input 1

(2) ARTIK5 Pin mapping(J24)

List of examples

■ Photoresistor(CdS) example

- Using C compiler

■ Infrared(IR) sensor example

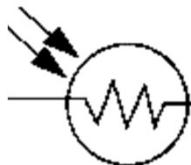
- Using Linux command line
- Using C compiler

■ Humidity/Temperature sensor example

- Using C compiler
- Using Arduino IDE

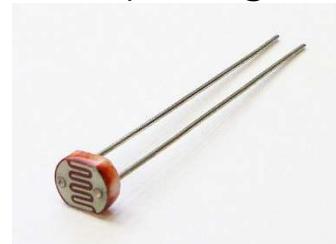
Photoresistor (CdS)

■ Symbol of Photoresistor

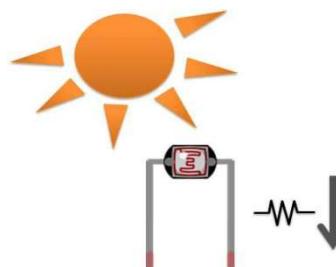


■ CdS Sensor

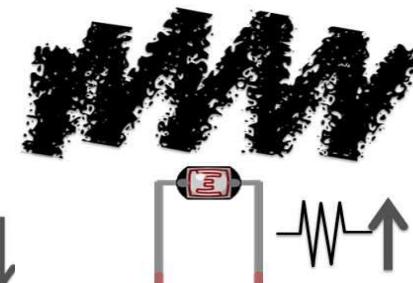
- A sensor varying with the intensity of the light
- Features
 - The photoconductive sensor, which is an N-type semiconductor, diffuses resistance according to the illuminance because the electrons in the N-type semiconductor become free electrons when the light energy is received.
 - Small diameter CdS has a smaller illuminance measurement range
 - The allowable operating temperature range is -30 ~ +60 °C



(1) 조도센서(cds)



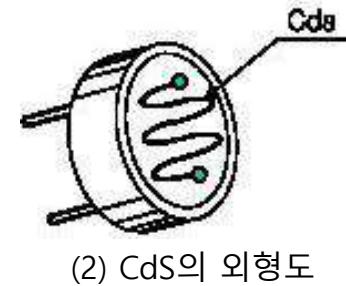
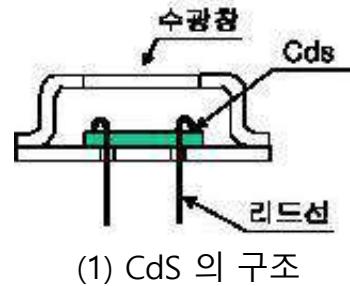
(2) 조도에 따른 저항값의 변화



Photoresistor (CdS)

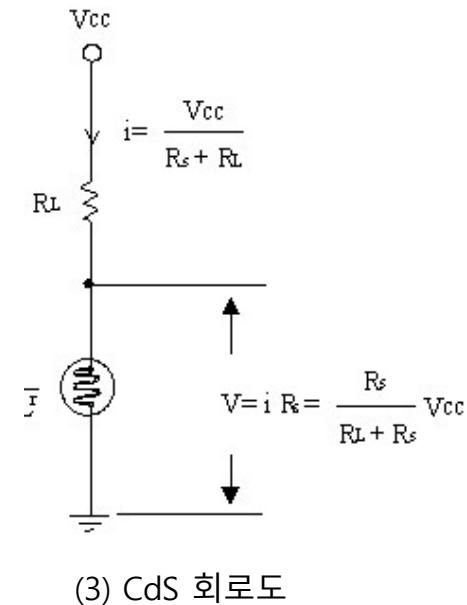
■ Structure of Photoresistor

- CdS is placed in a sealed container, and the light receiving window is made of transparent plastic or glass
- Two leads with no polarity



■ Output voltage

- The CdS resistance (R_s) decreases, the current increases when light comes in.
- CdS resistance(R_s)and load resistance(R_L) is direct connection
- Current $i = \frac{V_{cc}}{R_s + R_L}$
- CdS voltage $V = \frac{R_s}{R_s + R_L} V_{cc}$

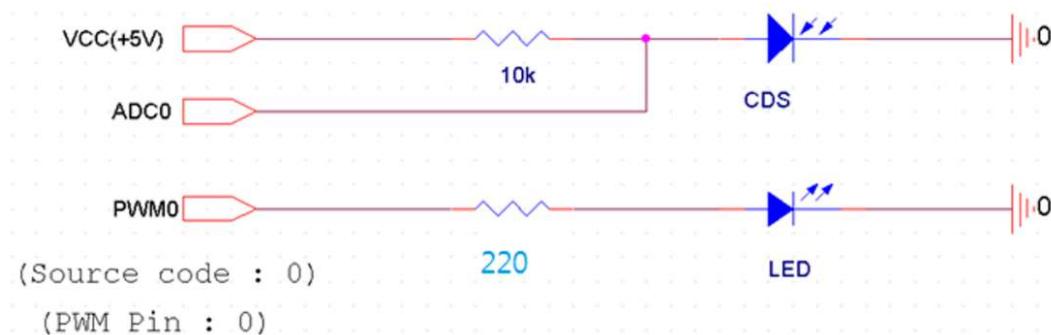


C Compiler Example

■ Required Hardware

- ARTIK 5 beta developer kit
- CdS
- LED
- 10K ohm resistor, 220 ohm resistor
- Breadboard
- Connector wires

■ Circuit Configuration



C Compiler Example

■ Source Code

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <unistd.h>

#define HIGH 1
#define LOW 0

#define ANALOG_PIN 0      // A0
#define PWM_PIN 0

#define PERIOD 1000000

bool pwmPin(int pin, int period, int duty_cycle, int val) {
    FILE * fd;
    char fName[128];

    // Exporting the pin to be used
    fd = fopen("/sys/class/pwm/pwmchip0/export", "w");
    if (!fd) {
        printf("Error: unable to export pin\n");
        return false;
    }
    fprintf(fd, "%d\n", pin);
    fclose(fd);

    // Setting period
    sprintf(fName, "/sys/class/pwm/pwmchip0/pwm%d/period", pin);
    fd = fopen(fName, "w");
    if (!fd) {
        printf("Error: can't open pin direction\n");
        return false;
    }
    fprintf(fd, "%d\n", period);
    fclose(fd);

    // Setting duty cycle
    sprintf(fName, "/sys/class/pwm/pwmchip0/pwm%d/duty_cycle", pin);
    fd = fopen(fName, "w");
    if (!fd) {
        printf("Error: can't open pin direction\n");
        return false;
    }
    fprintf(fd, "%d\n", duty_cycle);
    fclose(fd);

    // Setting value
    sprintf(fName, "/sys/class/pwm/pwmchip0/pwm%d/enable", pin);
    fd = fopen(fName, "w");
    if (!fd) {
        printf("Error: can't open pin direction\n");
        return false;
    }
    fprintf(fd, "%d\n", val);
    fclose(fd);
}
```

C Compiler Example

■ Source Code (2)

```

if (!fd) {
    printf("Error: can't open pin value\n");
    return false;
}
if (val == HIGH) fprintf(fd, "1\n");
else fprintf(fd, "0\n");
fclose(fd);
return true;
}

int analogRead(int pin) {
    FILE * fd;
    char fName[64];
    char val[8];

    // open value file
    sprintf(fName, "/sys/devices/126c0000.adc/iio:device0/
in_voltage%d_raw", pin);
    fd = fopen(fName, "r");
    if (!fd) {
        printf("Error: can't open analog voltage value\n");
        return 0;
    }
    fgets(val, 8, fd);
    fclose(fd);
    return atoi(val);
}

int Con_duty(int duty) {
    pwmPin(PWM_PIN, PERIOD, duty, 1);
}

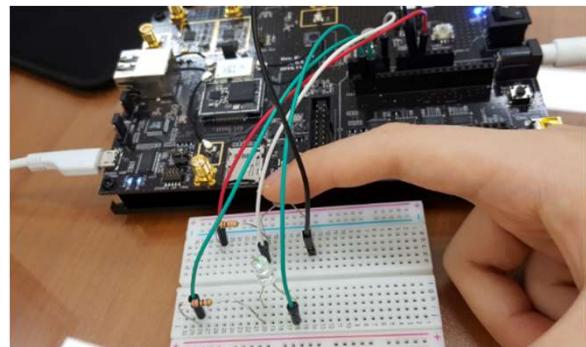
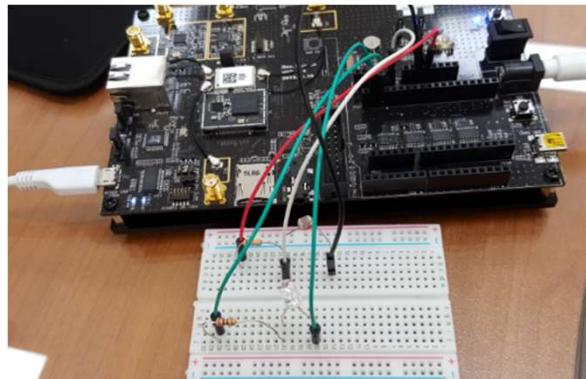
int main(void) {
    long SenVal = 0;
    while(1) {
        SenVal = analogRead(ANALOG_PIN);
        printf("val : %d\n", SenVal);

        if (SenVal > 500) Con_duty(1000000);
        else if (SenVal <= 500) Con_duty(0);
        sleep(1);
    }
    return 0;
}

```

C Compiler Example

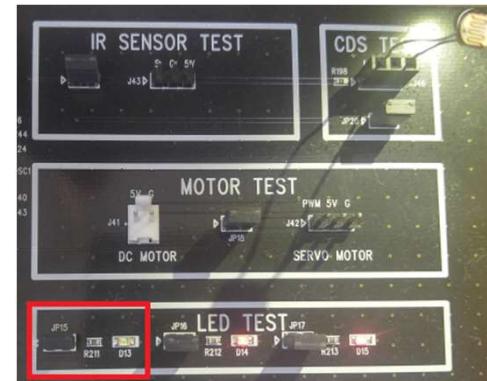
■ Execution result



ARTIK 520

▪ PS-ED500으로 실습 시에는

- LED 핀을 PWM이 아니라 해당 LED가 연결된 GPIO 핀으로 변경
- 마찬가지로 밝기에 따른 ON/OFF 조건문도 Con_duty에서 digitalWrite로 변경

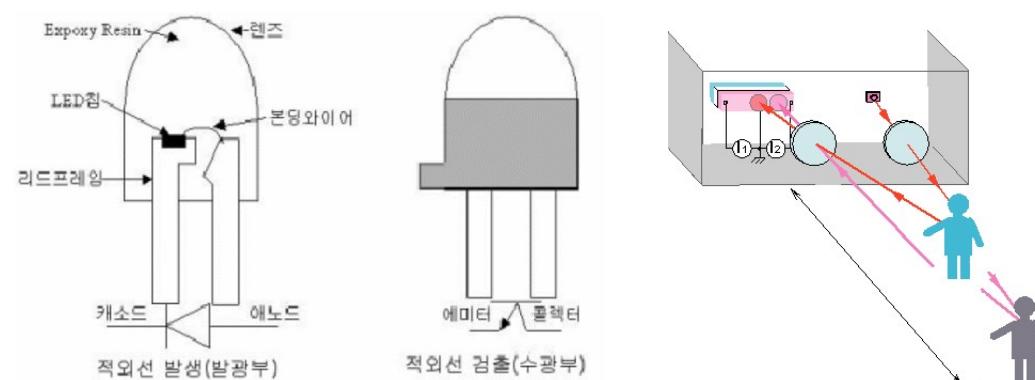
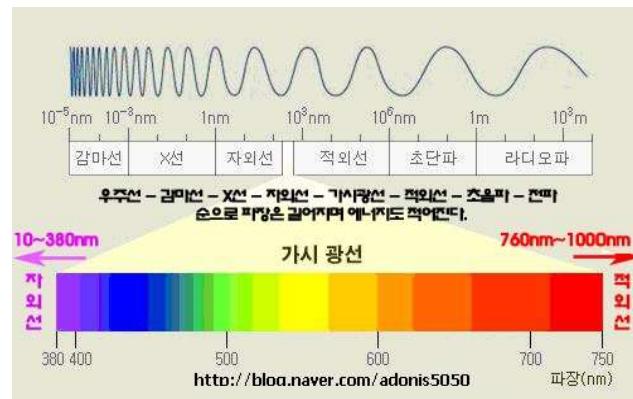


PS-ED500

Infrared(IR) Sensor

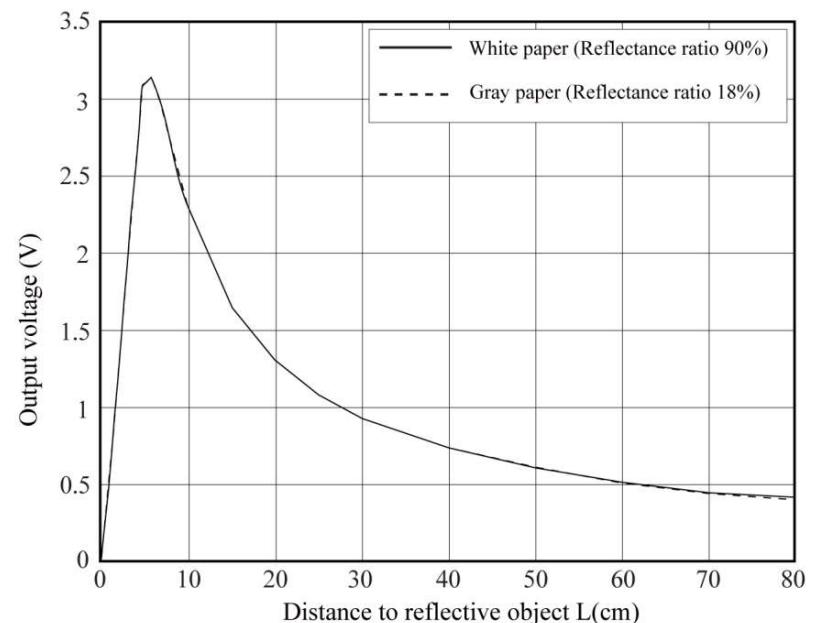
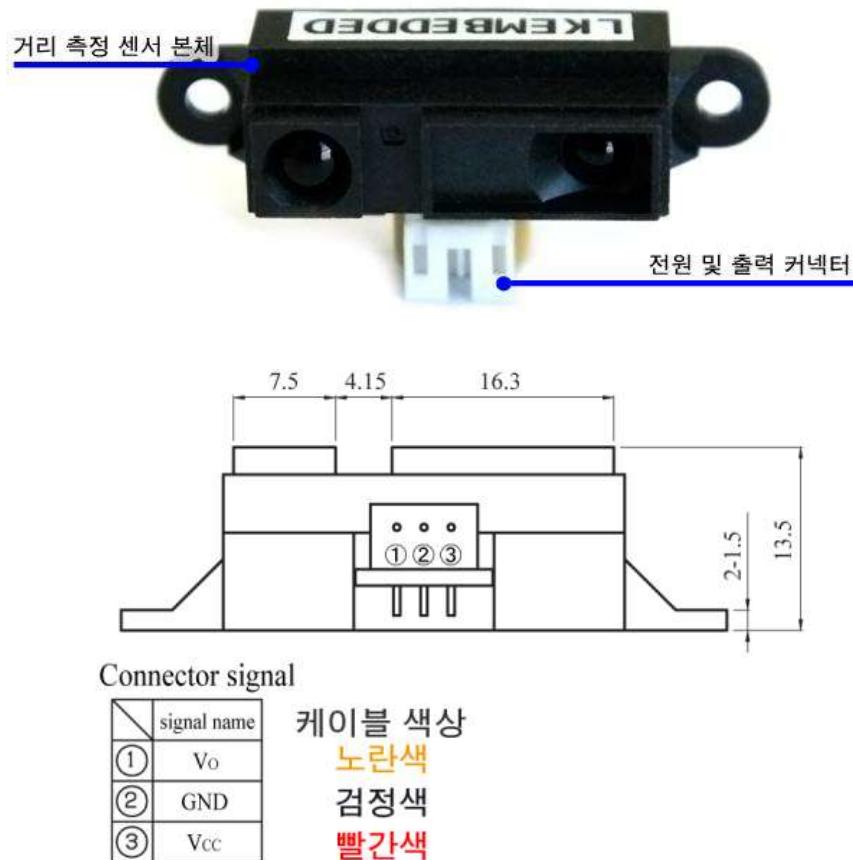
■ Infrared(IR) Sensor

- A device that detects physical quantities and stoichiometries such as temperature, pressure, and intensity of radiation using infrared rays and converts them into electrical quantities capable of signal processing
- In general, conversion formula is needed to measure physical quantities by sensors.
- Sending-light part : Emits light of a certain frequency
- Receiving-light part : Detect diverging light
- Sending-light part → Reflection on an object → Receiving-light part ⇒ Confirm the object existence and calculate the distance



Infrared(IR) Sensor

■ Infrared(IR) Sensor

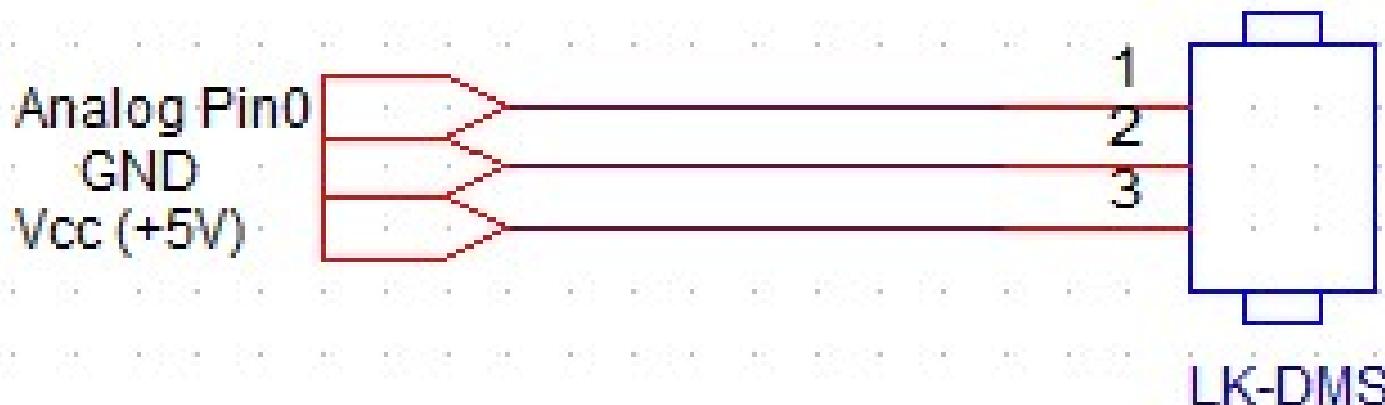


Using Command Line

■ Required Hardware

- ARTIK 5 beta developer kit
- IR sensor (LK-DMS)
- Breadboard
- Connector wires

■ Circuit Configuration



Using Command Line

■ Command

- Read a raw voltage from ADC0 of ARTIK 520
 - *cat /sys/devices/126c0000.adc/iio:device0/in_voltage0_raw*
- Read a raw voltage from ADC1 of ARTIK 520
 - *cat /sys/devices/126c0000.adc/iio:device0/in_voltage1_raw*

```
[root@localhost ~]# cat /sys/devices/126c0000.adc/iio:device0/in_voltage0_raw
2074
[root@localhost ~]# cat /sys/devices/126c0000.adc/iio:device0/in_voltage0_raw
2029
```

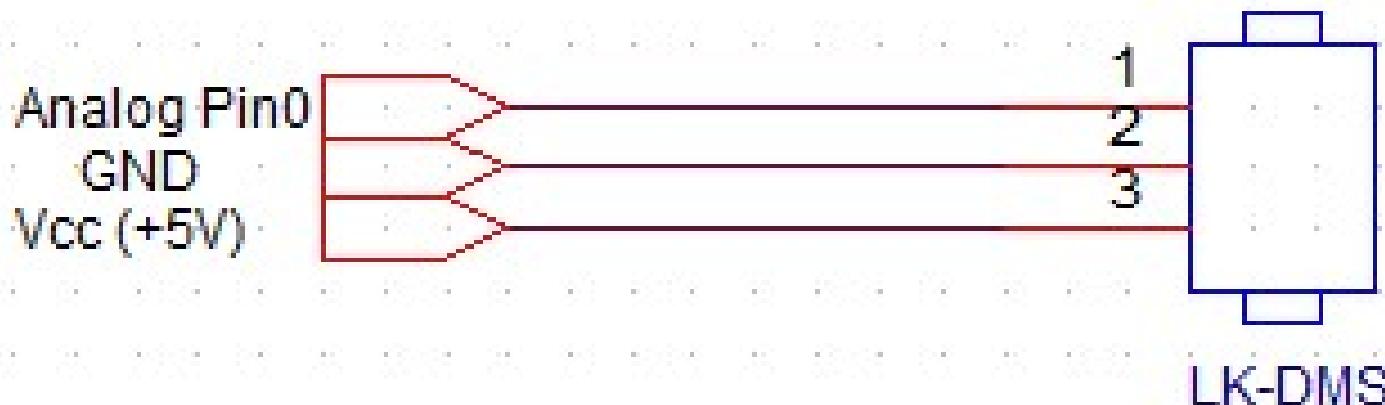
- It returns a raw measurement value, so you should convert it to a voltage measurement. The equation :
 - (Voltage) = (in_voltage X_raw)*0.439453125 [mV], for ARTIK module
 - (Voltage) = (in_voltage X_raw)*0.439453125*2 [mV], for ARTIK development board

C Compiler Example

■ Required Hardware

- ARTIK 5 beta developer kit
- IR sensor (LK-DMS)
- Breadboard
- Connector wires

■ Circuit Configuration



C Compiler Example

■ Source Code

```
#include "wiringARTIK.h"
#include <unistd.h>

#define ANALOG_PIN 0 // A0

int main(void) {
    long SenVal = 0;

    while(1) {
        SenVal = analogRead(ANALOG_PIN);
        printf("val : %d\n", SenVal);
        sleep(1);
    }
}
```

C Compiler Example

■ Execution result



ARTIK 520



PS-ED500

```
[root@localhost ~]# ./ir
val: 520
val: 489
val: 760
val: 1066
val: 2588
val: 2818
val: 2974
val: 486
val: 448
val: 497
val: 522
val: 508
val: 545
```

II. ARTIK 기술 교육

4. Analog Control(2)



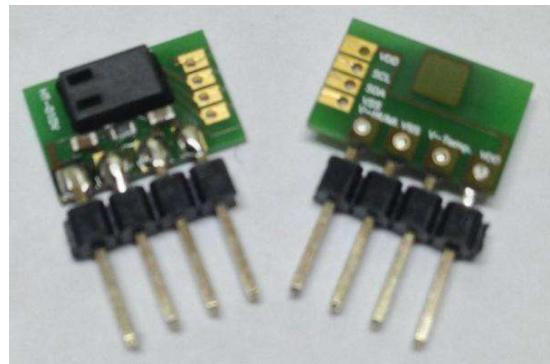
Humidity/Temperature Sensor

■ Humidity/Temperature sensor

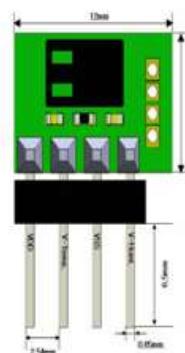
- Humidity sensor : Use the react to humidity and the influence of electric current flow
- Temperature sensor : Use a thermal resistor whose resistance varies with temperature
- Humidity/Temperature sensor : Modular product with temperature sensor and humidity sensor on one board

■ HT-01DV & ETH-01DV

- Integrated sensor of capacitive humidity sensor and band gap temperature sensor
- Convert digital measurement to voltage and output



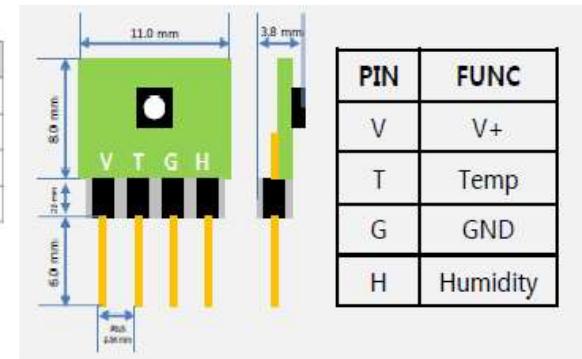
(1) 온/습도 센서(HT-01DV)



(2) 온/습도 센서(HT-01DV) Pin

Name	Pin Function
V-Humi.	RH Voltage Output
VSS	Ground
V-Temp.	Temp Voltage Output
VDD	DC Power

** VDD : 2.3~5.5V
** Pin pitch : 2.54mm



(3) 온/습도 센서(ETH-01DV) Pin

PIN	FUNC
V	V+
T	Temp
G	GND
H	Humidity

Humidity/Temperature Sensor

■ HT-01DV specifications

- Specifications

항목	상대습도			온도		
	Min	Typ	Max	Min	Typ	Max
Resolution	14 bit			14 bit		
Operating Range	0		100%RH	-40 °C		120 °C
Accuracy	Typical		±2.5%RH		±0.5 °C	
	Maximal	Fig. 1 참조		Fig. 2 참조		
Hysteresis	< ±1.5%RH		n/a			
Response Time	10 sec		15 sec		25 sec	

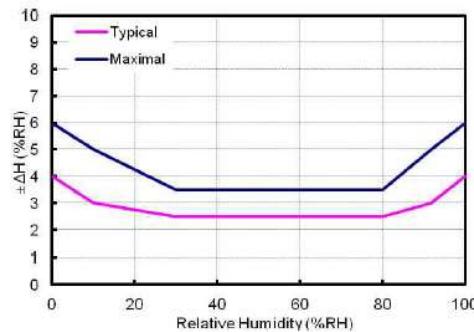


Fig.1

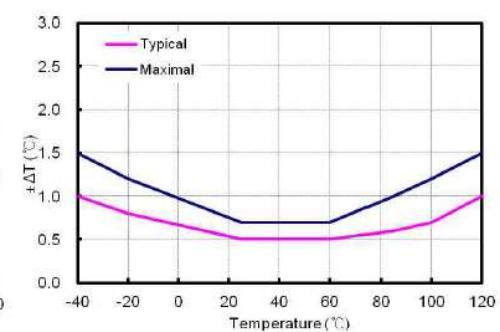
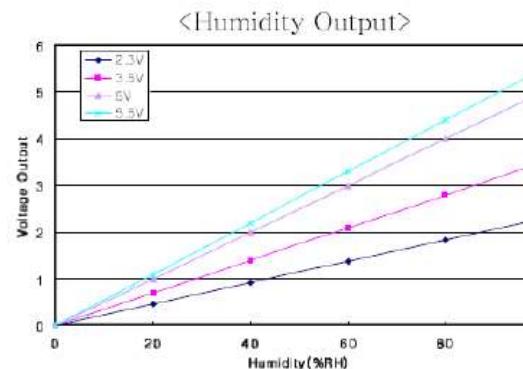


Fig.2

- Signal output

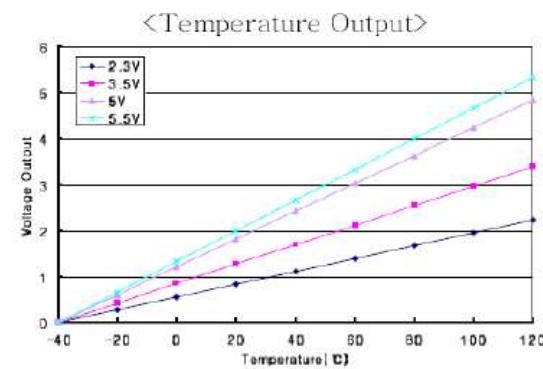
- Change the analog input data to the actual temperature / humidity unit



Equations

$$V_{out} = (V_{in}/100) \times \%RH$$

$$\%RH = V_{out}/(V_{in}/100)$$



Equations

$$V_{out} = (V_{in}/165) \times (°C + 40)$$

$$°C = V_{out}/(V_{in}/165) - 40$$

Humidity/Temperature Sensor

■ ETH-01DV specifications

- Specifications

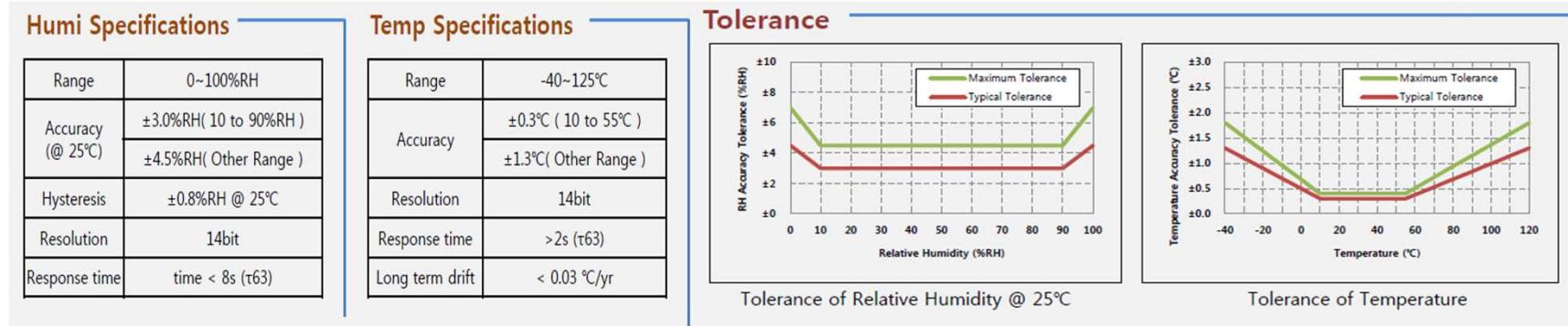


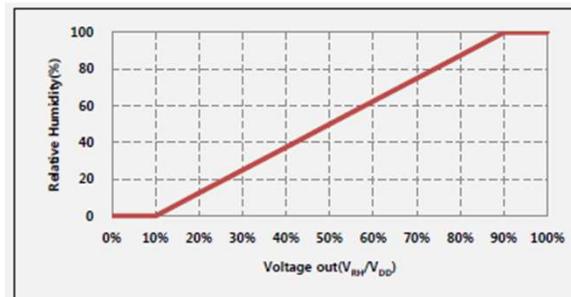
Fig.1

Fig.2

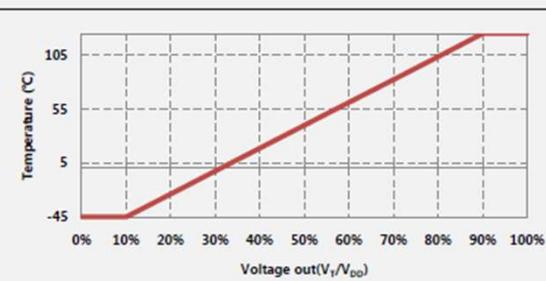
- Signal output

- Change the analog input data to the actual temperature / humidity unit

<Relative Humidity output>



<Temperature output>



Changing point

HT-01DV

ETH-01DV

Output Formula

$$\text{RH}[\%] = (V_{RH} \div V_{DD}) \times 100$$

$$T[^\circ\text{C}] = (V_T \div V_{DD}) \times 165 - 40$$

$$\text{RH}[\%] = (V_{RH} \div V_{DD}) \times 125 - 12.5$$

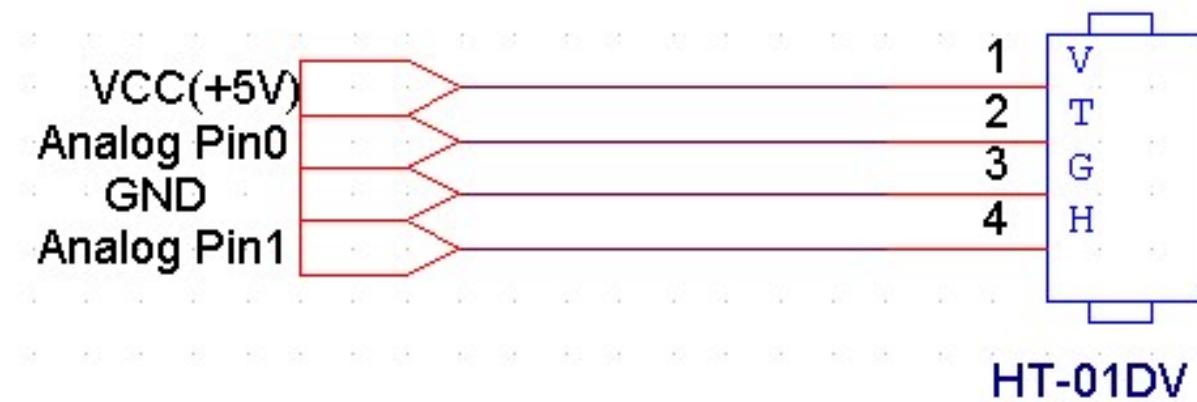
$$T[^\circ\text{C}] = (V_T \div V_{DD}) \times 217.75 - 66.875$$

C Compiler Example

■ Required Hardware

- ARTIK 5 beta developer kit
- Humidity/Temperature sensor (HT-01DV)
- Breadboard
- Connector wires

■ Circuit Configuration



C Compiler Example

■ Source Code

```
#include "wiringARTIK.h"
#include <unistd.h>

#define ANALOG_PIN0 0 // A0
#define ANALOG_PIN1 1 // A1

int main(void) {
    float SenVal1 = 0;
    float SenVal2 = 0;
    float vol1 = 0; // temperature voltage
    float vol2 = 0; // humidity voltage
    float tem = 0;
    float hum = 0;

    while(1) {
        SenVal1 = analogRead(ANALOG_PIN0);
        SenVal2 = analogRead(ANALOG_PIN1);

        vol1 = (SenVal1*0.439453125)*2; // mV단위로 변환
        vol2 = (SenVal2*0.439453125)*2;

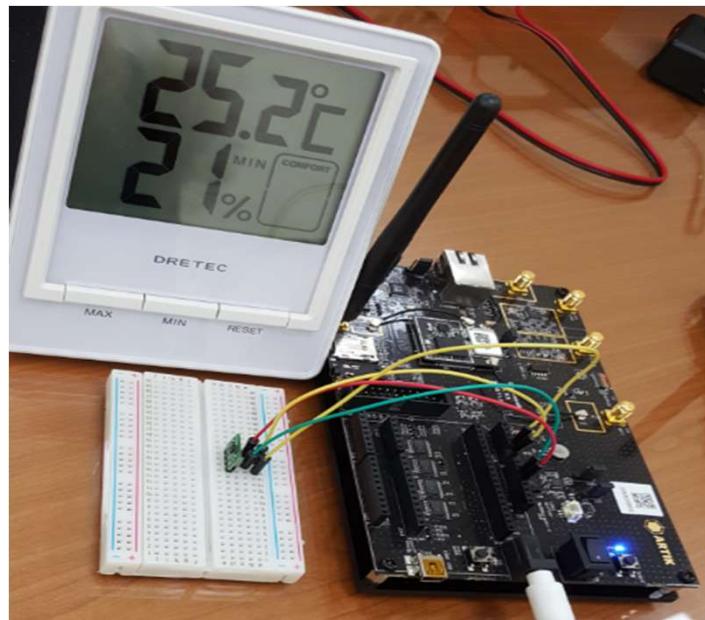
        tem = -66.875 + 218.75*(vol1/5000);
        hum = -12.5 + 125*(vol2/5000);

        printf("=====\\n");
        printf("val1 : %2.2f | temperature : %2.2f\\n", SenVal1, tem);
        printf("val2 : %2.2f | humidity : %2.2f\\n", SenVal2, hum);
        printf("=====\\n");

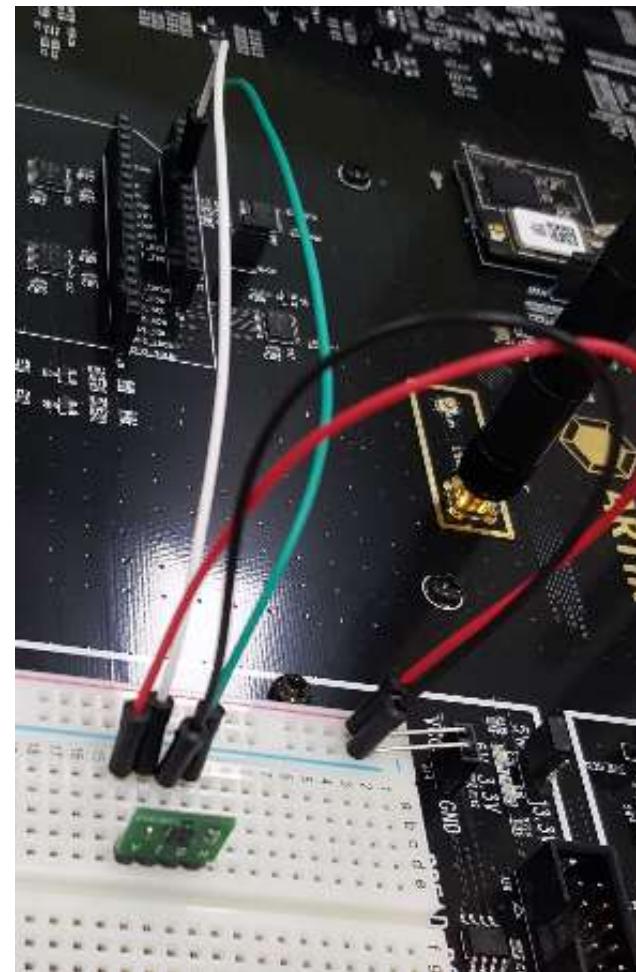
        sleep(1);
    }
    return 0;
}
```

C Compiler Example

■ Execution result



ARTIK 520



PS-ED500

C Compiler Example

■ Execution result

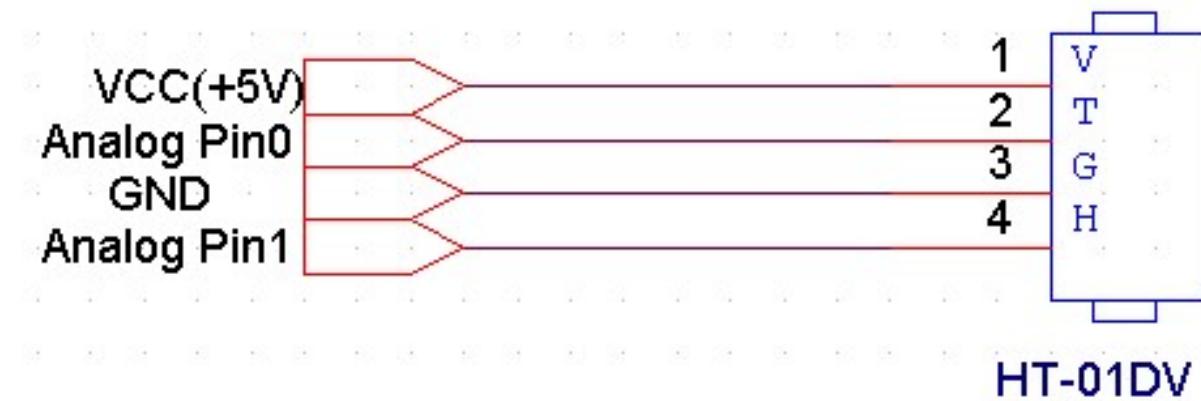
```
[root@localhost ~]# ./sensor
=====
val1 : 2404.00 | temerature : 25.14
val2 : 1535.00 | humidity   : 21.23
=====
=====
val1 : 2404.00 | temerature : 25.14
val2 : 1531.00 | humidity   : 21.14
=====
=====
val1 : 2407.00 | temerature : 25.26
val2 : 1530.00 | humidity   : 21.12
=====
```

Using Arduino IDE

■ Required Hardware

- ARTIK 5 beta developer kit
- Humidity/Temperature sensor (HT-01DV)
- Breadboard
- Connector wires

■ Circuit Configuration



Using Arduino IDE

■ Arduino Source Code

```
HTSensor

#include <DebugSerial.h>

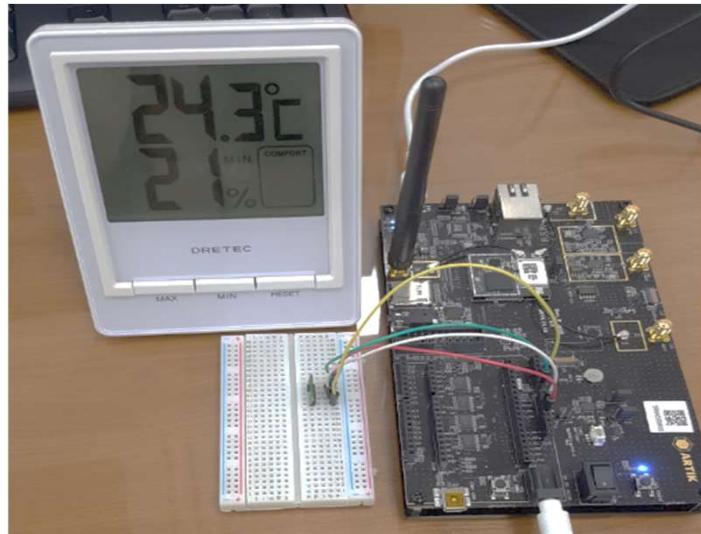
int voltage_raw0, voltage_raw1;
float voltage0, voltage1;
float temperature, humidity;
|
void setup() {
    DebugSerial.begin(9600);
}

void loop() {
    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);

    DebugSerial.print("Temperature : ");
    DebugSerial.println(temperature);
    DebugSerial.print("Humidity : ");
    DebugSerial.println(humidity);
    delay(5000);
}
```

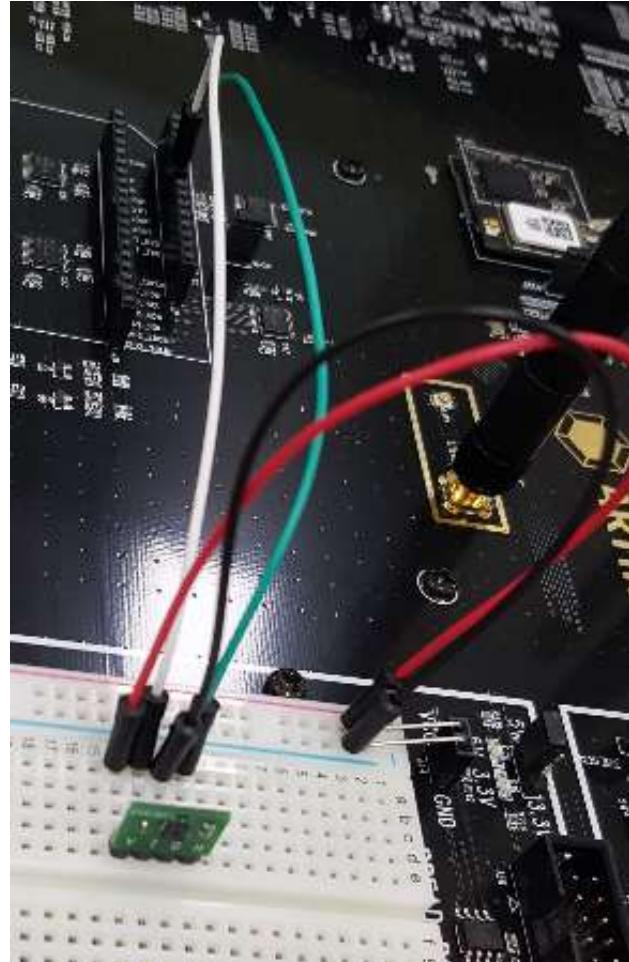
Using Arduino IDE

■ Execution result



ARTIK 520

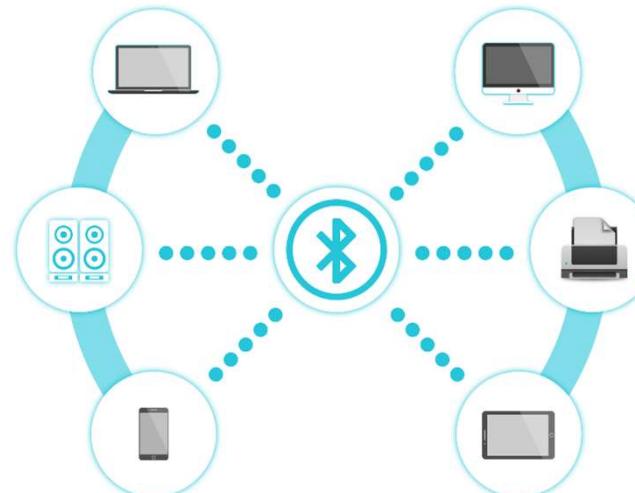
```
Temperature : 24.34
Humidity : 21.56
Temperature : 25.06
Humidity : 22.04
Temperature : 24.45
Humidity : 21.71
Temperature : 25.06
Humidity : 21.93
Temperature : 25.10
Humidity : 22.00
Temperature : 24.41
Humidity : 21.71
```



PS-ED500

II. ARTIK 기술 교육

5. Bluetooth



Bluetooth

■ Bluetooth

- Bluetooth is a small, low-cost, low-power
- Bluetooth is a standard for wireless connectivity in small areas (10 m to 100 m), including smartphones, laptops, and other peripheral device
- In recent years, the Bluetooth Research Group(SIG) released Bluetooth 4.1, which better supports Internet (IoT) and Wearable Device and can coexist with 4G

■ The advantages and disadvantages of Bluetooth

advantages

- Low cost and low power(100mW)
- High security due to separate transmission
- Communicate even if there is obstacle between devices
- No signal attenuation due to angle

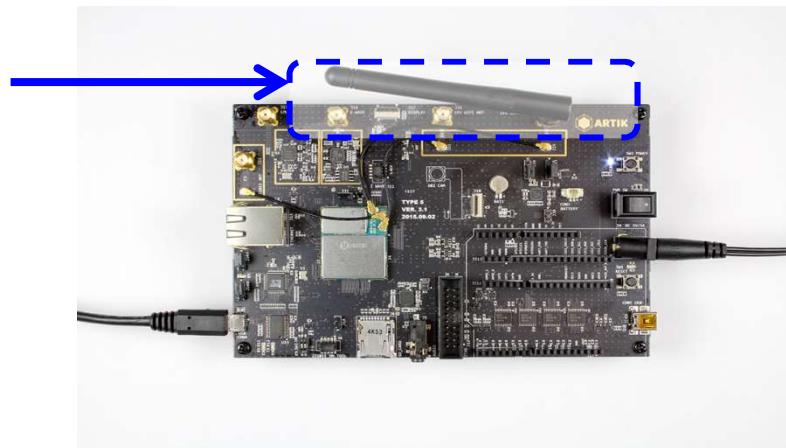
disadvantages

- Bluetooth module needs its own power supply
- Crosstalk during data transmission

Using Bluetooth on ARTIK 5

■ How to pair between ARTIK 5 and other device

- Step 1. Attach a antenna



- Step 2. Input commands

- *# bluetoothctl*
- *# agent on*
- *# default-agent*
- *# scan on*

- Step 3. Check device

```
[root@localhost ~]# bluetoothctl
[NEW] Controller 00:32:44:ED:7D:5A ARTIK5 [default]
[NEW] Device 76:BA:47:15:47:29 76-BA-47-15-47-29
[NEW] Device 3C:86:A8:03:5C:2D 3C-86-A8-03-5C-2D
[NEW] Device 38:01:95:E4:DB:43 [TV] UN60J6350
[bluetooth]# agent on
Agent registered
[bluetooth]# default-agent
Default agent request successful
[bluetooth]# scan on
Discovery started
[CHG] Controller 00:32:44:ED:7D:5A Discovering: yes
[CHG] Device 38:01:95:E4:DB:43 RSSI: -102
[NEW] Device 94:D7:71:EE:DB:18 Summer H (SM-N900K)
```

Using Bluetooth on ARTIK 5

■ How to pair between ARTIK 5 and other device

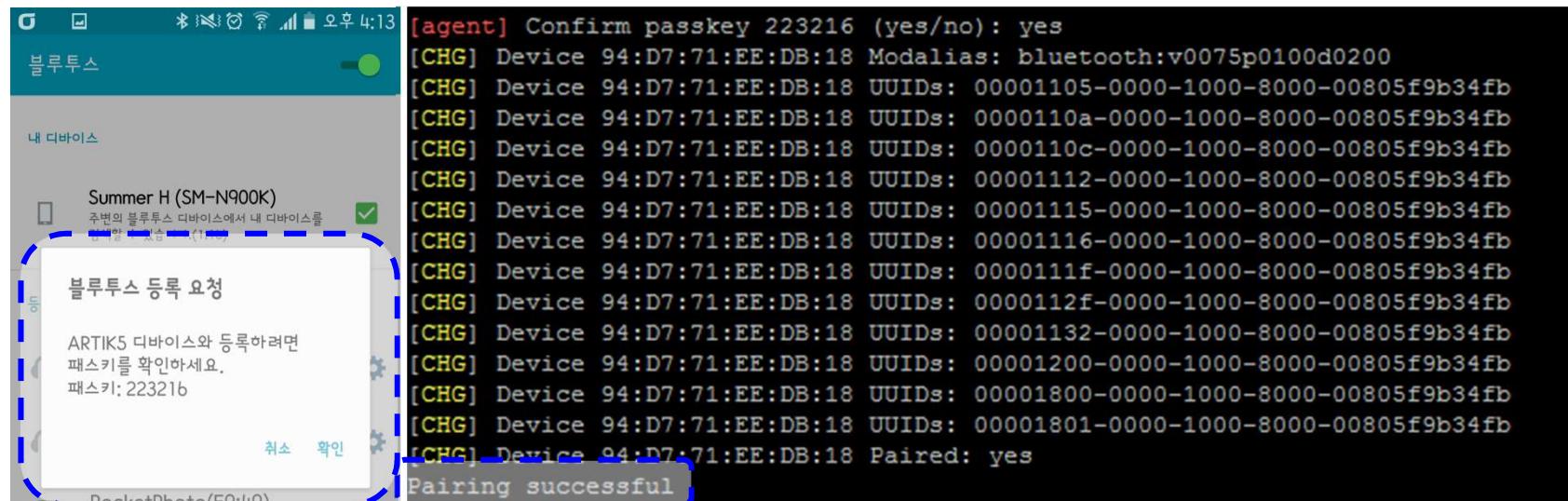
▪ Step 4. Input command

- # pair xx : xx : xx : xx : xx : xx

```
[NEW] Device 94:D7:71:EE:DB:18 Summer H (SM-N900K)
[bluetooth]# pair 94:D7:71:EE:DB:18
Attempting to pair with 94:D7:71:EE:DB:18
[CHG] Device 94:D7:71:EE:DB:18 Connected: yes
```

▪ Step 5. Pairing check

- check ARTIK 5 and device



Using Bluetooth on ARTIK 5

■ Note

- Case of the pairing fail
 - Once you reconnect after pairing Bluetooth
 - ARTIK is paired to another device
- Solution
 - *# remove xx : xx : xx : xx : xx : xx*
 - Return to Step 2

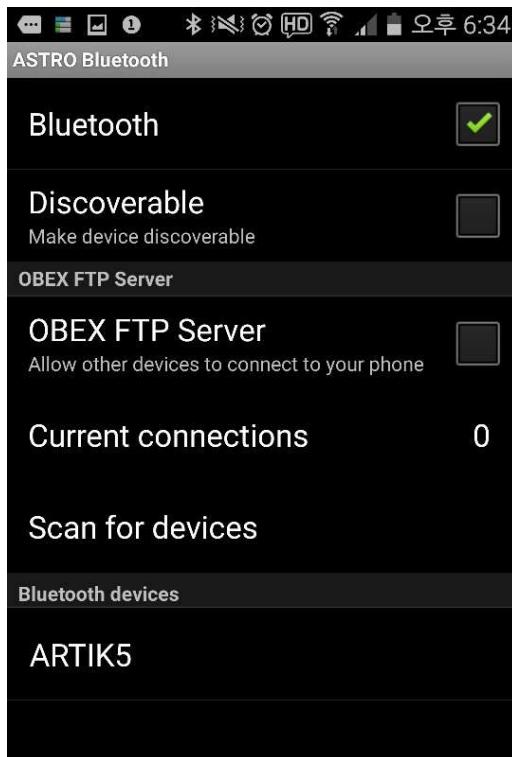
```
Failed to pair: org.bluez.Error.AlreadyExists
[CHG] Device 38:01:95:E4:DB:43 RSSI: -96
[CHG] Device 38:01:95:E4:DB:43 RSSI: -104
[CHG] Device 38:01:95:E4:DB:43 RSSI: -85
[CHG] Device 38:01:95:E4:DB:43 RSSI: -105
[CHG] Device 38:01:95:E4:DB:43 RSSI: -84
[CHG] Device 38:01:95:E4:DB:43 RSSI: -93
[CHG] Device 38:01:95:E4:DB:43 RSSI: -106
[CHG] Device 3C:86:A8:03:5C:2D RSSI: -94
[CHG] Device 38:01:95:E4:DB:43 RSSI: -96
[CHG] Device 38:01:95:E4:DB:43 RSSI: -78
[CHG] Device 38:01:95:E4:DB:43 RSSI: -99
[CHG] Device 94:D7:71:EE:DB:18 RSSI: -52
[CHG] Device 38:01:95:E4:DB:43 RSSI: -88
[CHG] Device 38:01:95:E4:DB:43 RSSI: -99
[CHG] Device 38:01:95:E4:DB:43 RSSI: -91
[CHG] Device 38:01:95:E4:DB:43 RSSI: -102
[CHG] Device 94:D7:71:EE:DB:18 RSSI: -60
[bluetooth]# remove 94:D7:71:EE:DB:18
```

Bluetooth(Obex FTP – BT File Transfer)

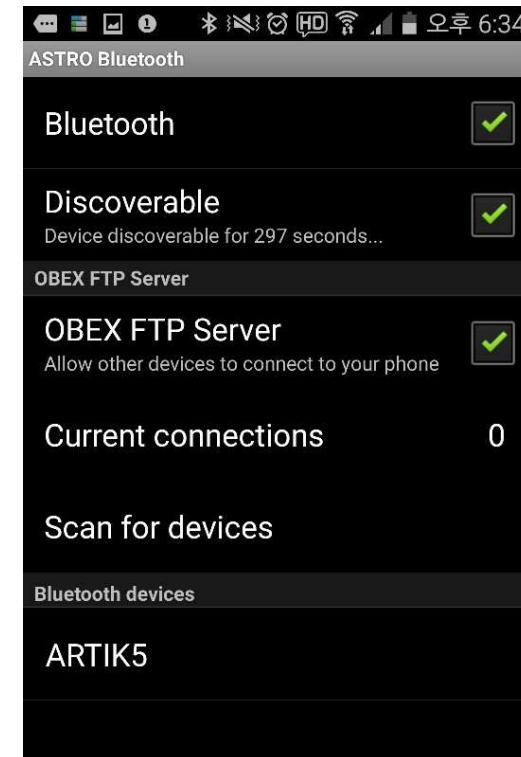
■ Download the ASTRO Bluetooth App & Settings



→ Download



Bluetooth
Discoverable
OBEX FTP Server
Check



Bluetooth(Obex FTP – BT File Transfer)

■ On the ARTIK Board

ARTIK \longleftrightarrow Phone
Pairing

```
[root@localhost test]# bluetoothctl
[NEW] Controller 00:EB:5D:B0:28:DF ARTIK5 [default]
[bluetooth]# scan on
Discovery started
[CHG] Controller 00:EB:5D:B0:28:DF Discovering: yes
[NEW] Device 94:D7:71:EE:DB:18 ○ L ○ (Galaxy Note3)
[bluetooth]# discoverable on
Changing discoverable on succeeded
[bluetooth]# pair 94:D7:71:EE:DB:18
Attempting to pair with 94:D7:71:EE:DB:18
[CHG] Device 94:D7:71:EE:DB:18 Connected: yes
[CHG] Device 94:D7:71:EE:DB:18 Modalias: bluetooth:v0075p0100d0200
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 00001105-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 0000110a-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 0000110c-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 00001112-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 00001115-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 00001116-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 0000111f-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 0000112f-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 00001132-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 00001200-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 00001800-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 00001801-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 Paired: yes
Pairing successful
[bluetooth]# exit
[DEL] Controller 00:EB:5D:B0:28:DF ARTIK5 [default]
```

Bluetooth(Obex FTP – BT File Transfer)

■ List folders on phone

#obexftp -b mac address of phone -l

```
[root@localhost test]# obexftp -b 94:D7:71:EE:DB:18 -l
Browsing 94:D7:71:EE:DB:18 ...
Connecting..done
Receiving "(null)"...<!DOCTYPE folder-listing SYSTEM "obex-folder-listing.dtd">
<folder-listing version="1.0">
<folder name=".face" modified="20160824T125956Z" user-perm="RW"/>
<folder name="Android" modified="20160726T123204Z" user-perm="RW"/>
<folder name="Music" modified="20140101T090600Z" user-perm="RW"/>
<folder name="Podcasts" modified="20140101T090600Z" user-perm="RW"/>
<folder name="Ringtones" modified="20160726T120456Z" user-perm="RW"/>
<folder name="Alarms" modified="20140101T090600Z" user-perm="RW"/>
<folder name="Notifications" modified="20160726T123034Z" user-perm="RW"/>
<folder name="Pictures" modified="20160824T120809Z" user-perm="RW"/>
<folder name="Movies" modified="20160727T051224Z" user-perm="RW"/>
<folder name="Download" modified="20160818T031505Z" user-perm="RW"/>
<folder name="DCIM" modified="20160824T121108Z" user-perm="RW"/>
<folder name="olleh_market" modified="20160726T120442Z" user-perm="RW"/>
<folder name=".SpenSDK30" modified="20160726T120453Z" user-perm="RW"/>
<file name=".enref" modified="20160726T120500Z" user-perm="RW" size="15"/>
<folder name="msc" modified="20160814T115758Z" user-perm="RW"/>
<folder name="iconnect" modified="20160728T013559Z" user-perm="RW"/>
<folder name="KakaoTalk" modified="20160817T011123Z" user-perm="RW"/>
<folder name="data" modified="20160728T013641Z" user-perm="RW"/>
<folder name="Playlists" modified="20160726T014914Z" user-perm="RW"/>
<folder name="melon" modified="20160811T092320Z" user-perm="RW"/>
<folder name=".System_CPO" modified="20160726T071035Z" user-perm="RW"/>
6<file name=".cache.idl" modified="20160825T041350Z" user-perm="RW" size="5967"/>
<folder name="inappad" modified="20160813T035635Z" user-perm="RW"/>
<folder name="igaw" modified="20160822T054439Z" user-perm="RW"/>
<folder name="System" modified="20160812T042504Z" user-perm="RW"/>
<folder name="WhoWho" modified="20160726T040347Z" user-perm="RW"/>
<file name=".cache.ble" modified="20160824T062931Z" user-perm="RW" size="32"/>
<folder name="WDM" modified="20160726T040828Z" user-perm="RW"/>
<folder name="nelo" modified="20160726T065741Z" user-perm="RW"/>
<folder name="NPKI" modified="20160726T053714Z" user-perm="RW"/>
<folder name="NPUSH" modified="20160726T065744Z" user-perm="RW"/>
<folder name="pino" modified="20160726T072016Z" user-perm="RW"/>
<folder name=".thumbnails" modified="20160727T024652Z" user-perm="RW"/>
<folder name="ZW4U99SQQ3.com.linecorp.trident.shared" modified="20160727T050715Z" user-perm="RW"/>
<folder name="com.facebook.orca" modified="20160727T073448Z" user-perm="RW"/>
<folder name="Pikicast" modified="20160812T051634Z" user-perm="RW"/>
<folder name="MyRemocon" modified="20160802T044532Z" user-perm="RW"/>
<folder name="Naver" modified="20160816T033351Z" user-perm="RW"/>
<folder name=".ValuePotion" modified="20160807T074117Z" user-perm="RW"/>
<folder name="DioDict4B" modified="20160808T034610Z" user-perm="RW"/>
<folder name="video" modified="20160810T040712Z" user-perm="RW"/>
<folder name="SNOW" modified="20160812T112110Z" user-perm="RW"/>
<folder name="Uva_Camera" modified="20160822T062849Z" user-perm="RW"/>
<folder name="LazyList" modified="20160817T050530Z" user-perm="RW"/>
<file name=".profif.os" modified="20160820T093123Z" user-perm="RW" size="36"/>
<folder name="KakaoTalkDownload" modified="20160824T125627Z" user-perm="RW"/>
</folder-listing>done
Disconnecting..done
```

Bluetooth(Obex FTP – BT File Transfer)

■ Source code(1)

```
#include "wiringARTIK.h"
#include <stdlib.h>

#define SHV_E300S      "BC:44:86:AB:EB:F6"
#define FILE_PATH      "/root/BT/BT_FILE.txt"

void BT_import_file(){
    FILE *fd;
    char val[20];

    system("ovexftp -b BC:44:86:AB:EB:F6 -q BT FILE.txt");
    sleep(1);

    if((fd = fopen(FILE_PATH, "r")) == NULL){
        printf("Error : can't open file\n");
        return 1;
    }

    fgets(val, 20, fd);
    fclose(fd);

    printf("%s\n", val);
}
```

```
void BT_export_file(){
    FILE *fd;
    char val[20];

    scanf("%s\n", &val);

    if((fd = fopen(FILE_PATH, "w+")) == NULL){
        printf("Error : can't open file\n");
        return 1;
    }

    rewind(fd);
    fprintf(fd, val);
    fclose(fd);
    system("obexftp -b BC:44:86:AB:EB:F6 -p BT FILE.txt");

    printf("\n");
}
```

Bluetooth(Obex FTP – BT File Transfer)

■ Source code(2)

```
void main(){
    int loop = 1;
    int read_write = 0;

    printf("=====\\n");
    printf("r : read // w : write // q : quit \\n")
    printf("=====\\n");

    while(loop){
        if(kbhit()){
            read_write = getchar();
        }
        switch(read_write){
            case 'r':           // r : read
                BT_import_file();
                break;
            case 'w':           // w : write
                BT_export_file();
                break;
            case 'q':           // q : quit
                loop = 0;
                break;
        }
    }
}
```

<Download Files>

#obexftp -b mac_address_of_phone -g file_name

<Upload Files>

#obexftp -b mac_address_of_phone -p file_name

<Options>

- b : Bluetooth device
- g : Send file from Phone to ARTIK
- p : Send file from ARTIK to Phone

txt, jpg, wmv ...

Bluetooth(Obex FTP – BT File Transfer)

■ Bluetooth File Transfer (→ Upload File)

- <On the ARTIK Board>

```
[root@localhost test]# gcc -o bt bt.c
[root@localhost test]# ls
bt  bt.c  wiringARTIK.h
[root@localhost test]# ./bt
=====
r : read // w : write // q : quit
=====
data : 12345
Browsing 94:D7:71:EE:DB:18 ...
Connecting..done
Sending "BT FILE.txt".../done
Disconnecting..-done

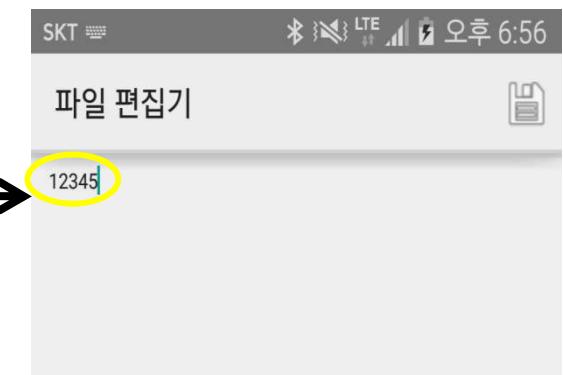
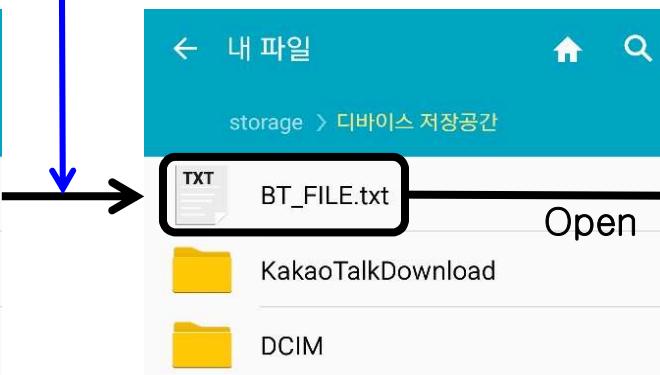
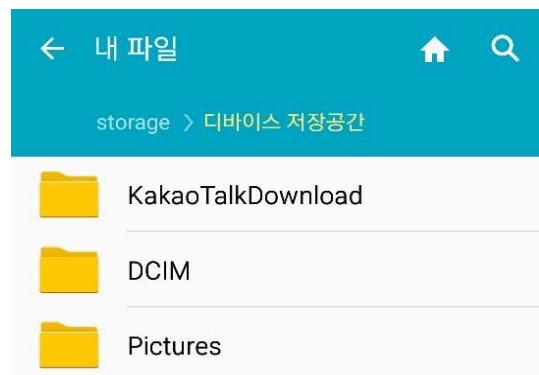
[root@localhost test]# ls
bt  bt.c  BT_FILE.txt  wiringARTIK.h
[root@localhost test]# cat BT_FILE.txt
12345[root@localhost test]#
```

‘w’ key press

Created file

- <On the Phone>

ARTIK → Phone



Bluetooth(Obex FTP – BT File Transfer)

■ Bluetooth File Transfer (→ Download File)

- <On the ARTIK Board>

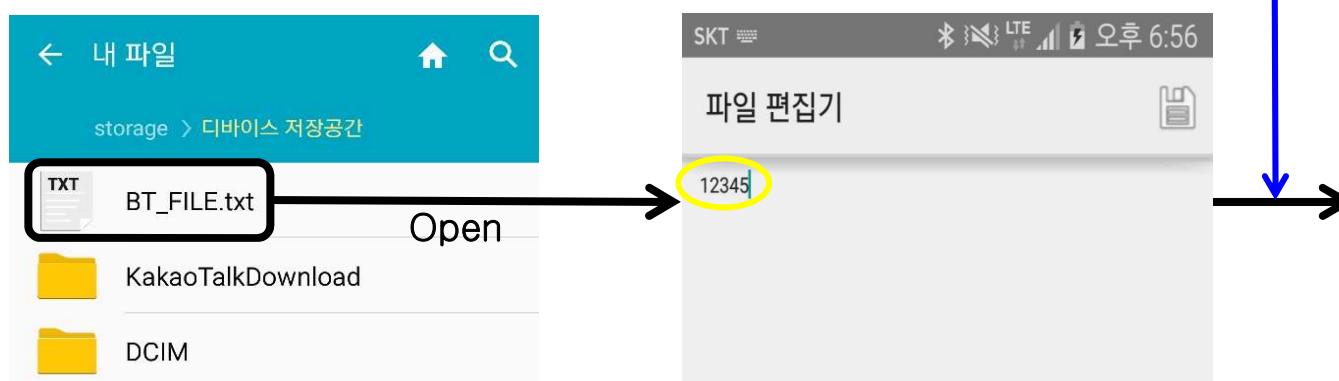
```
[root@localhost test]# vi BT_FILE.txt
123 ←
~ '123' modified
"BT FILE.txt" 1L, 4C written
[root@localhost test]# ./bt
=====
r : read // w : write // q : quit
=====
Browsing 94:D7:71:EE:DB:18 ...
Connecting..\qdone
Receiving "BT FILE.txt"...\done
Disconnecting..|done
12345
[root@localhost test]# cat BT_FILE.txt
12345[root@localhost test]#
```

'123' modified

'r' key press

overwrite

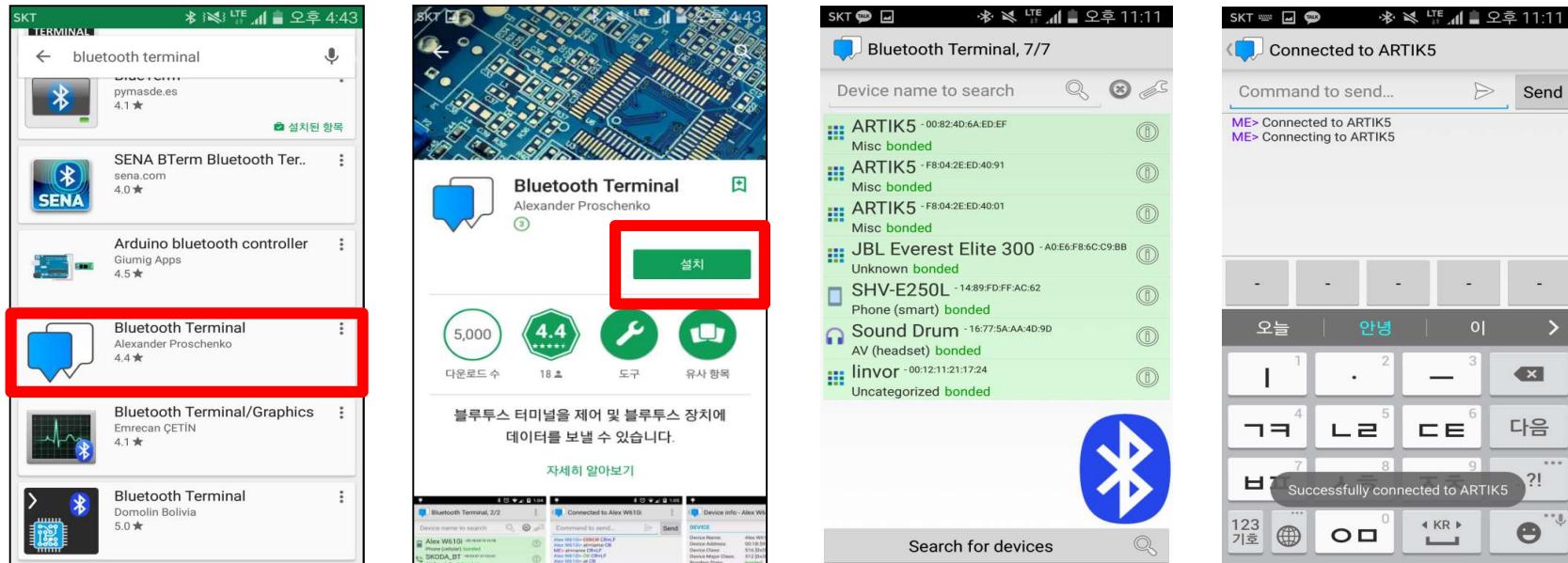
- <On the Phone>



Bluetooth(Data Transfer)

■ How to pair between ARTIK6 and Bluetooth App

- Step 1: Download Bluetooth Terminal app
 - Development Company: Alexander Proshchenko
 - Start the App
 - Click the ARTIK 5



Bluetooth(Data Transfer)

■ How to pair between ARTIK6 and Bluetooth App

- Step 2: Confirm ARTIK 5 connection

```
[root@localhost ~]# rfcomm listen hci0&
[1] 1936
[root@localhost ~]# Waiting for connection on channel 1
Connection from BC:44:86:AB:EB:F6 to /dev/rfcomm0
Press CTRL-C for hangup
[
```

- Press 'Enter'

```
[root@localhost ~]# Waiting for connection on channel 1
Connection from BC:44:86:AB:EB:F6 to /dev/rfcomm0
Press CTRL-C for hangup

[root@localhost ~]# [
```

Bluetooth(Data Transfer)

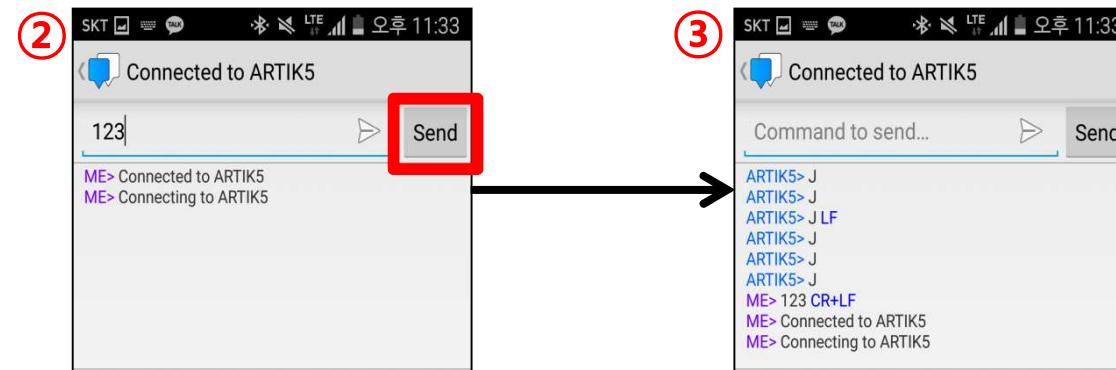
■ How to pair between ARTIK6 and Bluetooth App

- Step 3: Checking the transferred data

- <ARTIK 5> # cat /dev/rfcomm0
- <App> Send the data (Ex.123)

①

```
[root@localhost ~]# cat /dev/rfcomm0
```



④

```
[root@localhost ~]# cat /dev/rfcomm0
```

```
123
```

II. ARTIK 기술 교육

6. Wi-Fi



■ Wi-Fi(Wireless Fidelity)

- Where a wireless access point is installed , A wireless local area network (WLAN) capable of wireless Internet access within a certain distance using radio waves or infrared transmission
- In 1997, the Institute of Electrical and Electronics Engineers (IEEE) standardized Wi-Fi as IEEE 802.11

■ The advantages and disadvantages of Wi-Fi

advantages

- Compatibility is high due to the generalization of Wi-Fi
- High Data transmission speed
- AP installation is easy and installation cost is low
- Various versions such as a, b, g, n (IEEE 802.11)



disadvantages

- Security risk due to simultaneous access of multiple devices to one AP
- Wireless Internet is available only near AP
- Relatively low communication range
- Interference between AP radio waves

Wi-Fi in ARTIK

■ Wi-Fi

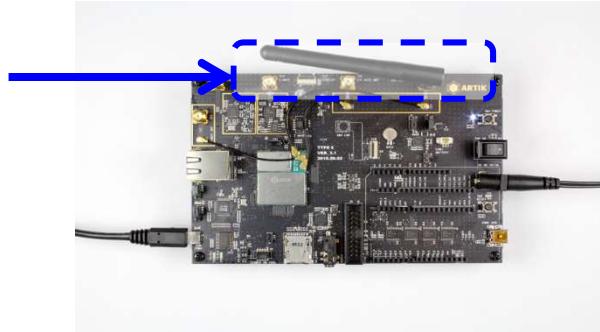
- Client : Mode to connect to an Internet-connected AP
Required for most development work.
- Access Point : Mode to connect to Internet through Ethernet LAN in ARTIK.
Enables wireless connection of other devices
- Direct or Wi-Fi Point-to-Point (P2P) : Mode to connect to other Wi-Fi client devices



Using Wi-Fi on ARTIK 5

■ How to connect to WiFi

- Step 1. Attach a antenna



- Step 2. Scan for wireless access points
 - *# wpa_cli scan results*

- *# wpa_cli scan_results*

- #### ▪ Step 3. Configure wpa_supplicant.conf

- ```
• # wpa_passphrase "SSID" "PASSWORD" >> /etc/wpa_supplicant/wpa_supplicant.conf
```

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

# Using Wi-Fi on ARTIK 5

## ■ How to connect to WiFi

- Step 4. Restart wpa\_supplicant

- *# systemctl restart wpa\_supplicant*

- Step 5. Get an IP address

- *# dhclient wlan0*
  - *# ifconfig*

```
[root@localhost ~]# ifconfig
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
 inet 192.168.0.66 netmask 255.255.255.0 broadcast 192.168.0.255
 ether e8:50:9b:94:b0:79 txqueuelen 1000 (Ethernet)
 RX packets 1152 bytes 71245 (69.5 KiB)
 RX errors 0 dropped 9 overruns 0 frame 0
 TX packets 17 bytes 2103 (2.0 KiB)
 TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
[root@localhost ~]#
```

# WiFi AP Realization

## ■ Introduction

Wireless or wired networks are the most important meanings for ARTIK to communicate with other devices. We can configure the network by connecting Ethernet to the wire or connecting to Wi-Fi, where ARTIK can act as a Wi-Fi access point as well as a Wi-Fi client. There have been a detailed SoftAP mode in ARTIK.

# Methodology

## ■ Wi-Fi Access Point setting

Step 1. Connect the WiFi antenna in the lower left (ANT1) of ARTIK

Step 2. dnsmasq(DNS Masquerade server) setting

- Enter `vi /etc/dnsmasq.conf` to input the following command in the file or uncomment the command, which sets the range of IP addresses for DHCP allocation.

`bind-interfaces`

`dhcp-range=192.168.1.2,192.168.1.100`

- Interrupt connman(connection manager) service.

`# systemctl stop connman`

- Change the settings of the network driver. If you enter the following command in order, you can see that val = 0 changes to val = 1.

`modprobe : used to add or remove modules`

`# modprobe -r dhd`

`# modprobe dhd_mode=2`

```
[root@localhost wpa_supplicant]# modprobe -r dhd
[489.199935] dhd_wlan_set_carddetect: notify_func=c04bd124, mmc
_host_dev=d8ed5410, val=0
[root@localhost wpa_supplicant]# modprobe dhd op_mode=2
[493.074776] dhd_wlan_set_carddetect: notify_func=c04bd124, mmc
_host_dev=d8ed5410, val=1
```

# Methodology

## ■ Wi-Fi Access Point setting

- Set the IP address of Wlan0

```
ifconfig wlan0 192.168.1.1 up
```

```
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
 inet 192.168.1.1 netmask 255.255.255.0 broadcast 192.168.1.255
 inet6 fe80::ee1f:72ff:fed5:1995 prefixlen 64 scopeid 0x20<link>
 ether ec:1f:72:d5:19:95 txqueuelen 1000 (Ethernet)
 RX packets 0 bytes 0 (0.0 B)
 RX errors 0 dropped 3 overruns 0 frame 0
 TX packets 29 bytes 3225 (3.1 KiB)
 TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

We can see that the IP address you entered is assigned to wlan0 when entering the command.

- Dnsmasq Start

```
dnsmasq -C /etc/dnsmasq.conf
```

# Methodology

## ■ Wi-Fi Access Point setting

Step 3. iptables setting

- iptables : a kind of firewall, usually software that enables NAT (Network Address Translation) used by Linux.

Enter the following command to change the configuration of iptables

```
sysctl net.ipv4.ip_forward=1
iptables --flush
iptables -t nat --flush
iptables --delete-chain
iptables -t nat --delete-chain
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
iptables -A FORWARD -i wlan0 -j ACCEPT
```

# Methodology

## ■ Wi-Fi Access Point setting

### Step 4. hostapd setting

- Enter `vi /etc/hostapd/hostapd.conf` and then enter the text below. ssid and wpa\_passphrase are changed into WiFi ID and password.

```
interface=wlan0
driver=nl80211
ssid=ARTIK_AP
auth_algs=1
hw_mode=g
channel=6
wpa=2
wpa_passphrase=artik@iot
wpa_pairwise=TKIP CCMP
rsn_pairwise=CCMP
```

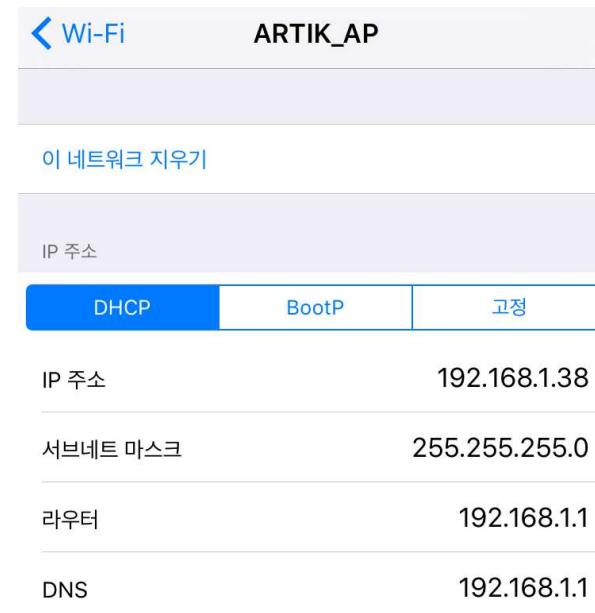
- `# hostapd /etc/hostapd/hostapd.conf -B` Activate hostapd.

# Methodology

## ■ Wi-Fi Access Point setting

### Step 5. WiFi connection

You can connect to ARTIK\_AP using mobile phone as the picture on the right.



### Step 6. Exit

- Enter the following command to exit AP mode, .  
`killall hostapd  
modprobe -r dhd  
modprobe dhd op_mode=0  
ifconfig wlan0 up  
if [ -f "/usr/lib/systemd/system/wpa_supplicant.service" ]; then  
systemctl restart wpa_supplicant  
fi`

# Methodology

## ■ AP mode scripting

- We can confirm that AP mode works well in the part of **Wi-Fi Access Point setting**. However, there are so many commands to input, so let us write a script to turn on and off the AP mode for convenience.

Most of the code consists of commands from the part of **Wi-Fi Access Point setting**.

However, if you have previously execute AP mode and exit without entering a normal exit code, when you attempt to reconnect again, you will see an error message (delete /var/run/hostapd/wlan0 file) and fail to reconnect.

So we added a command (**rm -f var/run/hostapd/wlan0**) to delete the file before the script.

# Methodology

## ■ AP mode scripting

### ▪ Source Code

#### APmode.sh

```
#!/bin/bash
rm -f /var/run/hostapd/wlan0
echo "[Configuring dnsmasq...]"
sed -i 's/#bind-interfaces/bind-interfaces/g' /etc/dnsmasq.conf
echo dhcp-
range=192.168.1.2,192.168.1.100>>/etc/dnsmasq.conf
systemctl stop connman
ifconfig eth0 up
dhclient eth0
modprobe -r dhd
modprobe dhd op_mode=2
ifconfig wlan0 192.168.1.1 up
dnsmasq -C /etc/dnsmasq.conf
echo "[Configuring iptables...]"
sysctl net.ipv4.ip_forward=1
iptables --flush
iptables -t nat --flush
iptables --delete-chain
iptables -t nat --delete-chain
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
iptables -A FORWARD -i wlan0 -j ACCEPT
echo "[Configuring hostapd...]"
sed -
i 's/#interface=/interface=wlan0/g' /etc/hostapd/hostapd.conf
sed -
```

```
i 's/#driver=/driver=nl80211/g' /etc/hostapd/hostapd.conf
sed -
i 's/#ssid=/ssid=ARTIK_AP/g' /etc/hostapd/hostapd.conf
sed -
i 's/#auth_algs/auth_algs=1/g' /etc/hostapd/hostapd.conf
sed -i 's/#hw_mode=/hw_mode=g/g' /etc/hostapd/hostapd.conf
sed -i 's/#channel=/channel=6/g' /etc/hostapd/hostapd.conf
sed -
i 's/#wpa_passphrase=/wpa_passphrase=artik@iot/g' /etc/hostapd/hostapd.conf
sed -i 's/#wpa_pairwise=TKIP/wpa_pairwise=TKIP
CCMP/g' /etc/hostapd/hostapd.conf
sed -
i 's/#rsn_pairwise=CCMP/rsn_pairwise=CCMP/g' /etc/hostapd/hostapd.conf
sed -i 's/#wpa=3/wpa=2/g' /etc/hostapd/hostapd.conf
hostapd /etc/hostapd/hostapd.conf -B
```

#### APmode\_end.sh

```
#!/bin/bash
killall hostapd
modprobe -r dhd
modprobe dhd op_mode=0
ifconfig wlan0 up
if [
f "/usr/lib/systemd/system/wpa_supplicant.service"]; then
systemctl restart wpa_supplicant
fi
```

# Result

## ■ Execution result

Wi-Fi Access Point connects ARTIK to the Internet via Ethernet LAN (eth0), and then supplies wireless network to other devices. Therefore Network Address Translation (NAT) to set up dnsmasq, iptables, and hostapd is necessary. Though it is much easier to input the command for the step and check the normal operation, it is more convenient for us to write the script

- When the script is executed, it is output as below and it is confirmed that a WiFi named ARTIK\_AP is created.

```
[root@localhost ~]# ./APmode.sh
[Configuring dnsmasq...]
[506.383260] dhd_wlan_set_carddetect: notify_func=c04bd124, mmc_host_dev=d8ec1410, val=0
[506.713325] dhd_wlan_set_carddetect: notify_func=c04bd124, mmc_host_dev=d8ec1410, val=1
[Configuring iptables...]
net.ipv4.ip_forward = 1
[Configuring hostapd...]
Configuration file: /etc/hostapd/hostapd.conf
nl80211: Could not re-add multicast membership for vendor events: -2 (No such file or directory)
Using interface wlan0 with hwaddr ec:1f:72:d5:19:95 and ssid "ARTIK_AP"
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
```

# Result

## ■ Execution result

- Ifconfig shows that eth0 is automatically assigned an IP address when wired LAN is connected and wlan0 is set to 192.168.1.1 shown as below.

```
[root@localhost ~]# ifconfig
eth0: flags=-28605<UP,BROADCAST,RUNNING,MULTICAST,DYNAMIC> mtu 1500
 inet 192.168.0.201 netmask 255.255.255.0 broadcast 192.168.0.255
 inet6 fe80::3440:93ff:fee6:355f prefixlen 64 scopeid 0x20<link>
 ether 36:40:93:e6:35:5f txqueuelen 1000 (Ethernet)
 RX packets 1259 bytes 118406 (115.6 KiB)
 RX errors 0 dropped 0 overruns 0 frame 0
 TX packets 151 bytes 15425 (15.0 KiB)
 TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
 device interrupt 32

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
 inet 127.0.0.1 netmask 255.0.0.0
 inet6 ::1 prefixlen 128 scopeid 0x10<host>
 loop txqueuelen 0 (Local Loopback)
 RX packets 196 bytes 16343 (15.9 KiB)
 RX errors 0 dropped 0 overruns 0 frame 0
 TX packets 196 bytes 16343 (15.9 KiB)
 TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
 inet 192.168.1.1 netmask 255.255.255.0 broadcast 192.168.1.255
 inet6 fe80::ee1f:72ff:fed5:1995 prefixlen 64 scopeid 0x20<link>
 ether ec:1f:72:d5:19:95 txqueuelen 1000 (Ethernet)
 RX packets 8 bytes 1090 (1.0 KiB)
 RX errors 0 dropped 6 overruns 0 frame 0
 TX packets 34 bytes 3624 (3.5 KiB)
 TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

- Output is shown as below when exiting.

```
[root@localhost ~]# ./APmode_end
[538.357931] dhd_wlan_set_carddetect: notify_func=c04bd124, mmc_host_dev=d8ec1410, val=0
[538.702991] dhd_wlan_set_carddetect: notify_func=c04bd124, mmc_host_dev=d8ec1410, val=1
```

# Conclusions

- ARTIK5 has used Access Point Mode, which can supply wireless network to other devices using Wi-Fi.
- Since ARTIK5 has two Wi-Fi antennas at 2.4GHz and 5GHz, the initial goal was to have two AP modes run at the same time.
- In other words, wlan0 and wlan1 were controlled at the same time, but it was not able to proceed because of limitation of one built-in wireless LAN card in hardware.

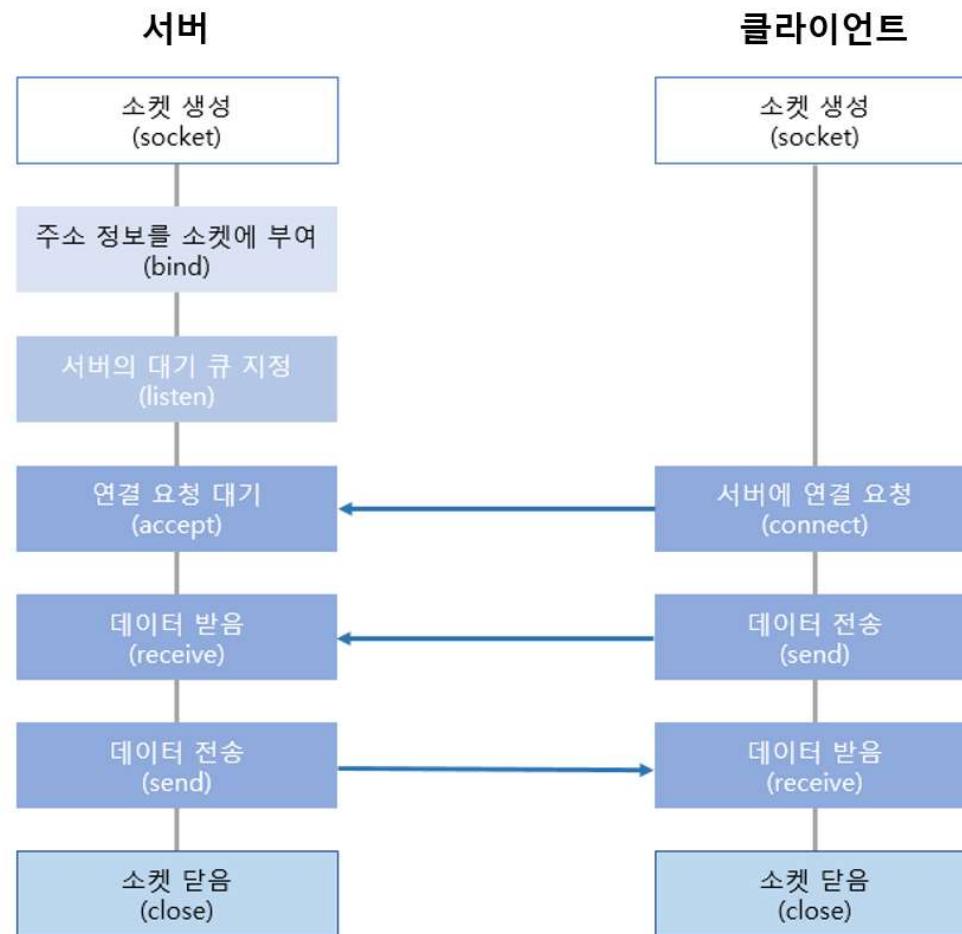
## II. ARTIK 기술 교육

### 7. Socket Communication



# Socket Communication using Wi-Fi

## ■ TCP Socket Communication Process



# Socket Communication using Wi-Fi

## ■ Functions

- #include <sys/types.h>, #include <sys/socket.h>

|                                                                                 |                                                                 |
|---------------------------------------------------------------------------------|-----------------------------------------------------------------|
| <b>int socket</b><br>(int domain, int type, int protocol);                      | 소켓 생성, return 소켓 디스크립터<br>(프로토콜 도메인, 서비스 유형, protocol(보통 0))    |
| <b>int bind</b><br>(int s, const struct sockaddr* name, socklen_t *namelen);    | 소켓 디스크립터를 이용하여 소켓에 주소 부여<br>(소켓 디스크립터, 바인드할 소켓 주소, 주소의 크기)      |
| <b>int listen</b><br>(int s, int backlog);                                      | 소켓 활성화 및 연결 요청 개수 지정<br>(소켓 디스크립터, 최대 대기 클라이언트 수)               |
| <b>int accept</b><br>(int s, struct sockaddr* addr, socklen_t *addrlen);        | 연결 설정, return 클라이언트 소켓 디스크립터<br>(소켓 디스크립터, 클라이언트 소켓 주소, 주소의 크기) |
| <b>int connect</b><br>(int s, const struct sockaddr* name, socklen_t *namelen); | 클라이언트가 서버에 연결요청 시 사용<br>(소켓 디스크립터, 서버 소켓 주소)                    |
| <b>ssize_t send</b><br>(int s, const void *msg, size_t len, int flags);         | 데이터 전송<br>(소켓 디스크립터, 전송하려는 데이터, flags)                          |
| <b>ssize_t recv</b><br>(int s, const void *buf, size_t len, int flags);         | 데이터 수신<br>(소켓 디스크립터, 수신 버퍼, flags)                              |

# Source code of Wi-Fi Communication

## ■ Source Code(Server-1)

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <signal.h>

#include <sys/types.h>
#include <sys/wait.h>

#define PORT 20162
#define BUFFER_SIZE 3490
#define BUFF_SIZE 100

#define BACKLOG 5
```

```
int main() {

 int sockfd, new_fd; // 소켓(파일) 디스크립터
 struct sockaddr_in my_addr; // 나의 ip 주소
 memset(&my_addr, 0, sizeof(my_addr)); // 초기화

 ssize_t recv_num; // recv 함수 리턴값 체크를 위한 변수
 char recv_data[BUFFER_SIZE]; // 소켓 통신에서 받는 데이터
 char send_data[BUFFER_SIZE]; // 소켓 통신에서 보내는 데이터
 pid_t pid; // 프로세스 id

 // TCP 소켓 생성
 sockfd = socket(AF_INET, SOCK_STREAM, 0);
 my_addr.sin_family = AF_INET; // IPv4
 my_addr.sin_port = htons(PORT); // Source Port(현재 3490)
 my_addr.sin_addr.s_addr = htonl(INADDR_ANY); // 자신의 ip 주소 대입

 // 소켓 바인딩
 if (bind(sockfd, (struct sockaddr *) &my_addr, sizeof(my_addr)) == -1) {
 printf("bind error\n");
 return -1;
 }

 // 클라이언트로부터의 연결대기
 if (listen(sockfd, BACKLOG) == -1) {
 printf("listen error\n");
 return -1;
 }
}
```

# Source code of Wi-Fi Communication

## ■ Source Code(Server-2)

```

printf("Waiting clients\n");

while (1) {
 struct sockaddr_in their_addr, peer_addr; // 상대방의 ip 주소

 socklen_t their_addrLength = sizeof(their_addr);

 // 클라이언트로부터의 연결요청이 들어온 경우 accept
 while((new_fd = accept(sockfd, (struct sockaddr*)&their_addr,
 (socklen_t *)&their_addrLength)) >= 0) {
 getpeername(new_fd, (struct sockaddr*)&peer_addr,
 &their_addrLength);

 char peer_ip[sizeof(peer_addr.sin_addr) + 1] = { 0 };
 sprintf(peer_ip, "%s", inet_ntoa(peer_addr.sin_addr));

 // 클라이언트 ip 출력
 if(strcmp(peer_ip,"0.0.0.0") != 0) {
 printf("Client IP : %s\n", peer_ip);
 }

 if (new_fd < 0) {
 printf("accept error\n");
 exit(0);
 }

 pid = fork();
 }
}

```

```

if(pid == 0) {
 close(sockfd);
 ssize_t recv_num;

 // 데이터를 클라이언트로부터 receive
 while((recv_num = recv(new_fd, recv_data, BUFF_SIZE, 0)) > 0) {
 recv_data[recv_num] = '\0';
 fputs(recv_data, stdout);
 fflush(stdout);

 // 받은 데이터를 클라이언트에 전송
 sprintf(send_data,"%s",recv_data);
 send(new_fd, send_data, strlen(send_data), 0);
 }

 close(new_fd);
 return 0;
}

else {
 close(new_fd);
}
}

close(sockfd);

return 0;
}

```

# Source code of Wi-Fi Communication

## ■ Source Code(Client)

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <signal.h>

#include <sys/types.h>
#include <sys/wait.h>

#define PORT 3490
#define MAXDATASIZE 100

int main(int argc, char** argv)
{
 if (argc != 2) {
 printf("실행 커맨드 뒤에 서버 IP 주소를 입력하세요", argv[0]);
 return -1;
 }

 int sockfd = socket(AF_INET, SOCK_STREAM, 0);
 struct sockaddr_in their_addr;
 memset(&their_addr, 0, sizeof(their_addr));
 their_addr.sin_family = AF_INET;
 their_addr.sin_port = htons(PORT);
 inet_aton(argv[1], (struct in_addr*)&their_addr.sin_addr);
}
```

```
if (connect(sockfd, (struct sockaddr*) &their_addr, sizeof(their_addr)) == -1)
{
 printf("connect error\n");
 return -1;
}

else
{
 int recv_datasize, send_datasize;
 char send_data[MAXDATASIZE];
 char recv_data[MAXDATASIZE];

 while (1)
 {
 //서버에 문자열을 보낸 뒤 서버가 보낸 echo를 받아 출력.
 printf("전송할 데이터 입력: ");

 fgets(send_data,MAXDATASIZE,stdin);

 write(sockfd, send_data, strlen(send_data));

 recv_datasize = read(sockfd, recv_data, MAXDATASIZE);
 printf("%d bytes read\n", recv_datasize);
 recv_data[recv_datasize] = '\0';
 fputs(recv_data, stdout);
 fflush(stdout);
 }
}
close(sockfd);

return 0;
}
```

# Wi-Fi ex. result

## ■ Execution result

- Server ARTIK

```
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# ./server
recv = 1
recv = 12
recv = 123
recv = 1234
recv = 12345
exit
[
```

- Client ARTIK

```
[root@localhost ~]#
[root@localhost ~]# ./client 192.168.0.104
send : 1
send : 12
send : 123
send : 1234
send : 12345
send : 999
[
```

# Source code of LED Control using Wi-Fi

## ■ Source Code(Server-1)

```
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <sys/wait.h>
#include "wiringARTIK.h"

#define MYPORT 3490
#define BACKLOG 10
#define MAXDATASIZE 100

//=====GPIO=====
#define PINNUMBER 135
#define HIGH 1
#define LOW 0
#define INPUT 1
#define OUTPUT 0
//=====GPIO=====

int main() {
 int sockfd, sockfd2; // 소켓의 파일 디스크립터
 struct sockaddr_in server_addr, client_addr; // 소켓 주소
 int numbytes; // receive한 데이터의 크기
 int on = 1; // 소켓 옵션 정보를 저장
 int sin_size; // 클라이언트 소켓 주소의 크기
 int recv_data; // receive 데이터를 실수로 저장할 공간
 char buf[MAX_DATA_SIZE]; // receive 데이터를 저장할 버퍼

 // set GPIO pin mode to OUTPUT
 if (!digitalPinMode(PINNUMBER, OUTPUT)) {
 printf("Fail to set pin mode");
 return -1;
 }

 // 소켓 생성
 sockfd = socket(AF_INET, SOCK_STREAM, 0);
 if (sockfd == -1) {
 perror("socket");
 exit(1);
 }

 // 호스트 IP주소와 포트 번호 기록
 server_addr.sin_family = AF_INET;
 server_addr.sin_addr.s_addr = htonl(INADDR_ANY);
 server_addr.sin_port = htons(PORT);
 bzero(&(server_addr.sin_zero), 8);

 // 소켓의 세부사항을 설정
 if (setsockopt(sockfd, SOL_SOCKET, SO_REUSEADDR,
&on, sizeof(on)) == -1) {
 perror("setsockopt");
 exit(1);
 }

 // 소켓에 주소 부여
 if (bind(sockfd, (struct sockaddr *)&server_addr,
sizeof(server_addr)) == -1) {
 perror("bind");
 exit(1);
 }
}
```

# Source code of LED Control using Wi-Fi

## ■ Source Code(Server-2)

```
// 소켓 활성화 및 연결 설정 요구 최대 수 지정
if (listen(sockfd, BACKLOG) == -1) {
 perror("listen");
 exit(1);
}

while(1) {
 // 연결 요청 대기
 sockfd2 = accept(sockfd, (struct sockaddr *)&client_addr,
&sin_size);
 if (sockfd2 == -1) {
 perror("accept");
 continue;
 }

 // child 프로세스 생성
 if (fork() == 0) {
 while(1) {
 // 클라이언트로부터 전송 받은 데이터 읽기
 numbytes = recv(sockfd2, buf, MAX_DATA_SIZE, 0);
 if (numbytes == -1) {
 perror("recv");
 exit(1);
 }
 buf[numbytes] = '\0';

 // 데이터를 실수로 변환하고
 // 999를 전송하면 통신 종료
 recv_data = atof(buf);
 printf("recv_data : %d\n", recv_data);
 if (recv_data == 999.000000) {
 printf("exit\n");
 close(sockfd2);
 return -1;
 } else if (recv_data == 1) {
 printf("LED is ON\n");
 digitalWrite(PINNUMBER, HIGH);
 } else if (recv_data == 0) {
 printf("LED is OFF\n");
 digitalWrite(PINNUMBER, LOW);
 } else {
 printf("Enter 1,0, or 999\n");
 }
 while (waitpid(-1, NULL, WNOHANG) > 0);
 }
 }
}
```

# Source code of LED Control using Wi-Fi

## ■ Source Code(Client)

```
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>
#include <netdb.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include "wiringARTIK.h"

#define PORT 3490
#define MAXDATASIZE 100

int main(int argc, char *argv[])
{
 int send_data;
 int sockfd, numbytes;
 char buf[MAXDATASIZE];
 struct hostent *he;
 struct sockaddr_in their_addr;

 if (argc != 2) {
 fprintf(stderr, "usage: ./client <host-name>\n");
 exit(1);
 }

 if ((he= gethostbyname(argv[1])) == NULL) {
 perror("gethostbyname");
 exit(1);
 }

 if ((sockfd = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
 perror("socket");
 exit(1);
 }

 their_addr.sin_family = AF_INET;
 their_addr.sin_port = htons(PORT);
 their_addr.sin_addr = *((struct in_addr *)he->h_addr);
 bzero(&(their_addr.sin_zero), 8);
 if (connect(sockfd, (struct sockaddr *)&their_addr, sizeof(struct
sockaddr)) == -1) {
 perror("connect");
 exit(1);
 }

 while(strcmp(buf, "quit")){
 printf(" Send Data = ");
 scanf("%d",&send_data);
 sprintf(buf, "%d", send_data);

 if (send(sockfd, buf, strlen(buf), 0) == -1) {
 perror("send");
 exit(0);
 }
 }
 close(sockfd);

 return 0;
}
```

# Wi-Fi ex. result

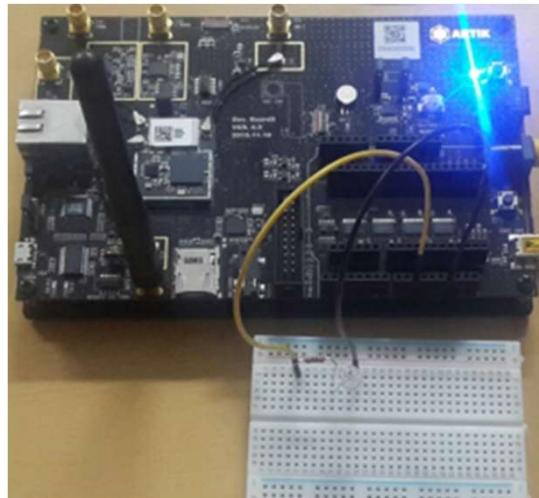
## ■ Execution result

- Server ARTIK

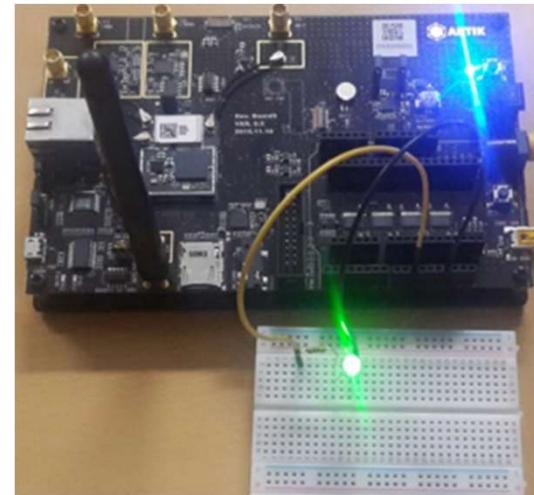
```
[root@localhost wifi]# ./server
Recieve Data = 1
LED is on
Recieve Data = 2
LED is off
```

- Client ARTIK

```
[root@localhost ~]# ./client 192.168.0.107
Send Data = 1
Send Data = 2
Send Data =
```



Before



After

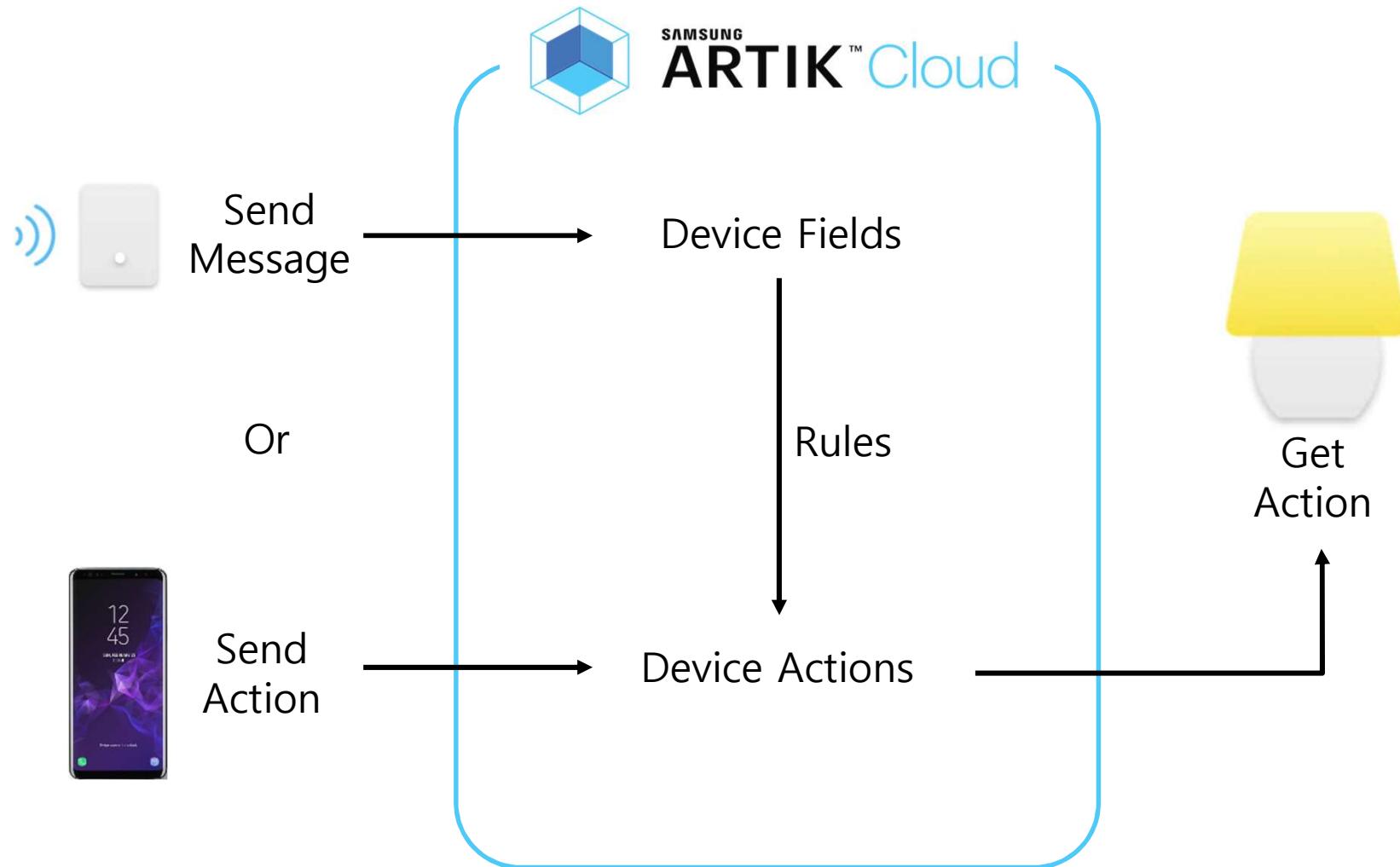
## II. ARTIK 기술 교육

### 8. ARTIK Cloud (1) - IoT System



# ARTIK Cloud

## ■ IoT System Using ARTIK Cloud



## ■ IoT System Using ARTIK Cloud

- Device Fields : Describe fields for each piece of data produced by the device
- Device Actions : Describe actions that the device is capable of receiving
- Rules : Send actions according to the conditions of the field
- Send Message, Send Action, Get Action
  - : By REST API, MQTT, Device Simulator, Node-RED...
- Example Process
  - Create Device Type – Create Device
  - Send Messages : Device Simulator, REST API, MQTT
  - Get Actions : REST API, MQTT
  - Scenes, Node-RED, App Inventor

## II. ARTIK 기술 교육

### 9. ARTIK Cloud (2) - Create Device



# ARTIK Cloud

## ■ ARTIK Cloud Sign up

SAMSUNG ARTIK™ | cloud services Works with... Pricing Blog Contact Us

Cloud Developer My Cloud

ARTIK

Make Connections, Not Silos

The ARTIK IoT platform enables...

Select one of these three

SAMSUNG Sign in with Samsung

Sign in with Google

N Sign in with Naver

Email

Password

Forgot password?

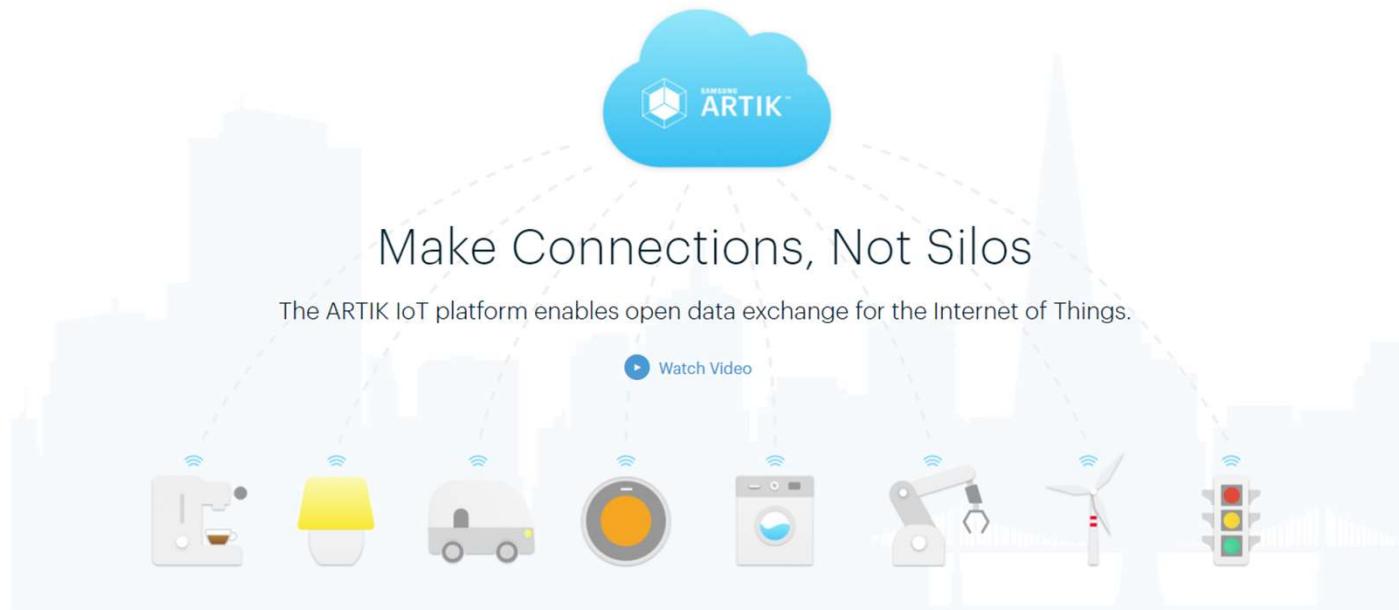
SIGN IN

Don't have an account yet? SIGN UP

# ARTIK Cloud

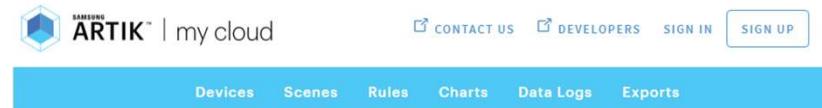
## ■ ARTIK Cloud

- My Cloud Menu
- Cloud Developers Menu

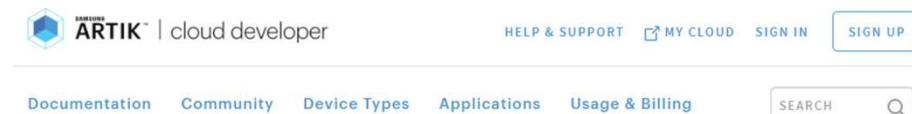


# ARTIK Cloud

## ■ ARTIK Cloud Menu



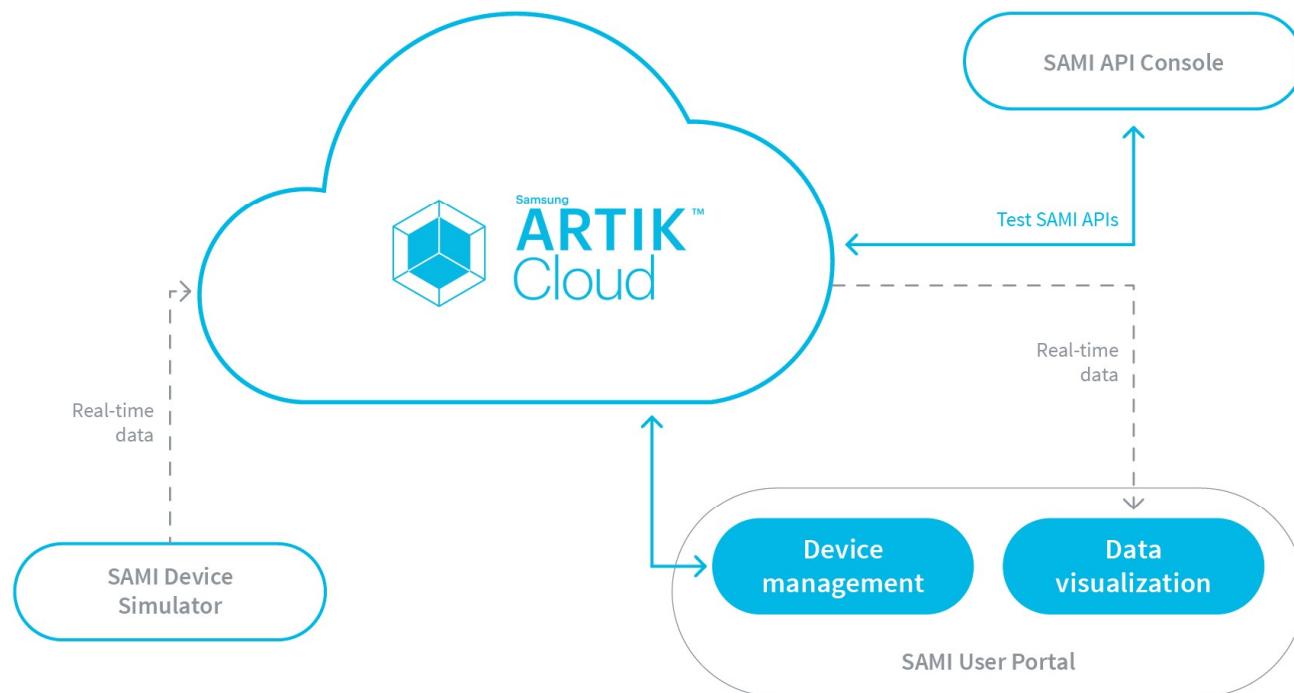
- My cloud : Interface for sending and receiving data to connected devices
  - Devices : Add or delete devices and check information about connected devices, Generate token for device
  - Scenes : Send a bunch of actions to one or more devices
  - Rules : Create rules for sending and receiving data
  - 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
  - Usage & Billing : View data usage and artik cloud services pricing

# 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]

The screenshot shows the ARTIK Cloud developer dashboard. At the top, there is a navigation bar with links for Documentation, Community, Device Types (which is highlighted with a red box), Applications, and Usage & Billing. To the right of the navigation bar are links for HELP & SUPPORT and MY CLOUD, along with a SEARCH bar. Below the navigation bar, there is a large blue banner with the text "One cloud. Many options. New SmartThings Cloud will incorporate Samsung Connect Cloud and ARTIK Cloud into a single platform. Learn more". Below the banner, there is a large blue area labeled "My Dashboard" with a white cloud icon.

The screenshot shows the ARTIK Cloud developer dashboard. At the top, there is a navigation bar with links for Documentation, Community, Device Types (which is highlighted with a red box), Applications, and Usage & Billing. To the right of the navigation bar are links for HELP & SUPPORT and MY CLOUD, along with a SEARCH bar. Below the navigation bar, there is a navigation menu with "DEVICE TYPES" and "Overview" items. In the main area, there is a section titled "Overview" with a dropdown menu set to "Last 7 days". To the right of the dropdown is a blue button with the text "+ NEW DEVICE TYPE" which is highlighted with a red box.

## ■ ARTIK Cloud tutorial

- Step 1 : Make a device type
  - Input Device Display Name (ex) Temp to LED
  - Input Unique Name (ex) com.artik.temptoled
  - Click [Create Device Type]

New Device  
Type



DEVICE DISPLAY NAME

 53

UNIQUE NAME

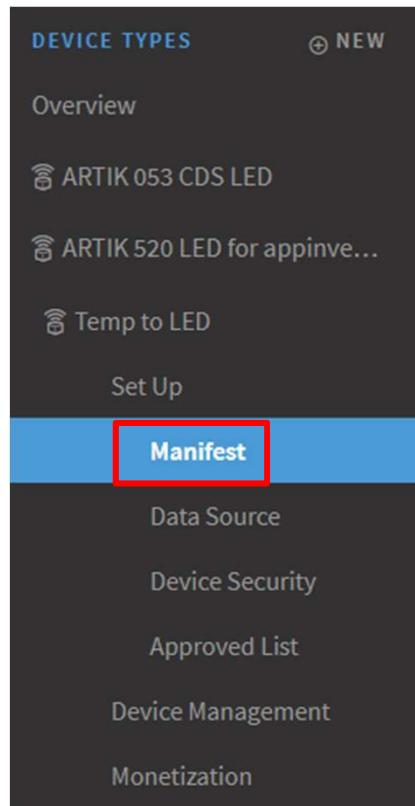
 236

**CREATE DEVICE TYPE**   **CANCEL**

## ■ ARTIK Cloud tutorial

### ▪ Step 2 : Make a manifest

- Check 'Temp to LED'
- Click [Setup] – [Manifest] – [+ New Manifest]



Create a manifest for ARTIK new type

ARTIK cloud services 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 services.



## ■ ARTIK Cloud tutorial

### ▪ Step 2 : Make a manifest

- Make messages field (from ARTIK to ARTIK cloud)
- Type 'temp' and 'hum' in 'Field Name' and select 'double' in 'Data Type'
- Description is option

The screenshot shows the 'Device Fields' section of the ARTIK Cloud interface. It displays two fields: 'temp' and 'hum'. Both fields have their 'FIELD NAME' set to 'temp' and 'hum' respectively. Under 'DATA TYPE', both are set to 'Double'. Under 'UNIT OF MEASUREMENT', 'temp' is set to 'SI.CELSIUS' and 'hum' is set to '%'. Both fields also have their 'ACCEPTABLE VALUE' set to 'Any Value'. The 'BROWSE STANDARD FIELDS' button is visible next to each field's input field.

**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 services platform.

**temp**

Is Collection (if the field contains an array)

**DATA TYPE**  
Double

**UNIT OF MEASUREMENT** SI.CELSIUS

**hum**

Is Collection (if the field contains an array)

**DATA TYPE**  
Double

**UNIT OF MEASUREMENT** %

**ACCEPTABLE VALUE**

Any Value  Range of Values  Selected Values

**temp**

**hum**

**STANDARD TEMPERATURE**

**DOUBLE**

**NEW FIELD** **NEW FIELD GROUP**

## ■ ARTIK Cloud tutorial

### ▪ Step 2 : Make a manifest

- Make a action (from ARTIK cloud to ARTIK)
- Click [NEXT : DEVICE ACTIONS]
- Make 'setOn', 'setOff' action

The active manifest describes the capabilities of your device type to other users and devices on the ARTIK cloud services 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 services platform.

**ACTION** [BROWSE STANDARD ACTIONS >](#)

`setOn` [STANDARD](#) [SET\\_ON](#)

**DESCRIPTION**

Set state to On

**SAVE** **CANCEL**

**NEXT: ACTIVATE MANIFEST**

**setOn** [STANDARD](#) [EDIT](#)  
Set state to On  
+ NEW PARAMETER

**setOff** [STANDARD](#) [EDIT](#)  
Set state to Off  
+ NEW PARAMETER

+ NEW ACTION

## ■ ARTIK Cloud tutorial

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

The screenshot shows the 'Simple Manifest' creation page for a device type named 'Temp to LED'. The page includes sections for 'Device Fields' and 'Device Actions', and a prominent 'Activate Manifest' button.

**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 services platform.

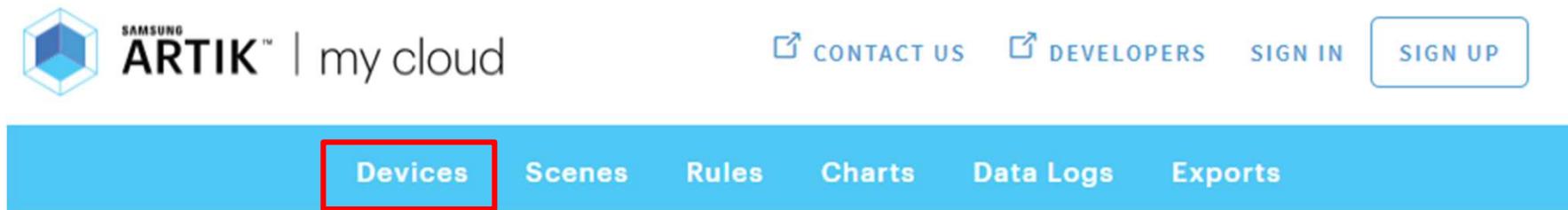
Your manifest is ready to be activated and does not require approval before going live. Activating this manifest will replace the current manifest. The device type will stay private.

| Fields                       | Actions          |
|------------------------------|------------------|
| temp<br>Double<br>SI.CELSIUS | setOn<br>Action  |
| hum<br>Double<br>%           | setOff<br>Action |

**ACTIVATE MANIFEST**   **CANCEL**

## ■ ARTIK Cloud tutorial

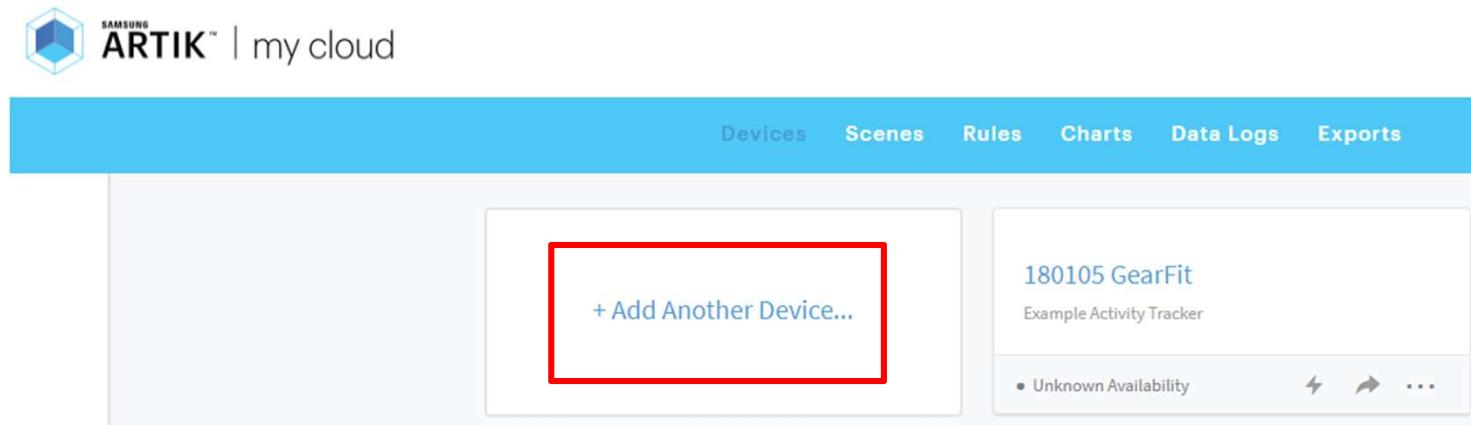
- Step 3 : Make a device
  - Open the ARTIK cloud menu
  - Click 'Devices'



## ■ ARTIK Cloud tutorial

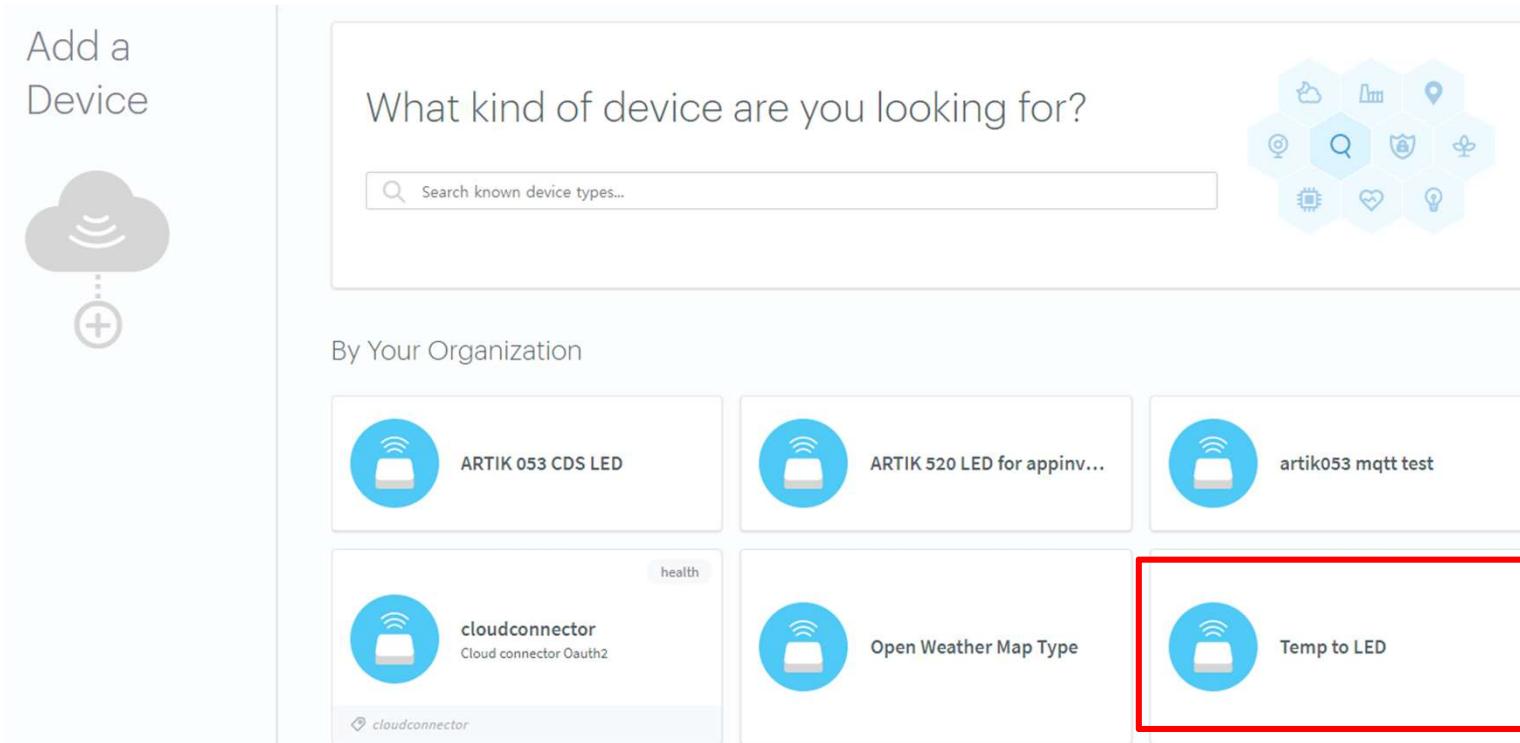
### ▪ Step 3 : Make a device

- You can see all your devices in this page
- Click [+Add another device...]



## ■ ARTIK Cloud tutorial

- Step 3 : Make a device
  - You can find 'Temp to LED', which is created at step 2.



# ARTIK Cloud

## ■ ARTIK Cloud tutorial

### ▪ Step 3 : Make a device

- Type 'Give Your Device a Name' something. (ex)ARTIK520 Temp to LED
- And you will see this device on 'My ARTIK Cloud' – 'Devices' page.

Give Your Device a Name

ARTIK520 Temp to LED

ADD DEVICE

< CHOOSE A DIFFERENT TYPE

Temp to LED

UNIQUE NAME

com.artik.temptoled

## ■ ARTIK Cloud tutorial

- Step 4 : Get Device ID / Token

ARTIK520 Temp to LED  

DEVICE INFO DATA TRANSFER

|                                 |                                    |
|---------------------------------|------------------------------------|
| DEVICE ADDED ON                 | DEVICE TYPE                        |
| 27/Mar/2018                     | Temp to LED                        |
| DEVICE ID                       | DEVICE TYPE ID                     |
| 010a/debb3a31500011b0222705e51b | dt1404246Rc5194R67942R0957#5tGR16J |
| DEVICE TOKEN                    | REVOKE TOKEN                       |
| XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  |                                    |

## II. ARTIK 기술 교육

### 10. ARTIK Cloud (3) - Send Message



# Device Simulator

## ■ Device Simulator on ARTIK 5

- Step 1 : 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
-rwxr--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
-rwxr--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=[user token]
```

# Device Simulator

## ■ Device Simulator on ARTIK 5

- Step 1 : Run Device simulator in ARTIK 5 dev. kit
  - How to get user token ( $\neq$  Device token)
  - <https://developer.artik.cloud/api-console/> -> Get Current User Profile API

Users

**①** **GET** Get Current User Profile </users/self>

Get the current user's profile

| Parameter | Value | Type | Location | Description |
|-----------|-------|------|----------|-------------|
|-----------|-------|------|----------|-------------|

**②** **TRY IT!** **CLEAR RESULTS**

Call  
<https://api.artik.cloud/v1.1/users/self>

Request Headers

```
{
 "Content-Type": "application/json",
 "Authorization": "Bearer USER TOKEN"
}
```

**③** → Copy&Paste

**LIST METHODS** **EXPAND METHOD**

# Device Simulator

## ■ Device Simulator on ARTIK 5

- Step 2 : 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   |
|----------------------------------|-------------------------------------|------------------------------|-----------------------|-----------------|----------------|---------|
| 48e3d2a08ae143edb03e06dcec1d5ade | dt013005c9302a428990073c9d6359b354  | 180105                       | GearFit               | 1               | LATEST         | a727495 |
| b31d17b8e92a4bf0ab18695149af070a | dtcfffc4b739fe447d08492687849b90264 | 180116                       | LED                   | 1               | LATEST         | dc33679 |
| c687886c8a35483faac0dd079f181c91 | dt8888ced85b524b81a991d712b77433f6  | 180116                       | TempSensor            | 3               | LATEST         | a3b79a4 |
| f1a15b1483294d1cb15a72653fc96a9  | dtf72ab948ed14460aad35aadd37a99222c | 180307                       | ARTIK Cloud Connector | 1               | LATEST         |         |
| 52b6b7b503b24c1a81338c42e99912b9 | dt71896a9e27e042c6b331a6594579f1a7  | appinventor float            | 6                     | LATEST          | a002b21        |         |
| 644fb0ffe8a3464ba22064b4a7f52268 | dta5bf8b79fd6f449285899938d485a4d6  | ARTIK 053 CDS LED device     | 1                     | LATEST          | b89b750        |         |
| 016a7dcbb3a345698f1fb0222795e51b | dt14042468c519486794283953e5f681b3  | ARTIK520 Temp to LED         | 1                     | LATEST          | 9177acd        |         |
| db9c722d25a64ef5b28101762fdd090a | dte299d6470a1b4462aae9a408c19178d7  | mqtt action from cloud - 053 | 1                     | LATEST          | de4bb28        |         |
| 8e516c5fc32549329cfdf7d53d758a49 | dt896778f5703c427cb4b0f028fd8a2045  | OpenWeatherMapConnector      | 1                     | LATEST          | 313c930        |         |

- Using command 'gs', you can get scenario sample. (gs : guess scenario)  
\$ gs [device ID, did] [Scenario file name]

```
$ gs 016a7dcbb3a345698f1fb0222795e51b scenario0327
Scenario saved to /root/016a7dcbb3a345698f1fb0222795e51b/scenario0327.json
```

- You can find scenario file in /[did]/scenario0327.json

```
$ [root@localhost ~]# cd /root/016a7dcbb3a345698f1fb0222795e51b
[root@localhost 016a7dcbb3a345698f1fb0222795e51b]# ls -l
total 4
-rw-r--r-- 1 root root 296 Mar 27 15:31 scenario0327.json
```

# Device Simulator

## ■ Device Simulator on ARTIK 5

### ▪ Step 2 : Guess Scenario and Run scenario

- Using vi editor, we will modify ID and NAME in scenario0327.json
- Modify field value, function, period, etc.
  - In the picture below, 'hum' field is deleted and the range of random function is modified



```

{
 "period": 1000,
 "data": {
 "hum": 0,
 "temp": 0
 },
 "sdid": "7591611d20ae48af8c274e17cbcd9b4b",
 "api": "POST",
 "config": {
 "hum": {
 "min": 0,
 "max": 10000,
 "function": "random",
 "type": "Double"
 },
 "temp": {
 "min": 0,
 "max": 10000,
 "function": "random",
 "type": "Double"
 }
 },
 "deviceToken": "beb80f93989548d4b3a2151bb31c83a5"
}

{
 "period": 1000,
 "data": {
 "hum": 0,
 "temp": 0
 },
 "sdid": "7591611d20ae48af8c274e17cbcd9b4b",
 "api": "POST",
 "config": {
 "temp": {
 "min": 10,
 "max": 100,
 "function": "random",
 "type": "Double"
 }
 },
 "deviceToken": "beb80f93989548d4b3a2151bb31c83a5"
}

```

| 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.                                     |

# Device Simulator

## ■ Device Simulator on ARTIK 5

- Step 2 : Guess Scenario and Run scenario
  - Restart Device Simulator and run scenario.

\$ rs [did] [scenario file name]

```
$ rs 7591611d20ae48af8c274e17cbcd9b4b s0328
Loading scenario from /root/7591611d20ae48af8c274e17cbcd9b4b/s0328.json
Reading file: /root/7591611d20ae48af8c274e17cbcd9b4b/s0328.json
$ Using this token to send the messages: beb80f93989548d4b3a2151bb31c83a5
Send #0 {"hum":0,"temp":15.590513229370117}
Got MID: 43ca5eaf4e048a486f828951b6e17eb
Send #1 {"hum":0,"temp":64.31137084960938}
Got MID: abaf7a4af53f41b59d097aa8c5a06d9c
Send #2 {"hum":0,"temp":67.51248168945312}
Got MID: 32a5f545b86849438f10ec800cbaf58c
Send #3 {"hum":0,"temp":88.52179718017578}
Got MID: 4a22c8a3418b4656aa2f3f60bc7ac38b
Send #4 {"hum":0,"temp":78.38431549072266}
Got MID: 053febeeee9d240998a9bd96cecc54c007
Send #5 {"hum":0,"temp":48.79450988769531}
Got MID: f401c067509543089dc4f13d549973cb
Send #6 {"hum":0,"temp":48.69914627075195}
Got MID: 07db09afbbcd4ce68392237b3ae395ee
Send #7 {"hum":0,"temp":22.736963272094727}
Got MID: cc7b91e2316c4d42b82a394e4ef1c064
Send #8 {"hum":0,"temp":42.271446228027344}
Got MID: 582992ddc33d455e95b1021eee1969b6
Send #9 {"hum":0,"temp":13.832640647888184}
Got MID: 5f13a0e7340f403c9d834ec8e664b13b
Send #10 {"hum":0,"temp":40.074405670166016}
Got MID: a2fb8d9fd2ca46e4ab8137c33156f442
```

- You can see the sending message

# Device Simulator

## ■ Device Simulator on ARTIK 5

- Step 3 : Check data in ARTIK Cloud
  - Click 'Charts'
  - Check the data that you want

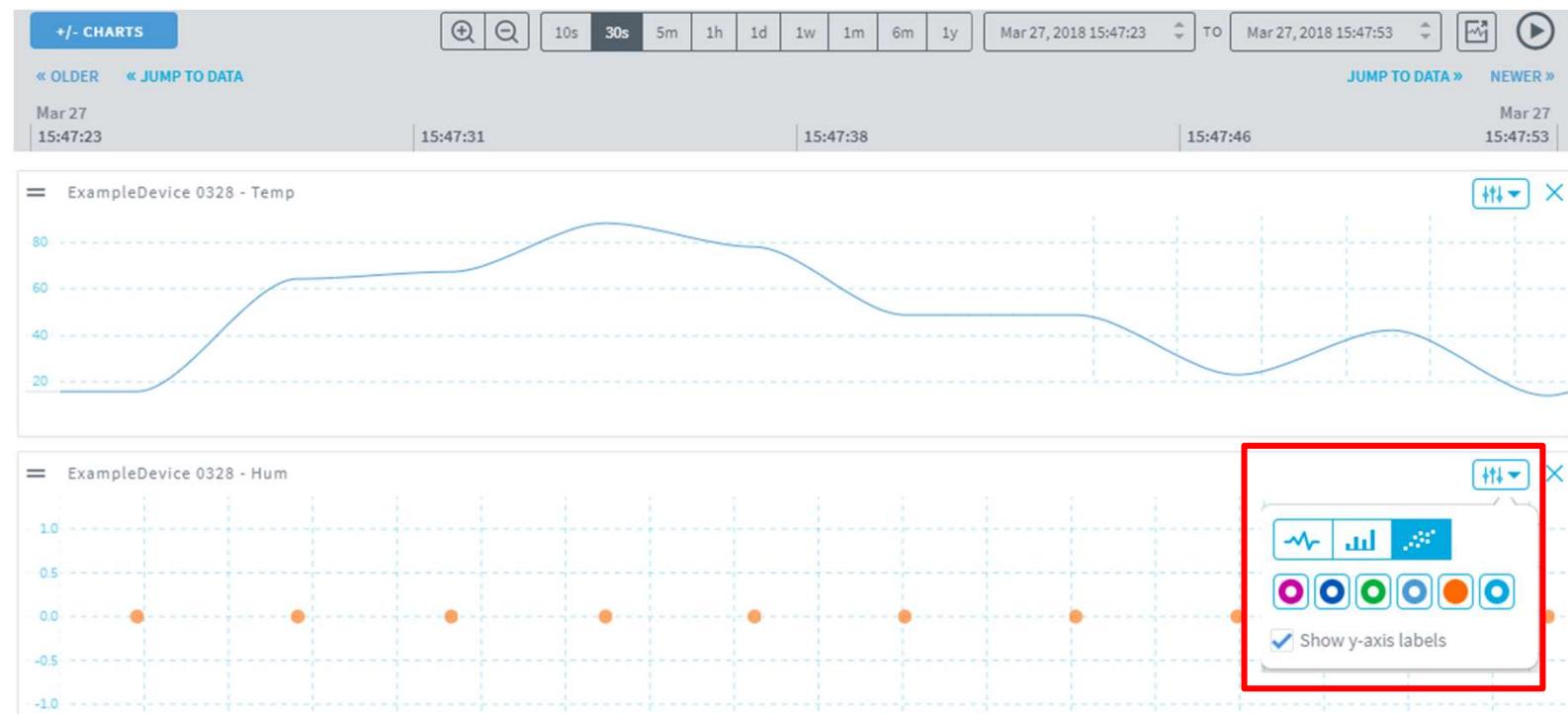
The screenshot shows the ARTIK Cloud interface with the 'Charts' tab highlighted by a red box. On the left, there is a button labeled '+/- CHARTS' also highlighted with a red box. Below the header, there are search and filter options, including time intervals (10s, 30s, 5m, 1h, 1d, 1w, 1m, 6m, 1y) and date range (Mar 27, 2018 15:47:23 TO Mar 27, 2018 15:47:53). The main area displays four data points with timestamps from Mar 27, 15:47:23 to 15:47:53.

This screenshot shows a detailed view of the selected chart. It displays the message '2 CHARTS SELECTED FOR DISPLAY (50 MAXIMUM)' and a single chart entry: 'ARTIK520 Temp To LED'. Within this entry, two data series are listed: 'Hum' and 'Temp', both marked with a checked checkbox icon.

# Device Simulator

## ■ Device Simulator on ARTIK 5

- Step 3 : Check data in ARTIK Cloud
  - You can change the graph type and color



# Device Simulator

## ■ Device Simulator on ARTIK Cloud

- Step 1 : Check data in ARTIK Cloud
  - Click 'Devices' – [Show Simulator]

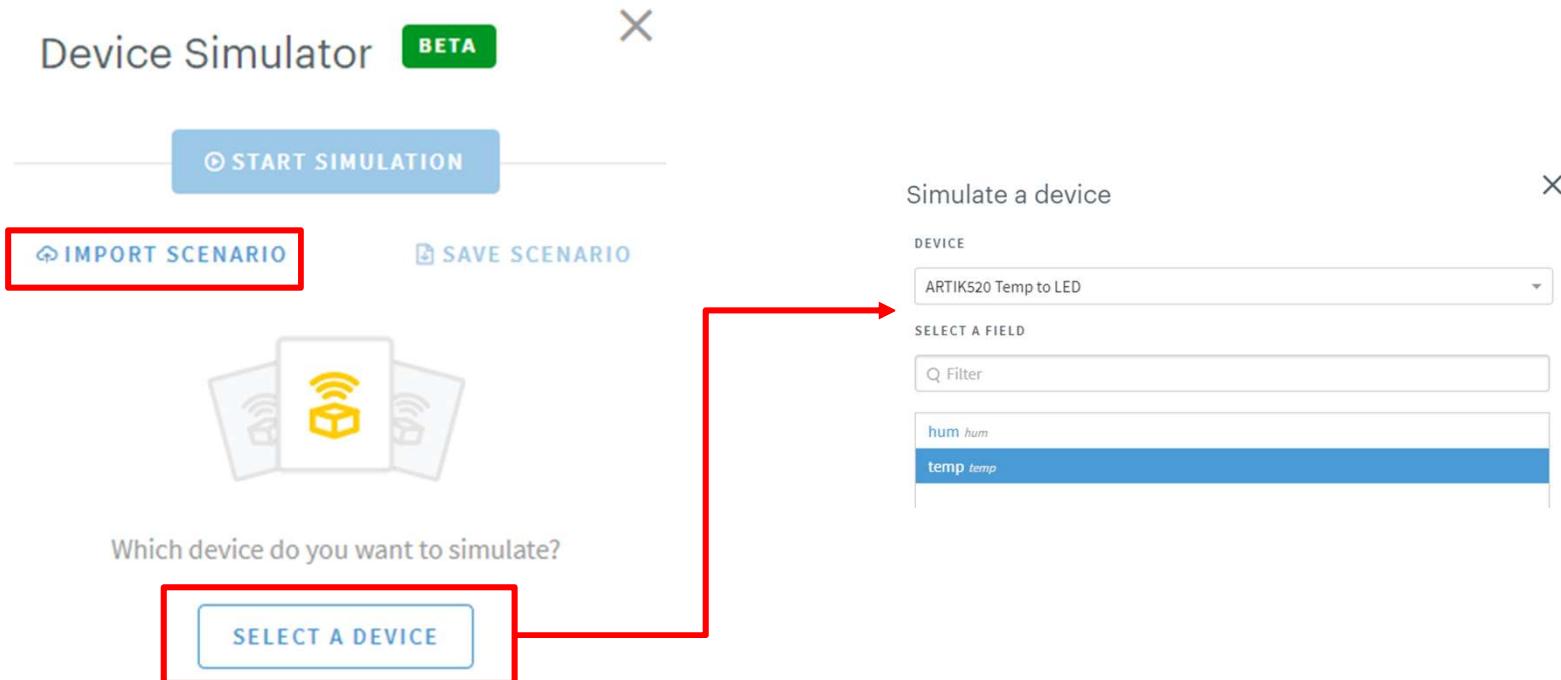
The screenshot shows the ARTIK Cloud interface. At the top, there's a navigation bar with the ARTIK logo and links for 'CONTACT US' and 'DEVELOPERS'. Below the navigation bar is a blue header with tabs: 'Devices' (which is highlighted with a red box), 'Scenes', 'Rules', 'Charts', 'Data Logs', and 'Exports'. The main content area has a sidebar on the left labeled 'Devices' with options: 'All', 'Shared with me', and 'My Devices'. A 'VIEW YOUR DATA' button is also in this sidebar. The main panel lists two devices: '180105 GearFit' (Example Activity Tracker) and '180116 LED' (ARTIK Smart Parking LED). Each device entry includes its name, a description, and a set of three small icons. At the bottom of the sidebar, there's a 'DEVELOPER TOOL' section with a 'SHOW SIMULATOR' button, which is also highlighted with a red box.

# Device Simulator

## ■ Device Simulator on ARTIK Cloud

### ▪ Step 2 : Import or Make Scenario

- Click 'Devices' – [Show Simulator]
- You can import json scenario file (ex- scenario0327.json)
- Or Click [Select a device] to make scenario



# Device Simulator

## ■ Device Simulator on ARTIK Cloud

- Step 2 : Import or Make Scenario
  - Edit Field Simulation interval, function(data pattern), range, step, value, etc.

Edit Field Simulation X

**FIELD** temp INTERVAL Every 5000 ms

**DATA PATTERN** Constant value DATA TYPE / UNIT Double / °C

**VALUE** 36.5 X

SAVE CHANGES

Edit Field Simulation X

**FIELD** hum INTERVAL Every 5000 ms

**DATA PATTERN** Increment DATA TYPE / UNIT Double / %

**START VALUE** 10 **MAXIMUM** 100

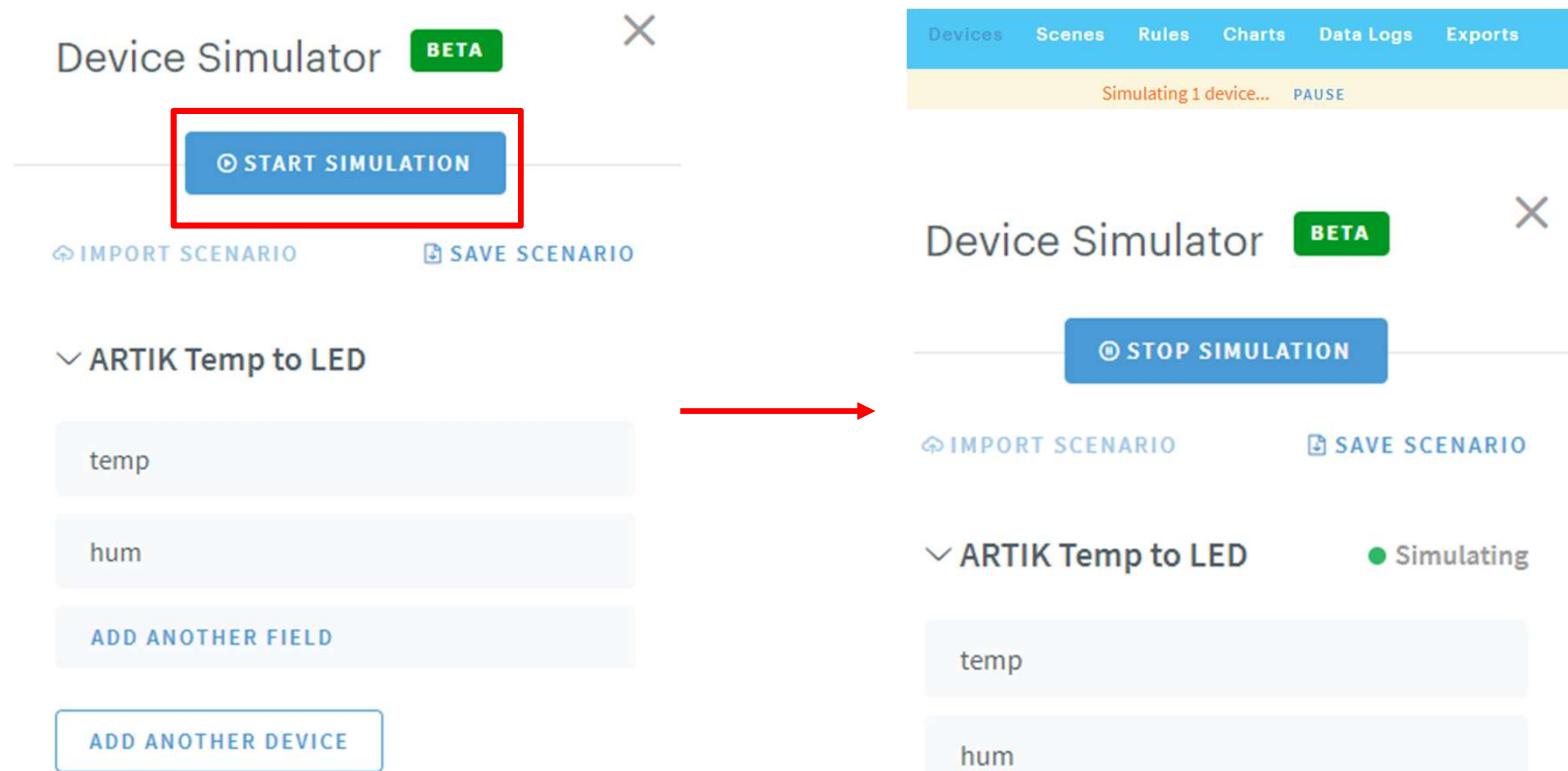
**INCREASE EACH STEP** 15  Start again when limit is reached

SAVE CHANGES

# Device Simulator

## ■ Device Simulator on ARTIK Cloud

- Step 3 : Start Simulation
  - Click [Start Simulation]
  - You can download json scenario file just created.



# Device Simulator

## ■ Device Simulator on ARTIK Cloud

- Step 3 : Check data in ARTIK Cloud



# 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.  |

# REST API Protocol

## ■ REST API Protocol

### Request Header Parameters

Content-Type "application/json"

Authorization "Bearer \_your\_device\_token\_"

### Response Body Parameters

mid Message ID.

# REST API Protocol

## ■ Send Message using Python SDK

- Step 1 : Check Python version ( 2.7 or 3.4+ )
  - PPT Source Code : 2.7 version
- Step 2 : Install ARTIK Cloud Python SDK
  - *# pip install artikcloud*
- Step 3 : Get Device ID, Token -> Save as config.json

```
{
 "device_id" : "_device_ID_",
 "device_token" : "_device_Token_"
}
```

## ■ Send Message using Python SDK

- Step 5 : ARTIK Cloud API Console : POST Send Message API
  - <https://developer.artik.cloud/api-console/>

POST Send Message /messages

Send a Message from a Source Device

| Parameter | Value                                 | Type   | Location | Description                                                                    |
|-----------|---------------------------------------|--------|----------|--------------------------------------------------------------------------------|
| data      | <input type="text" value="required"/> | string | body     | Data payload.                                                                  |
| sdid      | <input type="text"/>                  | string | body     | Source Device ID.                                                              |
| ts        | <input type="text"/>                  | number | body     | Timestamp (past, present or future). Defaults to current time if not provided. |
| type      | <input type="text" value="message"/>  | string | body     | Type.                                                                          |

[TRY IT!](#)

- data
  - Temperature Data
- sdid
  - Device ID
- type
  - POST Send Message API로 message

# REST API Protocol

## ■ Source Code (send-temp.py)

```
import artikcloud
from artikcloud.rest import ApiException
import sys, getopt
import time, json
from pprint import pprint

def main(argv):
 DEFAULT_CONFIG_PATH = 'config.json'

 with open(DEFAULT_CONFIG_PATH, 'r') as config_file:
 config = json.load(config_file)
 print(config)

 # Configure Oauth2 access token for the client application
 artikcloud.configuration = artikcloud.Configuration()
 artikcloud.configuration.access_token = config['device_token']

 # Read ADC Pin (Temperature sensor)
 adcPATH = "/sys/devices/126c0000.adc/iio:device0/in_voltage0_raw"
 target = open(adcPATH, 'r')
 voltage = float(target.read(8))
 temper = 217.75*voltage*0.439453125/2500 - 66.875
 target.close();
 print(' ')
 print 'Temper is', str(temper)
 print(' ')
```

```
Create an instance of the API Class
api_instance = artikcloud.MessagesApi()
device_message = {}
device_message['temp'] = temper # device의 temp 필드로 보낼 메시지
device_sdid = config['device_id'] # device ID
ts = None # timestamp

Construct a Message Object for request
data = artikcloud.Message(device_message, device_sdid, ts)

try:
 pprint(artikcloud.configuration.auth_settings()) # Debug Print
 api_response = api_instance.send_message(data) # Send Message
 pprint(api_response)
except ApiException as e:
 pprint("Exception when calling MessagesApi->send_message: %s\n" % e)

if __name__ == "__main__":
 while True:
 main(sys.argv[1:])
 time.sleep(5)
```

# REST API Protocol

## ■ Execution result of send-temp.py

```
[root@localhost pythonSample]# python send-temp.py
{u'device_token': u'██████████', u'device_id': u'████████'}
████████

Temper is 38.4998388672

{'artikcloud_oauth': {'in': 'header',
 'key': 'Authorization',
 'type': 'oauth2',
 'value': u'Bearer ██████████'},
 'data': {'mid': '5e8c5d599dcb4680b9d143df8ae61058'}}
{u'device_token': u'██████████', u'device_id': u'████████'}

Temper is 36.011875

{'artikcloud_oauth': {'in': 'header',
 'key': 'Authorization',
 'type': 'oauth2',
 'value': u'Bearer ██████████'},
 'data': {'mid': 'ba2af82a3e1647b49db4f5b9ecbb1b38'}}
```

|                      |                             |                             |                             |
|----------------------|-----------------------------|-----------------------------|-----------------------------|
| Temp to LED artik520 | Feb 20 2018<br>17:02:58.349 | Feb 20 2018<br>17:02:58.349 | {"temp":36.011875}          |
| Temp to LED artik520 | Feb 20 2018<br>17:02:52.979 | Feb 20 2018<br>17:02:52.979 | {"temp":38.499838867187506} |

## ■ MQTT Protocol

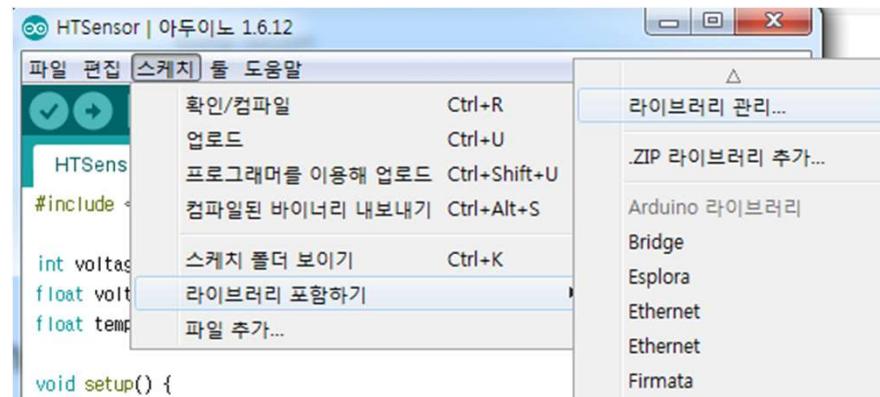
- MQTT is a lightweight messaging protocol.
- It is suitable for IoT, since it is bandwidth-efficient and uses little battery power.
- ARTIK Cloud devices can publish a data-only message to ARTIK Cloud or subscribe to receive Actions from ARTIK Cloud.

| 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>   |

# MQTT Protocol

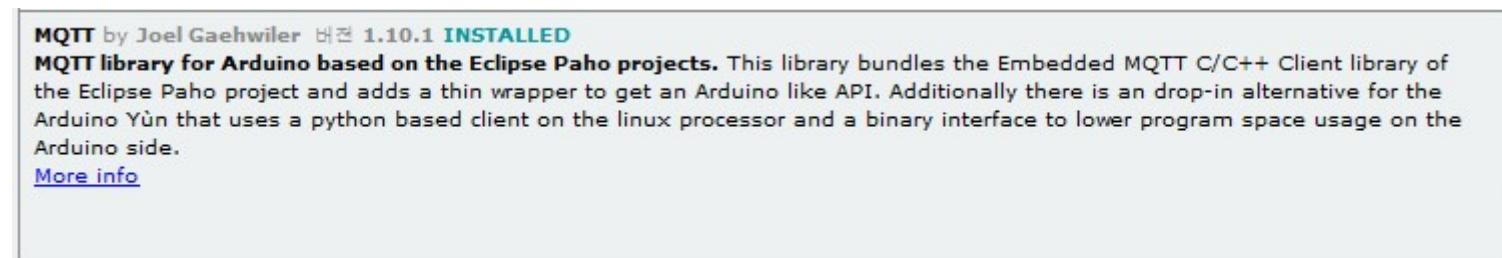
## ■ Install Arduino IDE libraries

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



- Step 2 : Install MQTT library

- Add #include <MQTTClient.h> to your sketch.



## ■ 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.

## ■ Connect a new device and get a device token

### Fields

|      |        |            |
|------|--------|------------|
| temp | Double | SI.CELSIUS |
| hum  | Double | %          |

## ■ Source Code – BasicCode

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

// MQTT Parameters
char mqttCloudServer[] = "api.artik.cloud";
int mqttCloudPort = 8883;
char mqttCloudClientName[] = "_Client_Name_"; // or
whatever you prefer
char mqttCloudUsername[] = "_device_ID_";
char mqttCloudPassword[] = "_device_token_";
char mqttCloudTopic[] = "/v1.1/messages/_device_ID_";
char buf[100] = "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("connected!");
}

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) {
}
```

## ■ 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          |  |
|--------------------|-----------------------------|-----------------------------|---------------|--|
| ExampleDevice 0328 | Mar 29 2018<br>14:52:47.084 | Mar 29 2018<br>14:52:47.084 | {"temp":19.3} |  |
| ExampleDevice 0328 | Mar 29 2018<br>14:52:44.152 | Mar 29 2018<br>14:52:44.152 | {"temp":19.3} |  |
| ExampleDevice 0328 | Mar 29 2018<br>14:52:41.001 | Mar 29 2018<br>14:52:41.001 | {"temp":19.3} |  |

## ■ Source Code – AutomationJson (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[] = "_Client_Name_"; // or whatever
you prefer
char mqttCloudUsername[] = "_device_ID_";
char mqttCloudPassword[] = "_device_Token_";
char mqttCloudTopic[] = "/v1.1/messages/_device_ID_";
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(".");
 }
}

void loop() {
 mqttCloudClient.loop();

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

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

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

}

int loadBuffer(double temp) {
 StaticJsonBuffer<200> jsonBuffer;
 JsonObject& dataPair = jsonBuffer.createObject();

 dataPair["temp"] = temp;
 dataPair.printTo(buf, sizeof(buf));
}
```

## ■ 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.

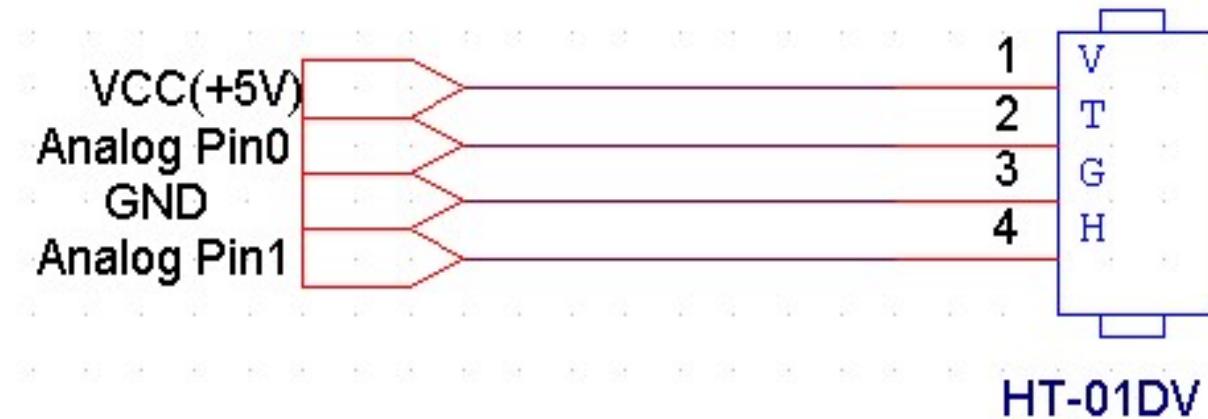
The screenshot shows a table with four columns: DEVICE, RECORDED AT, RECEIVED AT, and DATA. There is one row of data. The 'DEVICE' column contains 'ExampleDevice 0328'. The 'RECORDED AT' column contains 'Mar 29 2018 15:01:42.980'. The 'RECEIVED AT' column contains 'Mar 29 2018 15:01:42.980'. The 'DATA' column contains '[{"temp":24.5}]'. The table has a header row with 'MESSAGES' and 'ACTIONS' buttons. There are also 'FILTERS' and 'EXPORT DATA' buttons at the top.

| DEVICE             | RECORDED AT              | RECEIVED AT              | DATA            |
|--------------------|--------------------------|--------------------------|-----------------|
| ExampleDevice 0328 | Mar 29 2018 15:01:42.980 | Mar 29 2018 15:01:42.980 | [{"temp":24.5}] |

## ■ Required Hardware

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

## ■ Circuit Configuration



## ■ Source Code - RealSensor(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_ ";
char buf[100];

WiFiSSLClient ipCloudStack;
MQTTClient mqttCloudClient;

// Sensor Parameters
double temperature;
int voltage_raw0;
double voltage0;

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

 connect();
}

void loop() {
 mqttCloudClient.loop();

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

## ■ Source Code - RealSensor(2)

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

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

 DebugSerial.println("Connected!");
}

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

}

void loadBuffer(double temp.) {
 StaticJsonBuffer<200> jsonBuffer;

 JsonObject& dataPair = jsonBuffer.createObject();

 dataPair["temp"] = temp;

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

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

 voltage0 = voltage_raw0*0.439453125*2;

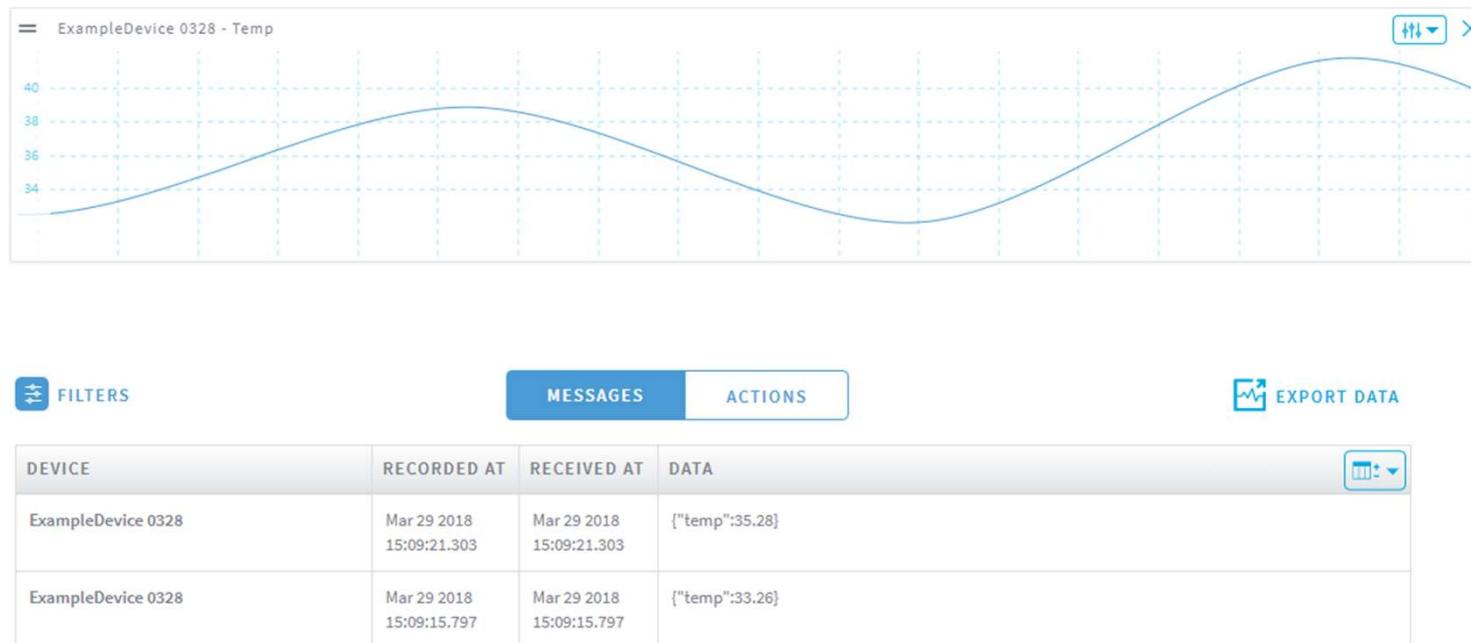
 temperature = -66.875 + 217.75*(voltage0/5000);
}
```

## ■ 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.



## **II. ARTIK 기술 교육**

### **11. ARTIK Cloud (4) - Get Action**



## ■ Set Rules

- Click 'Rules' – [+New Rule]



## ■ Set Rules

### ▪ Add Condition

- Send 'setOn' when 'temp' is more than 33
- Send 'setOff' when 'temp' less than or equal to 33

Choose device activity to monitor

IF

Actual Value ARTIK520 Temp to LED temp X

more than 33 °C

ADD DURATION

+ NEW CONDITION

This rule can run at any time on any day

Send actions to your devices

THEN

ARTIK520 Temp to LED setOn X

+ NEW ACTION

> SEND SETON TO ARTIK520 TEMP TO LED IF ARTIK520 Temp to LED temp is more than 33 THEN send to ARTIK520 Temp to LED the action setOn

> SEND SETOFF TO ARTIK520 TEMP TO LED IF ARTIK520 Temp to LED temp is less than or equal to 33 THEN send to ARTIK520 Temp to LED the action setOff

The screenshot shows the ARTIK Cloud Rule Editor interface. It consists of several sections: 'Choose device activity to monitor' (with a red box around the 'IF' condition), 'Send actions to your devices' (with a red box around the 'THEN' action), and two summary cards at the bottom. The 'IF' condition is set to 'Actual Value' for 'ARTIK520 Temp to LED temp', comparing it to '33' with the operator 'more than'. The 'THEN' action is 'ARTIK520 Temp to LED setOn'. Below these, two cards summarize the rules: one for setting on when temperature is above 33 and one for setting off when it is 33 or below.

## ■ Get Action using Python SDK

- Step 6 : ARTIK Cloud API Console : Get Normalized Actions API
  - <https://developer.artik.cloud/api-console/>

| Parameter | Value         | Type    | Location | Description                                                  |
|-----------|---------------|---------|----------|--------------------------------------------------------------|
| count     | 1             | integer | query    | Number of items to return per query.                         |
| endDate   | 1519111120875 | number  | query    | Timestamp of latest message (in milliseconds since epoch).   |
| mid       |               | string  | query    | Message ID of the Action being searched.                     |
| offset    |               | string  | query    | Offset for pagination.                                       |
| order     | desc *        | string  | query    | Ascending or descending sort order.                          |
| ddid      |               | string  | query    | Destination device ID being searched for actions.            |
| startDate | 1519107520875 | number  | query    | Timestamp of earliest actions (in milliseconds since epoch). |
| uid       |               | string  | query    | Owner's user ID being searched for actions.                  |

- count
  - 받아올 action의 수 = 1
- endDate
  - 현재 시간(timestamp, ms)
- startDate
  - 현재 시간에서 Send Temp 주기보다 긴 시간을 뺀 값 (timestamp, ms)
  - Send Temp 주기는 5초이므로 이것보다 길면 됨 (get-action.py Source Code에서는 1일)
- order
  - desc (가장 최근 Action을 받기 위한 내림차순 설정)

## ■ Source Code (get-action.py) -(1)

```
import artikcloud
from artikcloud.rest import ApiException
import sys, getopt
import time, json
from pprint import pprint

ledPin = 135
path_export = '/sys/class/gpio/export'
path_unexport = '/sys/class/gpio/unexport'
path_dir = '/sys/class/gpio/gpio%d/direction' % ledPin
path_val = '/sys/class/gpio/gpio%d/value' % ledPin

def main(argv):
 DEFAULT_CONFIG_PATH = 'config.json'

 with open(DEFAULT_CONFIG_PATH, 'r') as config_file:
 config = json.load(config_file)
 print(config)

 artikcloud.configuration = artikcloud.Configuration()
 artikcloud.configuration.access_token = config['device_token']

 # create an instance of the API class
 api_instance = artikcloud.MessagesApi()
 count = 1
 start_date = int(time.time()*1000) - 86400000 # 24시간
 전(ms)
 end_date = int(time.time()*1000) # 현재 시간(ms)
 order = 'desc'

try:
 # Get Normalized Actions
 api_response = api_instance.get_normalized_actions(count=count,
 end_date = end_date, start_date = start_date, order = order)
 pprint(api_response)
 actionName = api_response.data[0].data.actions[0].name
 if actionName == "setOn":
 pinVAL = open(path_val, "wb", 0)
 pinVAL.write(str(1))
 print('LED Turned On')
 pinVAL.close()
 else:
 pinVAL = open(path_val, "wb", 0)
 pinVAL.write(str(0))
 print('LED Turned Off')
 pinVAL.close()

except ApiException as e:
 print("Exception when calling MessagesApi->get_normalized_actions: %s\n" % e)
```

## ■ Source Code (get-action.py) -(2)

```
if __name__ == "__main__":

 # export
 pinCTL = open(path_export, "wb", 0)
 try:
 pinCTL.write(str(ledPin))
 print "Exported pin", str(ledPin)
 except:
 print "Pin ", str(ledPin), "has been exported"
 pinCTL.close()

 # direction
 pinDIR = open(path_dir, "wb", 0)
 try:
 pinDIR.write("out")
 print "Set pin ", str(ledPin), "as digital output"
 except:
 print "Failed to set pin direction"
 pinDIR.close()

 # value & unexport
 while True:
 main(sys.argv[1:])
 time.sleep(5)
```

## ■ Temperature-controlled LED using Python SDK

- Step 7

- Background : send-temp.py

*# python send-temp.py & 또는 # python send-temp.py → [ctrl + z]*

- Foreground : get-action.py

*# python get-action.py*

- 백그라운드 작업 목록 확인 *#jobs*

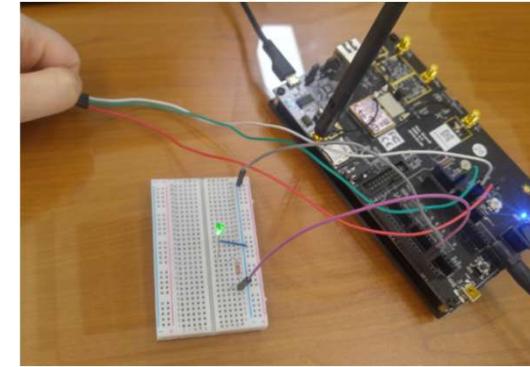
- 백그라운드 작업 종료

*# kill %1 또는 # fg %1* (1번 작업일 경우) 로 foreground에 불러온 후 종료

# REST API Protocol

## ■ Execution result

|                               |                         |
|-------------------------------|-------------------------|
| Temper is 29.1986816406       | Temper is 28.0121142578 |
| Exported pin 135              | LED Turned Off          |
| Set pin 135 as digital output | Temper is 27.7441796875 |
| LED Turned Off                | LED Turned Off          |
| Temper is 28.1269433594       | Temper is 33.2942529297 |
| LED Turned Off                | LED Turned On           |
| Temper is 31.4569873047       | Temper is 37.23671875   |
| LED Turned Off                | LED Turned On           |



|                      |                             |                             |                             |
|----------------------|-----------------------------|-----------------------------|-----------------------------|
| Temp to LED artik520 | Feb 21 2018<br>13:35:03.988 | Feb 21 2018<br>13:35:03.988 | {"temp":37.23671874999994}  |
| Temp to LED artik520 | Feb 21 2018<br>13:34:58.671 | Feb 21 2018<br>13:34:58.671 | {"temp":33.2942529296875}   |
| Temp to LED artik520 | Feb 21 2018<br>13:34:53.357 | Feb 21 2018<br>13:34:53.357 | {"temp":27.74417968749994}  |
| Temp to LED artik520 | Feb 21 2018<br>13:34:48.081 | Feb 21 2018<br>13:34:48.081 | {"temp":28.0121142578125}   |
| Temp to LED artik520 | Feb 21 2018<br>13:34:42.748 | Feb 21 2018<br>13:34:42.748 | {"temp":31.4569873046875}   |
| Temp to LED artik520 | Feb 21 2018<br>13:34:37.437 | Feb 21 2018<br>13:34:37.437 | {"temp":28.126943359375005} |
| Temp to LED artik520 | Feb 21 2018<br>13:34:32.054 | Feb 21 2018<br>13:34:32.054 | {"temp":29.198681640624997} |

Messages



|                                                                     |                             |                                             |        |
|---------------------------------------------------------------------|-----------------------------|---------------------------------------------|--------|
| Temp to LED artik520<br>Temp to LED artik520<br>6b7d1037b50e4ba2... | Feb 21 2018<br>13:35:04.057 | Temp to LED artik520<br>6b7d1037b50e4ba2... | setOn  |
| Temp to LED artik520<br>Temp to LED artik520<br>6b7d1037b50e4ba2... | Feb 21 2018<br>13:34:58.791 | Temp to LED artik520<br>6b7d1037b50e4ba2... | setOn  |
| Temp to LED artik520<br>Temp to LED artik520<br>6b7d1037b50e4ba2... | Feb 21 2018<br>13:34:53.467 | Temp to LED artik520<br>6b7d1037b50e4ba2... | setOff |
| Temp to LED artik520<br>Temp to LED artik520<br>6b7d1037b50e4ba2... | Feb 21 2018<br>13:34:48.188 | Temp to LED artik520<br>6b7d1037b50e4ba2... | setOff |
| Temp to LED artik520<br>Temp to LED artik520<br>6b7d1037b50e4ba2... | Feb 21 2018<br>13:34:42.868 | Temp to LED artik520<br>6b7d1037b50e4ba2... | setOff |
| Temp to LED artik520<br>Temp to LED artik520<br>6b7d1037b50e4ba2... | Feb 21 2018<br>13:34:37.534 | Temp to LED artik520<br>6b7d1037b50e4ba2... | setOff |
| Temp to LED artik520<br>Temp to LED artik520<br>6b7d1037b50e4ba2... | Feb 21 2018<br>13:34:32.173 | Temp to LED artik520<br>6b7d1037b50e4ba2... | setOff |

Actions

## ■ 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

## ■ 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_ ";
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);
```

## ■ Source Code (2)

```
delay(5000);
}

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

 DebugSerial.println("Connected!");
}

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));
}

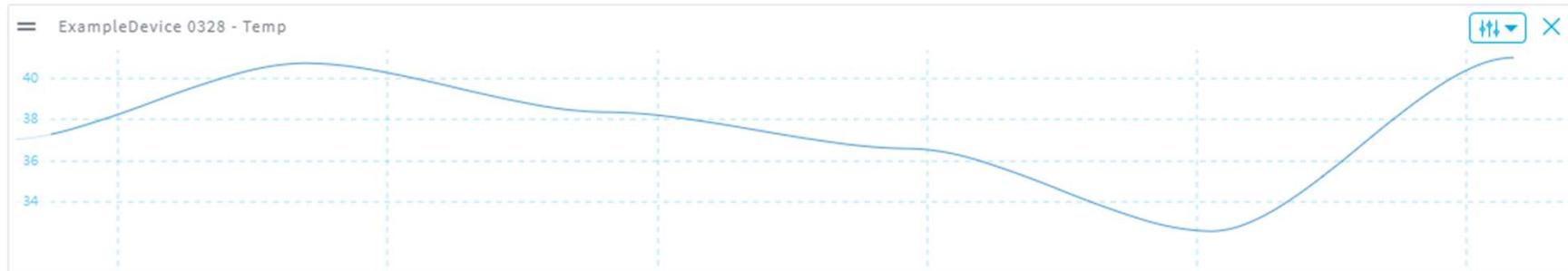
void readSensor() {
 voltage_raw0 = analogRead(0);
 voltage_raw1 = analogRead(1);

 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);
}
```

## ■ 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.



# MQTT Protocol

# Execution Result

- You can use Action messages to control LED or motor.

## II. ARTIK 기술 교육

### 12. ARTIK Cloud (5) - Scenes



# Scenes

## ■ Send a bunch of actions using Scenes

- Step 1 : Create Devices
  - Create some devices to test Scenes

The image shows two side-by-side screenshots of the Set Up / Manifests interface for different scenes.

**ScenesEx1:** The interface shows a manifest version V1 labeled as CURRENT. It includes a "DOWNLOAD MANIFEST" section and a "Actions" section. The "Actions" section contains two entries: "lock" and "unlock". The entire "Actions" section is highlighted with a red box.

**ScenesEx2:** The interface shows a manifest version V1 labeled as CURRENT. It includes a "DOWNLOAD MANIFEST" section and a "Actions" section. The "Actions" section contains three entries: "This\_is\_Example", "exAction1", and "exAction2". The entire "Actions" section is highlighted with a red box.

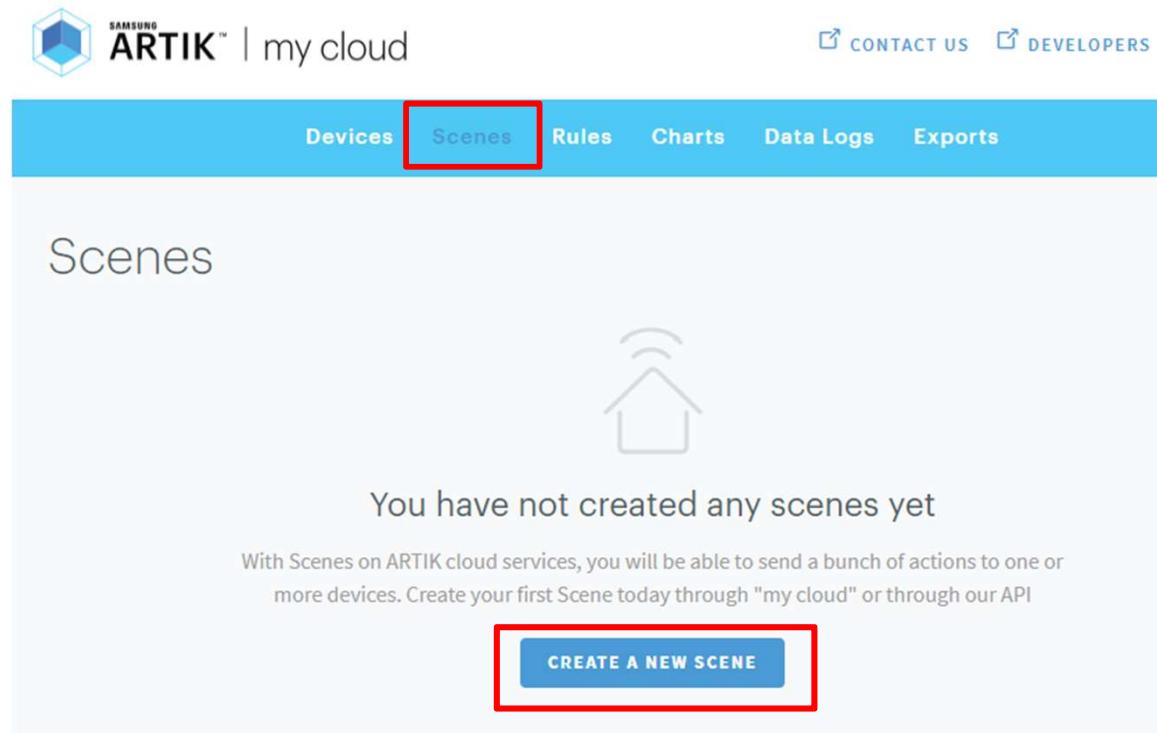
**SceneDevice1 (ScenesEx1):** A separate window shows the configuration for SceneDevice1. It lists "SceneDevice1" under "ScenesEx1" and indicates "Unknown Availability".

**SceneDevice2 (ScenesEx2):** A separate window shows the configuration for SceneDevice2. It lists "SceneDevice2" under "ScenesEx2" and indicates "Unknown Availability".

# Scenes

## ■ Send a bunch of actions using Scenes

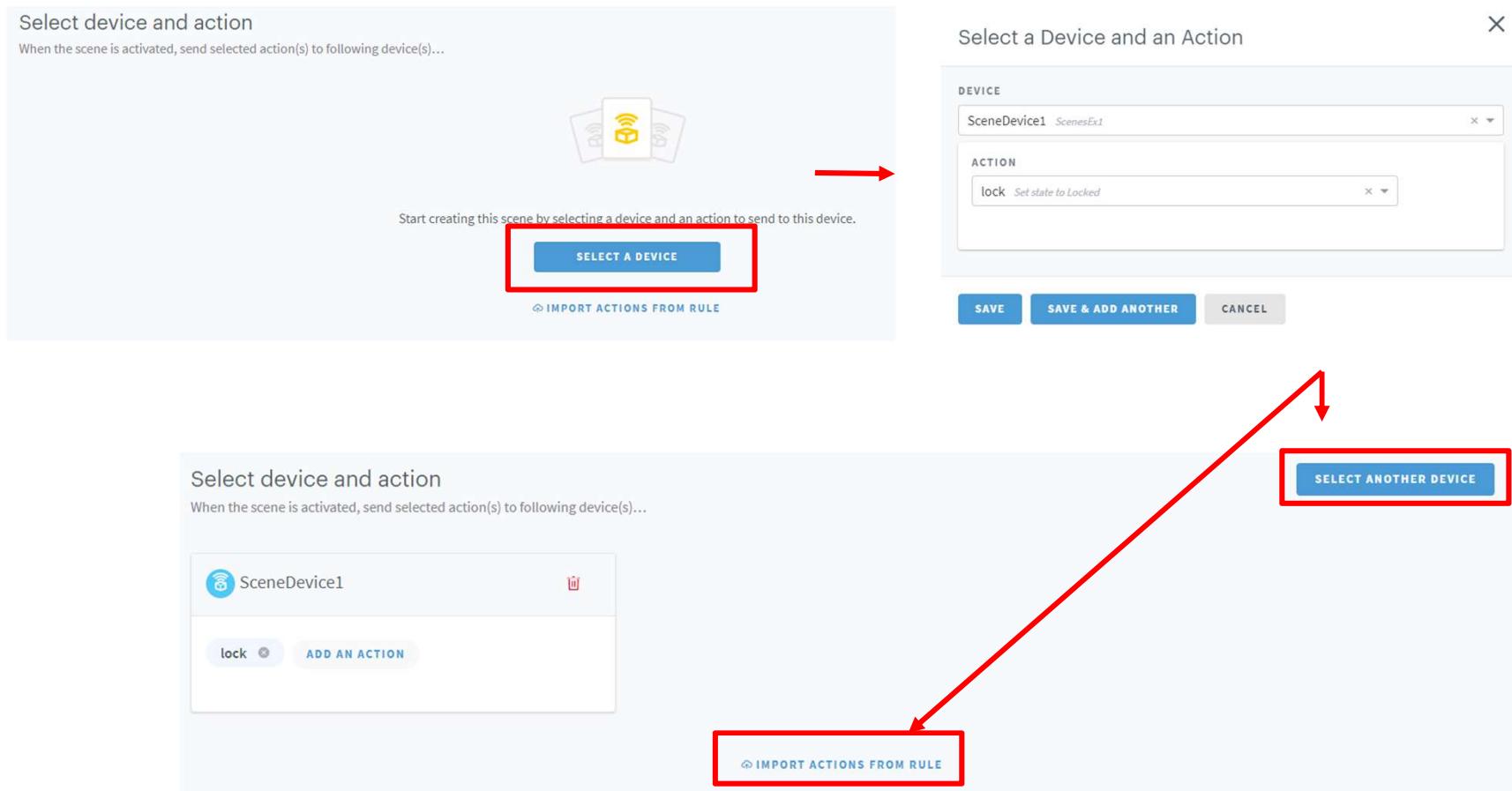
- Step 2 : Create Scenes
  - Click 'Scenes' – [Create a new scenes]



# Scenes

## ■ Send a bunch of actions using Scenes

- Step 3 : Select Devices / Import Actions from Rules



# Scenes

## ■ Send a bunch of actions using Scenes

- Step 3 : Select Devices / Import Actions from Rules

The image displays two overlapping windows from a software interface:

**Edit Action** window (left):

- DEVICE:** SceneDevice2\_ScenesEx2
- ACTION:** This\_is\_Example example action
- EXACTION1:** 1
- EXACTION2:** 0

**Import Actions From Rule** window (right):

- SELECT A RULE:**
  - Send setOff to 180116 LED  
send to 180116 LED the action setOff
  - Send setOn to 180116 LED  
send to 180116 LED the action setOn
  - Send setOn to Temp to LED artik520  
send to Temp to LED artik520 the action setOn
  - Send setOff to Temp to LED artik520

Both windows have a red box highlighting the "SELECT ANOTHER DEVICE" button in the top left of the first window and the "IMPORT ACTIONS FROM RULE" button in the top right of the second window.

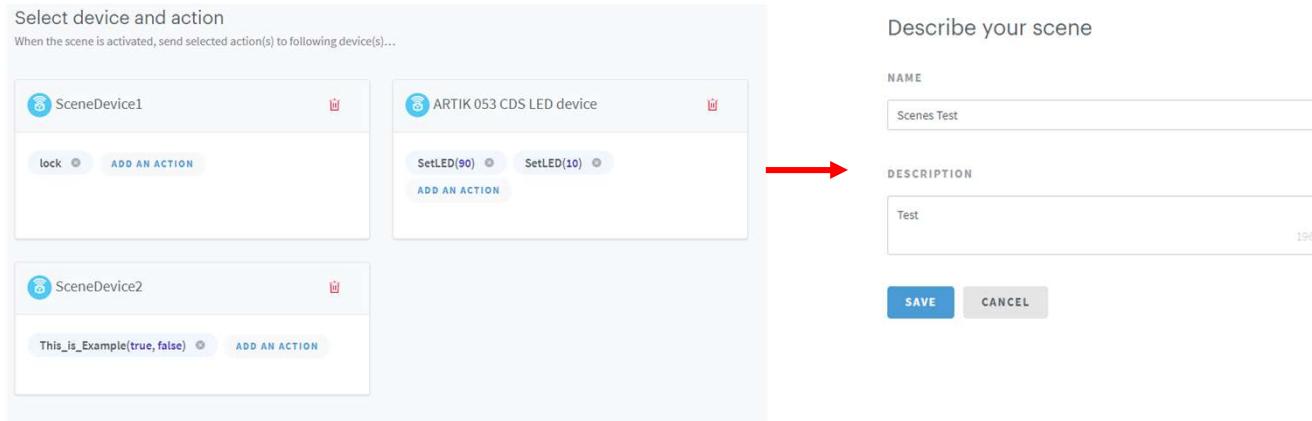
# Scenes

## ■ Send a bunch of actions using Scenes

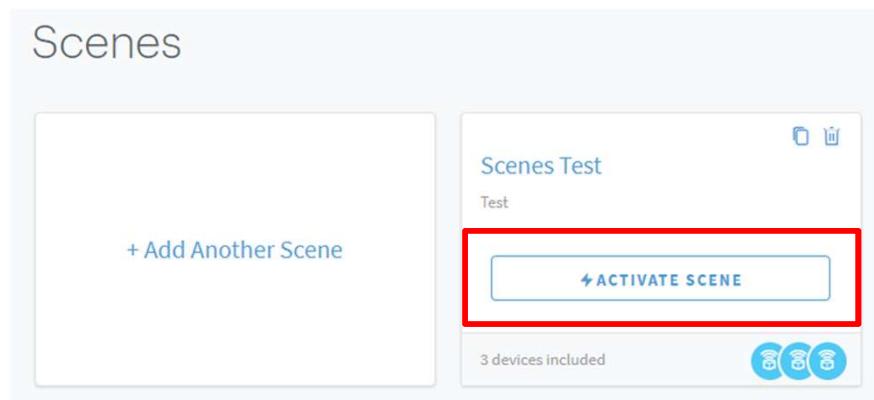
### ▪ Step 4 : Save Scene

Select device and action

When the scene is activated, send selected action(s) to following device(s)...



### ▪ Step 5 : Activate Scene



# Scenes

## ■ Send a bunch of actions using Scenes

- Step 6 : Check Data logs

A screenshot of a data log table. The table has columns: DEVICE, RECORDED AT, RECEIVED AT, and DATA. A single row is shown: "Scenes Test" at "Mar 29 2018 16:31:33.091" received at "Mar 29 2018 16:31:33.101". The "DATA" column contains the JSON object: {"activatedOn":1522308693072}. The entire row is highlighted with a red box. To the right of the table, the text "Message from Scene" is displayed.

| DEVICE      | RECORDED AT              | RECEIVED AT              | DATA                          |
|-------------|--------------------------|--------------------------|-------------------------------|
| Scenes Test | Mar 29 2018 16:31:33.091 | Mar 29 2018 16:31:33.101 | {"activatedOn":1522308693072} |

A screenshot of a data log table. The table has columns: DESTINATION DEVICE, SENT AT, SOURCE DEVICE, ACTION NAME, and ACTION PARAMETERS. Four rows are shown. The first three rows have their entire content highlighted with a red box. The fourth row is partially visible. The data is as follows:

| DESTINATION DEVICE                                                     | SENT AT                  | SOURCE DEVICE                       | ACTION NAME     | ACTION PARAMETERS                    |
|------------------------------------------------------------------------|--------------------------|-------------------------------------|-----------------|--------------------------------------|
| ARTIK 053 CDS LED device<br>ARTIK 053 CDS LED<br>644fb0ffe8a3464ba2... | Mar 29 2018 16:31:33.201 | Scenes Test<br>a9c294592fe1421fb... | SetLED          | {"brightness":10}                    |
| ARTIK 053 CDS LED device<br>ARTIK 053 CDS LED<br>644fb0ffe8a3464ba2... | Mar 29 2018 16:31:33.201 | Scenes Test<br>a9c294592fe1421fb... | SetLED          | {"brightness":90}                    |
| SceneDevice2<br>ScenesEx2<br>2e8415699aa144829...                      | Mar 29 2018 16:31:33.201 | Scenes Test<br>a9c294592fe1421fb... | This_is_Example | {"exAction1":true,"exAction2":false} |
| SceneDevice1<br>ScenesEx1<br>8de159637cd0492681...                     | Mar 29 2018 16:31:33.201 | Scenes Test<br>a9c294592fe1421fb... | lock            | None                                 |

Actions from Devices

## II. ARTIK 기술 교육

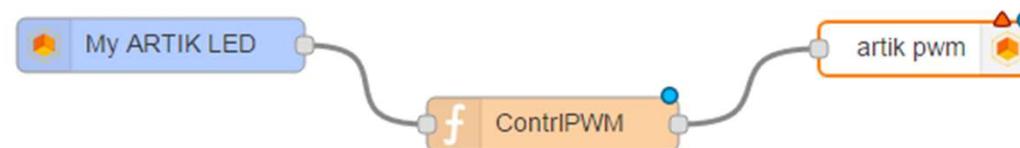
### 13. ARTIK Cloud (6) - Node-RED



# 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.



# Node-RED

## ■ Install Node-RED

### ▪ Step 1 : Connect to Wi-Fi

- Use the DHCP client services program dhclient.
- Check whether DHCP client succeeded in 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!
```

## ■ 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

## ■ 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
```

## ■ 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.

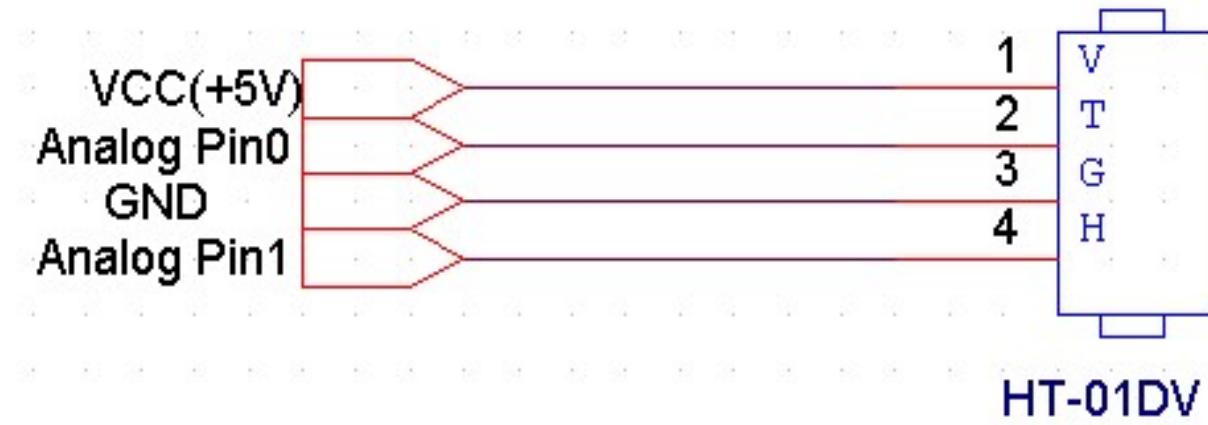
```
npm cache clean
npm install -g --unsafe-perm node-red
cd ~/.node-red
npm rebuild
```

# Node-RED Ex. Read H/T Sensor

## ■ Required Hardware

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

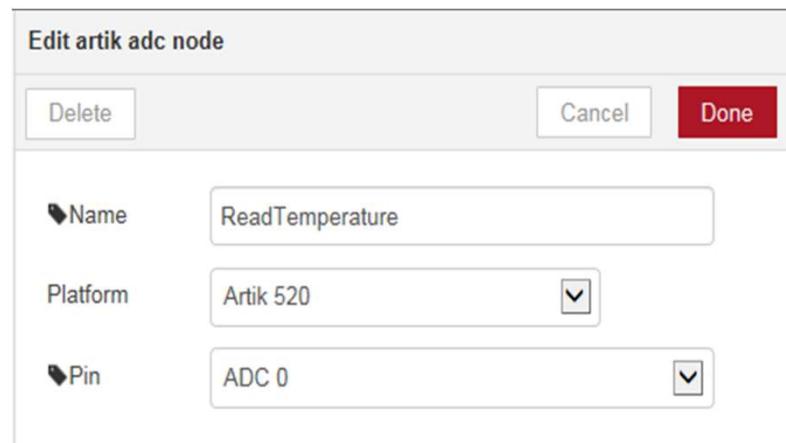
## ■ Circuit Configuration



# Node-RED Ex. Read H/T Sensor

## ■ Developing Node-RED application

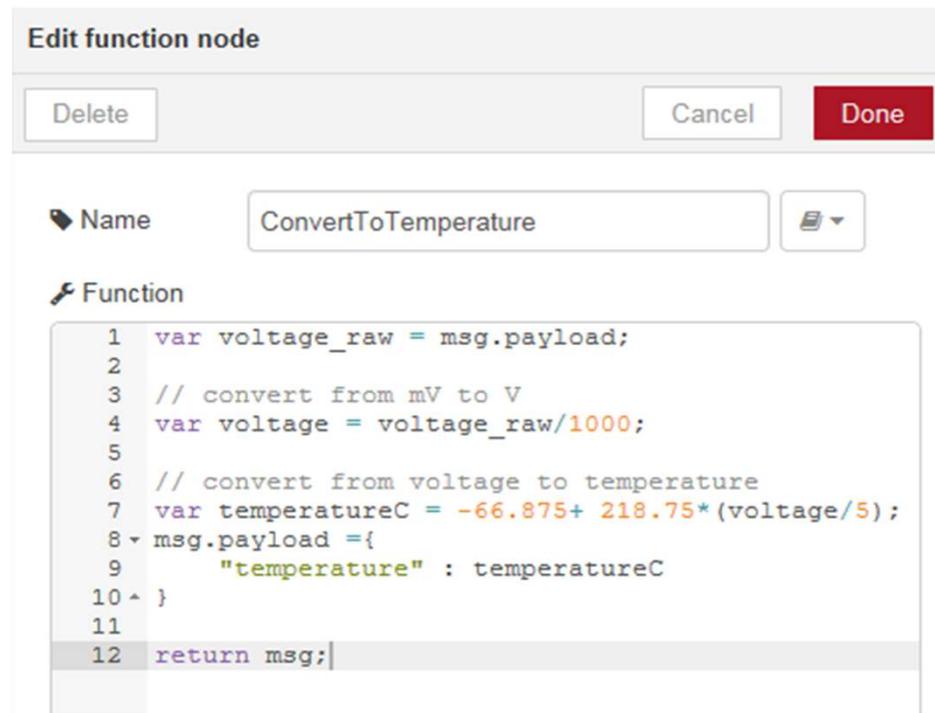
- Step 1 : Select an “Inject” node
  - Click and drag an “Inject” input node from the palette in left pane to the canvas.
  - Then, “Inject” will change to “timestamp”.
  - Configure the inject node to trigger data reading every 5s.
- Step 2 : Select an “Artik ADC” node and double-click to set :
  - “Artik 520” as target platform
  - “ADC 0” as the analog pin from which data will be read.
  - Name as “ReadTemperature”



# Node-RED Ex. Read H/T Sensor

## ■ Developing Node-RED application

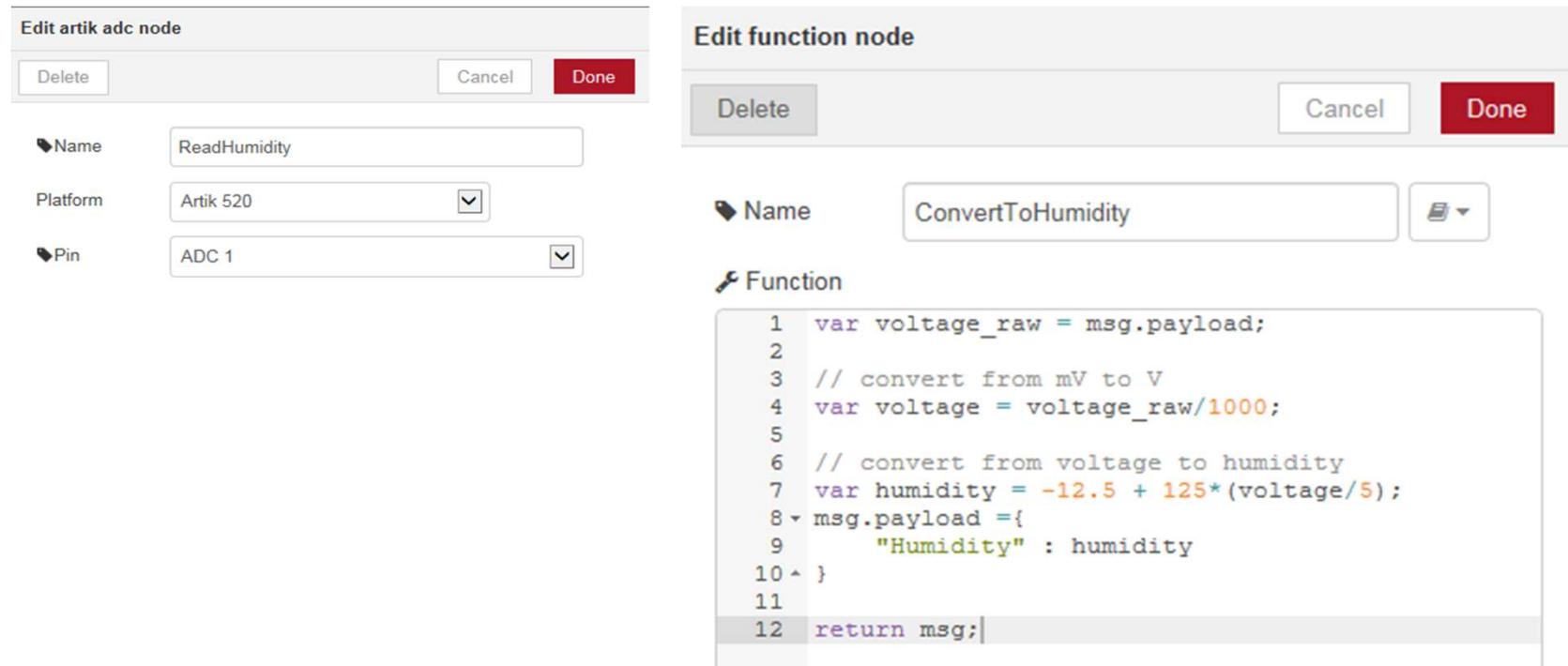
- Step 3 : Select “function” node
  - Drag a function node to the right of the aritk adc node.
  - It will convert the voltage(mV) reading from ADC 0 pin to a temperature.
  - Double-click to define the function as following.



# Node-RED Ex. Read H/T Sensor

## ■ Developing Node-RED application

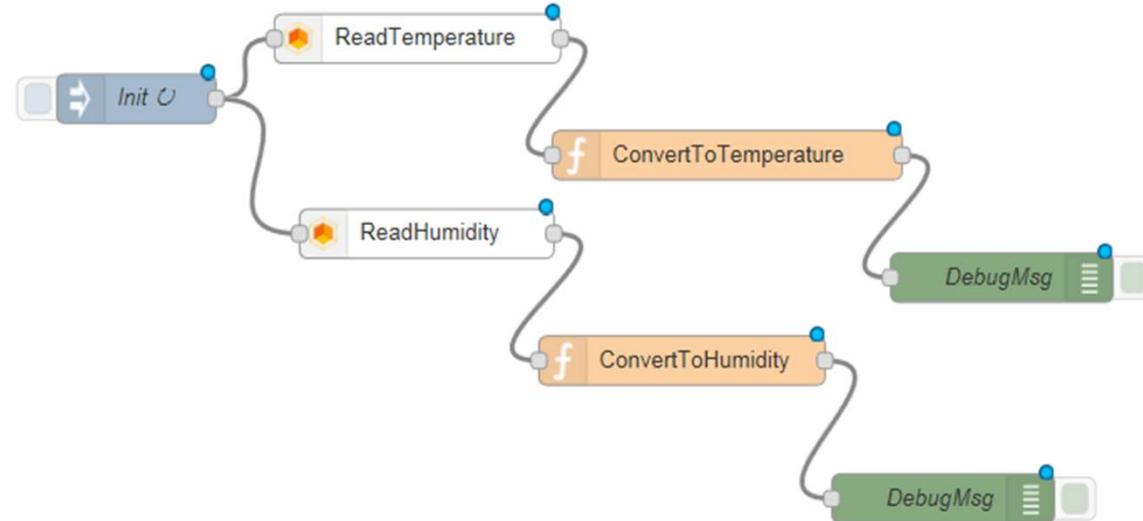
- Step 4 : Select "Artik ADC" node and "Function" node
  - Set ADC 1 as a pin of artik adc node.
  - Define the function as following, where will convert voltage(mV) to humidity.



# Node-RED Ex. Read H/T Sensor

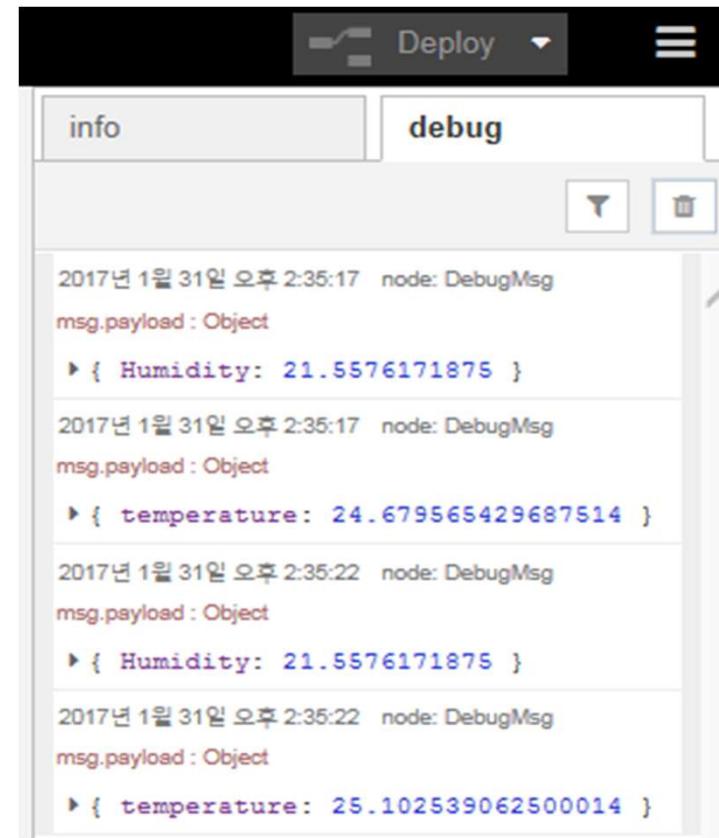
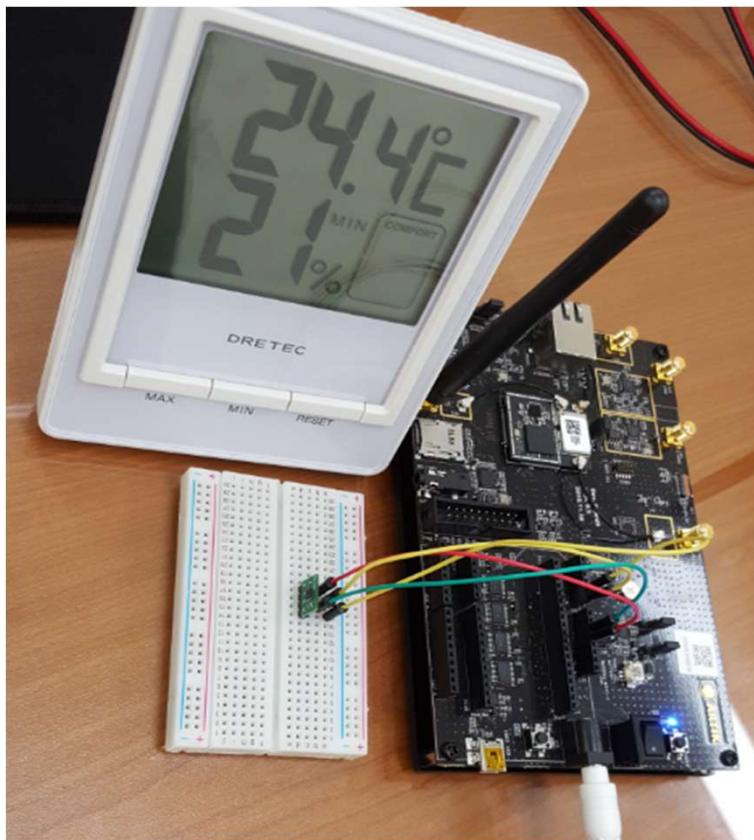
## ■ Developing Node-RED application

- 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 }
```

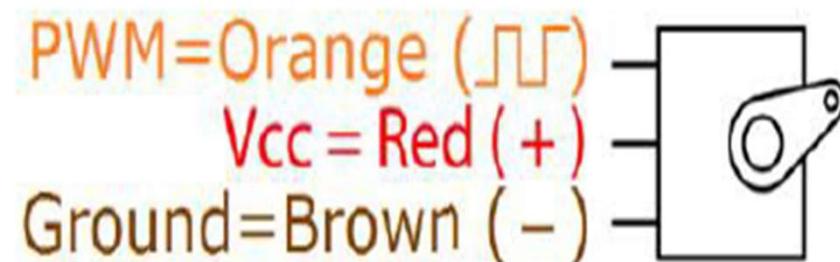
# Node-RED Ex. Control Servo Motor

## ■ 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.

```
1 var newMsg = {
2 payload:{
3 state:1,
4 period:20000000,
5 dutyCycle:2300000
6 }
7 }
8 return newMsg;
```

```
1 var newMsg = {
2 payload:{
3 state:1,
4 period:20000000,
5 dutyCycle:1500000
6 }
7 }
8 return newMsg;
```

# Node-RED Ex. Control Servo Motor

## ■ Developing Node-RED application

The image shows two separate 'Edit function node' windows from the Node-RED interface. Both windows have a 'Name' field and a 'Function' code editor.

**Left Window (|90):**

- Name: |90
- Function:

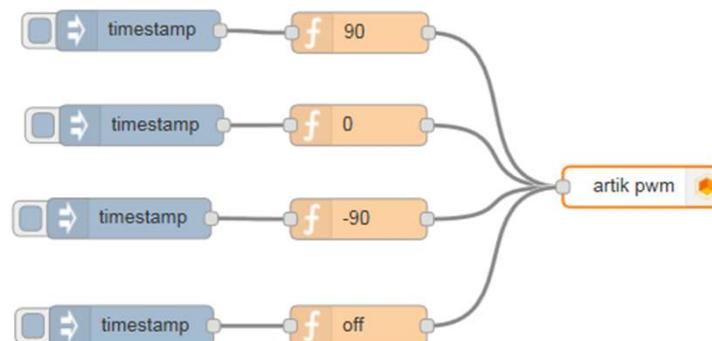
```
1 * var newMsg = {
2 * payload:{
3 * state:1,
4 * period:2000000,
5 * dutyCycle:700000
6 * }
7 * }
8 * return newMsg;
```

**Right Window (|off):**

- Name: |off
- Function:

```
1 * var newMsg = {
2 * payload:{
3 * state:0
4 * }
5 * }
6 * return newMsg;
```

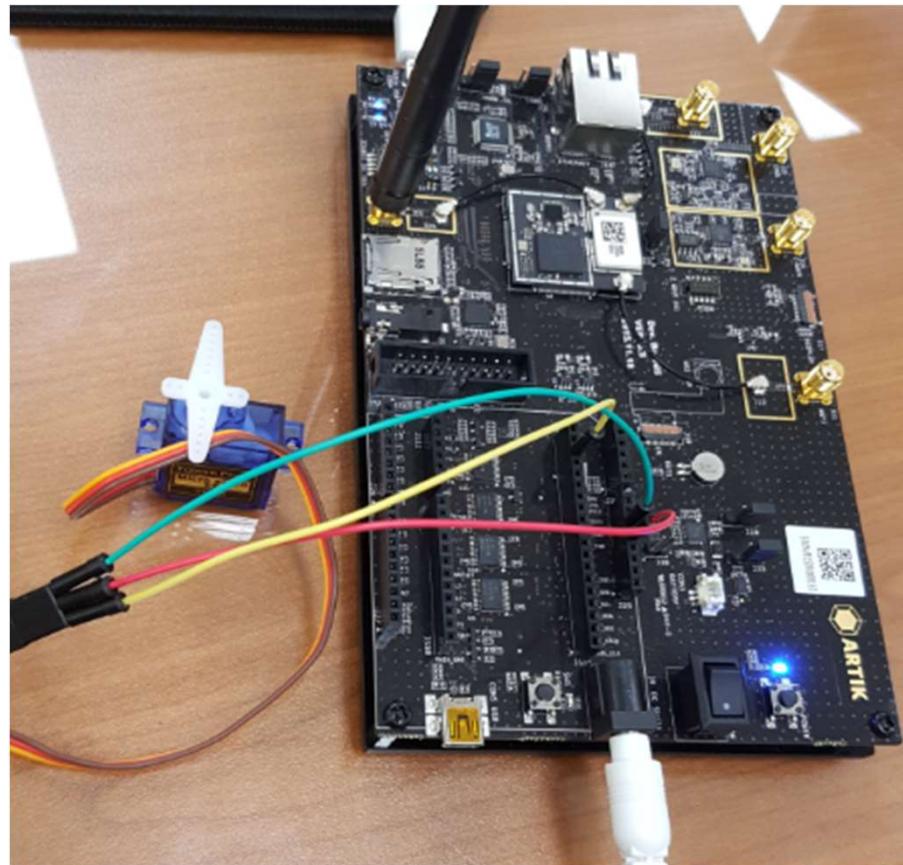
- Step 3 : Wire up four inject nodes to the beginning of the flow and click "Deploy".



# 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 click 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.

The image consists of two side-by-side screenshots from the ARTIK Cloud interface.

**Left Screenshot: Connect another device**

This screenshot shows the initial step of connecting a new device. It features a search bar with the text "Temp Sensor", a field to "NAME YOUR NEW DEVICE" containing "Temp Sensor", and a blue "CONNECT DEVICE..." button.

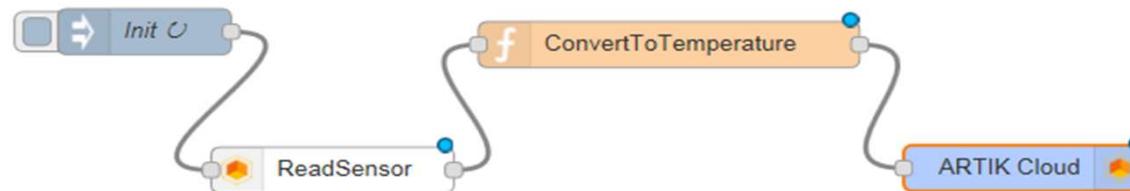
**Right Screenshot: Device Info**

This screenshot shows the details of a connected device named "Temp Sensor". It includes fields for "DEVICE TYPE" (Temp Sensor), "CONNECTED SINCE" (January 24, 2017), "LAST DATA TRANSFER" (Never), and "DEVICE ID" (highlighted with a red box). Below this, there is a "DEVICE TOKEN" field (also highlighted with a red box) which contains a long string of characters, and a "REVOKE TOKEN" link. At the bottom are "SAVE CHANGES" and "DELETE" buttons.

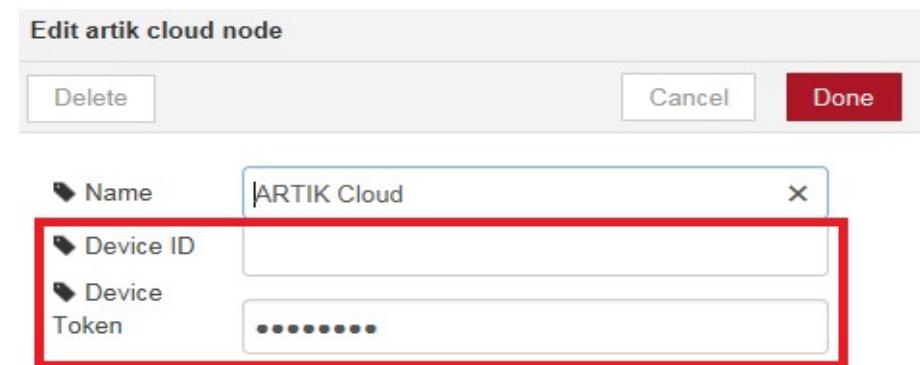
# Node-RED and ARTIK Cloud

## ■ 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.



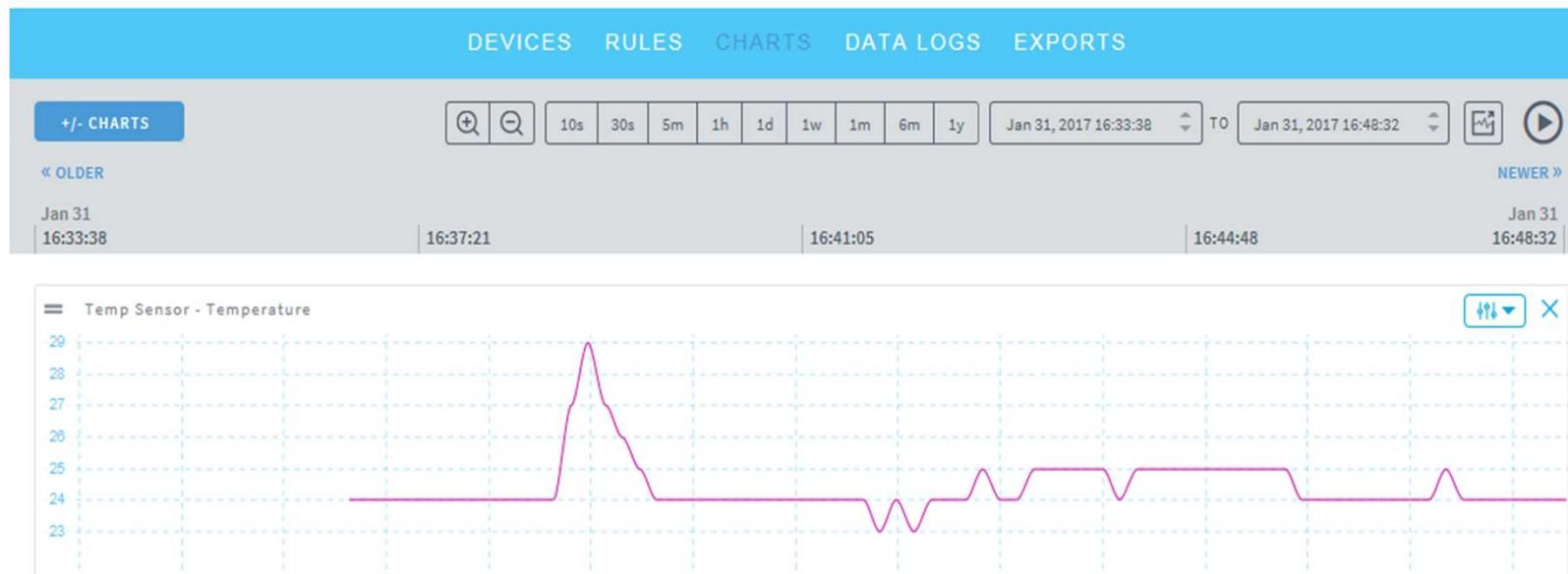
- Configure "ARTIK Cloud" output node as follows. Enter from the Device Info page.



# Node-RED and ARTIK Cloud

## ■ Sending Node-RED data to ARTIK Cloud

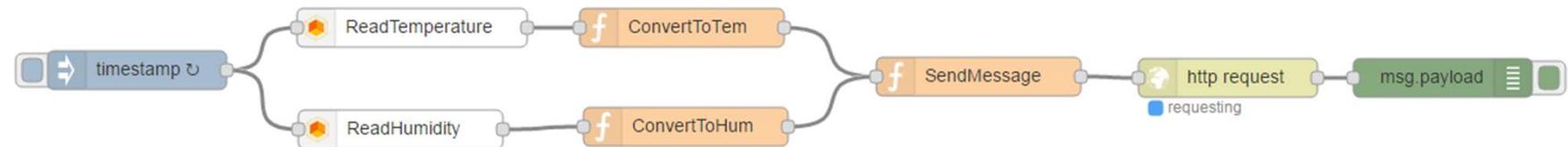
- Step 4 : Click "Deploy"
  - Go back to My ARTIK Cloud and CHARTS, then you can see the streamed data.



# Node-RED and ARTIK Cloud

## ■ Sending Node-RED 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.



# Node-RED and ARTIK Cloud

## ■ Sending Node-RED data to ARTIK Cloud via REST protocol

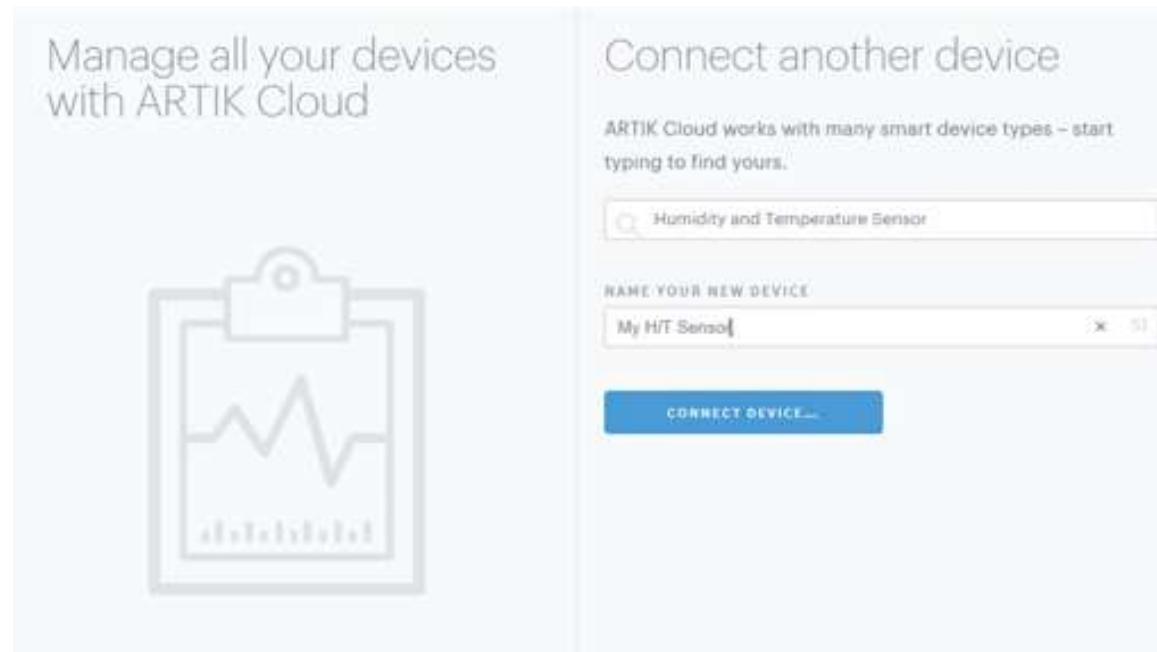
- Step 1 : Create a new Device type and set Manifest.
  - Go to ARTIK Cloud Developer site and create a new device type named as "Humidity and Temperature Sensor".
  - Set Manifest to add two Device Fields as follows.

The screenshot shows the 'Simple Manifest' section of the ARTIK Cloud developer portal. On the left, there's a sidebar titled 'New Manifest' with a document icon and a plus sign. The main area has a title 'Humidity and Temperature Sensor' with a Wi-Fi icon, and a 'Switch to Advanced' link. Below the title, a descriptive text states: '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.' A 'LEARN MORE »' link is present. The interface is divided into three sections: 'Device Fields', 'Device Actions', and 'Activate Manifest'. The 'Device Fields' section contains two fields: 'humidity' (type: DOUBLE) and 'temperature' (type: DOUBLE). Each field has an edit icon. At the bottom of this section are buttons for '+ NEW FIELD' and '+ NEW FIELD GROUP'. At the very bottom are 'NEXT: DEVICE ACTIONS' and 'CANCEL' buttons.

# Node-RED and ARTIK Cloud

## ■ Sending Node-RED data to ARTIK Cloud via REST protocol

- Step 2 : 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.



# Node-RED and ARTIK Cloud

## ■ Sending Node-RED data to ARTIK Cloud via REST protocol

- Step 3 : 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.

Edit function node

Delete Cancel Done

Name: SendMessage

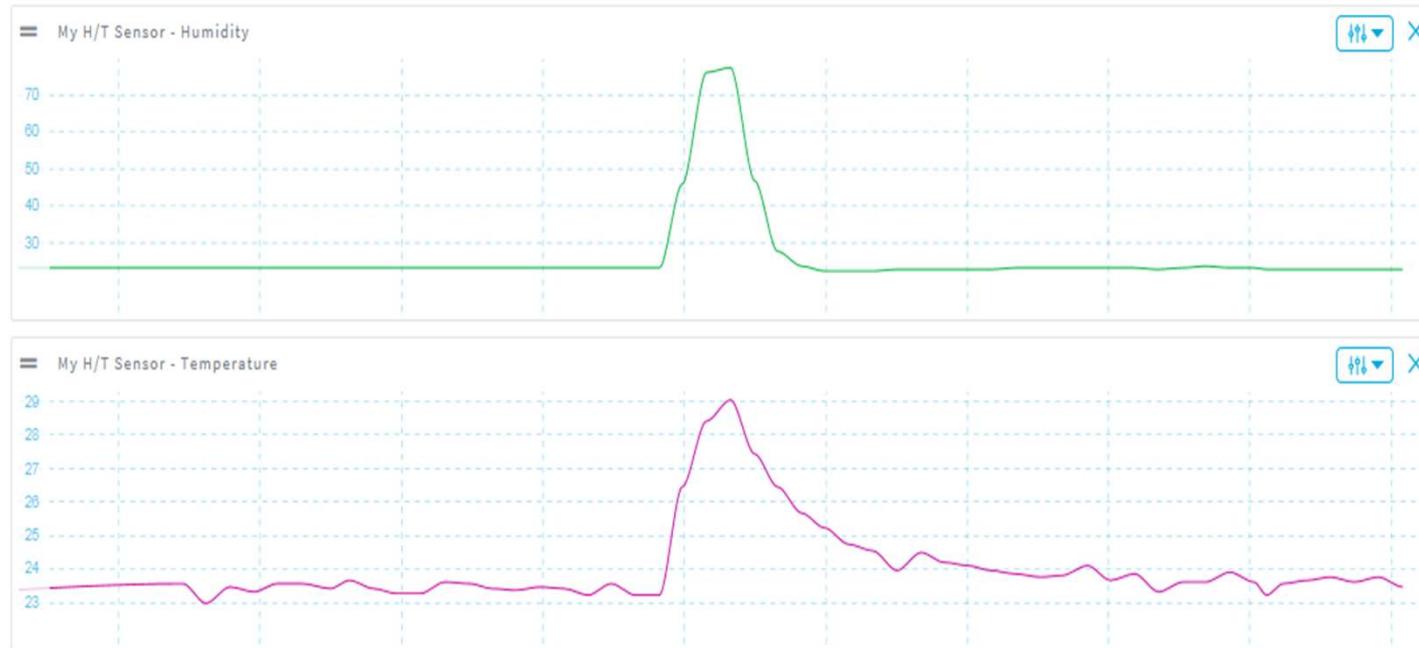
Function

```
1 var tem = msg.payload.temperature;
2 var hum = msg.payload.humidity;
3
4 msg.headers = {
5 "Content-Type": "application/json",
6 "Authorization": "Bearer device token
7 }
8
9 msg.payload = {
10 "sdid": "device id",
11 "type": "message",
12 "data": {
13 "temperature": tem,
14 "humidity": hum
15 }
16 }
17
18 return msg;
```

# Node-RED and ARTIK Cloud

## ■ Sending Node-RED data to ARTIK Cloud via REST protocol

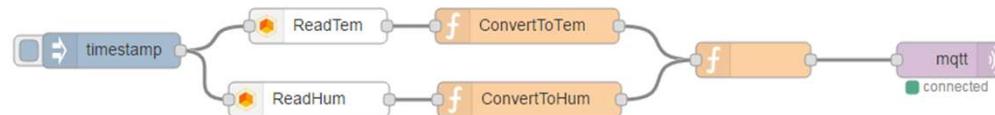
- Step 4 : Click "Deploy" and go back to My ARTIK Cloud CHARTS.



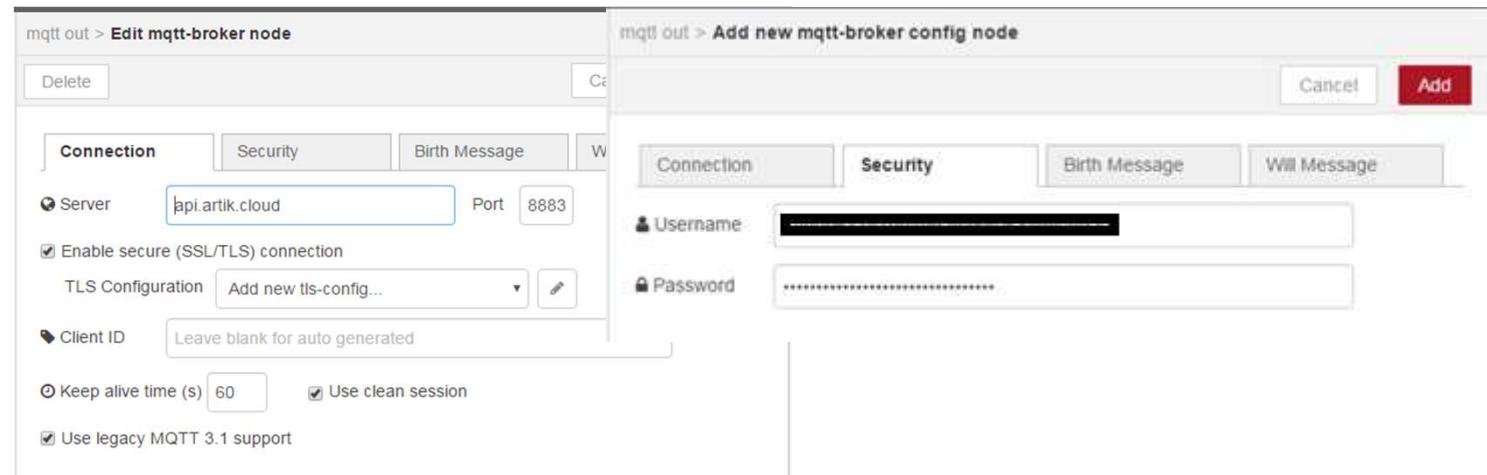
# Node-RED and ARTIK Cloud

## ■ Sending Node-RED 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".



# Node-RED and ARTIK Cloud

## ■ Sending Node-RED data to ARTIK Cloud via MQTT protocol

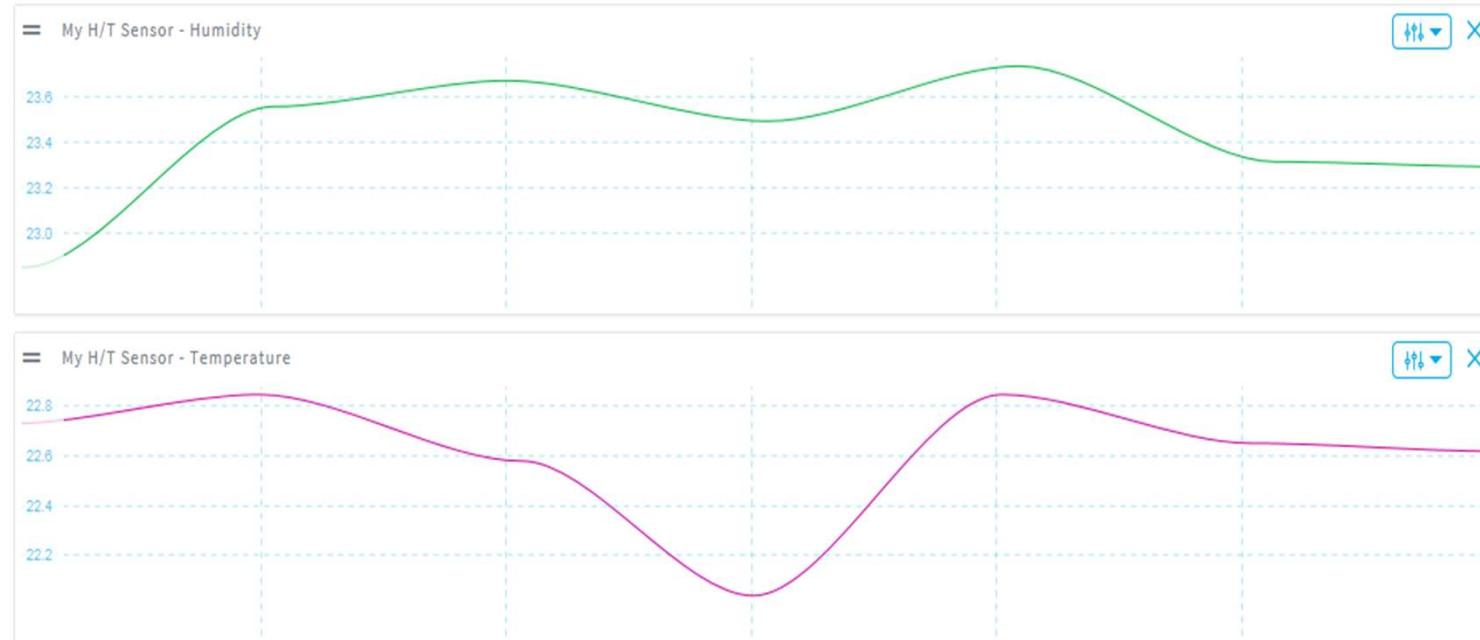
- Step 3 : Edit “function” node
  - You don't need msg.headers since MQTT is a lightweight protocol.
  - Add a topic as shown with your Device ID inserted.



# Node-RED and ARTIK Cloud

## ■ Sending Node-RED data to ARTIK Cloud via MQTT protocol

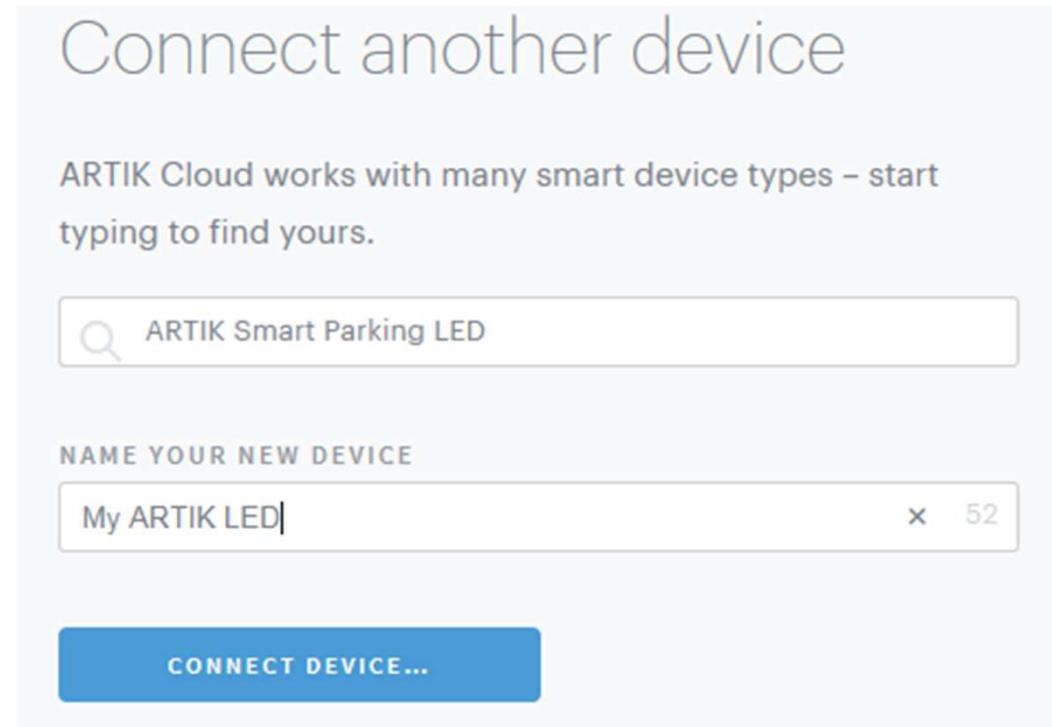
- Step 4 : Click "Deploy" and go to My ARTIK Cloud CHARTS



# 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.

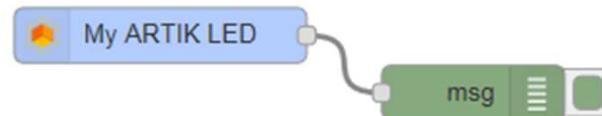


# Node-RED and ARTIK Cloud

## ■ 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°C, ARTIK Cloud sends actions "setOn".
  - If temperature is more than or equal to 26°C, 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"
```

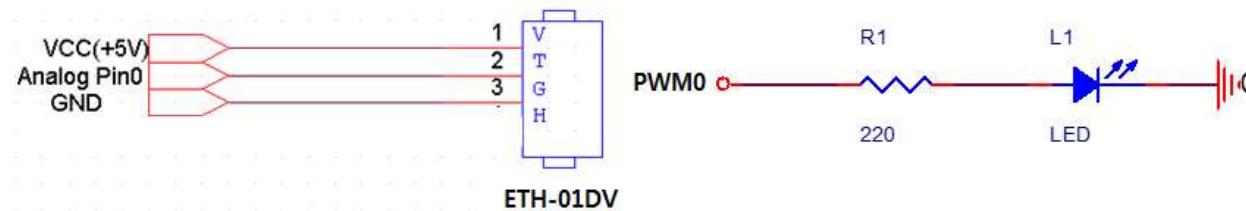
```
2017. 2. 2. 오후 3:07:47 node: ef1093fe.beeee
msg.payload : Object
▶ { temperature: 27.973837890624992 }

2017. 2. 2. 오후 3:07:48 node: Debug
msg : Object
 ▶ object
 ▶ actions: array[1]
 ▶ 0: object
 name: "setOff"
 ▶ parameters: object
 _msgid: "e8722d42.578dd"
```

# Node-RED and ARTIK Cloud

## ■ Using Actions, control LED on/off

- Step 1 : Configure circuit as follows.



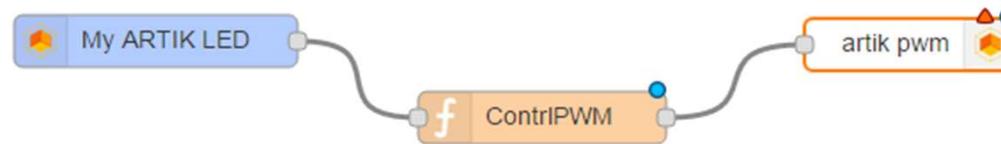
- Step 2 : In Node-RED, select "function" node.
  - It will check whether we received a "setOn" or "setoff" action.



# 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°C, LED blink.
  - If temperature is more than or equal to 26°C, you can see LED turned off.

## II. ARTIK 기술 교육

### 14. Android App 연동



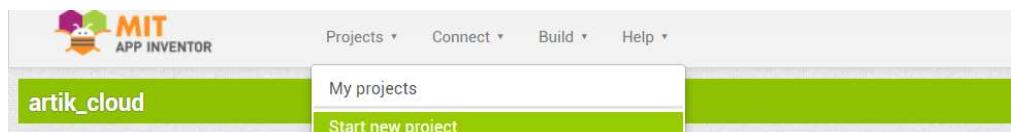
# Android App Inventor

## ■ App Inventor

- <http://appinventor.mit.edu.explore>
- [Create Apps!]



- [Projects] – [Start new Project]



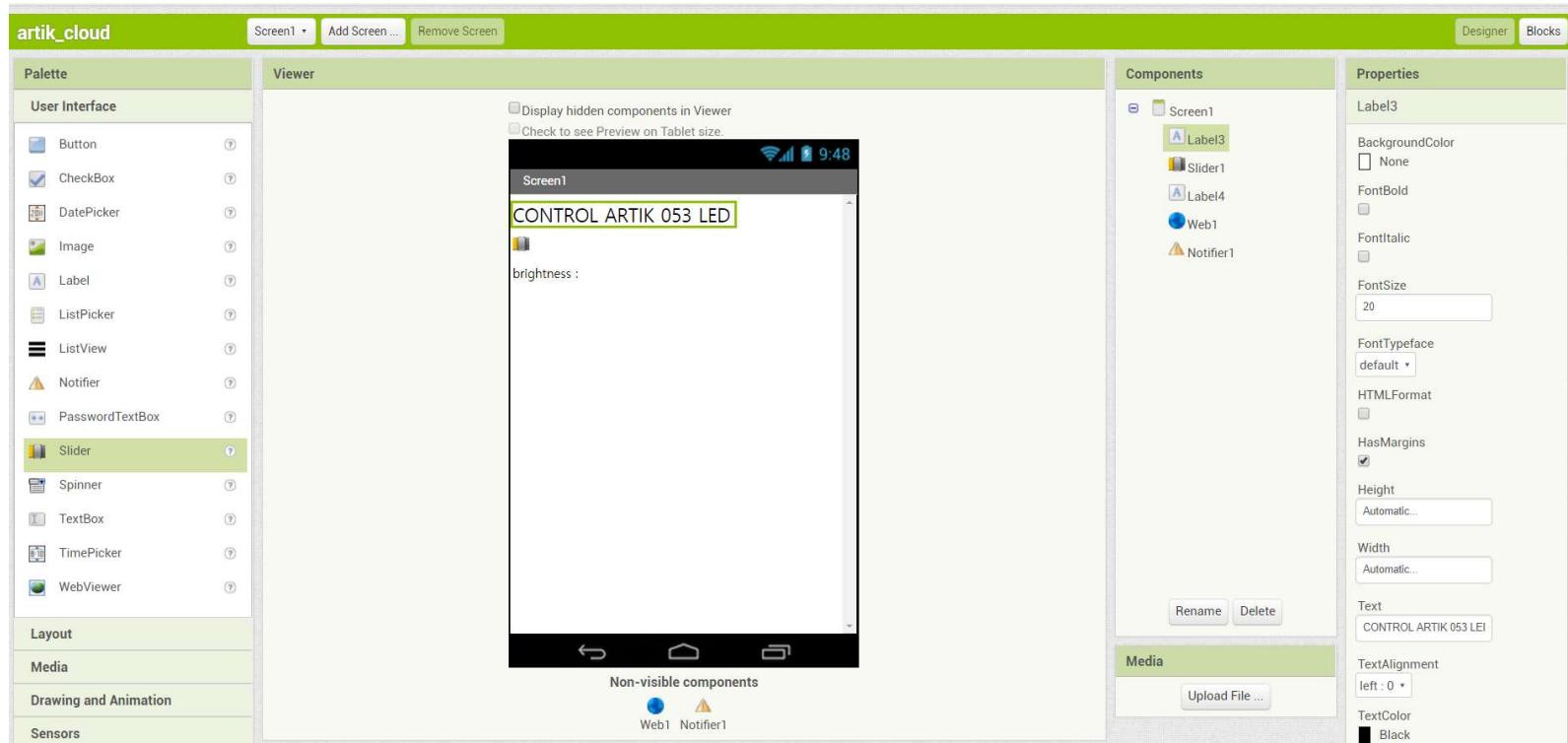
# Android App Inventor

## ■ App Inventor Example

- App inventor로 Application을 구현하여 클라우드에 Action 전송
- 이를 통해 ARTIK 520에 연결된 LED 밝기 제어

# Android App Inventor

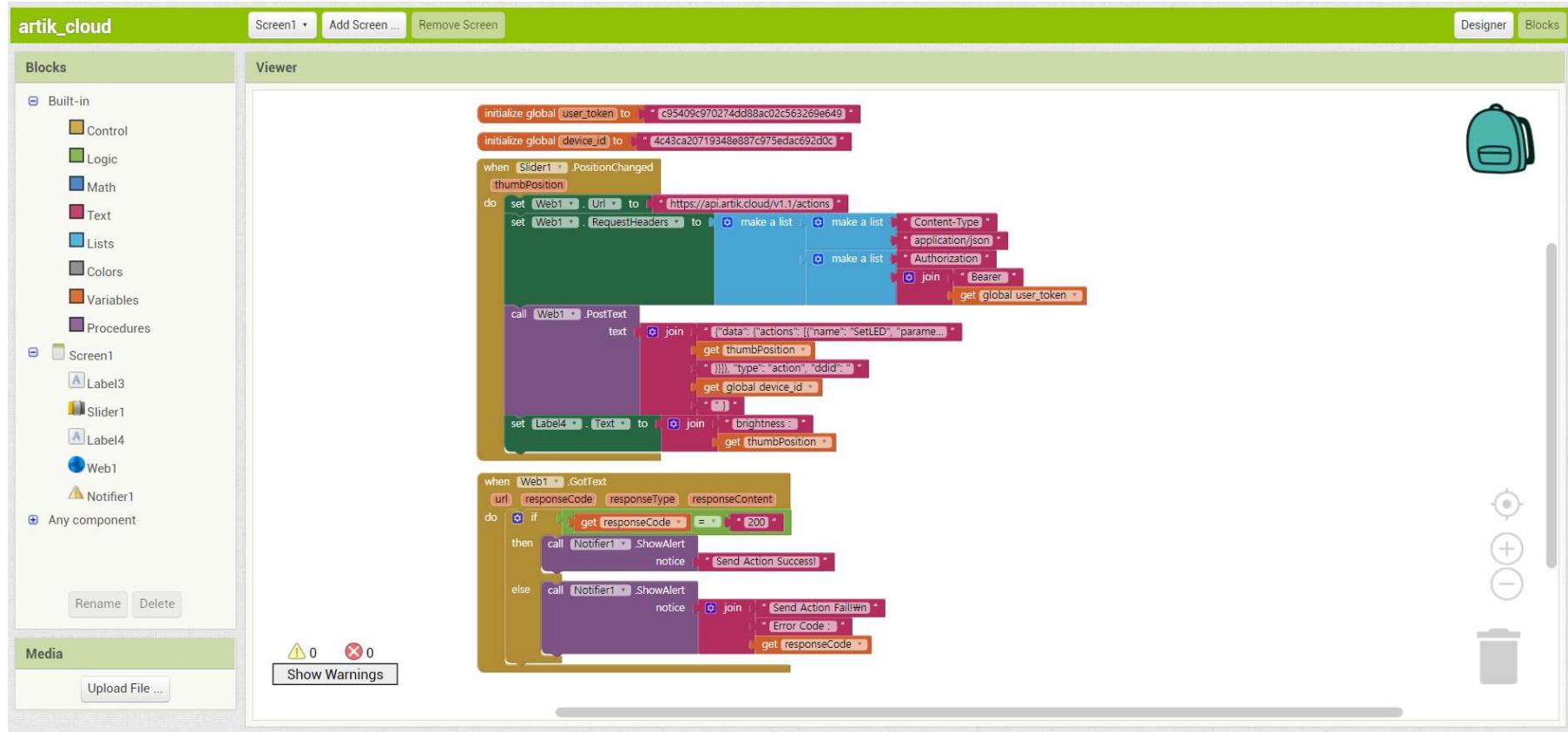
## ■ App Inventor : Designer Tab



| Components | Description                 |
|------------|-----------------------------|
| Label      | Text를 표시한다.                 |
| Notifier   | 일시적인 알림을 표시한다.              |
| Slider     | 슬라이드 바의 커서를 움직일 때 Event를 발생 |
| Web        | HTTP Request & Response     |

## ■ App Inventor : Block Tab

- Application에서 event 발생 시 수행할 일을 설정 가능



## ■ App Inventor : Block Tab

- 전역 변수 설정
  - USER TOKEN (API consol에서 얻을 수 있음)
  - Device ID
- Slider1.PositionChanged do
  - 슬라이더 위치 변경 시 event 발생
- set Seb1/Url to
  - HTTP Request URL 설정
- set Web1.RequestHeaders to
  - HTTP Request Header 설정
  - Header field name과 header field value로 이루어진 sublist들의 list
- call Web1.PostText text
  - text 데이터를 Request Body에 요청하여 POST method 요청
- set Label4.Text to
  - Label Text 설정

## ■ App Inventor : Block Tab

- When Web1.GotText do
  - HTTP Response 발생 시 url, responseCode, reponseType, responseContent를 변수로 저장
- If then else
  - If문 실행
- =
  - 비교 로직
- Call Notifier1.ShowAlert notice
  - 텍스트를 입력받아 알람을 띄운다

# Android App Inventor

## ■ How to get USER TOKEN

- <https://developer.artik.cloud/api-console/> -> **POST Send Actions API**
- TRY IT!

Response Code  
200

Response Headers

```
{ "allow": "*", "content-type": "application/json", "request-time": "1512051612", "x-rate-limit-limit": "100/100000", "x-rate-limit-reset": "1512051612/1512345600", "x-rate-limit-remaining": "99/99892", "x-quota-max-payload-size": "1024", "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": "Sun, 03 Dec 2017 07:11:42 GMT", "content-length": "53", "cache-control": "no-cache", "pragma": "no-cache", "strict-transport-security": "max-age=31536000; includeSubDomains; preload", "x-content-type-options": "nosniff", "connection": "close" }
```

Response Body [Select body](#)

```
{ "data": { "mid": "f632e2cd8a76457fad33d79c34a280ec" } }
```

- You can get USER TOKEN at Request Headers

[TRY IT!](#) [CLEAR RESULTS](#)

Call

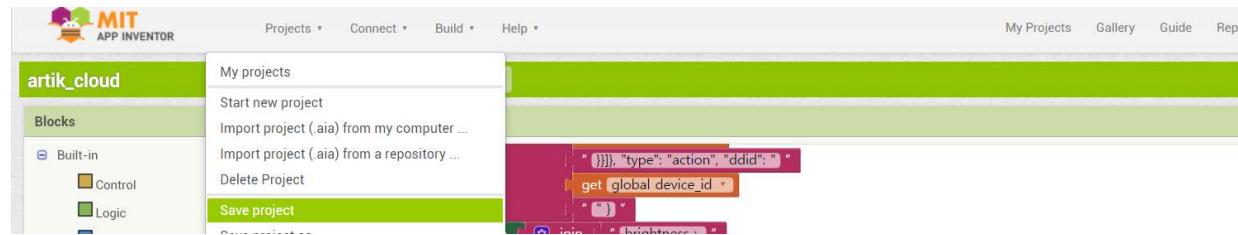
<https://api.artik.cloud/v1.1/actions>

Request Headers

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

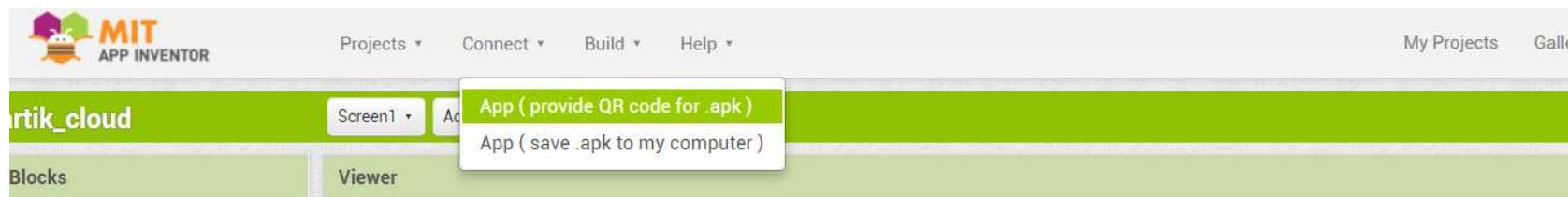
# Android App Inventor

## ■ Save Project

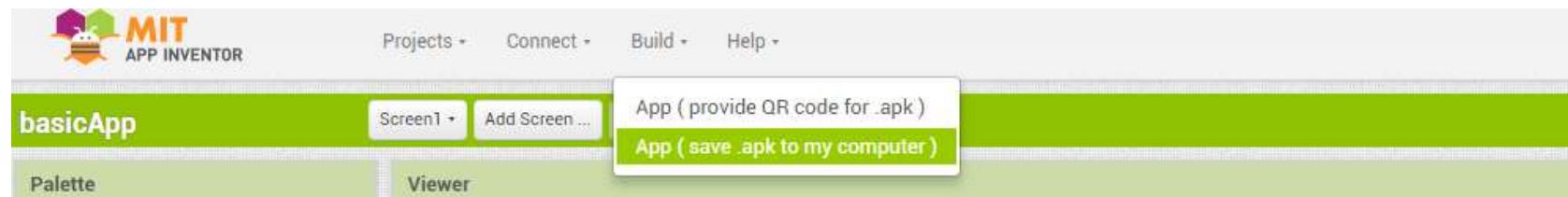


## ■ Download Application

- QR Code 를 통해 다운로드 (일반 QR Code App으로도 가능)

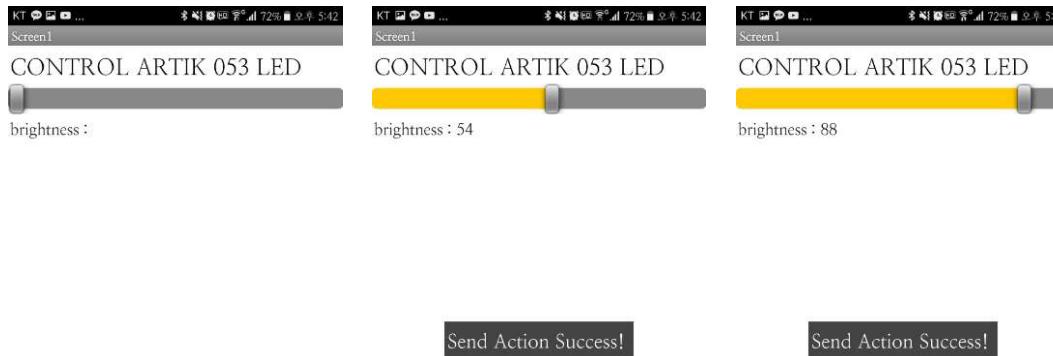


- PC에 직접 apk 파일 다운로드



# Android App Inventor

## ■ Download Application



- 연결한 ARTIK Cloud Device의 Action Parameter가 double 또는 float여야 함
- 그렇지 않을 경우(integer) 슬라이더가 정수 값이 아닐 시 에러 발생

## ■ Source Code (get-brightness)-(1)

```
import artikcloud
from artikcloud.rest import ApiException
import sys, getopt
import time, json
from pprint import pprint

ledPin = 135
path_export = '/sys/class/gpio/export'
path_unexport = '/sys/class/gpio/unexport'
path_dir = '/sys/class/gpio/gpio%d/direction' % ledPin
path_val = '/sys/class/gpio/gpio%d/value' % ledPin

def main(argv):
 DEFAULT_CONFIG_PATH = 'config.json'

 with open(DEFAULT_CONFIG_PATH, 'r') as config_file:
 config = json.load(config_file)
 print(config)

 artikcloud.configuration = artikcloud.Configuration()
 artikcloud.configuration.access_token = config['device_token']

 # create an instance of the API class
 api_instance = artikcloud.MessagesApi()
 count = 1
 start_date = int(time.time()*1000) - 86400000 # 24시간
 전(ms)
 end_date = int(time.time()*1000) # 현재 시간(ms)
 order = 'desc'

 try:
 # Get Normalized Actions
 api_response = api_instance.get_normalized_actions(count=count,
end_date = end_date, start_date = start_date, order = order)
 pprint(api_response)
 actionName = api_response.data[0].data.actions[0].parameters
 if actionName["brightness"] >= 50:
 pinVAL = open(path_val, "wb", 0)
 pinVAL.write(str(1))
 print('LED Turned On')
 pinVAL.close()
 else:
 pinVAL = open(path_val, "wb", 0)
 pinVAL.write(str(0))
 print('LED Turned Off')
 pinVAL.close()

 except ApiException as e:
 print("Exception when calling MessagesApi->get_normalized_actions: %s\n" % e)
```

## ■ Source Code (get-brightness)-(2)

```
if __name__ == "__main__":

 # export
 pinCTL = open(path_export, "wb", 0)
 try:
 pinCTL.write(str(ledPin))
 print "Exported pin", str(ledPin)
 except:
 print "Pin ", str(ledPin), "has been exported"
 pinCTL.close()

 # direction
 pinDIR = open(path_dir, "wb", 0)
 try:
 pinDIR.write("out")
 print "Set pin ", str(ledPin), "as digital output"
 except:
 print "Failed to set pin direction"
 pinDIR.close()

 # value & unexport
 while True:
 main(sys.argv[1:])
 time.sleep(5)
```

## II. ARTIK 기술 교육

### 15. ARTIK Cloud Connector



# Introduction to Cloud Connector

## ■ Cloud Connector

- ARTIK Cloud can connect to the third-party cloud and use it as the data source for a device type.
- An infrastructure that enables you to connect the third-party cloud to ARTIK Cloud is called *Cloud Connectors*.
- ARTIK Cloud can retrieve the device's data sent to that cloud.

## ■ Eligibility

- You can build a Cloud Connector for the third-party cloud if it:
  - Uses OAuth1 or OAuth2 for authentication.
  - Provides subscribe/notification APIs for ARTIK Cloud to subscribe and receive notifications about new data.
  - Has public APIs to read data from the cloud.

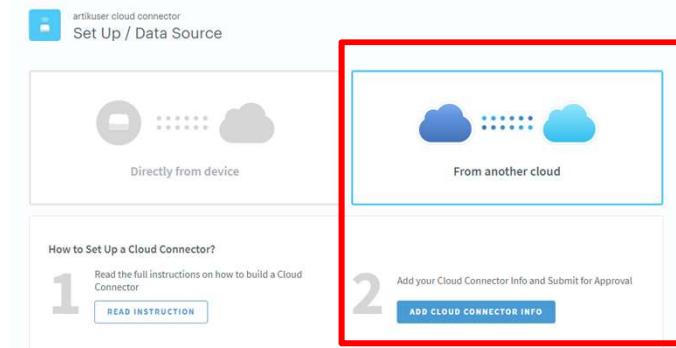
## ■ Building a Cloud Connector as a developer

- There are two parts to building a Cloud Connector:
  - Create and configure an ARTIK Cloud device type configured to receive data from a cloud service on ARTIK Cloud.
  - Create and configure an "ARTIK Cloud Connector" application on the third-party cloud. This enables ARTIK Cloud devices of the above type to interact with that cloud.

# Introduction to Cloud Connector

## ■ Steps to build a Cloud Connector

- In ARTIK Cloud Developers, create a new device type.
- After creating, enter the Data Source menu of the device type.
  - Check "From another cloud" → Click "Add Cloud Connection Info"
- In Cloud Authentication tap, four authentication methods can be selected.
  - OAuth1
  - OAuth2
  - Custom
  - None



The screenshots show the 'Cloud Authentication' configuration screen. The top screenshot shows the general configuration with a red box around the 'Cloud Authentication' section. The bottom screenshot shows a detailed view of the 'OAuth2' configuration with a red box around the 'AUTHENTICATION TYPE' dropdown, which is set to 'OAuth2'.

# Introduction to Cloud Connector

## ■ OAuth1 Parameters

| PARAMETER NAME         | DESCRIPTION                                                                                                                                                                                                                                                                                                                        | DEFAULT VALUE             |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|
| Request Token URL      | The request token URL provided by the third-party cloud.                                                                                                                                                                                                                                                                           |                           |
| AccessTokenURL         | The access token URL provided by the third-party cloud.                                                                                                                                                                                                                                                                            |                           |
| AuthorizationURL       | The authorization URL provided by the third-party cloud.                                                                                                                                                                                                                                                                           |                           |
| Consumer Key           | The consumer ID provided by the third-party cloud.                                                                                                                                                                                                                                                                                 |                           |
| Consumer Secret        | The consumer secret provided by the third-party cloud.                                                                                                                                                                                                                                                                             |                           |
| Signature Method       | The signature method to use in the Oauth 1 flow (HMAC-SHA1, RSA-SHA1, Plaintext)                                                                                                                                                                                                                                                   | HMAC-SHA1                 |
| Transmission Method    | When making an OAuth-authenticated request, protocol parameters as well as any other parameter using the "oauth_" prefix SHALL be included in the request using one and only one of the following locations, listed in order of decreasing preference: HTTP Authorization header, HTTP request entity-body, HTTP request URI query | HTTP Authorization header |
| Credentials Parameters | A map to identify the fields "userid", "token", "secret", "external_id" in the request                                                                                                                                                                                                                                             | Identity                  |

# Introduction to Cloud Connector

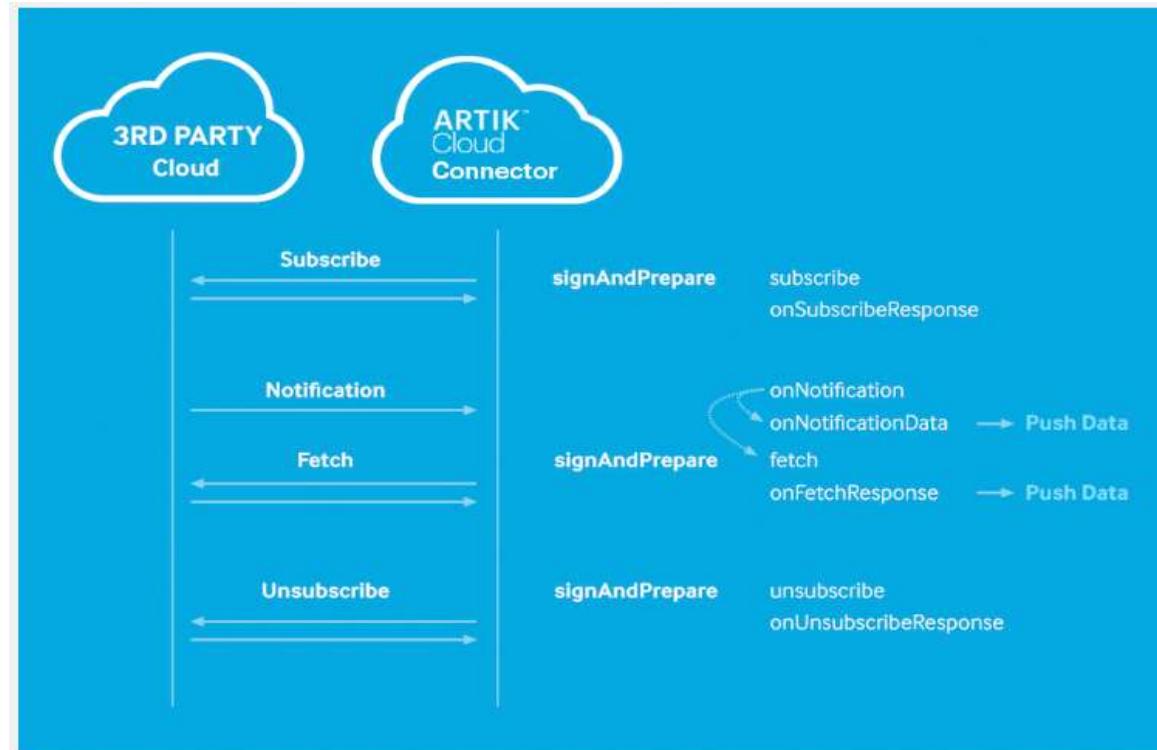
## ■ OAuth2 Parameters

| PARAMETER NAME     | DESCRIPTION                                                                                                                                                                                  | DEFAULT VALUE |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| Authorization URL  | The authorization URL provided by the third-party cloud.                                                                                                                                     |               |
| Access Token URL   | The access token URL provided by the third-party cloud.                                                                                                                                      |               |
| Client ID          | The client ID provided by the third-party cloud.                                                                                                                                             |               |
| Client Secret      | The client secret provided by the third-party cloud.                                                                                                                                         |               |
| scope              | The OAuth 2 scope parameter provided by the the third-party cloud.                                                                                                                           |               |
| Credentials Params | A map to identify the following fields in the request: <code>access_token</code> , <code>token_type</code> , <code>refresh_token</code> , <code>expires_in</code> , <code>external_id</code> | Identity      |

# Introduction to Cloud Connector

## ■ Cloud Connector Groovy Code

- Cloud Connector uses subscribe/notification APIs to access data in three steps:
  - Subscribe: subscribe to notification.
  - Notification: Receive notification.
  - Fetch: Call endpoint(s) to get data if needed.



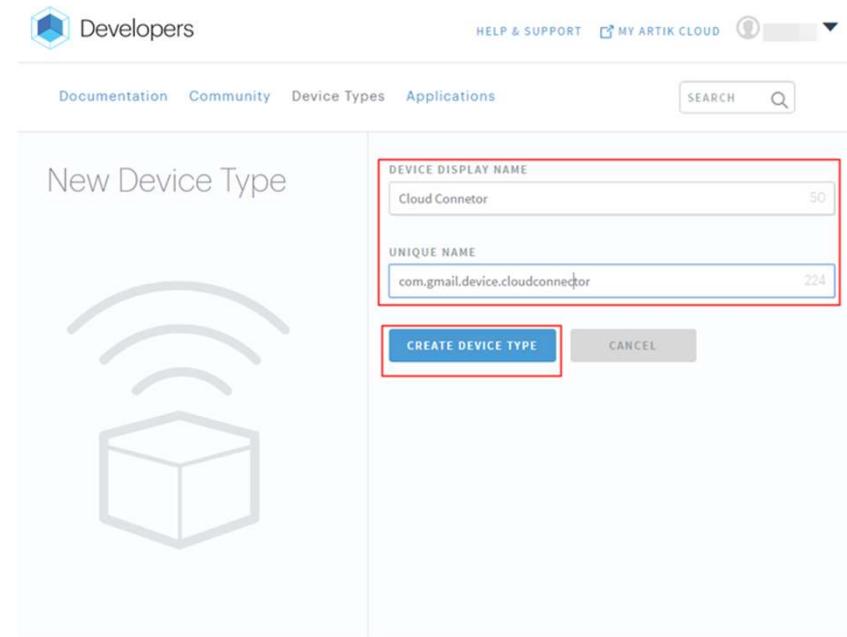
# Cloud Connector for the Moves Cloud

## ■ Moves Cloud

- Moves is an Android/iOS app that tracks activity data.
- Moves Cloud uses OAuth2 as Authentication.

## ■ Build a Cloud Connector for the Moves

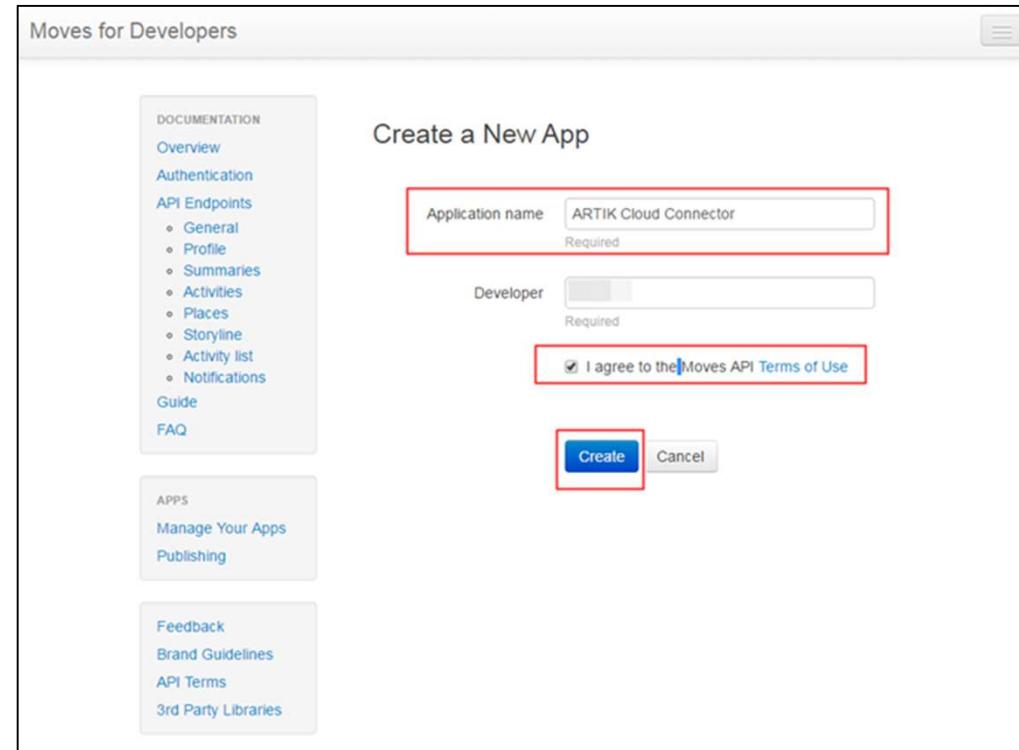
- Step1 : Create a new Device Type.
  - Go to the ARTIK Cloud Developers site, and log in.
  - <https://developer.artik.cloud>
  - Create a new device type 'Cloud Connector.'



# Cloud Connector for the Moves Cloud

## ■ Build a Cloud Connector for the Moves

- Step2 : Create a Moves app.
  - Log into the Moves Developer Portal.
  - <https://dev.moves-app.com/>
  - Click on "Manage Your Apps" and then "Create a New App."
  - Give your app a name like "ARTIK Cloud Connector", enter a developer name, and click "Create".

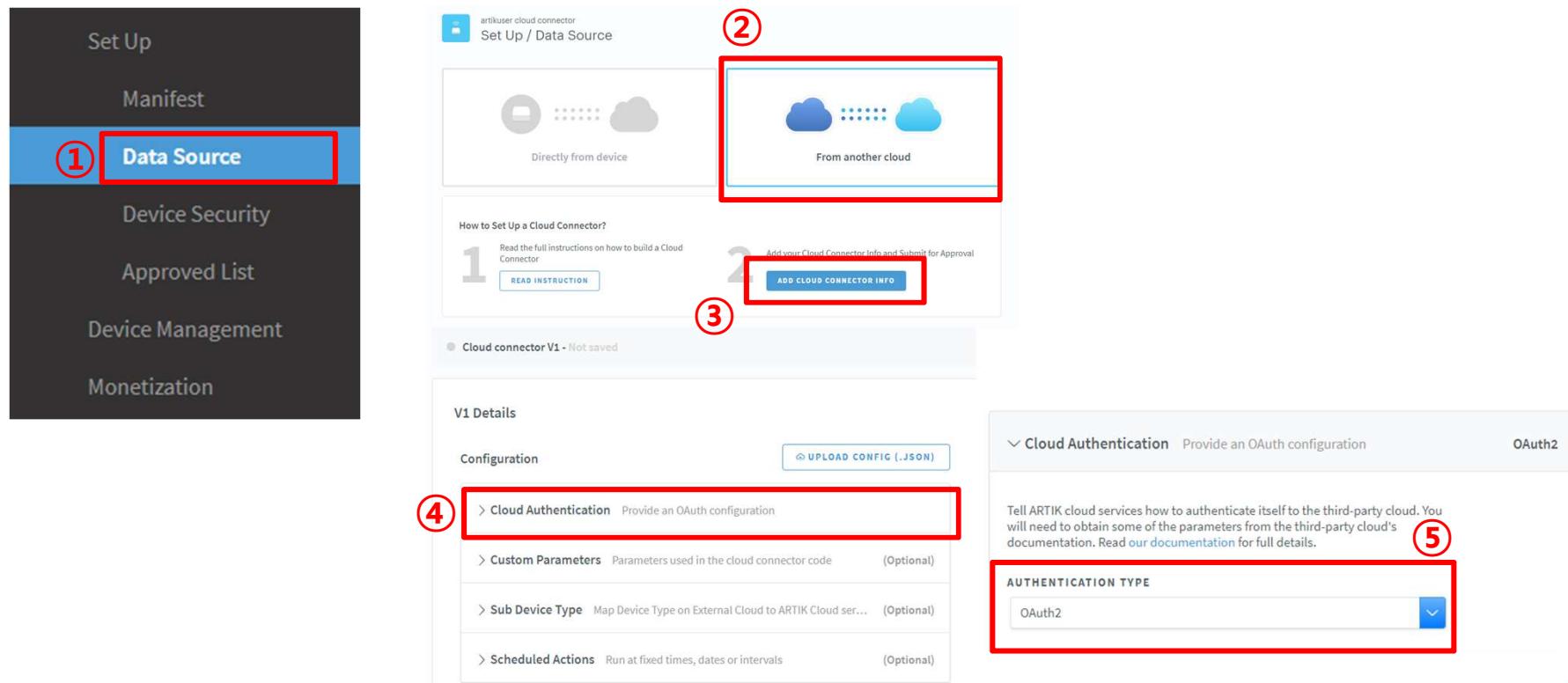


# Cloud Connector for the Moves Cloud

## ■ Build a Cloud Connector for the Moves

- Step3 : Configure ARTIK Cloud Authentication to Moves Cloud

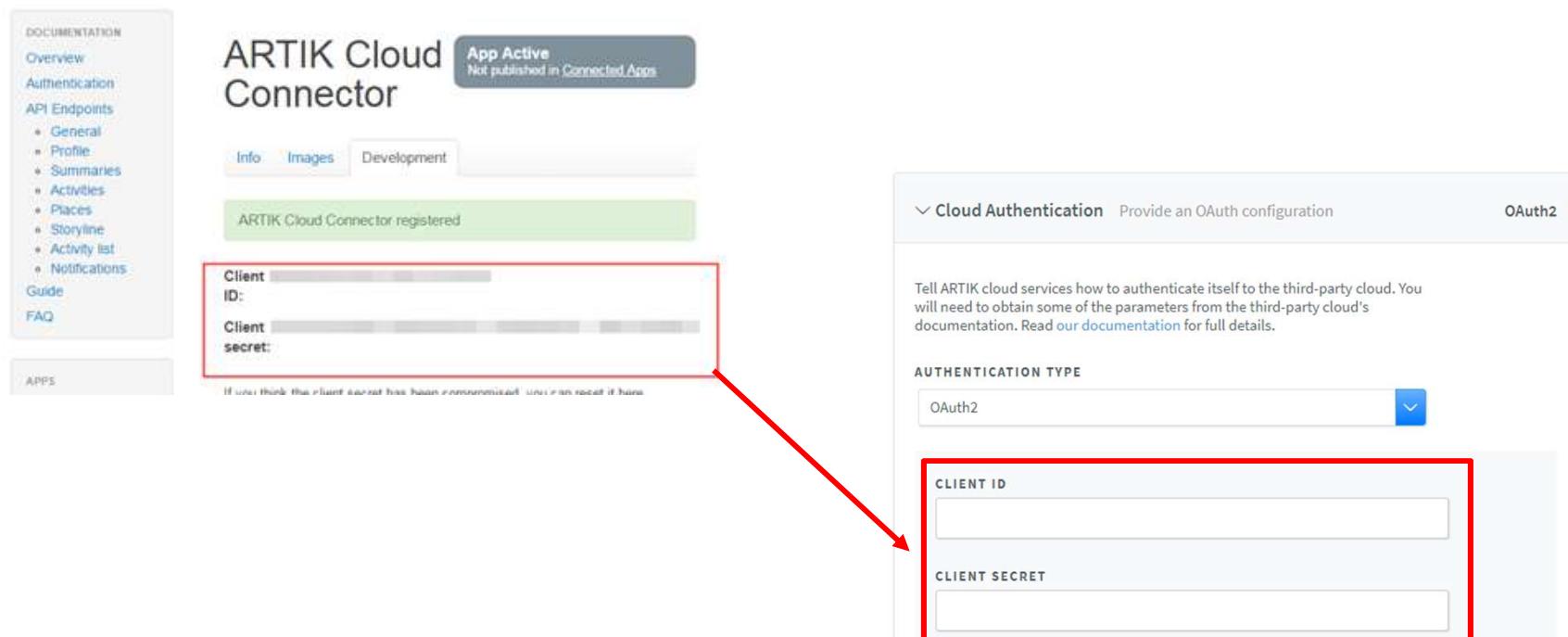
- Now, return to the ARTIK Cloud Developer Dashboard, and enter 'Device Info' tab.
- Check "This Device type sends data to a cloud that ARTIK Cloud receives data from".
- Choose "OAuth2" as Authentication Type.



# Cloud Connector for the Moves Cloud

## ■ Build a Cloud Connector for the Moves

- Step3 : Configure ARTIK Cloud Authentication to Moves Cloud
  - In the Moves Developer Portal, navigate to the development tab and copy the *Client ID* and *Client Secret*.
  - Paste the Client ID and Client Secret you copied from Moves to ARTIK Cloud Authentication Configuration.



# Cloud Connector for the Moves Cloud

## ■ Build a Cloud Connector for the Moves

- Step3 : Configure ARTIK Cloud Authentication to Moves Cloud
  - Fill in the following information from the Moves Authentication doc:

|                                             |                                                                                                                         |
|---------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| Client ID                                   |                                                                                                                         |
| Client Secret                               |                                                                                                                         |
| Permission Scope (Comma Separated)          | default, activity, location                                                                                             |
| Authorization URL                           | <a href="https://api.moves-app.com/oauth/v1/auth">https://api.moves-app.com/oauth/v1/auth</a><br>orize                  |
| Authorization Grant Type                    | code                                                                                                                    |
| Access Token URL                            | POST <a href="https://api.moves-app.com/oauth/v1/access_token">https://api.moves-app.com/oauth/v1/<br/>access_token</a> |
| Change Credentials Parameters - external_id | user_id                                                                                                                 |

- Copy the *Redirect URL* and *Notification URL* displayed in Authentication.

The screenshot shows the 'REDIRECT URL' and 'NOTIFICATION URL' fields highlighted with a red box. Below them is a note: 'Provide alternative names for the parameters used when passing credentials via the URL.' To the right, there is a 'CHANGE CREDENTIALS PARAMETERS' section with a 'external\_id' field highlighted with a blue box. A blue arrow points from the 'external\_id' field to the 'external\_id' parameter in the 'ACCESS TOKEN URL' section.

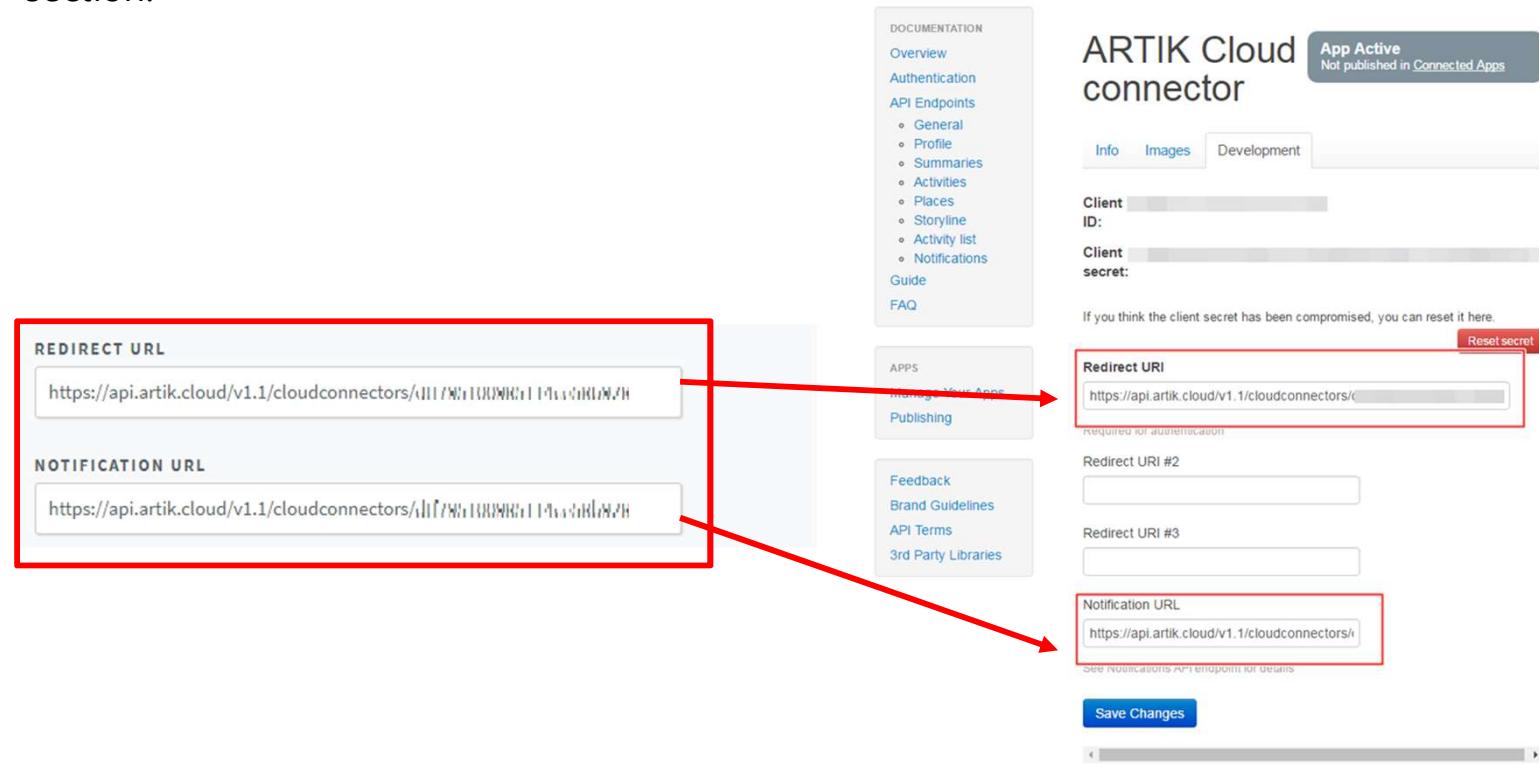
|                                                                                         |                                                                                                                                                           |
|-----------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| REDIRECT URL                                                                            | <a href="https://api.artik.cloud/v1.1/cloudconnectors/111111111111111111111111">https://api.artik.cloud/v1.1/cloudconnectors/111111111111111111111111</a> |
| NOTIFICATION URL                                                                        | <a href="https://api.artik.cloud/v1.1/cloudconnectors/111111111111111111111111">https://api.artik.cloud/v1.1/cloudconnectors/111111111111111111111111</a> |
| Provide alternative names for the parameters used when passing credentials via the URL. |                                                                                                                                                           |
| CHANGE CREDENTIALS PARAMETERS                                                           | external_id                                                                                                                                               |
| ACCESS TOKEN URL                                                                        | access_token                                                                                                                                              |
|                                                                                         | scope                                                                                                                                                     |
|                                                                                         | token_type                                                                                                                                                |
|                                                                                         | refresh_token                                                                                                                                             |
|                                                                                         | expires_in                                                                                                                                                |
|                                                                                         | external_id                                                                                                                                               |

# Cloud Connector for the Moves Cloud

## ■ Build a Cloud Connector for the Moves

### ▪ Step4 : Complete Moves setup.

- Go back to the Moves Developer Portal and look at the “Manage Your Apps” page.
- Navigate to the “Development” tab for your “ARTIK Cloud Connector” app.
- Fill in the *Redirect URL* and *Notification URL* that you copied from ARTIK Cloud in the previous section.



# Cloud Connector for the Moves Cloud

## ■ Build a Cloud Connector for the Moves

- Step5 : Upload Cloud Connector code to ARTIK Cloud
  - Switch to the Connector Code tab and copy-and-paste the following Groovy code.

```
// Sample CloudConnector, that can be used as a bootstrap to write a new
CloudConnector.
// Lot of code is commented, and everything is optional.
// The class can be named as you want no additional import allowed
// See the javadoc/scaladoc of
cloud.artik.cloudconnector.api_v1.CloudConnectorpackage com.sample

import org.scalactic./*
import org.joda.time.format.DateTimeFormat
import org.joda.time./*
import groovy.transform.CompileStatic
import groovy.transform.ToString
import groovy.json.JsonSlurper
import groovy.json.JsonOutput
import cloud.artik.cloudconnector.api_v1./*
import static java.net.HttpURLConnection./*

//@CompileStatic
class MyCloudConnector extends CloudConnector {
 static final mdateFormat = DateTimeFormat.forPattern("yyyy-MM-
dd").withZoneUTC()
 static final queryParams = ["timeZone": "UTC"]
 static final receivedDateFormat =
 DateTimeFormat.forPattern("yyyyMMdd'T'HHmmssZ").withZoneUTC().withOffsetParsed
()
 static final requestDateFormat =
 DateTimeFormat.forPattern("yyyyMMdd").withZoneUTC()
 static final reasonToFetchSummaryData = ["DataUpload"]
 JsonSlurper slurper = new JsonSlurper()

 def summaryEndpoint(String date) {
 "https://api.moves-app.com/api/1.1/user/summary/daily/" + date
 }
}
```

```
@Override
def Or<NotificationResponse, Failure> onNotification(Context ctx, RequestDef req)
{
 def json = slurper.parseText(req.content)
 def extId = json.userId.toString()
 def storyLineFiltered =
 json.storylineUpdates.findAll{ reasonToFetchSummaryData.contains(it.reason) }
 def datesFromStoryLines = storyLineFiltered.collect { e ->
 //We try to recover a valid Date from the storyLine event, and use it to fetch
 summary data.
 String dtStr = (
 (e.endTime)?e.endTime:
 (e.startTime)?e.startTime:
 (e.lastSegmentStartTime)?e.lastSegmentStartTime:
 null
)
 DateTime dt = (dtStr != null) ?DateTime.parse(dtStr, receivedDateFormat):
 DateTime.now()
 requestDateFormat.print(dt)
 }.unique()
 def requestsToDo = datesFromStoryLines.collect{ dateStr ->
 new RequestDef(summaryEndpoint(dateStr)).withQueryParams(queryParams)
 }
 new Good(new NotificationResponse([new ThirdPartyNotification(new
 ByExternalId(extId), requestsToDo)]))
}

@Override
def Or<RequestDef, Failure> fetch(Context ctx, RequestDef req, DeviceInfo info) {
 new Good(req.addHeaders(["Authorization": "Bearer " + info.credentials.token]))
}

def isSameDay(DateTime d1, DateTime d2) {
 (d1.year== d2.year) && (d1.dayOfYear == d2.dayOfYear)
}
```

# Cloud Connector for the Moves Cloud

## ■ Build a Cloud Connector for the Moves

- Step5 : Upload Cloud Connector code to ARTIK Cloud
  - Switch to the Connector Code tab and copy-and-paste the following Groovy code. (Cont.)

```
def isToday(DateTime d) {
 isSameDay(d, DateTime.now())
}

def getTimestampOfTheEndOfDay(DateTime date) {
 date.minusMillis((date.millisOfDay().get() + 1)).plusDays(1)
}

/**
 * Since we recover the summary data of the day, we want to have a meaningful
timestamp from source:
 * If the date is not today : the day is finished.
 * We set the source timestamp to the last second of this past day.
 * If the date is today : the day is not finished, data can continue to evolve for the
day,
 * we set the timestamp to now()
 */

def getTimestampFromDate(DateTime date, DateTimeZone dtz =
DateTimeZone.UTC) {
 def now = new DateTime(dtz).toDateTime(dtz)
 def returnedDate = isSameDay(date, now)? now :
getTimestampOfTheEndOfDay(date)
 returnedDate.getMillis()
}

def extractSummaryNotification(jsonNode, long ts) {
 if (jsonNode.summary) {
 jsonNode.summary.collect {js -> new Event(ts, "{W\"summaryW\":"+ +
JsonOutput.toJson(js) + "}")]
 } else {
 []
 }
}
```

```
def extractCaloriesIdle(jsonNode, long ts) {
 if (jsonNode.caloriesIdle) {
 [new Event(ts, "{W\"caloriesIdleW\":"+ +
JsonOutput.toJson(jsonNode.caloriesIdle) + "}")]
 } else {
 []
 }
}

@Override
def Or<List<Event>, Failure> onFetchResponse(Context ctx, RequestDef req,
DeviceInfo info, Response res) {
 switch(res.status) {
 case HTTP_OK:
 def content = res.content.trim()
 if (content == "") {
 ctx.debug("ignore response valid respond: '${res.content}'")
 return new Good(Empty.list())
 } else if (res.contentType.startsWith("application/json")) {
 def json = slurper.parseText(content)
 def events = json.collectMany { jsData ->
 def ts = (jsData.date)?
getTimestampFromDate(DateTime.parse(jsData.date, requestDateFormat)): ctx.now()
 extractSummaryNotification(jsData, ts) + extractCaloriesIdle(jsData, ts)
 }
 return new Good(events)
 }
 return new Bad(new Failure("unsupported response ${res} ...
${res.contentType} .. ${res.contentType.startsWith("application/json")}")
 default:
 return new Bad(new Failure("http status : ${res.status} is not OK
(${HTTP_OK}))")
 }
}
```

# Cloud Connector for the Moves Cloud

## ■ Build a Cloud Connector for the Moves

- Step5 : Upload Cloud Connector code to ARTIK Cloud
  - Then, click "submit for approval".
  - Once you submit it, Cloud Connector code is in approval state.
  - It maybe takes a few hours to receive approval e-mail.

Cloud Connector Code

BROWSE & UPLOAD CODE (.GROOVY)

Paste or upload Groovy code that tells ARTIK cloud services how to interact with the third-party APIs in order to get data from the cloud. See our [documentation](#) for details.

```
View sample code. view in fullscreen
1 // Sample CloudConnector, that can be used as a bootstrap to write a new CloudConnector.
2 // Lot of code is commented, and everything is optional.
3 // The class can be named as you want no additional import allowed
4 // See the javadoc/scaladoc of cloud.artik.cloudconnector.api_v1.CloudConnectorpackage com.samp
5
6 import org.scalactic.-
7 import org.joda.time.format.DateTimeFormat
8 import org.joda.time.-
9 import groovy.transform.CompileStatic
10 import groovy.transform.ToString
11 import groovy.json.JsonSlurper
12 import groovy.json.JsonOutput
13 import static cloud.artik.cloudconnector.api_v1.-
14 import static java.net.HttpURLConnection.-
15
16
```

SUBMIT FOR APPROVAL    SAVE DRAFT

# Cloud Connector for the Moves Cloud

## ■ Build a Cloud Connector for the Moves

### ▪ Step 6 : Create device type Manifest

- Go to 'Manifest' tab, you can create the Manifest for the Cloud Connector device, by uploading JSON file.
- Here you can upload manifest.json, which has the following content.

```
{
 "fields": [
 {
 "children": [
 {
 "name": "activity",
 "type": "CUSTOM",
 "valueClass": "STRING"
 },
 {
 "name": "group",
 "type": "CUSTOM",
 "valueClass": "STRING"
 },
 {
 "name": "duration",
 "type": "CUSTOM",
 "valueClass": "LONG"
 },
 {
 "name": "calories",
 "type": "CUSTOM",
 "valueClass": "LONG"
 }
]
 }
]
}
```

```
{
 "name": "steps",
 "type": "CUSTOM",
 "valueClass": "LONG"
},
{
 "name": "distance",
 "type": "CUSTOM",
 "valueClass": "LONG"
},
{
 "name": "summary"
},
{
 "name": "caloriesIdle",
 "type": "CUSTOM",
 "valueClass": "Double",
 "unit": "StandardUnits.KILO_CALORIE",
 "isCollection": false,
 "description": "daily idle burn in kcal. Available if user has at least once enabled calories",
 "tags": []
},
{
 "messageFormat": "json",
 "actions": []
}
```

# Cloud Connector for the Moves Cloud

## ■ Build a Cloud Connector for the Moves

- Step 6 : Create device type Manifest (Cont.)

The screenshot shows the Moves Cloud interface for managing device type manifests. At the top, there's a blue header bar with the title 'Set Up / Manifests'. Below it, a navigation bar includes a 'cloudconnector' icon, the text 'Set Up / Manifests', a '+ NEW VERSION' button, and a dropdown menu. The main content area displays a table titled 'Fields' with the following data:

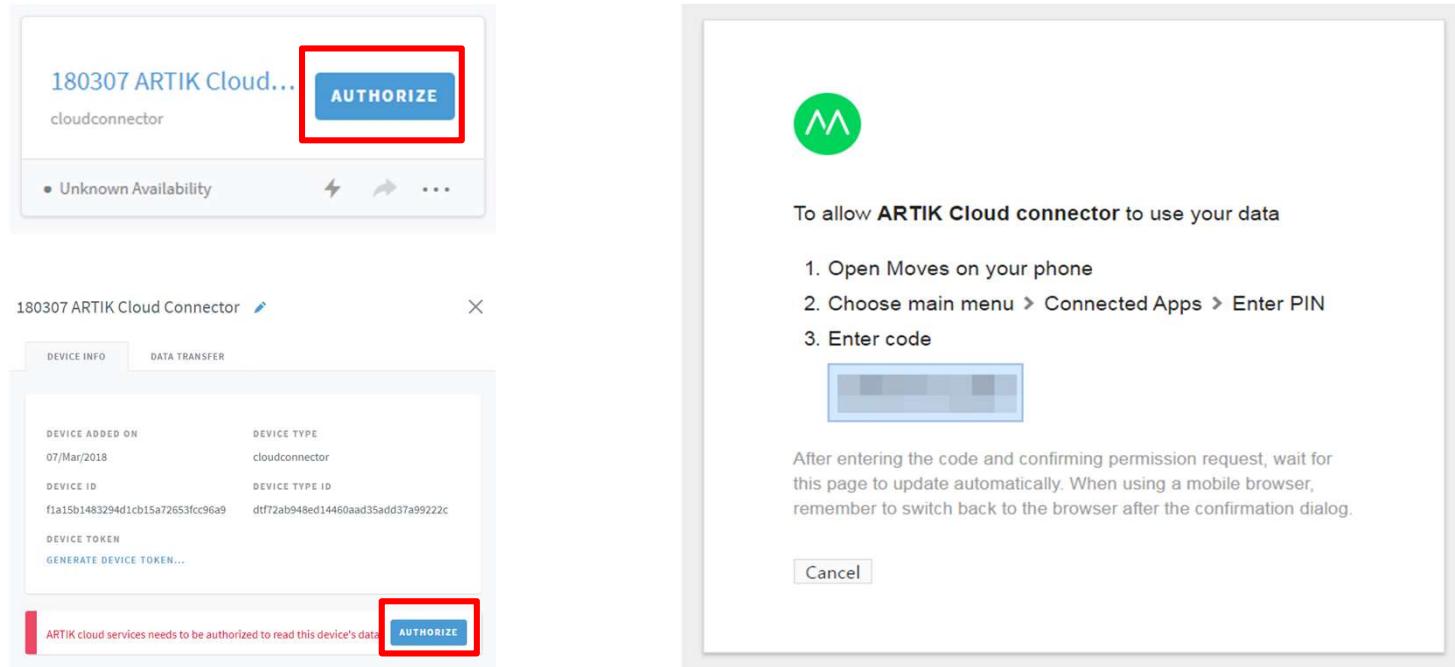
| Field Name   | Type   | Description |
|--------------|--------|-------------|
| caloriesidle | Double | J*4.184     |
| summary      | Group  |             |
| activity     | String | None        |
| calories     | Long   | None        |
| distance     | Long   | None        |
| duration     | Long   | None        |
| group        | String | None        |
| steps        | Long   | None        |

Below the table, there are buttons for 'DOWNLOAD MANIFEST' and 'VIEW SAMPLE MESSAGE' with the timestamp '03/07/2018 1:23'.

# Cloud Connector for the Moves Cloud

## ■ Build a Cloud Connector for the Moves

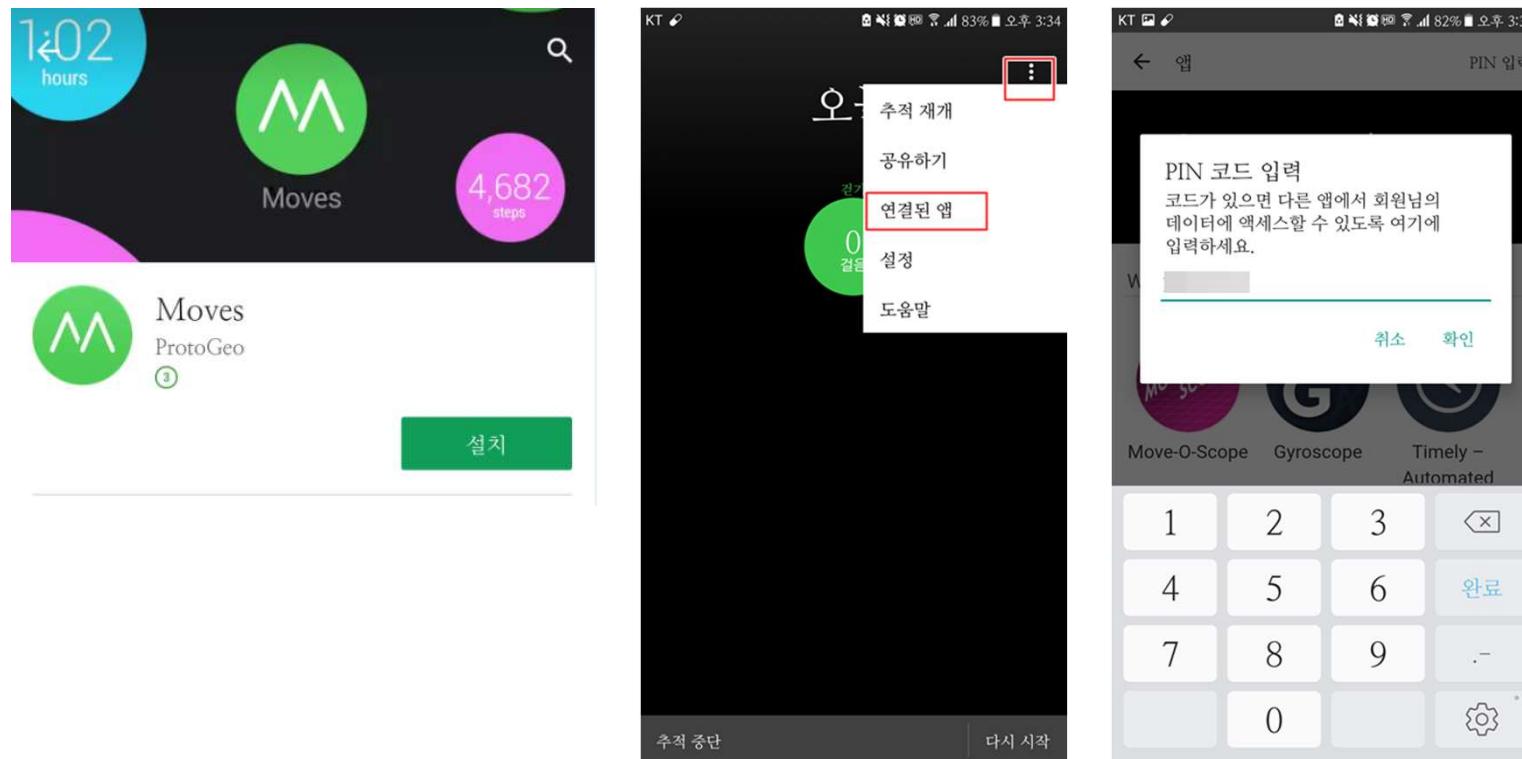
- Step 7 : Connect a device in My ARTIK Cloud.
  - Once Cloud Connector is approved, go to My ARTIK Cloud and connect a device using your newly created Cloud Connector device type.
  - And then, authorize the device to grant ARTIK Cloud access to your Moves data.



# Cloud Connector for the Moves Cloud

## ■ Build a Cloud Connector for the Moves

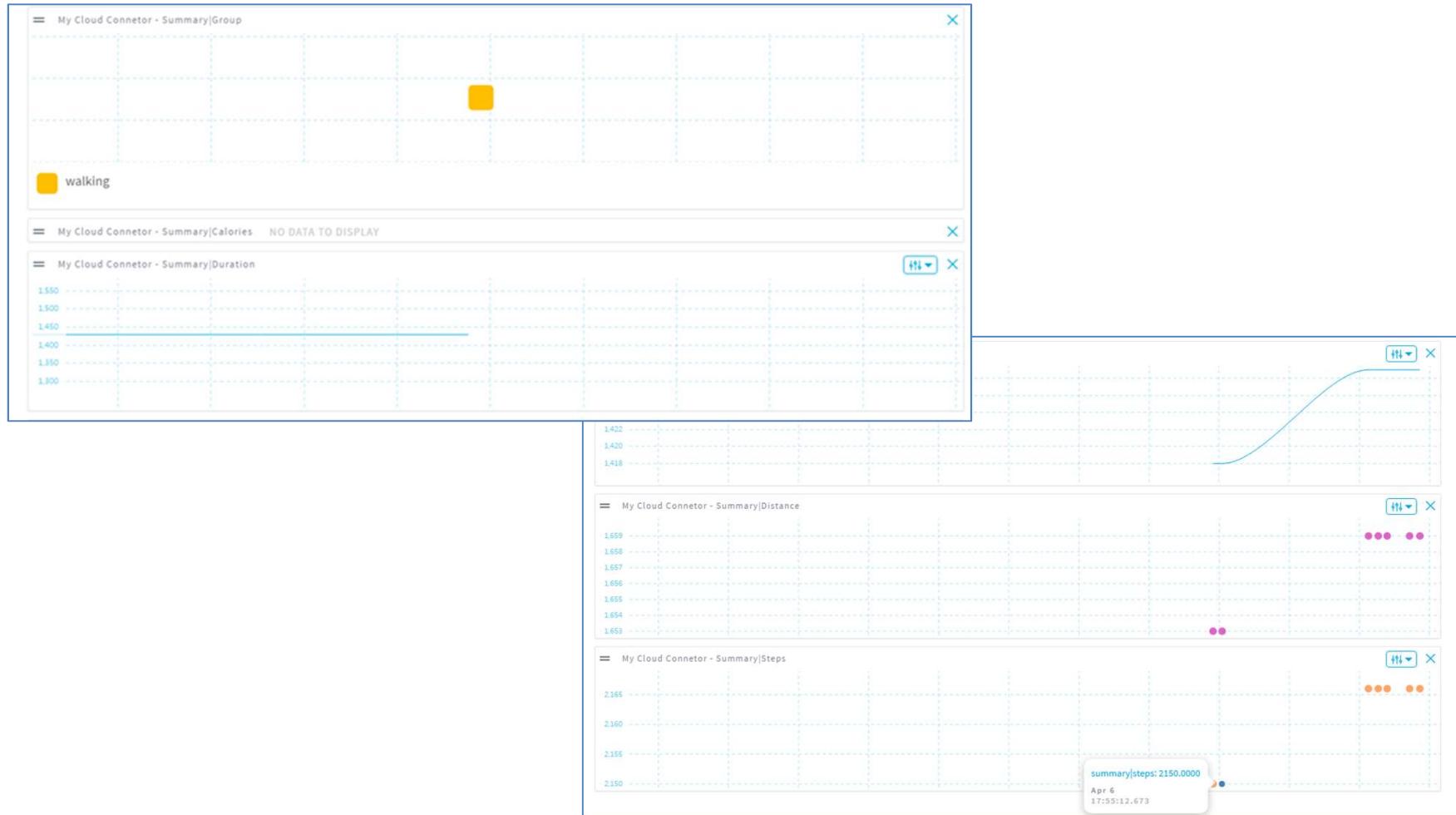
- Step 8 : Install Moves app on your smart phone.
  - Choose main menu – Connected Apps
  - Enter Pin code



# Cloud Connector for the Moves Cloud

## ■ Build a Cloud Connector for the Moves

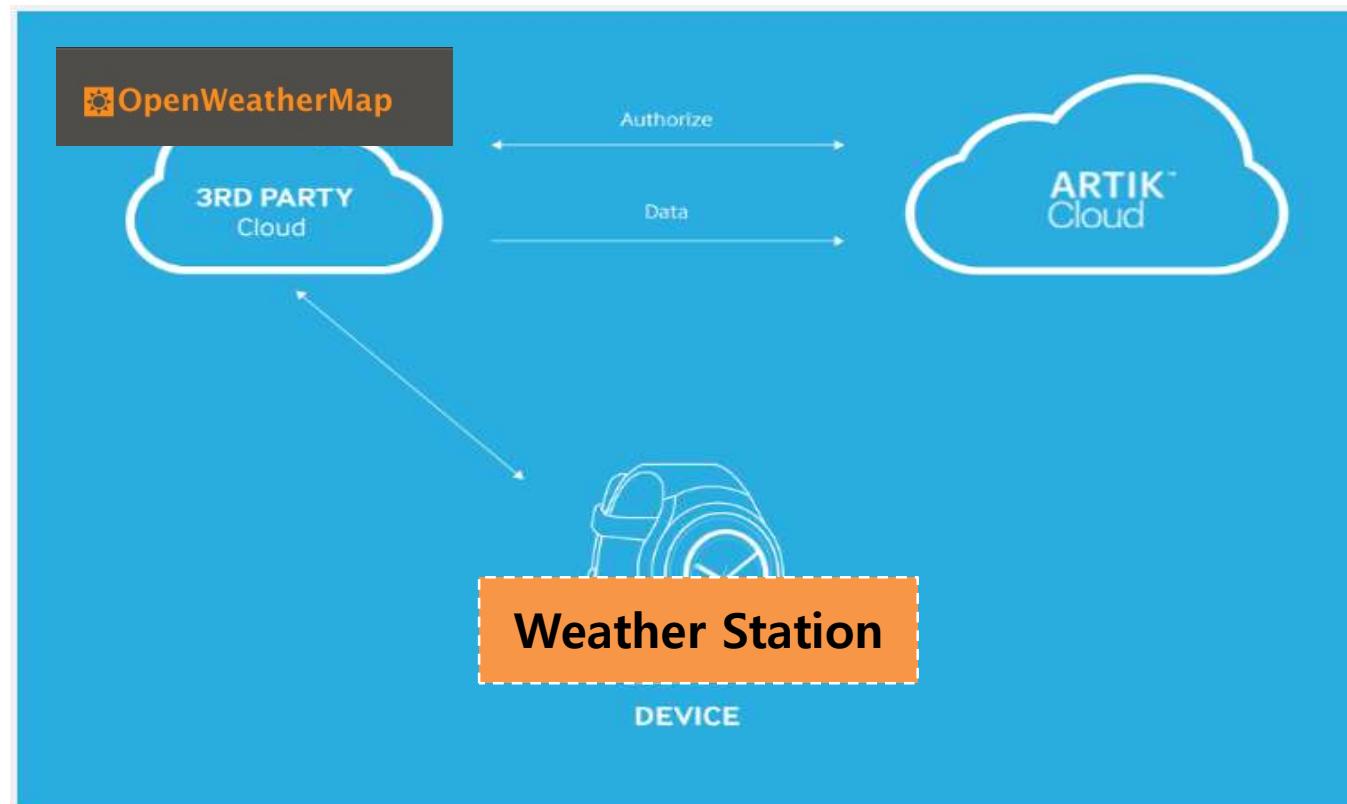
- Step 9 : Check data in My ARTIK Cloud Charts.



# Open Weather Map Cloud

## ■ Introduction to Open Weather Map Cloud

- Open Weather Map is a cloud that provides weather data across the country.
- Unlike the Moves cloud, Open Weather Map does not use authentication, but it requires an App ID that is issued after logging in as a parameter when calling the API.



# Open Weather Map Cloud

## ■ Introduction to Open Weather Map Cloud

- Before you build Cloud Connector, let's call the Open Weather Map API and get a rough idea of how to collect data from the cloud.
- Go to Open Weather Map site.
  - <https://openweathermap.org/>
- On the Open Weather Map site, click API on the top menu to view the available API Documentation.

The screenshot shows the OpenWeatherMap website with a dark header bar. The header includes the logo, navigation links for Weather, Maps, API, Price, Partners, Stations, Widgets, News, and About, and two buttons: 'API doc' and 'Subscribe'.

**Current weather data**

- Access current weather data for any location including over 200,000 cities
- Current weather is frequently updated based on global models and data from more than 40,000 weather stations
- Data is available in JSON, XML, or HTML format
- Available for Free and all other paid

**5 day / 3 hour forecast**

- 5 day forecast is available at any location or city
- 5 day forecast includes weather data every 3 hours
- Forecast is available in JSON and XML
- Available for Free and all other paid accounts

**16 day / daily forecast**

- 16 day forecast is available at any location or city
- 16 day forecast includes daily weather
- Forecast is available in JSON and XML
- Available for Developer, Professional and Enterprise accounts

Our weather API is simple, clear and free. We also offer higher levels of support, please see our [paid plan options](#). To access the API you need to sign up for an [API key](#) if you are on a free or paid plan.

# Open Weather Map Cloud

## ■ Introduction to Open Weather Map Cloud

- Before you build Cloud Connector, let's call the Open Weather Map API and get a rough idea of how to collect data from the cloud.
- Go to Open Weather Map site.
  - <https://openweathermap.org/>
- On the Open Weather Map site, click API on the top menu to view the available API Documentation.

The screenshot shows the OpenWeatherMap website with a dark header bar. The header includes the logo, navigation links for Weather, Maps, API, Price, Partners, Stations, Widgets, News, and About, and two buttons: 'API doc' and 'Subscribe'.

**Current weather data**

- Access current weather data for any location including over 200,000 cities
- Current weather is frequently updated based on global models and data from more than 40,000 weather stations
- Data is available in JSON, XML, or HTML format
- Available for Free and all other paid

**5 day / 3 hour forecast**

- 5 day forecast is available at any location or city
- 5 day forecast includes weather data every 3 hours
- Forecast is available in JSON and XML
- Available for Free and all other paid accounts

**16 day / daily forecast**

- 16 day forecast is available at any location or city
- 16 day forecast includes daily weather
- Forecast is available in JSON and XML
- Available for Developer, Professional and Enterprise accounts

Our weather API is simple, clear and free. We also offer higher levels of support, please see our [paid plan options](#). To access the API you need to sign up for an [API key](#) if you are on a free or paid plan.

# Open Weather Map Cloud

## ■ Introduction to Open Weather Map Cloud

- To get an API Key from the cloud, go to the Price tab on the top menu and select the account you want to use and click 'Get API Key and Start' or 'Subscribe'.

The screenshot shows the OpenWeatherMap Cloud interface. At the top, there's a navigation bar with 'Support Center', a search bar ('Weather in your city'), and links for 'Sign In' and 'Sign Up'. Temperature units are set to °C and °F. The main menu includes 'Weather', 'Maps', 'API', 'Price' (which is highlighted with a red box), 'Partners', 'Stations', 'Widgets', 'News', and 'About'. Below the menu, the 'Price' section is displayed, featuring a heading 'Current weather and forecasts collection' and a table comparing 'Free', 'Startup', and 'Enterprise' plans. The 'Enterprise' plan is currently selected. A call-to-action button 'Get API key and Start' is highlighted with a red box. To the right, the 'API keys' section is shown, also with a red box around the 'API keys' tab in the navigation bar. It contains a note about activation times and a form for generating a new API key, with fields for 'Key' and 'Name'.

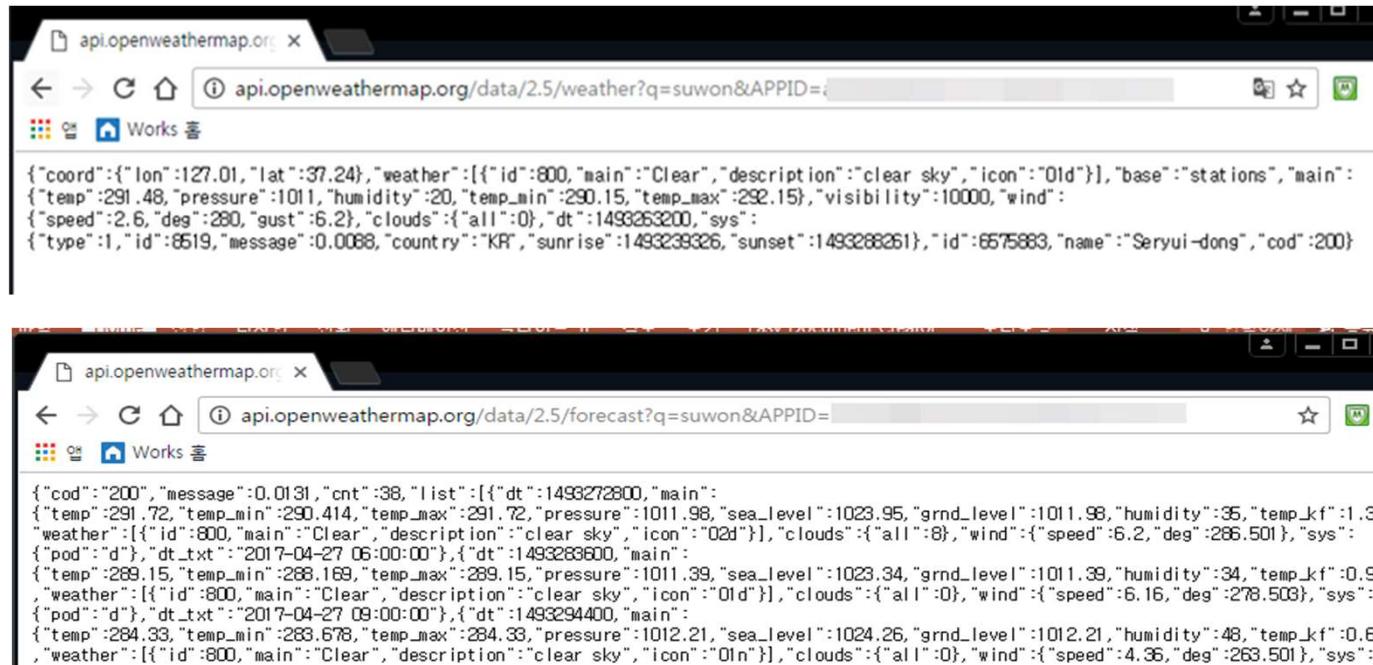
|                                 | Free                                  | Startup                   | Enterprise |
|---------------------------------|---------------------------------------|---------------------------|------------|
| Price                           | Free                                  | 40 USD / month            |            |
| Subscribe                       | <a href="#">Get API key and Start</a> | <a href="#">Subscribe</a> |            |
| Calls per minute (no more than) | 60                                    | 600                       | 3,000      |
| Current weather API             | ✓                                     | ✓                         | ✓          |
| 5 days/3 hour forecast API      | ✓                                     | ✓                         | ✓          |
| 16 days/daily forecast API      | -                                     | -                         | ✓          |
| Weather maps API                | ✓                                     | ✓                         | ✓          |
| Bulk download                   | -                                     | -                         | -          |
| UV index (beta)                 | ✓                                     | ✓                         | ✓          |
| Air pollution (beta)            | ✓                                     | ✓                         | ✓          |
| Weather alerts (beta)           | ✓                                     | ✓                         | ✓          |

# Open Weather Map Cloud

## ■ Introduction to Open Weather Map Cloud

### ▪ Example of API Calls

- Call current weather data for one location
  - [api.openweathermap.org/data/2.5/weather?q={city name}&APPID={API Key}](http://api.openweathermap.org/data/2.5/weather?q={city name}&APPID={API Key})
- Call 5 day / 3 hour forecast data
  - [api.openweathermap.org/data/2.5/forecast?q={city name}&APPID={API Key}](http://api.openweathermap.org/data/2.5/forecast?q={city name}&APPID={API Key})



The image contains two screenshots of a web browser window. Both screenshots show the URL [api.openweathermap.org/data/2.5/weather?q=suwon&APPID=...](http://api.openweathermap.org/data/2.5/weather?q=suwon&APPID=...) in the address bar.

**Screenshot 1 (Top): Current Weather Data**

```
{"coord": {"lon": 127.01, "lat": 37.24}, "weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01d"}], "base": "stations", "main": {"temp": 291.48, "pressure": 1011, "humidity": 20, "temp_min": 290.15, "temp_max": 292.15}, "visibility": 10000, "wind": {"speed": 2.6, "deg": 280, "gust": 6.2}, "clouds": {"all": 0}, "dt": 1493263200, "sys": {"type": 1, "id": 8519, "message": 0.0088, "country": "KR", "sunrise": 1493239326, "sunset": 1493288261}, "id": 6675883, "name": "Seryui-dong", "cod": 200}
```

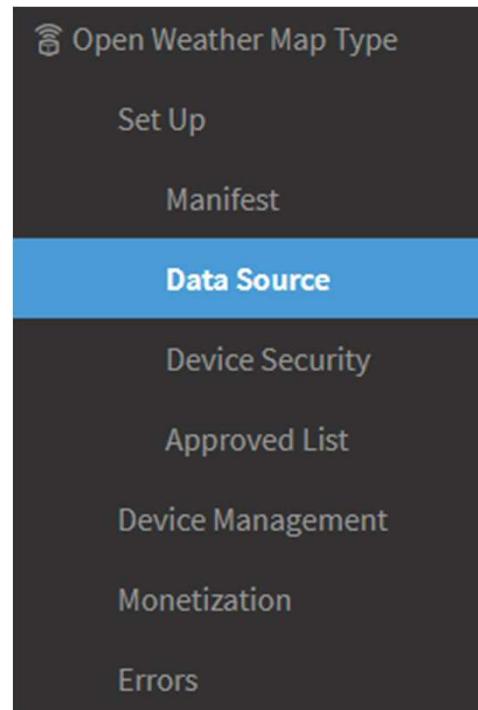
**Screenshot 2 (Bottom): 5 Day Forecast Data**

```
{"cod": "200", "message": 0.0131, "cnt": 38, "list": [{"dt": 1493272800, "main": {"temp": 291.72, "temp_min": 290.414, "temp_max": 291.72, "pressure": 1011.98, "sea_level": 1023.95, "grnd_level": 1011.98}, "humidity": 35, "temp_kf": 1.3}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "02d"}], "clouds": {"all": 0}, "wind": {"speed": 6.2, "deg": 286.501}, "sys": {"pod": "d"}, "dt_txt": "2017-04-27 06:00:00"}, {"dt": 1493280600, "main": {"temp": 289.15, "temp_min": 288.168, "temp_max": 289.15, "pressure": 1011.99, "sea_level": 1023.34, "grnd_level": 1011.39}, "humidity": 34, "temp_kf": 0.98}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01d"}], "clouds": {"all": 0}, "wind": {"speed": 6.16, "deg": 278.503}, "sys": {"pod": "d"}, "dt_txt": "2017-04-27 08:00:00"}, {"dt": 1493294400, "main": {"temp": 284.33, "temp_min": 283.678, "temp_max": 284.33, "pressure": 1012.21, "sea_level": 1024.26, "grnd_level": 1012.21}, "humidity": 48, "temp_kf": 0.65}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-04-27 12:00:00"}, {"dt": 1493302200, "main": {"temp": 283.5, "temp_min": 282.848, "temp_max": 283.5, "pressure": 1013.04, "sea_level": 1025.01, "grnd_level": 1013.04}, "humidity": 52, "temp_kf": 0.56}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-04-27 18:00:00"}, {"dt": 1493310000, "main": {"temp": 282.67, "temp_min": 281.918, "temp_max": 282.67, "pressure": 1013.87, "sea_level": 1025.78, "grnd_level": 1013.87}, "humidity": 56, "temp_kf": 0.47}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-04-28 00:00:00"}, {"dt": 1493317800, "main": {"temp": 281.84, "temp_min": 281.088, "temp_max": 281.84, "pressure": 1014.7, "sea_level": 1026.55, "grnd_level": 1014.7}, "humidity": 60, "temp_kf": 0.38}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-04-28 06:00:00"}, {"dt": 1493325600, "main": {"temp": 281.01, "temp_min": 280.258, "temp_max": 281.01, "pressure": 1015.53, "sea_level": 1027.32, "grnd_level": 1015.53}, "humidity": 64, "temp_kf": 0.29}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-04-28 12:00:00"}, {"dt": 1493333400, "main": {"temp": 280.18, "temp_min": 279.428, "temp_max": 280.18, "pressure": 1016.36, "sea_level": 1028.09, "grnd_level": 1016.36}, "humidity": 68, "temp_kf": 0.2}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-04-28 18:00:00"}, {"dt": 1493341200, "main": {"temp": 279.35, "temp_min": 278.6, "temp_max": 279.35, "pressure": 1017.19, "sea_level": 1028.86, "grnd_level": 1017.19}, "humidity": 72, "temp_kf": 0.11}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-04-29 00:00:00"}, {"dt": 1493348000, "main": {"temp": 278.52, "temp_min": 277.768, "temp_max": 278.52, "pressure": 1018.02, "sea_level": 1029.63, "grnd_level": 1018.02}, "humidity": 76, "temp_kf": 0.02}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-04-29 06:00:00"}, {"dt": 1493355800, "main": {"temp": 277.69, "temp_min": 276.938, "temp_max": 277.69, "pressure": 1018.85, "sea_level": 1030.4, "grnd_level": 1018.85}, "humidity": 80, "temp_kf": -0.89}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-04-29 12:00:00"}, {"dt": 1493363600, "main": {"temp": 276.86, "temp_min": 276.1, "temp_max": 276.86, "pressure": 1019.68, "sea_level": 1031.17, "grnd_level": 1019.68}, "humidity": 84, "temp_kf": -0.78}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-04-29 18:00:00"}, {"dt": 1493371400, "main": {"temp": 276.03, "temp_min": 275.278, "temp_max": 276.03, "pressure": 1020.51, "sea_level": 1031.94, "grnd_level": 1020.51}, "humidity": 88, "temp_kf": -0.67}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-04-30 00:00:00"}, {"dt": 1493379200, "main": {"temp": 275.2, "temp_min": 274.448, "temp_max": 275.2, "pressure": 1021.34, "sea_level": 1032.71, "grnd_level": 1021.34}, "humidity": 92, "temp_kf": -0.56}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-04-30 06:00:00"}, {"dt": 1493387000, "main": {"temp": 274.37, "temp_min": 273.618, "temp_max": 274.37, "pressure": 1022.17, "sea_level": 1033.48, "grnd_level": 1022.17}, "humidity": 96, "temp_kf": -0.45}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-04-30 12:00:00"}, {"dt": 1493394800, "main": {"temp": 273.54, "temp_min": 272.788, "temp_max": 273.54, "pressure": 1023.0, "sea_level": 1034.25, "grnd_level": 1023.0}, "humidity": 100, "temp_kf": -0.34}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-04-30 18:00:00"}, {"dt": 1493402600, "main": {"temp": 272.71, "temp_min": 271.958, "temp_max": 272.71, "pressure": 1023.83, "sea_level": 1035.02, "grnd_level": 1023.83}, "humidity": 104, "temp_kf": -0.23}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-05-01 00:00:00"}, {"dt": 1493410400, "main": {"temp": 271.88, "temp_min": 271.128, "temp_max": 271.88, "pressure": 1024.66, "sea_level": 1035.79, "grnd_level": 1024.66}, "humidity": 108, "temp_kf": -0.12}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-05-01 06:00:00"}, {"dt": 1493418200, "main": {"temp": 271.05, "temp_min": 270.3, "temp_max": 271.05, "pressure": 1025.49, "sea_level": 1036.56, "grnd_level": 1025.49}, "humidity": 112, "temp_kf": 0.01}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-05-01 12:00:00"}, {"dt": 1493426000, "main": {"temp": 270.22, "temp_min": 269.468, "temp_max": 270.22, "pressure": 1026.32, "sea_level": 1037.33, "grnd_level": 1026.32}, "humidity": 116, "temp_kf": 0.9}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-05-01 18:00:00"}, {"dt": 1493433800, "main": {"temp": 269.39, "temp_min": 268.638, "temp_max": 269.39, "pressure": 1027.15, "sea_level": 1038.1, "grnd_level": 1027.15}, "humidity": 120, "temp_kf": 0.79}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-05-02 00:00:00"}, {"dt": 1493441600, "main": {"temp": 268.56, "temp_min": 267.8, "temp_max": 268.56, "pressure": 1027.98, "sea_level": 1038.87, "grnd_level": 1027.98}, "humidity": 124, "temp_kf": 0.68}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-05-02 06:00:00"}, {"dt": 1493449400, "main": {"temp": 267.73, "temp_min": 267.0, "temp_max": 267.73, "pressure": 1028.81, "sea_level": 1039.64, "grnd_level": 1028.81}, "humidity": 128, "temp_kf": 0.57}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-05-02 12:00:00"}, {"dt": 1493457200, "main": {"temp": 266.9, "temp_min": 266.17, "temp_max": 266.9, "pressure": 1029.64, "sea_level": 1040.41, "grnd_level": 1029.64}, "humidity": 132, "temp_kf": 0.46}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-05-02 18:00:00"}, {"dt": 1493465000, "main": {"temp": 266.07, "temp_min": 265.34, "temp_max": 266.07, "pressure": 1030.47, "sea_level": 1041.18, "grnd_level": 1030.47}, "humidity": 136, "temp_kf": 0.35}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-05-03 00:00:00"}, {"dt": 1493472800, "main": {"temp": 265.24, "temp_min": 264.51, "temp_max": 265.24, "pressure": 1031.3, "sea_level": 1041.95, "grnd_level": 1031.3}, "humidity": 140, "temp_kf": 0.24}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-05-03 06:00:00"}, {"dt": 1493480600, "main": {"temp": 264.41, "temp_min": 263.68, "temp_max": 264.41, "pressure": 1032.13, "sea_level": 1042.72, "grnd_level": 1032.13}, "humidity": 144, "temp_kf": 0.13}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-05-03 12:00:00"}, {"dt": 1493488400, "main": {"temp": 263.58, "temp_min": 262.85, "temp_max": 263.58, "pressure": 1032.96, "sea_level": 1043.49, "grnd_level": 1032.96}, "humidity": 148, "temp_kf": 0.02}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-05-03 18:00:00"}, {"dt": 1493496200, "main": {"temp": 262.75, "temp_min": 262.02, "temp_max": 262.75, "pressure": 1033.79, "sea_level": 1044.26, "grnd_level": 1033.79}, "humidity": 152, "temp_kf": -0.91}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-05-04 00:00:00"}, {"dt": 1493504000, "main": {"temp": 261.92, "temp_min": 261.19, "temp_max": 261.92, "pressure": 1034.62, "sea_level": 1045.03, "grnd_level": 1034.62}, "humidity": 156, "temp_kf": -0.8}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-05-04 06:00:00"}, {"dt": 1493511800, "main": {"temp": 261.09, "temp_min": 260.36, "temp_max": 261.09, "pressure": 1035.45, "sea_level": 1045.8, "grnd_level": 1035.45}, "humidity": 160, "temp_kf": -0.69}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-05-04 12:00:00"}, {"dt": 1493519600, "main": {"temp": 260.26, "temp_min": 259.53, "temp_max": 260.26, "pressure": 1036.28, "sea_level": 1046.57, "grnd_level": 1036.28}, "humidity": 164, "temp_kf": -0.58}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-05-04 18:00:00"}, {"dt": 1493527400, "main": {"temp": 259.43, "temp_min": 258.7, "temp_max": 259.43, "pressure": 1037.11, "sea_level": 1047.34, "grnd_level": 1037.11}, "humidity": 168, "temp_kf": -0.47}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-05-05 00:00:00"}, {"dt": 1493535200, "main": {"temp": 258.6, "temp_min": 257.87, "temp_max": 258.6, "pressure": 1037.94, "sea_level": 1048.11, "grnd_level": 1037.94}, "humidity": 172, "temp_kf": -0.36}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-05-05 06:00:00"}, {"dt": 1493543000, "main": {"temp": 257.77, "temp_min": 257.04, "temp_max": 257.77, "pressure": 1038.77, "sea_level": 1048.88, "grnd_level": 1038.77}, "humidity": 176, "temp_kf": -0.25}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-05-05 12:00:00"}, {"dt": 1493550800, "main": {"temp": 256.94, "temp_min": 256.21, "temp_max": 256.94, "pressure": 1039.6, "sea_level": 1049.65, "grnd_level": 1039.6}, "humidity": 180, "temp_kf": -0.14}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-05-05 18:00:00"}, {"dt": 1493558600, "main": {"temp": 256.11, "temp_min": 255.38, "temp_max": 256.11, "pressure": 1040.43, "sea_level": 1050.42, "grnd_level": 1040.43}, "humidity": 184, "temp_kf": 0.03}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-05-06 00:00:00"}, {"dt": 1493566400, "main": {"temp": 255.28, "temp_min": 254.55, "temp_max": 255.28, "pressure": 1041.26, "sea_level": 1051.19, "grnd_level": 1041.26}, "humidity": 188, "temp_kf": 0.92}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-05-06 06:00:00"}, {"dt": 1493574200, "main": {"temp": 254.45, "temp_min": 253.72, "temp_max": 254.45, "pressure": 1042.09, "sea_level": 1051.96, "grnd_level": 1042.09}, "humidity": 192, "temp_kf": 0.81}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-05-06 12:00:00"}, {"dt": 1493582000, "main": {"temp": 253.62, "temp_min": 252.89, "temp_max": 253.62, "pressure": 1042.92, "sea_level": 1052.73, "grnd_level": 1042.92}, "humidity": 196, "temp_kf": 0.7}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-05-06 18:00:00"}, {"dt": 1493589800, "main": {"temp": 252.79, "temp_min": 252.06, "temp_max": 252.79, "pressure": 1043.75, "sea_level": 1053.5, "grnd_level": 1043.79}, "humidity": 200, "temp_kf": 0.59}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-05-07 00:00:00"}, {"dt": 1493597600, "main": {"temp": 251.96, "temp_min": 251.23, "temp_max": 251.96, "pressure": 1044.58, "sea_level": 1054.27, "grnd_level": 1044.58}, "humidity": 204, "temp_kf": 0.48}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-05-07 06:00:00"}, {"dt": 1493605400, "main": {"temp": 251.13, "temp_min": 250.4, "temp_max": 251.13, "pressure": 1045.41, "sea_level": 1055.04, "grnd_level": 1045.41}, "humidity": 208, "temp_kf": 0.37}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-05-07 12:00:00"}, {"dt": 1493613200, "main": {"temp": 250.3, "temp_min": 249.57, "temp_max": 250.3, "pressure": 1046.24, "sea_level": 1055.81, "grnd_level": 1046.24}, "humidity": 212, "temp_kf": 0.26}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-05-07 18:00:00"}, {"dt": 1493621000, "main": {"temp": 249.47, "temp_min": 248.74, "temp_max": 249.47, "pressure": 1047.07, "sea_level": 1056.58, "grnd_level": 1047.07}, "humidity": 216, "temp_kf": 0.15}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-05-08 00:00:00"}, {"dt": 1493628800, "main": {"temp": 248.64, "temp_min": 247.91, "temp_max": 248.64, "pressure": 1047.9, "sea_level": 1057.35, "grnd_level": 1047.9}, "humidity": 220, "temp_kf": 0.04}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-05-08 06:00:00"}, {"dt": 1493636600, "main": {"temp": 247.81, "temp_min": 247.08, "temp_max": 247.81, "pressure": 1048.73, "sea_level": 1058.12, "grnd_level": 1048.73}, "humidity": 224, "temp_kf": -0.93}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-05-08 12:00:00"}, {"dt": 1493644400, "main": {"temp": 246.98, "temp_min": 246.25, "temp_max": 246.98, "pressure": 1049.56, "sea_level": 1058.89, "grnd_level": 1049.56}, "humidity": 228, "temp_kf": -0.82}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2017-05-08 18:00:00"}, {"dt": 1493652200, "main": {"temp": 246.15, "temp_min": 245.42, "temp_max": 246.15, "pressure": 1050.39, "sea_level": 1059.66, "grnd_level": 1050.39}, "humidity": 232, "temp_kf": -0.71}, {"weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 0}, "wind": {"speed": 4.36, "deg": 263.501}, "sys": {"pod": "n"}, "dt_txt": "2
```

# Open Weather Map Cloud

## ■ Build Cloud Connector for Open Weather Map

- Step 1 : Create a new Device Type
  - Go to the ARTIK Cloud Developer site, and create a device type on the dashboard.



# Open Weather Map Cloud

## ■ Build Cloud Connector for Open Weather Map

- Step 2 : Configure ARTIK Cloud authentication
  - Go to the 'Data Source' tab and make some settings for building the Cloud Connector.
  - Choose "From another cloud" and select 'None' for the Authentication type.

Configuration      [DOWNLOAD CONFIG](#)      [UPLOAD CONFIG \(.JSON\)](#)

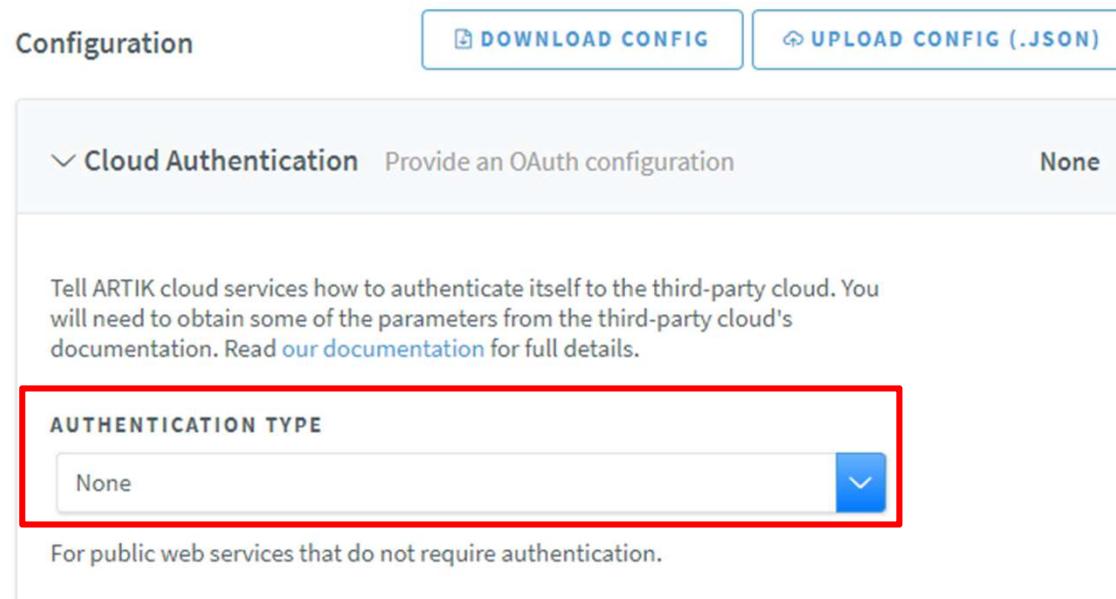
Cloud Authentication    Provide an OAuth configuration      None

Tell ARTIK cloud services how to authenticate itself to the third-party cloud. You will need to obtain some of the parameters from the third-party cloud's documentation. Read [our documentation](#) for full details.

AUTHENTICATION TYPE

None

For public web services that do not require authentication.



# Open Weather Map Cloud

## ■ Build Cloud Connector for Open Weather Map

- Step 2 : Write Cloud Connector Groovy code.
  - Go to the Github and download the samples of Cloud Connector Data source for ARTIK Cloud.
  - <https://github.com/artikcloud/artikcloud-cloudconnector-samples.git>
  - Navigate to '`artikcloud-cloudconnector-samples-master\open-weather-map\src\main\groovy\cloudconnector`'
  - Then, open the file '`MyCloudConnector.groovy`'
  - Copy the Groovy source code and upload it to the ARTIK Cloud.

The screenshot shows the ARTIK Cloud interface for uploading a Cloud Connector Groovy script. On the left, there are three files: 'cfg' (with a gear icon), 'manifest' (with a document icon), and 'MyCloudConnector' (with a document icon). The 'MyCloudConnector' file is highlighted with a red box. A red arrow points from this file to a larger red box enclosing the code editor on the right. The code editor contains the Groovy script:

```
View sample code. view in fullscreen
1 package cloudconnector
2
3 import cloud.artik.cloudconnector.api_v1.*
4 import groovy.json.JsonOutput
5 import groovy.json.JsonSlurper
6 import org.joda.time.DateTime
7 import org.joda.time.DateTimeZone
8 import org.joda.time.format.DateTimeFormat
9 import org.joda.time.format.DateTimeFormatter
10 import org.scalactic.Bad
11 import org.scalactic.Good
12 import org.scalactic.Or
13 import scala.math.BigDecimal
14
15 import java.math.MathContext
```

Below the code editor are two buttons: 'SUBMIT FOR APPROVAL' and 'SAVE DRAFT'. At the top right of the code editor is a button labeled 'BROWSE & UPLOAD CODE (.GROOVY)'. The entire code editor area is also enclosed in a red box.

# Open Weather Map Cloud

## ■ Build Cloud Connector for Open Weather Map

- Step 2 : Write Cloud Connector Groovy code. (Cont.)
  - Before saving your Cloud Connector code, add the following code lines in the class.

```
@Override
Or<NotificationResponse, Failure> onNotification(Context ctx, RequestDef req) {
 ctx.debug("onNotification: " + req)
 if (req.url.endsWith("thirdpartynotifications/postsubscription")) {
 def did = slurper.parseText(req.content())?.did
 //return new Good(new NotificationResponse([new ThirdPartyNotification(new ByDid(did), [])]))
 return new Good(new NotificationResponse([]))
 } else if (req.contentType() == CT_JSON && req.content().trim().length() > 0) {
 def did = req.headers()['notificationId']
 if (did == null) {
 ctx.debug('Bad notification (where is did in following req :) ' + req)
 return new Bad(new Failure('Impossible to recover device id from token request.'))
 }
 def content = req.content()
 def json = slurper.parseText(content)

 def dataToFetch = json.messages.collect { e ->
 new RequestDef("${ctx.parameters().endpoint}/messages/${e}")
 }
 return new Good(new NotificationResponse([new ThirdPartyNotification(new ByDid(did), dataToFetch)]))
 } else {
 // nothing todo
 return new Good(new NotificationResponse([]))
 }
}
```

# Open Weather Map Cloud

## ■ Build Cloud Connector for Open Weather Map

- Step 3 : Add Custom Parameter.
  - Add an apiKey entry to the Custom Parameter and enter API Key you have get from Open Weather Map Cloud.

The screenshot shows a 'Custom Parameters' section with the following details:

- Custom Parameters**: Parameters used in the cloud connector code. **1 added**.
- Description: Provide **custom parameters** found in your Cloud Connector Code. With custom parameters, you can easily change the value of a parameter in the future, instead of changing the Groovy code.
- NAME**: apiKey
- VALUE**: 052973e147514754415d133a241 (A red box surrounds this field)
- ADD PARAMETER** button

- Then, save your changes.
- Once you submit it, Cloud Connector code is in approval state.

# Open Weather Map Cloud

## ■ Build Cloud Connector for Open Weather Map

- Step 4 : Create a New Manifest of device type.
  - Go back to the sambpe SDK folder you downloaded from Github.
  - Navigate to '[ARTIKcloud-cloudconnector-samples-master#open-weather-map#src#main#groovy#cloudconnector](#)', then you should see '[manifest.JSON](#)' file.
  - Upload this file to the ARTIK Cloud and activate Manifest.

The screenshot shows two side-by-side views of the ARTIK Cloud interface. On the left, there is a file browser-like view with three files: 'cfg' (with a gear icon), 'manifest' (with a document icon), and 'MyCloudConnect LDF' (with a tablet icon). A red box highlights the 'manifest' file. A red arrow points from this file towards the right-hand interface. On the right, the 'Open Weather Map Type Set Up / Manifests' page is displayed. It shows a table of fields and actions. The 'Fields' section includes: afternoon (Group), icon (String, None), main (String, None), temp\_max (Double, °C), temp\_min (Double, °C), text (String, None), wind (Group), beaufort (Integer, None), speed (Double, None), text (String, None), clouds (Group), all (Integer, %), coord (Group), lat (Double, None), long (Double, None), date (String, None), dt (Long, None), and evening (Group). The 'Actions' section includes: getCurrentWeatherByCity (city String, None, countryCode String, None), getCurrentWeatherByGPSLocation (lat Double, None, long Double, None), getForecastWeatherByCity (city String, None, countryCode String, None), daysToForecast (daysToForecast String, None), getForecastWeatherByGPSLocation (lat Double, None, long Double, None), getWeatherSummary (for String, None, in String, None).

# Open Weather Map Cloud

## ■ Build Cloud Connector for Open Weather Map

- Step 5 : Connect a new device.
  - Go to the My ARTIK Cloud site, connect another device on Device menu.

Give Your Device a Name

OpenWeatherMapConnector

ADD DEVICE

< CHOOSE A DIFFERENT TYPE

Open Weather Map Type

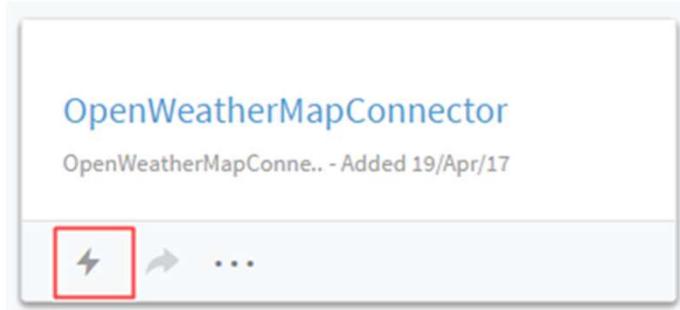
UNIQUE NAME

com.nvyr.openweathermap

# Open Weather Map Cloud

## ■ Build Cloud Connector for Open Weather Map

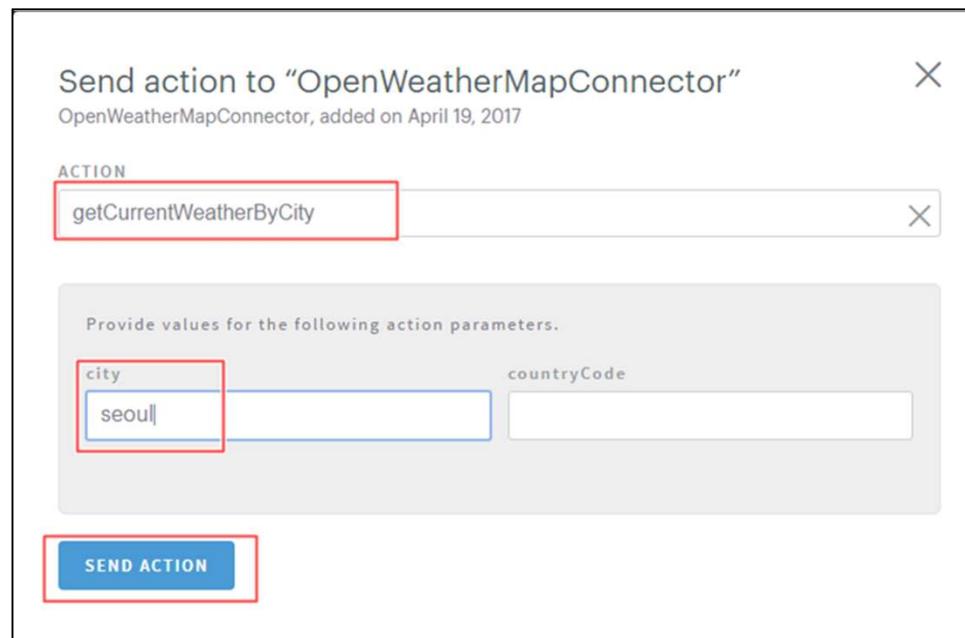
- Step 6 : Send Action to Cloud Connector device.
  - Once the device is connected, send an Action to that device and collect messages received from Open Weather Map cloud.
  - There are many ways to send Action in ARTIK Cloud.
  - You can send one Action by clicking the icon on the Device menu.



# Open Weather Map Cloud

## ■ Build Cloud Connector for Open Weather Map

- Step 6 : Send Action to Cloud Connector device. (Cont.)
  - Cloud Connector for Open Weather Map has several Actions, among which we will send the city name along with the getCurrentWeatherByCity Action to collect the current weather data for that city.
  - By sending this Action to the device, ARTIK will run the methods of the Cloud Connector Code to collect the appropriate data from the connected cloud.



# Open Weather Map Cloud

## ■ Build Cloud Connector for Open Weather Map

- Step 7 : Check the weather data.
  - Go to My ARTIK Cloud and click Data Log tab.
  - Then, you can see that JSON type data including local name, temperature, air pressure, humidity, country name, wind etc.

### Action

| DESTINATION DEVICE                                                               | SENT AT                    | SOURCE DEVICE                                    | ACTION NAME             | ACTION PARAMETERS |
|----------------------------------------------------------------------------------|----------------------------|--------------------------------------------------|-------------------------|-------------------|
| OpenWeatherMapConnector<br><i>Open Weather Map Type</i><br>8e516c5fc32549329c... | Mar 9 2018<br>14:21:52.274 | OpenWeatherMapConnector<br>8e516c5fc32549329c... | getCurrentWeatherByCity | {"city":"seoul"}  |



### Message

| DEVICE                  | RECORDED AT                | RECEIVED AT                | DATA                                                                                                      |
|-------------------------|----------------------------|----------------------------|-----------------------------------------------------------------------------------------------------------|
| OpenWeatherMapConnector | Mar 9 2018<br>14:21:52.893 | Mar 9 2018<br>14:21:52.934 | {"name":"Seoul","dt":1520569800,"main":{"temp_min":7,"pressure":1024,"humidity":49,"temp":7.58,"tem ... ▼ |

# III. ARTIK 유닛 프로젝트 사례



# 유닛 프로젝트 종합

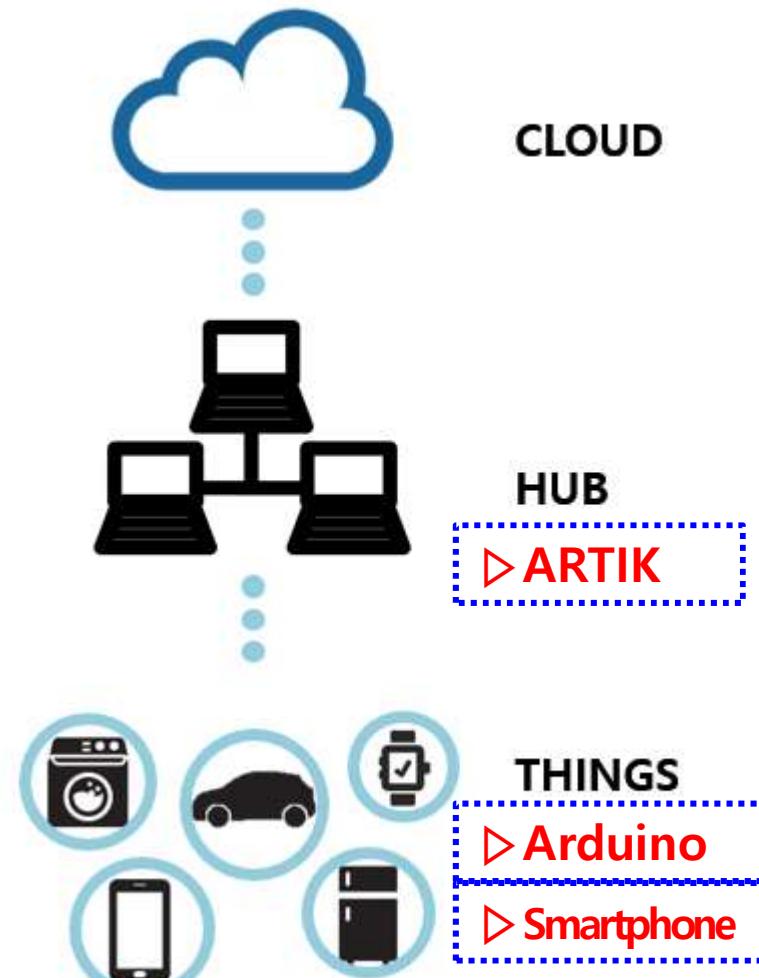
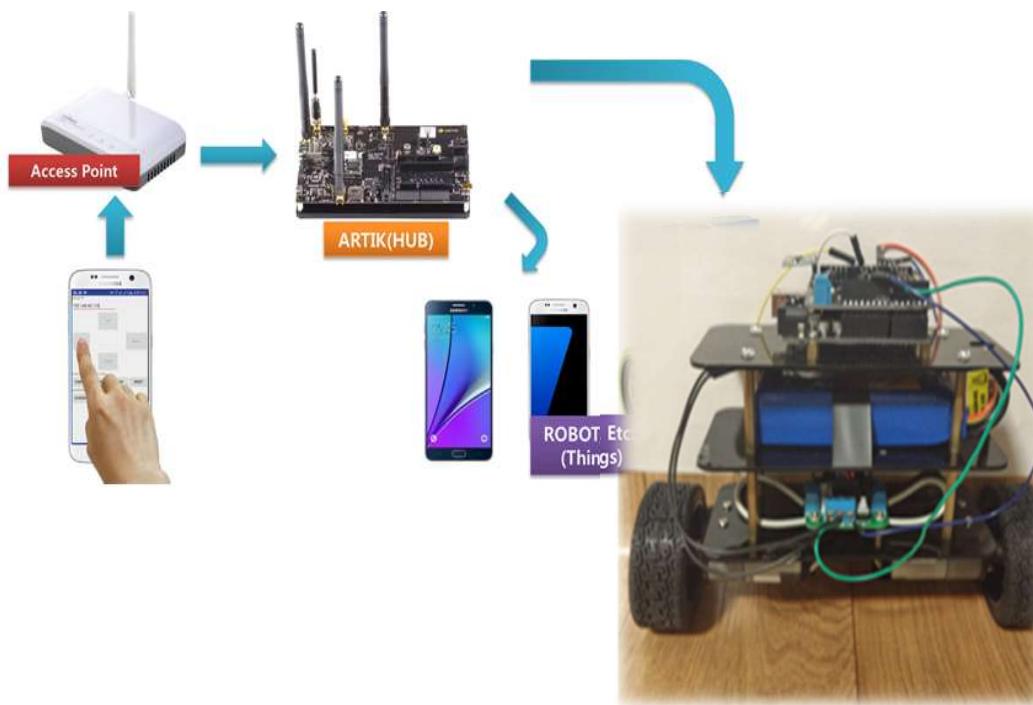
| 로봇 분야                                                                                                                                                                                                                                                                                          | 웨어러블 디바이스 분야                                                                                                                                                                                                                               | 스마트 홈 분야                                                                                                                                                                                                                                                                                                                          |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>밸런싱 로봇</li> <li>• 볼핀타이어를 활용한 세그웨이</li> <li>• 전기자전거</li> <li>• 탐사용 RC카</li> <li>• 손동작 인식 센서를 활용한 RC카</li> <li>• 긴급제동 시스템</li> <li>• StarWars BB-8 드로이드 설계</li> </ul>  | <ul style="list-style-type: none"> <li>장애물 알림용 모자</li> <li>• 거동이 불편한 사람을 위한 케어박스</li> <li>• 독거노인을 위한 심장마비 감지 팔찌</li> <li>• 움직임을 따라하는 로봇 손</li> </ul>  | <ul style="list-style-type: none"> <li>스마트 환기 시스템</li> <li>• 화재대응 시스템</li> <li>• 냉장고 IoT 시스템</li> <li>• 스마트폰을 이용한 도어락</li> <li>• 스마트폰을 이용한 무드등</li> <li>• 클라우드를 활용한 원격 멀티탭</li> <li>• 스마트 요람</li> <li>• 스마트 욕조</li> <li>• 스마트 화분</li> </ul>  |

특히 출원 준비중

# 유닛 프로젝트 – 로봇분야

## ■ 밸런싱 로봇

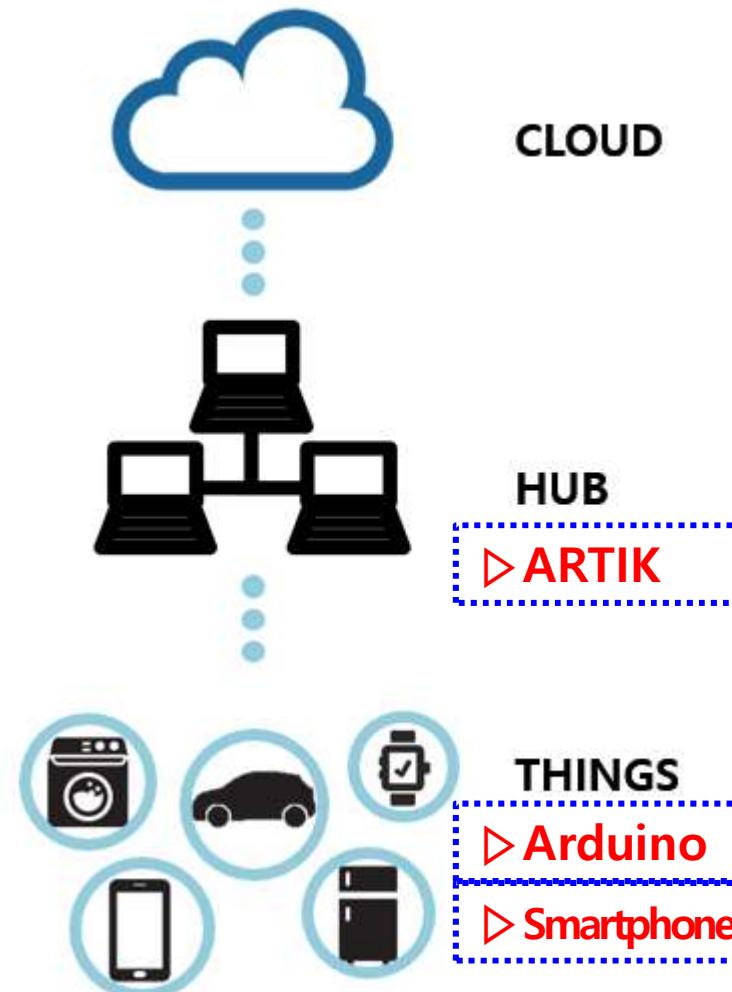
- ✓ 2바퀴로 균형을 잡는 로봇
- ✓ 자이로/가속도 센서를 이용
- ✓ 로봇 본체(Thing)는 Arduino로 구현
- ✓ 이동 명령 및 수행 임무는 스마트폰을 통해 ARTIK(Hub)를 거쳐 로봇 본체가 받음



# 유닛 프로젝트 – 로봇분야

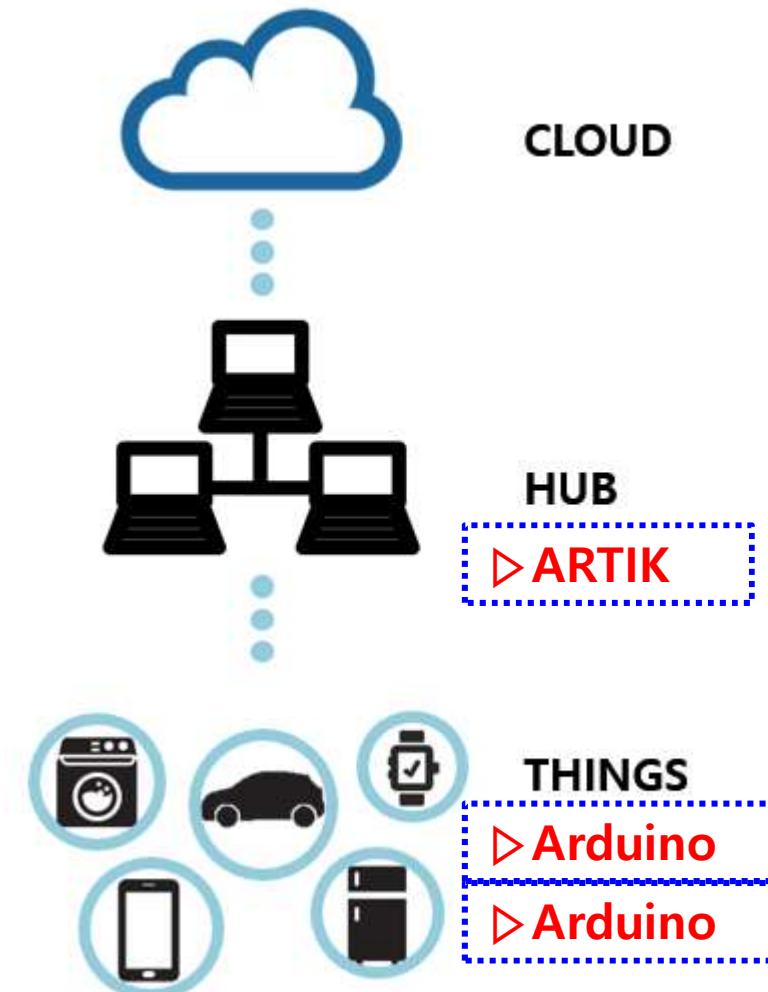
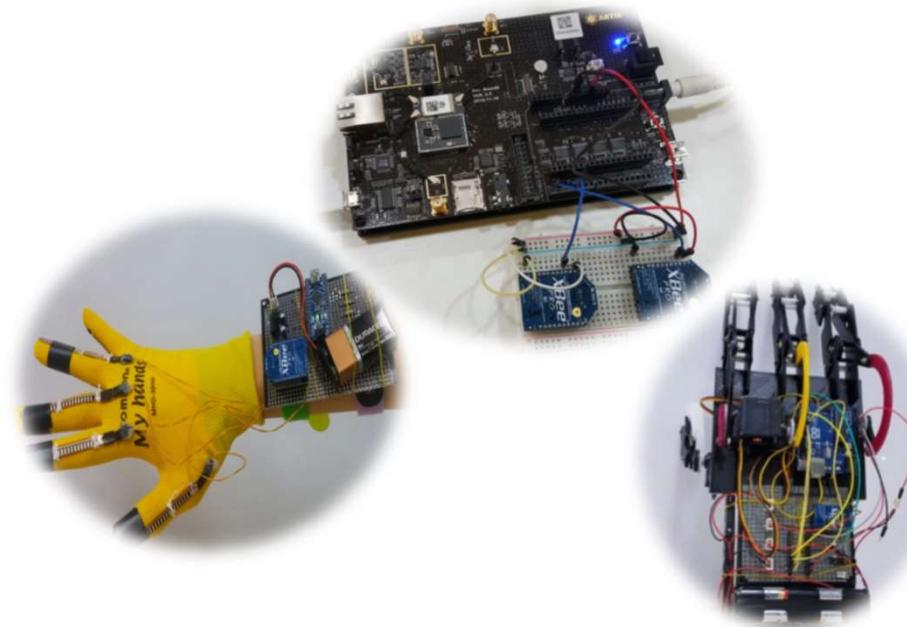
- 탐사용 RC카

- ✓ 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
  - ✓ 휠 센서를 통한 사용자 모션 인식
  - ✓ 사용자 모션에 따른 서보모터 제어
  - ✓ Arduino와 ARTIK 사이의 무선통신
    - ✓ ZigBee 모듈 활용



# 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 구마다 개별 제어

