台揚企業集團
MTI Group

NOTE: APPROVAL ON THIS PAGE SIGNIFIES THE ENTIRE DOCUMENT APPROVED.

MTI MINI ME APP

REV	ECO NO.	DATE	CHANGE INFORMATION	AUTHOR

DRAWING NO.: MTI MINI ME APP

APPROVED BY

PAGE NO.: **1** OF **26**

A01			Initial Draft	
REVISION HISTORY				
DRAWING NO.: MTI MINI ME APP			TITLE: MTI MINI ME APP	

Index

Figure

Table

1 Purpose

This document describes the software architecture, design concept and operation flow of the MTI RFID ME GUI APP which used to control the MTI RFID MINI ME (RU-827) by Android-powered device.

2 Architecture

2.1 Android Manifest

2.1.1 USB Host

Due to the RU-827 is controlled by Android-powered device the following list describes should be implemented before working with the USB host APIs.

1. The minimum SDK version should be 12 or higher, because the USB host APIs are not supports earlier SDK.
2. Declaring the **android.hardware.usb.host** in the **<uses-feature>** element.
3. To be notified of the RU-827 attached the **android.hardware.usb.action. SB_DEVICE_ATTACHED** should be added in the **<intent-filter>** element.
4. Adding the **android.hardware.usb.action.USB_DEVICE_ATTACHED** in the **<meta-data>** element and points to device_filter.xml which defines the specific device.

2.1.2 Permissions

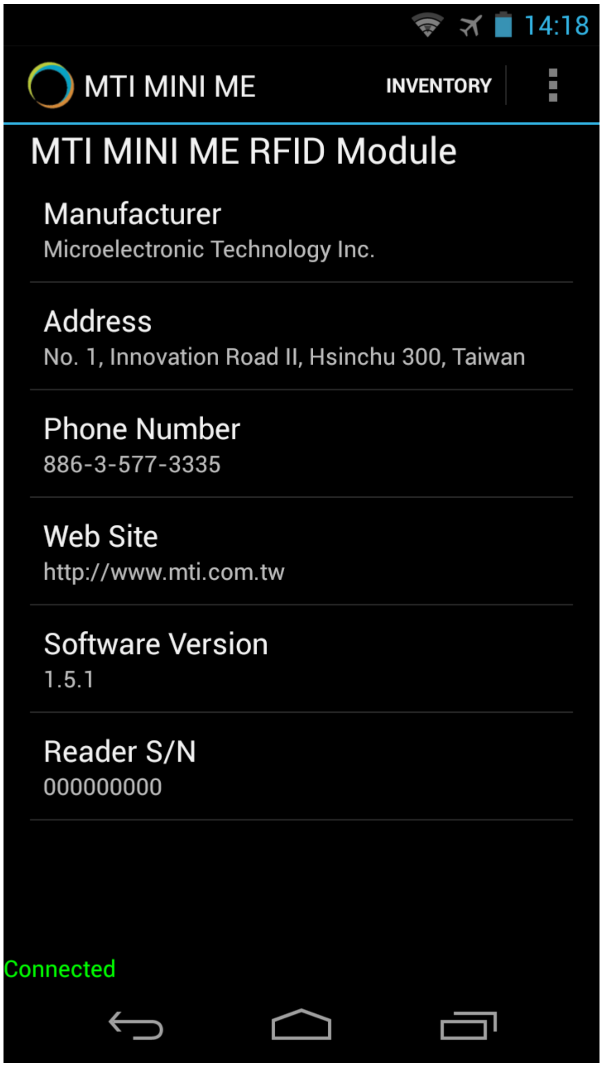
1. For searching the Tag on the internet, the **android.permission.INTERNET** should be added.
2. Because the handset device can't be upside down, add the **android.permission. SET_ORIENTATION** permission for controlling the orientation manually.

2.2 User Interface

The user interface of this App is introduced into fragment for variety screen size device. The fragment here not only provides better vision for tablet and handset, but without additional coding effort for both of them. Instead of calling activities with each other traditionally, there is only one activity in this App. The switching of screens are based on the frame in the activity and variety fragments. All of the fragments are transit in the frame which embedded in the activity. But the App title, option menu, and connection status are belonging to activity. They will be shown from beginning to end no matter how the fragment transit. The Android SDK has detail descriptions of fragment (http://developer.android.com/guide/components/fragments.html).

2.2.1 About Fragment

The Figure 1 is the screenshot of the about fragment, its layout file is frag_about.xml inherits from preference. All of the titles and summaries besides Reader S/N are defined in the strings.xml, the Reader S/N will be updated after RFID reader attached.



MTI MINI ME RFID Module

Manufacturer
Microelectronic Technology Inc.

Address
No. 1, Innovation Road II, Hsinchu 300, Taiwan

Phone Number
886-3-577-3335

Web Site
http://www.mti.com.tw

Software Version
1.5.1

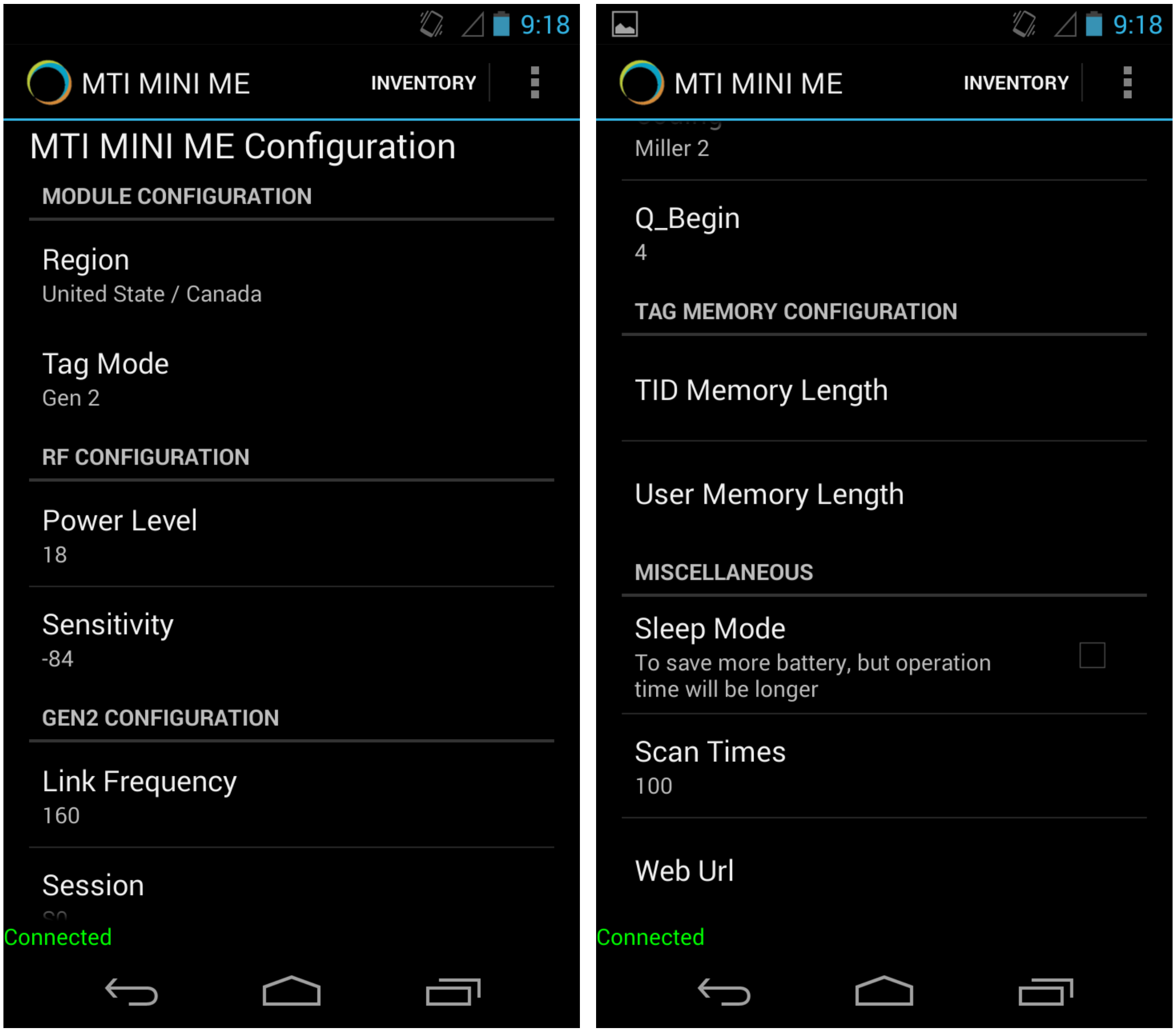
Reader S/N
000000000

Connected

Figure 1. About Fragment

2.2.2 Configure Fragment

The Figure 2 is the screenshot of the configure fragment, and its layout file is frag_config.xml also inherits from preference. The settings of this fragment effect upon the RFID reader or the operation method of this App. The Region and Tag Mode are read only values in this App, they only show what the RFID reader default setting. The Power Level, Sensitivity, Link Frequency, Session, Coding, Q_Begin, Sleep Mode, and Scan Times are the configure parameters of the RFID reader; the detail description can reference the document MTI RU-827 RFID Module Command Reference Manual. The TID Memory Length, User Memory Length, and Web Url will not affect the RFID reader settings. They only effect how this App manipulation. The first two are used to define the length of TID and user memory banks of the tag which want to access, and the last one is the web url, while long press the tag id in the tag fragment the App will based on this url to search the tag id.



MTI MINI ME Configuration

MODULE CONFIGURATION

Region
United State / Canada

Tag Mode
Gen 2

RF CONFIGURATION

Power Level
18

Sensitivity
-84

GEN2 CONFIGURATION

Link Frequency
160

Session
S0

Connected

Miller 2

Q_Begin
4

TAG MEMORY CONFIGURATION

TID Memory Length

User Memory Length

MISCELLANEOUS

Sleep Mode ☐
To save more battery, but operation time will be longer

Scan Times
100

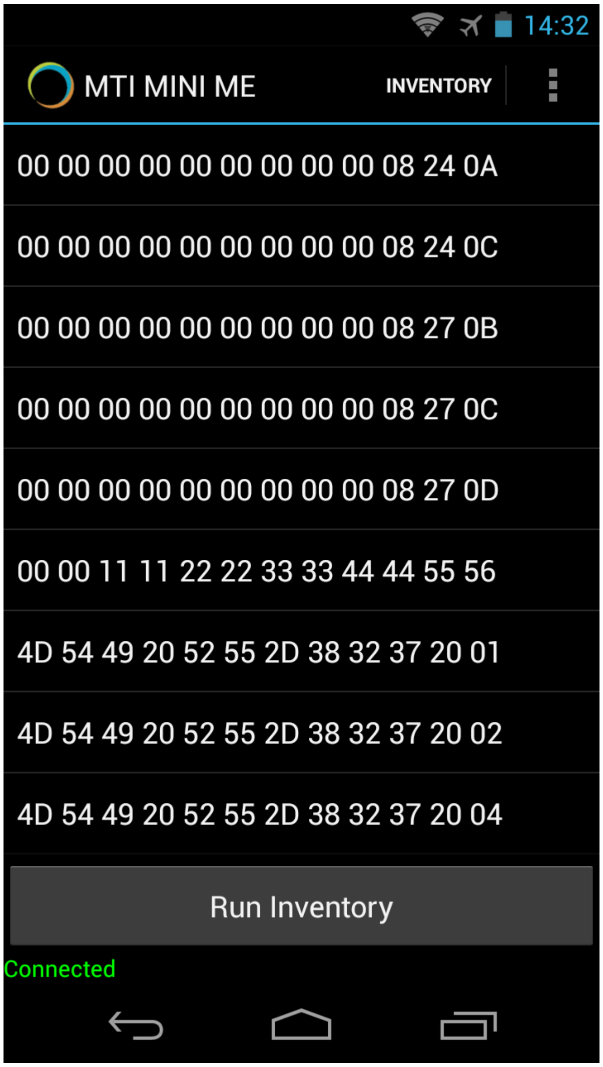
Web Url

Connected

Figure 2. Configure Fragment

2.2.3 Tag Fragment

The Figure 3 is the screenshot of the tag fragment, and its layout file is frag_tag.xml. This is a variant of the list view, it add an inventory button for scanning tags and display on the list view. Clicking one of the scanned tag on the list view, the fragment will be transit to detail fragment. Besides, long pressing it, the web fragment will display.



```
00 00 00 00 00 00 00 00 00 08 24 0A
00 00 00 00 00 00 00 00 00 08 24 0C
00 00 00 00 00 00 00 00 00 08 27 0B
00 00 00 00 00 00 00 00 00 08 27 0C
00 00 00 00 00 00 00 00 00 08 27 0D
00 00 11 11 22 22 33 33 44 44 55 56
4D 54 49 20 52 55 2D 38 32 37 20 01
4D 54 49 20 52 55 2D 38 32 37 20 02
4D 54 49 20 52 55 2D 38 32 37 20 04
```

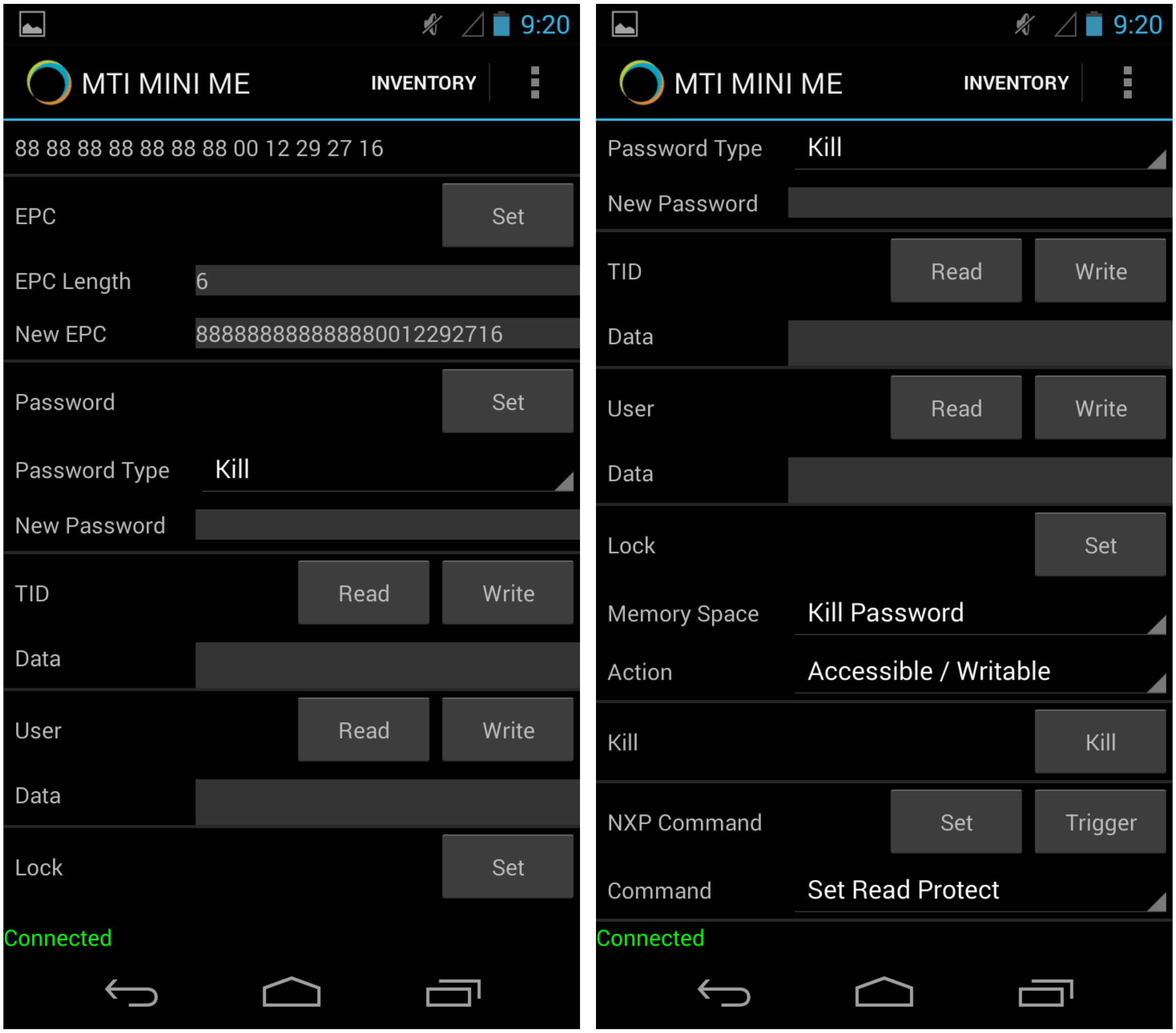
Connected

Figure 3. Tag Fragment

The preceding three fragments can transit by selecting the option menu.

2.2.4 Detail Fragment

The Figure 4 is the screenshot of the detail fragment, and its layout file is frag_detail.xml. After inventory, clicking one of a tag id on the list view will transit to this fragment. The features of this fragment are to manipulate the specific tag like read, write, kill, and lock. The detail description can reference the document MTI RU-827 RFID Module Command Reference Manual.



88 88 88 88 88 88 88 00 12 29 27 16

EPC

EPC Length: 6

New EPC: 888888888888880012292716

Password

Password Type: Kill

New Password:

TID

Data:

User

Data:

Lock

Connected

Password Type: Kill

New Password:

TID

Data:

User

Data:

Lock

Memory Space: Kill Password

Action: Accessible / Writable

Kill

NXP Command

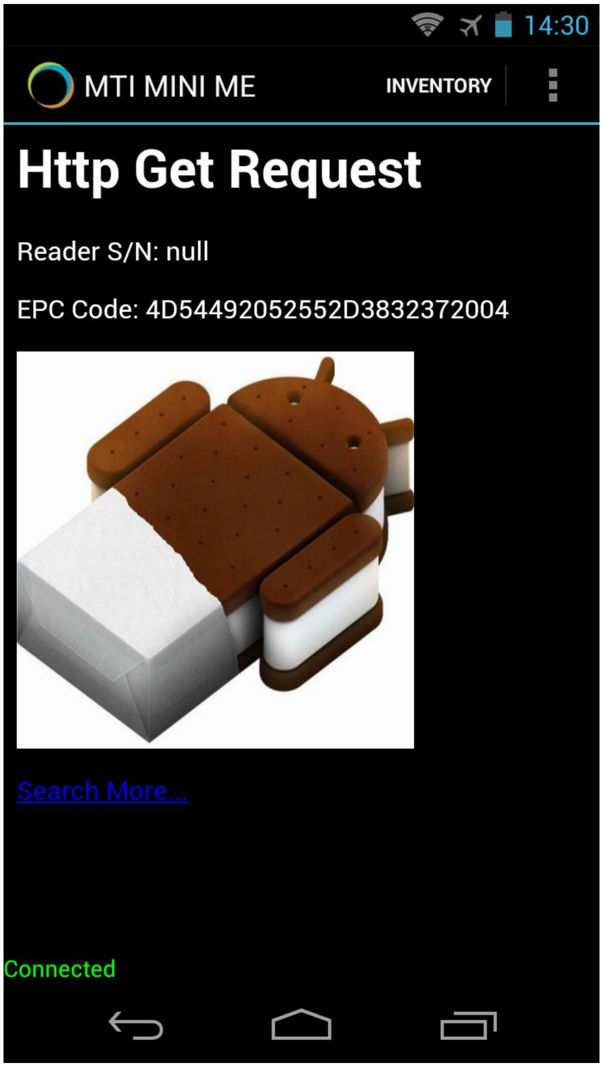
Command: Set Read Protect

Connected

Figure 4. Detail Fragment

2.2.5 Web Fragment

The Figure 5 is the screenshot of the web fragment, its layout file is frag_web.xml. After inventory, long pressing one of a tag id on the list view will transit to this fragment. There should be a web server to achieve this function.



Http Get Request

Reader S/N: null

EPC Code: 4D54492052552D3832372004

Search More...

Connected

Figure 5. Web Fragment

3 Flowchart

The flowcharts here are only illustrating the programming concept but without the life cycle of Android itself.

The Figure 6 illustrates the flow of the activity.

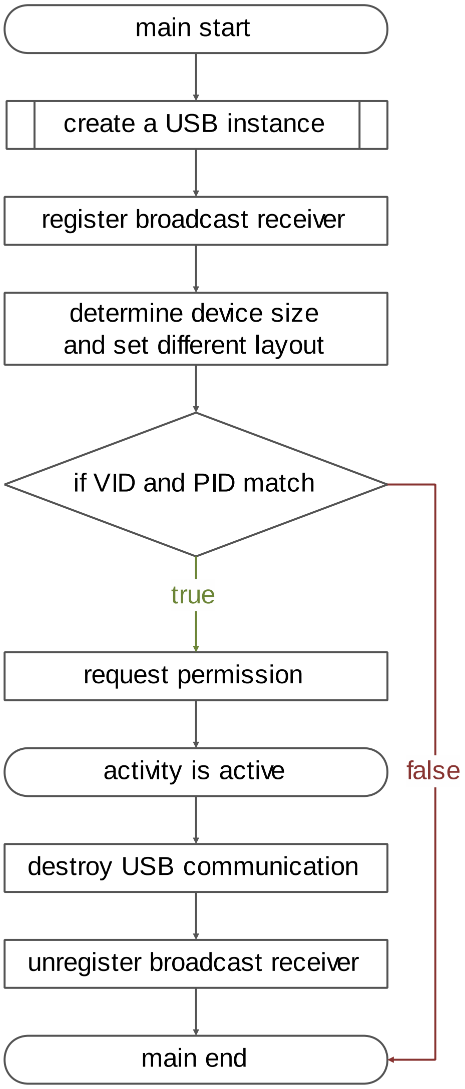


Figure 6. Main Activity Flowchart

The Figure 7 illustrates the flow of the received broadcast. There are three broadcast will be received: USB attached, USB detached, and USB permission.

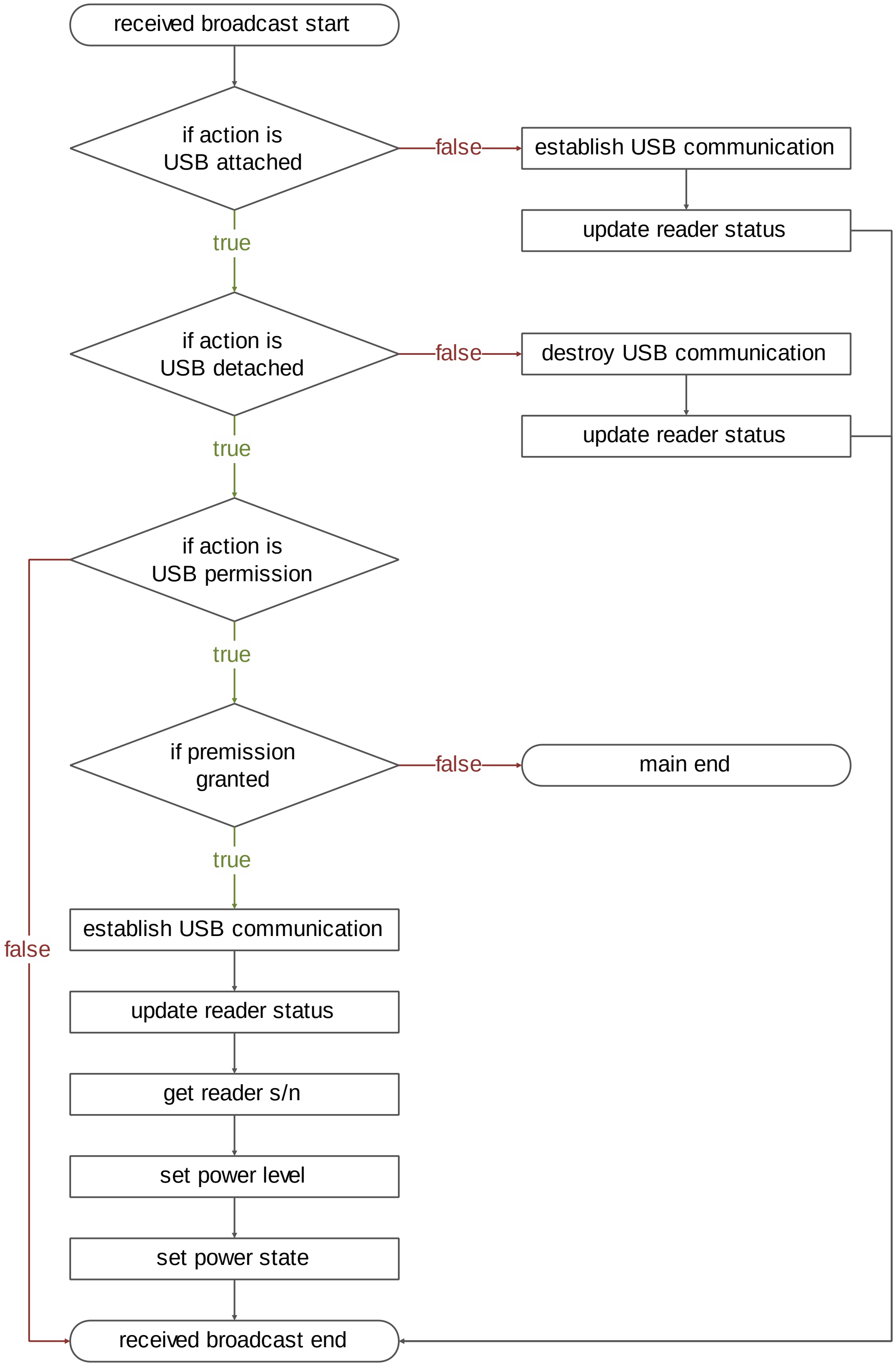


Figure 7. Received Broadcast Flowchart

The Figure 8 illustrates the flow of the fragment transition. The tag, configure, and about fragments will be transited by selecting the option menu. By the way, the detail and web fragment will be transited by clicking or long pressing a tag on the tag fragment. This flowchart is for the handset device, because of the small screen size all of the fragment should be transition. For the tablet device, because of the tag fragment is fixed on the left or top of screen. Other fragments will transit on the right or bottom screen.

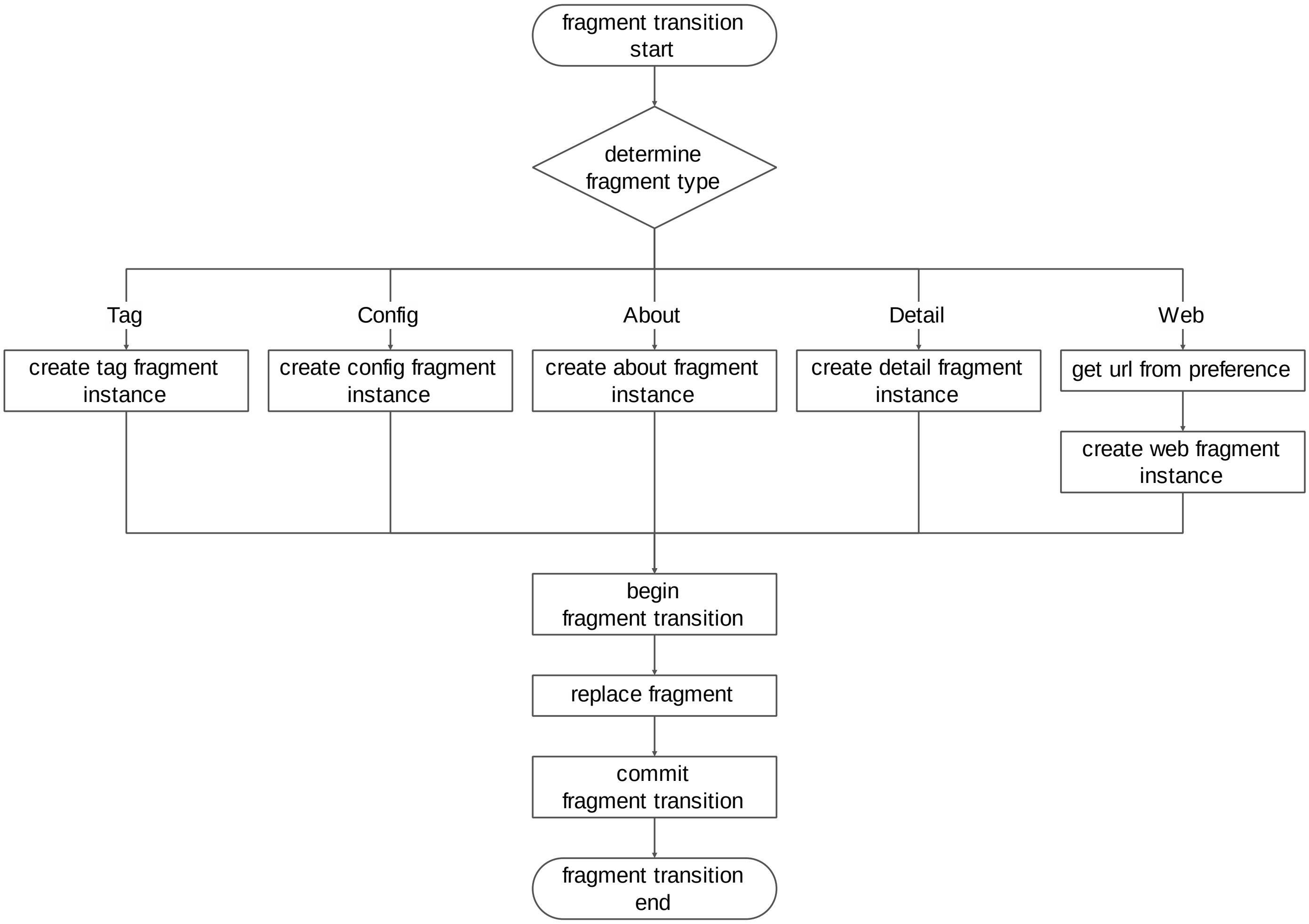

fragment transition start
determine fragment type
Tag
Config
About
Detail
Web
create tag fragment instance
create config fragment instance
create about fragment instance
create detail fragment instance
get url from preference
create web fragment instance
begin fragment transition
replace fragment
commit fragment transition
fragment transition end

Figure 8. Fragment Transition Flowchart

The Figure 9 illustrates the flow of the USB lower layer communication. This class inherits from application, it means the life cycle is the same as application's life.

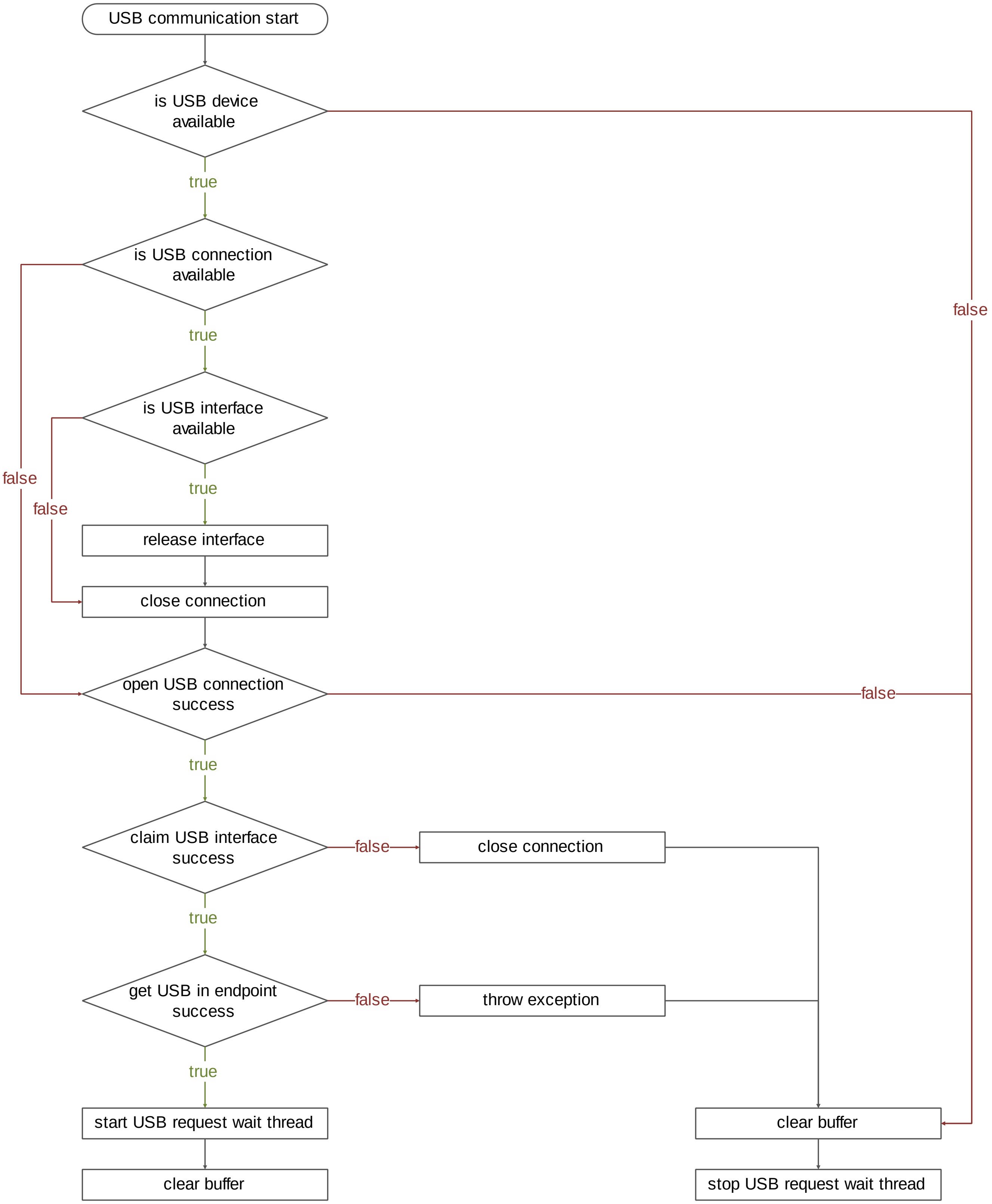

USB communication start
is USB device available
true
is USB connection available
true
is USB interface available
true
release interface
close connection
open USB connection success
true
claim USB interface success
true
get USB in endpoint success
true
start USB request wait thread
clear buffer
false
false
false
false
close connection
false
throw exception
false
clear buffer
stop USB request wait thread

Figure 9. USB Communication Flowchart

The Figure 10 illustrates the flow of the configure fragment which can access and show the configuration of the RU-827.

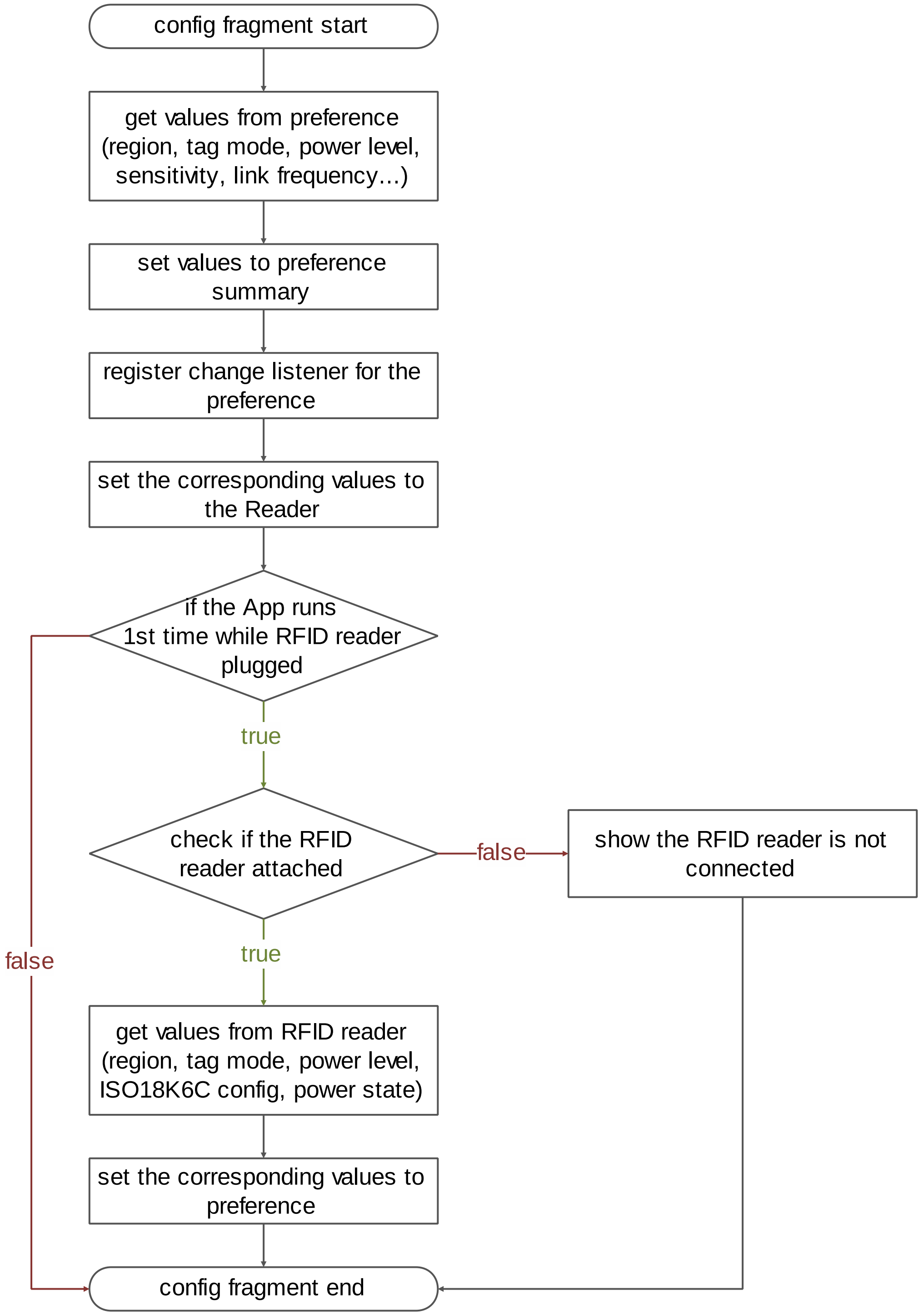

config fragment start
get values from preference (region, tag mode, power level, sensitivity, link frequency…)
set values to preference summary
register change listener for the preference
set the corresponding values to the Reader
if the App runs 1st time while RFID reader plugged
true
false
check if the RFID reader attached
false
show the RFID reader is not connected
true
get values from RFID reader (region, tag mode, power level, ISO18K6C config, power state)
set the corresponding values to preference
config fragment end

Figure 10. Configure Fragment Flowchart

The Figure 11 illustrates the flow of the tag fragment which can inventory, and transit to detail or web fragment.

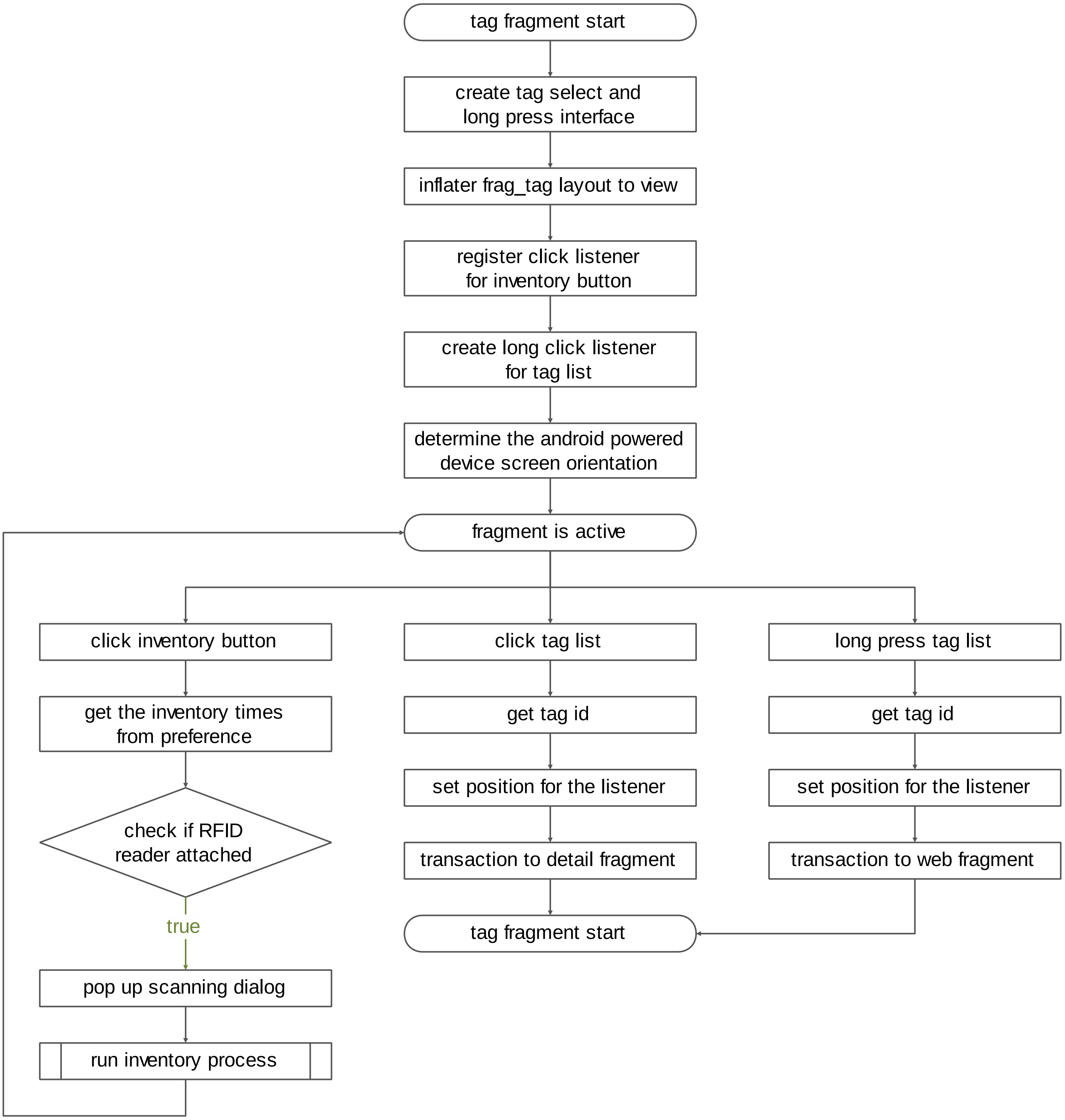

tag fragment start
create tag select and long press interface
inflater frag_tag layout to view
register click listener for inventory button
create long click listener for tag list
determine the android powered device screen orientation
fragment is active
click inventory button
get the inventory times from preference
check if RFID reader attached
true
pop up scanning dialog
run inventory process
click tag list
get tag id
set position for the listener
transaction to detail fragment
tag fragment start
long press tag list
get tag id
set position for the listener
transaction to web fragment

Figure 11. Tag Fragment Flowchart

The Figure 12 illustrates the flow of the inventory.

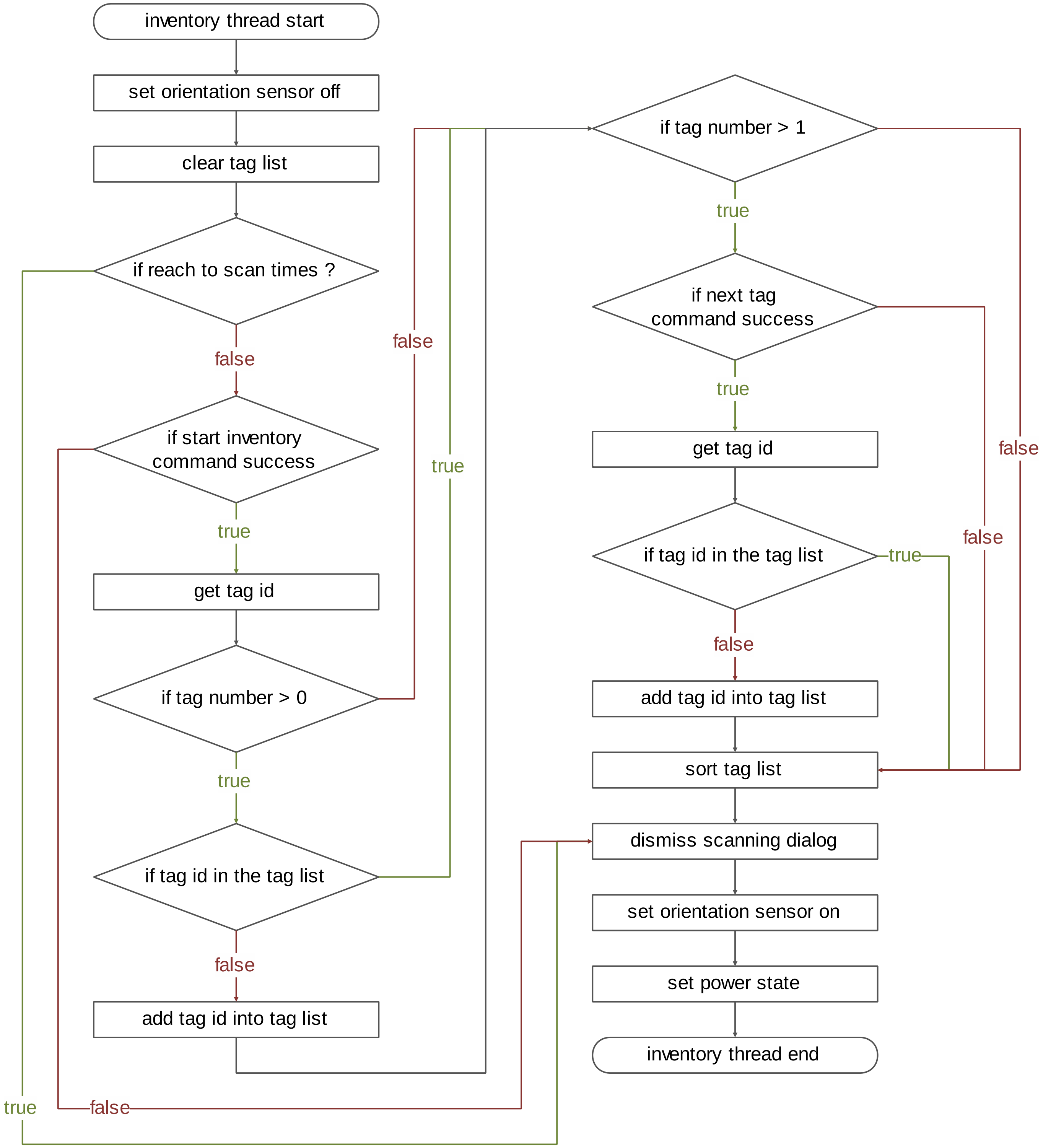


Figure 12. Inventory Flowchart

The Figure 13 illustrates the flow of the detail fragment which can access and show the tag informations.

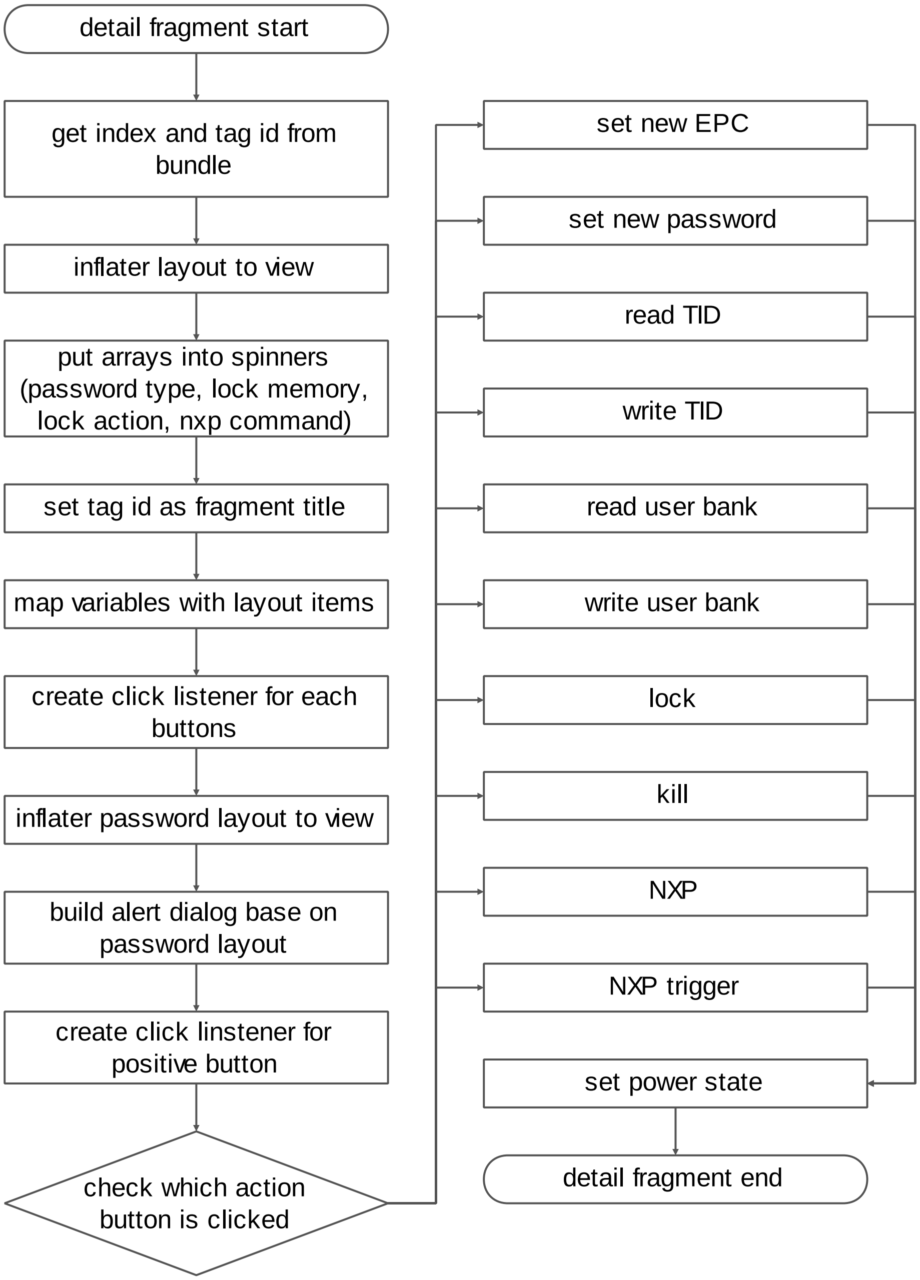

detail fragment start
get index and tag id from bundle
inflater layout to view
put arrays into spinners (password type, lock memory, lock action, nxp command)
set tag id as fragment title
map variables with layout items
create click listener for each buttons
inflater password layout to view
build alert dialog base on password layout
create click linstener for positive button
check which action button is clicked
set new EPC
set new password
read TID
write TID
read user bank
write user bank
lock
kill
NXP
NXP trigger
set power state
detail fragment end

Figure 13. Detail Fragment Flowchart

4 Flies List

The Figure 14 and 15 illustrates the files tree structure.

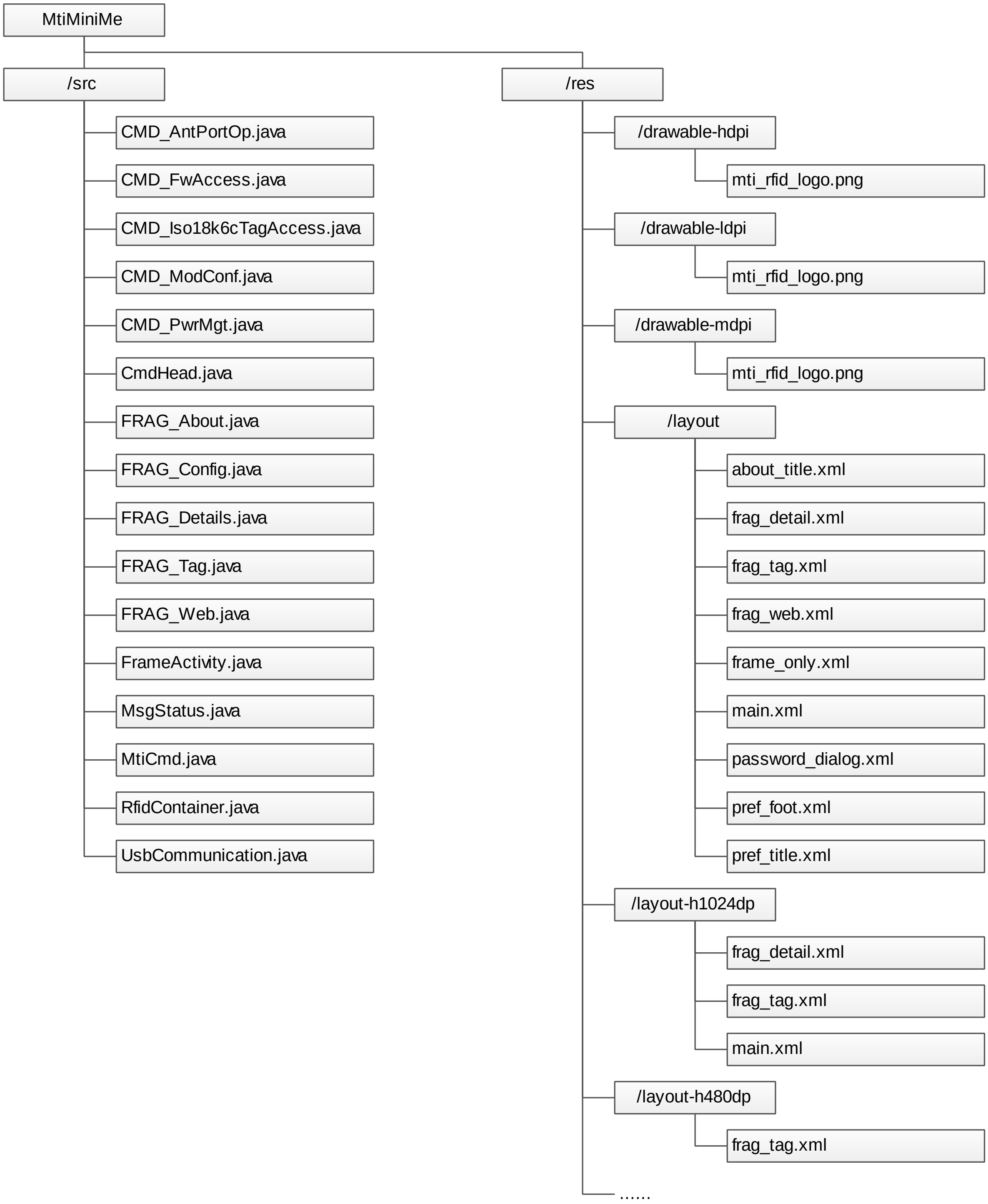


Figure 14. Files Tree Structure

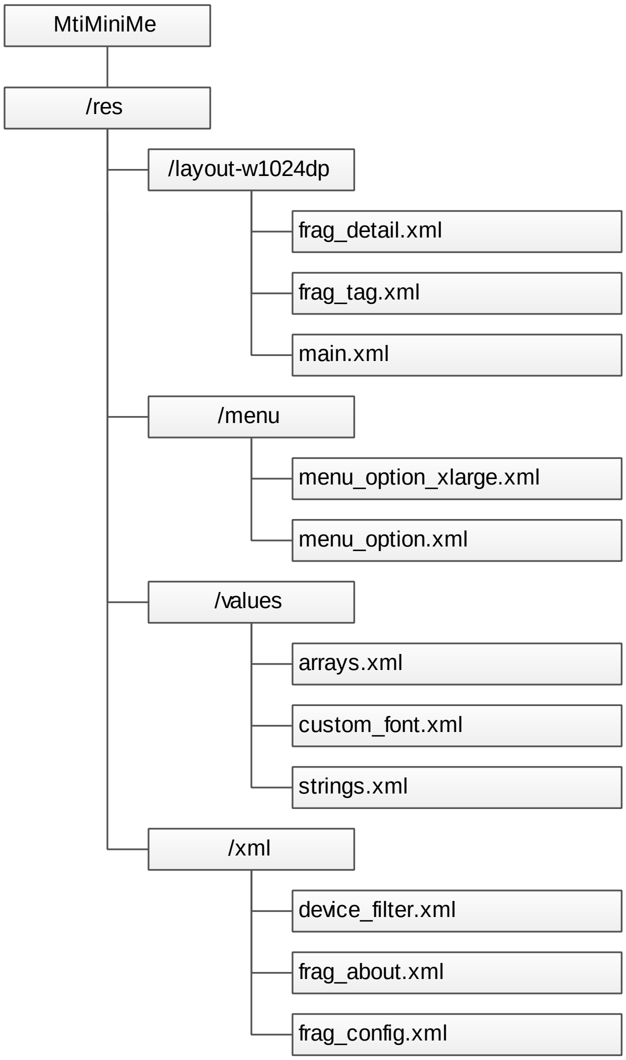


Figure 15. Files Tree Structure (Cont.)

The table below describes the purpose of files in the project.

path	file name	purpose
/	AndroidManifest.xml	Android manifest file.
/src		The source codes directory.
	CMD_AntPortOp.java	The antenna port operation classes.
	CMD_FwAccess.java	The RFID module firmware access classes.
	CMD_Iso18k6cTagAccess.java	The ISO 18000-6C tag access classes.
	CMD_ModConf.java	The RFID module configuration classes.
	CMD_PwrMgt.java	The RFID module power management classes.
	CmdHead.java	The enumeration of the command headers.
	FRAG_About.java	The RFID about fragment.
	FRAG_Config.java	The RFID configuration fragment.
	FRAG_Details.java	The tag information fragment.
	FRAG_Tag.java	The tag inventory fragment.
	FRAG_Web.java	The web fragment.
	FrameActivity.java	The functions to transaction fragments (This file is not used).
	MsgStatus.java	The enumeration of the response message status.
	MtiCmd.java	The functions to deal with sent commands and received data.
	RfidContainer.java	The entry of this App.
	UsbCommunication.java	The lower layer to communicate with USB.
/res/drawable-hdpi		This resource directory is to put high density components.
	mti_rfid_logo.png	The logo for the high density.
/res/drawable-ldpi		This resource directory is to put low density components.
	mti_rfid_logo.png	The logo for the low density.
/res/drawable-mdpi		This resource directory is to put medium density

		components.
	mti_rfid_logo.png	The logo for the medium density.
/res/layout		This resource directory is to put common layout components.
	about_title.xml	The title layout of the about fragment.
	frag_detail.xml	The layout of the tag information fragment.
	frag_tag.xml	The layout of the tag inventory fragment.
	frag_web.xml	The layout of the web fragment.
	frame_only.xml	This frame is only for handset device (This file is not used).
	main.xml	The main layout.
	password_dialog.xml	The layout of the password dialog.
	pref_foot.xml	The foot layout of the preference.
	pref_title.xml	The title layout of the preference.
/res/layout-h1024dp		The layout components resource directory for the screen height higher than 1024dp.
	frag_detail.xml	The layout of the tag information fragment.
	frag_tag.xml	The layout of the tag inventory fragment.
	main.xml	The main layout.
/res/layout-h480dp		The layout components resource directory for the screen height higher than 480dp.
	frag_tag.xml	The layout of the tag inventory fragment.
/res/layout-w1024dp		The layout components resource directory for the screen width wider than 1024dp.
	frag_detail.xml	The layout of the tag information fragment.
	frag_tag.xml	The layout of the tag inventory fragment.
	main.xml	The main layout.
/res/menu		This resource directory is to put menu components.
	menu_option_xlarge.xml	The options menu for xlarge screen.

	menu_option.xml	The options menu for normal screen.
/res/values		This directory is to put constant values.
	arrays.xml	The strings array for spinner.
	custom_font.xml	The styles and themes for different layout.
	strings.xml	The maps of variables and constant strings.
/res/xml		This directory is to put some special and customize xml file.
	device_filter.xml	There are VID and PID defined in it.
	frag_about.xml	The preference of the RFID about.
	frag_config.xml	The preference of the RFID configuration.

Table 1. Files List

There are some replicated files in the different directories under the res directory. They have the same file name but the contents are different. They have variety layout and font size to compliant corresponding screen size, density, or orientation. The Android SDK has detail descriptions of supporting multiple screens (http://developer.android.com/guide/practices/screens_support.html).

5 Appendix

5.1 Git Branch

The Git is used to control the source code version, and this document describes the branch **master**. There are 6 branches in this Git repository. The branches besides impinj are all AMS solution, in other words only the impinj branch is Impinj solution. The table 2 sketches the usage of each branch.

branch	Usage
impinj	This branch is used for Impinj solution product, ex. RU-824, RU-861 etc.
itotani	This branch is specific for Itotani which can accept 0x3993 PID and 0x24e9 VID product.
master	The main trunk which this document describes.
orlando	This branch is customized for Universal Studio Orlando.
rfidShow	This branch is designed for RFID show.
service	This branch is not a complete one, which has some Android service codes in it.

Table 2. Git branches

5.2 Publish to Google Play

In order to publish this APP to Google play, there are some informations should know:

To export .apk file, there should be a keystore. The setting of it is as follows:

field	value
File name	mtirfid.keystore
Password	mti50-5000
Alias	dp
Password	mti50-5000
Validity	40
First and Last Name	sw mti
Organization Unit	eng
Organization	mti

City or Locality	hsinchu
State or Province	hsinchu
Country Code (xx)	TW

Table 3. Keystore Information

6 Reference

[1] Android SDK (https://developer.android.com/guide/index.html)

[2] MTI RU-827 RFID Module Command Reference Manual.