



Tutorial on Basic Android Setup

EE368/CS232 Digital Image Processing, Winter 2014



Linux Version

Introduction

In this tutorial, we will learn how to set up the Android software development environment and how to implement image processing operations on an Android mobile device. Android is an open-source platform developed by Google and the Open Handset Alliance on which interesting and powerful new applications can be quickly developed and distributed to many mobile devices. There is a large, growing community of Android developers and a vast selection of Android devices, which includes smartphones, tablets, and TV setup boxes. Android also comes with an extension library of useful functions, including functions for user interfaces, image/bitmap manipulation, and camera control that we will frequently use in EE368/CS232. We look forward to seeing your novel image processing algorithms and applications running on Android devices as the quarter progresses.

The tutorial is split into two parts. In the first part, we will explain how to download and install the Android software tools onto your computer. Then, in the second part, we will explain how to develop image processing programs that can run on an Android mobile device.

Estimated time to complete this tutorial: 2 hours

Part I: Creating the Software Development Environment ¹

We will use the Google Android SDK, the Java JDK, and the Eclipse IDE to design, implement, and debug Android-compatible programs in this class. Make sure your Linux environment can display graphics (e.g., “ssh -X” if you are connecting to a server).

Downloading and Installing Java JDK

The Java JDK from SUN/Oracle is required for development.

1. Download the latest version of the JDK from this website:
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>
2. Execute the downloaded installer.

Downloading and Installing the Android SDK Bundle

1. If you have a 64-bit OS, you must install the following library:

```
sudo apt-get install ia32-libs
```

¹ Parts of this tutorial borrow explanations from the official Android developers' website (developer.android.com).

2. Download “ADT Bundle for Linux” from this website:
<http://developer.android.com/sdk/index.html>
3. Unzip the downloaded file to a convenient location on your machine, for example:
`/home/yourname/Android/adt-bundle-linux-x86_64-20130729`
4. Open the Eclipse program included in the bundle, for example:

```
cd /home/yourname/Android/adt-bundle-linux-x86_64-20130729/eclipse
./eclipse &
```
5. When asked to choose a default workspace, pick a folder that is easy to remember and access, for example:
`/home/yourname/Android/workspace`
6. Verify that Eclipse starts properly and an IDE window like in Figure 1 is shown. You can display the primary development console by choosing Window > Show View > Console, so that a console like in Figure 2 appears.

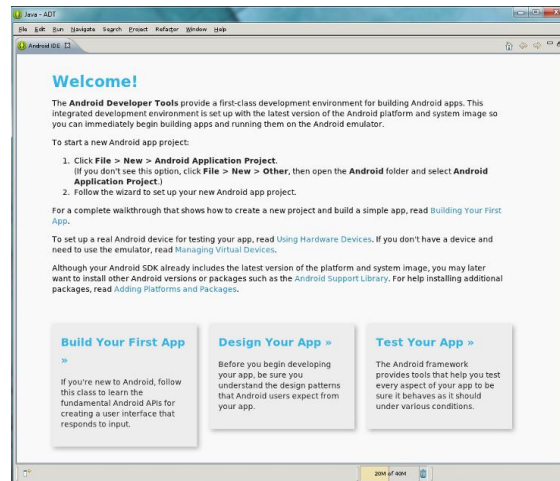


Figure 1. Initial start-up screen of the Eclipse IDE.

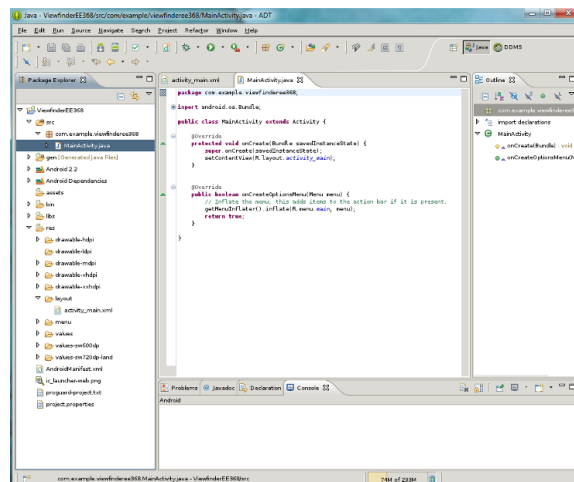


Figure 2. Development console within the Eclipse IDE.

Updating the Android SDK

1. Add the location where you installed the SDK to your system PATH by adding a line like the following to your Linux shell's start-up file (e.g., .bashrc, .cshrc, etc. in your home directory):

```
export ANDR_SDK=/home/username/Android/adt-bundle-linux-x86_64-20130729/sdk
export PATH=${ANDR_SDK}/tools:${ANDR_SDK}/platform-tools:${PATH}
```

OR

```
setenv ANDR_SDK /home/username/Android/adt-bundle-linux-x86_64-20130729/sdk
setenv PATH $ANDR_SDK/tools:$ANDR_SDK/platform-tools:$PATH
```

Be sure to “source .bashrc”, “source .cshrc”, etc.

2. In Eclipse, select Window > Android SDK Manager. A window like that in Figure 3 should pop up.
3. In the Android SDK Manager that pops up, check at least the following boxes under “Packages”:
Tools, Android 4.0, Android 3.0, Android 2.3, Android 2.2, Android 2.1, Android 1.6, Extras.
4. Click “Install <number> packages”, choose “Accept License” for all items listed, and click “Install”. The selected packages will now be downloaded and copied to your Android SDK installation folder. You can monitor the download/installation progress at the bottom of the Android SDK Manager window.
5. During the download, if you are asked for Motorola or HTC developer account information, you can register for free accounts at:
<http://developer.motorola.com>
<http://htcdev.com>

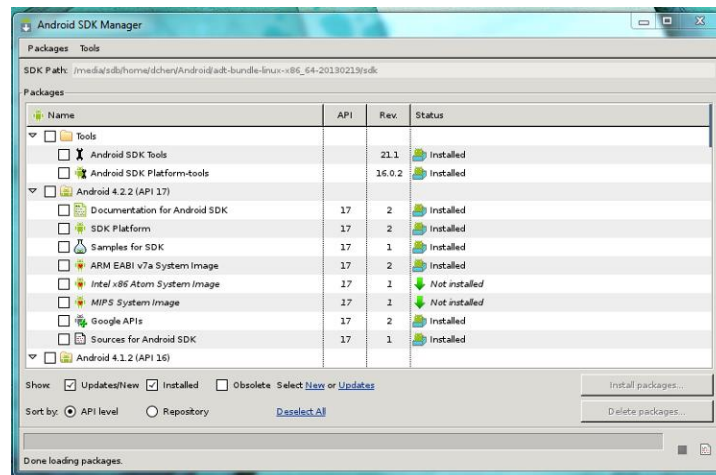


Figure 3. Android SDK manager panel in Eclipse.

6. If you encountered problems in this section, please take a look at the tips on these sites:
<http://developer.android.com/sdk/index.html>
<http://developer.android.com/sdk/installing/bundle.html>

Linking Your Phone to Your Computer

1. Connect your phone to your computer via USB. Turn on your phone.
 2. Go to the home screen.
 3. Select Settings > Applications > Development and then enable USB debugging.
 4. After you have downloaded updates for the Android SDK in Eclipse above, the USB driver should have been included. Install the USB driver on your computer, following the tips on the following page:
<http://developer.android.com/guide/developing/device.html#setting-up>
-

Part II: Developing Image Processing Programs for Android

Now that the Google Android SDK and the Eclipse IDE are all set up on your computer, we are ready to start writing image processing programs that can run an Android-compatible mobile device.

Hello World Example

First, we will build a simple Android program in Eclipse. This simple example will also help you to become familiar with how to create an Android project, how to (auto) compile source code, and how to run the generated executable on the mobile device. Please follow the instructions on this page to develop the “My First App” program:

<http://developer.android.com/training/basics/firstapp/index.html>

Note: In case there are strange errors when the project is created about “R.java”, simply add a comment line (e.g., “// dummy comment”) at the top of “gen : com.example.myfirstapp : R.java” and save the file. This will make the errors disappear.

In the external “My First App” tutorial, they only run the “My First App” program in an emulator. Additionally, we will now also run the program on the actual Android phone. Make sure your phone is properly linked to your computer.

1. In Eclipse, select the project MyFirstApp in the Project Explorer. Then, select Project > Properties > Android and pick the highest Android version supported by your mobile device, e.g., Android 2.1 for the Droid phone.
2. Select Run > Run Configurations > Android Application > MyFirstApp > Target. Choose “Always prompt to pick device”.
3. Select Run, and in the Device Chooser dialog, select your Android phone. The “My First App” program will be sent to and automatically started on your phone, and you should see the screen similar to Figure 4 on your phone.



Figure 4. “My First App” program running on an Android phone.

EE368 Viewfinder Example

Now, having grasped the fundamentals of building and running an Android application, we will create a more complicated project involving the onboard camera and real-time image processing.

1. Create a new Android project with the following parameters.
 Application name: Viewfinder EE368
 Project name: ViewfinderEE368
 Package name: com.example.viewfinderee368
 Minimum Required SDK: API 7: Android 2.1 (Éclair)
 Target SDK: API 7: Android 2.1 (Éclair)
 Compile With: API 7: Android 2.1 (Éclair)
 Theme: None
 Create Activity: Blank Activity
2. Copy the text in the following document into `AndroidManifest.xml`. This defines the main activities and permissions for this program.
<http://ee368.stanford.edu/Android/ViewfinderEE368/AndroidManifest.xml.txt>
3. Copy the text in the following document into `src : com.example.viewfinderee368 : ViewfinderEE368.java`. This defines the classes in this program.
<http://ee368.stanford.edu/Android/ViewfinderEE368/ViewfinderEE368.java>
4. Check to make sure everything is copied correctly into the project. If there are compilation errors, a red X will appear in the Package Explorer.
5. Select Run and in the Device Chooser dialog, select your phone. You should see something like Figure 5 on your phone. Point the camera at different objects around you

to see how the mean, standard deviation, and histogram of each color channel changes dynamically. You are augmenting the viewfinder in real time!



Figure 5. “Viewfinder EE368” program running on an Android phone.

Real-time Phone Debugging in Eclipse

It is actually possible to view real-time messages from the phone in Eclipse, which can be very helpful for debugging and code development.

1. Select Window > Open Perspective > DDMS.
2. A new tab entitled “DDMS” should appear next to the default “Java” tab. Click on the “DDMS” tab.
3. Select your Android device from the list of Devices.
4. Select Window> Show View > LogCat. The LogCat view shows a sequential list of real-time messages from the phone. In particular, error messages in red can be very useful when trying to debug a problem.

Taking a Screenshot of the Phone

At some point, it may be useful to take a screenshot of the phone, e.g., to use as a figure in your project report.

1. Go to the DDMS perspective in Eclipse.
2. Select your Android device, as shown in Figure 6.

- Click the camera icon (circled in red in Figure 6), and a panel like Figure 7 should pop up.
- Finally, when you have the desired screen shown, click Save or Copy to extract the screen shot.

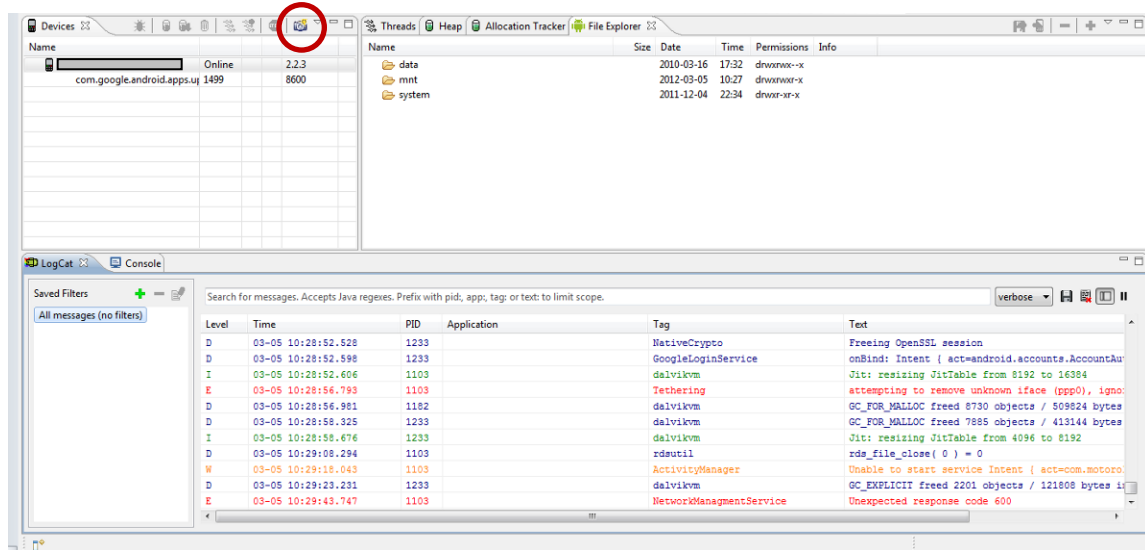


Figure 6. Dalvik Debug Monitor panel.

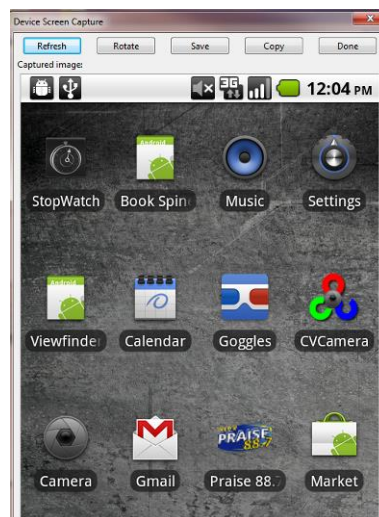


Figure 7. Device screen capture panel.