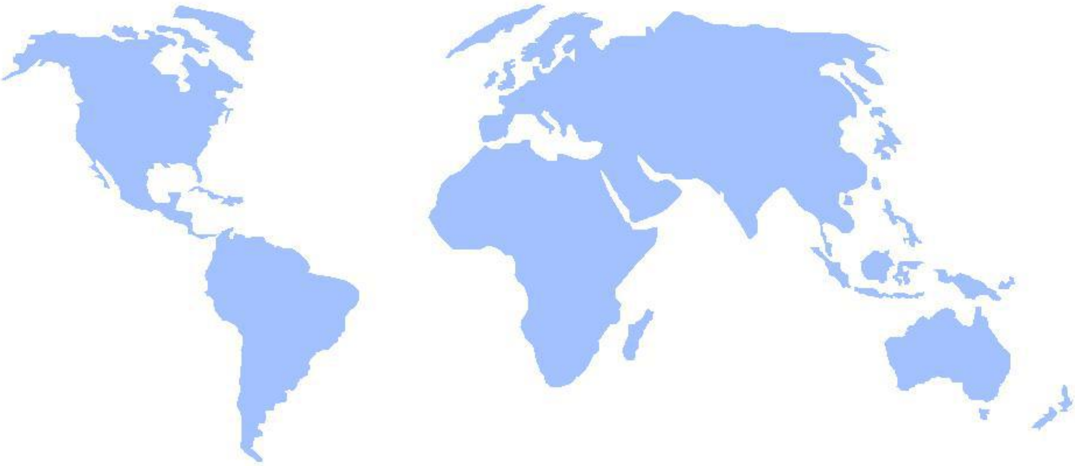
Android UHF Interface User Manual



www.seuic.com

Revising index

Version	Update date	Revising details	By
1.0	9/1/2016	Create the Chinese version	Xiaodong Feng

Content

Abstract

This document is created to provide the users the manual for the interface related to D500 device, to help them better use the special functions of the terminal product.

Hardware platform

SDK is applicable to the devices listed below：

- AUTOID 9U

Without further limits definition in functional interface instruction, its using range shall be redeemed as applicable to all.

Software platform

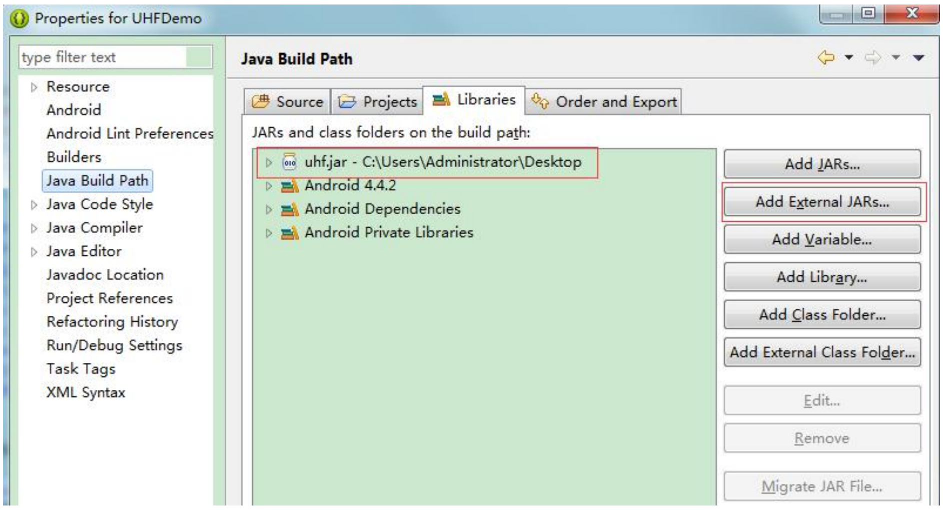
SDK is based on Android 4.4, and it supports the eclipse development tool.

Object oriented

It's designed for the developers who would like to use UHF module besides the Android standard functions.

Quote method

1. Take Eclipse for example, right click Program, then Properties, Java Build Path, Libraries, Add External JARs, select uhf.jar.



2. AndroidManifest.xml uses *uses-library* to ID the *uhf.jar* is quoted package.

```
   <application
……
          <uses-library
               android:name="com.seuic.uhf" />
          ……
    </application>
```

UHF Interface

Package name	com.seuic.uhf
Package file	uhf.jar
System package	Yes
Class	UHFService
Function	Provide the control interface related to UHF

UHFService class

Using method

```
import com.seuic.uhf.UHFService;
UHFService mDevice = UHFService.getInstance();
```

Function interface

Function	Specification
open	Switch UHF on
close	Switch UHF off
getFirmwareVersion	Get the firmware version number
getTemperature	Get the temperature
getPower	Get the power
setPower	Set the power
getRegion	Get field
setRegion	Set field
inventoryOnce	Singe tag reading
inventoryStart	Start continuous tag reading
inventoryStop	Stop continuous tag reading
getTagIDCount	Get the quantity of the continuous tags reading
getTagIDs	Get the continuous tag reading ID
readTagData	Read the label

writeTagData	Write the label

1. Switch UHF on

boolean open ()

Parameters

non

Return value

boolean; true for success and false for failure

2. Switch UHF off

void close ()

Parameters

Non

Return value

non

3. Get the firmware version number

String getFirmwareVersion ()

Parameters

non

Return value

String: not null for success and null for failure

4. Get the temperature

String getTemperature ()

Parameters

non

Return value

String: not null for success and null for failure (℃)

5. Get the power

int getPower ()

Parameters

non

Return value

int; not-o for success and 0 for failure (0~30 dBm)

6. Set the power

boolean setPower (int power)

Parameters

power

Set the power as (0~30 dBm)

Return value

boolean; true for success and false for failure

7. Get field

String getRegion ()

Parameters

non

Return value

String; non blank for success and null for failure

Range value including："FCC" , "ETSI", "China1" , "China2"

8. Set field

boolean setRegion (String region)

Parameters

region

Range value including："FCC" , "ETSI", "China1" , "China2"

Return value

boolean; true for success and false for failure

9. Singe tag reading

boolean inventoryOnce(EPC epc, int timeout)

Parameters

epc

EPC class oriented object

timeout

Order time out (0~500 ms)

Return value

boolean; true for success and false for failure

10. Start continuous tag reading

boolean inventoryStart ()

Parameters

non

Return value

boolean; true for success and false for failure

11. Stop start tag reading

boolean inventoryStop ()

Parameters

non

Return value

boolean; true for success and false for failure

12. Get the quantity of the continuous tags reading

int getTagIDCount ()

Parameters

non

Return value

int; non-0 for the quantity of the labels got, and 0 for noting getting any label

13. Get the continuous tag reading ID

List<EPC> getTagIDs ()

Parameters

non

Return value

List<EPC>; The List collection of all EPC oriented objects

14. Read the label

boolean readTagData (byte[] Epc, byte[] PassWord, int Bank, int Offset ,int Len, byte[] Data)

Parameters

Epc

Label ID（PC+EPC）

PassWord

Visit password (length as 4 bytes)

Bank

Label saving area(0 for password,1 for EPC, 2 for TI and 3 for users)

Offset

Starting address (unit as byte)

Len

the length to read, with unit as byte

Data

save the data to be written in the label

Return value

boolean; true for success and false for failure

15. Write the label

boolean writeTagData (byte[] Epc, byte[] PassWord, int Bank, int Offset ,int Len, byte[] Data)

Parameters

Epc

Label ID（PC+EPC）

PassWord

Visit password (length as 4 bytes)

Bank

Label saving area(0 for password,1 for EPC, 2 for TI and 3 for users)

Offset

Starting address (unit as byte)

Len

the length to read, with unit as byte

Data

save the data to be written in the label

Return value

boolean; true for success and false for failure

EPC class

Class member variable

```
byte[] id;
int len;
int rssi;
int count;
```

Using method

```
import com.seuic.uhf.EPC;
EPC epc = new EPC();
```

Function interface

1. Get the ID data from EPC oriented object

String getId()

Parameters

non

Return value

String; a hexadecimal string

2. Match the ID number

boolean equals(Object obj)

Parameters

EPC oriented object

Return value

boolean; true for success and false for failure