SAMSUNG
ARTIK™ Modules

Lab 2: Develop with ARTIK SDK

Samsung Training Lab

TABLE OF CONTENTS

VERSION HISTORY

Revision	Date	Description	Maturity
V1.0	June 1, 2018	First release	Released
V2.0	June 17, 2018	Update content based on latest SDK 1.7.1 release	Review

Objective

Develop for ARTIK530s by using ARTIK SDK.

Prerequisites

Complete Lab 1, **Getting Started with the ARTIK Development Board**.

Goals

Control I/Os programmatically.

ARTIK Development Board Overview

Figure 1 below details the parts and layout of the ARTIK Development Board:

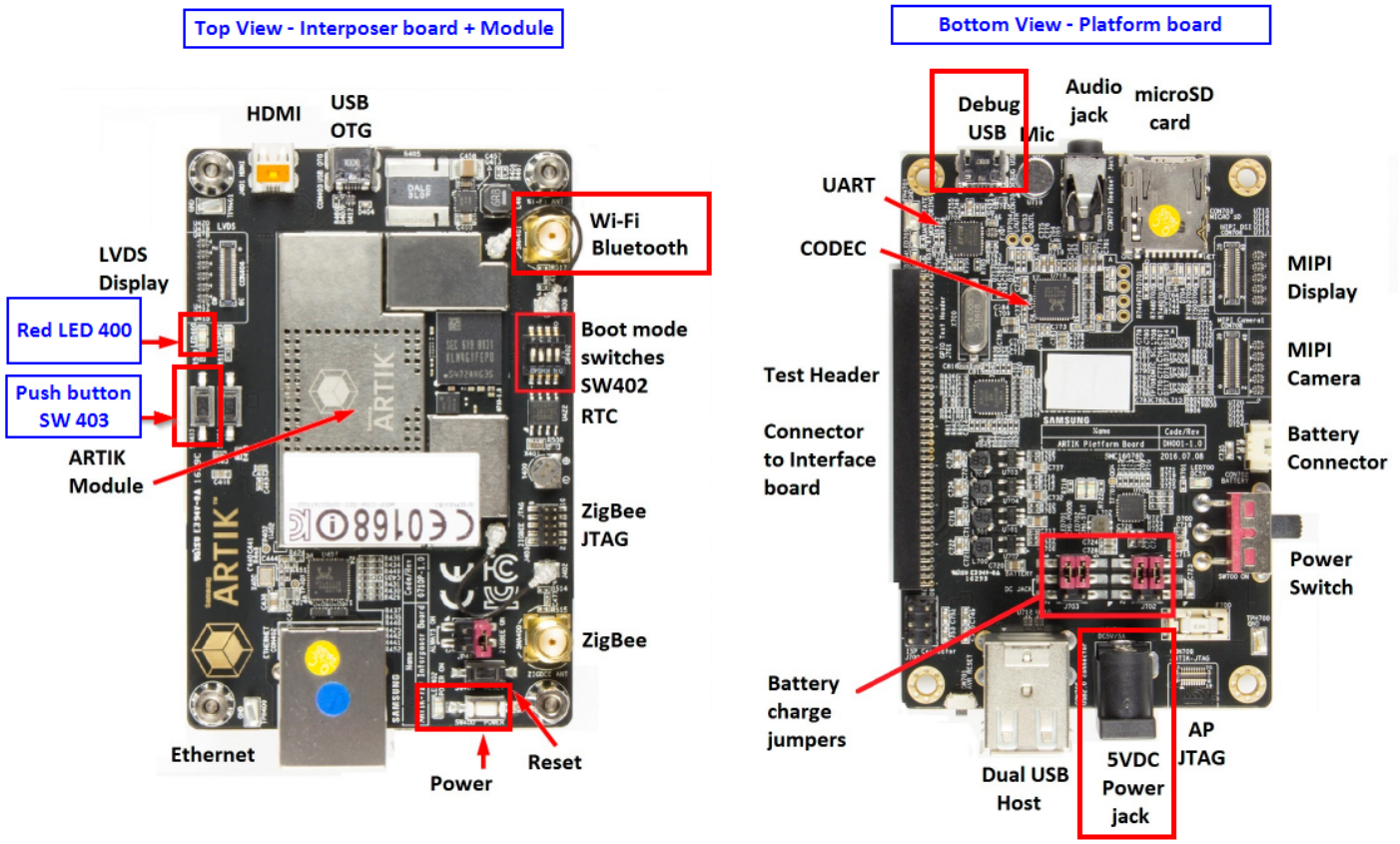


Figure 1. ARTIK Development Board Overview

PART 1: GET ARTIK ONLINE

CONFIGURE WIFI OR ETHERNET ON ARTIK MODULE

If you plugged an Ethernet cable into your ARTIK530s Ethernet port, please move on to step 5.

1. Wi-Fi is managed by the wpa client service. The following configuration needs to be done only once. Subsequently, Wi-Fi should automatically come up after boot.

2. Configure the ARTIK Development Board to connect to a Wi-Fi access point by executing the following command on the terminal shell:

   NOTE: Substitute "***<SSID>***" and "***<password>***" in the command below with the proper SSID and password given to you by your Lab presenter.

   ```
   wpa_passphrase <SSID> <password> >> /etc/wpa_supplicant/wpa_supplicant.conf
   ```

3. Enable and restart the wpa client service, by typing the following command on the shell:

   ```
   systemctl restart wpa_supplicant
   ```

4. Restart the dhcp client to get an IP address, by typing the following command on the shell:

   ```
   dhclient wlan0
   ```

   NOTE:

   Each time after reboot, the above command needs to be executed to obtain an IP address from the router

5. Check to see if you receive an IP address on your WiFi adapter, by typing the following command on the shell:

   ```
   ifconfig wlan0
   ```

   You should see something similar to the following example:

   ```
   [root@localhost ~]# ifconfig wlan0
   wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
          inet 10.0.1.13  netmask 255.255.255.0  broadcast 10.0.1.255
          ether ec:1f:72:d5:18:25  txqueuelen 1000  (Ethernet)
          RX packets 11  bytes 1218 (1.1 KiB)
          RX errors 0  dropped 9  overruns 0  frame 0
          TX packets 32  bytes 4285 (4.1 KiB)
          TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

   [root@localhost ~]#
   ```

   If you are using Ethernet, use the command below to check your IP address.

   ```
   [root@localhost ~]# ifconfig eth0
   eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.2  netmask 255.255.255.0  broadcast 10.0.0.255
        inet6 2601:647:4e01:7b45:489d:21ff:fe85:6c27  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::489d:21ff:fe85:6c27  prefixlen 64  scopeid 0x20<link>
        ether 4a:9d:21:85:6c:27  txqueuelen 1000  (Ethernet)
   ```

6. Execute the following command to make sure the network connection is working and look for 0% packet loss

```
ping –c 3 www.google.com
```

You should see something similar to the following example:

```
[root@localhost ~]# ping www.google.com
PING www.google.com (216.58.192.4) 56(84) bytes of data.
64 bytes from nuq04s29-in-f4.1e100.net (216.58.192.4): icmp_seq=1 ttl=52 time=26.6 ms
64 bytes from nuq04s29-in-f4.1e100.net (216.58.192.4): icmp_seq=2 ttl=52 time=32.2 ms
64 bytes from nuq04s29-in-f4.1e100.net (216.58.192.4): icmp_seq=3 ttl=52 time=81.7 ms

--- www.google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 26.692/46.919/81.768/24.748 ms
[root@localhost ~]#
```

Note: If the ping command does not succeed as shown above, there may be an issue with the Internet connection to the access point or due to connecting to the wrong access point.

Try the following debugging steps:

a. Make sure that the p2p device is not configured as the default in the routing table. To check, execute the following command:

```
route
```

b. If the "Iface" value on the "default" destination is "p2p", execute the following command:

```
ifconfig p2p0 down
```

c. Comment out any other SSID entries that are in the wireless range in the ***/etc/wpa_supplicant/wpa_supplicant.conf*** file.

d. Make sure there is a gateway entry in the ***/etc/resolv.conf*** file.

IDE/SDK Setup

Please follow our online instruction at https://developer.artik.io/documentation/artik/getting-started/prepare-ide.html.

File transfer from host to ARTIK

This section describes the steps to transfer a file from host development environment (PC or Mac) to an ARTIK board. File transfer from host to ARTIK can be done through SCP application using wireless LAN **in the same network**

1. For Mac® and Linux® systems:

   Using the default scp client application, from the terminal program send the file to the board as mentioned below,

```
scp file_to_send root@ARTIK_IP_ADDRESS

scp testfile.c root@10.0.1.13:/root/
```

2. For windows® systems:

   Please install winSCP or Filezilla application in the host PC to send a file to ARTIK

PART 2: BUILD AN ARTIK530S EXAMPLE WITH ARTIK SDK

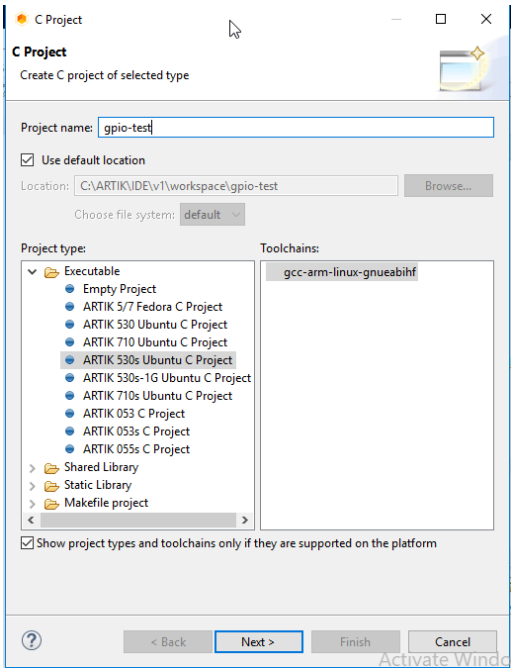
INSTALL ARTIK SDK LIBRARY

Run the command below to install ARTIK SDK library from your ARTIK530s serial console,

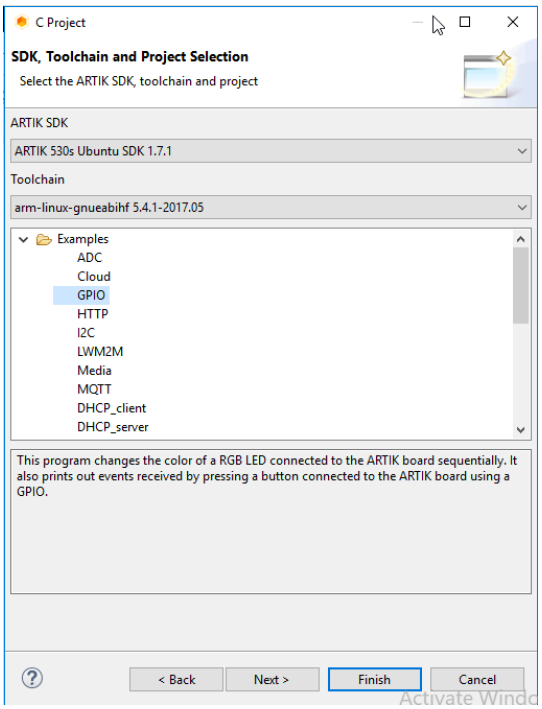
```
[root@artik ~]# apt-get install libartik-sdk-*
…
Setting up libartik-sdk-base (1.8-1) ...
Setting up libartik-sdk-base-dev (1.8-1) ...
Setting up libartik-sdk-bluetooth (1.8-1) ...
Setting up libartik-sdk-bluetooth-dev (1.8-1) ...
Setting up libartik-sdk-connectivity (1.8-1) ...
Setting up libartik-sdk-connectivity-dev (1.8-1) ...
Setting up libartik-sdk-systemio (1.8-1) ...
Setting up libartik-sdk-media (1.8-1) ...
Setting up libartik-sdk-sensor (1.8-1) ...
Setting up libartik-sdk-wifi (1.8-1) ...
…
```

BUILD AN APPLICATION FOR ARTIK530S

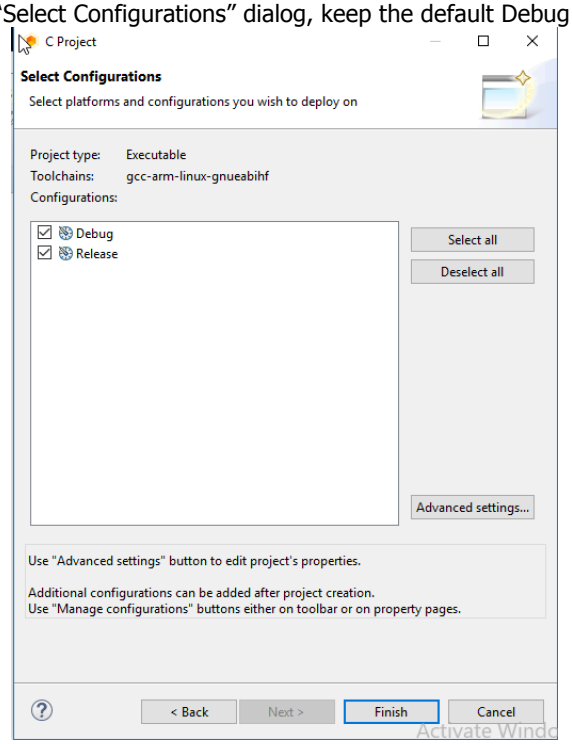
1. Launch your ARTIK IDE, go to File->New->C Project.
2. In the "C Project" dialog, select "ARTIK 530s Ubuntu C Project", and choose "gcc-arm-linux-gnueabihf" as the default toolchain. Name the Project name as "gpio-test". Then, click Next.



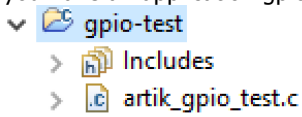
3. In "SDK, Toolchain and Project Selection" dialog, select "ARTIK 530s Ubuntu SDK 1.7.1" as the target SDK version. Click on "GPIO" example, we will use this example as our template to generate an ARTIK530s application. Then, click Next.



4. In "Select Configurations" dialog, keep the default Debug and Release builds settings, and click Finish.



5. Now you have an application gpio-test created in your IDE Project Explorer.



6. Delete existing artik_gpio_test.c by right clicking the file and select Delete.

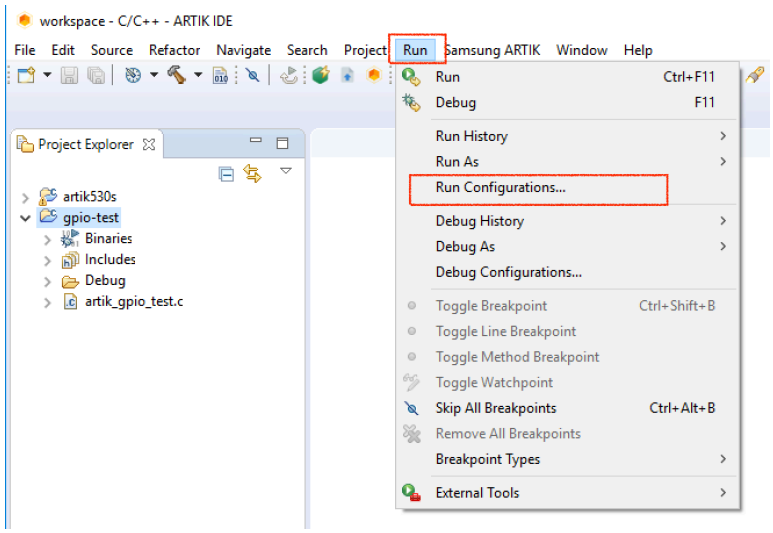
7. Drag and drop the new artik_gpio_test.c to gpio_test project.

8. To build the application, right-click on the application project and select Build Project. This will cross-compile the application for ARTIK530s modules.

SET UP A "RUN CONFIGURATION"

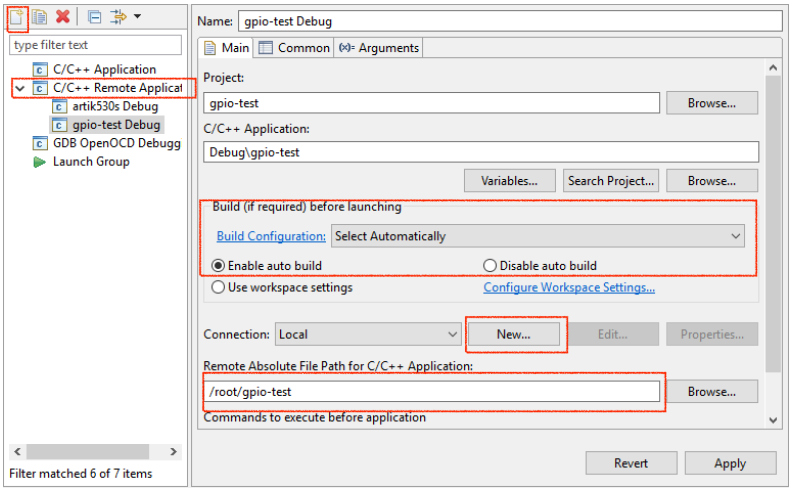
ARTIK IDE communicates with the ARTIK board via WiFi or Ethernet to load, run or debug the application. You need to set up a remote "run configuration" for your board.

1. Click Run, then Run Configurations…

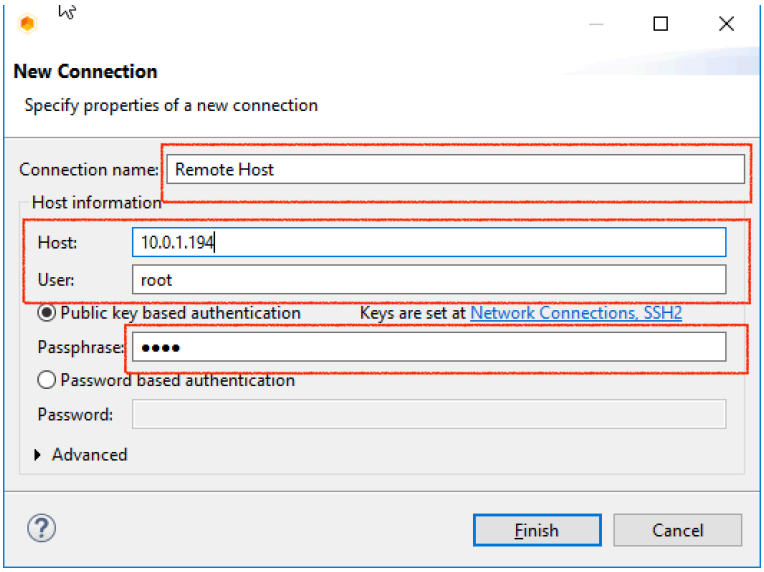


2. Add a C/C++ Remote Application and configure it as follow.

   a. Click on "C/C++ Remote Application" and click the New button to create a new remote connection; enter information like below.



   b. On the "Connection:" line, select New, then select SSH as the connection type. Configure the SSH connection by adding a name, the ARTIK board IP address, and username/password(root/root by default).



c. Apply the new settings.

Run the application from IDE

Right click your project and select Run, this will build the project, generate a binary image and transfer the image to your ARTIK board over SSH.

If required,

1. Select Run >> Run Configurations… to select the target board for your project.
2. Right click on the application project and select Run As >> ARTIK 5/7 Program
3. Observe the output of the application program in the build console window.

You should be able to observe LED400 and LED401 are turned on and off.