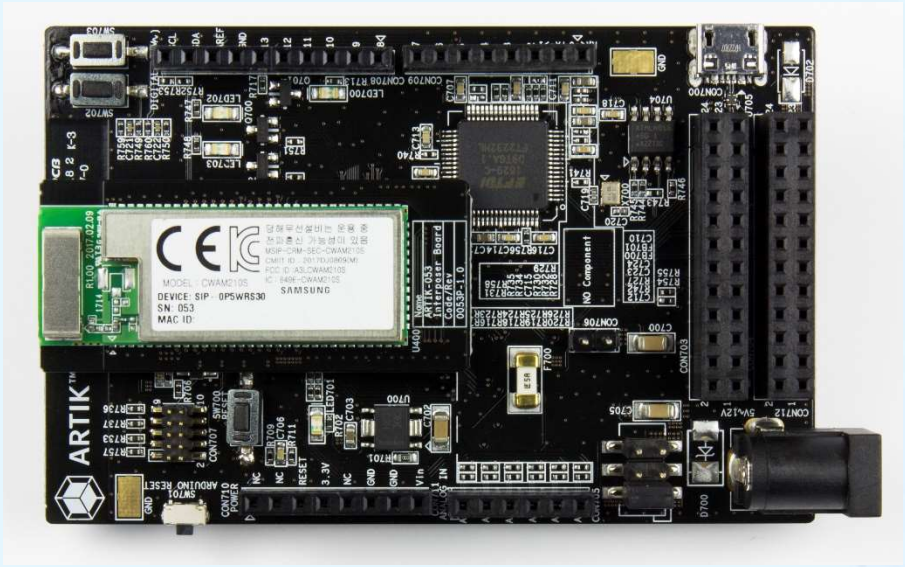
# ARTIK 053 SPEC & Setup

- ARTIK 053 Specification
- ARTIK 053 Power Configuration
- ARTIK 053 GPIO Header Map
- ARTIK 053 Arduino Shield & Sensors
- ARTIK 053 Setup, Driver Installation
- ARTIK 053 Demo (On-board Example)

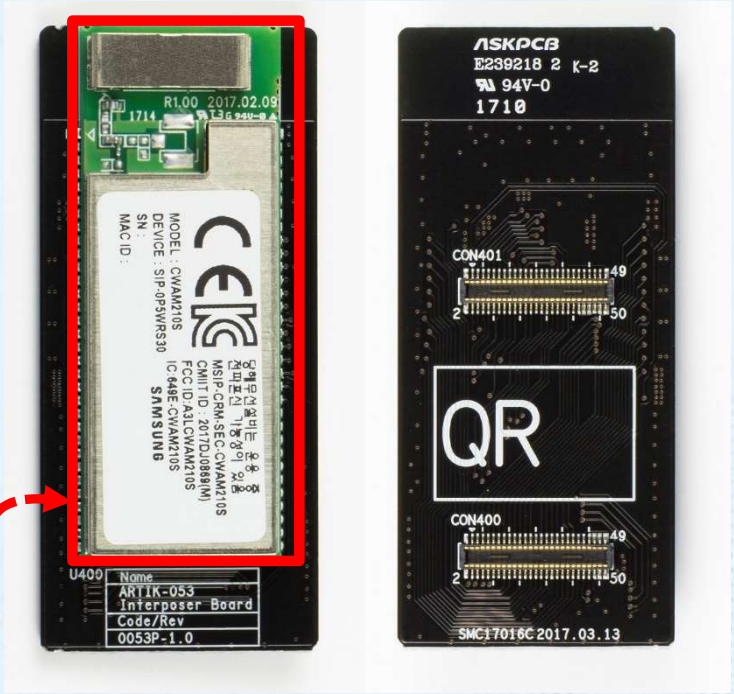
# ARTIK 053 Specification

## Overview

**Evaluation Kit**



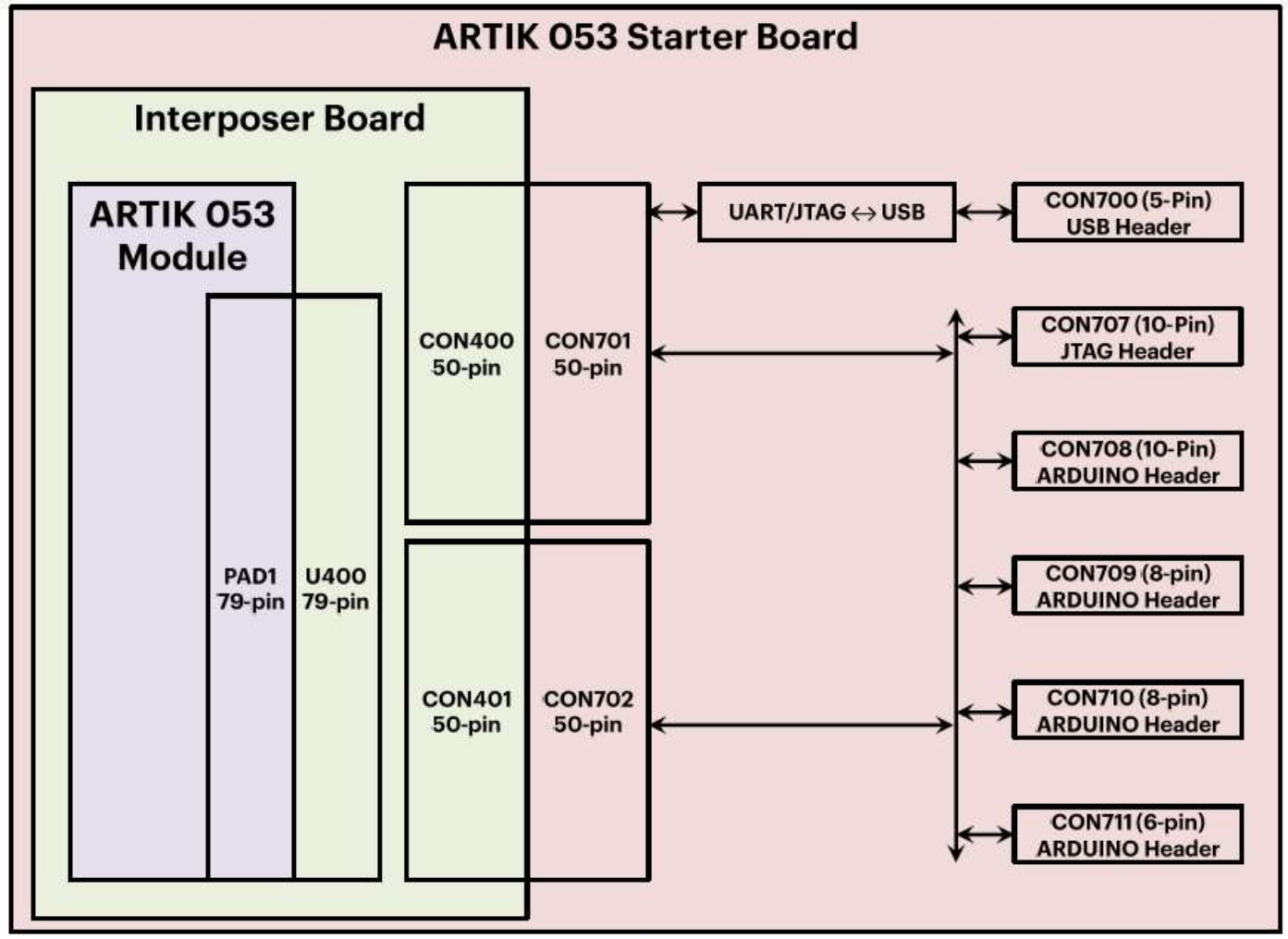
**Interposer Board**




**ARTIK 053 Module**

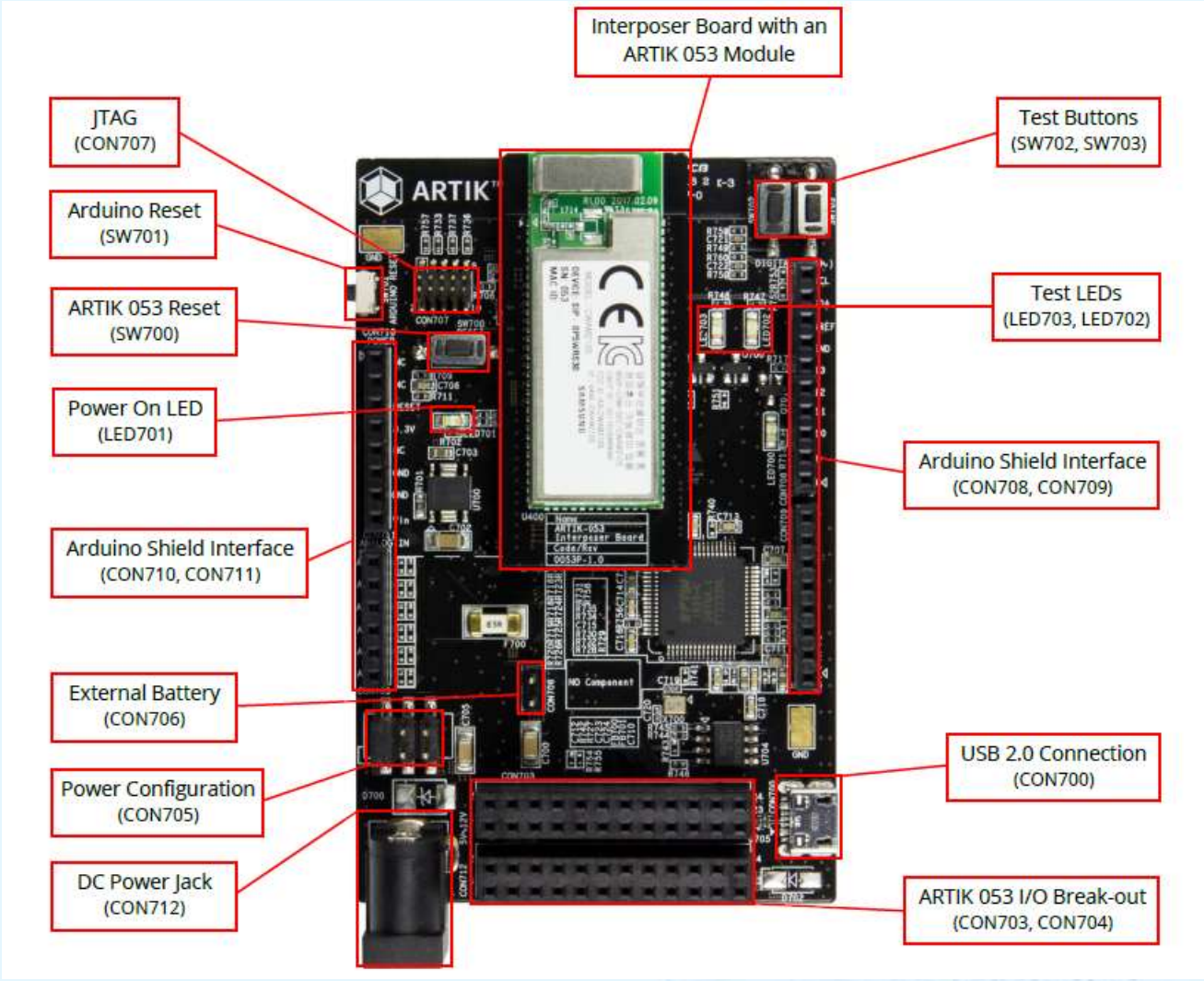
# ARTIK 053 Specification

## Starter Kit Block Diagram



# ARTIK 053 Specification

## Board Interface



# ARTIK 053 Specification

- Main Features
  - CPU: 32bit ARM Cortex R4 (@320MHz), I-Cache/D-Cache 32KB
  - RAM: 1280 KB (General Use)
    128 KB (Global IPC data)
  - Flash: 8 MB
  - Secure System: AES/DES/TDES, SHA-1/SHA-2, PKA, PRNG/DTRNG, Secure key storage
  - PUF: Physical Unclonable Function
  - Wi-Fi: Certified IEEE 802.11 b/g/n, 2.4 GHz radio
  - Regulatory: FCC(U.S), IC(Canada), CE(EU), KC(Korea), SRRC(China)
  - Power Supply: 5~12V
  - I/O: UART, I2C, SPI, PWM, ADC, GPIO
  - Dimension: 15mm(W) x 40mm(H) x 3mm (D)

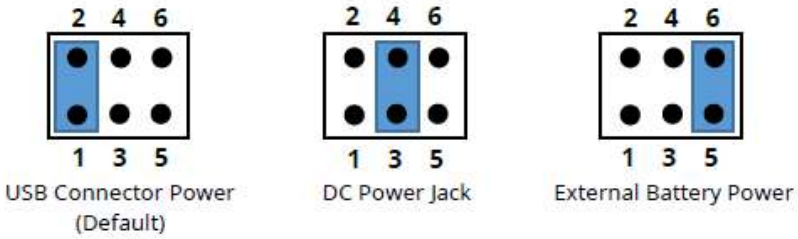
# ARTIK 053 Power Configuration

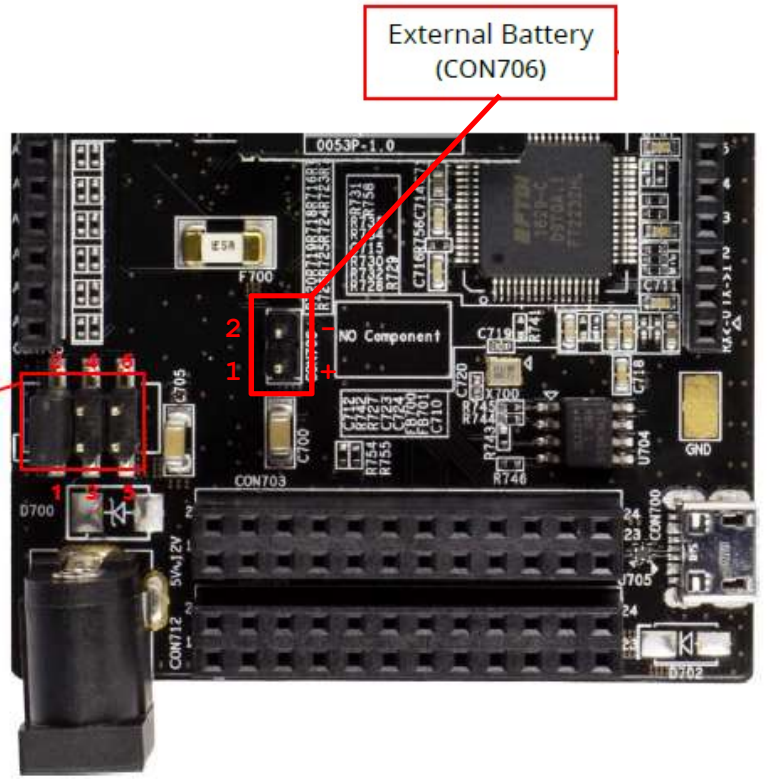
## Power & Battery Configuration

| Pin Number | Description | Comment |
|---|---|---|
| 1 | Plus (Square pad) | 5V-12V External Battery Power. Example, 4 AA batteries |
| 2 | GND (Round pad) | Ground |

Power Configuration (CON705 Header)

| Pin Number | Description | Comment |
|---|---|---|
| 1 - 2 | VCC_USB5P0 | USB Connector Power Source |
| 3 - 4 | DC Jack | DC Jack Power Source |
| 5 - 6 | Battery | External Battery Power Source |



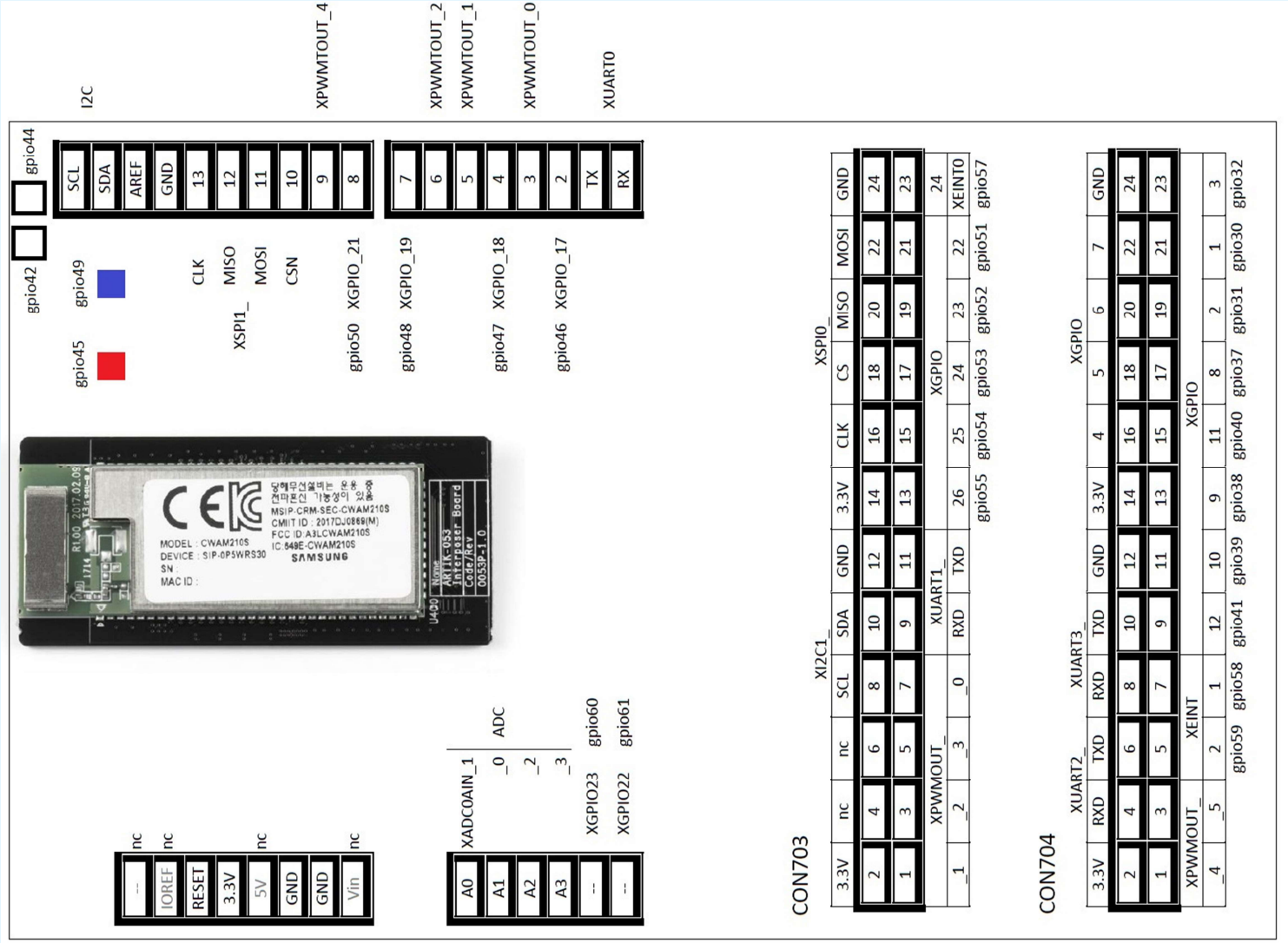

Power settings jumper location

# ARTIK 053 GPIO Header Map

- Header Map(1/2)
  - The signal levels and positions are Arduino-compatible for certain tested shields
  - Other shields may require adaptation
    - XGPIO pins used as outputs might require a pre-driver circuit to meet the drive strength requirements of some Arduino shields.
    - XGPIO pins used as inputs for 5V signals would need to be buffered as the ARTIK 053 pins are not 5V-tolerant.
  - Tizen RT source repository
    - https://github.com/SamsungARTIK/Tizen RT
  - Tizen RT source code for GPIO Mapping: *tinyara/os/arch/arm/src/artik053/src/artik053_boot.c*

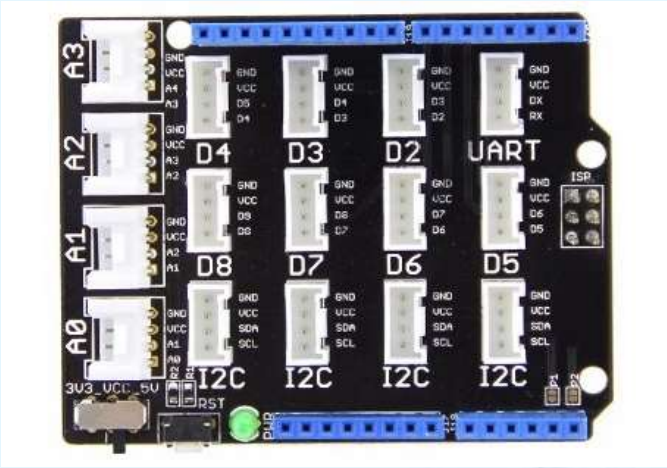
# ARTIK 053 GPIO Header Map

## Header Map(2/2)

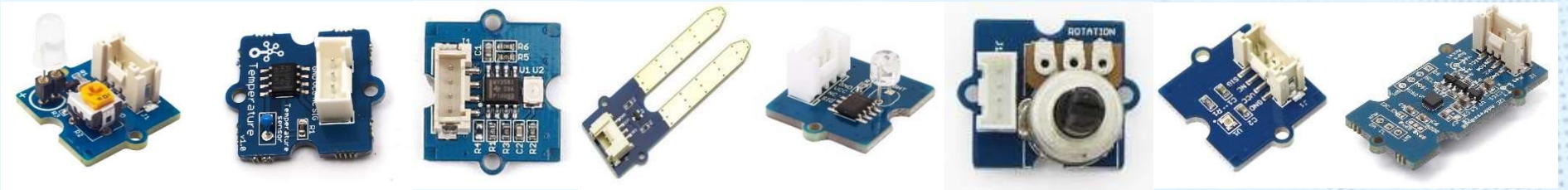


# ARTIK 053 Arduino Shield & Sensor

- ARTIK 053 Compatible Arduino Shield
  - Groove Base Shield




  - Sensors



LED (Digital) | Temp. | UV | Moisture | Light | Angle | Luminance (Analog) | Accel (I2C)

# ARTIK 053 Arduino Shield & Sensor

## ARTIK 053 Compatible Arduino Shield

### Groove Shield and Sensor Details

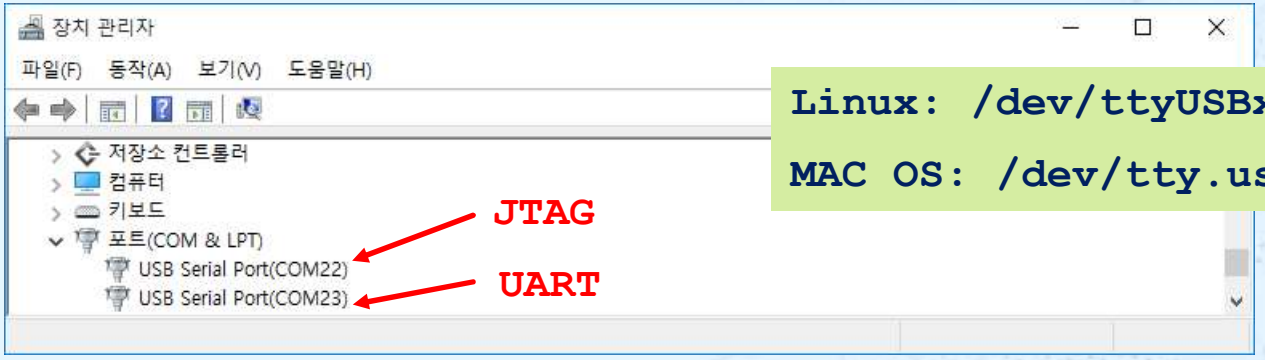
| Type | URL |
|---|---|
| Base Shield | http://wiki.seeed.cc/Base_Shield_V2/ |
| LED Socket | http://wiki.seeed.cc/Grove-LED_Socket_Kit/ |
| Temperature Sensor | http://wiki.seeed.cc/Grove-Temperature_Sensor/ |
| UV Sensor | http://wiki.seeed.cc/Grove-UV_Sensor/ |
| Moisture Sensor | http://wiki.seeed.cc/Grove-Moisture_Sensor/ |
| Light Sensor | http://wiki.seeed.cc/Grove-Light_Sensor/ |
| Rotary Angle Sensor | http://wiki.seeed.cc/Grove-Rotary_Angle_Sensor/ |
| Luminance Sensor | http://wiki.seeed.cc/Grove-Luminance_Sensor/ |
| MPU9250 Accelerometer | http://wiki.seeed.cc/Grove-IMU_9DOF_v2.0/ |

# ARTIK 053 Setup

- Getting Up and Running
  - Power the board
    - Verify that the ARTIK 053 module Interposer board is firmly snapped onto the base Starter board
    - Attach the USB cable to the Starter board micro type B receptacle and to your host PC to provide serial communications
    - Optionally, in the future: Connect an external power adapter. (Don't connect an external power supply for the initial testing here, as it will complicate power sequencing.)

# ARTIK 053 Setup

- Getting Up and Running
  - Enable the debug interface
    - The ARTIK 053 Starter Kit communicates through an FTDI USB-to-serial device
    - The driver exposes two COM port devices:
      - UART interface for console debug (provide CLI)
      - JTAG interface for program transfer
    - Once the base COM port drivers are installed on your development PC and the ARTIK board is attached, the two serial COM ports become accessible:




```
Linux: /dev/ttyUSBx
MAC OS: /dev/tty.usbserial.xxx
```

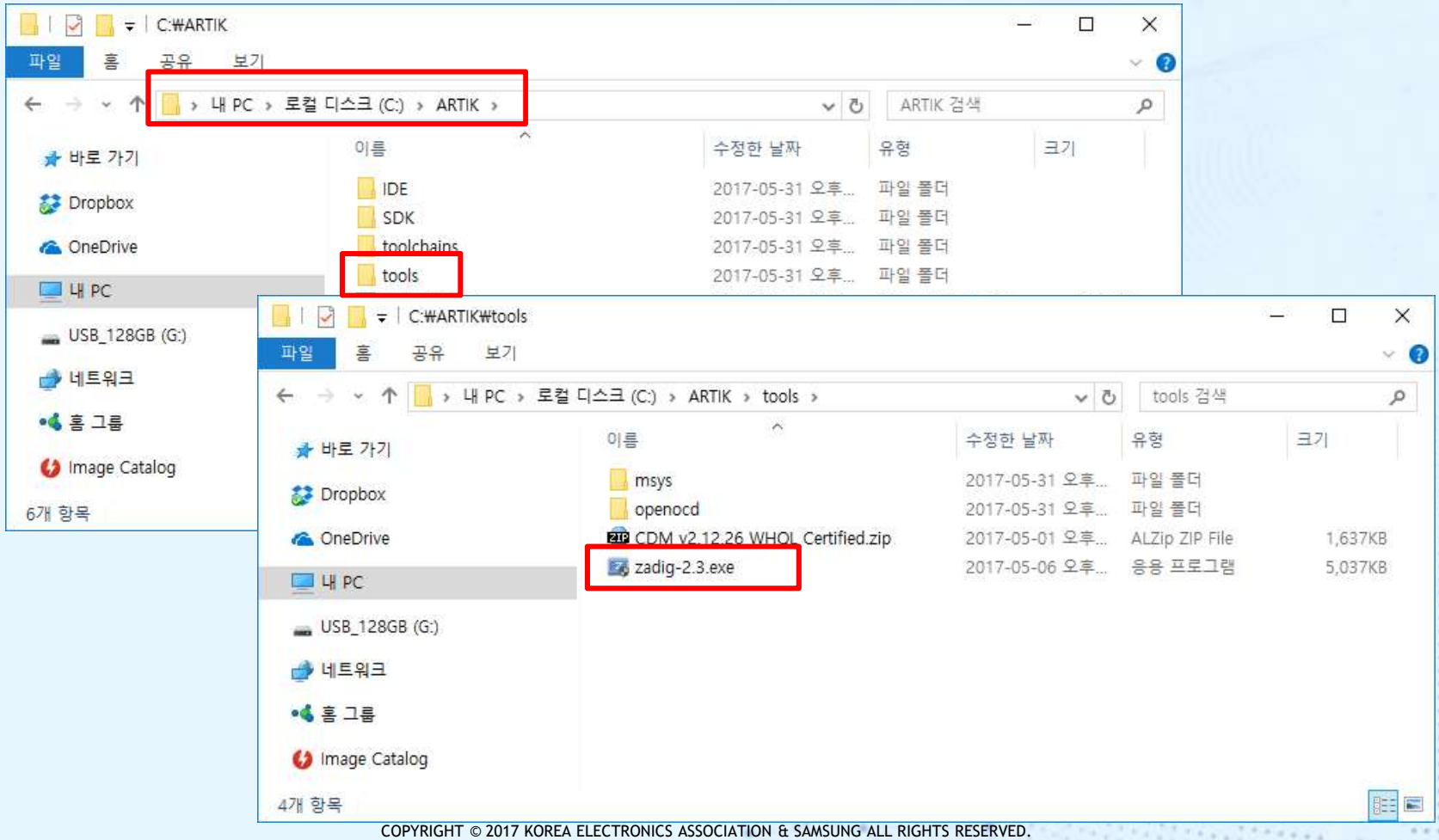
# ARTIK 053 Setup

- JTAG Driver Installation
  - The port that handles JTAG transfers will need a new driver installed for flashing and debug activities
    - Browse to the <ARTIK IDE Installation Directory>\tools directory, and open the zadig-2.3.exe application
    - Under the Devices drop-down list, select Interface 0 of the dual COM port interfaces
    - Replace the FTDIBUS driver with the WinUSB driver by clicking Replace Driver

# ARTIK 053 Setup

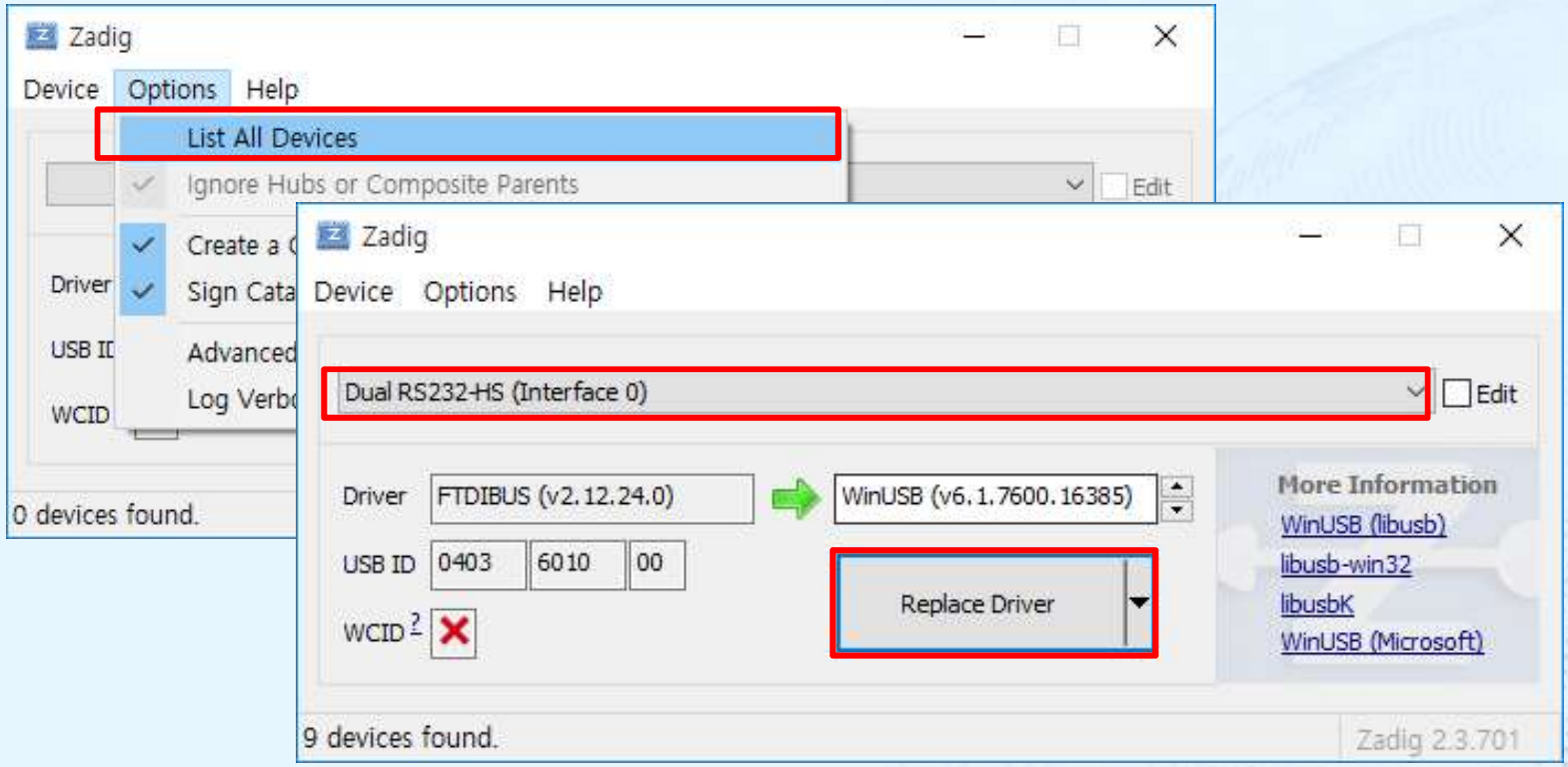
## JTAG Driver Installation

- Browse to the <ARTIK IDE Installation Directory>\tools directory, and open the zadig-2.3.exe application



# ARTIK 053 Setup

- JTAG Driver Installation
  - Zadig Menu → Options → List All Devices
  - Select "Dual RS232-HS (Interface 0)"

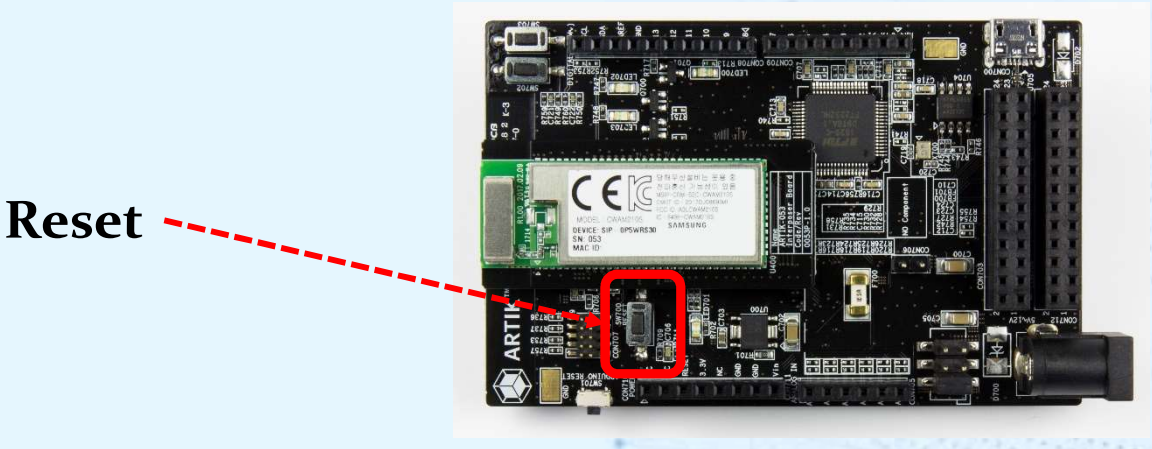


# ARTIK 053 Setup

- JTAG Driver Installation
  - JTAG Interface: Dual RS232-HS

# ARTIK 053 Setup

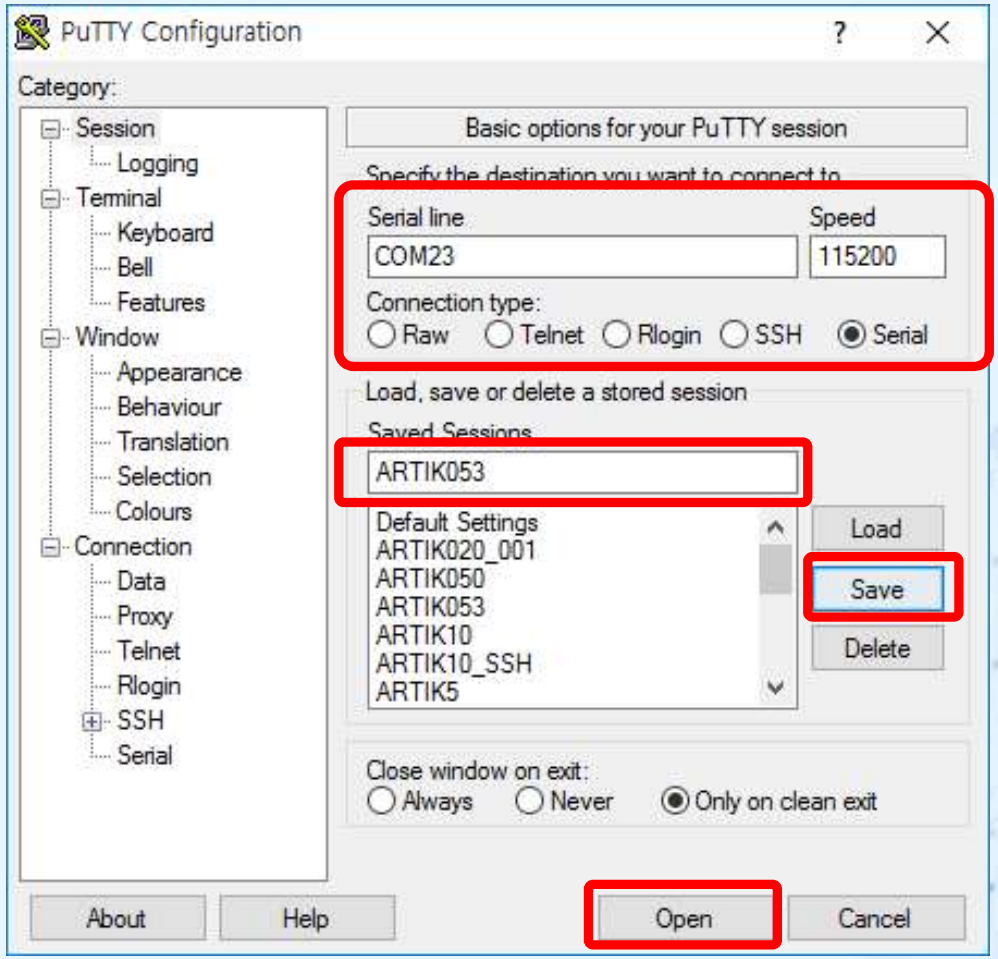
- Trying It Out
  - You're ready to start! First, a word about Reset: **Don't press it** unless directed.
  - Here's why: The ARTIK 053 module resets itself when you apply power (here, by connecting the USB cable).
  - After that, the pre-loaded firmware is designed to enter one of two modes:
    - Command Line Interface(CLI) mode (default)
    - on-boarding mode (if you press Reset)



# ARTIK 053 Setup

## Terminal Program Setup (Putty)

### Putty UART Configuration




**Connection type: Serial**

**Port Number: COM#**

**baud rate: 115200 bps**

**Save session**

**Open(Connect)**

# ARTIK 053 Setup

- CLI Mode
  - To Enter CLI Mode: Just Type Enter

```
U-Boot 2017.01-00064-gd5533a7a97d3 (Mar 13 2017 - 18:38:26 +0900)

CPU:   Exynos200 @ 320 MHz
Model: ARTIK-051 based on Exynos T20 (DA variant)
DRAM:  722 KiB
WARNING: Caches not enabled
BL1 released at 2017-3-13 15:00
Flash: 8 MiB
*** Warning - bad CRC, using default environment

In:    serial@80180000
Out:   serial@80180000
Err:   serial@80180000
Hit any key to s}=========================================
 TinyAra version: 0.98-rc3 (0053GC02-098-rc3-01Q3)
 ------------------------------------
 Build User : junhwan@Linux
 Build Time : 2017-04-19 10:04:05
=========================================
Starting supplicant as daemon...
TASH>>Starting AP ARTIK_286d9740126b
Web server started
ARTIK Onboarding Service started

TASH>>
```

CLI Mode

# ARTIK 053 Setup

- Security Credentials
  - Each ARTIK 053 module comes with unique security credentials in tamper-proof storage
  - The module is pre-loaded with a Tizen RT demonstration image that provides easy and secure connectivity to Wi-Fi and to ARTIK Cloud
    - The Samsung ARTIK App connects your device to Wi-Fi and registers it with ARTIK Cloud – instantly allowing you to remotely control and monitor ARTIK 053 GPIO functions. There is no charge for the app or service
    - If you prefer, you can bypass onboarding. You will need to manually set up Wi-Fi using the normal terminal emulator console interface.

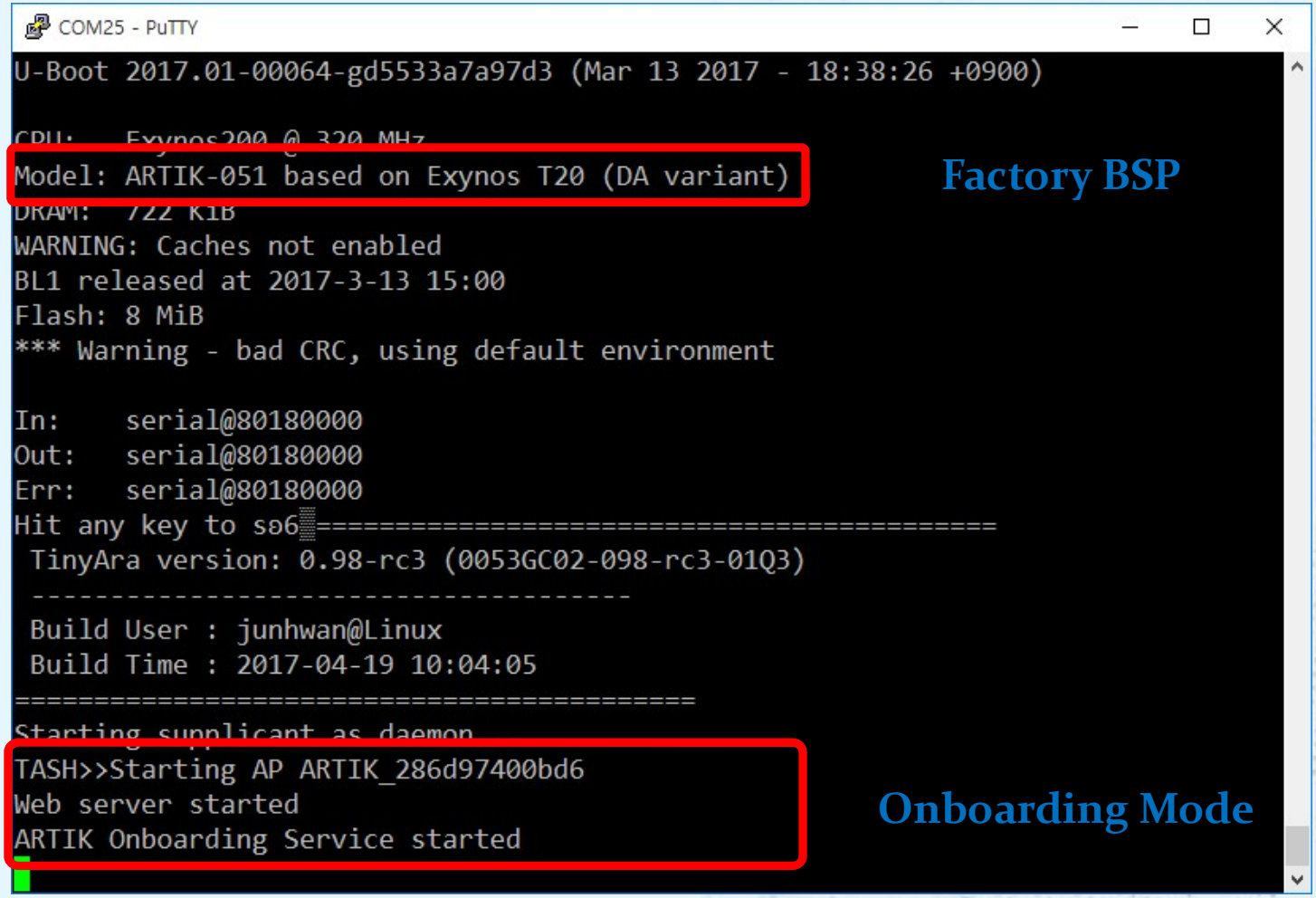
| AUTO-CONNECT TO ARTIK CLOUD | USE COMMAND LINE INTERFACE (CLI) |
|---|---|
| Attach USB cable and press **reset** (on-boarding mode) | Attach USB cable but do **not** press reset (CLI mode) |
| Follow auto-connect instructions | Enter commands at TASH> prompt on console |

# ARTIK 053 Demo

- Auto-connect ARTIK 053 to WiFi and ARTIK Cloud
  - You'll need to "on-board" your kit once initially and any time you load new firmware
  - Install ARTIK IDE, and then install the free Samsung ARTIK app to your Android phone
  - With everything connected and ready, press **reset** on the Starter Kit board
  - You'll see a message on the terminal emulator console that you're in on-boarding mode

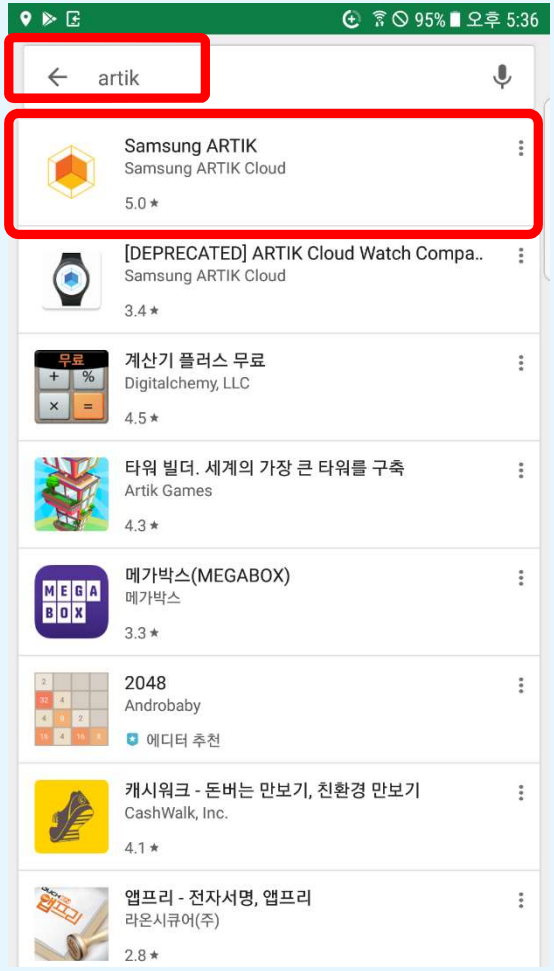
# ARTIK 053 Demo

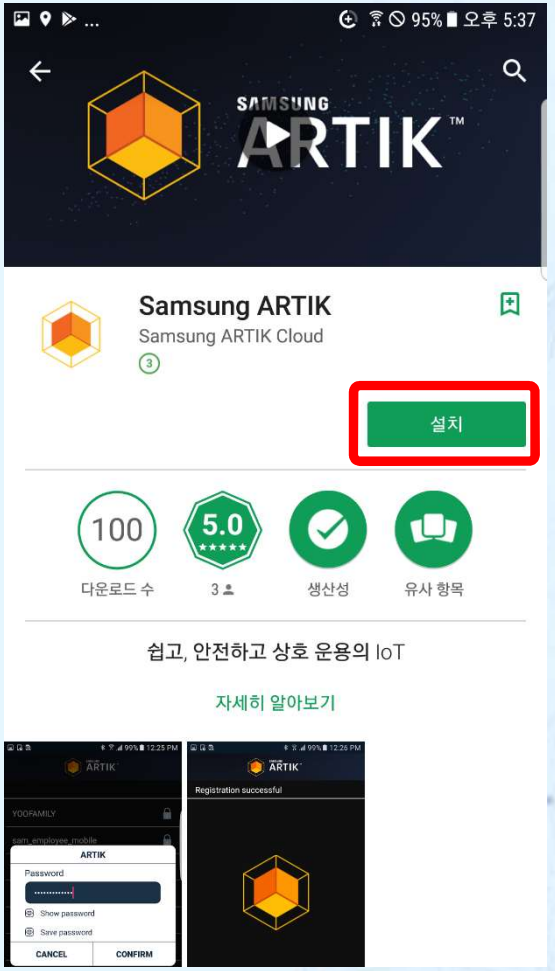
- On-boarding Mode
  - ARTIK Operate AP Mode

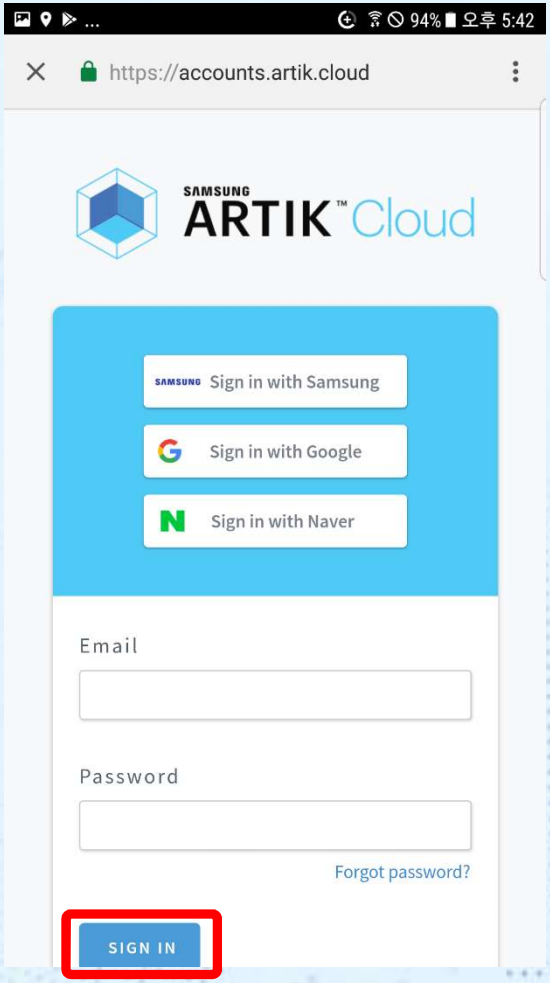


# ARTIK 053 Demo

- On-boarding Mode with Smart Phone App
  - Install ARTIK App



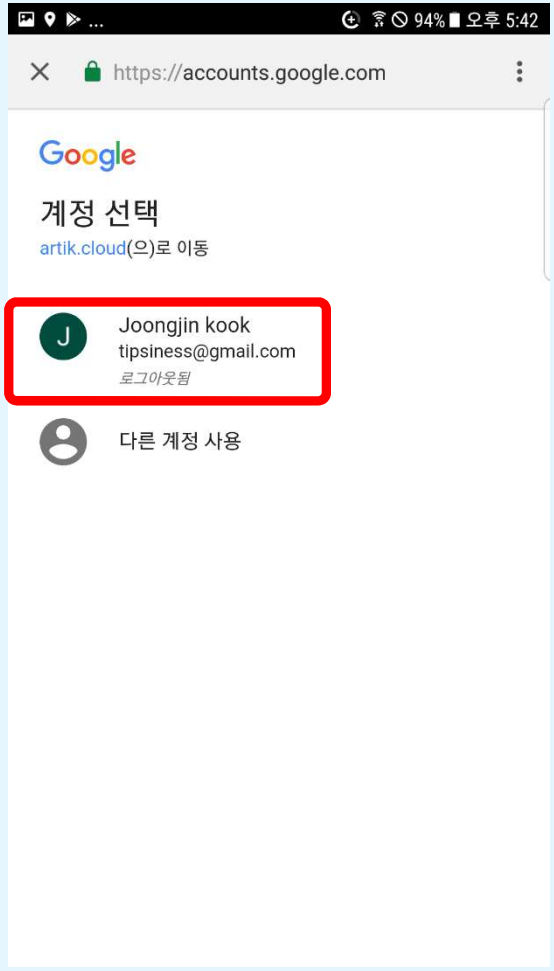



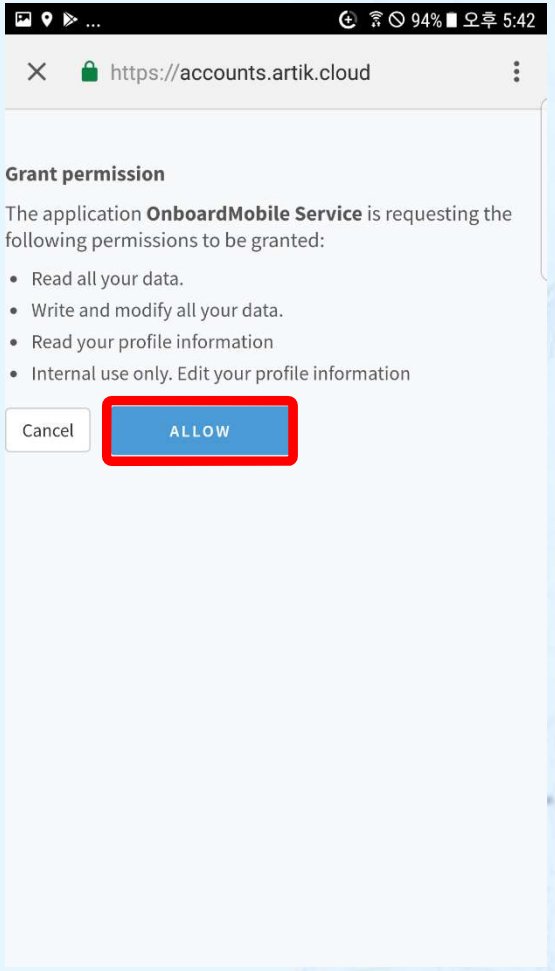



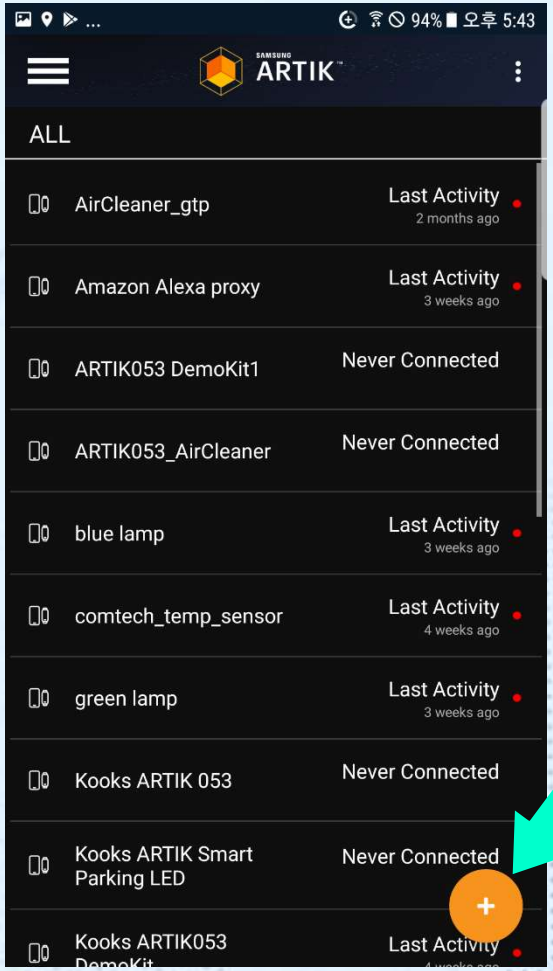
# ARTIK 053 Demo

## On-boarding Mode with Smart Phone App

### Sign In ARTIK App

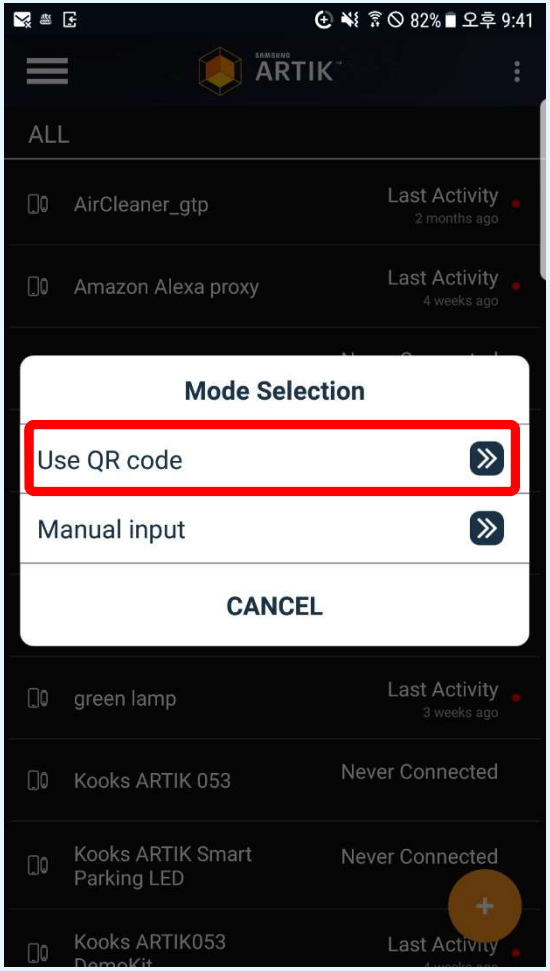


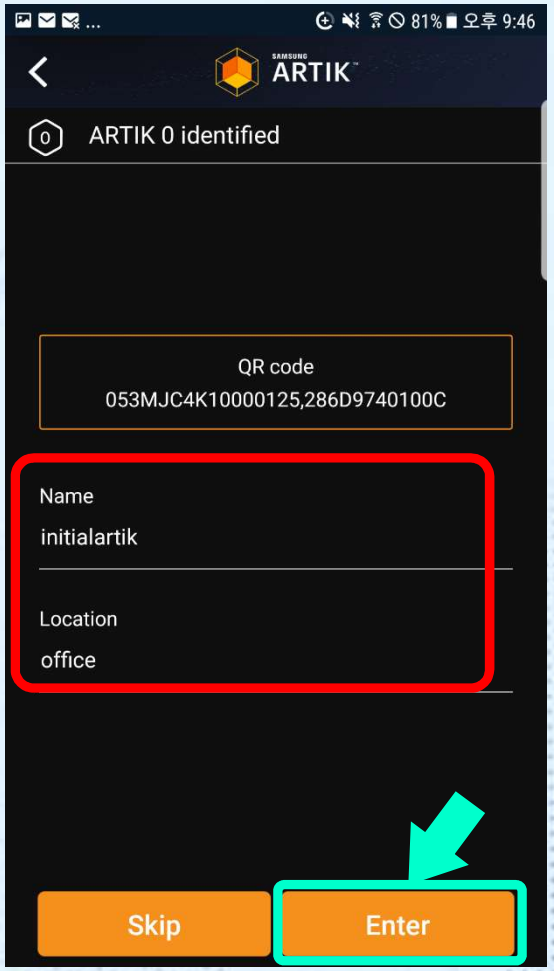



# ARTIK 053 Demo

- On-boarding Mode with Smart Phone App
  - Register ARTIK 053 to ARTIK App

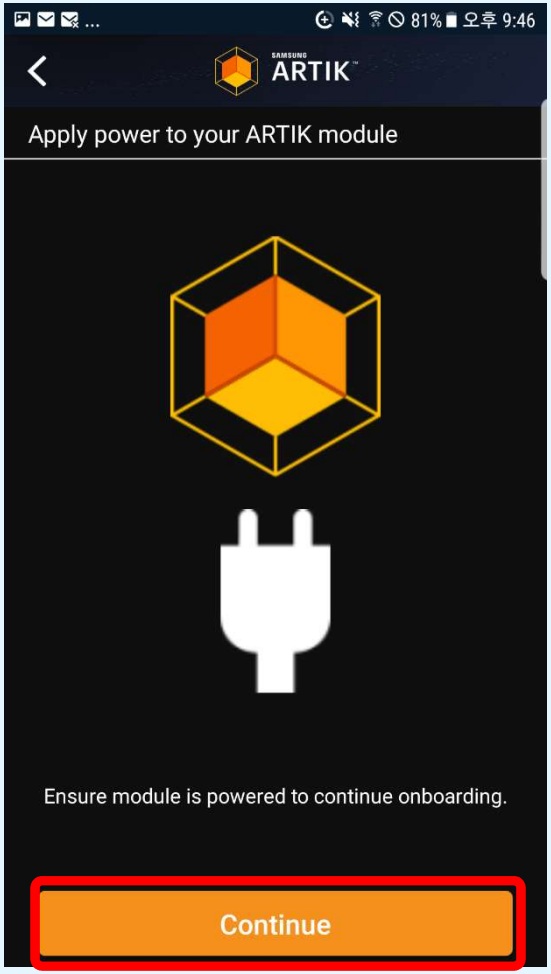


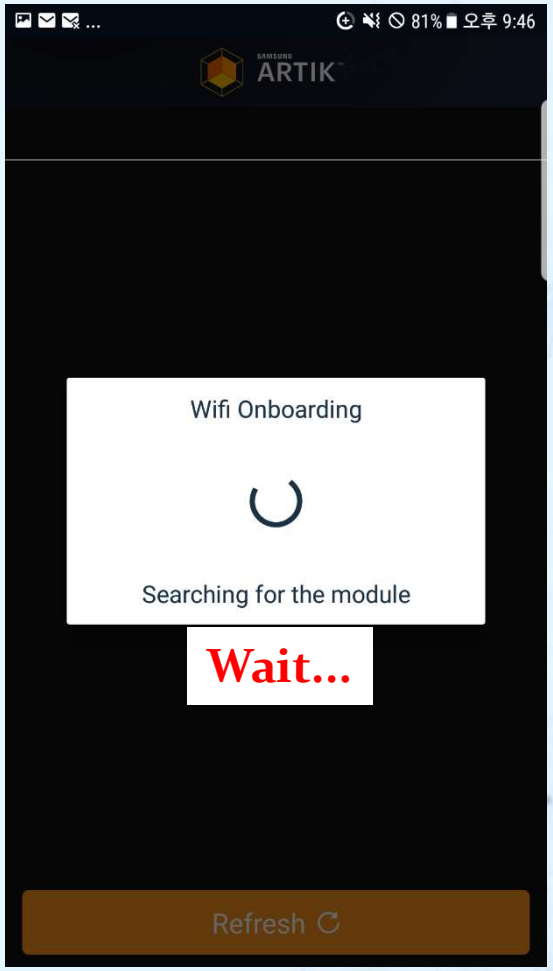


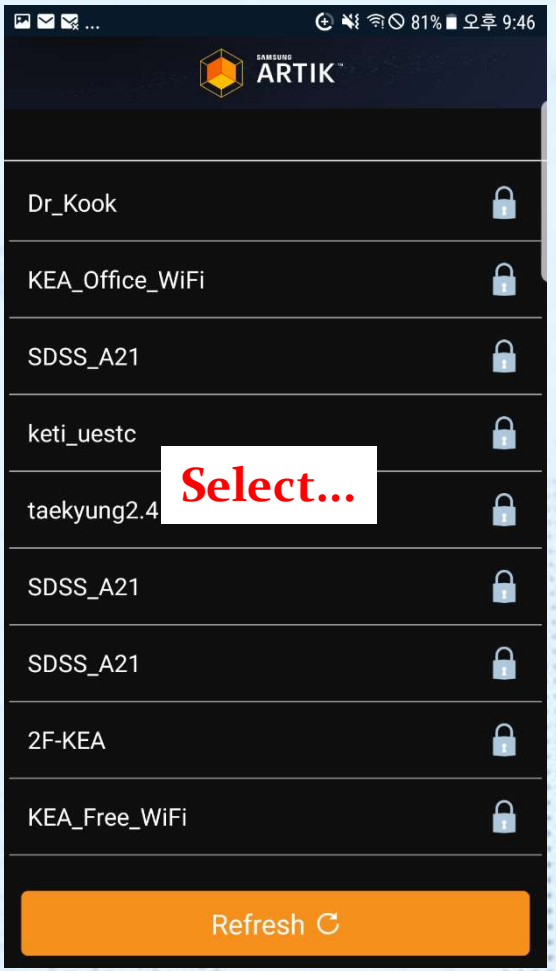


# ARTIK 053 Demo

- On-boarding Mode with Smart Phone App
  - Configure WiFi

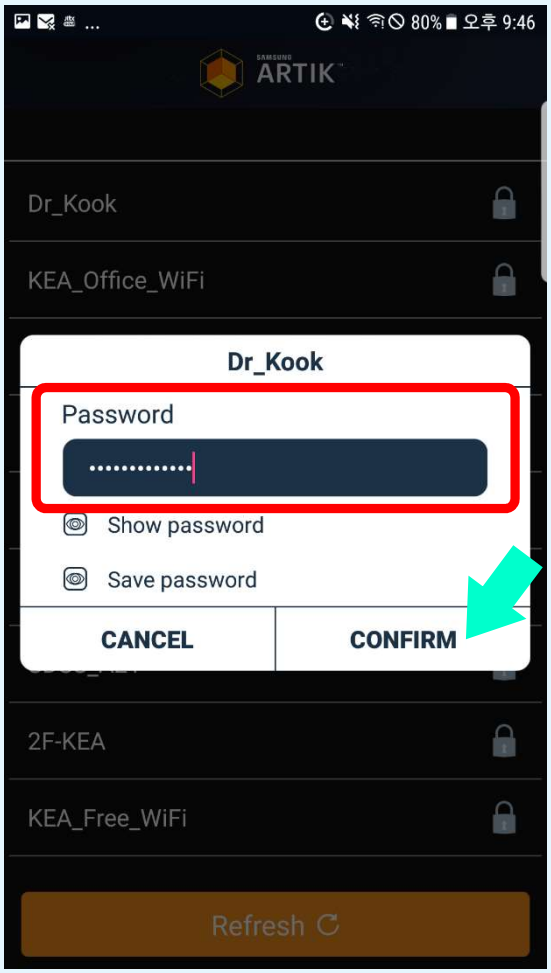


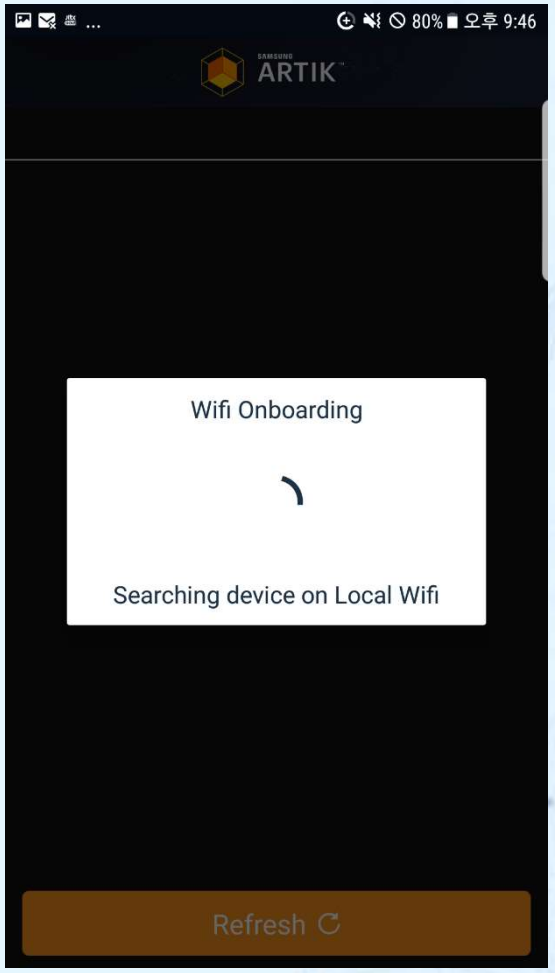



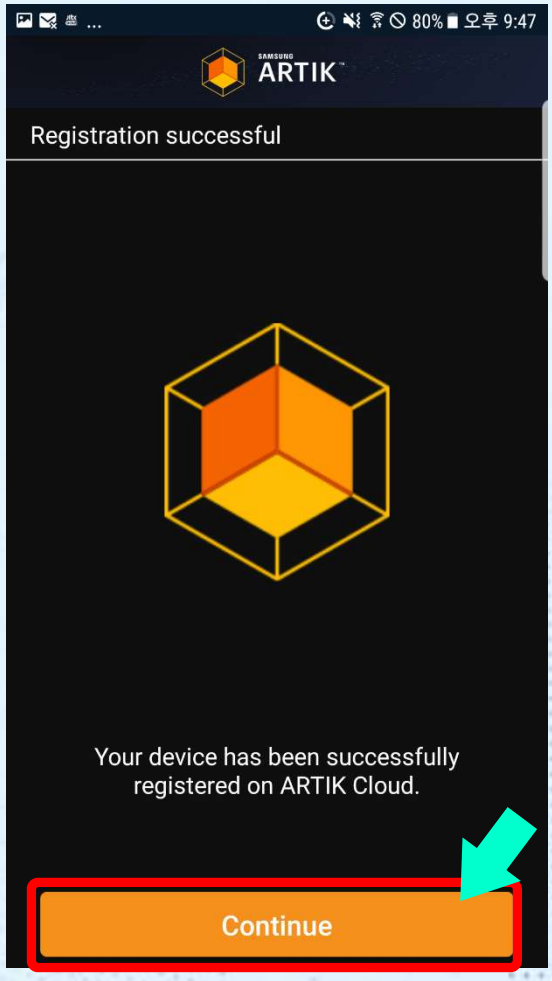



# ARTIK 053 Demo

- On-boarding Mode with Smart Phone App
  - Configure WiFi

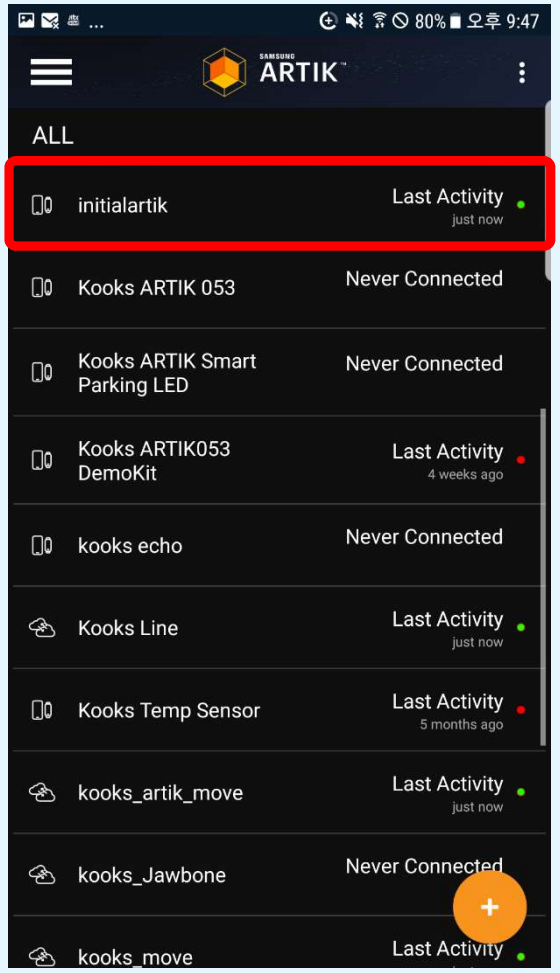


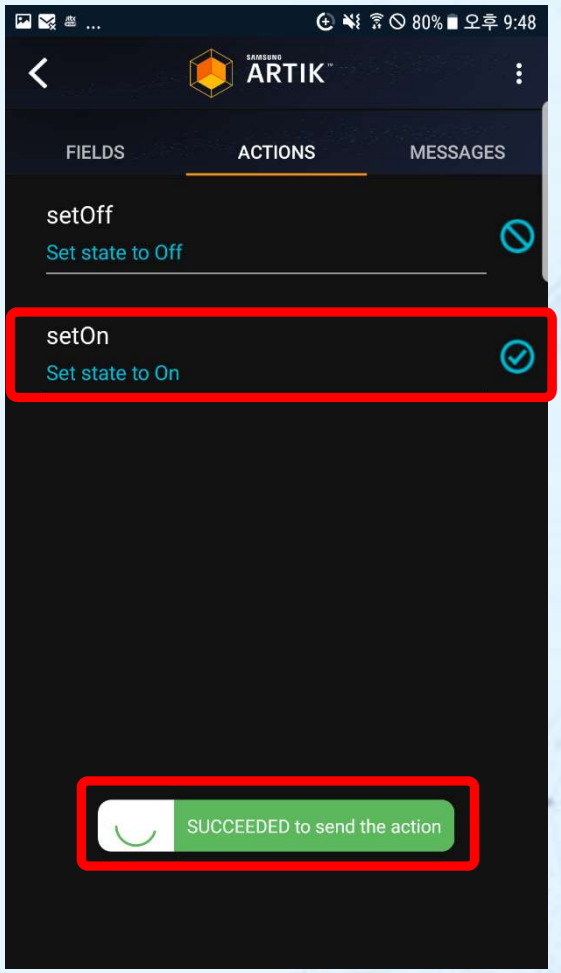



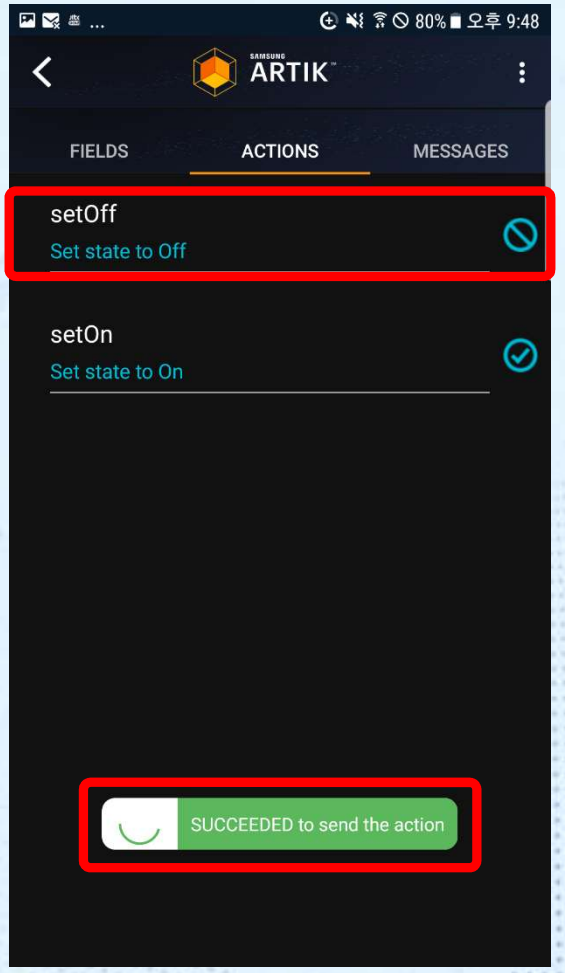



# ARTIK 053 Demo

- Onboarding Mode with Smart Phone App
  - Send an Action to ARTIK 053



# Tizen RT

- NuttX
- Tizen RT Outlook
- Tizen RT Sub-systems

# NuttX

- NuttX RTOS
  - NuttX was first released in 2007 by Gregory Nutt
  - Emphasis on standards compliance and small footprint
  - Scalable from 8-bit to 32-bit microcontroller environments
  - Governing standards in NuttX are Posix and ANSI standards
  - Supported Platforms
    - ARM7TDMI, ARM920T, ARM926EJS, ARM Cortex-A5/8/9, ARM Cortex-R4, ARM Cortex-M0/3/4//7
    - AVR, AVR32
    - 80x86
    - Freescale M68HCS12, MicroChip PIC32MX/Z, LM32, Espressif ESP32, ...

# NuttX

- NuttX RTOS
  - File system
    - Tiny in-memory, root pseudo-file-system
    - VFS
    - FAT12/16/32
    - NFS Client
    - NXFFS: Tiny NuttX wear-leveling FLASH file system
    - SMART
    - ROMFS, BINFS, Union
    - procfs

# NuttX

- NuttX RTOS
  - Device Drivers
    - VFS supports character and block drivers
    - Asynchronous I/O (AIO)
    - Network, USB (host), USB (device), serial, I2C, I2S, NAND, CAN, ADC, DAC, PWM, Quadrature Encoder, generic timer, and watchdog timer driver architectures
    - RAMDISK, pipes, FIFO, /dev/null, /dev/zero, /dev/random, and loop drivers
    - Generic driver for SPI-based or SDIO-based MMC/SD/SDH cards
    - Power Management sub-system
    - Graphics Devices: framebuffer drivers, graphic and segment LCD drivers
    - Audio Subsystem: CODECs, audio input and output drivers

# NuttX

- NuttX RTOS
  - Device Drivers
    - Crytopgraphic sub-system
    - Input Devices: Touchscreen, USB keyboard, USB mouse, analog/discrete joystock, GPIO-based buttons and keypads
    - Memory Technology Devices (MTD)
    - Analog Devices: Support for Analog-to-Digital conversion (ADC), Digital-to-Analog conversion (DAC), multiplexers, and amplifiers

# NuttX

- NuttX RTOS
  - Networking
    - Multiple network interface support; multiple network link layer support
    - IPv4, IPv6, TCP/IP, UDP, ARP, ICMP, ICMPv6, IGMPv2 (client) stacks
    - User space stacks
    - Stream, datagram, and raw packet sockets
    - Raw socket and local, Unix domain socket support
    - DNS name resolution / NetDB
    - IEEE 802.11 FullMac
    - IEEE 802.15.4 MAC + 6loWPAN

# NuttX

- NuttX RTOS
  - Networking
    - SLIP, TUN/PPP, local loopback devices
    - A cJSON port
    - BSD compatible socket layer.
    - Networking utilities (DHCP server/client, SMTP, TELNET, FTP server/client, TFTP, HTTP server/client, PPPD, NTP
    - ICMPv6 autonomous auto-configuration
    - NFS Client

# NuttX

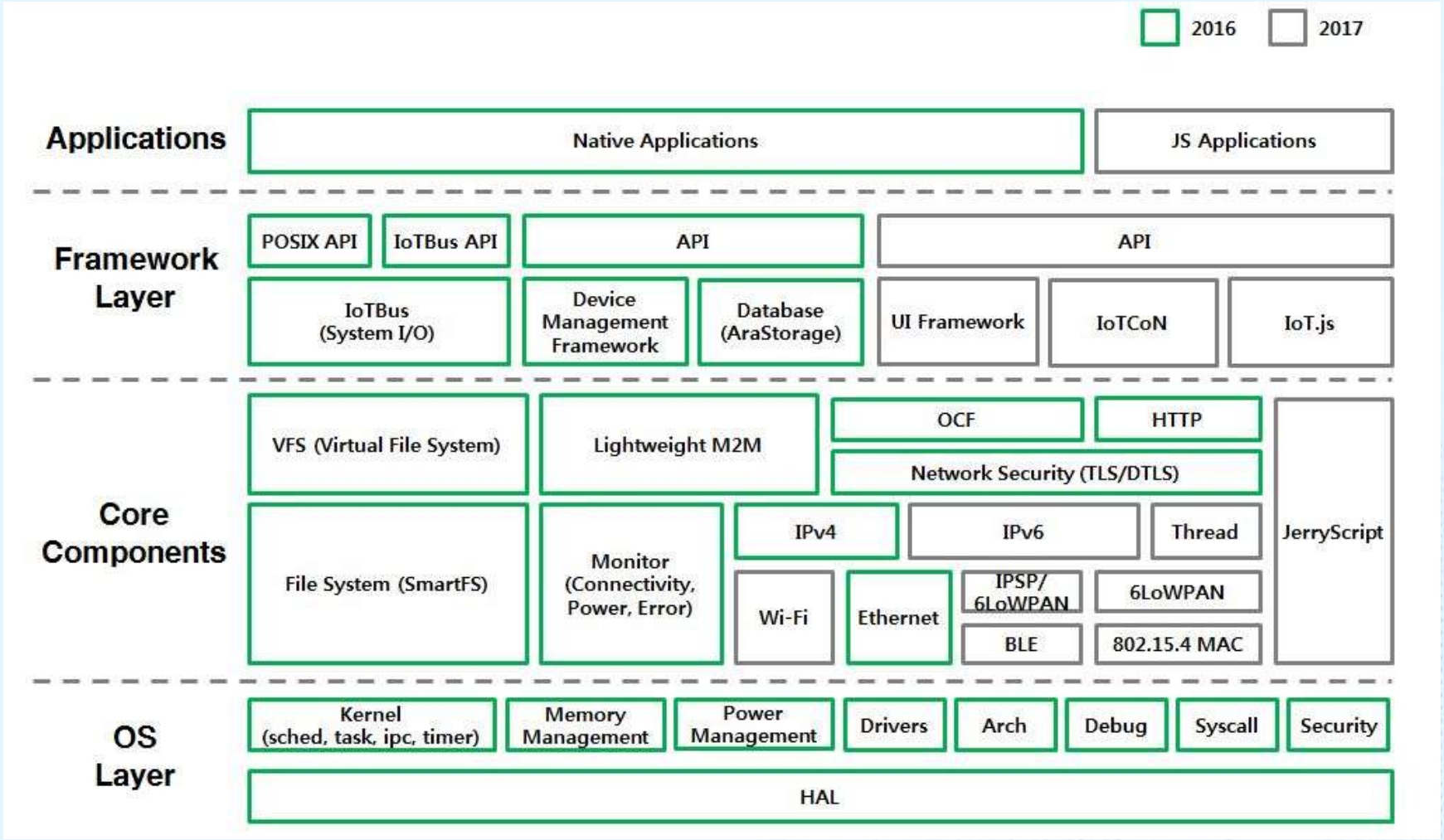
- NuttX RTOS
  - Projects using NuttX
    - Thingsee IoT development device
    - Autopilot PX4 is using NuttX to control a variety of autonomous platforms
    - Pixhawk: An Advanced, User-Friendly Autopilot
    - Biffer Board supports NuttX besides many other RTOS
    - OsmocomBB is using NuttX to develop an operating system for cell phones
    - A homebrew steer-by-wire system implemented using NuttX
    - Motorola Moto Z
    - Sony is using NuttX in their audio processors
    - ***Samsung announces Tizen RT based on NuttX RTOS***

# Tizen RT Outlook

## Tizen RT Overview

# Tizen RT Outlook

## Tizen RT Architecture

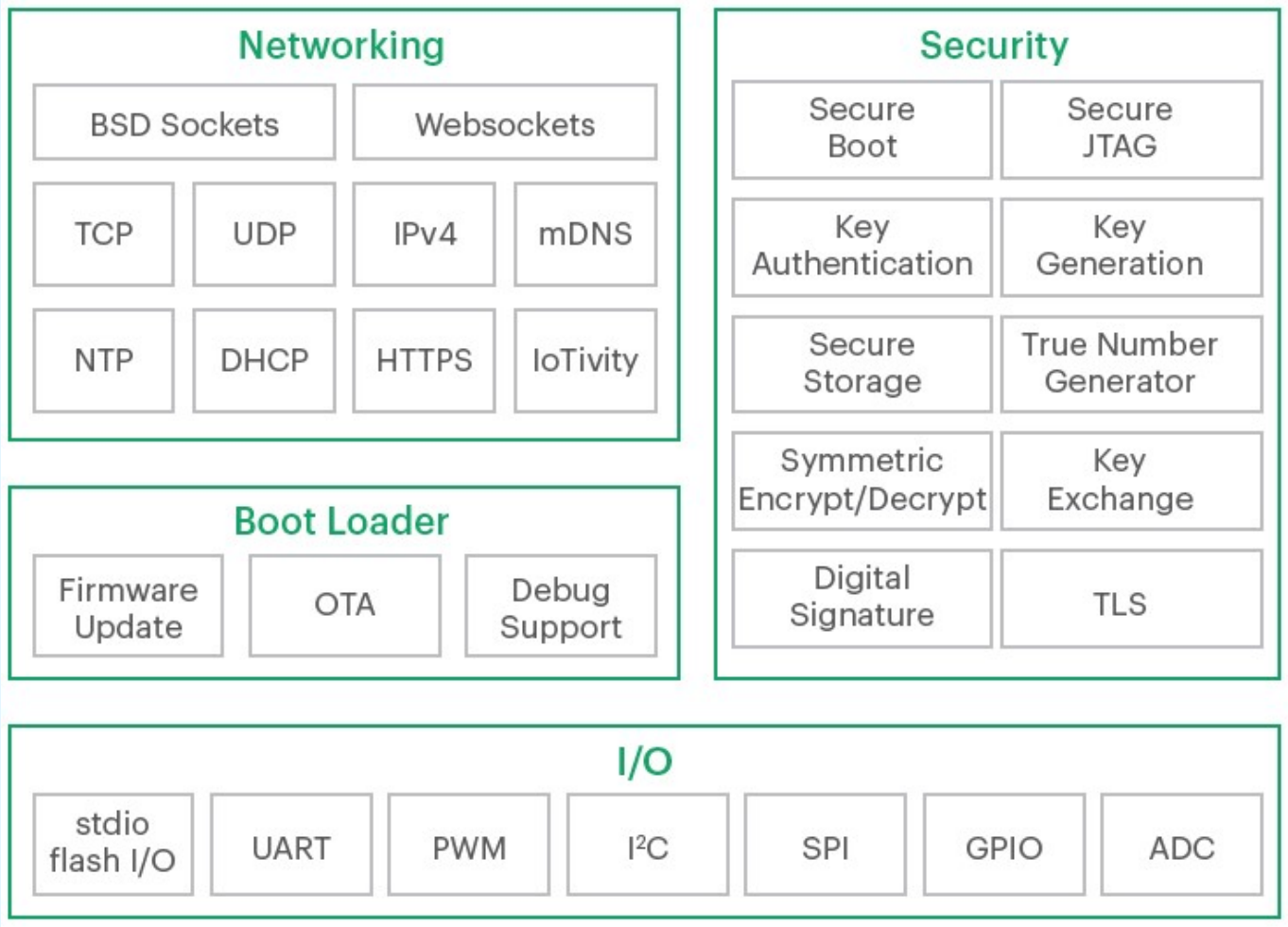


# Tizen RT Outlook

- Tizen RT Features
  - Real time: Tasks, threads, queues, mutex, semaphore, signal
  - Time: RTC, date/time, timer, sleep
  - Internet: DHCP, NTP client, DNS client, mDNS, BSD sockets, Websockets
  - Services: Web client/server, WiFi management, cJSON
  - Libc Compatibility: Flash-based stdio, stdlib, string, unistd, time libraries
  - Encryption: AES 128/256, RSA 1024/2048, ECC BP/NIST 192/224/256/384/512
  - Authentication: HMAC 128/256, certificate
  - Certifate Storage: Secure Flash storage
  - Firmware Integrity: Secure boot and JTAG protection

# Tizen RT Outlook

## Tizen RT Block Diagram



# Tizen RT Sub-systems

- IoT Data Management
  - File System Support
    - Proc File System for debug
    - ROM File System for read only data
    - SmartFS for flash file system (wear-leveling, bad sector mgmt, transaction logging based journaling)
    - MTD and MTD Partition
    - VFS (provide a common interface set in the form of POSIX API)
  - Database(AraStorage) Support
    - AraStorage (Lightweight database): manipulate collected sensor data with SQL-compatible interface
      - b+ tree based indexing algorithm
      - Cursor structore to improve usability for application layer

# Tizen RT Sub-systems

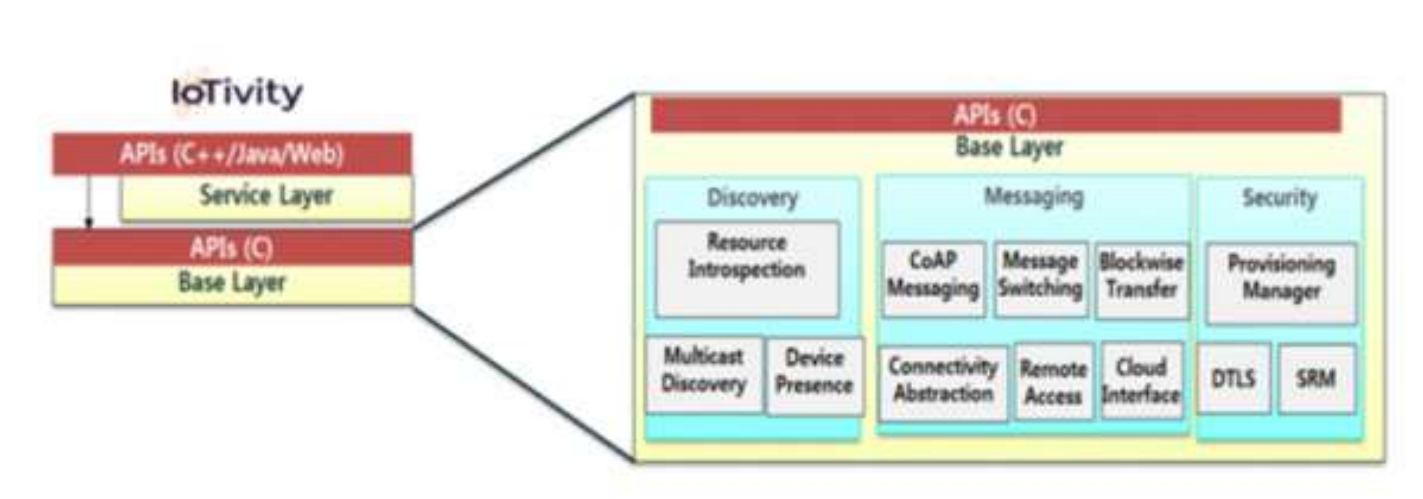
- Device Management
  - OMA based Lightweight M2M (LWM2M) protocol for Device Management
  - Application layer communication protocol between an LWM2M server and an LWM2M client
  - LWM2M Interfaces
    - Bootstrap, Register, Device Management and Service Enablement, Information Reporting
  - LWM2M Protocol Stack
    - wakaama is ported (https://github.com/eclipse/wakaama)
      * Wakaama (formerly liblwm2m) is an implementation of the Open Mobile Alliance's LightWeight M2M protocol (LWM2M).
    - CoAP, DTLS Security, UDP

# Tizen RT Sub-systems

- IP Network
  - TCP/UDP/IPv4 protocols, LWIP is ported on Tizen RT
  - Regarding IPv6, uIP-based stack is already implemented and is granted IPv6 Ready Logo from IPv6 Forum
  - Transition between IPv4 and IPv6 is also required
  - Suppose that sensor devices are equipped with only IPv6 over IEEE 802.15.4 or IPSP/BLE
  - Tizen RT will get ready to fit into these relaying IoT devices by implementing the transition functions between IPv4 and IPv6

# Tizen RT Sub-systems

- IoTivity
  - Tizen RT 1.0 supports IoTivity 1.1.0 now
  - IoTivity 1.2 Base layer Support (OCF 1.0 Base Layer Ready)
    - Tizen RT supports IoTivity base layer for constrained device communication in IoT world.
    - It supports IoTivity 1.2 release as base code with OCF 1.0 spec ready.
    - Currently IP transport(TCP/UDP over Wi-Fi) is supported in Connectivity Abstraction(CA) Layer.



# Tizen RT Sub-systems

- IoTivity
  - Features in IoTivity 1.2 of Tizen RT

| Component (Base layer) | Feature | Description |
|---|---|---|
| Discovery | Multicast Discovery, Device Presence | Discover resource, check device presence |
| | Resource Introspection | Resource Type/property management |
| Messaging | CoAP Messaging | Transmit message between devices |
| | Message switching | Tizen RT does not support message switching |
| | Connectivity Abstraction | Currently Wi-Fi transport support. Feature support for BT, BLE, NFC etc. |
| | Block-wise transfer | Block data transfer(More than 1KB data) |
| | CoAP over TCP | Reliable transmission. It can be used for messaging between device and cloud. |
| Security | DTLS | secure channel with data encryption for UDP. |
| | Security Resource Manager | Access control mechanism |
| | Security Provisioning Manager | transmit credential for authentication |

# Tizen RT Sub-systems

- IoTivity
  - Features support in IoTivity 1.2
    - Tizen RT supports the IoTivity 1.2.0 base layer stack (csdk layer) with Wi-Fi transport.
    - It supports resource creation and publish for resource discovery. (resource registration, discovery , update, delete).
    - It supports Device to Device Communication with UDP over Secured DTLS channel.
    - Wi-Fi transport over IPv4 is supported in Tizen 4.0.
    - It supports CoAP over TCP for communicating with the IoT cloud. (resource registration, discovery , update, delete).
    - TLS support for the TCP to enable security for cloud communication.
    - Support presence for server side and provide presence callback for client side

# Tizen RT Sub-systems

- IoTivity
  - Features support in IoTivity 1.2
    - Onboarding support (Wi-Fi provisioning) for new devices to enable easy setup feature on to the network.
    - Cloud provisioning to connect the device to the cloud and to publish the resource on to it.
    - Keep-alive mechanism to keep the cloud session active with CoAP over TCP.
    - Direct pairing support for credential delivery to transfer the ownership of the device for easy setup.
    - This supports multiple ownership transfer and multiple ownership structure in Security Resource Model.
    - Message-oriented communication interface for the cloud. This interface can be used for a publish/subscribe based information exchange. A resource model for a CoAP-based message broker will be provided.

# Tizen RT Sub-systems

- IoT Bus Framework
  - GPIO
    - Provides functions to control generic pins. It can be configured to be input or output because GPIO pins have no predefined purpose
  - I2C
    - Provides functions to read values of I2C devices or write command to I2C devices. It is typically used to connect sensor devices or for intra board communications
  - SPI
    - Provides functions to communicate with SPI devices. It supports synchronous serial communication interfaces used for a short distance communication. Full duplex modes using a master-slave architecture with a single master is also served

# Tizen RT Sub-systems

- IoT Bus Framework
  - PWM
    - Provides functions to get/set duty cycles and periods of PWM devices. It is typically used to control servo motors or LEDs.
  - UART
    - Provides functions to read/write for asynchronous serial communication. And it is usually used in conjunction with communication standards such as RS-232, RS-422 or RS-485.

# Tizen RT Sub-systems

- Device Management Framework
  - Configuration
    - LWM2M client will be configured with a set of parameters that include LWM2M bootstrap server address, bootstrap server port, and the LWM2M session lifetime
    - If a direct connection to LWM2M server is preferred, then the client will be configured with the LWM2M server address and port information
  - Temporary halt and resumption
    - Wireless links, especially in indoor deployments are prone to intermittent failures, and may momentarily halt an on-going LWM2M session
    - Taking this into account, the Device Management Framework should gracefully close any all LWM2M sessions with their respective servers, and also logically resume the sessions once the wireless link is restored

# Tizen RT Sub-systems

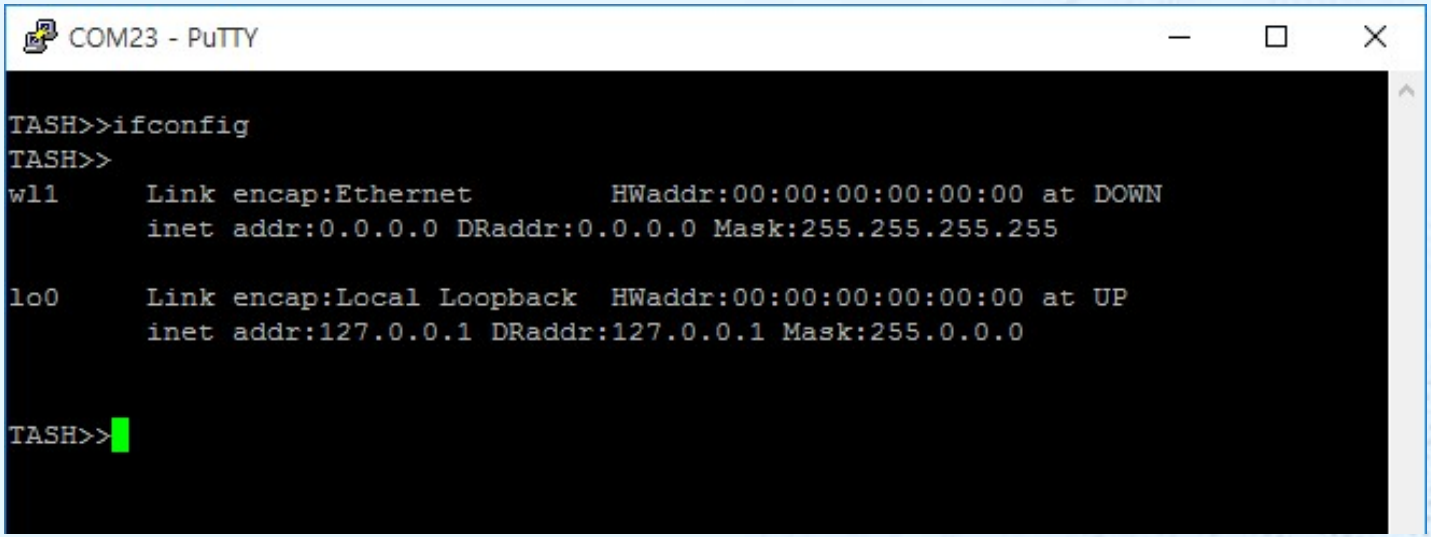
- Device Management Framework
  - Support for Multiple Servers
    - The LWM2M specification allows multiple servers to perform Device Management with a registered LWM2M client
    - To this end, the framework should facilitate the seamless addition of LWM2M server information
  - Device Management Services
    - **Connectivity Monitoring**: which relates to details such as client IP address, network type, signal strength, and effective data rate.
    - **Power Monitoring**: which relates to available power states of a device, its current power state, and the time spent in different power states.
    - **Error reporting**: which relates to out of memory conditions, and also temporary loss of wireless connection.
    - **S/W Update**: which relates to querying a firmware repository for updates, version checking, downloading and installing the firmware package.

# Command Line Interface(CLI)

- CLI Operations
- CLI Commands
- GPIO Commands (gpio)
- File system Command (smartfs)
- Ramdisk Command (mkrd)
- WiFi Configuration (wifi)

# CLI Operations

- Command Line Interface(CLI) Operation
  - This bash-like shell is available for the simple command line interpreter (CLI) that interacts with the module.
  - Use it to run any of the pre-loaded sample applications (e.g. ping, wifi)



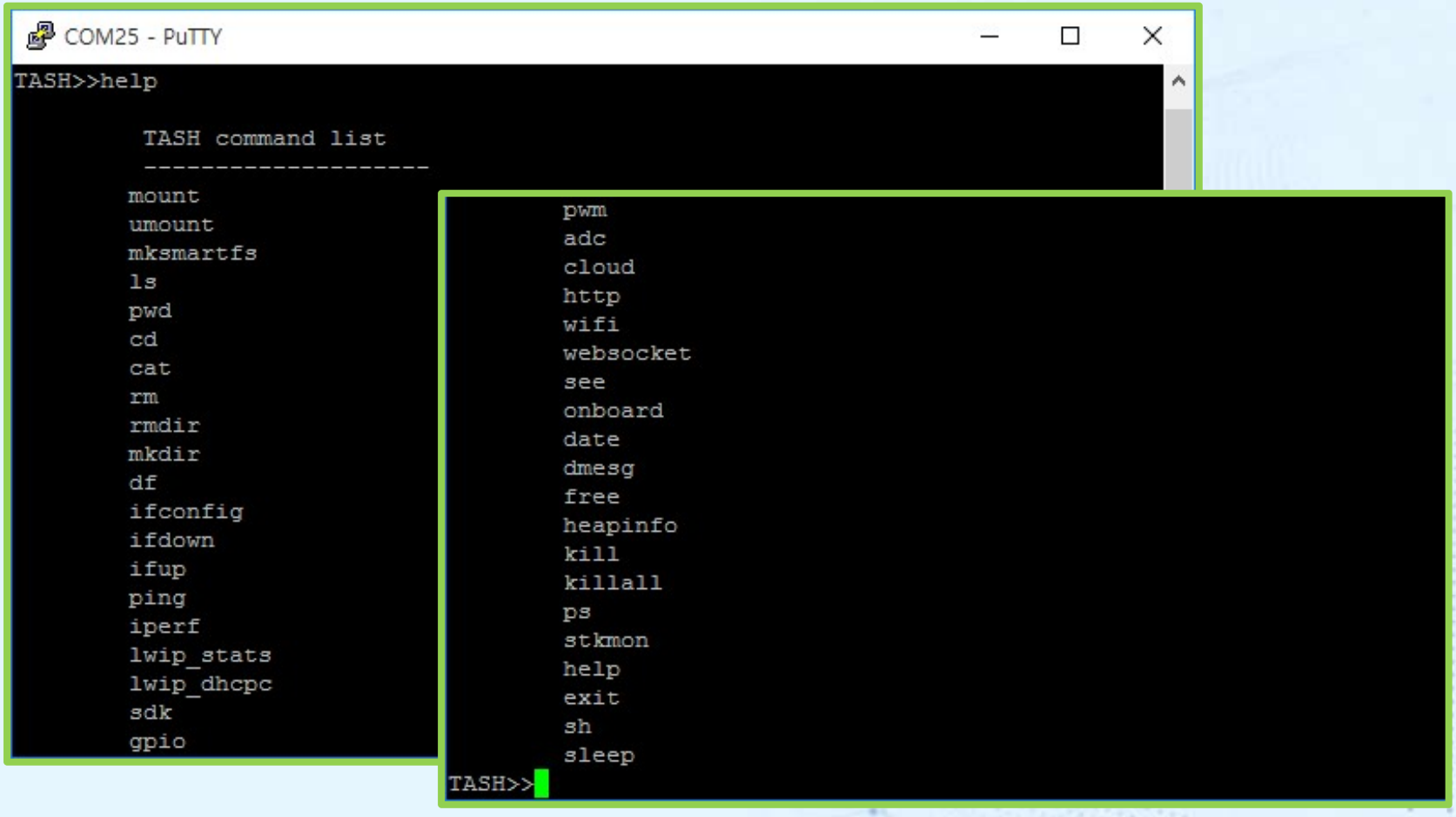
# CLI Operations

- Command Line Interface(CLI) Operation
  - Depending on the build configuration (minimal, typical, extra), the command configuration may vary
  - The configuration between the Initial ARTIK 053 BSP command and the command that updates the binaries through ARTIK IDE is different

| COMMAND | CATEGORY |
|---|---|
| wifi iperf ping<br>ifconfig ifup ifdown | network operations |
| mkdir rmdir pwd cd<br>ls cat<br>rm<br>mount umount | file system operations<br>(note that cp and mv are not present) |
| ps kill killall heapinfo<br>free stkmon sleep exit sh | process-related commands |
| setenv unsetenv getenv date | environment variable access |
| mkrd mksmartfs* | RAM disk / flash file system creation |

# CLI Commands

- Commands List
  - TASH>> help

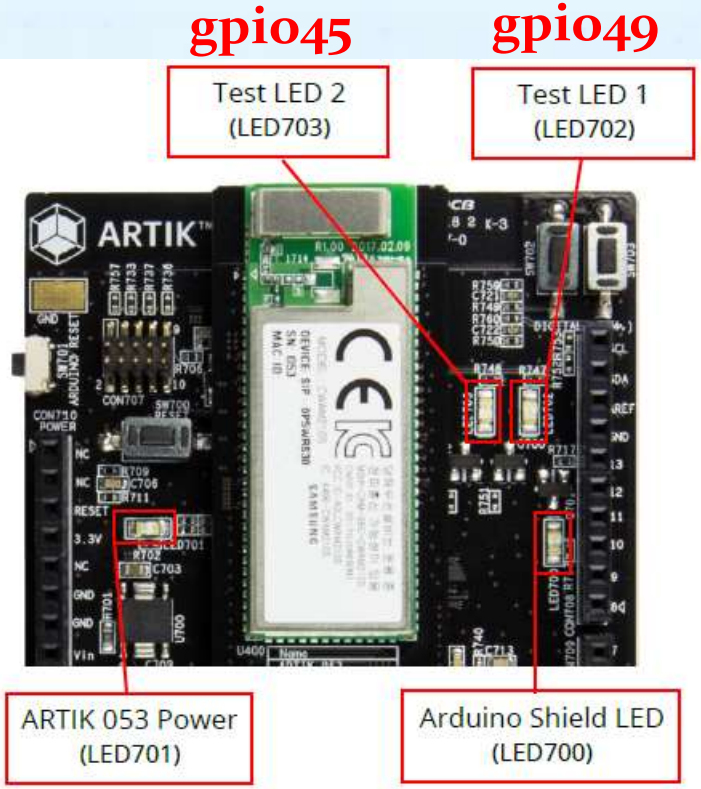


# GPIO Command - gpio

- Internal Buttons & LEDs
  - LEDs

LED descriptions

| LED Name | Description | Comment |
|---|---|---|
| LED700 | Arduino Shield LED | This red LED is illuminated from pin 6 of Arduino Shield 1 Header CON708. Is used by Arduino Shield applications. See *Table 7* for more details. |
| LED701 | ARTIK 053 Power | This red LED is illuminated whenever Starter Kit board power is applied. |
| LED702 | Test LED 1 | This blue LED is connected to signal XGPIO20. Can be used for application specific output. |
| LED703 | Test LED 2 | This red LED is connected to signal XGPIO16. Can be used for application specific output. |



# GPIO Command - gpio

- ARTIK 053 GPIO Pin Map
  - tinyara/os/arch/arm/src/artik053/src/artik053_boot.c

```
} pins[] = {
    { 30, GPIO_INPUT | GPIO_PULLDOWN | GPIO_PORTG0 | GPIO_PIN1 }, /* ARTIK_A053_XGPIO1 */
    { 31, GPIO_INPUT | GPIO_PULLDOWN | GPIO_PORTG0 | GPIO_PIN2 }, /* ARTIK_A053_XGPIO2 */
    { 32, GPIO_INPUT | GPIO_PULLDOWN | GPIO_PORTG0 | GPIO_PIN3 }, /* ARTIK_A053_XGPIO3 */
    { 37, GPIO_INPUT | GPIO_PULLDOWN | GPIO_PORTG1 | GPIO_PIN0 }, /* ARTIK_A053_XGPIO8 */
    { 38, GPIO_INPUT | GPIO_PULLDOWN | GPIO_PORTG1 | GPIO_PIN1 }, /* ARTIK_A053_XGPIO9 */
    { 39, GPIO_INPUT | GPIO_PULLDOWN | GPIO_PORTG1 | GPIO_PIN2 }, /* ARTIK_A053_XGPIO10 */
    { 40, GPIO_INPUT | GPIO_PULLDOWN | GPIO_PORTG1 | GPIO_PIN3 }, /* ARTIK_A053_XGPIO11 */
    { 41, GPIO_INPUT | GPIO_PULLDOWN | GPIO_PORTG1 | GPIO_PIN4 }, /* ARTIK_A053_XGPIO12 */
    { 42, GPIO_INPUT | GPIO_PULLDOWN | GPIO_PORTG1 | GPIO_PIN5 }, /* ARTIK_A053_XGPIO13 */
    { 43, GPIO_INPUT | GPIO_PULLDOWN | GPIO_PORTG1 | GPIO_PIN6 }, /* ARTIK_A053_XGPIO14 */
    { 44, GPIO_INPUT | GPIO_PULLDOWN | GPIO_PORTG1 | GPIO_PIN7 }, /* ARTIK_A053_XGPIO15 */
    { 45, GPIO_INPUT | GPIO_PULLDOWN | GPIO_PORTG2 | GPIO_PIN0 }, /* ARTIK_A053_XGPIO16 */
    { 46, GPIO_INPUT | GPIO_PULLDOWN | GPIO_PORTG2 | GPIO_PIN1 }, /* ARTIK_A053_XGPIO17 */
    { 47, GPIO_INPUT | GPIO_PULLDOWN | GPIO_PORTG2 | GPIO_PIN2 }, /* ARTIK_A053_XGPIO18 */
    { 48, GPIO_INPUT | GPIO_PULLDOWN | GPIO_PORTG2 | GPIO_PIN3 }, /* ARTIK_A053_XGPIO19 */
    { 49, GPIO_INPUT | GPIO_PULLDOWN | GPIO_PORTG2 | GPIO_PIN4 }, /* ARTIK_A053_XGPIO20 */
    { 50, GPIO_INPUT | GPIO_PULLDOWN | GPIO_PORTG2 | GPIO_PIN5 }, /* ARTIK_A053_XGPIO21 */
    { 51, GPIO_INPUT | GPIO_PULLDOWN | GPIO_PORTG2 | GPIO_PIN6 }, /* ARTIK_A053_XGPIO22 */
    { 52, GPIO_INPUT | GPIO_PULLDOWN | GPIO_PORTG2 | GPIO_PIN7 }, /* ARTIK_A053_XGPIO23 */
    { 53, GPIO_INPUT | GPIO_PULLDOWN | GPIO_PORTG3 | GPIO_PIN0 }, /* ARTIK_A053_XGPIO24 */
    { 54, GPIO_INPUT | GPIO_PULLDOWN | GPIO_PORTG3 | GPIO_PIN1 }, /* ARTIK_A053_XGPIO25 */
    { 55, GPIO_INPUT | GPIO_PULLDOWN | GPIO_PORTG3 | GPIO_PIN2 }, /* ARTIK_A053_XGPIO26 */
    { 57, GPIO_INPUT | GPIO_PULLDOWN | GPIO_PORTA0 | GPIO_PIN0 }, /* ARTIK_A053_XEINT0 */
    { 58, GPIO_INPUT | GPIO_PULLDOWN | GPIO_PORTA0 | GPIO_PIN1 }, /* ARTIK_A053_XEINT1 */
    { 59, GPIO_INPUT | GPIO_PULLDOWN | GPIO_PORTA0 | GPIO_PIN2 }, /* ARTIK_A053_XEINT2 */
};
```

# GPIO Command - gpio

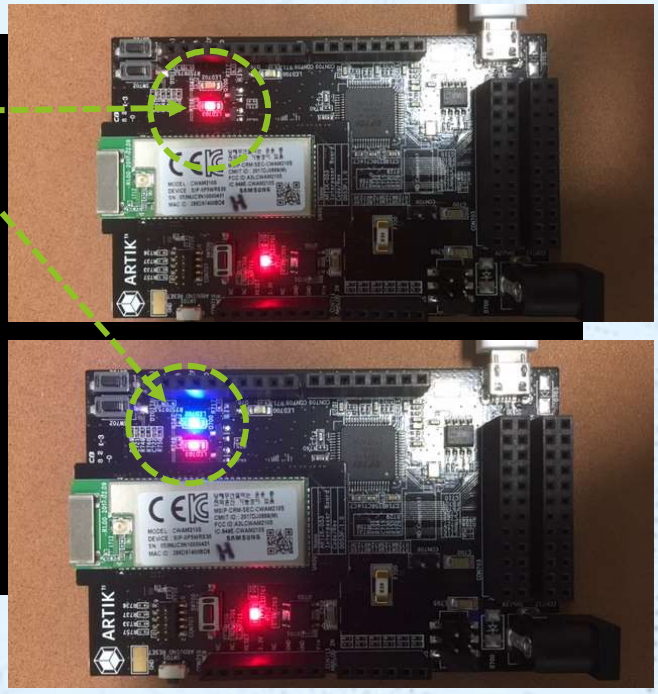
- Turn on a GPIO pin to light an LED
  - On the pre-loaded firmware, you'll find a test app called gpio that you can use to set GPIO pin function
    - TASH>> gpio write 45 1
    - TASH>> gpio write 49 1

where is a 'gpio' command?

```
TASH>>gpio write 45 1
TASH>>gpio write 49 1
TASH>>gpio write 49 0
TASH>>gpio write 45 0
TASH>>
```

Red LED On

Blue LED On



# GPIO Command - gpio

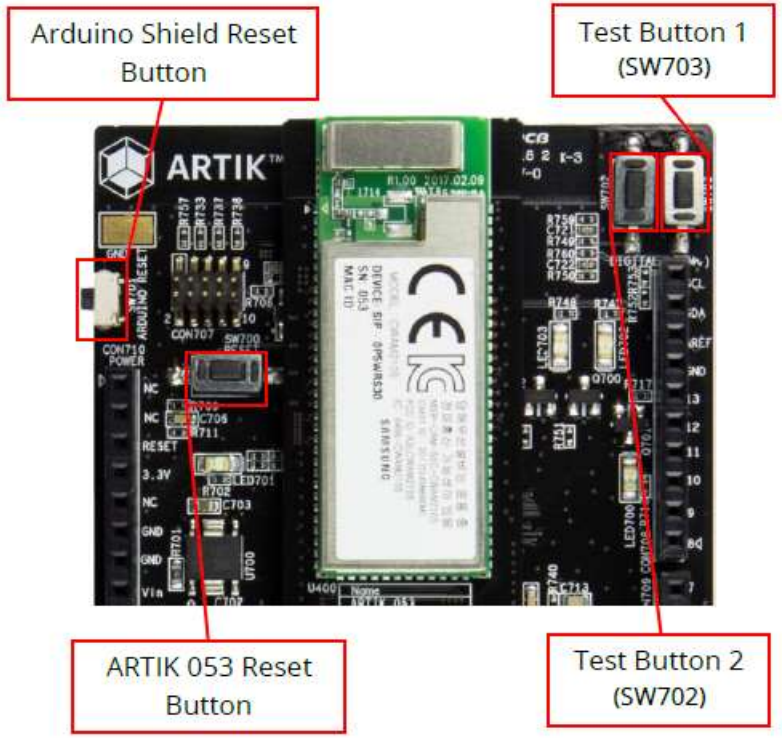
## Internal Buttons & LEDs

### Buttons

Button descriptions

| Button Name | Description | Comment |
|---|---|---|
| SW700 | ARTIK 053 Reset Button | Connects signal XRESET_N to GND on CON702 (pin 1 and 21). See *Table 17* for more details. |
| SW701 | Arduino Reset Button | A reset signal on Arduino interface. The reset signal is part of CON710. See *Table 9* for more details. |
| SW702 | Test Button 2 | When pressed, SW702 connects signal XGPIO13 to GND. Can be used for application specific input. |
| SW703 | Test Button 1 | When pressed, SW703 connects signal XGPIO15 to GND. Can be used for application specific input. |

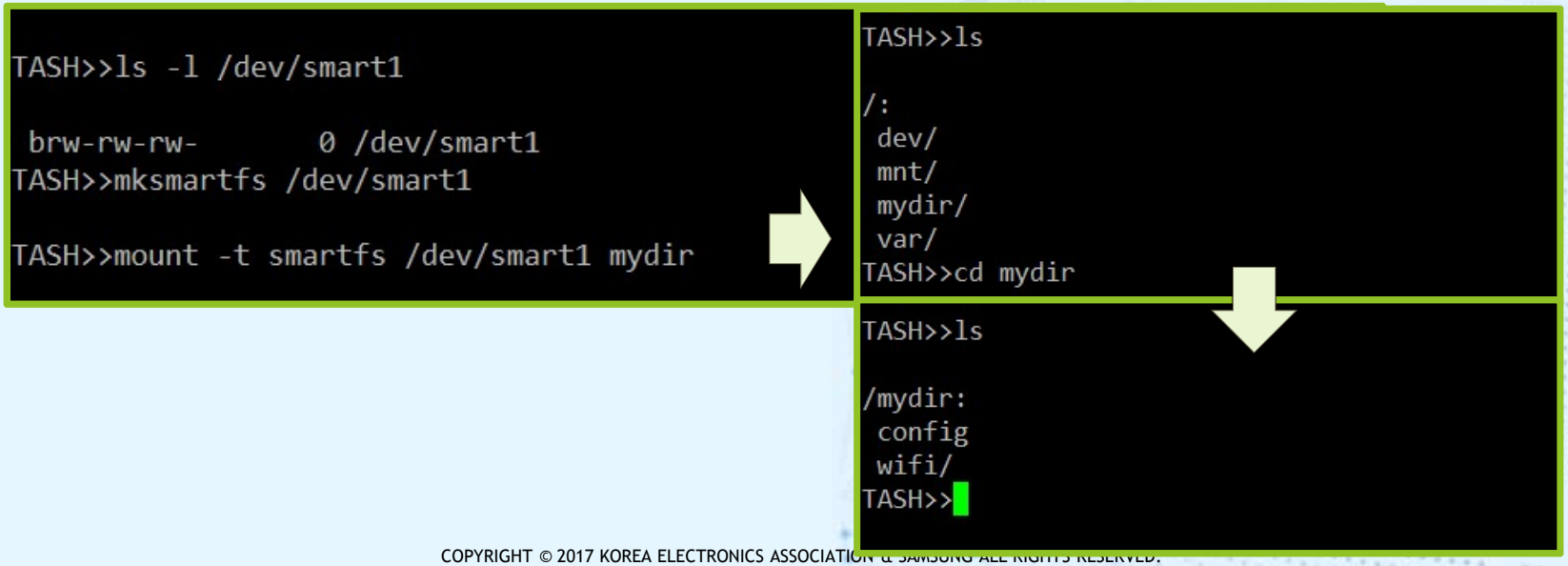
gpio44




gpio42

# File System Command - smartfs

- Flash Drive Setup (smartfs)
  - Device file for Flash drive:
    - /dev/smart1
  - Format:
    - TASH>> mksmartfs /dev/smart1
  - Mount:
    - TASH>> mount -t smartfs /dev/smart1 mydir



# Ramdisk Command - mkrd

- RAM Disk Drive Setup (Not supported on Initial BSP)
  - Command:
    - mkrd
    - TASH>> mkrd 1

```
TASH>>ls /dev/ram0
ls : stat failed
TASH>>mkrd 1
TASH>>tash_mkrd: RAMDISK at 2060800

TASH>>ls /dev/ram0
 /dev/ram0
TASH>>
```

/dev/ram0 file is created

# WiFi Configuration - wifi

- WiFi Setup (Manually)
  - Start the Tizen RT version of the 'wpa_supplicant' service in client (STA) mode
    - TASH>> wifi  stop
    - TASH>> wifi  startsta

```
TASH>>wifi stop
TASH>>wifi startsta

Starting supplicant as daemon...
```

Stop AP Mode

# WiFi Configuration - wifi

- WiFi Setup (Manually)
  - Connect AP
    - TASH>> wifi connect SSID PASSWORD
    - TASH>> ifconfig

```
TASH>>wifi connect Dr Kook 1357924680abc

Connected to Dr_Kook
TASH>>ifconfig

wl1     Link encap:Ethernet        HWaddr:28:6d:97:40:0b:d6 at UP
        inet addr:192.168.10.1 DRaddr:192.168.10.1 Mask:255.255.255.0

lo0     Link encap:Local Loopback  HWaddr:00:00:00:00:00:00 at UP
        inet addr:127.0.0.1 DRaddr:127.0.0.1 Mask:255.0.0.0

TASH>>
```

# WiFi Configuration - wifi

- WiFi Setup (Manually)
  - WiFi Commands (Updated Image)
    - TASH>> help

```
TASH>>help
         TASH command list
         -------------------
artikwifi       cat             cd              date
exit            free            getenv          heapinfo
hello           help            ifconfig        ifdown
ifup            kill            killall         ls
mkdir           mkrd            mksmartfs       mount
ping            ps              pwd             reboot
rm              rmdir           setenv          sh
sleep           stkmon          umount          unsetenv


If you want to run an ASYNC command with specific priority and stacksize
use "setenv CMD_PRI" or "CMD_STACK"
TASH>>
```

# WiFi Configuration - wifi

- WiFi Setup (Manually)
  - WiFi Commands (Updated Image)
    - TASH>> artikwifi

```
TASH>>artikwifi
*************************************************************
* Samsung System LSI wifi application for t20               *
*************************************************************
Available commands:
   join
   leave
   scan
   startsta
   startap
   stop
   status
   opmode
   country
   txpower
   auto
   help
use the command help to get extended help about arguments for the
different commands
*************************************************************
TASH>>
```

# WiFi Configuration - wifi

- WiFi Setup (Manually)
  - Start STA Mode (Updated Image)
    - TASH>> artikwifi  startsta

```
TASH>>artikwifi startsta
**************************************************************
* Samsung System LSI wifi application for t20                *
**************************************************************
Link call back handles registered - per default!
Starting supplicant in foreground...
STA mode started successfully
TASH>>
```

# WiFi Configuration - wifi

- WiFi Setup (Manually)
  - AP Scanning (Updated Image)
    - TASH>> artikwifi  scan

```
TASH>>artikwifi scan
*****************************************************************
* Samsung System LSI wifi application for t20                   *
*****************************************************************
Link call back handles registered - per default!
Successfully started scan...waiting for result!
Scan Result - networks:
          BSSID                 RSSI     SECURITY
CH     SSID
      1) 90:9f:33:a2:72:5a     -26      [WPA2-PSK+AES]
12     Dr_Kook
      2) 06:b0:6d:12:6b:34     -72      [WPA2-PSK+AES]
1      KEA_Office_WiFi
      3) 00:27:1c:6a:81:00     -74      [WPA-PSK+TKIP]
1      \x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
      4) 00:26:66:ce:89:ac     -76      [WPA2-PSK+AES]
11     taekyung2.4
      5) 00:13:60:67:b8:c0     -76      [WPA-PSK+TKIP]
7      SDSS_A21
      6) 00:14:69:f3:11:b0     -78      [WPA-PSK+TKIP]
4      SDSS_A21
      7) 00:3a:98:68:1b:50     -77      [WEP]
8      2F-KEA
      8) 04:bf:6d:12:6b:34     -73      [NONE]
1      KEA Free WiFi
TASH>>
```

# WiFi Configuration - wifi

- WiFi Setup (Manually)
  - Connect an AP (Updated Image)
    - TASH>> artikwifi  join

```
TASH>>artikwifi join Dr_Kook 1357924680abc
*************************************************************
* Samsung System LSI wifi application for t20               *
*************************************************************
Link call back handles registered - per default!
Joining network Dr_Kook
Security: wpa2_aes
Passphrase: 1357924680abc
Connected to network: bssid: 90:9f:33:a2:72:5a, ssid: Dr_Kook
Successfully joined the network with SSID Dr_Kook
TASH>>
```

# WiFi Configuration - wifi

- WiFi Setup (Manually)
  - WiFi Status (Updated Image)
    - TASH>> artikwifi  status

```
TASH>>artikwifi status
**************************************************************
* Samsung System LSI wifi application for t20                *
**************************************************************
Connected to network: bssid: 90:9f:33:a2:72:5a, ssid: Dr_Kook
Link call back handles registered - per default!
Wi-Fi status - Connected : 1
TASH>>
```

# WiFi Configuration - wifi

- WiFi Setup (Manually)
  - IP Address (Updated Image)
    - TASH>> ifconfig

```
TASH>>ifconfig
wl1     Link encap: 28:6d:97:40:0b:d6   RUNNING: UP
        inet addr: 0.0.0.0      Bcast: 0.0.0.0  Mask: 255.255.255.255   MTU: 150
0

lo0     Link encap: 00:00:00:00:00:00   RUNNING: UP
        inet addr: 127.0.0.1    Bcast: 127.0.0.1        Mask: 255.0.0.0 MTU: 0

TASH>>
```

# WiFi Configuration - wifi

- WiFi Setup (Manually)
  - IP Address (Updated Image)
    - TASH>> ifconfig wl1 dhcp

```
TASH>>ifconfig wl1 dhcp
IP address 192.168.14.8
Netmask 255.255.255.0
Gateway 192.168.14.1
Default DNS 210.94.0.73
TASH>>ifconfig
wl1     Link encap: 28:6d:97:40:0b:d6   RUNNING: UP
        inet addr: 192.168.14.8 Bcast: 192.168.14.8     Mask: 255.255.255.0    M
TU: 1500

lo0     Link encap: 00:00:00:00:00:00   RUNNING: UP
        inet addr: 127.0.0.1    Bcast: 127.0.0.1        Mask: 255.0.0.0 MTU: 0

TASH>>
```

# WiFi Configuration - wifi

- WiFi Setup (Manually)
  - Configuration of DNS Address
    - Source file
      apps/system/utils/netcmd.c
      cmd_ifconfig()

# WiFi Configuration - wifi

- WiFi Setup (Manually)
  - Ping Test
    - TASH>> ping

```
TASH>>ping gnu.org
PING gnu.org (gnu.org) 60(88) bytes of data. count(10)
--- gnu.org ping statistics ---
10 packets transmitted, 0 received, -Infinity% packet loss,
TASH>>ping 192.168.14.1
PING 192.168.14.1 (192.168.14.1) 60(88) bytes of data. count(10)
 60 bytes from 192.168.14.1: icmp_seq=1 ttl=255 time=20 ms
 60 bytes from 192.168.14.1: icmp_seq=2 ttl=255 time=10 ms
 60 bytes from 192.168.14.1: icmp_seq=3 ttl=255 time=10 ms
 60 bytes from 192.168.14.1: icmp_seq=4 ttl=255 time=10 ms
 60 bytes from 192.168.14.1: icmp_seq=5 ttl=255 time=10 ms
 60 bytes from 192.168.14.1: icmp_seq=6 ttl=255 time=10 ms
 60 bytes from 192.168.14.1: icmp_seq=7 ttl=255 time=10 ms
 60 bytes from 192.168.14.1: icmp_seq=8 ttl=255 time=10 ms
 60 bytes from 192.168.14.1: icmp_seq=9 ttl=255 time=10 ms
 60 bytes from 192.168.14.1: icmp_seq=10 ttl=255 time=10 ms
--- 192.168.14.1 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss,
TASH>>
```