Tizen.IoT.Init

Tizen IoTivity를 시작하기 위한 환경 설정

Geunsun Lee

1

Installing Tizen Platform

- Flashing with the IoT Setup Wizard

2

Booting The RPi3

- Connecting a Host PC to RPi3 via UART

3

Building a Tizen project

- Cloning Tizen repositories
- Building the project with Tizen Studio

4

Running the Tizen project on RPi3

- Connecting RPi3 to the network
- Connecting RPi3 to Tizen Studio

Preparation

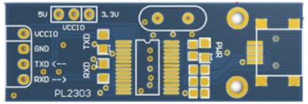
Micro SD Card

Infrared Motion Sensor
(HC-SR501)

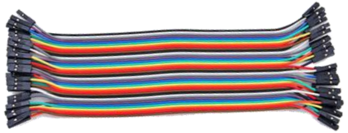
Note PC

Raspberry Pi 3

Power Charger



Serial Port (PL2303)



Jumper Cable

1

Installing Tizen Platform

- Flashing with the IoT Setup Wizard

2

Booting The RPi3

- Connecting a Host PC to RPi3 via UART

3

Building a Tizen project

- Cloning Tizen repositories
- Building the project with Tizen Studio

4

Running the Tizen project on RPi3

- Connecting RPi3 to the network
- Connecting RPi3 to Tizen Studio

Flashing with the IoT Setup Wizard

1. Installing the IoT Setup Wizard

https://developer.tizen.org/development/iot-preview/getting-started/flashing-tizen-images

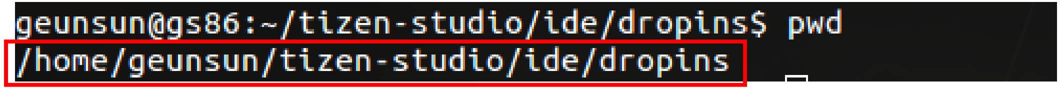
- Download the plugin

http://download.tizen.org/misc/iot-preview/org.tizen.iotsetupwizard_0.4.1.preview.jar

* *Note :* This IoT Setup Wizard version is a test version for developers.
The officially verified version is going to be released in the next release milestone.

- Add the plugin to the Tizen Studio

/home/<user>/tizen-studio/ide/**dropins**/



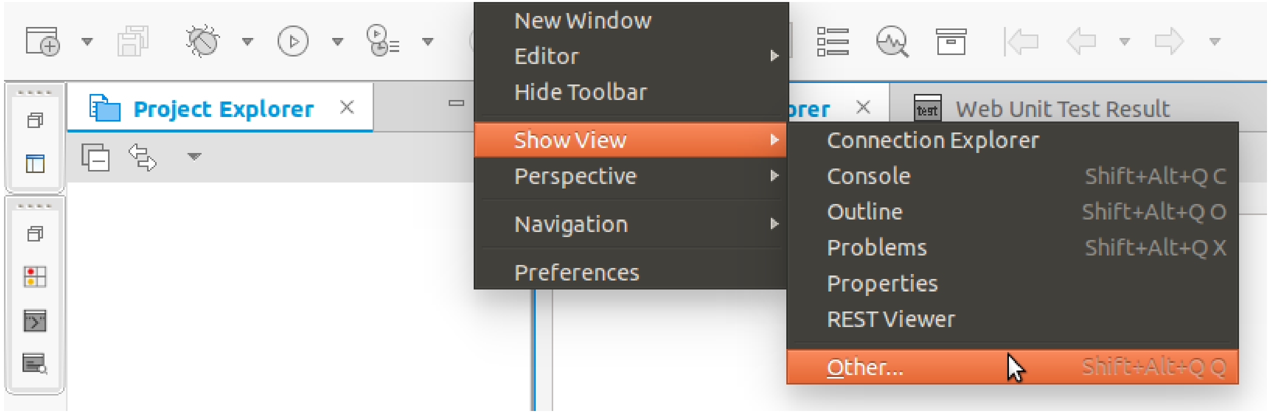
```
geunsun@gs86:~/tizen-studio/ide/dropins$ pwd
/home/geunsun/tizen-studio/ide/dropins
```

Flashing with the IoT Setup Wizard

1. Installing the IoT Setup Wizard

- Start the Tizen Studio
- Access the IoT Setup Wizard in the Tizen Studio menu by going to

Windows > Show View > Other

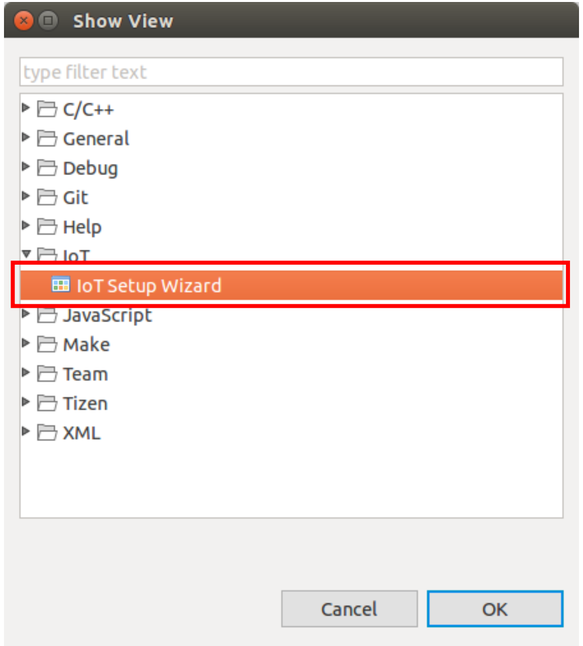


Flashing with the IoT Setup Wizard

1. Installing the IoT Setup Wizard

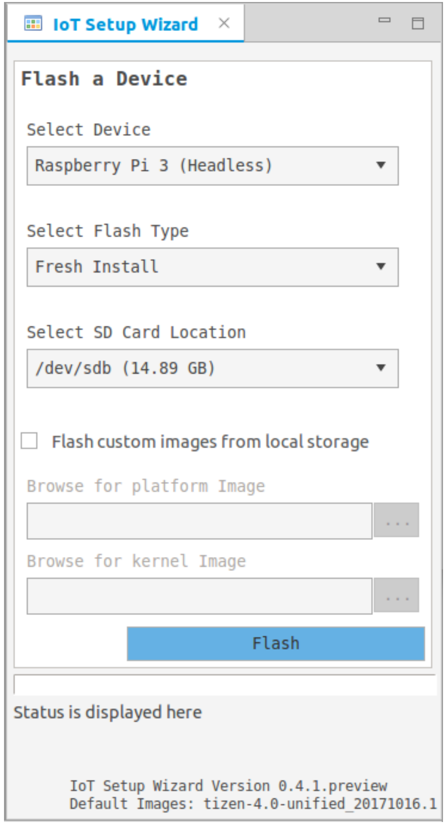
- In the Show View window, select

IoT > IoT Setup Wizard



Flashing with the IoT Setup Wizard

2. IoT Setup Wizard User Interface



Flash a Device

Select Device: Raspberry Pi 3 (Headless)

Select Flash Type: Fresh Install

Select SD Card Location: /dev/sdb (14.89 GB)

☐ Flash custom images from local storage

Browse for platform Image

Browse for kernel Image

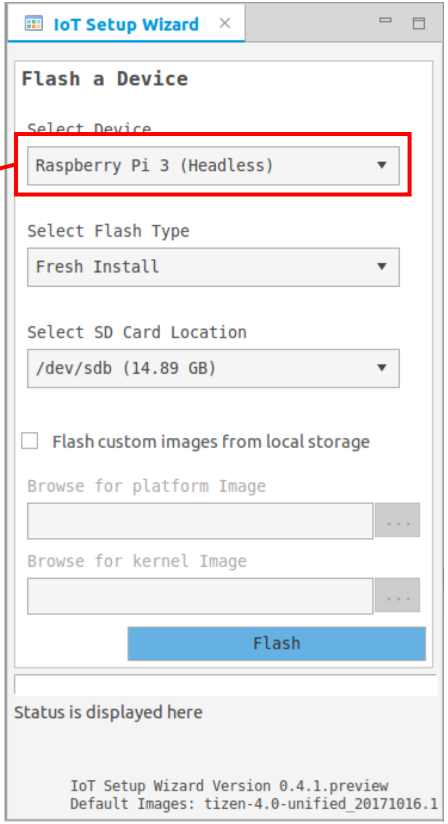
Status is displayed here

IoT Setup Wizard Version 0.4.1.preview
Default Images: tizen-4.0-unified_20171016.1

Flashing with the IoT Setup Wizard

2. IoT Setup Wizard User Interface

- Samsung Artik 530(Headless)
- Samsung Artik 530(Headed)
- Raspberry Pi 3(Headless)



Flash a Device

Select Device: Raspberry Pi 3 (Headless)

Select Flash Type: Fresh Install

Select SD Card Location: /dev/sdb (14.89 GB)

☐ Flash custom images from local storage

Browse for platform Image

Browse for kernel Image

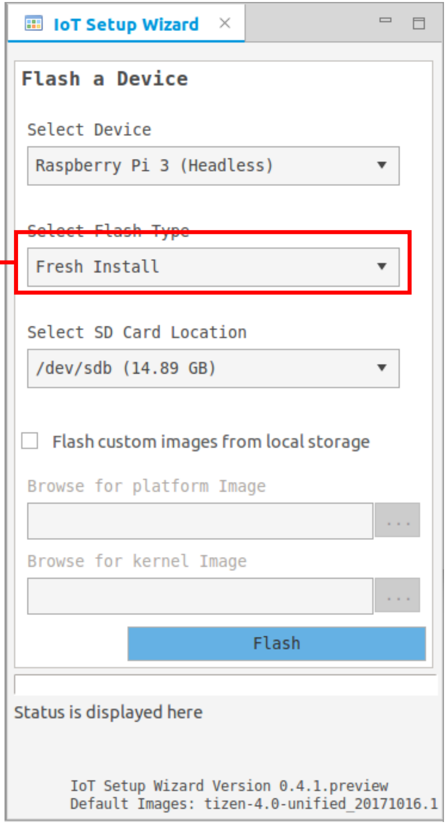
Status is displayed here

IoT Setup Wizard Version 0.4.1.preview
Default Images: tizen-4.0-unified_20171016.1

Flashing with the IoT Setup Wizard

2. IoT Setup Wizard User Interface

- Fresh Install
- Flash Kernel and Bootloader
- Flash Platform Image



Flash a Device

Select Device: Raspberry Pi 3 (Headless)

Select Flash Type: Fresh Install

Select SD Card Location: /dev/sdb (14.89 GB)

☐ Flash custom images from local storage

Browse for platform Image

Browse for kernel Image

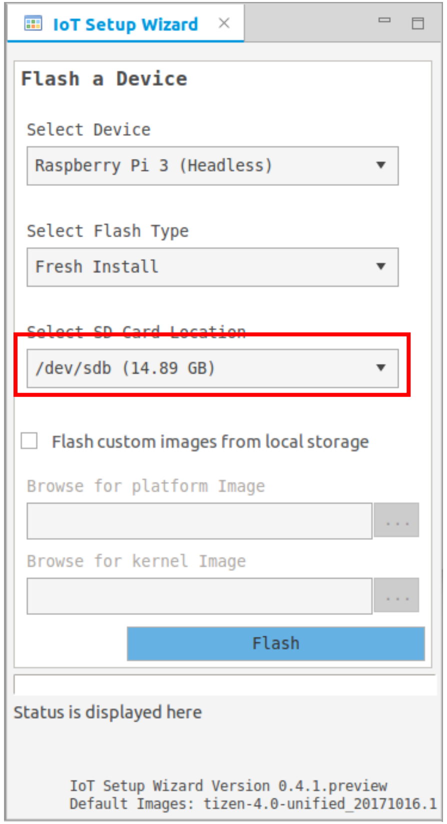
Status is displayed here

IoT Setup Wizard Version 0.4.1.preview
Default Images: tizen-4.0-unified_20171016.1

Flashing with the IoT Setup Wizard

2. IoT Setup Wizard User Interface

** Note :* If an SD card is not displayed even after inserting it into the system,
close the IoT Setup Wizard and open it again.



Flash a Device

Select Device
Raspberry Pi 3 (Headless)

Select Flash Type
Fresh Install

Select SD Card Location
/dev/sdb (14.89 GB)

☐ Flash custom images from local storage

Browse for platform Image

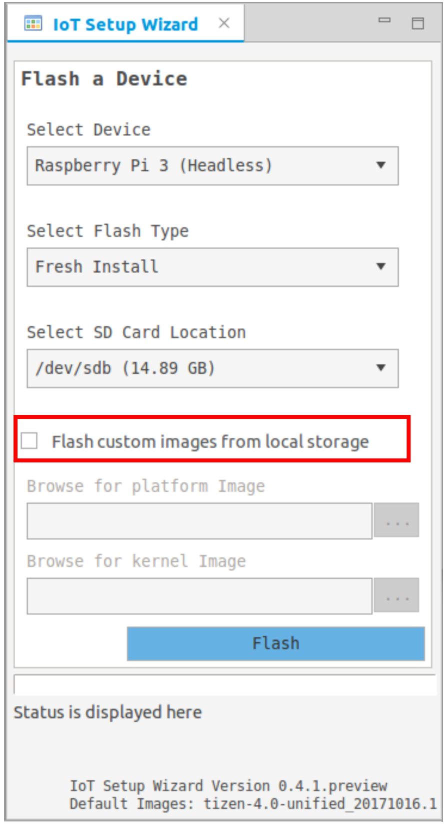
Browse for kernel Image

Status is displayed here

IoT Setup Wizard Version 0.4.1.preview
Default Images: tizen-4.0-unified_20171016.1

Flashing with the IoT Setup Wizard

2. IoT Setup Wizard User Interface



Flash a Device

Select Device: Raspberry Pi 3 (Headless)

Select Flash Type: Fresh Install

Select SD Card Location: /dev/sdb (14.89 GB)

☐ Flash custom images from local storage

Browse for platform Image

Browse for kernel Image

Status is displayed here

IoT Setup Wizard Version 0.4.1.preview
Default Images: tizen-4.0-unified_20171016.1

Flashing with the IoT Setup Wizard

2. IoT Setup Wizard User Interface

- Download the IoT Headless Image
  - Kernel & Module Image

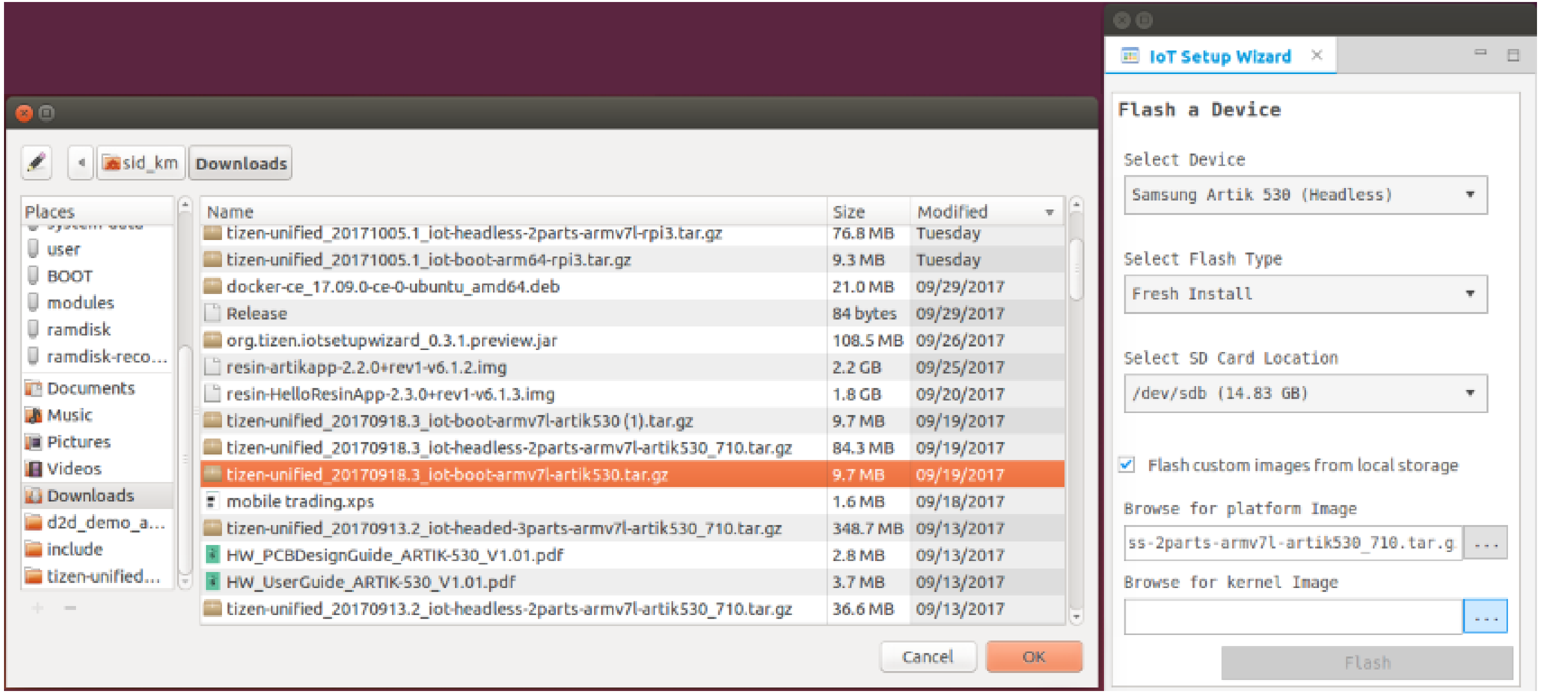
Access http://download.tizen.org/snapshots/tizen/unified/latest/images/standard/iot-boot-arm64-rpi3/
Download tizen-unified_2017xxxx.x_iot-boot-arm64-rpi3.tar.gz

- Tizen Platform Image

Access http://download.tizen.org/snapshots/tizen/unified/latest/images/standard/iot-headless-2parts-armv7l-rpi3/
Download **tizen-unified_2017xxxx.x_iot-headless-2parts-armv7l-rpi3.tar.gz**

Flashing with the IoT Setup Wizard

2. IoT Setup Wizard User Interface



sid_km / Downloads

Places

- user
- BOOT
- modules
- ramdisk
- ramdisk-reco...
- Documents
- Music
- Pictures
- Videos
- Downloads
- d2d_demo_a...
- include
- tizen-unified...

Name	Size	Modified
tizen-unified_20171005.1_iot-headless-2parts-armv7l-rpi3.tar.gz	76.8 MB	Tuesday
tizen-unified_20171005.1_iot-boot-arm64-rpi3.tar.gz	9.3 MB	Tuesday
docker-ce_17.09.0-ce-0-ubuntu_amd64.deb	21.0 MB	09/29/2017
Release	84 bytes	09/29/2017
org.tizen.iotsetupwizard_0.3.1.preview.jar	108.5 MB	09/26/2017
resin-artikapp-2.2.0+rev1-v6.1.2.img	2.2 GB	09/25/2017
resin-HelloResinApp-2.3.0+rev1-v6.1.3.img	1.8 GB	09/20/2017
tizen-unified_20170918.3_iot-boot-armv7l-artik530 (1).tar.gz	9.7 MB	09/19/2017
tizen-unified_20170918.3_iot-headless-2parts-armv7l-artik530_710.tar.gz	84.3 MB	09/19/2017
tizen-unified_20170918.3_iot-boot-armv7l-artik530.tar.gz	9.7 MB	09/19/2017
mobile trading.xps	1.6 MB	09/18/2017
tizen-unified_20170913.2_iot-headed-3parts-armv7l-artik530_710.tar.gz	348.7 MB	09/13/2017
HW_PCBDesignGuide_ARTIK-530_V1.01.pdf	2.8 MB	09/13/2017
HW_UserGuide_ARTIK-530_V1.01.pdf	3.7 MB	09/13/2017
tizen-unified_20170913.2_iot-headless-2parts-armv7l-artik530_710.tar.gz	36.6 MB	09/13/2017

IoT Setup Wizard

Flash a Device

Select Device: Samsung Artik 530 (Headless)

Select Flash Type: Fresh Install

Select SD Card Location: /dev/sdb (14.83 GB)

☑ Flash custom images from local storage

Browse for platform Image: ss-2parts-armv7l-artik530_710.tar.g

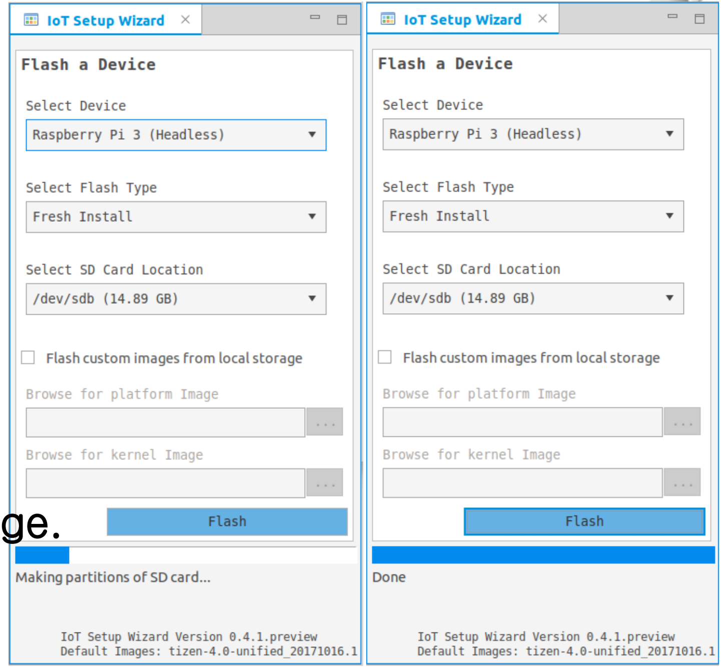
Browse for kernel Image:

Flashing with the IoT Setup Wizard

3. Flashing an SD card with the Wizard

To flash your SD card with the IoT Setup Wizard,

- Insert your SD card into the computer.
- In Tizen IoT Setup Wizard,
  select the device, flash type,
  and the SD card location.
- If you want to use your own image files
  instead of the default images,
  check the **Flash custom images from local storage.**
- Click **Flash.**



Flash a Device

Select Device: Raspberry Pi 3 (Headless)

Select Flash Type: Fresh Install

Select SD Card Location: /dev/sdb (14.89 GB)

☐ Flash custom images from local storage

Browse for platform Image

Browse for kernel Image

Making partitions of SD card...

IoT Setup Wizard Version 0.4.1.preview
Default Images: tizen-4.0-unified_20171016.1

Flash a Device

Select Device: Raspberry Pi 3 (Headless)

Select Flash Type: Fresh Install

Select SD Card Location: /dev/sdb (14.89 GB)

☐ Flash custom images from local storage

Browse for platform Image

Browse for kernel Image

Done

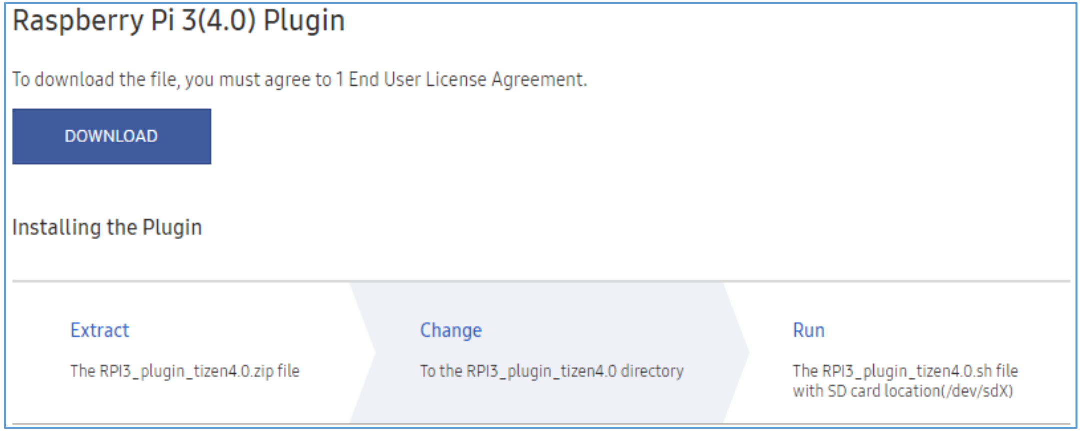
IoT Setup Wizard Version 0.4.1.preview
Default Images: tizen-4.0-unified_20171016.1

The IoT Setup Wizard takes about a minute to prepare your SD card, and once finished, displays the "Done" message in the status bar.

Flashing with the IoT Setup Wizard

4. Install the firmware for Wi-Fi and Bluetooth

- Download the plugin zip file from **Raspberry Pi 3(4.0) Plugin** section in http://developer.samsung.com/tizendevice/firmware and follow the provided instructions.



Raspberry Pi 3(4.0) Plugin

To download the file, you must agree to 1 End User License Agreement.

Installing the Plugin

Extract
The RPI3_plugin_tizen4.0.zip file

Change
To the RPI3_plugin_tizen4.0 directory

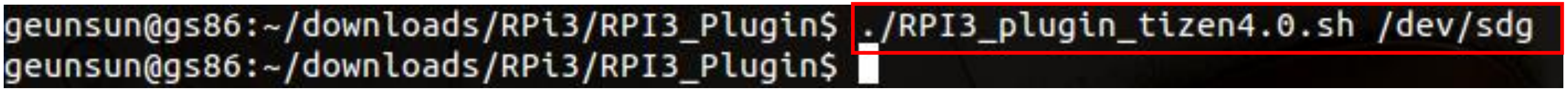
Run
The RPI3_plugin_tizen4.0.sh file with SD card location(/dev/sdX)

Flashing with the IoT Setup Wizard

4. Install the firmware for Wi-Fi and Bluetooth

- Run the script command given in the instructions in the “Linux shell”.

```
$ ./RPI3_plugin_tizen4.0.sh /dev/sd{X}   // {X} is a letter
```



```
geunsun@gs86:~/downloads/RPi3/RPI3_Plugin$ ./RPI3_plugin_tizen4.0.sh /dev/sdg
geunsun@gs86:~/downloads/RPi3/RPI3_Plugin$ 
```

1

Installing Tizen Platform

- Flashing with the IoT Setup Wizard

2

Booting The RPi3

- Connecting a Host PC to RPi3 via UART

3

Building a Tizen project

- Cloning Tizen repositories
- Building the project with Tizen Studio

4

Running the Tizen project on RPi3

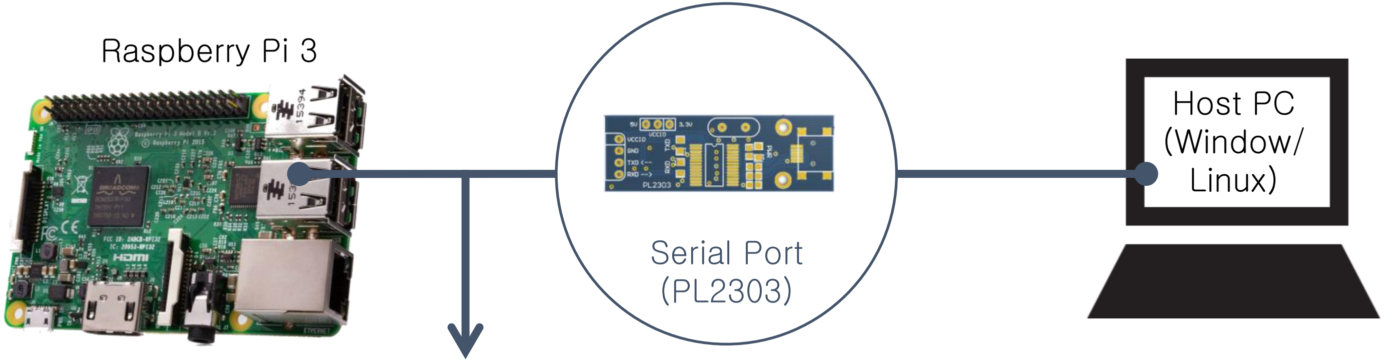
- Connecting RPi3 to the network
- Connecting RPi3 to Tizen Studio

Booting the RPi3

1. Inserting SD Card to RPi3.
2. Connecting a Host PC to RPi3 via UART Board.
3. Connecting power to RPi3.
4. Entering user ID and password

Booting the RPi3

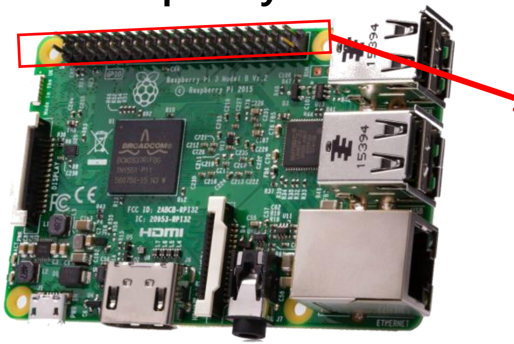
1. Inserting SD Card to RPi3.
2. Connecting a Host PC to RPi3 via UART Board.

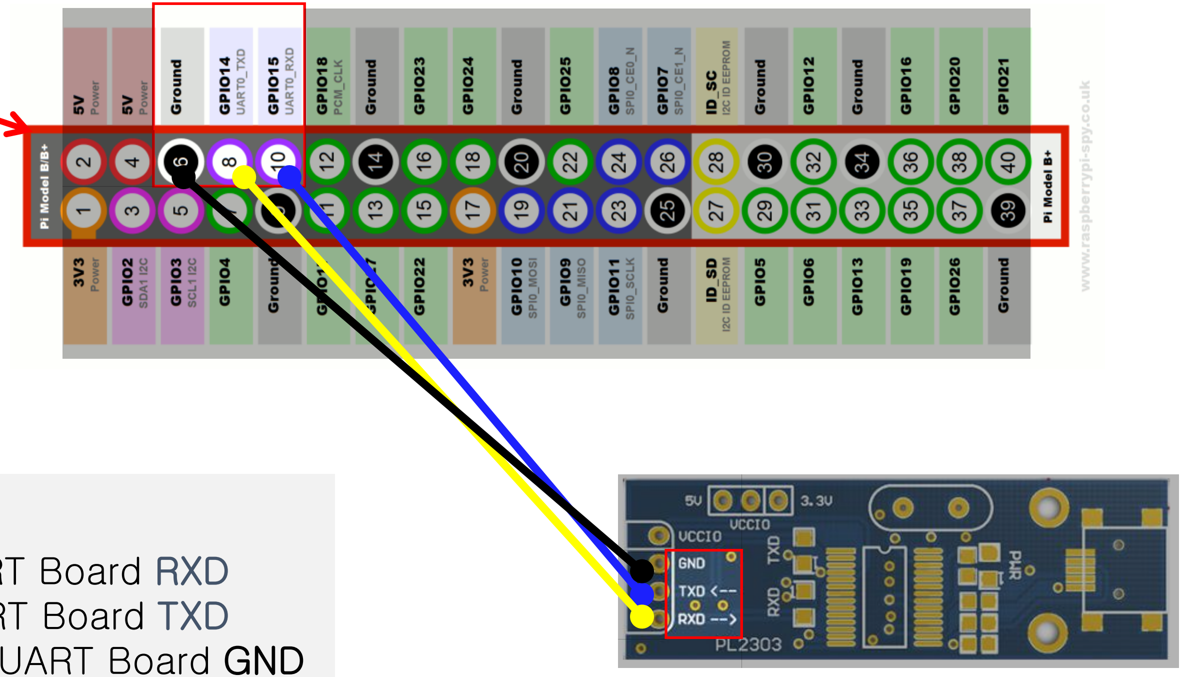


1) Information of connecting to UART Board*

1) Information of connecting to UART Board*

Raspberry Pi 3

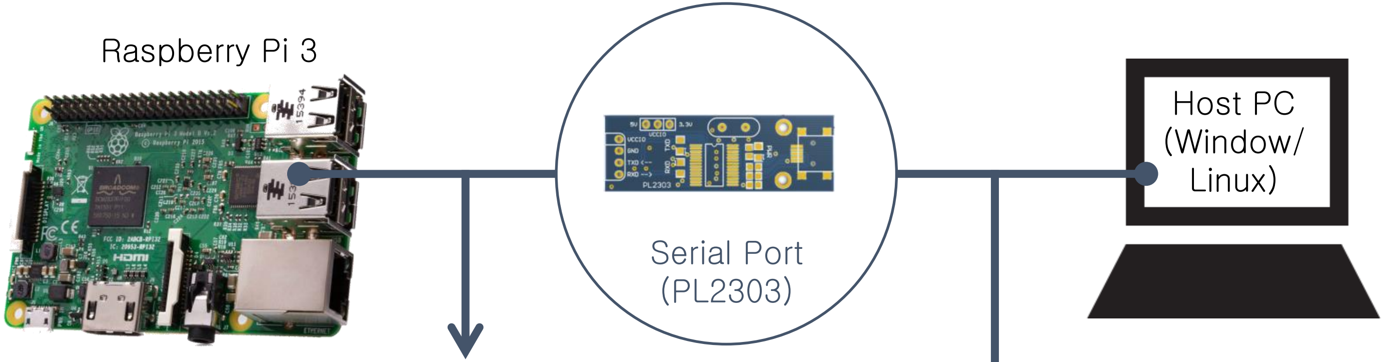




In case of **PL2303**,
RPi3 TXD(08pin) → UART Board RXD
RPi3 RXD(10pin) → UART Board TXD
RPi3 **Ground**(06pint) → UART Board **GND**

Booting the RPi3

1. Inserting SD Card to RPi3.
2. Connecting a Host PC to RPi3 via UART Board.



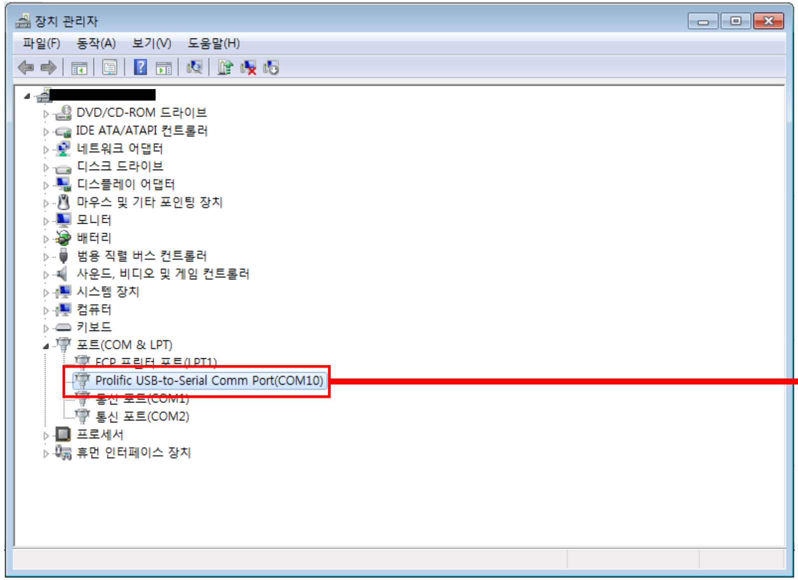
1) Information of connecting to UART Board*

2) Opening a Serial Terminal with Host PC(Windows/Linux)*

2) Opening a Serial Terminal with Host PC(Windows)*

1. Install the appropriate drivers.

Download the PL2303 Windows Driver : http://www.prolific.com.tw/US/ShowProduct.aspx?p_id=225&pcid=41



- [redacted]
  - DVD/CD-ROM 드라이브
  - IDE ATA/ATAPI 컨트롤러
  - 네트워크 어댑터
  - 디스크 드라이브
  - 디스플레이 어댑터
  - 마우스 및 기타 포인팅 장치
  - 모니터
  - 배터리
  - 범용 직렬 버스 컨트롤러
  - 사운드, 비디오 및 게임 컨트롤러
  - 시스템 장치
  - 컴퓨터
  - 키보드
  - 포트(COM & LPT)
    - ECP 프린터 포트(LPT1)
    - Prolific USB-to-Serial Comm Port(COM10)
    - 통신 포트(COM1)
    - 통신 포트(COM2)
  - 프로세서
  - 휴먼 인터페이스 장치

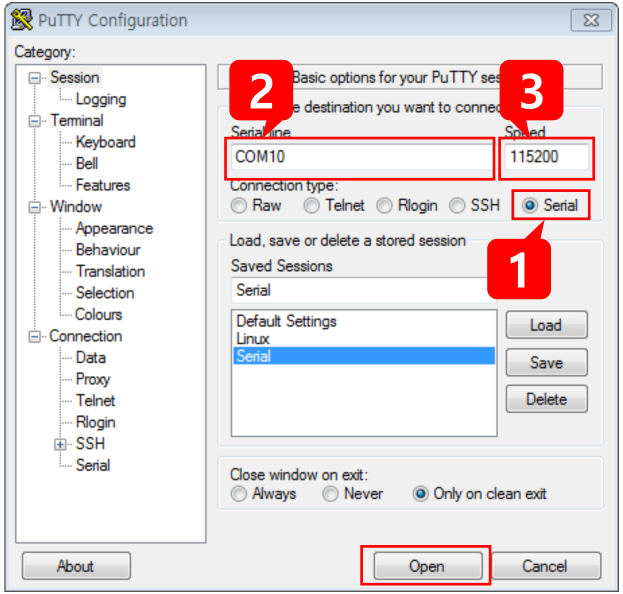
When the serial port is recognized:

Prolific USB–to–Serial Comm Port(COM#)

2) Opening a Serial Terminal with Host PC(Windows)*

1. Execute a **PuTTY** program to open serial terminal.

Download the **PuTTY** terminal emulator : http://www.putty.org/



Serial line: COM10

Speed: 115200

Connection type: ○ Raw ○ Telnet ○ Rlogin ○ SSH ● Serial

Saved Sessions: Serial

Close window on exit: ○ Always ○ Never ● Only on clean exit

Configure the **PuTTY** menu as follows:

a. Under Connection type, select **Serial**.
b. In the **Serial line** field, enter the COM# for your board, such as COM10.

* *Note :* If you did not identify your COM# when setting up your board, navigate to the Device Manager and check for an entry called **USB Serial Port**. The COM# is displayed next to the USB Serial Port entry.

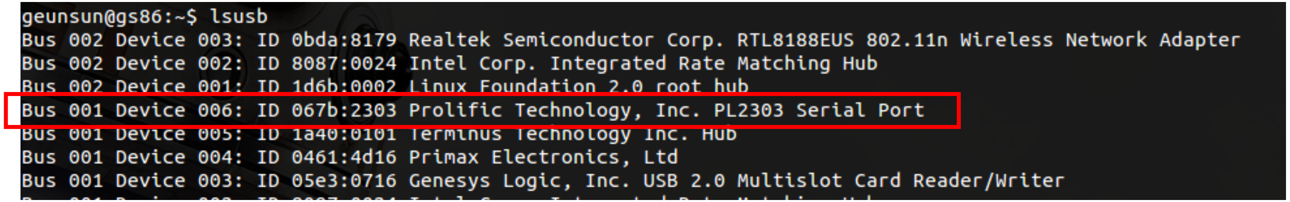
Source : https://software.intel.com/en-us/setting-up-serial-terminal-on-system-with-windows

2) Opening a Serial Terminal with Host PC(Linux)*

1. Run the Terminal program.

$ lsusb

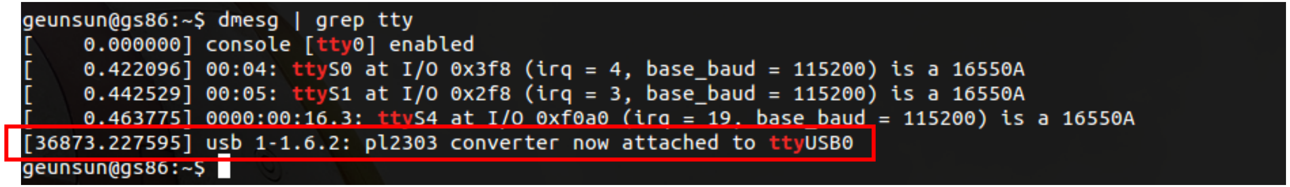
- 장치 연결 확인



```
geunsun@gs86:~$ lsusb
Bus 002 Device 003: ID 0bda:8179 Realtek Semiconductor Corp. RTL8188EUS 802.11n Wireless Network Adapter
Bus 002 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 006: ID 067b:2303 Prolific Technology, Inc. PL2303 Serial Port
Bus 001 Device 005: ID 1a40:0101 Terminus Technology Inc. Hub
Bus 001 Device 004: ID 0461:4d16 Primax Electronics, Ltd
Bus 001 Device 003: ID 05e3:0716 Genesys Logic, Inc. USB 2.0 Multislot Card Reader/Writer
```

$ dmesg | grep tty

- 장치 번호 확인



```
geunsun@gs86:~$ dmesg | grep tty
[    0.000000] console [tty0] enabled
[    0.422096] 00:04: ttyS0 at I/O 0x3f8 (irq = 4, base_baud = 115200) is a 16550A
[    0.442529] 00:05: ttyS1 at I/O 0x2f8 (irq = 3, base_baud = 115200) is a 16550A
[    0.463775] 0000:00:16.3: ttyS4 at I/O 0xf0a0 (irq = 19, base_baud = 115200) is a 16550A
[36873.227595] usb 1-1.6.2: pl2303 converter now attached to ttyUSB0
geunsun@gs86:~$ 
```

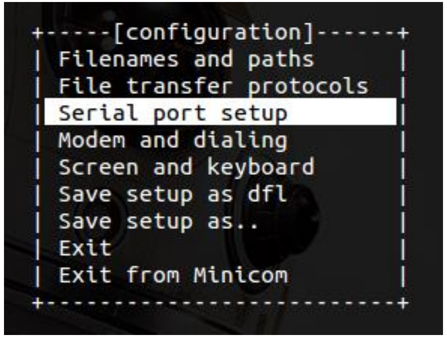
Source : https://software.intel.com/en-us/setting-up-serial-terminal-on-system-with-windows

2) Opening a Serial Terminal with Host PC(Linux)*

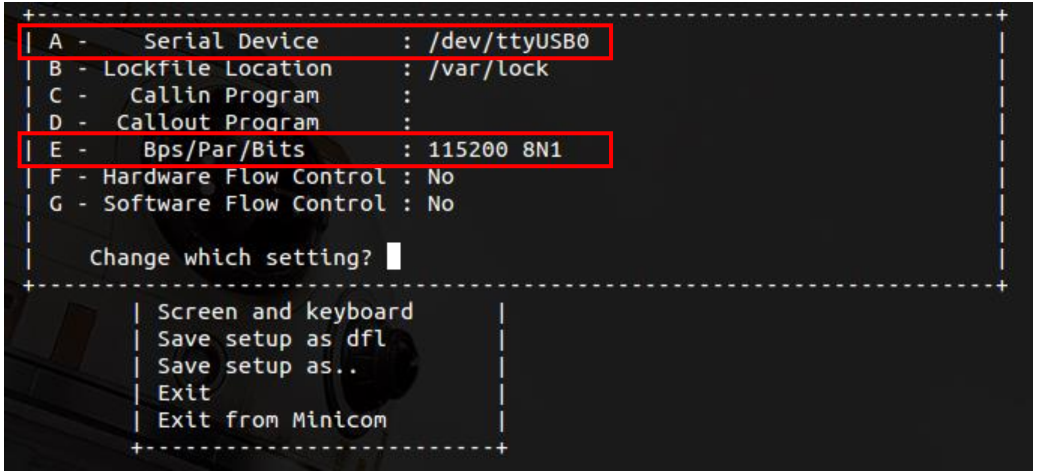
2. Run the Terminal program.

방법 1 $ sudo minicom –s (Serial Device : /dev/ttyUSB0, Bps : 115200)
방법 2 $ sudo putty (Serial line : /dev/ttyUSB0, Speed : 115200)

- minicom



```
+-----[configuration]------+
| Filenames and paths      |
| File transfer protocols  |
| Serial port setup        |
| Modem and dialing        |
| Screen and keyboard      |
| Save setup as dfl        |
| Save setup as..          |
| Exit                     |
| Exit from Minicom        |
+--------------------------+
```



```
+-----------------------------------------------------------------------+
| A -    Serial Device      : /dev/ttyUSB0                               |
| B - Lockfile Location     : /var/lock                                  |
| C -   Callin Program      :                                            |
| D -  Callout Program      :                                            |
| E -    Bps/Par/Bits       : 115200 8N1                                 |
| F - Hardware Flow Control : No                                         |
| G - Software Flow Control : No                                         |
|                                                                       |
|    Change which setting?                                              |
+-----------------------------------------------------------------------+
        | Screen and keyboard      |
        | Save setup as dfl        |
        | Save setup as..          |
        | Exit                     |
        | Exit from Minicom        |
        +--------------------------+
```

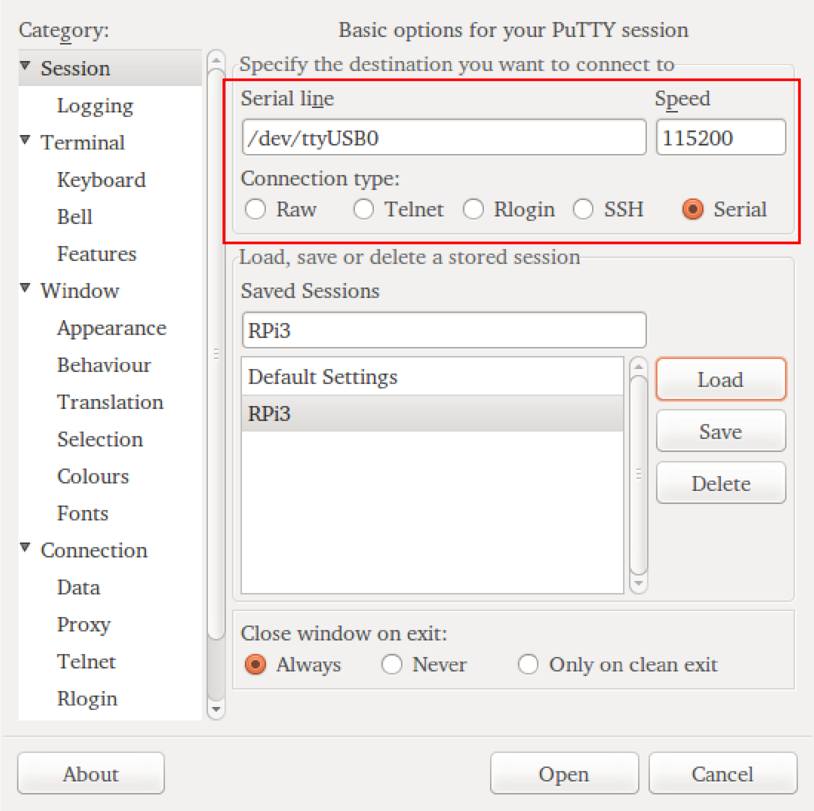
Source : https://software.intel.com/en-us/setting-up-serial-terminal-on-system-with-windows

2) Opening a Serial Terminal with Host PC(Linux)*

2. Run the Terminal program.

- putty

```
$ sudo apt-get install putty
$ sudo putty
```



Basic options for your PuTTY session

Specify the destination you want to connect to

Serial line: /dev/ttyUSB0

Speed: 115200

Connection type: ○ Raw ○ Telnet ○ Rlogin ○ SSH ● Serial

Load, save or delete a stored session

Saved Sessions: RPi3

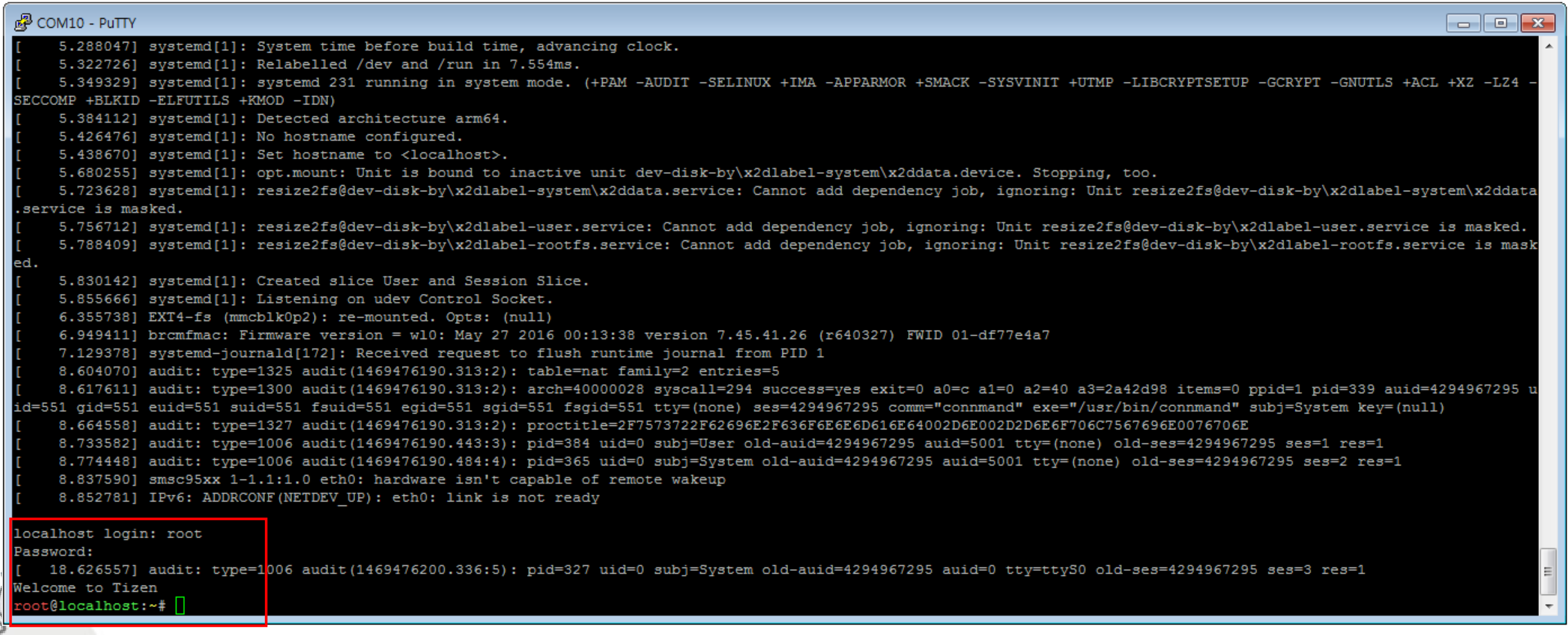
Default Settings
RPi3

Close window on exit: ● Always ○ Never ○ Only on clean exit

Source : https://software.intel.com/en-us/setting-up-serial-terminal-on-system-with-windows

Booting the RPi3

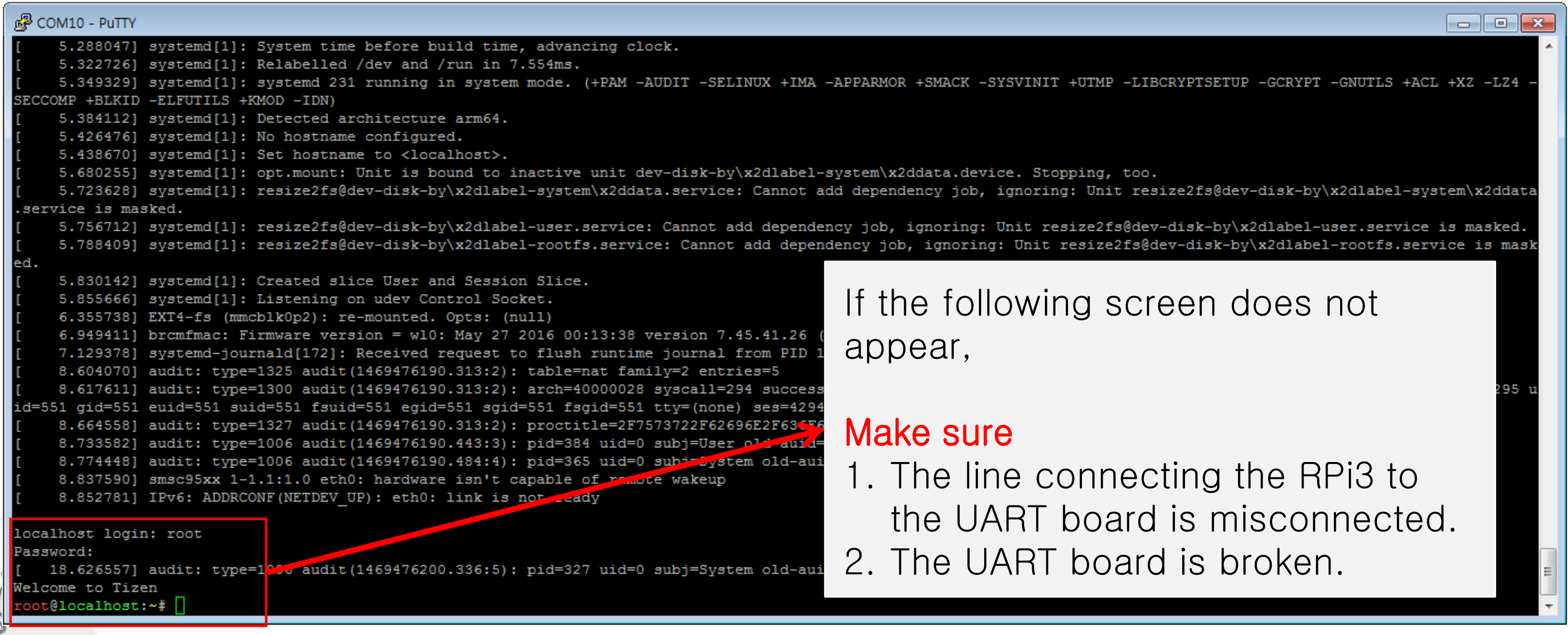
3. Connecting power to RPi3.
4. Entering user ID and password
    - User id : root / Password : tizen (The password is not shown when you enter it.)



```
[    5.288047] systemd[1]: System time before build time, advancing clock.
[    5.322726] systemd[1]: Relabelled /dev and /run in 7.554ms.
[    5.349329] systemd[1]: systemd 231 running in system mode. (+PAM -AUDIT -SELINUX +IMA -APPARMOR +SMACK -SYSVINIT +UTMP -LIBCRYPTSETUP -GCRYPT -GNUTLS +ACL +XZ -LZ4 -
SECCOMP +BLKID -ELFUTILS +KMOD -IDN)
[    5.384112] systemd[1]: Detected architecture arm64.
[    5.426476] systemd[1]: No hostname configured.
[    5.438670] systemd[1]: Set hostname to <localhost>.
[    5.680255] systemd[1]: opt.mount: Unit is bound to inactive unit dev-disk-by\x2dlabel-system\x2ddata.device. Stopping, too.
[    5.723628] systemd[1]: resize2fs@dev-disk-by\x2dlabel-system\x2ddata.service: Cannot add dependency job, ignoring: Unit resize2fs@dev-disk-by\x2dlabel-system\x2ddata
.service is masked.
[    5.756712] systemd[1]: resize2fs@dev-disk-by\x2dlabel-user.service: Cannot add dependency job, ignoring: Unit resize2fs@dev-disk-by\x2dlabel-user.service is masked.
[    5.788409] systemd[1]: resize2fs@dev-disk-by\x2dlabel-rootfs.service: Cannot add dependency job, ignoring: Unit resize2fs@dev-disk-by\x2dlabel-rootfs.service is mask
ed.
[    5.830142] systemd[1]: Created slice User and Session Slice.
[    5.855666] systemd[1]: Listening on udev Control Socket.
[    6.355738] EXT4-fs (mmcblk0p2): re-mounted. Opts: (null)
[    6.949411] brcmfmac: Firmware version = wl0: May 27 2016 00:13:38 version 7.45.41.26 (r640327) FWID 01-df77e4a7
[    7.129378] systemd-journald[172]: Received request to flush runtime journal from PID 1
[    8.604070] audit: type=1325 audit(1469476190.313:2): table=nat family=2 entries=5
[    8.617611] audit: type=1300 audit(1469476190.313:2): arch=40000028 syscall=294 success=yes exit=0 a0=c a1=0 a2=40 a3=2a42d98 items=0 ppid=1 pid=339 auid=4294967295 u
id=551 gid=551 euid=551 suid=551 fsuid=551 egid=551 sgid=551 fsgid=551 tty=(none) ses=4294967295 comm="connmand" exe="/usr/bin/connmand" subj=System key=(null)
[    8.664558] audit: type=1327 audit(1469476190.313:2): proctitle=2F7573722F62696E2F636F6E6E6D616E64002D6E002D2D6E6F706C7567696E0076706E
[    8.733582] audit: type=1006 audit(1469476190.443:3): pid=384 uid=0 subj=User old-auid=4294967295 auid=5001 tty=(none) old-ses=4294967295 ses=1 res=1
[    8.774448] audit: type=1006 audit(1469476190.484:4): pid=365 uid=0 subj=System old-auid=4294967295 auid=5001 tty=(none) old-ses=4294967295 ses=2 res=1
[    8.837590] smsc95xx 1-1.1:1.0 eth0: hardware isn't capable of remote wakeup
[    8.852781] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready

localhost login: root
Password:
[   18.626557] audit: type=1006 audit(1469476200.336:5): pid=327 uid=0 subj=System old-auid=4294967295 auid=0 tty=ttyS0 old-ses=4294967295 ses=3 res=1
Welcome to Tizen
root@localhost:~#
```

Booting the RPi3

3. Connecting power to RPi3.
4. Entering user ID and password
   - User id : root / Password : tizen (The password is not shown when you enter it.)



```
[    5.288047] systemd[1]: System time before build time, advancing clock.
[    5.322726] systemd[1]: Relabelled /dev and /run in 7.554ms.
[    5.349329] systemd[1]: systemd 231 running in system mode. (+PAM -AUDIT -SELINUX +IMA -APPARMOR +SMACK -SYSVINIT +UTMP -LIBCRYPTSETUP -GCRYPT -GNUTLS +ACL +XZ -LZ4 -
SECCOMP +BLKID -ELFUTILS +KMOD -IDN)
[    5.384112] systemd[1]: Detected architecture arm64.
[    5.426476] systemd[1]: No hostname configured.
[    5.438670] systemd[1]: Set hostname to <localhost>.
[    5.680255] systemd[1]: opt.mount: Unit is bound to inactive unit dev-disk-by\x2dlabel-system\x2ddata.device. Stopping, too.
[    5.723628] systemd[1]: resize2fs@dev-disk-by\x2dlabel-system\x2ddata.service: Cannot add dependency job, ignoring: Unit resize2fs@dev-disk-by\x2dlabel-system\x2ddata
.service is masked.
[    5.756712] systemd[1]: resize2fs@dev-disk-by\x2dlabel-user.service: Cannot add dependency job, ignoring: Unit resize2fs@dev-disk-by\x2dlabel-user.service is masked.
[    5.788409] systemd[1]: resize2fs@dev-disk-by\x2dlabel-rootfs.service: Cannot add dependency job, ignoring: Unit resize2fs@dev-disk-by\x2dlabel-rootfs.service is mask
ed.
[    5.830142] systemd[1]: Created slice User and Session Slice.
[    5.855666] systemd[1]: Listening on udev Control Socket.
[    6.355738] EXT4-fs (mmcblk0p2): re-mounted. Opts: (null)
[    6.949411] brcmfmac: Firmware version = wl0: May 27 2016 00:13:38 version 7.45.41.26 (
[    7.129378] systemd-journald[172]: Received request to flush runtime journal from PID 1
[    8.604070] audit: type=1325 audit(1469476190.313:2): table=nat family=2 entries=5
[    8.617611] audit: type=1300 audit(1469476190.313:2): arch=40000028 syscall=294 success                                     295 u
id=551 gid=551 euid=551 suid=551 fsuid=551 egid=551 sgid=551 fsgid=551 tty=(none) ses=4294
[    8.664558] audit: type=1327 audit(1469476190.313:2): proctitle=2F7573722F62696E2F63
[    8.733582] audit: type=1006 audit(1469476190.443:3): pid=384 uid=0 subj=User old-auid=
[    8.774448] audit: type=1006 audit(1469476190.484:4): pid=365 uid=0 subj=System old-aui
[    8.837590] smsc95xx 1-1.1:1.0 eth0: hardware isn't capable of remote wakeup
[    8.852781] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready

localhost login: root
Password:
[   18.626557] audit: type=1006 audit(1469476200.336:5): pid=327 uid=0 subj=System old-aui
Welcome to Tizen
root@localhost:~#
```

If the following screen does not appear,

Make sure

1. The line connecting the RPi3 to the UART board is misconnected.
2. The UART board is broken.

1

Installing Tizen Platform

- Flashing with the IoT Setup Wizard

2

Booting The RPi3

- Connecting a Host PC to RPi3 via UART

3

Building a Tizen project

- Cloning Tizen repositories
- Building the project with Tizen Studio

4

Running the Tizen project on RPi3

- Connecting RPi3 to the network
- Connecting RPi3 to Tizen Studio

Prerequisite

1. Downloading the Git.

   `https://www.git-scm.com/downloads`

2. Installing the Tizen Studio.

   `https://developer.tizen.org/development/tizen-studio/download`

Cloning Tizen repositories

Template source code

- position–finder–server

```
Git path : apps/native/position–finder–server
Branch : template
```

- position–finder–client

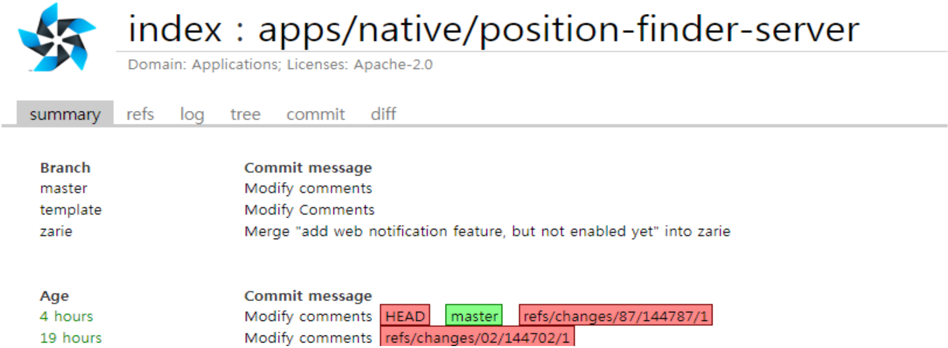
```
Git path : apps/native/position–finder–client
```

Cloning Tizen repositories

1) Access the Tizen Project GIT Repositories.

```
https://git.tizen.org/
```

2) Searching for position-finder-server.



index : apps/native/position-finder-server

Domain: Applications; Licenses: Apache-2.0

Branch	Commit message
master	Modify comments
template	Modify Comments
zarie	Merge "add web notification feature, but not enabled yet" into zarie

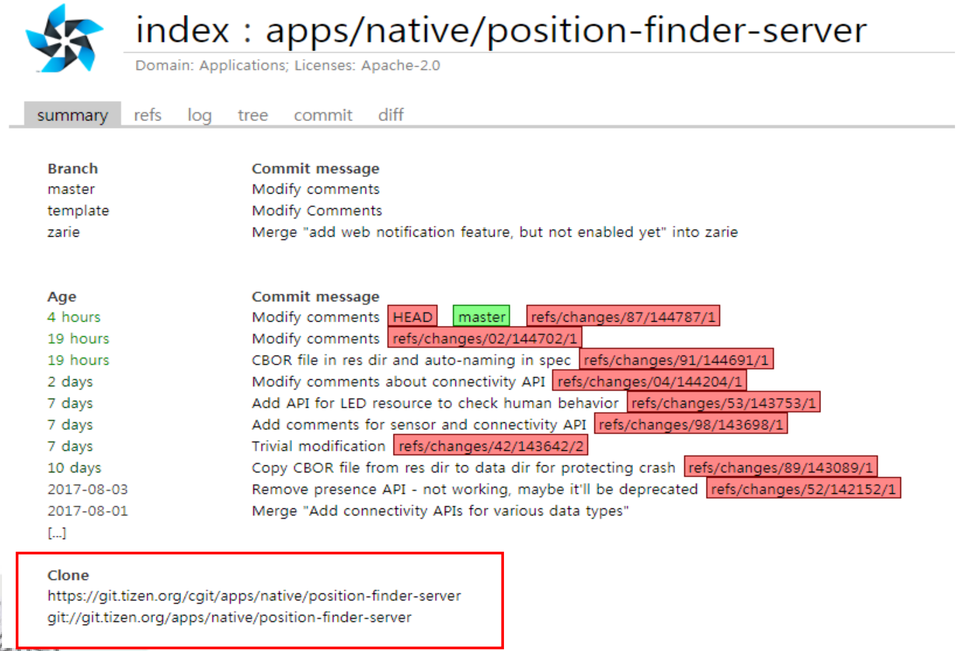
Age	Commit message
4 hours	Modify comments HEAD master refs/changes/87/144787/1
19 hours	Modify comments refs/changes/02/144702/1

3) How to clone the repository : apps/native/position-finder-server.*
4) Running a Git Bash and cloning the Tizen repository.*
5) How to change the branch of the project.*
6) Changing the branch of the Tizen Project.*

3) How to clone the repository : apps/native/position-finder-server*

```
$ git clone <Git address of the repository>
$ git clone git://git.tizen.org/apps/native/position-finder-server
```

https://git.tizen.org/



index : apps/native/position-finder-server

Domain: Applications; Licenses: Apache-2.0

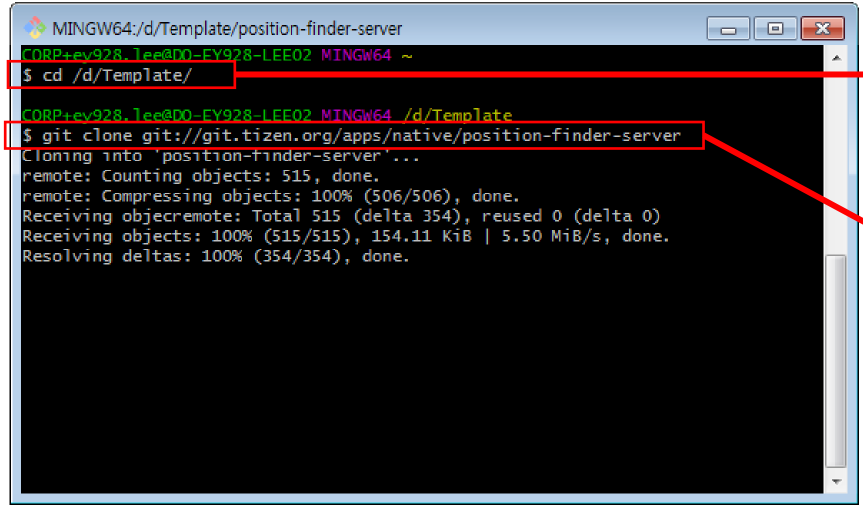
Branch	Commit message
master	Modify comments
template	Modify Comments
zarie	Merge "add web notification feature, but not enabled yet" into zarie

Age	Commit message
4 hours	Modify comments HEAD master refs/changes/87/144787/1
19 hours	Modify comments refs/changes/02/144702/1
19 hours	CBOR file in res dir and auto-naming in spec refs/changes/91/144691/1
2 days	Modify comments about connectivity API refs/changes/04/144204/1
7 days	Add API for LED resource to check human behavior refs/changes/53/143753/1
7 days	Add comments for sensor and connectivity API refs/changes/98/143698/1
7 days	Trivial modification refs/changes/42/143642/2
10 days	Copy CBOR file from res dir to data dir for protecting crash refs/changes/89/143089/1
2017-08-03	Remove presence API - not working, maybe it'll be deprecated refs/changes/52/142152/1
2017-08-01	Merge "Add connectivity APIs for various data types"
[...]	

Clone
https://git.tizen.org/cgit/apps/native/position-finder-server
git://git.tizen.org/apps/native/position-finder-server

4) Running a Git Bash and cloning the Tizen repository*

Git bash



```
CORP+ey928.lee@DO-EY928-LEE02 MINGW64 ~
$ cd /d/Template/

CORP+ey928.lee@DO-EY928-LEE02 MINGW64 /d/Template
$ git clone git://git.tizen.org/apps/native/position-finder-server
Cloning into 'position-finder-server'...
remote: Counting objects: 515, done.
remote: Compressing objects: 100% (506/506), done.
Receiving objecremote: Total 515 (delta 354), reused 0 (delta 0)
Receiving objects: 100% (515/515), 154.11 KiB | 5.50 MiB/s, done.
Resolving deltas: 100% (354/354), done.
```

$ cd <THE PATH>
Enters the path to download the source code.

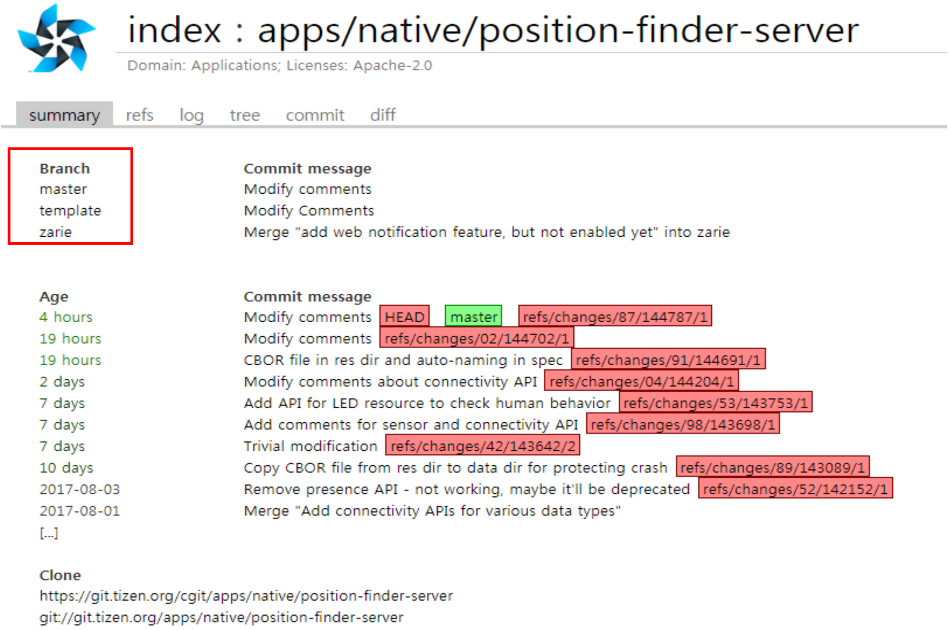
$ git clone <Git address of the repository>
Downloads the repository on your local PC using the *git clone* command.

ex) $ git clone git://git.tizen.org/apps/native/position-finder-server

5) How to change the branch of the project*

```
$ git checkout <BRANCH NAME>
$ git checkout template
```

https://git.tizen.org/



index : apps/native/position-finder-server

Domain: Applications; Licenses: Apache-2.0

Branch	Commit message
master	Modify comments
template	Modify Comments
zarie	Merge "add web notification feature, but not enabled yet" into zarie

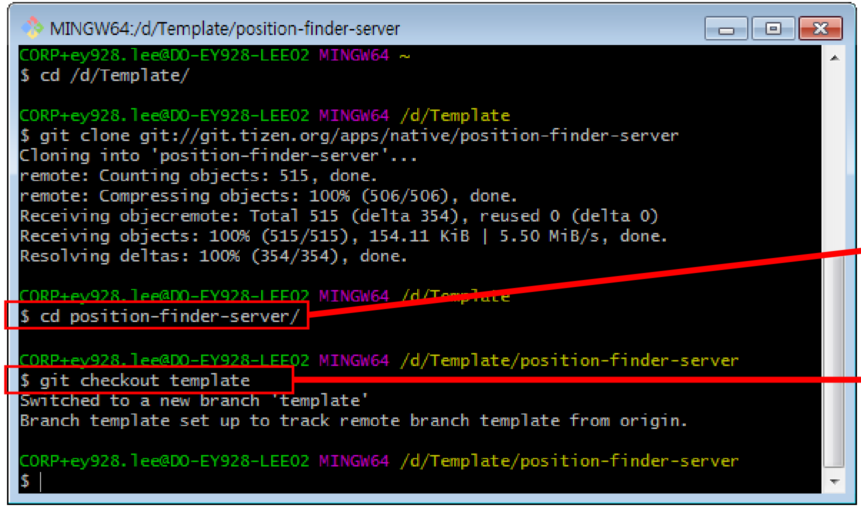
Age	Commit message
4 hours	Modify comments HEAD master refs/changes/87/144787/1
19 hours	Modify comments refs/changes/02/144702/1
19 hours	CBOR file in res dir and auto-naming in spec refs/changes/91/144691/1
2 days	Modify comments about connectivity API refs/changes/04/144204/1
7 days	Add API for LED resource to check human behavior refs/changes/53/143753/1
7 days	Add comments for sensor and connectivity API refs/changes/98/143698/1
7 days	Trivial modification refs/changes/42/143642/2
10 days	Copy CBOR file from res dir to data dir for protecting crash refs/changes/89/143089/1
2017-08-03	Remove presence API - not working, maybe it'll be deprecated refs/changes/52/142152/1
2017-08-01	Merge "Add connectivity APIs for various data types"
[...]	

Clone

https://git.tizen.org/cgit/apps/native/position-finder-server
git://git.tizen.org/apps/native/position-finder-server

6) Changing the branch of the Tizen Project*

Git bash


MINGW64:/d/Template/position-finder-server

```
CORP+ey928.lee@DO-EY928-LEE02 MINGW64 ~
$ cd /d/Template/

CORP+ey928.lee@DO-EY928-LEE02 MINGW64 /d/Template
$ git clone git://git.tizen.org/apps/native/position-finder-server
Cloning into 'position-finder-server'...
remote: Counting objects: 515, done.
remote: Compressing objects: 100% (506/506), done.
Receiving objecremote: Total 515 (delta 354), reused 0 (delta 0)
Receiving objects: 100% (515/515), 154.11 KiB | 5.50 MiB/s, done.
Resolving deltas: 100% (354/354), done.

CORP+ey928.lee@DO-EY928-LEE02 MINGW64 /d/Template
$ cd position-finder-server/

CORP+ey928.lee@DO-EY928-LEE02 MINGW64 /d/Template/position-finder-server
$ git checkout template
Switched to a new branch 'template'
Branch template set up to track remote branch template from origin.

CORP+ey928.lee@DO-EY928-LEE02 MINGW64 /d/Template/position-finder-server
$
```

$ cd position-finder-server
Enter the project.

$ git checkout <BRANCH NAME>
Change the branch of the project using the *git checkout* command.

ex) git checkout template

Building the project with Tizen Studio

1. Importing the cloned project into Tizen Studio.*

a. *File* > Select *Import…*
b. Tizen > Select Tizen Project > Next
c. *Select Directory / File* > Select Root directory
d. *Location* > Browse > Select the path of the cloned project(… .₩position-finder-server) > Next
e. Finish

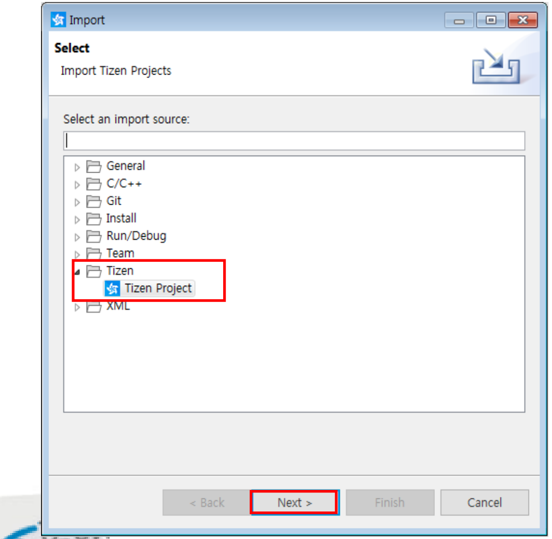
2. Building the project

Project > Build Project (F10)

Import the project into Tizen Studio.*

a. *File* > Select *Import…*
b. Tizen > Select Tizen Project > Next*
c. *Select Directory / File* > Select Root directory*
d. *Location* > Browse > Select the path of the cloned project(….₩position-finder-server) > Next*
e. Finish

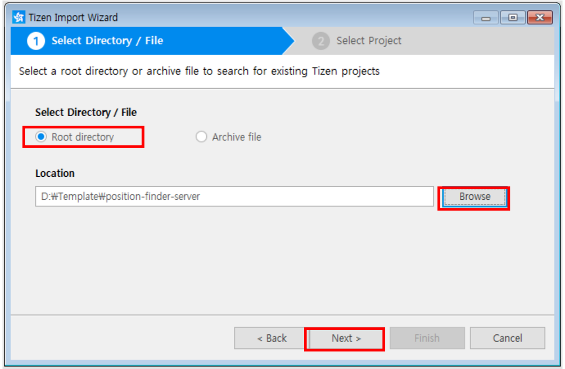
1-b.



Import Tizen Projects

Select an import source:

1-c.

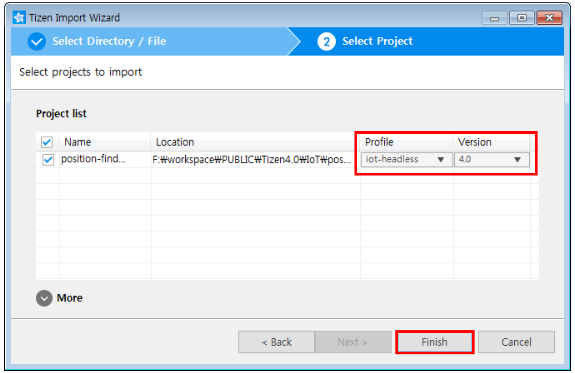


Select a root directory or archive file to search for existing Tizen projects

● Root directory ○ Archive file

Location: D:₩Template₩position-finder-server

1-d.



Select projects to import

Project list

☑	Name	Location	Profile	Version
☑	position-find…	F:₩workspace₩PUBLIC₩Tizen4.0₩IoT₩pos…	iot-headless	4.0

Building the project with Tizen Studio

1. Import the project into Tizen Studio.*

a. *File* > Select *Import…*
b. Tizen > Select Tizen Project > Next
c. *Select Directory / File* > Select Root directory
d. *Location* > Browse > Select the path of the cloned project(….₩position-finder-server) > Next
e. Finish

2. Build the project

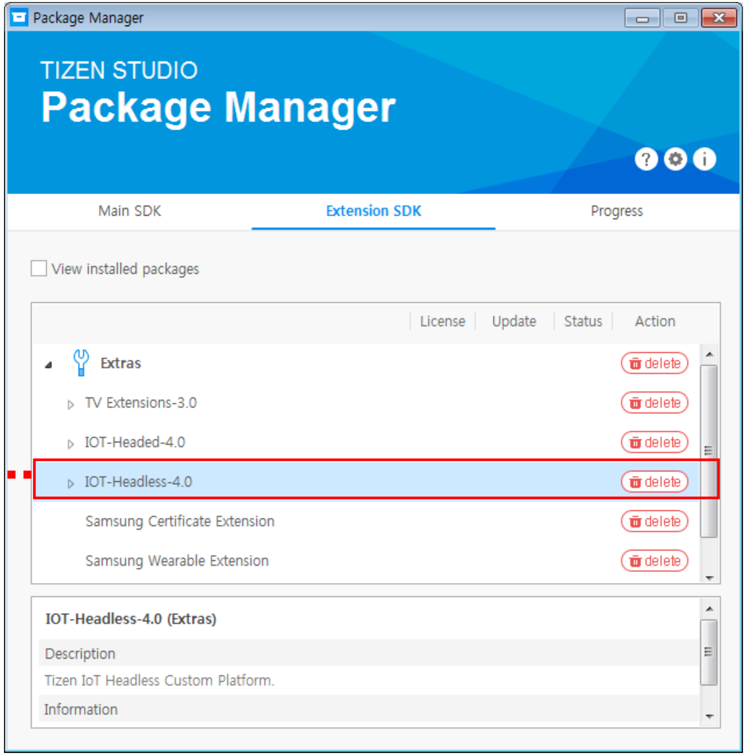
Project > Build Project (F10)

NOTE* (Native 개발 환경 설치)

Package Manager -> 개발하고자 하는 package 설치

** NOTE :* 해당 자료에서는 IOT-Headless-4.0 package가 설치되었음을 전제로 실습을 진행하였습니다.

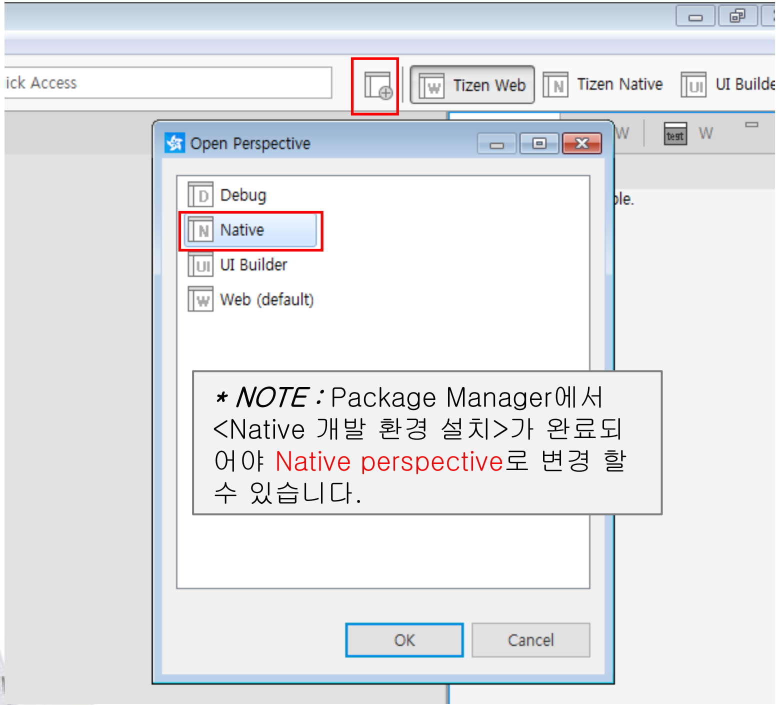
IOT-Headless-4.0



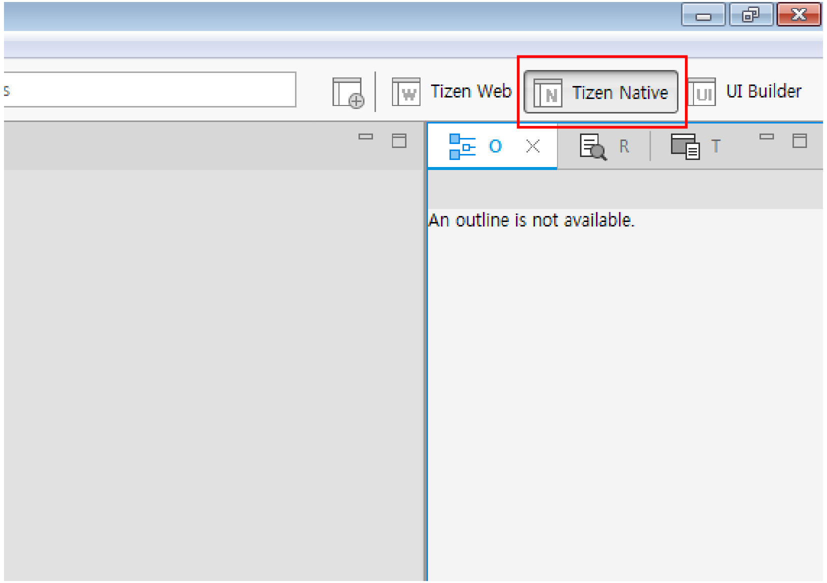
Tizen IoT Headless Custom Platform.

NOTE* (Native 개발 시 Perspective 설정)

Perspective 변경



* *NOTE :* Package Manager에서 <Native 개발 환경 설치>가 완료되어야 Native perspective로 변경 할 수 있습니다.



An outline is not available.

1

Installing Tizen Platform

- Flashing with the IoT Setup Wizard

2

Booting The RPi3

- Connecting a Host PC to RPi3 via UART

3

Building a Tizen project

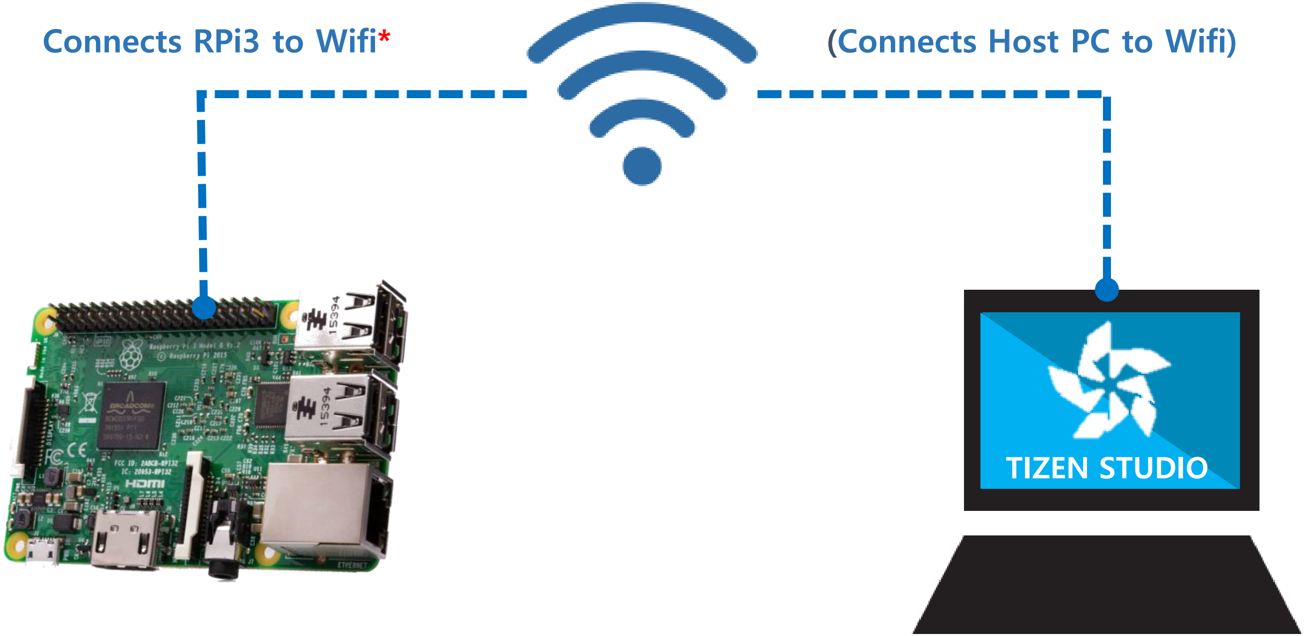
- Cloning Tizen repositories
- Building the project with Tizen Studio

4

Running the Tizen project on RPi3

- Connecting RPi3 to the network
- Connecting RPi3 to Tizen Studio

Connecting RPi3 to the network



Connects RPi3 to Wifi*

(Connects Host PC to Wifi)

Connects RPi3 to Wifi *

Raspberry Pi
3

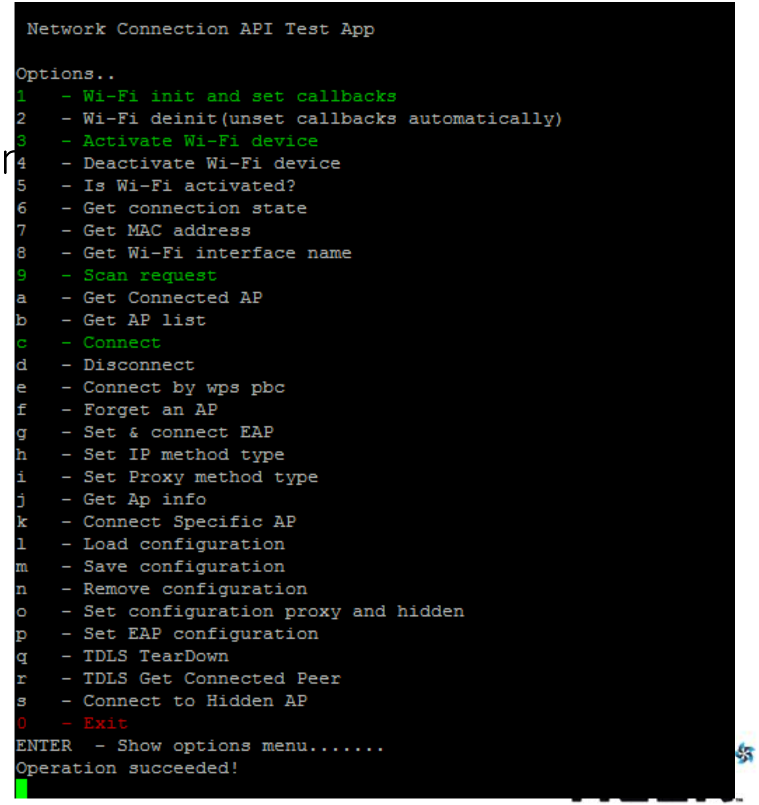
1. Connects a Host PC to RPi3 via UART.
2. Opens the serial terminal of RPi3.
3. Connects Wifi with the wifi_manager_test commar

```
$ wifi_manager_test
```

Follow the steps below:

(Input only numbers.)
1. Wi-Fi Init and set callbacks
3. Activate Wi-Fi device
c. Connect (Write AP name/password)
6. Get connection state (Verify the network is connected)
0. Exit

** Note :* You should not misspell a AP name and password when you enter it. If you get typo, you have to run it again from the beginning.



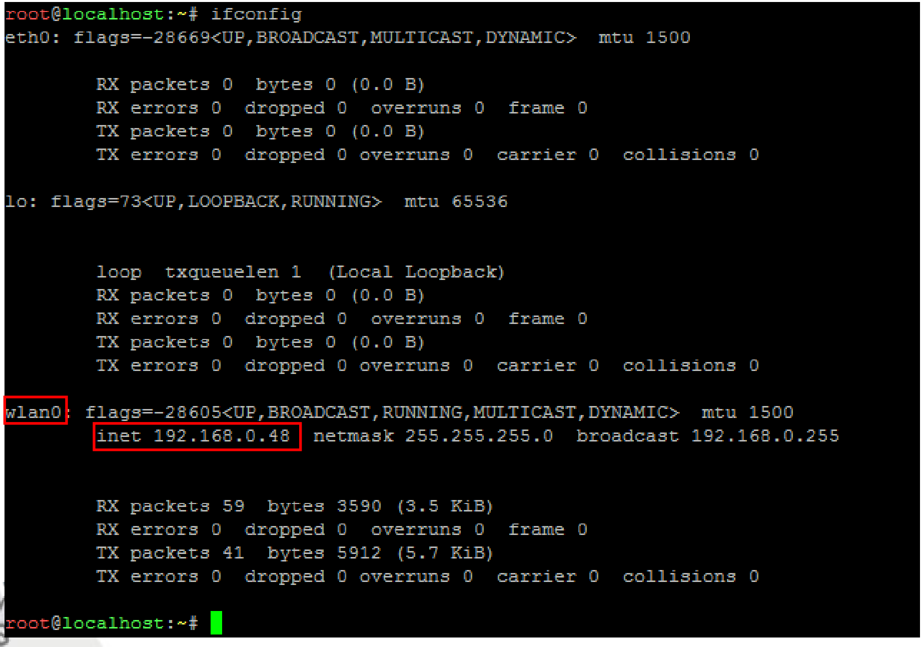
```
 Network Connection API Test App

Options..
1    - Wi-Fi init and set callbacks
2    - Wi-Fi deinit(unset callbacks automatically)
3    - Activate Wi-Fi device
4    - Deactivate Wi-Fi device
5    - Is Wi-Fi activated?
6    - Get connection state
7    - Get MAC address
8    - Get Wi-Fi interface name
9    - Scan request
a    - Get Connected AP
b    - Get AP list
c    - Connect
d    - Disconnect
e    - Connect by wps pbc
f    - Forget an AP
g    - Set & connect EAP
h    - Set IP method type
i    - Set Proxy method type
j    - Get Ap info
k    - Connect Specific AP
l    - Load configuration
m    - Save configuration
n    - Remove configuration
o    - Set configuration proxy and hidden
p    - Set EAP configuration
q    - TDLS TearDown
r    - TDLS Get Connected Peer
s    - Connect to Hidden AP
0    - Exit
ENTER  - Show options menu.......
Operation succeeded!
```

Connects RPi3 to Wifi *

Raspberry Pi 3

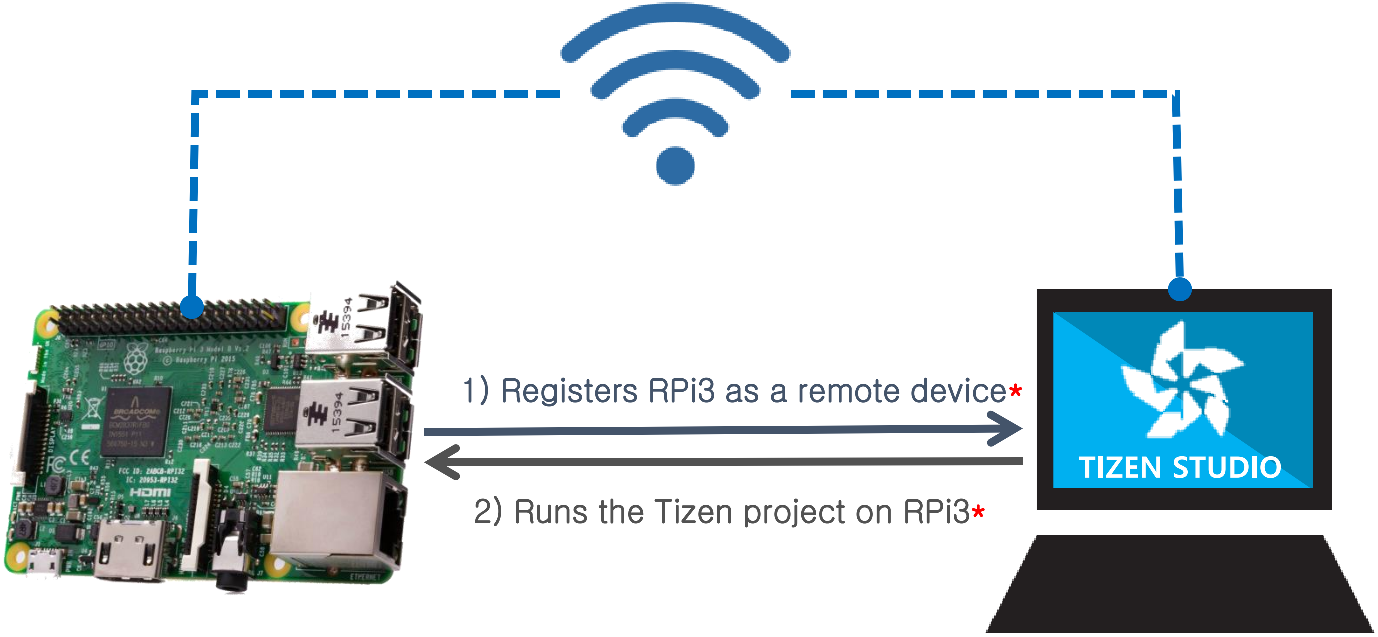
4. Checks the IP address of the connected Wi-Fi on RPi3.

```
$ ifconfig
```



```
root@localhost:~# ifconfig
eth0: flags=-28669<UP,BROADCAST,MULTICAST,DYNAMIC>  mtu 1500

        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536


        loop  txqueuelen 1  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=-28605<UP,BROADCAST,RUNNING,MULTICAST,DYNAMIC>  mtu 1500
        inet 192.168.0.48  netmask 255.255.255.0  broadcast 192.168.0.255


        RX packets 59  bytes 3590 (3.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 41  bytes 5912 (5.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@localhost:~#
```

Connecting RPi3 to Tizen Studio

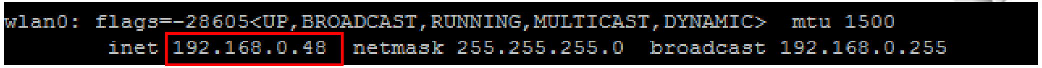


1) Registers RPi3 as a remote device*

2) Runs the Tizen project on RPi3*

1) Registers RPi3 as a remote device*

* *Note :* Ip address of RPi3 (Refer to **Connects RPi3 to Wifi***)



```
wlan0: flags=-28605<UP,BROADCAST,RUNNING,MULTICAST,DYNAMIC>  mtu 1500
        inet 192.168.0.48  netmask 255.255.255.0  broadcast 192.168.0.255
```

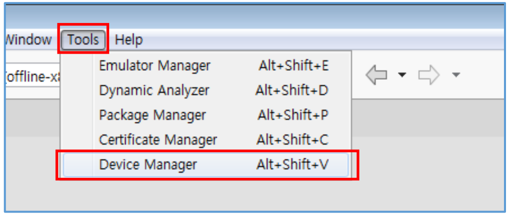
Tizen Studio

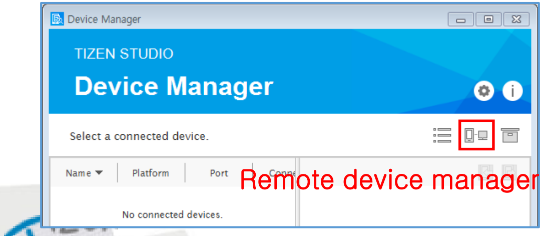
1. Adds RPi3 as a remote device.
   a. *Tools > Device Manager > Remote Device Manager*
   b. Scan (scanning remote devices)
2. Sets the connection state to ON.

If the connection fails,

1. Check if Wifi is disconnected or not.
2. Your RPi3 and PC are on the same local network.

1-a.


Window
Tools
Help
offline-x
Emulator Manager Alt+Shift+E
Dynamic Analyzer Alt+Shift+D
Package Manager Alt+Shift+P
Certificate Manager Alt+Shift+C
Device Manager Alt+Shift+V

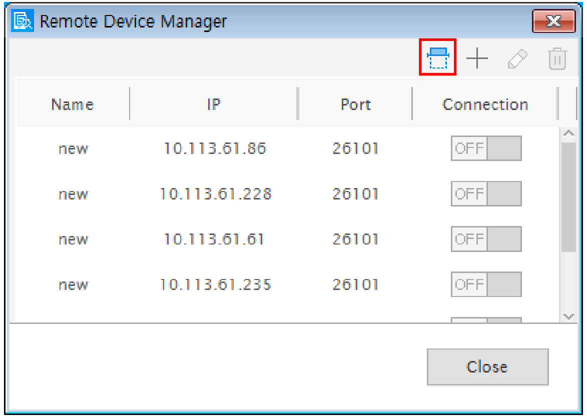


Select a connected device.

No connected devices.

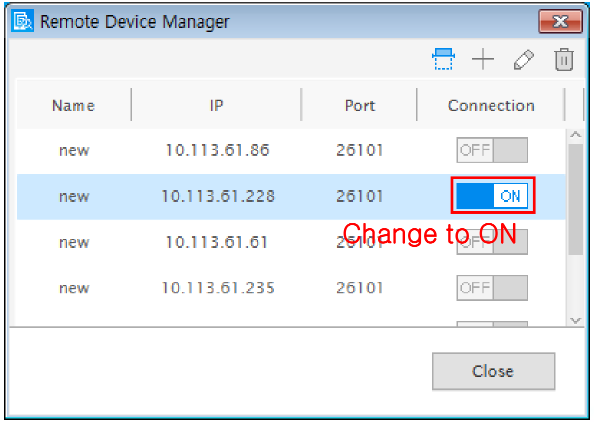
Remote device manager

1-b.



Name	IP	Port	Connection
new	10.113.61.86	26101	OFF
new	10.113.61.228	26101	OFF
new	10.113.61.61	26101	OFF
new	10.113.61.235	26101	OFF

2.



Name	IP	Port	Connection
new	10.113.61.86	26101	OFF
new	10.113.61.228	26101	ON
new	10.113.61.61	26101	OFF
new	10.113.61.235	26101	OFF

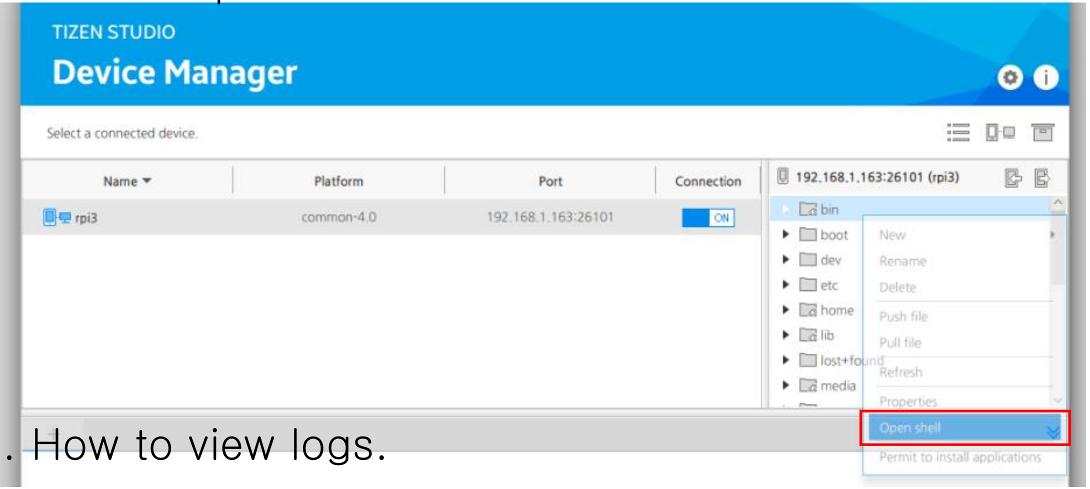
Change to ON

2) Runs the Tizen project on RPi3*

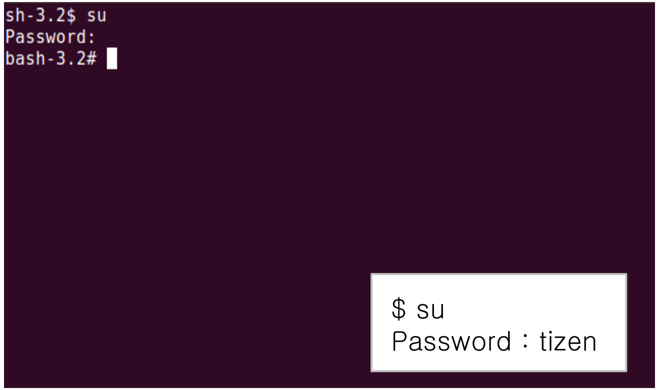
1. Running the project on RPi3.

Run > Run As > …

2. How to open the shell.



Name	Platform	Port	Connection
rpi3	common-4.0	192.168.1.163:26101	ON



```
sh-3.2$ su
Password:
bash-3.2#
```

$ su
Password : tizen

* *NOTE:* The password is not shown when you enter it.

3. How to view logs.

Execute ***dlogutil*** in the opened shell.

$ dlogutil <LOG_TAG> ex) dlogutil TT

Appendix

How to install IoT Headless image on RPi3 (for ubuntu)

1. Downloads a script that can install the image of tizen platform on the SD card.

- **rpi_sdcard_setup.sh**

```
$ wget https://git.tizen.org/cgit/apps/native/position-finder-client/plain/tool/rpi_sdcard_setup.sh
$ chmod +x rpi_sdcard_setup.sh
```

- You can also go to the location where the script is located and download it.
  - Clones position-finder-client repository with **Git Bash**.

```
$ git clone git://git.tizen.org/apps/native/position-finder-client (Branch : template)
```

  - The script(**rpi_sdcard_setup.sh**) is located in the position-finder-client/tools directory.

```
$ cd position-finder-client/tools/
```

How to install IoT Headless image on RPi3 (for ubuntu)

1. Downloads a script that can install the image of tizen platform on the SD card.

- **rpi_sdcard_setup.sh**

```
$ wget https://git.tizen.org/cgit/apps/native/position-finder-client/plain/tool/rpi_sdcard_setup.sh
$ chmod +x rpi_sdcard_setup.sh
```

- You can also go to the location where the script is located and download it.
  - Clones position-finder-client repository with **Git Bash**.

```
$ git clone git://git.tizen.org/apps/native/position-finder-client (Branch : template)
```

  - The script(**rpi_sdcard_setup.sh**) is located in the position-finder-client/tools directory.

```
$ cd position-finder-client/tools/
```

How to install IoT Headless image on RPi3 (for ubuntu)

2. Runs the downloaded script on the SD card.

1. Inserts SD card reader with an SD card into the PC.

2. Runs the script.

```
$ ./rpi_sdcard_setup.sh
```

3. There are three questions when the script runs.

```
#1 : Please type device node of usb [press enter key to use default(sdb)] :
```

```
#2 : Please type binary version of tizen [press enter key to use default(latest)] :
```

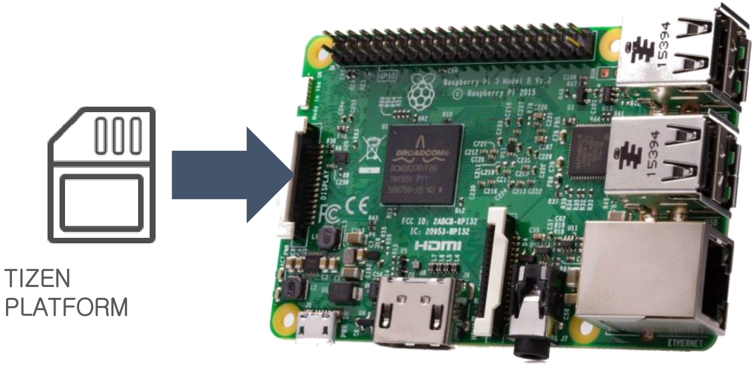
   - When you press enter, latest binary is downloaded. If you want another version, please enter the desired version date. (e.g. 20171110.3)
   - Information about the version can be found on http://download.tizen.org/snapshots/tizen/unified/

```
#3 : /dev/sdb will be formatted, Is it OK? [y/n]
```

What the script(rpi_sdcard_setup.sh) does

1. Downloading files needed for image installation
2. Downloading the IoT Headless Image
3. Making partitions of SD Card
4. Fusing boot & platform image to SD Card
5. Copying firmwares for Wi-Fi and Bluetooth

* *NOTE* : The above should be done on Ubuntu PC.



1. Downloading files needed for image installation

1. Installing pv package on Ubuntu PC.

```
$ sudo apt-get install pv
```

2. Downloading the fusing-script for RPi3.

```
$ wget https://git.tizen.org/cgit/platform/kernel/linux-
rpi3/plain/scripts/sd_fusing_rpi3.sh?h=submit/tizen/20170725.223437 --output-
document=$download_dir/$script_fusing
```

3. Downloading firmwares for Wi-Fi and Bluetooth.

```
$ wget https://github.com/RPi-Distro/firmware-nonfree/raw/master/brcm80211/brcm/brcmfmac43430-sdio.bin
$ wget https://github.com/RPi-Distro/firmware-nonfree/raw/master/brcm80211/brcm/brcmfmac43430-sdio.txt
$ wget https://github.com/OpenELEC/misc-firmware/raw/master/firmware/brcm/BCM43430A1.hcd
```

2. Downloading the IoT Headless Image

- Kernel & Module Image

Access http://download.tizen.org/snapshots/tizen/unified/latest/images/standard/iot-boot-arm64-rpi3/
Download tizen-unified_2017xxxx.x_iot-boot-arm64-rpi3.tar.gz

- Tizen Platform Image

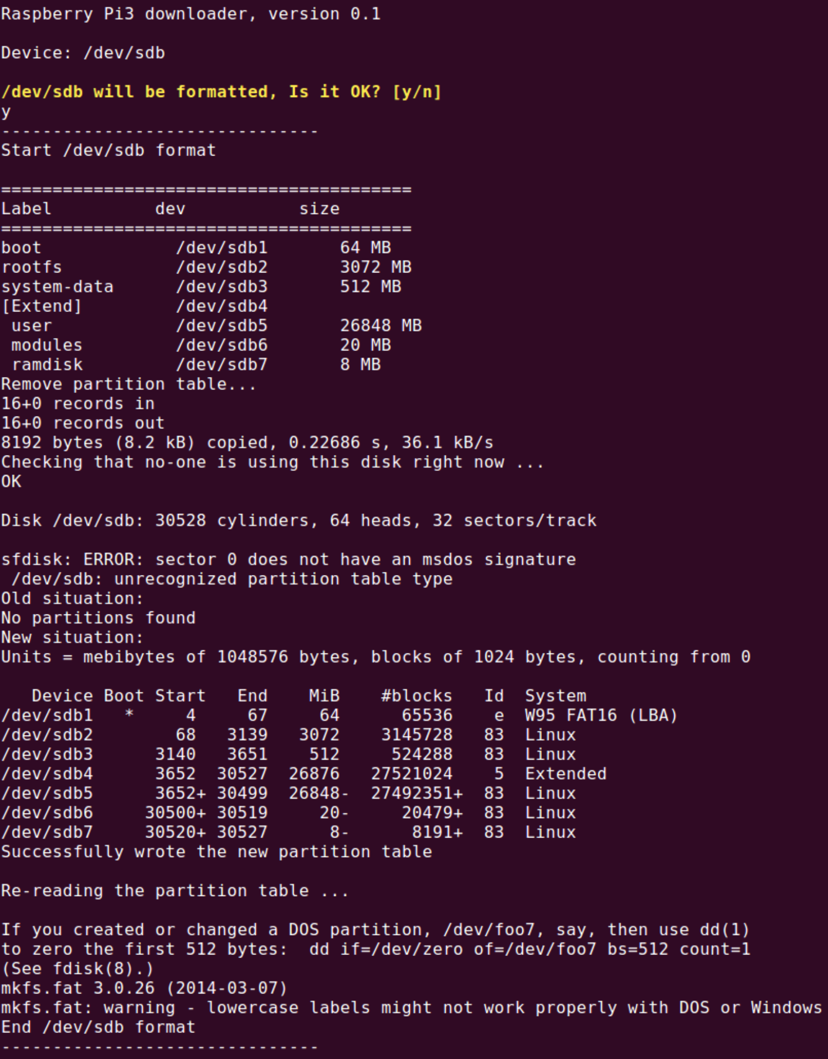
Access http://download.tizen.org/snapshots/tizen/unified/latest/images/standard/iot-headless-2parts-armv7l-rpi3/
Download **tizen-unified_2017xxxx.x_iot-headless-2parts-armv7l-rpi3.tar.gz**

3. Making partitions of SD Card

1. Check the device node of SD card. (ex. sdx : x is alphabet)
2. Make partitions using the fusing-script.*

```
$ sudo ./sd_fusing_rpi3.sh -d /dev/sdb --format
```

Screen when SD Card is formatted.*



```
Raspberry Pi3 downloader, version 0.1

Device: /dev/sdb

/dev/sdb will be formatted, Is it OK? [y/n]
y
-------------------------------
Start /dev/sdb format

========================================
Label          dev           size
========================================
boot             /dev/sdb1       64 MB
rootfs           /dev/sdb2       3072 MB
system-data      /dev/sdb3       512 MB
[Extend]         /dev/sdb4
 user            /dev/sdb5       26848 MB
 modules         /dev/sdb6       20 MB
 ramdisk         /dev/sdb7       8 MB
Remove partition table...
16+0 records in
16+0 records out
8192 bytes (8.2 kB) copied, 0.22686 s, 36.1 kB/s
Checking that no-one is using this disk right now ...
OK

Disk /dev/sdb: 30528 cylinders, 64 heads, 32 sectors/track

sfdisk: ERROR: sector 0 does not have an msdos signature
 /dev/sdb: unrecognized partition table type
Old situation:
No partitions found
New situation:
Units = mebibytes of 1048576 bytes, blocks of 1024 bytes, counting from 0

   Device Boot Start   End    MiB    #blocks   Id  System
/dev/sdb1   *      4     67     64      65536    e  W95 FAT16 (LBA)
/dev/sdb2         68   3139   3072    3145728   83  Linux
/dev/sdb3       3140   3651    512     524288   83  Linux
/dev/sdb4       3652  30527  26876   27521024    5  Extended
/dev/sdb5       3652+ 30499  26848-  27492351+  83  Linux
/dev/sdb6      30500+ 30519     20-     20479+  83  Linux
/dev/sdb7      30520+ 30527      8-      8191+  83  Linux
Successfully wrote the new partition table

Re-reading the partition table ...

If you created or changed a DOS partition, /dev/foo7, say, then use dd(1)
to zero the first 512 bytes:  dd if=/dev/zero of=/dev/foo7 bs=512 count=1
(See fdisk(8).)
mkfs.fat 3.0.26 (2014-03-07)
mkfs.fat: warning - lowercase labels might not work properly with DOS or Windows
End /dev/sdb format
-------------------------------
```

4. Fusing boot & platform image to SD Card

1. Write boot image to SD Card using the fusing-script.*

```
$ sudo ./sd_fusing_rpi3.sh -d /dev/sdb -b tizen-unified_2017xxxx.x_iot-boot-arm64-rpi3.tar.gz
```

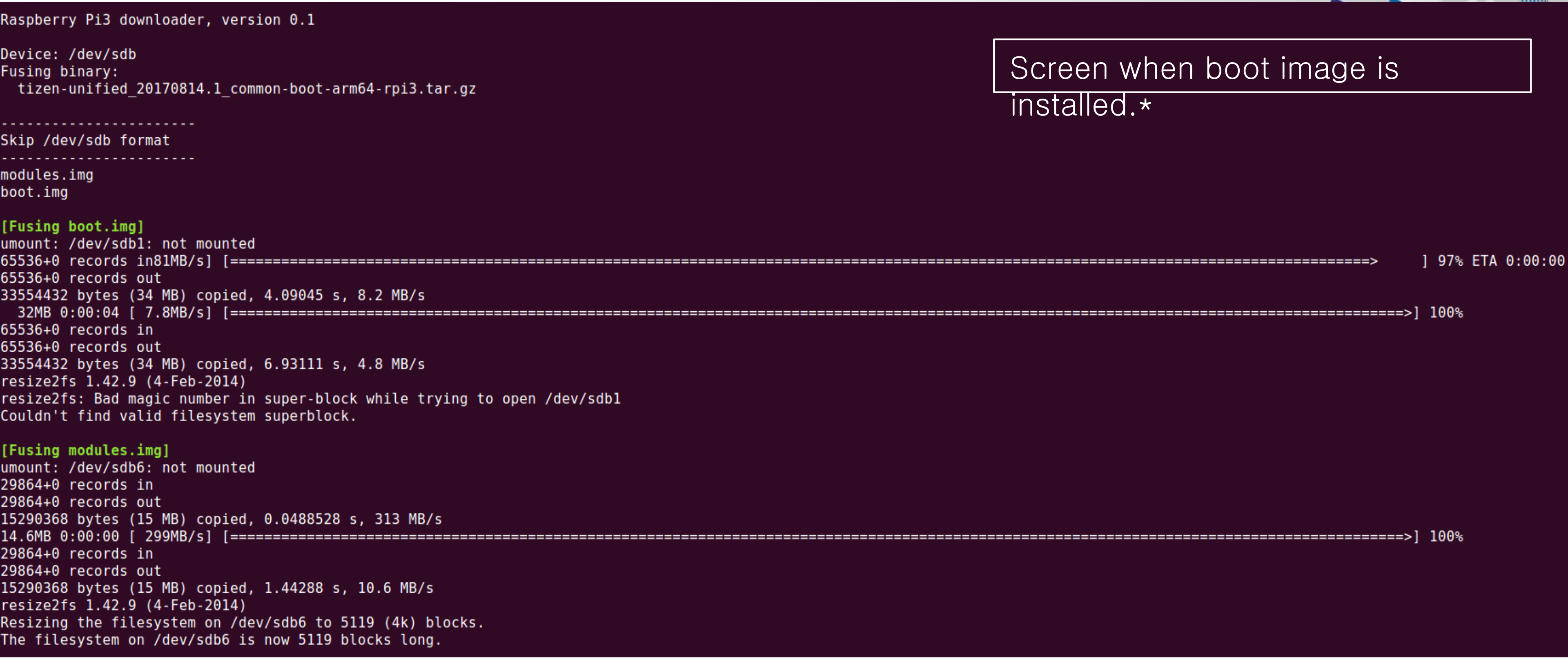
2. Write IoT Headless image to SD Card using the fusing-script.*

```
$ sudo ./sd_fusing_rpi3.sh -d /dev/sdb -b tizen-unified_2017xxxx.x_iot-headless-2parts-armv7l-rpi3.tar.gz
```

5. Copying firmwares for Wi-Fi and Bluetooth

Copy firmwares for Wi-Fi and Bluetooth to SD Card

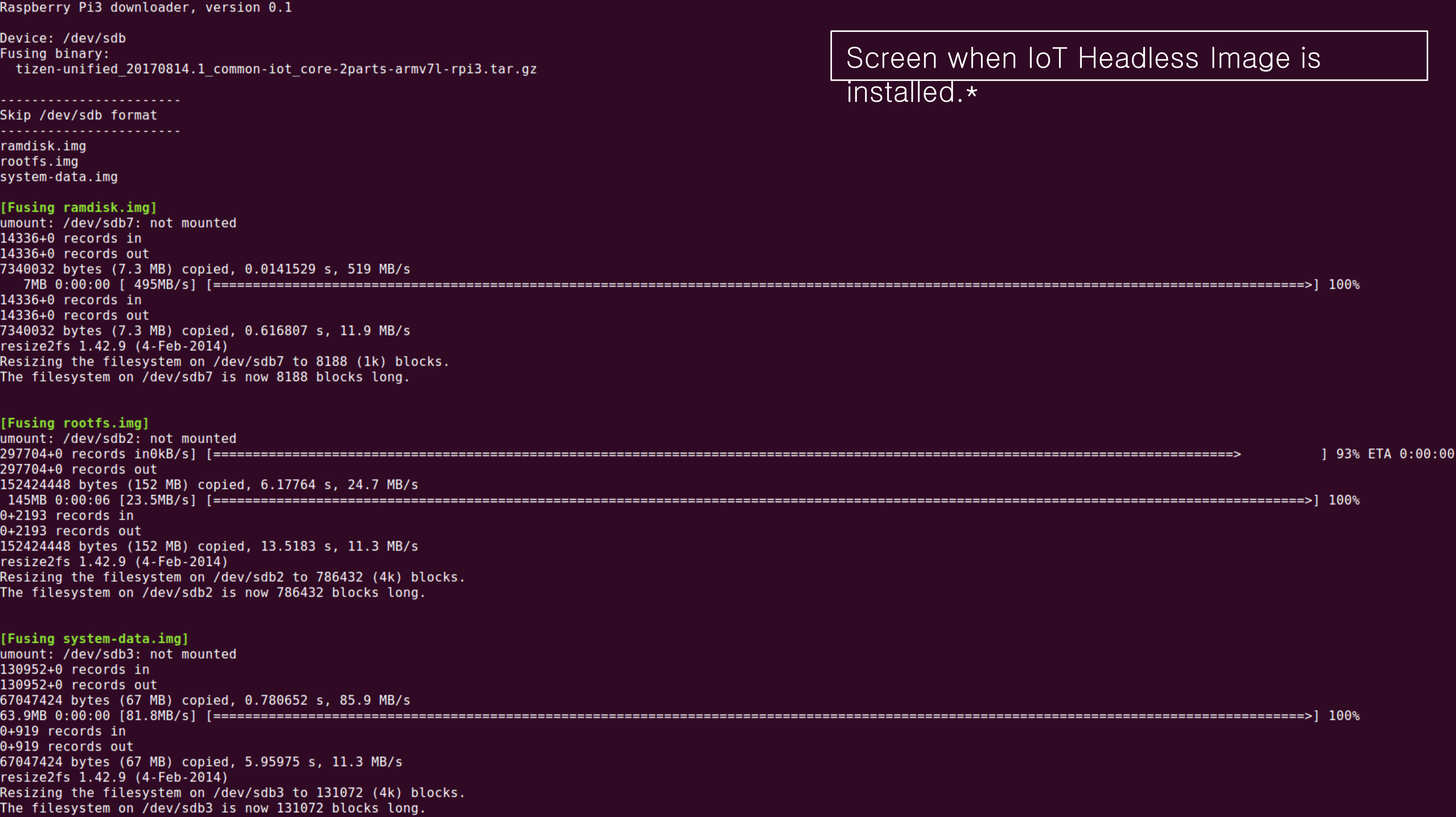
```
$ mkdir rootfs
$ sudo mount /dev/sdb2 rootfs
$ sudo mkdir -p rootfs/usr/etc/bluetooth
$ sudo cp BCM43430A1.hcd rootfs/usr/etc/bluetooth
$ sudo mkdir -p rootfs/usr/lib/firmware/brcm
$ sudo cp brcmfmac43430-sdio.* rootfs/usr/lib/firmware/brcm
$ sync
$ sudo umount rootfs
$ rmdir rootfs
```



```
Raspberry Pi3 downloader, version 0.1

Device: /dev/sdb
Fusing binary:
  tizen-unified_20170814.1_common-boot-arm64-rpi3.tar.gz

-----------------------
Skip /dev/sdb format
-----------------------
modules.img
boot.img

[Fusing boot.img]
umount: /dev/sdb1: not mounted
65536+0 records in81MB/s] [======================================================================================================================================>    ] 97% ETA 0:00:00
65536+0 records out
33554432 bytes (34 MB) copied, 4.09045 s, 8.2 MB/s
  32MB 0:00:04 [ 7.8MB/s] [==========================================================================================================================================>] 100%
65536+0 records in
65536+0 records out
33554432 bytes (34 MB) copied, 6.93111 s, 4.8 MB/s
resize2fs 1.42.9 (4-Feb-2014)
resize2fs: Bad magic number in super-block while trying to open /dev/sdb1
Couldn't find valid filesystem superblock.

[Fusing modules.img]
umount: /dev/sdb6: not mounted
29864+0 records in
29864+0 records out
15290368 bytes (15 MB) copied, 0.0488528 s, 313 MB/s
14.6MB 0:00:00 [ 299MB/s] [==========================================================================================================================================>] 100%
29864+0 records in
29864+0 records out
15290368 bytes (15 MB) copied, 1.44288 s, 10.6 MB/s
resize2fs 1.42.9 (4-Feb-2014)
Resizing the filesystem on /dev/sdb6 to 5119 (4k) blocks.
The filesystem on /dev/sdb6 is now 5119 blocks long.
```

Screen when boot image is installed.*

Screen when IoT Headless Image is installed.*



```
Raspberry Pi3 downloader, version 0.1

Device: /dev/sdb
Fusing binary:
  tizen-unified_20170814.1_common-iot_core-2parts-armv7l-rpi3.tar.gz

-----------------------
Skip /dev/sdb format
-----------------------
ramdisk.img
rootfs.img
system-data.img

[Fusing ramdisk.img]
umount: /dev/sdb7: not mounted
14336+0 records in
14336+0 records out
7340032 bytes (7.3 MB) copied, 0.0141529 s, 519 MB/s
   7MB 0:00:00 [ 495MB/s] [===========================================================================================================================================>] 100%
14336+0 records in
14336+0 records out
7340032 bytes (7.3 MB) copied, 0.616807 s, 11.9 MB/s
resize2fs 1.42.9 (4-Feb-2014)
Resizing the filesystem on /dev/sdb7 to 8188 (1k) blocks.
The filesystem on /dev/sdb7 is now 8188 blocks long.


[Fusing rootfs.img]
umount: /dev/sdb2: not mounted
297704+0 records in0kB/s] [==================================================================================================================================>         ] 93% ETA 0:00:00
297704+0 records out
152424448 bytes (152 MB) copied, 6.17764 s, 24.7 MB/s
 145MB 0:00:06 [23.5MB/s] [===========================================================================================================================================>] 100%
0+2193 records in
0+2193 records out
152424448 bytes (152 MB) copied, 13.5183 s, 11.3 MB/s
resize2fs 1.42.9 (4-Feb-2014)
Resizing the filesystem on /dev/sdb2 to 786432 (4k) blocks.
The filesystem on /dev/sdb2 is now 786432 blocks long.


[Fusing system-data.img]
umount: /dev/sdb3: not mounted
130952+0 records in
130952+0 records out
67047424 bytes (67 MB) copied, 0.780652 s, 85.9 MB/s
63.9MB 0:00:00 [81.8MB/s] [===========================================================================================================================================>] 100%
0+919 records in
0+919 records out
67047424 bytes (67 MB) copied, 5.95975 s, 11.3 MB/s
resize2fs 1.42.9 (4-Feb-2014)
Resizing the filesystem on /dev/sdb3 to 131072 (4k) blocks.
The filesystem on /dev/sdb3 is now 131072 blocks long.
```

Thank you