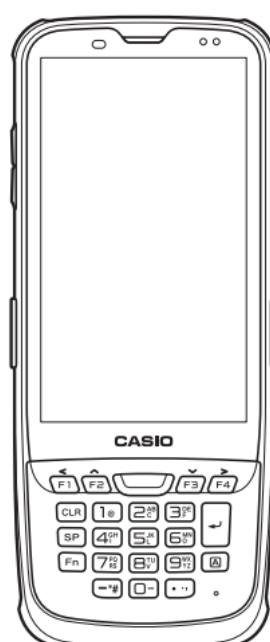


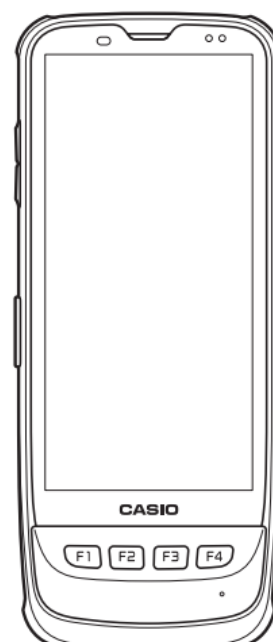
IT-G600/650 Series

Quick Start Guide

This document is a guide book for
IT-G600/650 application developers.



IT-G600



IT-G650

Cautions

- No part of this document may be produced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of CASIO Computer Co., Ltd. in Tokyo Japan.
- Information in this document is subject to change without advance notice.
- CASIO Computer Co., Ltd. makes no representations or warranties with respect to the contents or use of this manual and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose.

© 2020 CASIO COMPUTER CO.,LTD.

Microsoft, Windows are trademarks of Microsoft Corporation. Java is a trademark of Oracle Corporation.

Mac, Mac OS and OS X are trademarks of Apple Inc. Linux is a trademark of Linus Torvalds.

Intel is a trademark of Intel Corporation. AMD is a trademark of Advanced Micro Devices, Inc.

Android is a trademark of Google LLC.

Other company, product and service names used in this manual also may be trademarks or registered trademarks of others.

- Table of contents -

| | |
|--------------------------------------|-----------|
| 1. Overview | 2 |
| 2. Basic Development Kit | 3 |
| 2.1 Structure | 3 |
| 2.2 Version | 5 |
| 2.3 Notes | 5 |
| 3. Development environment | 6 |
| 3.1 Recommended environment | 6 |
| 3.2 Required system | 7 |
| 3.3 Construction steps | 8 |
| 3.3.1 Android Studio | 8 |
| 3.3.2 CASIO extended function | 21 |

1. Overview

The IT-G600/650 is the rugged smart handheld terminal equipped with Android. CASIO extends function, such as a hardware key and a barcode scanner, for Android smart devices equipped with a large-screen touch panel LCD. The IT-G600/650 was developed as a flagship model of a rugged smart handheld terminal that combines the functions of an Android smart device and a handheld terminal.

This manual is a guidebook for developers that outlines the "IT-G600/650 Basic Development Kit" and describes how to develop applications that control CASIO extended functions such as keys and barcode scanners.

2. Basic Development Kit

2.1 Structure

The following figure shows the structure of this Basic Development Kit.

Manuals

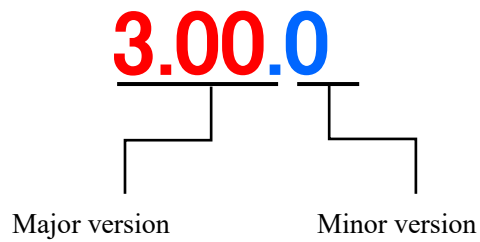
| Folders / Files | Descriptions |
|------------------------------|--|
| MANUAL/EN | Manual |
| QuickStartGuide.pdf | A guidebook for application developers. Describes how to build a development environment. |
| SoftwareManual.pdf | This is a manual for the software to be installed. Describes the software and its specifications. |
| HardwareManual.pdf | This is a manual for the hardware to be installed. This section describes notes on handling and operation. |
| DeviceControl.pdf | This is a manual describes how to control barcode scanners and keys. Describes how to create an application that controls devices. |
| javadoc | Reference manual for the device control library. This section describes the function specifications of the device control library. |
| KittingManual.pdf | This is a manual for the kitting. Describes the kitting procedure to make the terminal ready for business use. |
| CasioAndroidAddonsManual.pdf | This is a manual for the add-ons. Describes how to extend the terminal for business use. |
| SecurityToolManual.pdf | This is a manual for security tools. Describes how to use security tools to enhance security for business use, such as strengthening logins and setting restrictions. |

Software

| Folders / Files | Descriptions |
|----------------------------------|--|
| /Software | Software |
| /DeviceControl | Device Control folder |
| DeviceLibrary.aar | A library for controlling devices such as barcode scanner and keys etc. Used to create applications that control devices. |
| IntentManager.apk | An application for controlling devices such as barcode scanner etc. by intent. Used to create simple applications that control devices. |
| /Tools | Tools folder |
| /SecurityTools | Security tools folder |
| SecurityToolsxxxxxxxxx.apk | Applications to strengthen security in addition to Android standard security. Consists of Login Security, Home Security, and Blocking Security. |
| AuthenticateServicexxxxxxxxx.apk | |
| SecurityServicexxxxxxxxx.apk | |
| JobUserEditorxxxxxxxxx.apk | |
| LoginSettingxxxxxxxxx.apk | |
| JobMenuxxxxxxxxx.apk | |
| SettingsBlockerxxxxxxxxx.apk | |
| KitData.xls | A definition file for the KitDevice. |
| /Samples | Samples folder |
| KeyLibrarySample.zip | Sample program for KeyLibrary. |
| SymbolScan.zip | Sample programs for ScannerLibrary. |
| RangeScan.zip | |
| InverseScan.zip | |
| CenteringWindowScan.zip | |
| TriggerScan.zip | |
| ImageCapture.zip | Sample program for ScannerIntent. |
| ImageCaptureEx.zip | |
| SymbolScan_Intent.zip | |
| OSUpdateSample.zip | |
| | Sample program for OSUpdate. |

2.2 Version

The version number of this development kit consists of a combination of major version and minor version, as shown below.



The major number represents the version of DeviceLibrary (i.e. DeviceLibrary.aar). So, when a newer version of development kit is released, if it includes newer version of DeviceLibrary, its major version number is increased, and the minor version is returned to zero. If changes other than DeviceLibrary (i.e. manuals, tools, etc.) are included, its minor version number is increased.

2.3 Notes

Notes for using this development kit are as follows.

There is a dependency between the version of Android Studio and the version of the Basic Development Kit. For details, refer to "3.1 Recommended environment (p.6)".

3. Development environment

3.1 Recommended environment

To develop the application for the IT-G600/650, it needs the development environment of Android.

| Development language | Development platform (Recommended) |
|----------------------|---|
| Java | Android Studio 3.0 or above Android SDK (API level 28 above) Google USB driver JDK7 or above (Bundled with Android Studio) |

Note!

The basic development kit was confirmed to work with the following Android Studio combination. Use the following combination or above version.

However, there is a possibility that something wrong with work if use the latest unconfirmed Android Studio version.

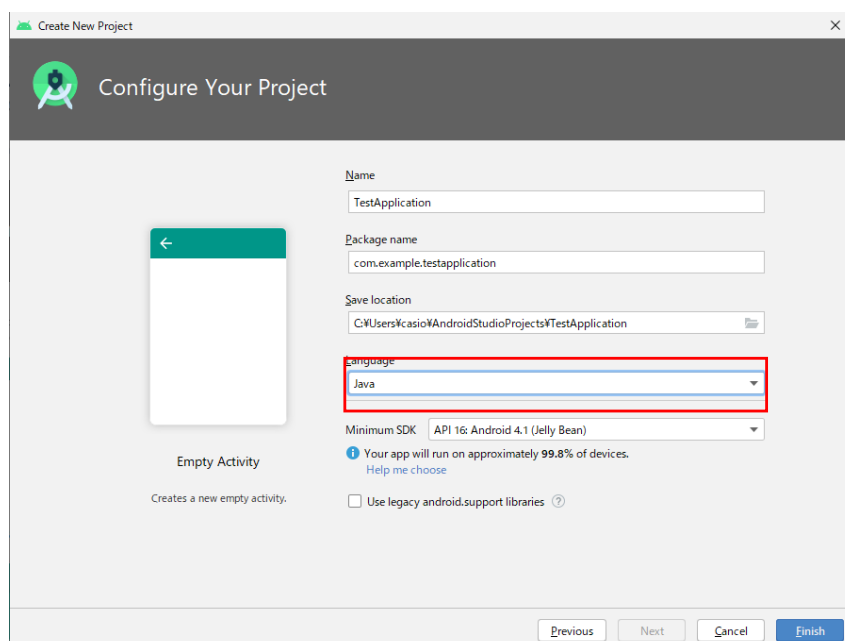
If there is something wrong with work, try with the confirmed below combination.

All released versions of Android Studio are available at the following web site.

<https://developer.android.com/studio/archive>

| Software | Version |
|---|---------|
| Android Studio | 4.0 |
| Gradle version (Android Studio) | 6.1.1 |
| Android Plugin Version (Android Studio) | 4.0.0 |
| Development Kit | 3.013.1 |

The default development language for Android Studio 3.4 or above is Kotlin. Select Java in case of creating a new project.



3.2 Required system

[Windows]

- Microsoft® Windows® 8/10 (64bit)
- 4 GB RAM minimum, 8 GB RAM recommended
- 2 GB of available disk space minimum, 4 GB Recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image)
- 1280 x 800 minimum screen resolution

[Mac]

- Mac® OS X® 10.10 (Yosemite) or higher, up to 10.14 (macOS Mojave)
- 4 GB RAM minimum, 8 GB RAM recommended
- 2 GB of available disk space minimum, 4 GB Recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image)
- 1280 x 800 minimum screen resolution

[Linux]

- GNOME or KDE desktop
- Tested on gLinuxbased on Debian (4.19.67-2rodete2)
- GNU C Library (glibc) 2.19 or later
- 4 GB RAM minimum, 8 GB RAM recommended
- 2 GB of available disk space minimum, 4 GB Recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image)
- 1280 x 800 minimum screen resolution

Note!

The explanation after the next chapter is all for Windows (64bit). If you are using Mac or Linux as a development environment, you are responsible for building the environment at your own risk.

If using windows (32bit), make sure Android Studio is working properly on the PC before installing this BDK. The final version of Android Studio for Windows (32bit) is 3.6.

3.3 Construction steps

The followings are the basic steps to construct the development environment.

Due to version upgrade of Android Studio, environment may not be constructed by the following procedure. In that case, refer to the procedure described on Google's website etc. and construct the environment.

- (1) Download and install Android Studio
- (2) CASIO extended function control

3.3.1 Android Studio

Download and install

Android Studio can be downloaded from the following site of Google.

<https://developer.android.com/studio/>

Agree to the dialog shown below and, then start downloading.

Executing the downloaded file, follow the instructions to proceed the installation.

Download Android Studio

Before downloading, you must agree to the following terms and conditions.

Terms and Conditions

This is the Android Software Development Kit License Agreement

1. Introduction

1.1 The Android Software Development Kit (referred to in the License Agreement as the "SDK" and specifically including the Android system files, packaged APIs, and Google APIs add-ons) is licensed to you subject to the terms of the License Agreement. The License Agreement forms a legally binding contract between you and Google in relation to your use of the SDK.

1.2 "Android" means the Android software stack for devices, as made available under the Android Open Source Project, which is located at the following URL: <https://source.android.com/>, as updated from time to time.

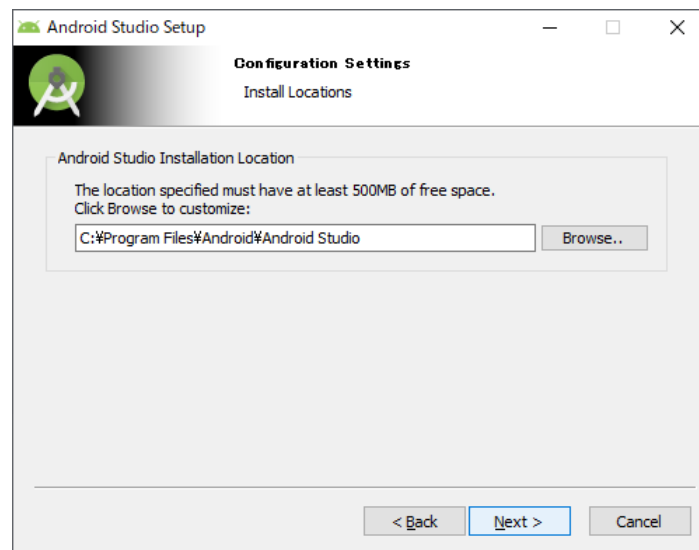
1.3 A "compatible implementation" means any Android device that (i) complies with the Android Compatibility Definition document, which can be found at the Android compatibility website

☒ I have read and agree with the above terms and conditions

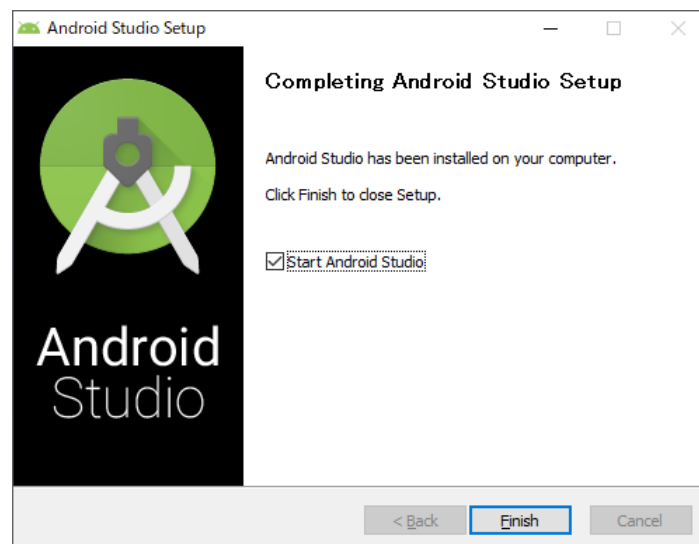
DOWNLOAD ANDROID STUDIO FOR WINDOWS

android-studio-ide-193.6514223-windows.exe

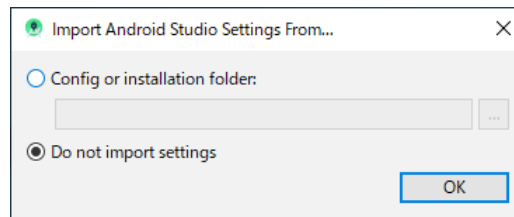
Prompted to specify the installation destination. Change it if necessary.



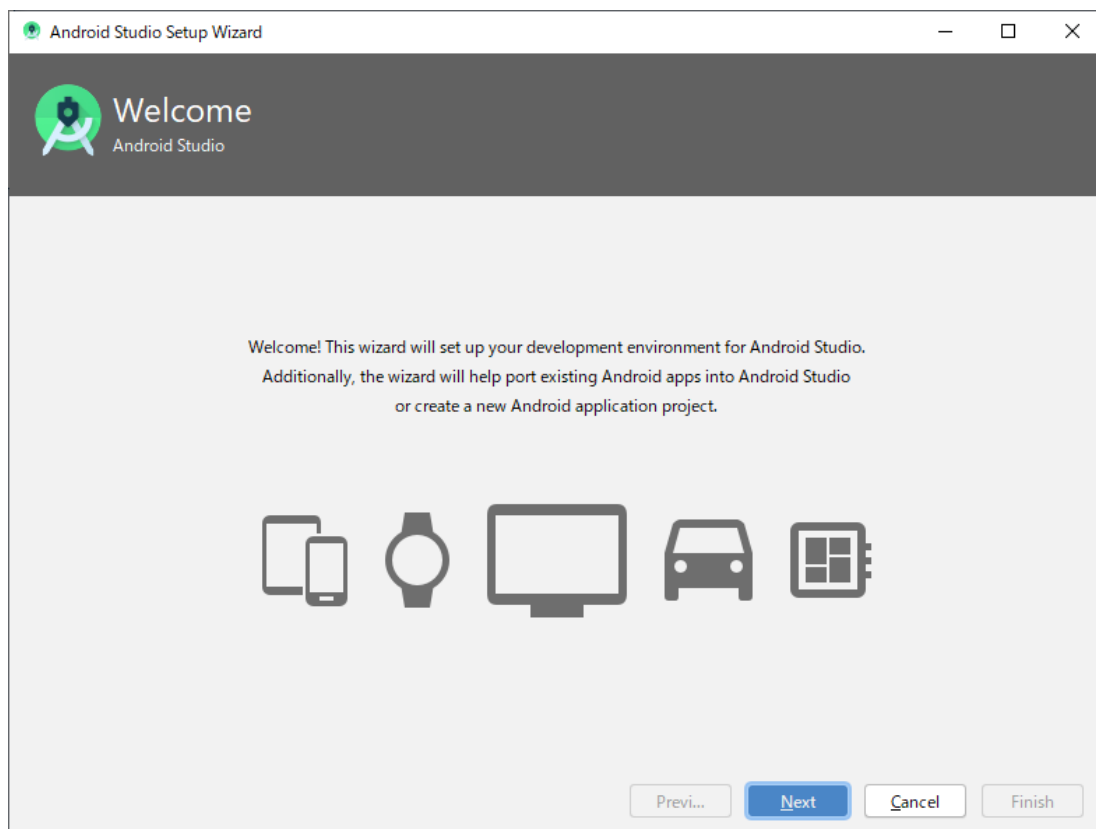
When the below dialog is displayed, installation of Android Studio is completed.



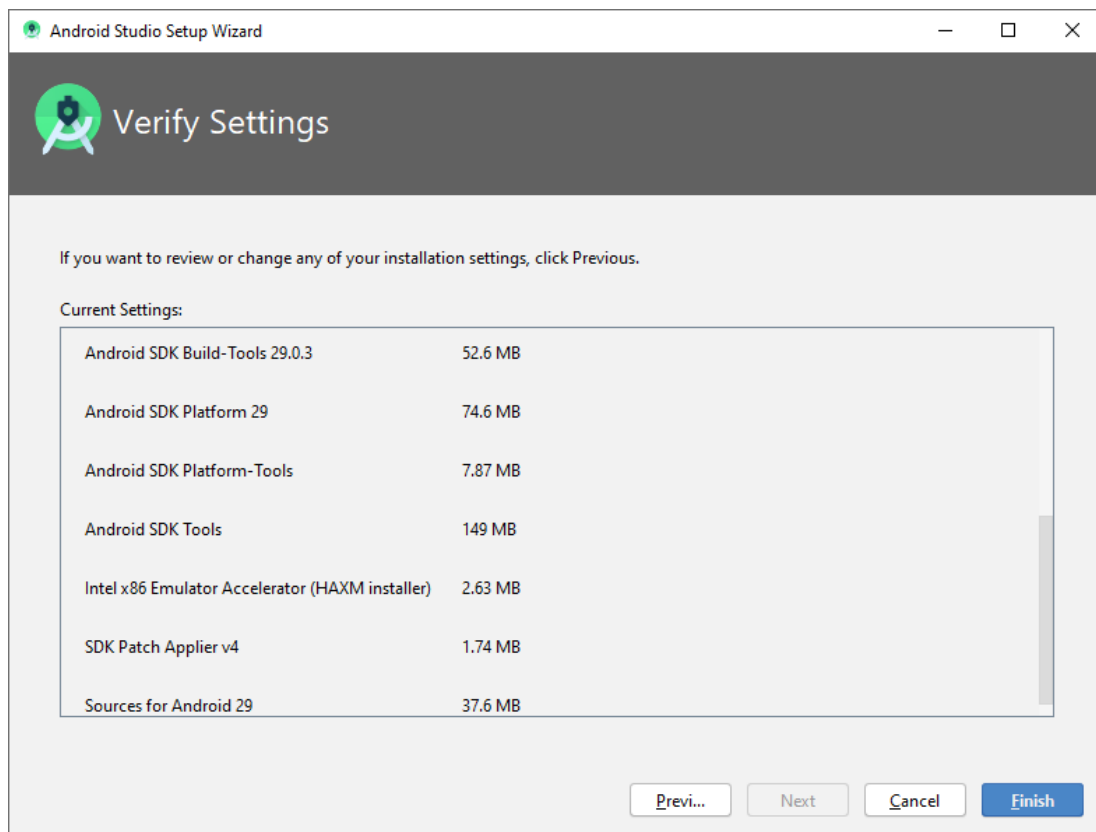
When start the installed Android Studio first time, select whether to use the previous setting or not. If do not want to use the previous setting, select "Do not import settings".



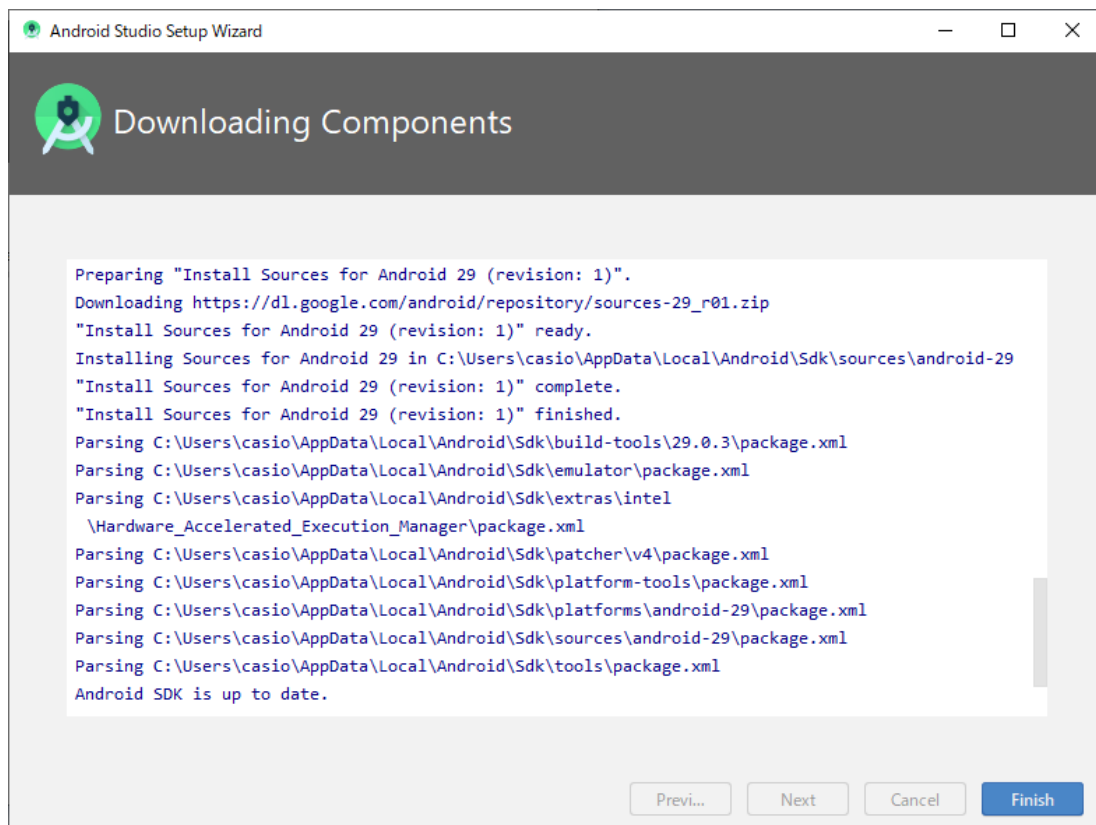
The Setup Wizard will start, so follow the instructions.



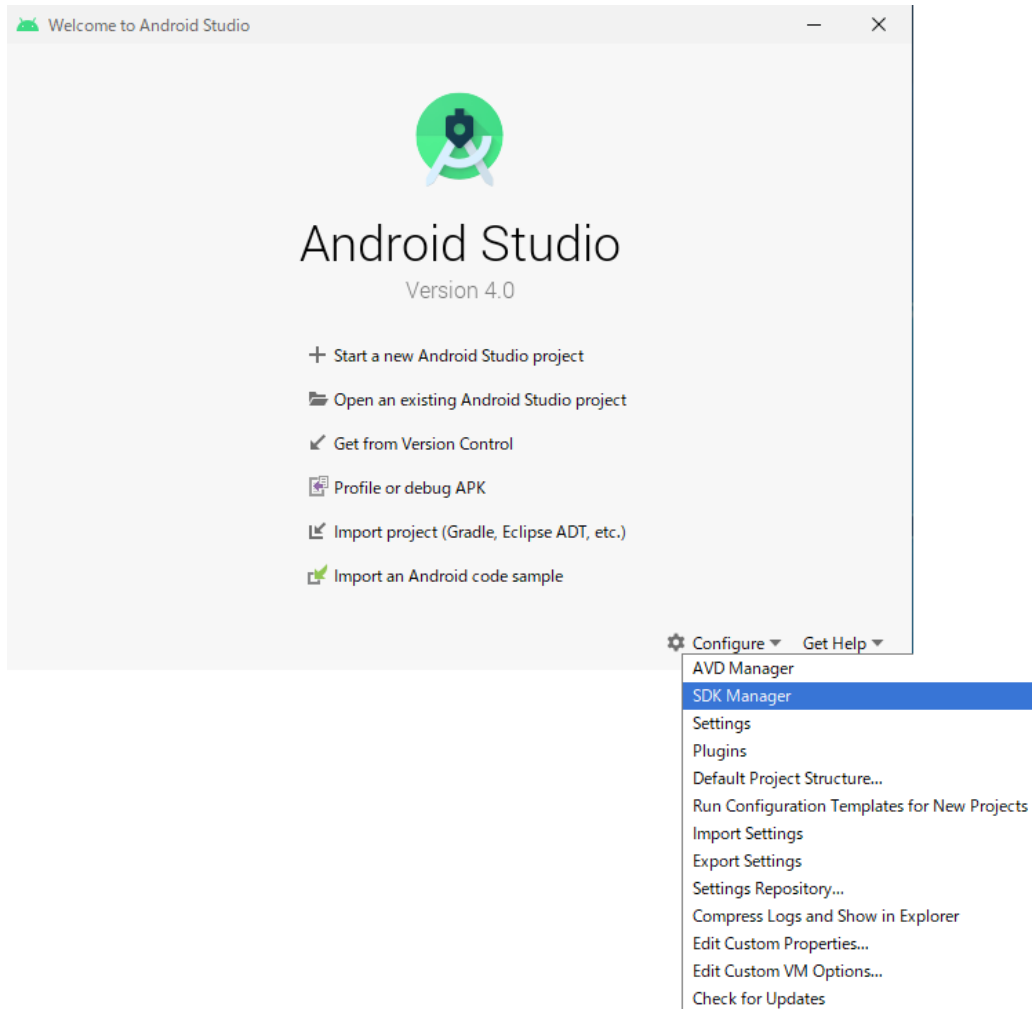
If do not have any problem after confirming "Finish".



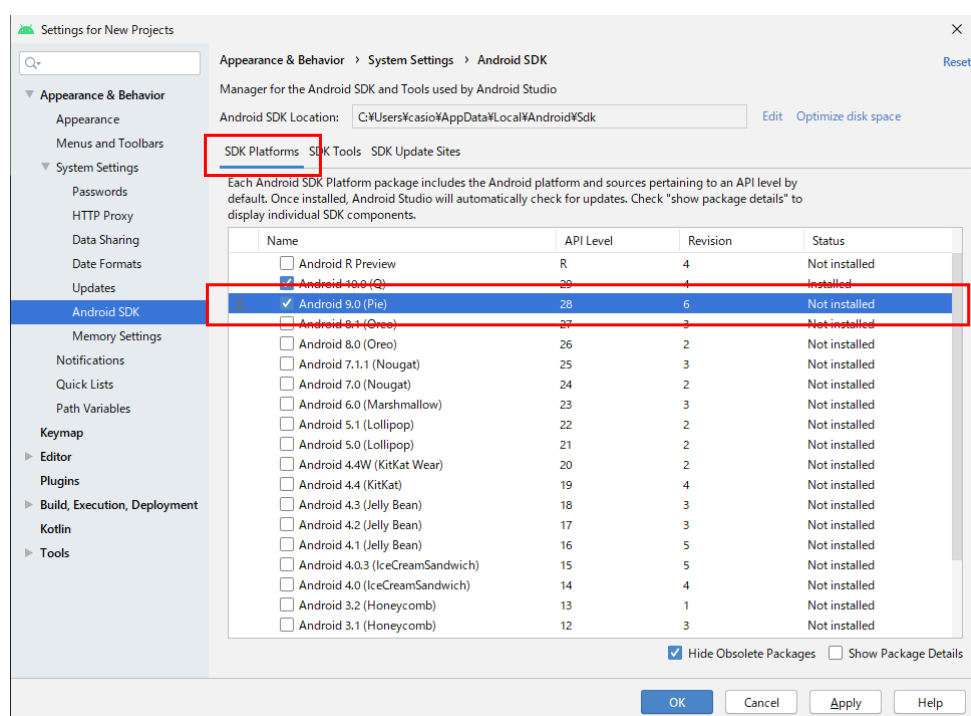
Downloading Components begins. When completed, Android Studio will start up.



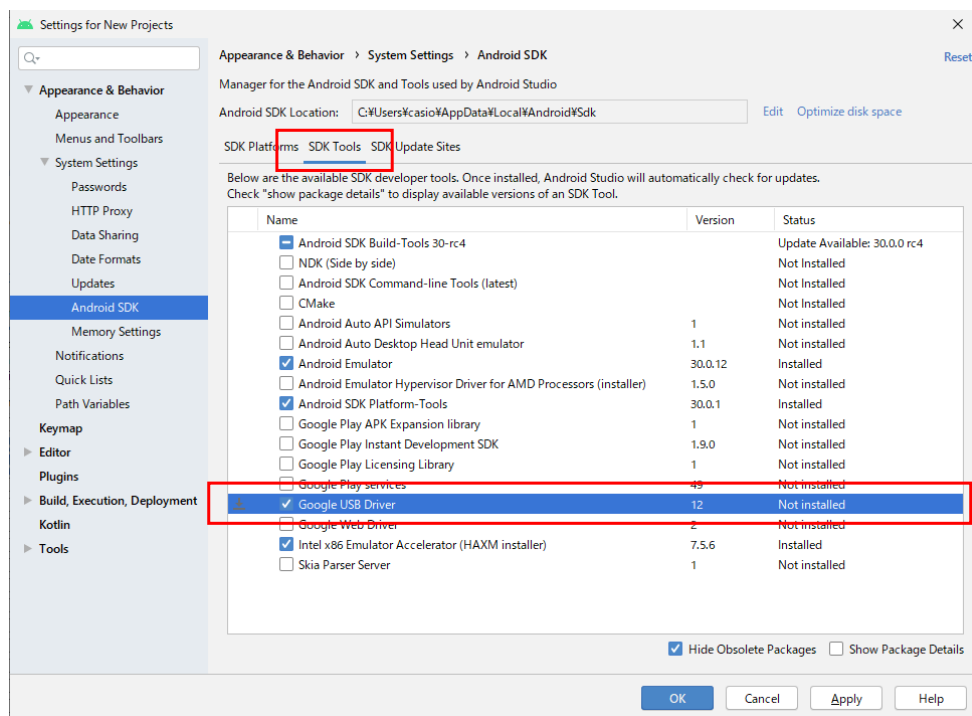
Launch the Android Studio, and chose "SDK Manager" from Configure tab.



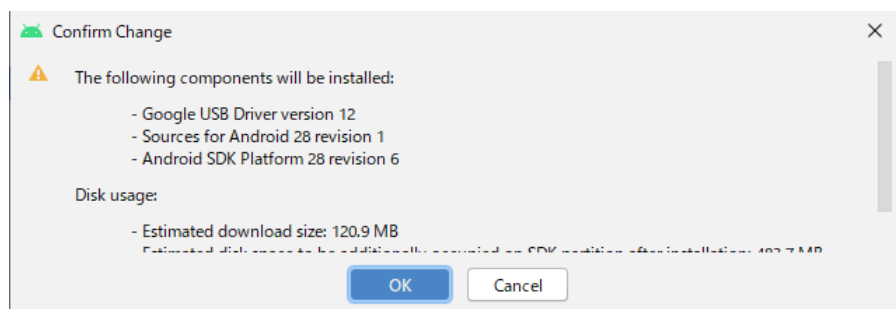
Select "SDK Platform" tab, then check "Android 9.0 (API Level 28)".



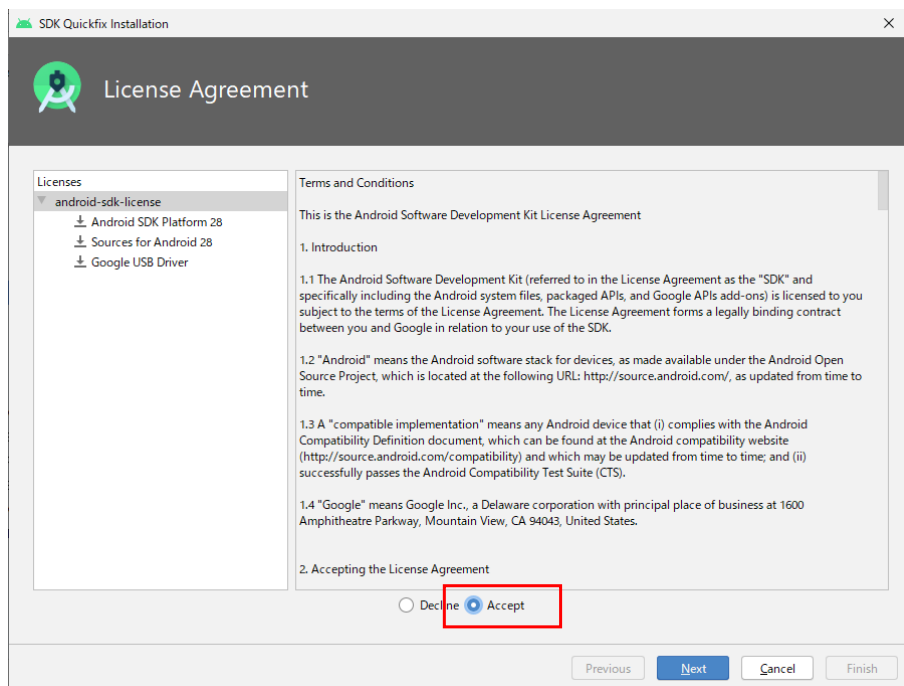
Next, select "SDK Tools" tab and check "Google USB Driver".



The following dialog will be displayed, press "OK".



Accept the "License Agreement", then press "OK" to start installation.



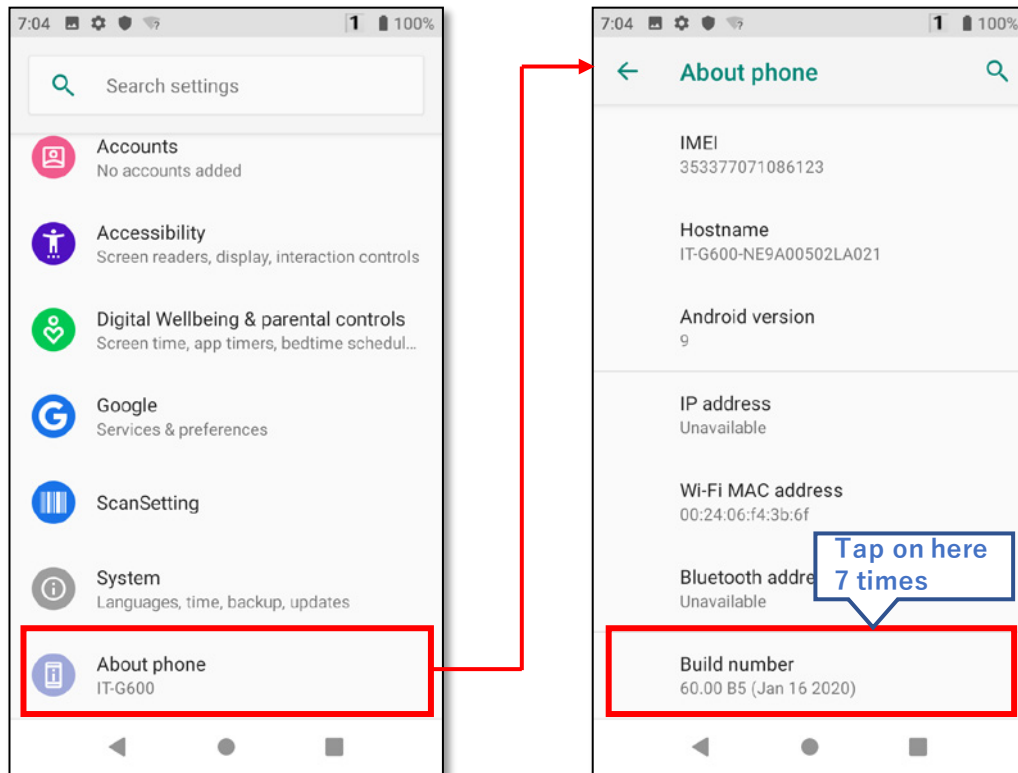
The USB driver is stored in "extras\google\usb_driver" under "Android SDK Installation Location".

e.g.) C:\Users\ xxx\ AppData\ Local\ Android\ Sdk\ extras\ google\ usb_driver
(xxx: username of computer)

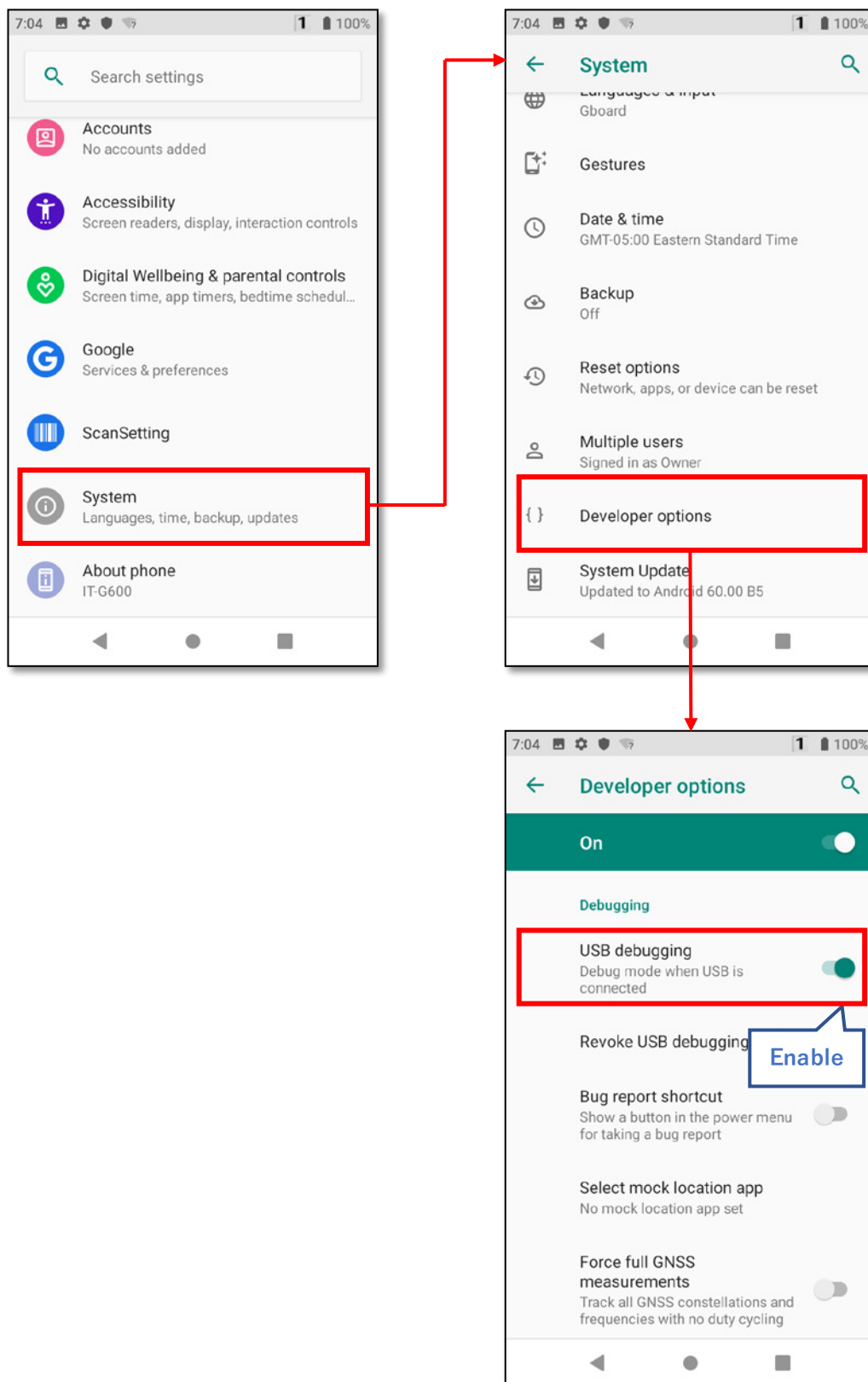
USB debugging

Since USB debugging is a function for developers, need to activate the Developer options by the following procedure first.

Open [Settings] -> [About phone] and tap "Build number" on the bottom line seven times, [Developer options] is added above [About phone].



Open [Settings] -> [About phone] -> [Developer options] and enable "USB debugging".



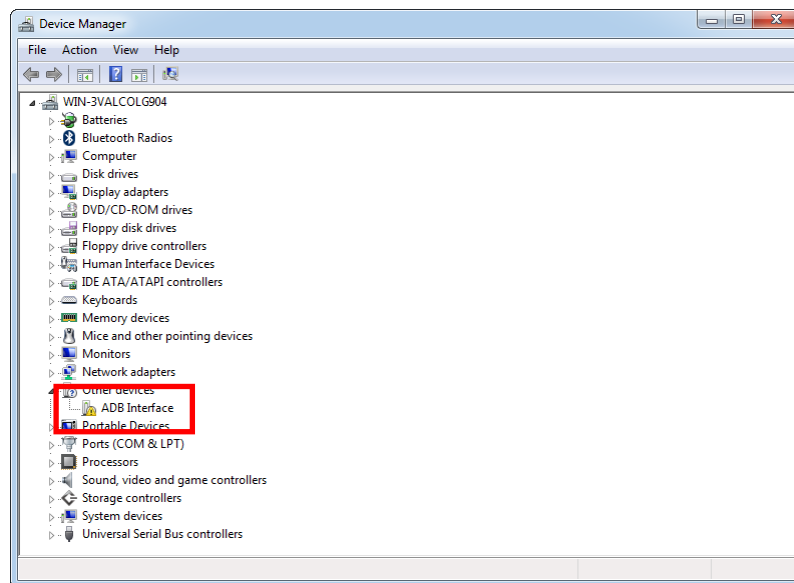
Install the USB driver for connecting the IT-G600/650 and PC with the ADB (Android Debug Bridge) protocol.

Skip from this procedure in case of recognizing the IT-G600/650 as ADB on PC already.

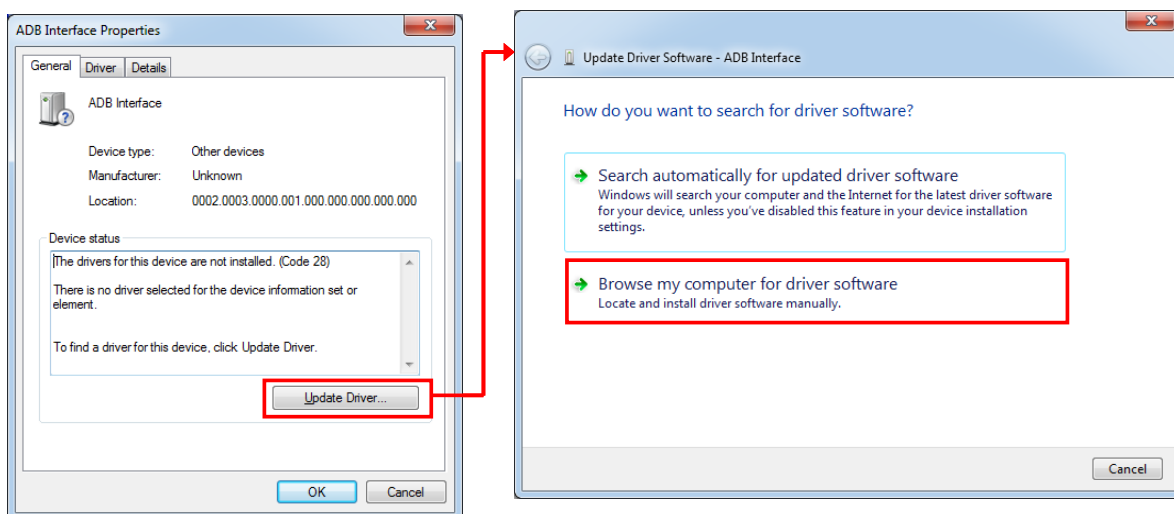
Connect the IT-G600/650 to a PC using a USB cable or a USB cradle.

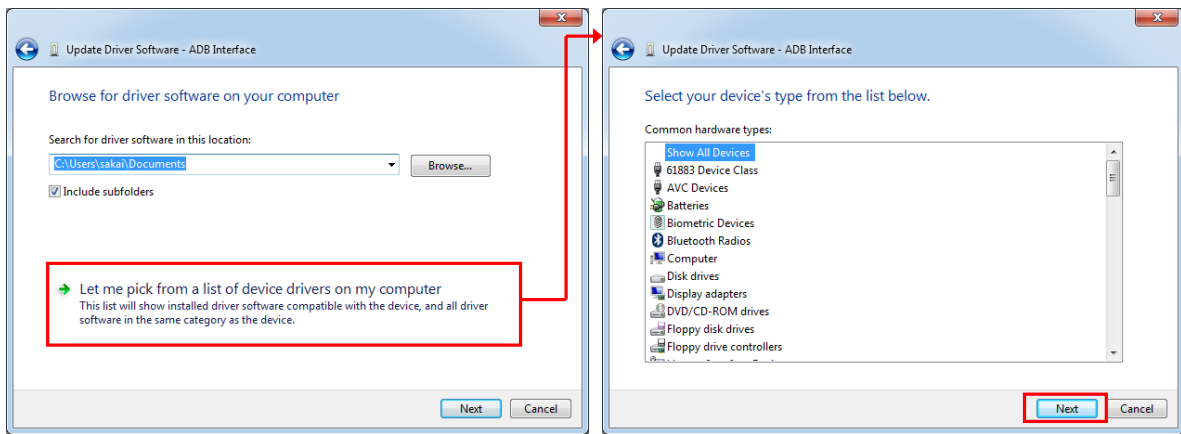
Then, open the Device Manager by [Control Panel] -> [Hardware and Sound] -> [Device Manager].

In the Device Manager, "ADB Interface" is displayed as an unknown device as shown below. By right-click the red frame ("Android") in the above figure, displays its properties, and press [Update Driver].

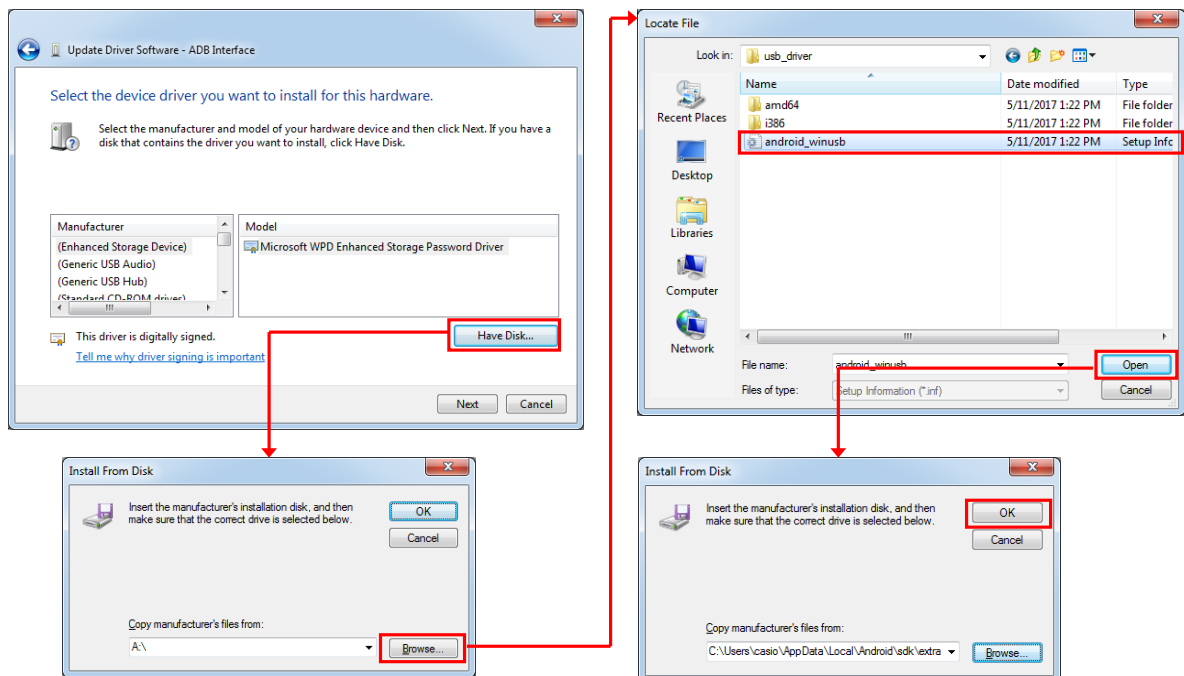


Press "Let me pick from a list of drivers on my computer " at the bottom. In the driver list, with "Show All Devices" is highlighted and press [Next].

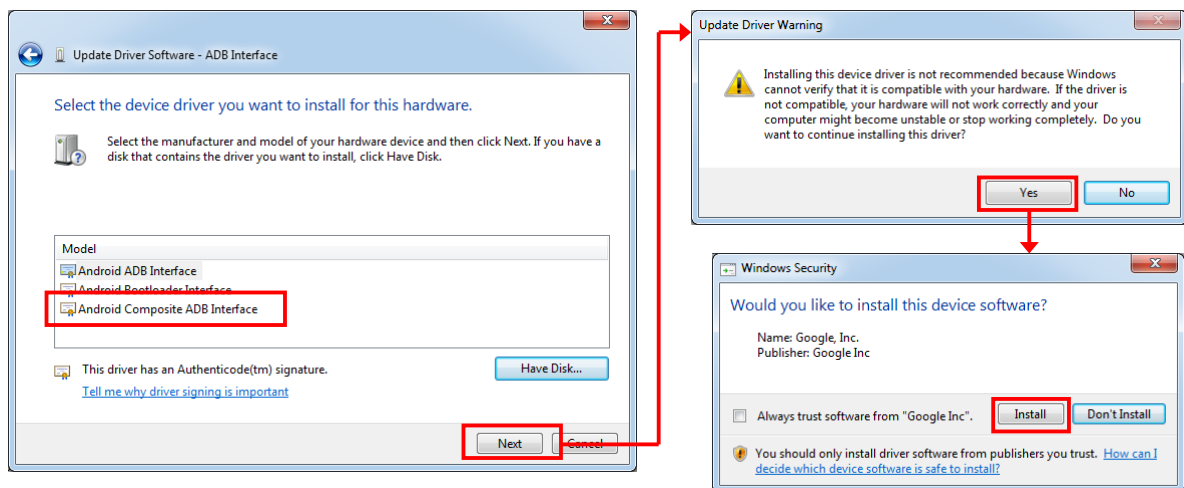




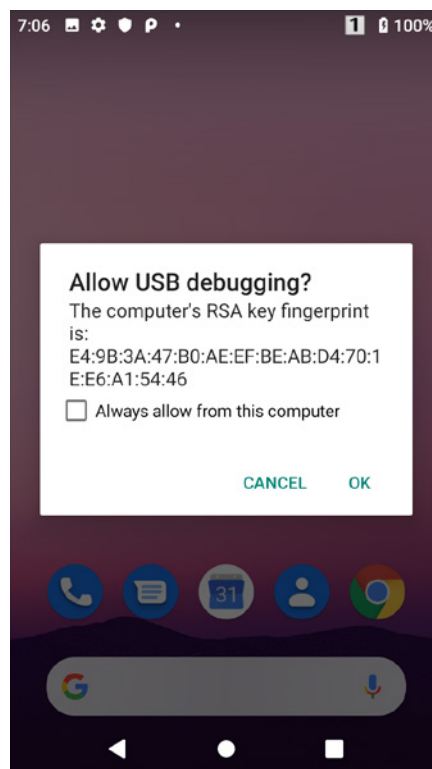
When click [Hard Disk...] in the displayed dialog, a dialog box for specifying the location of the driver opens, so click [Browse...] to open the file dialog box.
In the file dialog box, specify the downloaded USB driver.
(e.g. C:\Users\xxx\AppData\Local\Android\Sdk\extras\google\usb_driver\android_winusb.inf)



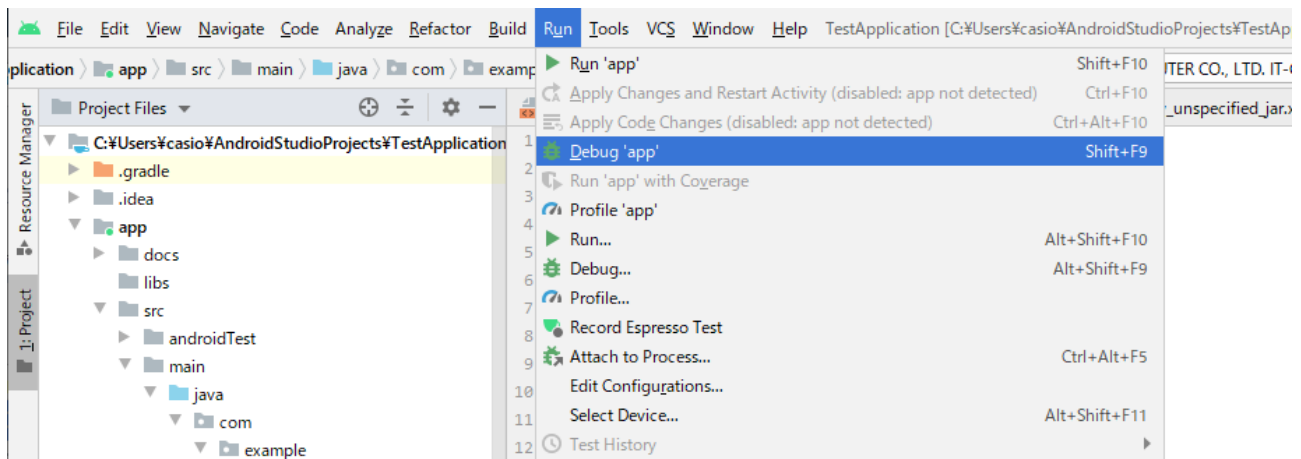
The following driver selection screen will appear. Select "Android Composite ADB Interface" and press [Next].
Answer "Yes" to "Driver Update Warning" and press "Install" of "Windows Security" dialog.



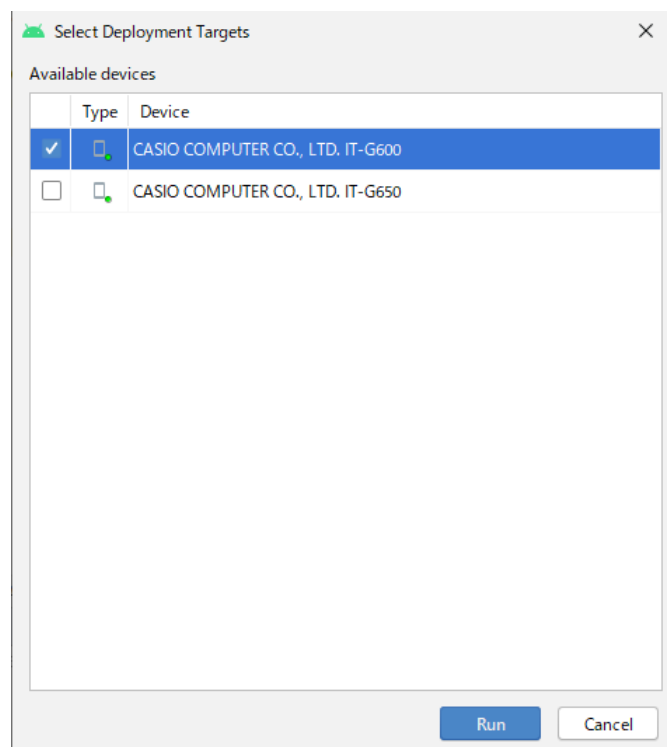
By the above, the installation of the USB driver for ADB protocol was completed. Press "OK" when the follow dialog is shown on the IT-G600/650.



Select "Run" -> "Debug 'app'" on Android Studio.



When the following dialog is shown, select the target device "CASIO COMPUTER CO., LTD. IT-G600" or "CASIO COMPUTER CO., LTD. IT-G650" from "Available devices" and press "Run". Start to debug an application on the IT-G600/650.



3.3.2 CASIO extended function

The following methods are available for developing applications that control keys and barcode scanners.

For details on each function, refer to the "Device Control Manual".

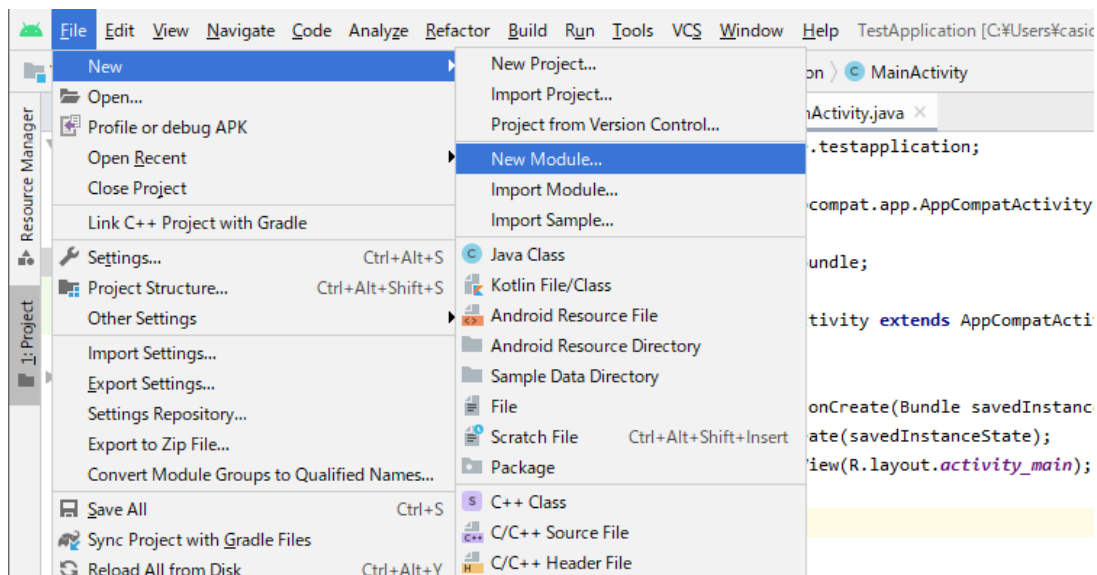
- Application development with imported device control library
- Install an intent manager and develop an application that issues intents

Device Library

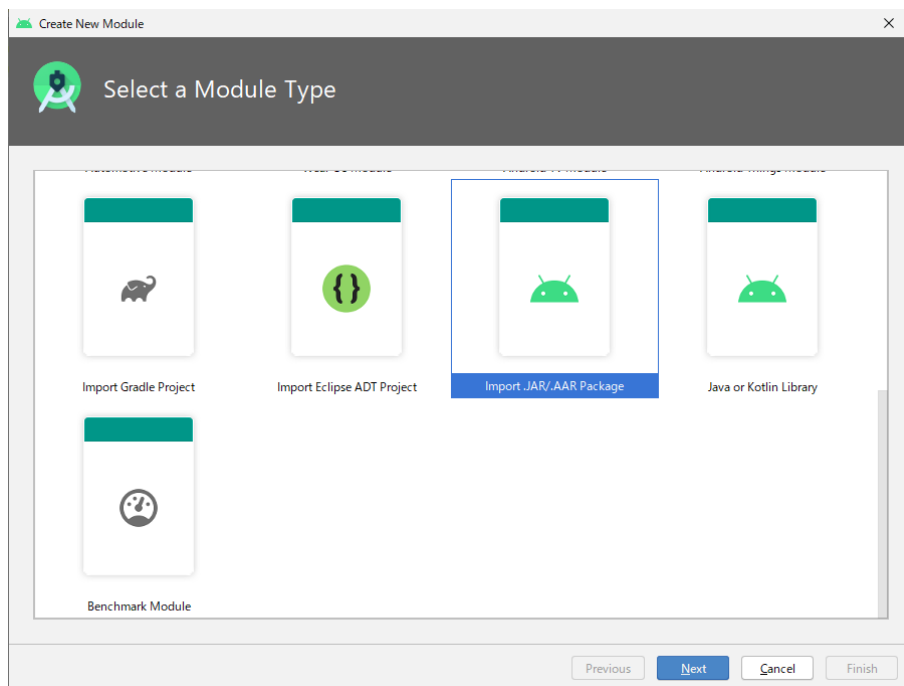
When developing an application that controls CASIO extended functions (barcode scanner, trigger key, etc.), import the DeviceLibrary. The library import is required for each project.


Import

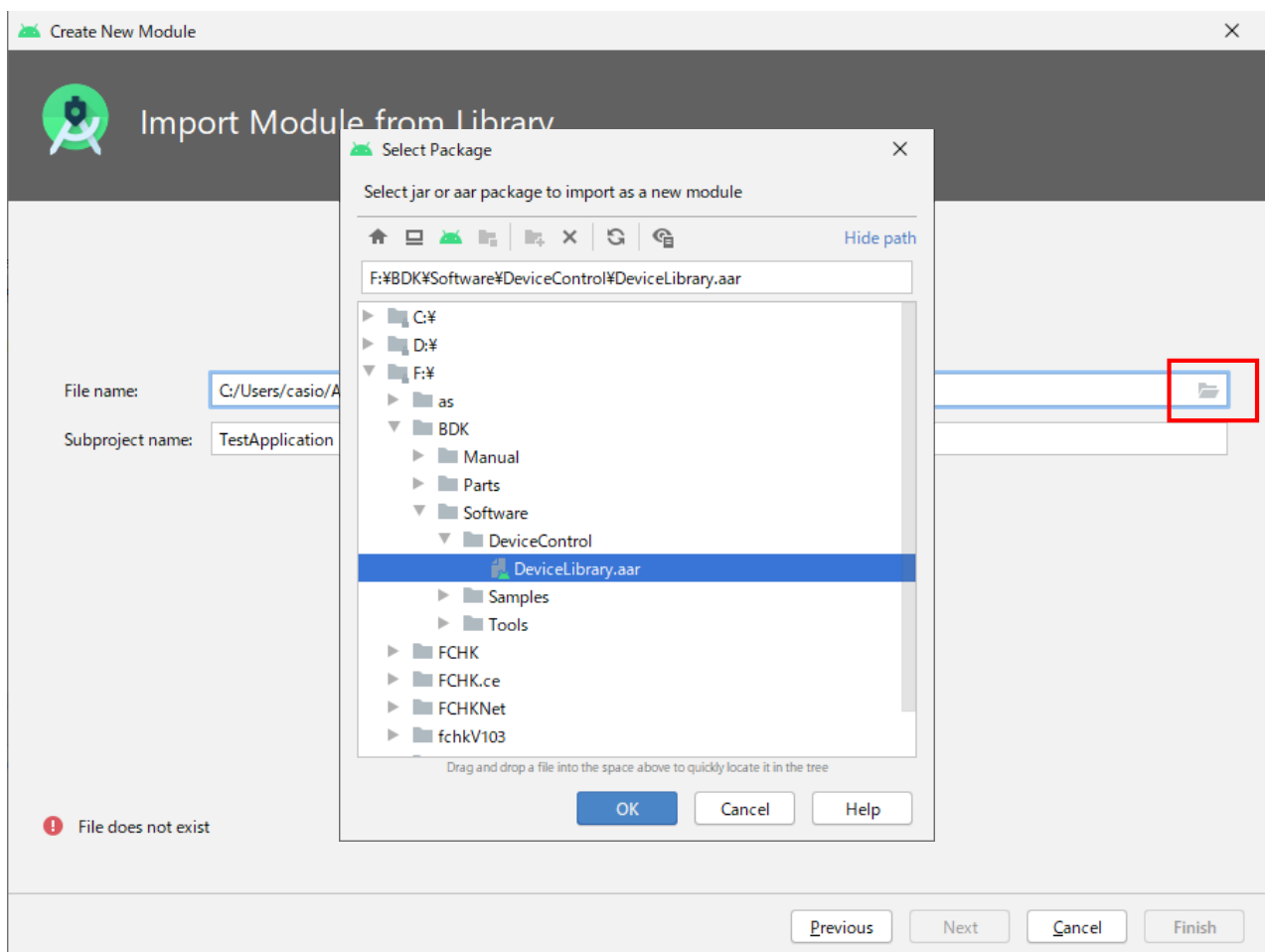
Open the project from the Android Studio, select "New Module" dialog with [File] -> [New] -> [New Module].



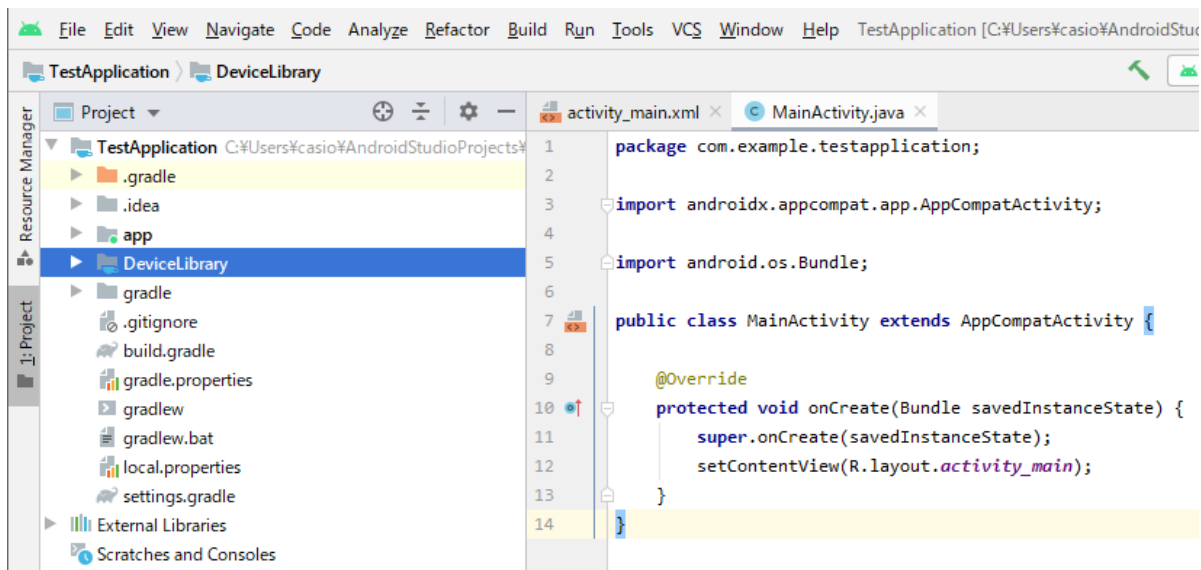
Select "Import .JAR/.AAR Package" and select "Next".



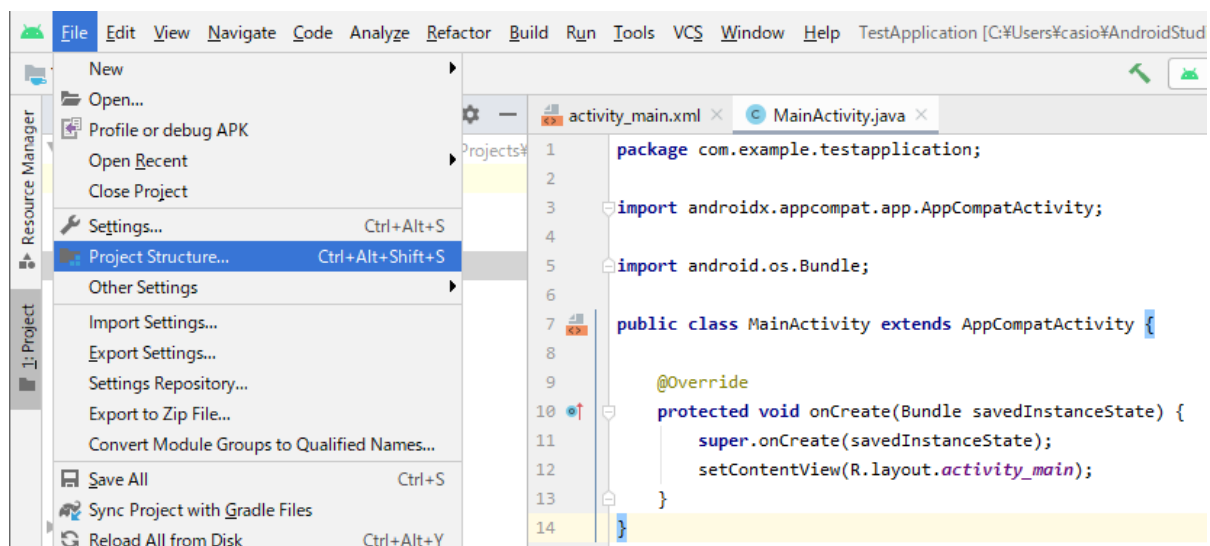
Press the  of "File name" and specify the DeviceLibrary (AAR) to be imported.



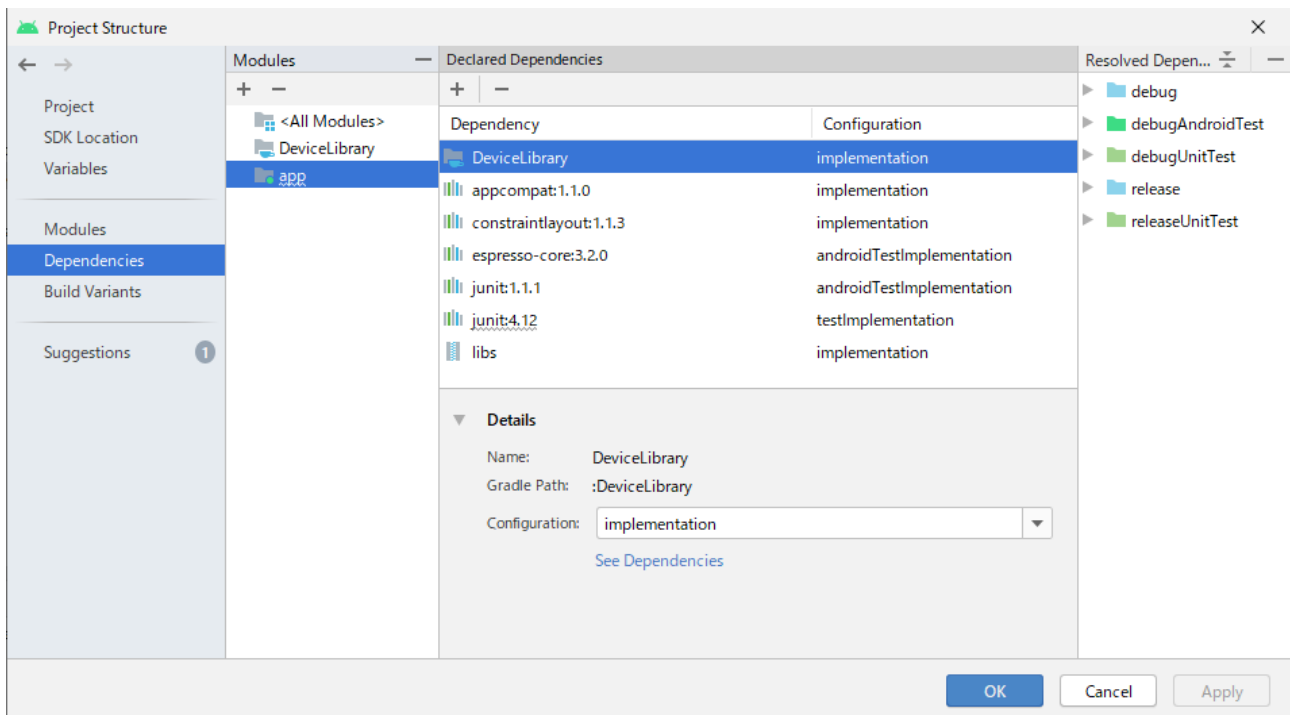
Confirm that the DeviceLibrary is added to the Android Studio Project tree.



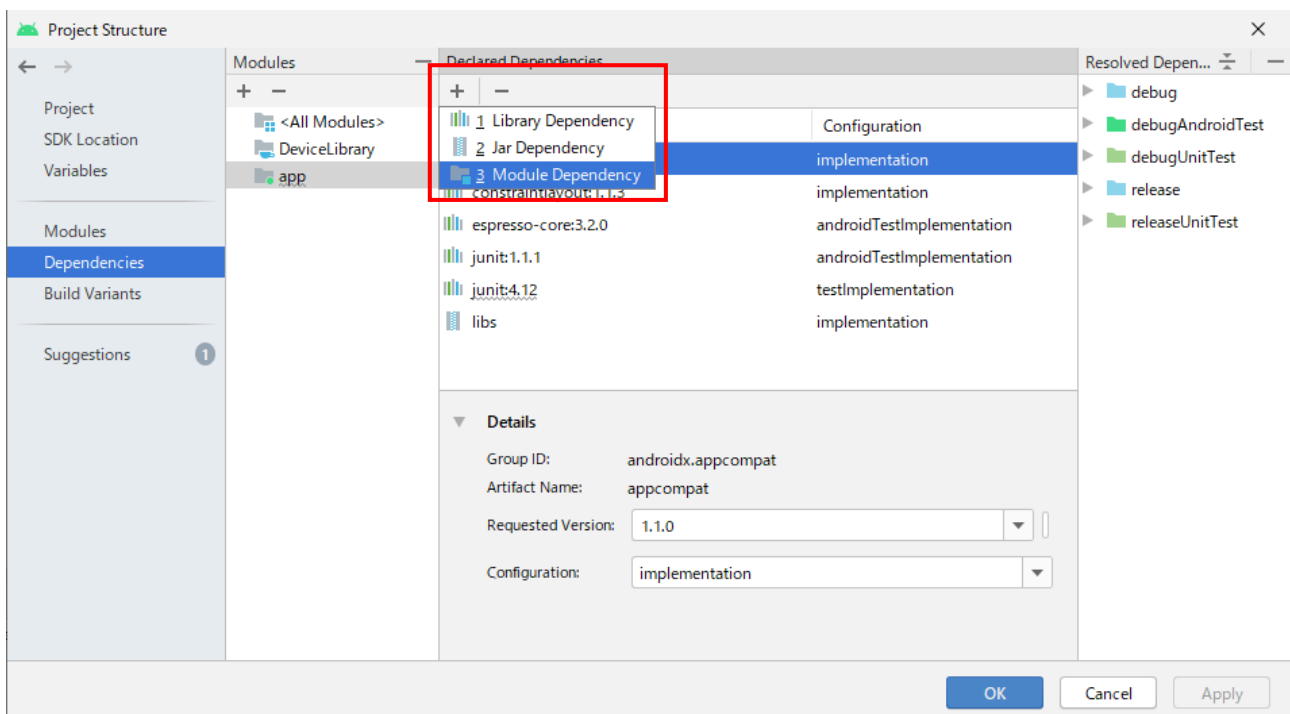
Select [File] -> [Project Structure ...].



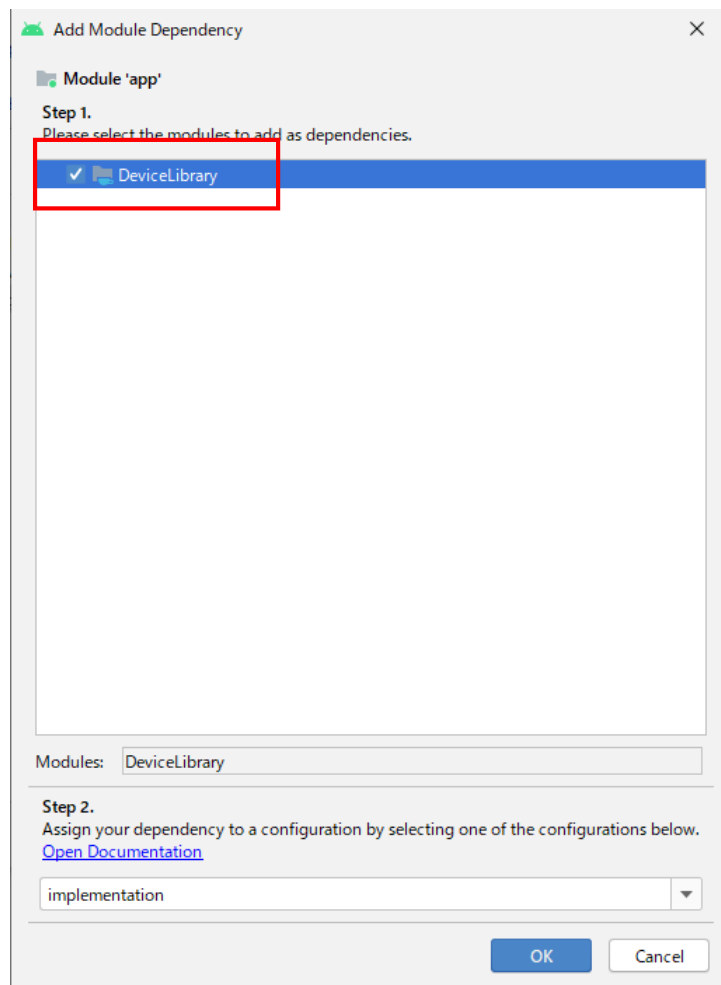
Open the [Dependencies] tab, and check if "DeviceLibrary" exists. If it exists, subsequent operations are unnecessary.



Select "+" button and select [Module Dependency].



Check the "DeviceLibrary" and select "OK".

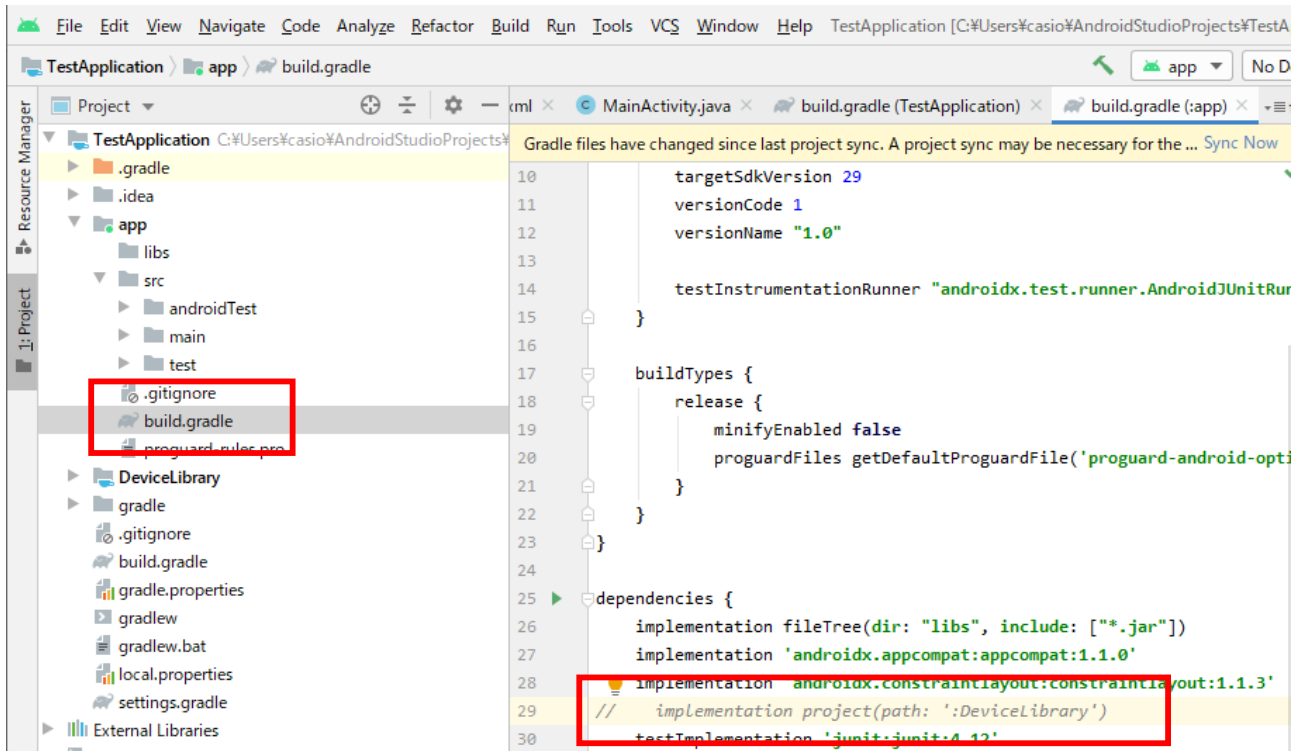


This completes the import of the DeviceLibrary.

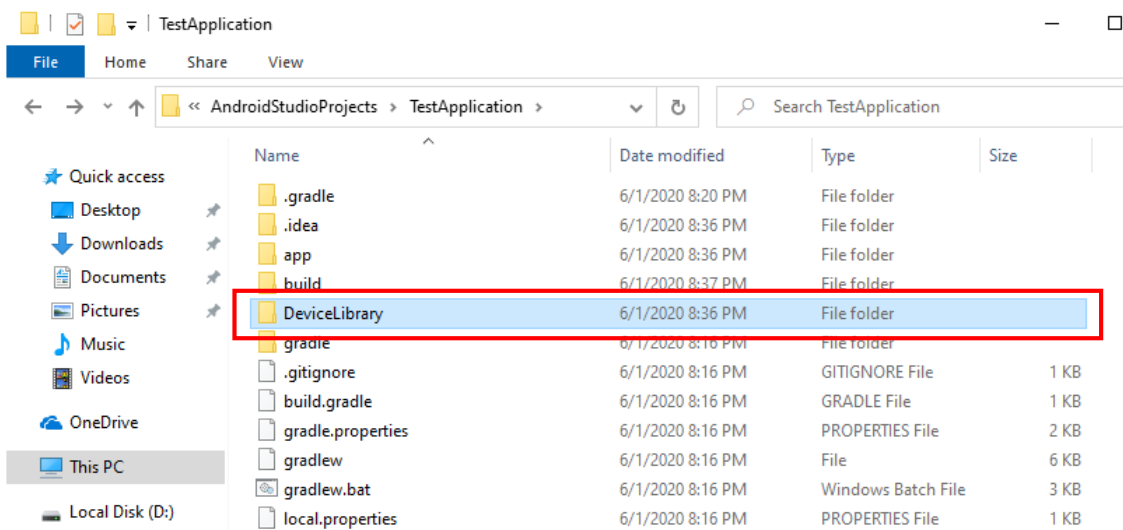
Updates

To update the DeviceLibrary imported in the project of Android Studio (exchange to the new version), need to unregister them and import new one.

To unregister the DeviceLibrary, remove the implementation of the DeviceLibrary in build.gradle.



Delete the DeviceLibrary folder in the root folder of the project.



Import new DeviceLibrary as in the described before section.

Finally, undo the comment out of build.gradle and select "Sync".

The update of the DeviceLibrary is complete.

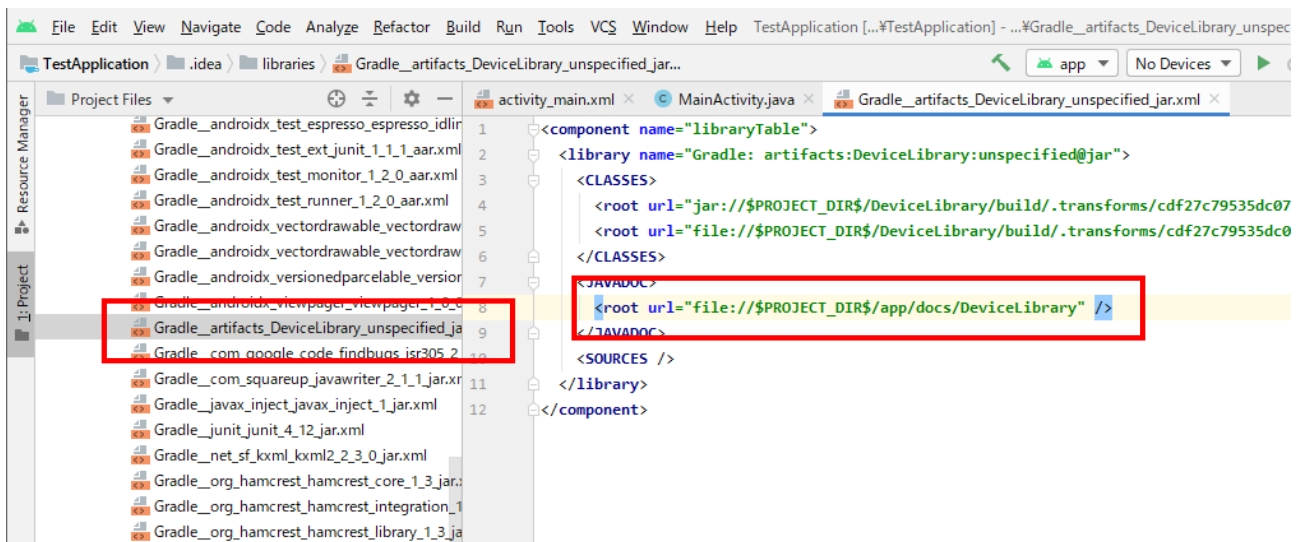
Help files

This section explains how to enable pop-up help (pops up help message when you move the cursor to the function name) of DeviceLibrary.

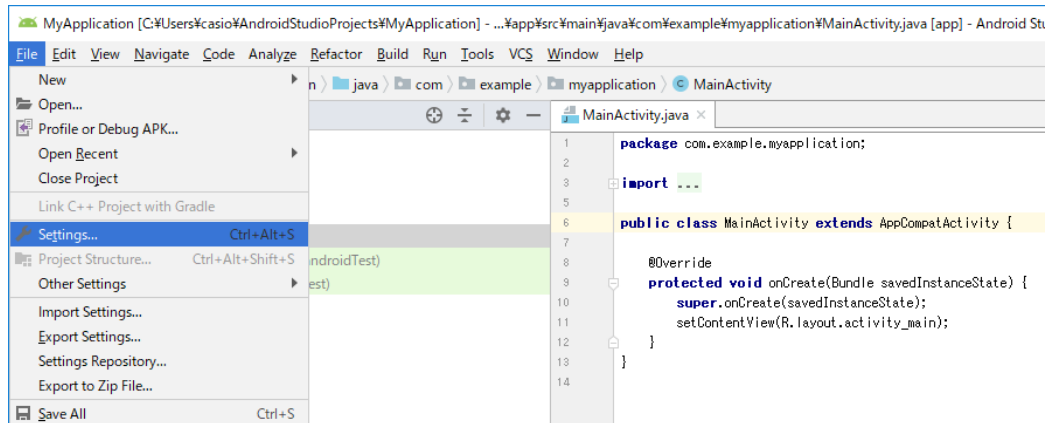
Copy the javadoc in the Basic Development Kit to any folder. Here, it is assumed that it is copied to the "app/docs/DeviceLibrary" folder.

With the import of the DeviceLibrary completed, open "Gradle__artifacts_DeviceLibrary.xml". For "<JAVADOC>" in this xml file, specify the location of the javadoc.
If a xml file is not created, restart Android Studio.

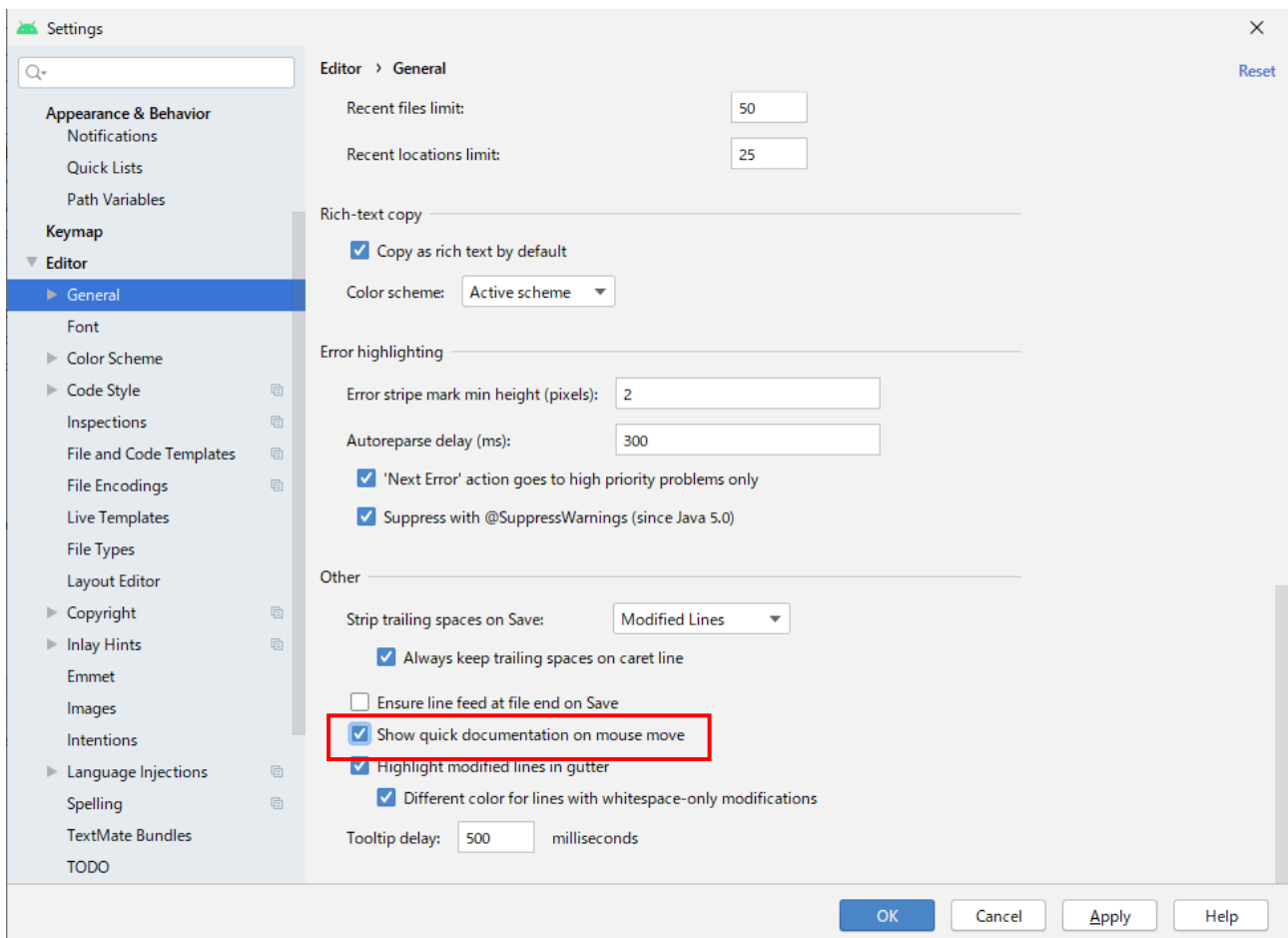
```
<JAVADOC>
<root url="file://$PROJECT_DIR$/app/docs/DeviceLibrary" />
</JAVADOC>
```



Next, select [File] -> [Settings ...] from the Android Studio.



Check "Show quick documentation on mouse move" in [Editor] -> [General].



The registration of help file is complete. Mouse over the function name in Android Studio to display help for the function.

Intent Manager

When creating an application that uses intents, first install the intent manager on the IT-G600/650.

If start the Intent Manager once after installation, it will run in the background and receive intents.

Next, create an application that issues the intent.

Refer to the "Device Control Manual" for details on issuing intents.