

ロボホン認定試験チュートリアル

2.環境構築

Version 1.0.1

Last update 2017/06/20

SHARP CORPORATION

更新履歴

Version	Description	Date
1.0.0	初版作成	2016/09/08
1.0.1	サンプルアプリケーションについて追記	2017/06/20

目次

更新履歴	2
目次	3
1. Android Studio のインストールと設定	9
1.1 JDK のダウンロードとインストール	9
REF _	
Toc46	
11004	
84 ¥h	
9	
1.2 Android Studio のダウンロード	11
REF _	
Toc46	
11004	
85 ¥h	
11	
1.3 Android Studio のインストール	11
REF _	
Toc46	
11004	
86 ¥h	

11

1.4 SDK の追加..... 13

REF _

Toc46

11004

87 ¥h

13

1.5 SDK ツールの設定..... 15

REF _

Toc46

11004

88 ¥h

15

1.6 Instant Run の無効化..... 16

REF _

Toc46

11004

89 ¥h

16

1.7 HVML 拡張子の登録.....	16
----------------------	----

REF _

Toc46

11004

90 ¥h

16

1.8 画面サイズの設定.....	17
-------------------	----

REF _

Toc46

11004

91 ¥h

17

2. ロボホンへの接続	19
-------------------	----

REF _

Toc46

11004

92 ¥h
19

2.1 ロボホンを開発者モードへ設定 19

REF _

Toc46

11004

93 ¥h

19

2.2 ADB Driver のインストール (Windows のみ) 20

REF _

Toc46

11004

94 ¥h

20

3. サンプル・アプリケーションの実行.....21

REF _

Toc46

11004

95 ¥h

21

3.1 サンプル・アプリケーションについて 21

REF _

Toc46

11004

96 ¥h

21

3.2 サンプルアプリの実行 22

REF _

Toc46

11004

97 ¥h

22

4. HVML とオーサリングツール23

REF _

Toc46

11004

98 ¥h

23

4.1 HVML とは 23

REF _

Toc46

11004

99 ¥h

23

4.2 HVML の簡単なサンプル 23

REF _

Toc46

11005

00 ¥h

23

4.3 HVML オーサリングツールエラー! ブックマークが定義されていません。

1. Android Studio のインストールと設定

1.1 JDK のダウンロードとインストール

ロボホンアプリの開発環境である、Android Studio をインストールする前に、Android の開発言語である、Java 言語の開発環境を用意する必要があります。

Java 言語の開発環境は Java Development Kit (JDK)と呼ばれており、お使いの OS に合わせた JDK をインストールする必要があります。

まだ、インストールされていない方は、以下の URL より、お使いの OS 用の JDK をダウンロード／インストールしてください。既にインストール済みの場合は次の章に進んでください。

Java SE Development Kit 8のダウンロード

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

(リンク先に各 OS 用のソフトへのリンクがあります。)

[Java SE Development Kit 8] (2016年8月時点最新版)

Java SE Development Kit 8u101		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
<input checked="" type="radio"/> Accept License Agreement <input type="radio"/> Decline License Agreement		
Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.77 MB	jdk-8u101-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.72 MB	jdk-8u101-linux-arm64-vfp-hflt.tar.gz
Linux x86	160.28 MB	jdk-8u101-linux-i586.rpm
Linux x86	174.96 MB	jdk-8u101-linux-i586.tar.gz
Linux x64	158.27 MB	jdk-8u101-linux-x64.rpm
Linux x64	172.95 MB	jdk-8u101-linux-x64.tar.gz
Mac OS X	227.36 MB	jdk-8u101-macosx-x64.dmg
Solaris SPARC 64-bit	139.66 MB	jdk-8u101-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	98.96 MB	jdk-8u101-solaris-sparcv9.tar.gz
Solaris x64	140.33 MB	jdk-8u101-solaris-x64.tar.Z
Solaris x64	96.78 MB	jdk-8u101-solaris-x64.tar.gz
Windows x86	188.32 MB	jdk-8u101-windows-i586.exe
Windows x64	193.68 MB	jdk-8u101-windows-x64.exe

Oracle Java SE Development Kit ダウンロードページ

(Accept License Agreement のラジオボタンをチェックします。)

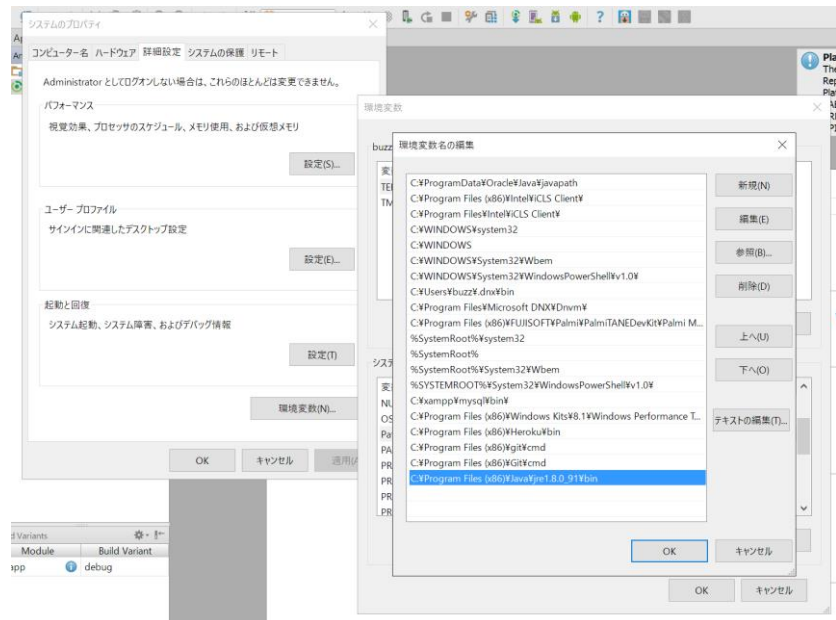
ダウンロードが終了したら、JDK のインストールを行ってください。

Windows の場合

インストール終了後、Windows の場合はパスの設定が必要です。

環境変数の設定から、Path にインストールした JDK の bin フォルダまでのパスを、JAVA_HOME にインストールフォルダまでのパスを設定します。

詳しくは、「JDK パス」などで検索すると手順が公開されています。



Windows でのパス設定画面の例

(システムのプロパティ > 環境変数 > 環境変数名の編集)

1.2 Android Studio のダウンロード

JDK のインストールが終了したら、Android Studio のダウンロードとインストールを行います。

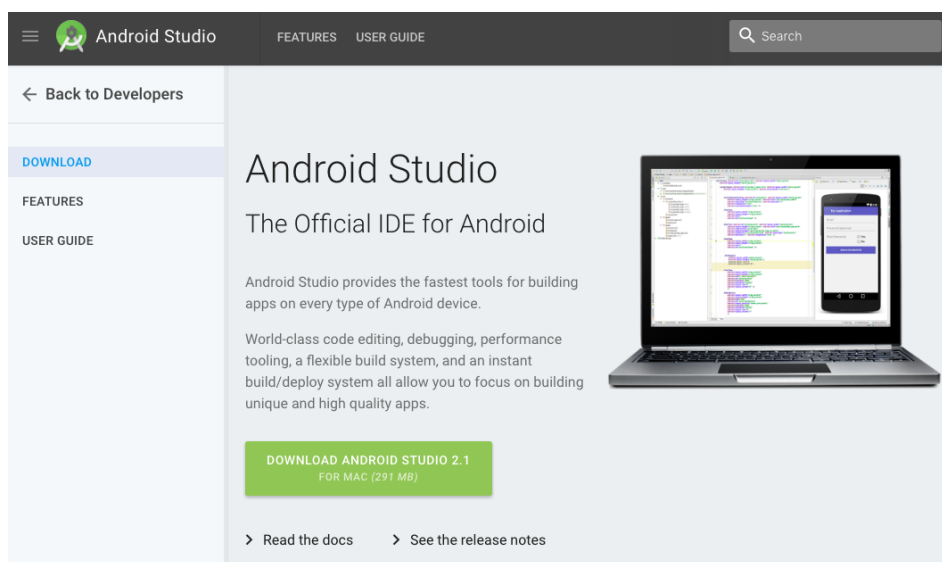
Android Studio は Android 用の総合開発環境です。ロボホンの開発ではバージョン 1.5 以上の Android Studio に対応しています。

Android Studio 2.1のダウンロード

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

(ダウンロードボタンをクリックすると、ご使用の OS に合わせたインストーラーをダウンロード出来ます。)

[Android Studio 2.1] (2016年8月時点最新版)



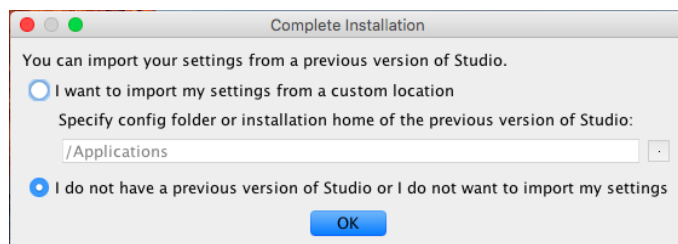
Android Studio ダウンロードページ

(DOWNLOAD ANDROID STUDIO 2.1 ボタンの下にお使いの OS 名が記載されている事を確認の上、ダウンロード)

1.3 Android Studio のインストール

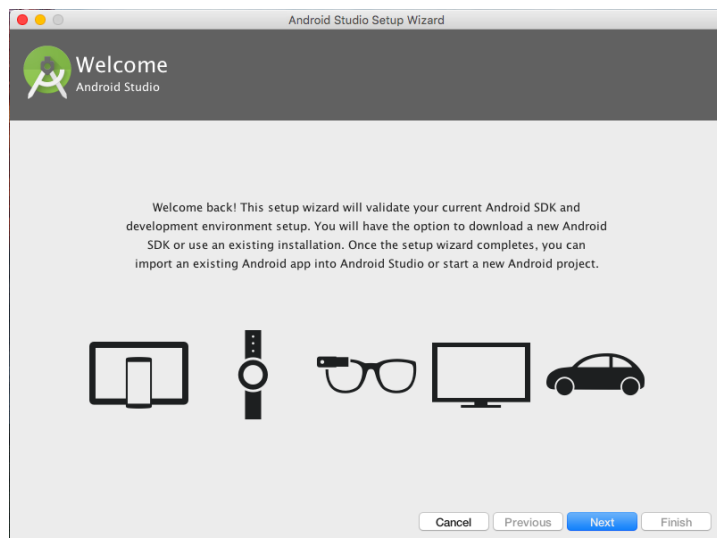
ダウンロードが終了したら、Android Studio をインストールします。

インストーラーは一般的な形ですので、使用されている OS に合わせてインストールを完了させてください。



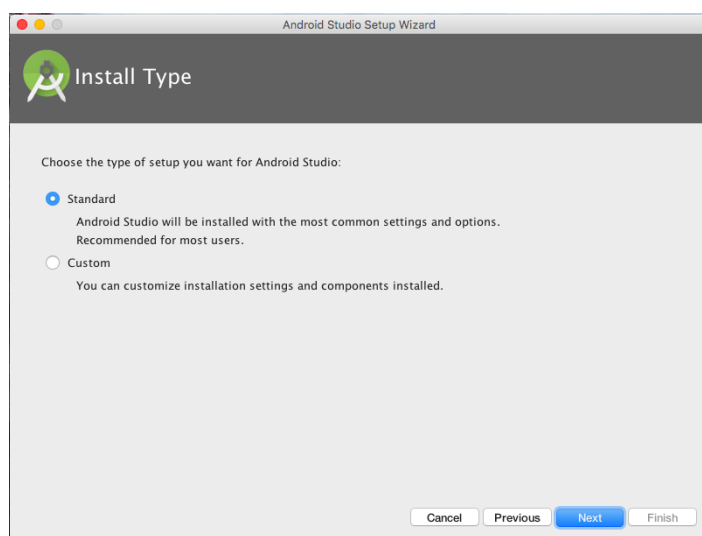
インストーラーを立ち上げると、既に存在する Android Studio の設定ファイルを引き継ぐかの確認画面が表示されます。初めてインストールする場合、過去にインストールしたデータを引き継がない場合は、「I do not have a

previous version of Studio or I do not want to import my settings」が選択された状態のまま OK ボタンをクリックします。



Android Studio インストーラー (macOS)

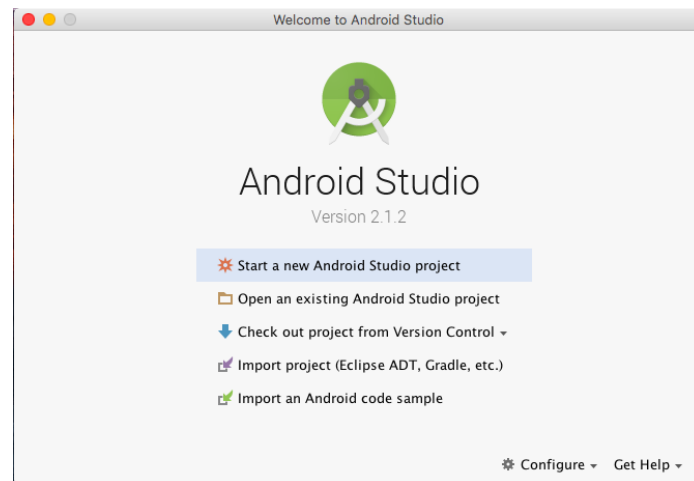
そのまま Next ボタンを押してインストール作業を進めます。



Android Studio インストーラー (macOS)

(Install Type の設定画面)

インストール途中で、Install Type の選択画面がありますが、Standard のままで設定を進めてください。

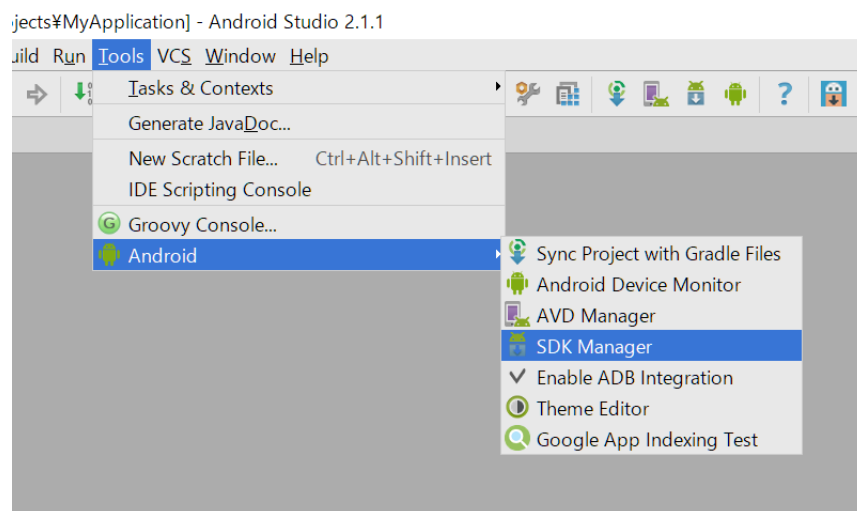


インストールが完了すると、Android Studio が起動し、ウェルカム画面が表示されます。
次の項目でこの画面を使用して設定を進めていきます。

1.4 SDK の追加

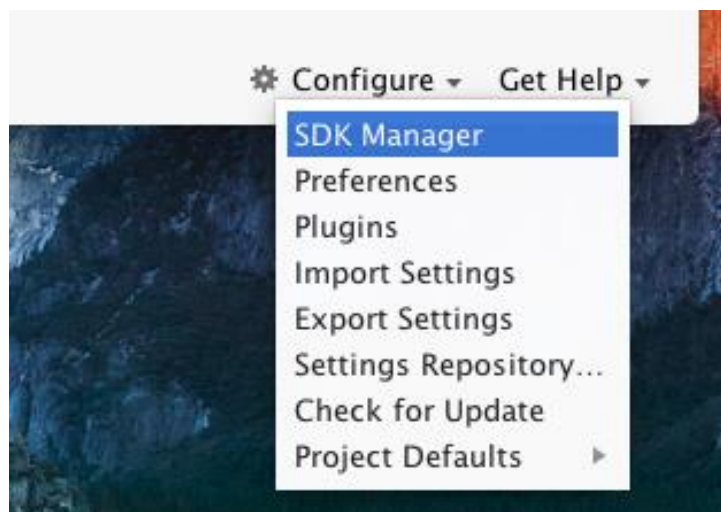
ロボホンの Android は 5.0 系(Lollipop) が使用されています。

Android Studio を Android5.0 系の OS 向けアプリ開発に対応させるため、Android5.0 用の SDK をインストールします。

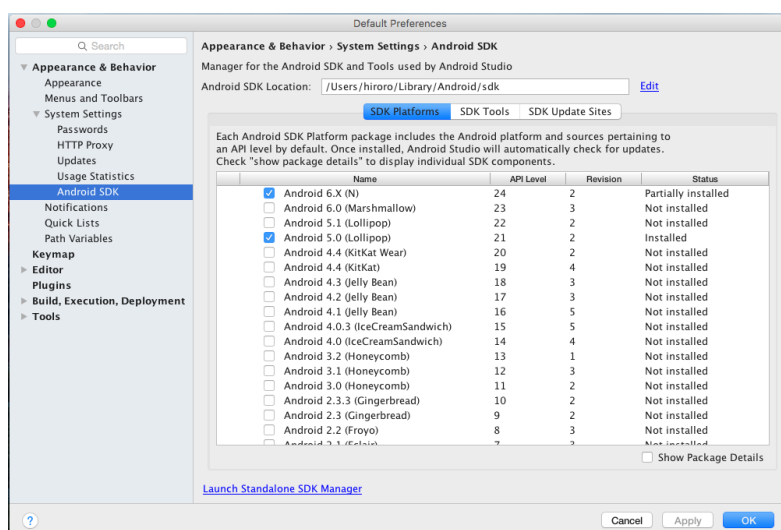


SDK マネージャー画面 (Windows)

Windows の場合は、ツールバーの **[Tools]** > **[Android]** > **[SDK Manager]** を選択してください。

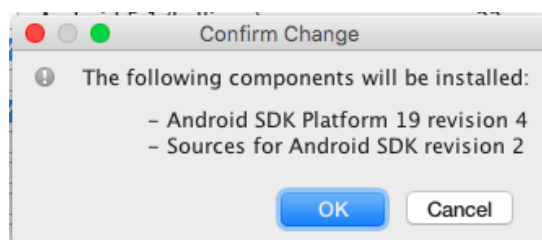


macOS の場合は、Android Studio が起動した際に表示されるウェルカム画面右下に表示されている [Configure] メニューから [SDK Manager] を選択してください。



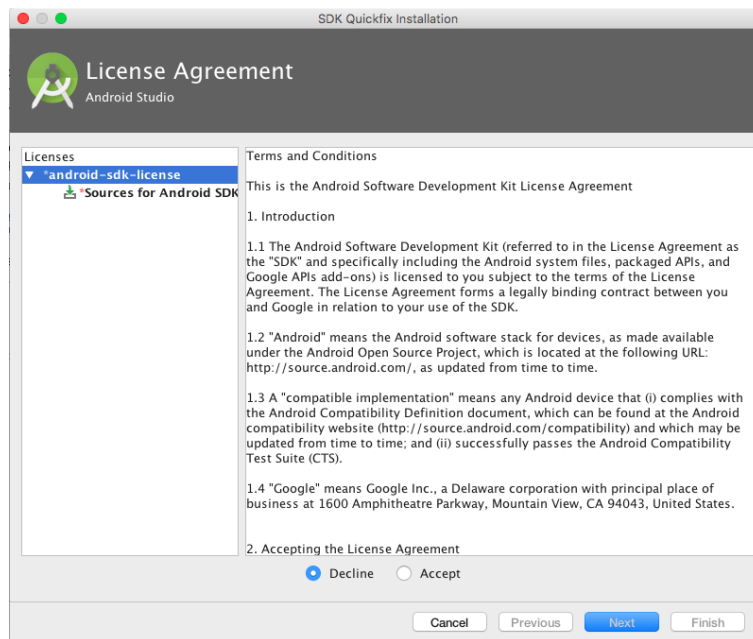
SDK マネージャー画面 (macOS)

[SDK Manager] が表示されたら、右側の pane 内のタブメニューの [SDK Platform] を選択してください。プラットフォームリストから、Android 5.0 (Lollipop)の項目を探し、Status が「Not Installed」であれば、左側のチェックボックスをチェックして、OK ボタンをクリックすると、SDK のインストール画面が表示されますので、インストールを行ってください。



インストールの確認アラート (macOS)

(OK ボタンをクリックしてインストールを進めてください。)



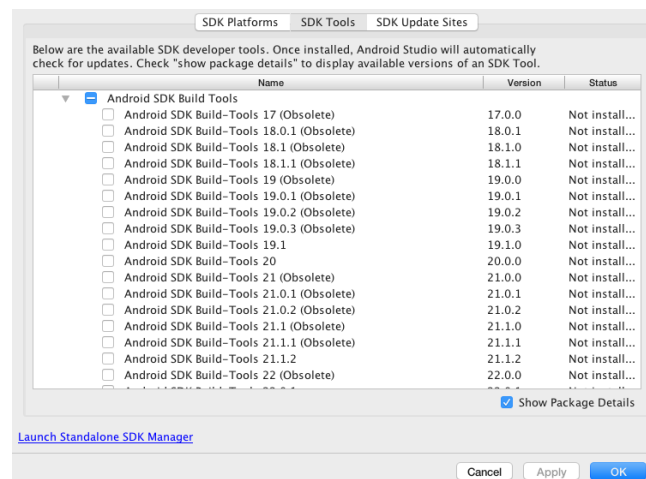
追加 SDK のインストール画面 (macOS)

(Next ボタンをクリックしてインストールを進めてください。)

1.5 SDK ツールの設定

SDK のインストールが完了したら、Android 5.0 用の Android Tools をインストールします。

再度、ウェルカム画面右下の [Configure] メニューから [SDK Manager] を選択し、先ほどの [SDK Platform] の隣のタブ [SDK Tools] を開きます。



SDK Tools の設定画面 (macOS)

リスト表示されている項目の中から、[Android SDK Build Tools] を選択し、チェックします。

その後、下部に表示されている [Show Package Details] をチェックすると、[Android SDK Build Tools] の小項目が表示されますので、表示された小項目の Version が 21 以降のものを全てチェックし、OK ボタンをクリックしてください。

1.6 Instant Run の無効化

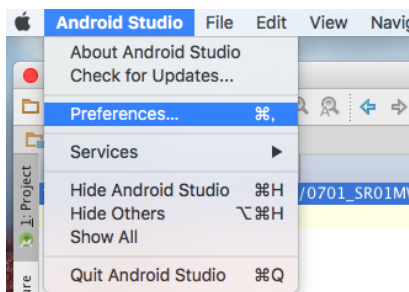
Instant Run 機能は Android アプリを開発中に、作動中のアプリを再インストールしなくても、動作の確認ができる機能です。

この機能が有効になっていると、ロボホン開発では、対話シナリオの実行テスト時に不具合が発生する可能性があるため、無効化する必要があります。

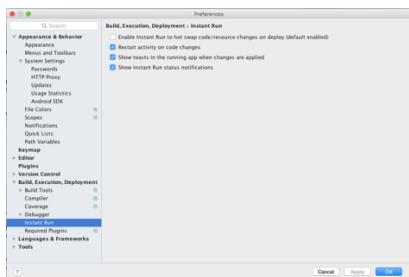
[設定方法]

Android Studio を起動し、新しくプロジェクトを作るか、既存のプロジェクトを読み込んで、Android Studio で開発を行う画面を表示させてください。

1. 左上部の [Android Studio] メニュー内の [Preferences] を選択します。(Windows の場合は File メニュー内に存在します。)



2. 表示されたウィンドウ内の [Build, Execution, Deployment] 項目の [InstantRun] リンクを選択します。



3. [Enable Instant Run to hot swap code/resource changes on deploy (default enabled)] の項目からチェックを外し、OK ボタンをクリックします。

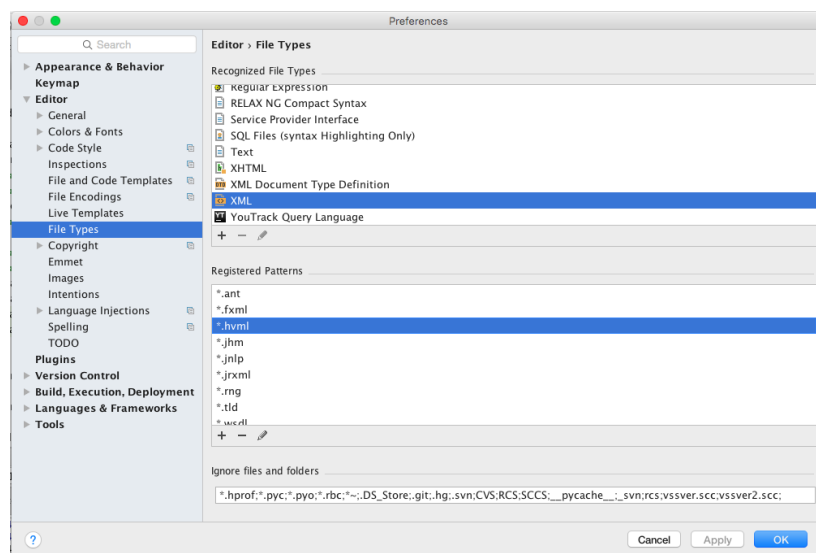
以上で InstantRun の無効化が完了です。

1.7 HVML 拡張子の登録

ロボホンにて音声を使用するアプリを開発する際、XML に準拠したシャープ独自のマークアップ言語である、HVML 言語を使用します。Android Studio をインストールした段階では HVML ファイルを開いても XML ファイルとして認識されないため、XML ファイルとして認識されるように、Android Studio の設定を変更します。

[設定方法]

1. 左上部の [Android Studio] メニュー内の [Preferences] を選択します。(Windows の場合は File メニュー内に存在します。)
2. [Editor] 項目の [File Types] を選択します。
3. [File Types] 項目内の [Recognized File Types] より、XML を選択し、Registered Patterns に*.hvmml を追加します。
4. 最後に OK ボタンを押して、[Preferences] を閉じます。



File Types の設定画面 (macOS)

1.8 画面サイズの設定

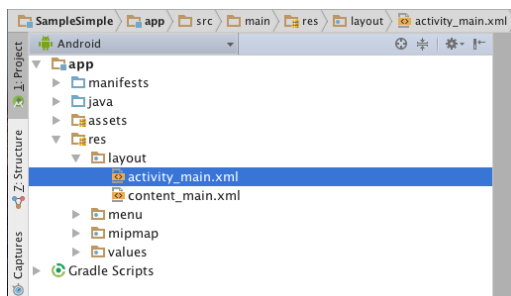
ロボホンには背面に LCD 液晶画面が搭載されており、通常の Android 端末と同様に、画面 UI を使用したアプリケーションの開発も可能です。

Android Studio に搭載されたバーチャル端末の画面レイアウトをロボホンと同じ画面サイズである Galaxy Nexus (4.7" 720 × 1280: xhdpi) に変更します。

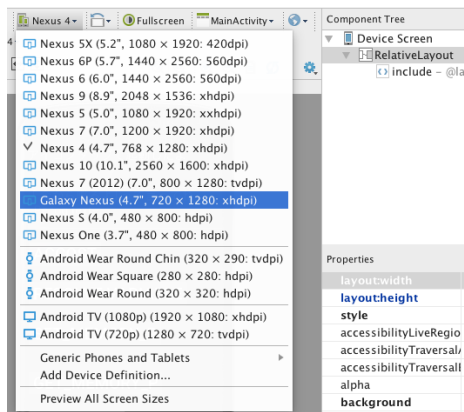
この設定は Android Studio の環境設定ではなく、作成するアプリプロジェクト単位で設定する必要があります。

[設定方法]

1. プロジェクトを開いた状態で、左側のツリービューを [Android] に設定します。



2. ツリービューから `app > res > layout > activity_main.xml` を開きます。



3. バーチャル端末が表示されている部分の上部に「Nexus4」など現在の端末レイアウトが表示されますので、Galaxy Nexus へ変更します。

2. ロボホンへの接続

2.1 ロボホンを開発者モードへ設定

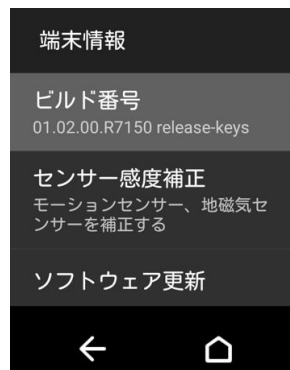
ロボホンは通常の Android 端末と同じように、そのままの状態では開発者モードが無効になっています。
ロボホン本体へアプリケーションを転送するには開発者モードを有効にする必要があります。

[設定方法]

1. ロボホン本体の液晶画面にて [設定] 項目を表示します。

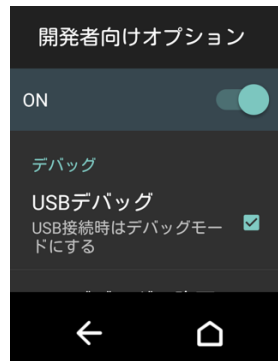


2. 設定項目内の [端末情報] を選択し、ビルド番号の記載された部分を 7回連続してタップ します。

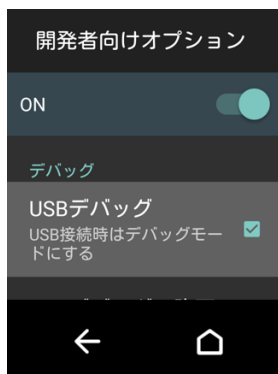


開発者モードが選択できるようになりました。続いて、開発者モードを有効化します。

3. ロボホン本体の液晶画面にて [設定] 項目を表示します。
4. [その他] の項目を選択し、[開発者向けオプション] をタップし、スライダーをONに設定します。



5. 4で設定したスライダーの下項目の **【USBデバッグ】** チェックボックスをチェックします。



2.2 ADB Driver のインストール (Windows のみ)

Windows マシンとロボホンを接続する際は、Windows マシンに ADB Driver をインストールする必要があります。
macOS の場合はインストールの必要がありませんので、この項目は飛ばしてください。

[設定方法]

1. ADV Driver (usb_driver_SHARP_RoBoHoN_r1.0.zip)を解凍します。
2. ロボホンとWindowsマシンを付属のUSBケーブルで接続します。
3. デバイスマネージャーが自動的に実行され、ドライバのインストールに失敗します。
4. デバイスマネージャーを開き、**【Android ADB Interface】** を右クリックし、ドライバーソフトウェアの更新を選択します。
5. 参照ボタンを押して先ほど解凍したフォルダを選択し次へをクリックします。
6. **【“Sharp Corporation”からのソフトウェアを常に信頼する】** をチェックし、インストールをクリックします。
7. インストールが完了したら、閉じるボタンで、ウィンドウを閉じます。

これで Windows マシンとロボホンの接続が可能となります。

3. サンプルアプリの実行

3.1 サンプルアプリについて

ロボホンの SDK には複数のサンプルアプリが付属しています。
アプリ開発の前に、サンプルアプリをロボホンで実行してみましょう。

「sampleSimple」

発話を中心に、幾つかのイベント処理を伴うサンプルアプリです。

「SampleScenario」

HVML で様々な発話を行うサンプルアプリです。

「SampleProjector」

ロボホンのプロジェクターで画像を投影するサンプルアプリです。

「SampleAddressBook」

アドレス帳に登録されたオーナー情報などを取得するサンプルアプリです。

「SampleDance」

ダンスのサンプルアプリです。

「SampleMultilingual」

多言語対応（日本語、英語、中国語）のサンプルアプリです。

「SampleCamera」

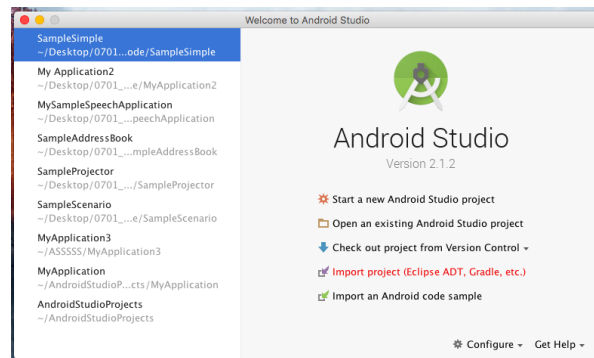
カメラ機能（顔認識、写真撮影、動画撮影）のサンプルアプリです。

3.2 サンプルアプリの実行

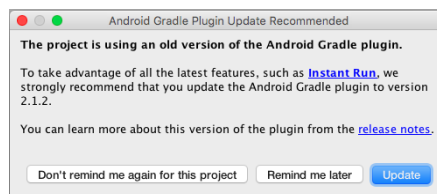
サンプルアプリを実行するには Android Studio にてサンプルアプリを読み込み、ロボホンに転送します。
サンプルアプリは SDK 関連資料内の zip ファイル「0701_SR01MW_SampleCode」内に含まれております。

[サンプルアプリの実行方法]

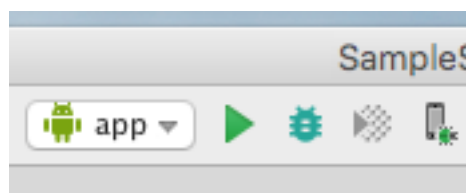
1. Android Studio の [ウェルカム画面] を開きます。



2. ウェルカム画面のメニューから、[Import project (Eclipse, ADT, Gradle, etc)] を選択します。
3. 実行したいサンプル・アプリのフォルダを選択します。
4. 読み込みが開始され、読み込み途中で、[Import project (Eclipse, ADT, Gradle, etc)] の update をリクエストする alert が表示されますので、update をクリックします。



5. プロジェクトの読み込みが完了したら、Android Studio の上部中央にある、[再生マーク] をクリックして、ロボホンへの転送を開始します。



ロボホン上でアプリが実行され、動作の確認を行うことができます。

4. HVML とオーサリングツール

4.1 HVML とは

HVML は電子機器と人間の対話を表現するためのシャープ独自のマークアップ言語です。

ロボホンの発声、音声認識など、対話機能を実装する際に使用します。

詳しい仕様等は SDK 付属資料の[0601_SR01MW_HVML2.0_Specification_V01_00_00.pdf]などをご参照ください。

4.2 HVML の簡単なサンプル

それでは、HVML を使用して、簡単な対話サンプルを作成してみましょう。

HVML のサンプルコード

```
<hvm1 version="2.0">
  <head>
    <version value="1.0"/>
    <producer>jp.co.sharp.producer</producer>
    <scene value="jp.co.sharp.producer.sample"/>
    <situation topic_id="0001" trigger="user-word">${Lvcsr:Basic} eq こんにちは</situation>
  </head>

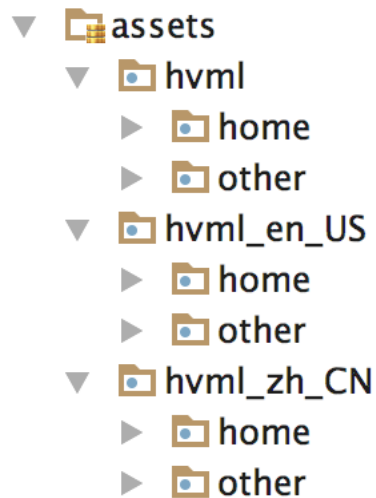
  <body>
    <topic id="0001" listen="false">
      <action index="1"><speech>こんにちは。今日もいい天気ですね。</speech></action>
    </topic>
  </body>
</hvm1>
```

上記の HVML はユーザーが「こんにちは」と発声した際に、ロボホンが「こんにちは。今日もいい天気ですね。」と返答する対話を表現したサンプルです。

このようなシンプルな対話以外にも、複数の返答パターンを設けたり、複数の対話をリンクしたり、色々な対話を表現できます。

4.3 多言語対応と HVMML

ロボホン SDK の 1.2.0 より、ロボホンに多言語（日本語、英語、中国語）を対応させることが出来るようになりました。本体の言語設定を変更することにより、対応した言語の HVMML を実行することが可能です。（アプリ内で一時的に言語設定を変更することも可能です。）



テンプレートからアプリを作成すると、hvml フォルダの他に、hvml_en_US フォルダと hvml_zh_CN フォルダが作成されます。それぞれのフォルダは、本体の言語設定に対応しているので、言語ごとに hvml ファイルを作成することが出来ます。

多言語対応を行うの TIPS

本体の言語設定に対応したフォルダが存在しない場合は、日本語のフォルダが自動的に選択されます。