

# **Programming Guide**

## ***RFID Android SDK***

unitech electronics Documentation  
Issue1, revision 2  
July. 2020

## Revision History

Release	Revision	Date	Changes
1	0	2019-12-11	First release
1	1	2020-03-30	Update SDK file structure and MercuryAPI document
1	2	2020-07-02	Update Preface and Pogo Library

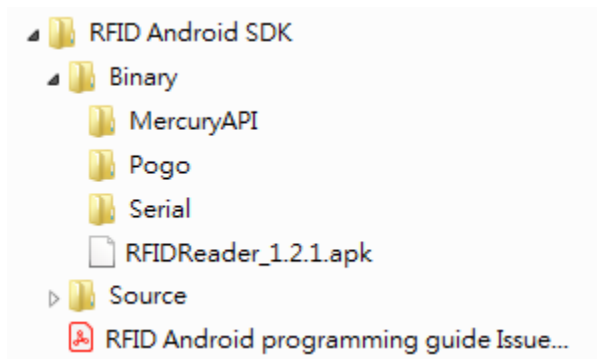
# Table of Contents

1.	PREFACE .....	1
2.	POGO LIBRARY .....	1
2.1	TURN ON POWER.....	1
2.2	TURN OFF POWER.....	2
3.	SERIAL LIBRARY .....	2
4.	MERCURY API.....	2

## 1. Preface

This document describes the RFID library of unitech portable Android RFID reader. The reader uses ThingMagic RFID module, and thus all RFID function calls are based on Mercury API.

RFID reader uses pogo pin to make serial connection to communicate with the Android device, so we provide pogo library, serial library and customized Mercury API in the SDK. The figure below shows the structure of SDK. The *Binary* folder includes three libraries and one demo application, and *Source* folder contains the source codes of demo application.



## 2. Pogo Library

Pogo library pogolib.jar can be found under /Binary/Pogo/. Please put this library file into your project and import it as library. In the sample project the library is in the folder /app/libs. Below we list the functions provided in Pogo library.

### 2.1 Turn on power

**Function Description:**

Set the power pin to high.

**Function call:**

```
void powerOn ()
```

**Return:**

None.

## 2.2 Turn off power

**Function Description:**

Set the power pin to low.

**Function call:**

void powerOff ()

**Return:**

None.

## 3. Serial Library

Serial library is JNI library. Please copy jni folder under /Binary/Serial/ folder to your project and specify the path in gradle. For example, we put jni folder at root folder of the project in the demo app, and have following specified in build.gradle.

```
android {  
    .....  
    sourceSets {  
        main {  
            jniLibs.srcDirs = ['jni']  
        }  
    }  
    .....  
}
```

Serial connection and communication are already embedded into customized Mercury API, so you can call function calls defined in MercuryAPI to make the connection and send command to the RFID module.

## 4. Mercury API

Customized Mercury API is revised from Mercury API to support serial communication. Therefore,

please specify the scheme as “ute” and Factory object as “new SerialTransportRS232.Factory()” while setting serial port of the reader. The URI should be “ute:///dev/ttyHS3” while making the connection. Below is the sample code, and you can also find the details in the ReaderConnect.java in the Source folder.

```
private static Reader reader;
if (reader == null) {
    Reader.setSerialTransport("ute", new SerialTransportRS232.Factory());
    reader = Reader.create("ute:///dev/ttyHS3");
}
reader.connect();
```

All other RFID related functions are kept the same. Please refer the official Mercury API programming guide and Java doc under the folder Binary/MercuryAPI.