Application Note

AN1011-00, 2014-03-14

Android Keyboard Wedge

Notice

Registered Trademark

ARETE
mobile

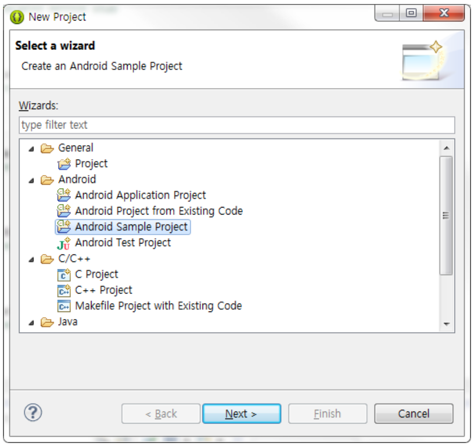
ARETE mobile is launched by PHYCHIPS Inc. for mobile device and currently registered worldwide with its ARETE mobile trade mark legally protected by law.

This document is produced by PHYCHIPS Inc. and protected by Copyright Laws.
Please note that there may be possible mistake or omission of information in this document.

Revision History

Version	Date	Changes
00	2014-03-14	Initially released

- Keyboard wedge is a virtual keyboard that allows keyboard data from a non-keyboard device. If you can simulate keyboard, you can input wanted data to anywhere you want. For example, a bar code interface is the most well-known keyboard wedge software. A bar code scanner can simulate keyboard. If you scan a bar code, bar code data can be pasted to notepad, web page, data base, etc.
- In Android system, keyboard wedge can be implemented with the Soft Keyboard sample app that is provided by Google SDK.



New Project

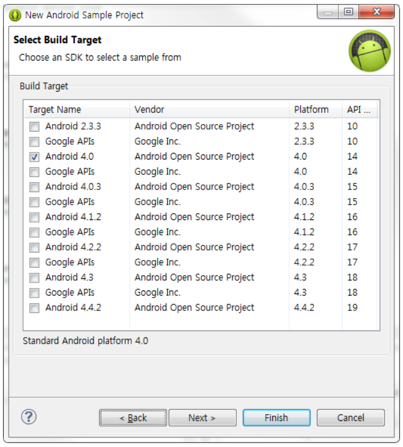
Select a wizard

Create an Android Sample Project

Wizards:

type filter text

- General
  - Project
- Android
  - Android Application Project
  - Android Project from Existing Code
  - Android Sample Project
  - Android Test Project
- C/C++
  - C Project
  - C++ Project
  - Makefile Project with Existing Code
- Java



New Android Sample Project

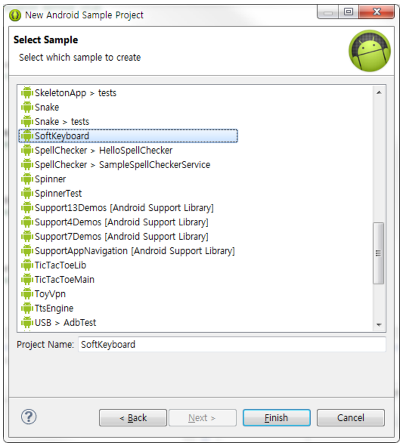
Select Build Target

Choose an SDK to select a sample from

Build Target

Target Name	Vendor	Platform	API ...
□ Android 2.3.3	Android Open Source Project	2.3.3	10
□ Google APIs	Google Inc.	2.3.3	10
☑ Android 4.0	Android Open Source Project	4.0	14
□ Google APIs	Google Inc.	4.0	14
□ Android 4.0.3	Android Open Source Project	4.0.3	15
□ Google APIs	Google Inc.	4.0.3	15
□ Android 4.1.2	Android Open Source Project	4.1.2	16
□ Google APIs	Google Inc.	4.1.2	16
□ Android 4.2.2	Android Open Source Project	4.2.2	17
□ Google APIs	Google Inc.	4.2.2	17
□ Android 4.3	Android Open Source Project	4.3	18
□ Google APIs	Google Inc.	4.3	18
□ Android 4.4.2	Android Open Source Project	4.4.2	19

Standard Android platform 4.0



New Android Sample Project

Select Sample

Select which sample to create

- SkeletonApp > tests
- Snake
- Snake > tests
- SoftKeyboard
- SpellChecker > HelloSpellChecker
- SpellChecker > SampleSpellCheckerService
- Spinner
- SpinnerTest
- Support13Demos [Android Support Library]
- Support4Demos [Android Support Library]
- Support7Demos [Android Support Library]
- SupportAppNavigation [Android Support Library]
- TicTacToeLib
- TicTacToeMain
- ToyVpn
- TtsEngine
- USB > AdbTest

Project Name: SoftKeyboard

- After including sample project into your eclipse workspace, you just add RFID function key and keyboard layout.

- To turn on the RFID reader, add code below in Softkeyboard.java

```
@Override public void onWindowShown()
   {
      super.onWindowShown();

     try
     {
     RcpApi.setRcpEvent(this);
     RcpApi.open();
     setVolumeMax();
     Thread.sleep(200);
     }
     catch (Exception e)
     {
            e.printStackTrace();
     }
   }
```

To turn the RFID reader off,

```
@Override
public void onWindowHidden()
{
        super.onWindowHidden();
        try
        {
        RcpApi.close();
        }
        catch (Exception e)
        {
        e.printStackTrace();
        }
}
```

For handling tag IDs,

```
@Override
public void onTagReceived(int[] arg0)
{
mRcpOn = false;
String tag = RcpLib.int2str(arg0);
System.out.println("tag :  " + tag);
Message message = new Message();
message.what = 0;
message.obj = tag;
handler.sendMessage(message);
}

Handler handler = new Handler()
{
        public void handleMessage(Message msg)
        {
                switch(msg.what)
                {
                case 0:
                String str = (String)msg.obj;
                getCurrentInputConnection().commitText(str,1);
                break;
                default:
                break;
                }
        }
};
```

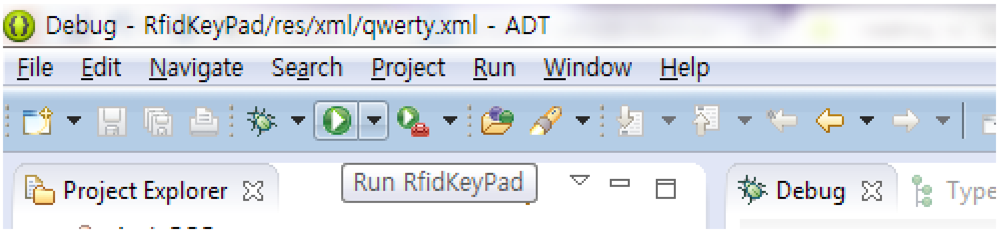
To get RFID key input,

```
public void onKey(int primaryCode, int[] keyCodes)
{
        if (isWordSeparator(primaryCode))
        {

                ...

        } else if (primaryCode == -99)
        { // RFID button
        try
        {
        if(!mRcpOn)
        {
        mRcpOn = true;
        RcpApi.startReadTags(1, 0, 0);
        }
        } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
        }
        } else {
        handleCharacter(primaryCode, keyCodes);
        }
}
```

To add the RFID key to keyboard layout, modify res/xml/qwerty.xml

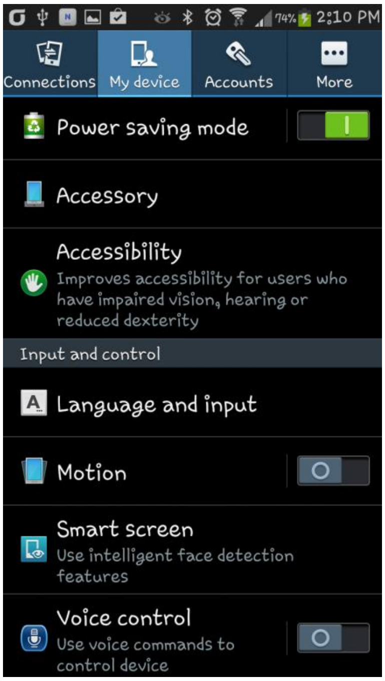
```
<Keyboard xmlns:android="http://schemas.android.com/apk/res/android"
    android:keyWidth="10%p"
    android:horizontalGap="0px"
    android:verticalGap="0px"
    android:keyHeight="@dimen/key_height"
    >
                                             ...

 <Row android:rowEdgeFlags="bottom">
        <Key android:codes="-3" android:keyIcon="@drawable/sym_keyboard_done"
                android:keyWidth="10%p" android:keyEdgeFlags="left"/>
        <Key android:codes="-2" android:keyLabel="123" android:keyWidth="15%p"/>
        <Key android:codes="32" android:keyIcon="@drawable/sym_keyboard_space"
                android:keyWidth="30%p" android:isRepeatable="true"/>
        <Key android:codes="46,44" android:keyLabel=". ,"
                android:keyWidth="15%p"/>
        <Key android:codes="-99" android:keyLabel="RFID"
                android:keyWidth="20%p"/>
        <Key android:codes="10" android:keyIcon="@drawable/sym_keyboard_return"
                android:keyWidth="10%p" android:keyEdgeFlags="right"/>
    </Row>
</Keyboard>
```

Click Run and install app. This app has no launcher activity. Never mind message.


Debug - RfidKeyPad/res/xml/qwerty.xml - ADT
File Edit Navigate Search Project Run Window Help
Run RfidKeyPad
Project Explorer
Debug
Type

```
[2014-03-14 13:37:45 - RfidKeyPad] ------------------------------
[2014-03-14 13:37:45 - RfidKeyPad] Android Launch!
[2014-03-14 13:37:45 - RfidKeyPad] adb is running normally.
[2014-03-14 13:37:45 - RfidKeyPad] No Launcher activity found!
[2014-03-14 13:37:45 - RfidKeyPad] The launch will only sync the application package on the
device!
[2014-03-14 13:37:45 - RfidKeyPad] Performing sync
[2014-03-14 13:37:45 - RfidKeyPad] Automatic Target Mode: using device '42f0fb05f6cf7f83'
[2014-03-14 13:37:45 - RfidKeyPad] Uploading RfidKeyPad.apk onto device '42f0fb05f6cf7f83'
[2014-03-14 13:37:45 - RfidKeyPad] Installing RfidKeyPad.apk...
[2014-03-14 13:37:48 - RfidKeyPad] Success!
[2014-03-14 13:37:48 - RfidKeyPad] \RfidKeyPad\bin\RfidKeyPad.apk installed on device
[2014-03-14 13:37:48 - RfidKeyPad] Done!
```

Select RFID Soft Keyboard and select it as default.



Power saving mode

Accessory

Accessibility
Improves accessibility for users who have impaired vision, hearing or reduced dexterity

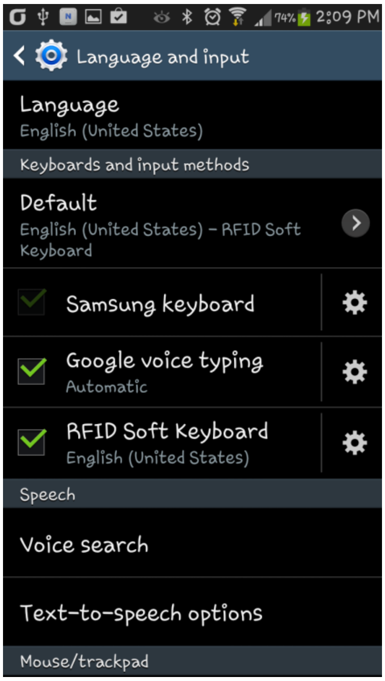
Input and control

Language and input

Motion

Smart screen
Use intelligent face detection features

Voice control
Use voice commands to control device



Language and input

Language
English (United States)

Keyboards and input methods

Default
English (United States) - RFID Soft Keyboard

□ Samsung keyboard

☑ Google voice typing
Automatic

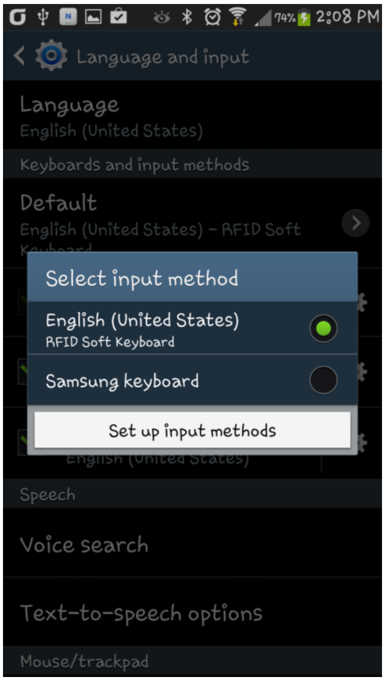
☑ RFID Soft Keyboard
English (United States)

Speech

Voice search

Text-to-speech options

Mouse/trackpad



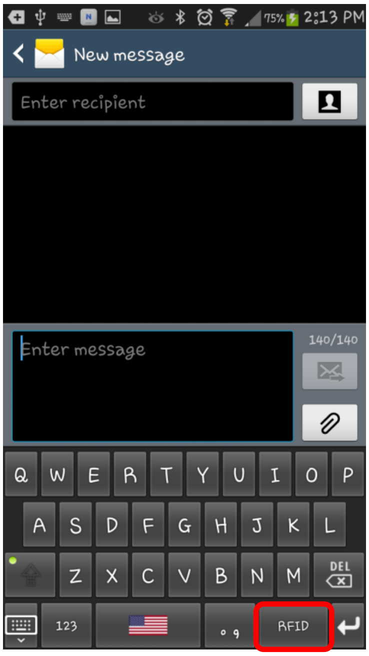
Select input method

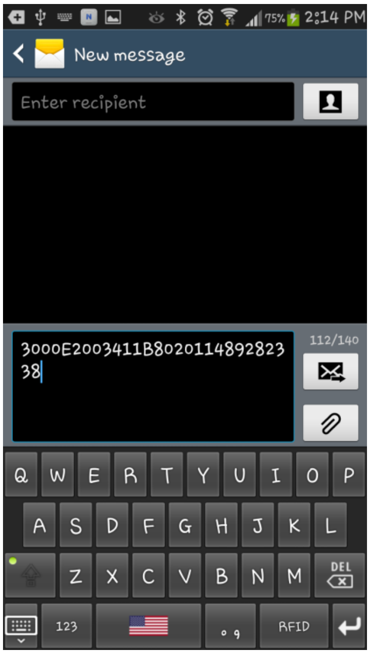
● English (United States)
RFID Soft Keyboard

○ Samsung keyboard

Set up input methods

Insert ARETE POP into your smart phone and open any app where you want to input RFID data. If you touch edit box, your RFID soft keyboard will appears. Touch RFID key to read a tag.





3000E2003411B8020114892823
38

ARETE mobile Customer Service

Phone: + 82 42 864 2402

Fax: + 82 42 864 2403

Email: sales@phychips.com

Address

PHYCHIPS Inc.

#104 Migun Technoworld 2, 187 Techno 2-ro, Yuseong-gu, Daejeon, Korea 305-500

Working Day and time

Monday to Friday

09:00~18:00(Korean Time, GMT Time + 9 hours)

Local Customer Service

Please contact where you buy.