Programming Guide

UnitechRFID

unitech electronics Documentation
Issue1, revision 1
August 2024

unitech electronics co., ltd.
5F., No.135, Ln. 235, Baoqiao Rd., Xindian Dist., New Taipei City 231, Taiwan (R.O.C.)
• Tel: + 886 2 89121122 • Fax: +886 2 89121810 • website: www.ute.com

Revision History

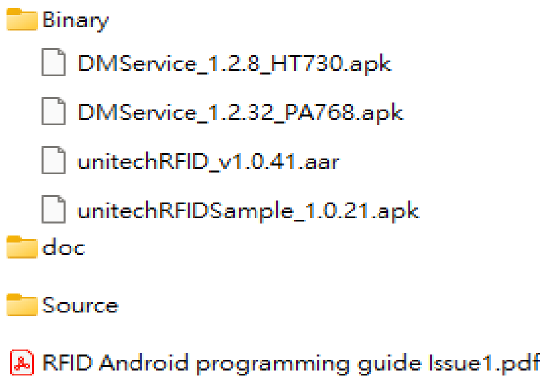
Release	Revision	Date	Changes
1	0	2022-03-14	Release for unitechRFID
1	1	2024-08-05	Support RG768

Table of Contents

1. Preface

This document describes the functionalities of the UnitechRFID. The SDK is delivered through the JAVA-based library “unitechRFID.aar” for user with the supported portable Android RFID readers from unitech. Those supported RFID readers use the modules ST25RU3993 and RM100.

The figure below shows the structure of SDK. The *Binary* folder includes the library and one sample application, and the *Source* folder contains the source codes of sample application, and the doc folder contains the javaDoc of the library “unitechRFID.aar”.



```
Binary
    DMService_1.2.8_HT730.apk
    DMService_1.2.32_PA768.apk
    unitechRFID_v1.0.41.aar
    unitechRFIDSample_1.0.21.apk
doc
Source
RFID Android programming guide Issue1.pdf
```

Folders & Files	Description
/UnitechRFID	The SDK package
/UnitechRFID/Binary/	The Binary folder contains the RFID Library (AAR) and the dependency apps & sample app.
/UnitechRFID/doc/	The doc folder contains the JAVADOC of the unitechRFID.aar library file. Please load index.html into your browsers to view
/UnitechRFID/Binary/unitechRFID_v1.0.41.aar	The unitechRFID.aar library file. Include this library in your Android Studio project to access the API functions.
/UnitechRFID/Binary/DMService_1.2.8_HT730.apk	The dependency app DMService required for HT730. Please make sure this APK is installed on the host device before using the unitechRFID.aar

/UnitechRFID/Binary/DMService_1.2.32_PA768.apk	The dependency app DMService required for PA768. Please make sure this APK is installed on the host device before using the unitechRFID.aar
/UnitechRFID/Binary/unitechRFIDSample_1.0.21.apk	The sample application shows the use of the unitechRFID.aar, The source code of this application is also included in this package for studying.
/UnitechRFID/Source/	The source code of the unitechRFIDSample application.

Note: The library "unitechRFID.aar" needs

- ***HT730: DMService V1.2.8 or later version***
- ***PA768: DMService V1.2.32 or later version***

to support. You could get it in the Binary folder.

Note: The BaseReader(HT730Reader, RG768Reader, RP902Reader) should be initialized in non-main thread.

Note: The application needs API version 28 or higher.

1.1 Prerequisites

The SDK supports the following RFID Reader peripherals:

- RP902 with an Android 9 or above.
- HT730 with Android 10.
- PA768 with Android 12

Note about the Dependency App "DMService"

The DMService application is one of the unitech helper applications that provide the functions of the UnitechSDK. The "unitechRFID.aar" UnitechRFID uses the UnitechSDK for some of its functionalities such as Power Settings. It is recommended that your application can take care of the installation of this DMService APK if the host device does not already have it installed, or if the minimum version required are not correct.

The UnitechRFID requires

- HT730: DMService V1.2.8 or later (inclusive)
- PA768: DMService V1.2.32 or later (inclusive)

to be installed on the supported host Android devices. The DMService APK files for the supported models can be found from the Binary folder inside the SDK package.

Note about Multithreaded Applications
The BaseReader class should be initilized in non-main thread.

2. uniechRFID API

The TransportBluetooth used to connect to RFID module by Bluetooth. Below is the sample code, and you can also find the details in the SampleFragment.java in the Source folder.

```
TransportBluetooth tb = new TrandportBluetooth(DeviceType.RP902, “RP902”,
“11:22:33:44:55:66”);
BaseReader baseReader = new RP902Reader(tb);
baseReader.addListener(this);
baseReader.connect();
```

Connect to RFID module with UART. Below is the sample code, and you can also find the details in the SampleFragment.java in the Source folder.
HT730:

```
BaseReader baseReader = new HT730Reader(getApplicationContext());
baseReader.addListener(this);
baseReader.connect();
```

RG768

```
BaseReader baseReader = new RG768Reader(getApplicationContext());
baseReader.addListener(this);
baseReader.connect();
```

Note about Multithreaded Applications
The BaseReader class should be initilized in non-main thread.