Institut für Informatik
Lehrstuhl Human Centered Multimedia

Prof. Dr. Elisabeth André
Simon Flutura

Universität Augsburg University

SSI on Android, SS 2015

1. Introduction

SSI - Social Signal Iterpretation Framework has been portet to Linux with an Android-port in the works. Devices like the Raspberry PI, as well as several mobile devices can now be used. This guide shows you how to set up SSI on Android

Task 1: Prepare the setup (**note:** crosscompile from Windows)

1. GCC and G++ 4.8 or later have to be installed, because SSI relies on C++11 standard.
2. CMake is needed as build-system.
3. Android SDK and NDK are needed and has to be setup on the your system.
4. Apache ANT is needed to build an APK and has to be in PATH. Instructions can be found here:
   `https://developer.android.com/ndk/guides/setup.html`

Task 2: Get SSI

SSI can be found here:

1. from git `https://hcm-lab.de/git/ssi/ssi-pub.git`
2. from subversion `https://hcm-lab.de/svn/Johannes/openssi/trunk/`
3. from `cleanport.tar.gz` archive from:
   `https://skathi.net/pub/simon/ssi/clean_port_.tar.gz`

Task 3: Prepare Buildsystem for APK

for apk creation some additional steps are neccessary.

1. copy * from `trunk/docs/apk` to trunk.
2. additionally copy `trunk/docs/apk/android/AndroidManifest.xml` to trunk.

3. edit android.apk.cmake to match your device regarding `ANDROID_APK_API_LEVEL`

note to rebuild the APK you have to:

```
make clean -f plugins\androidSensors\tools\Makefile
```

your apk will be build in `trunk/bin_cmake/Apk/bin/`

Task 4: Build SSI

Install CMake; create a build directory next to your SSI-root called trunk in the following text. Start CMake, select trunk as source directory and your build directory..

Note that your binaries will be put to `trunk/bin_cmake` with "make install" the final step.

following variables have to be set to match your device/emulator:

```
ANDROID_ABI=armeabi-v7a
ANDROID_NATIVE_API_LEVEL=android-21
```

following variables have to be set to match your setup:

```
ANDROID_NDK=absolute/path/to/ndk
CMAKE_MAKE_PROGRAM=filepath/to/make/binary (NDK/prebuilt/%PLATTFORM%/bin/make.exe)
```

Press configure and lets follow the wizard:

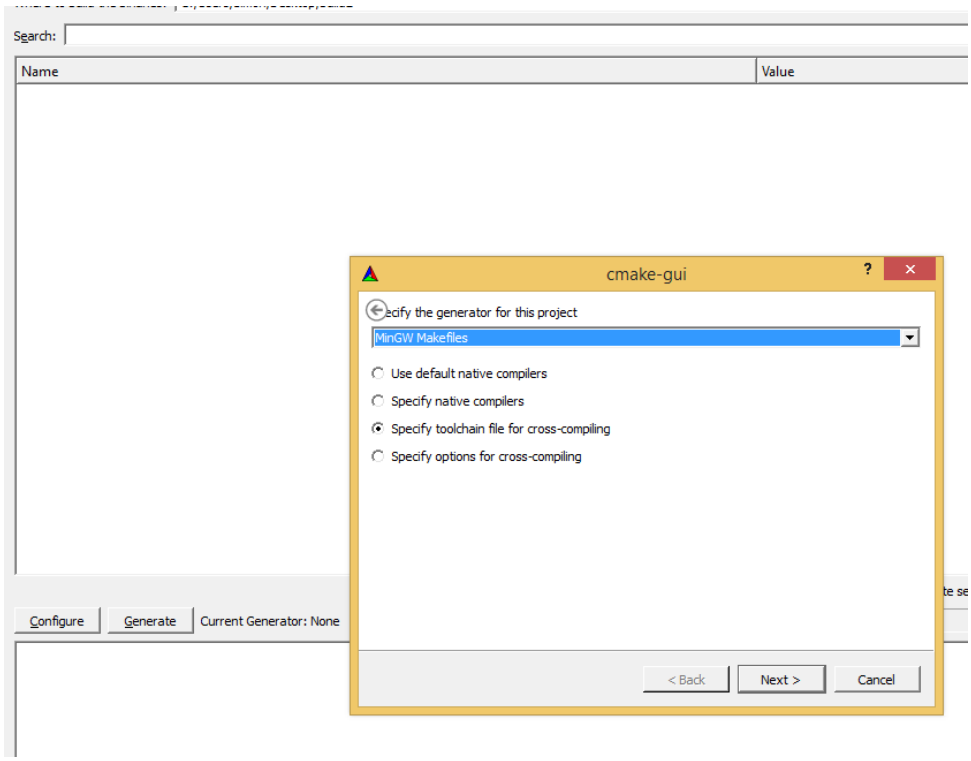

Figure 1: In CMake-GUI: select MinGW Makefiles; toolchain for crosscompiling

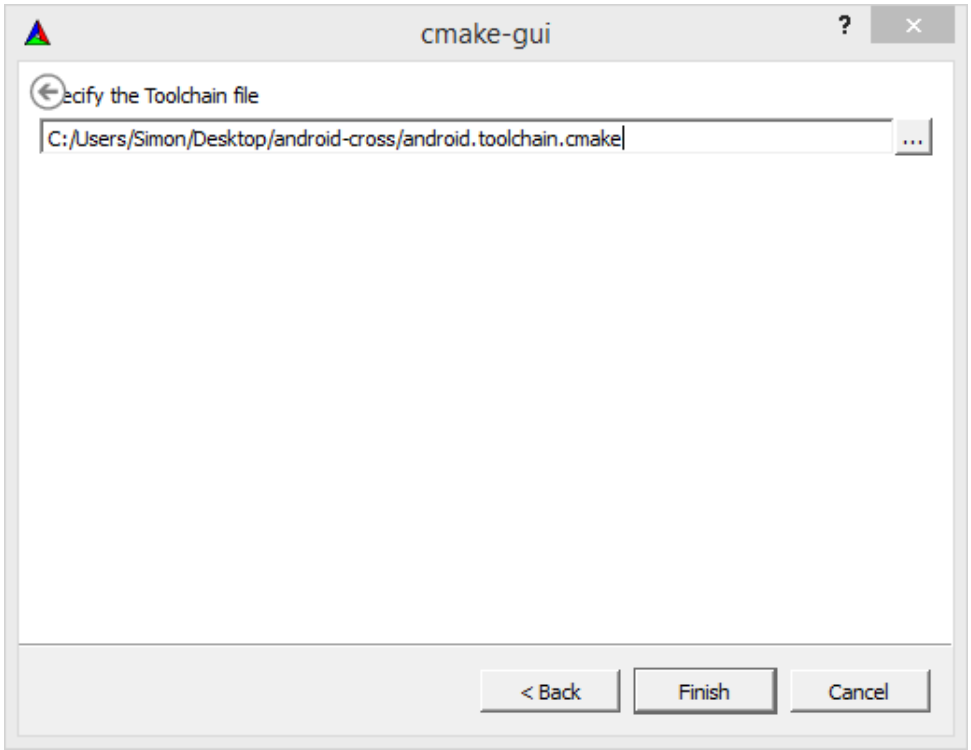

Figure 2: select android.toolchain.cmake from trunk/docs/ssi-port-cmake/intro-android-from-win

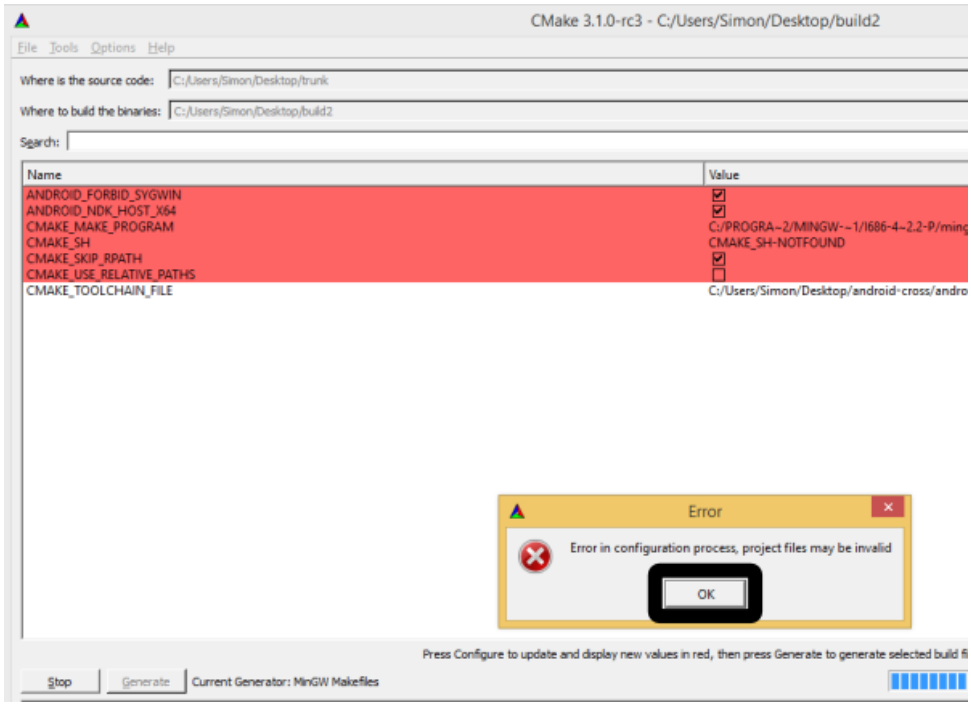

Figure 3: Ignore the error

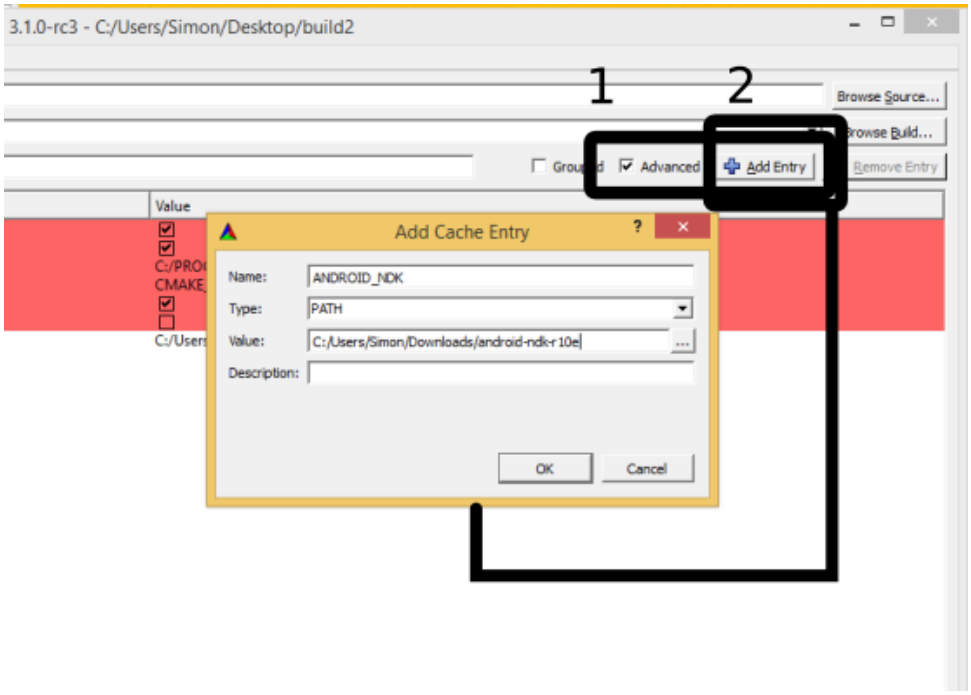

Figure 4: change to advanced view and add entries fitting your device.

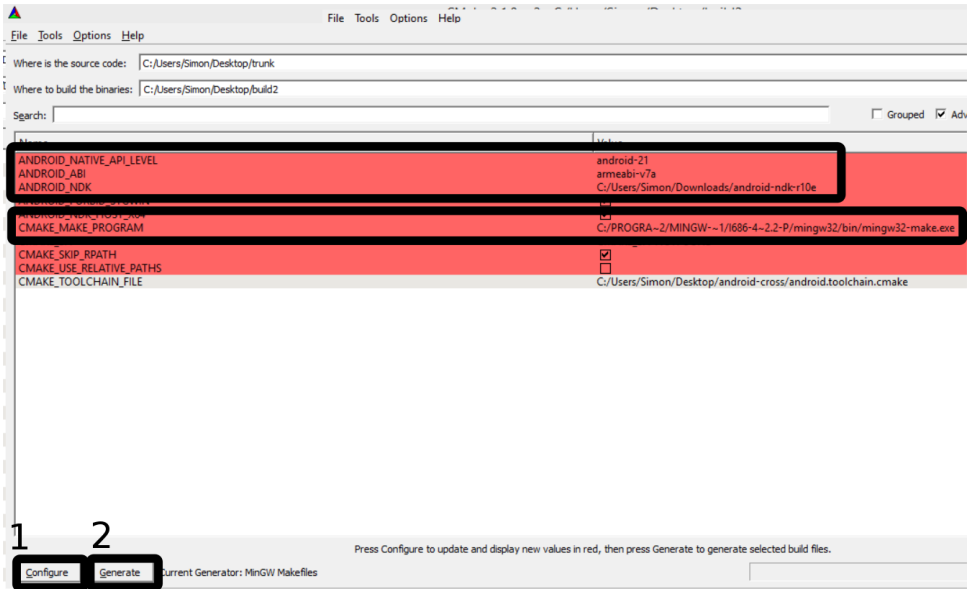

Figure 5: marked values should be set; configure; generate

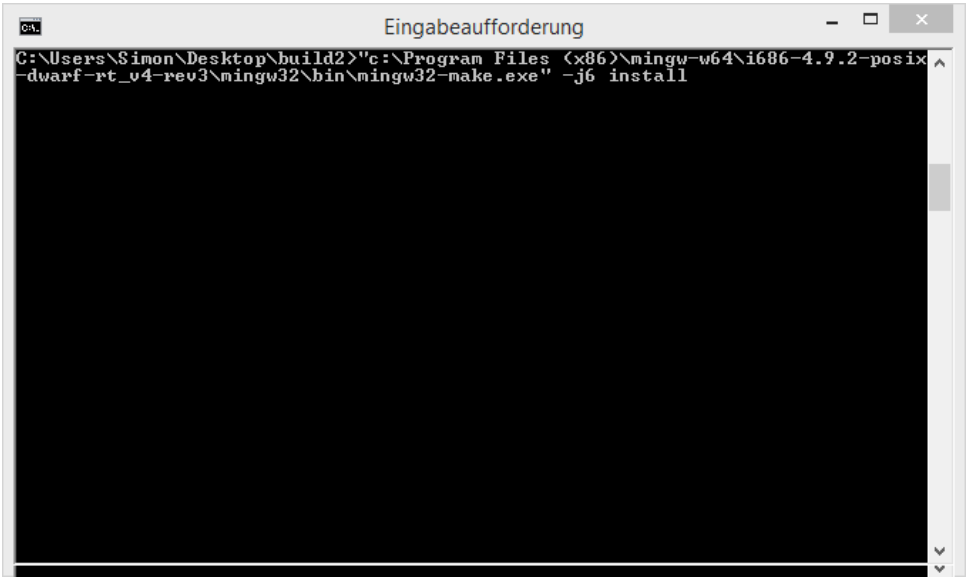

Figure 6: find your build system in your cmake build dir (where to build the binaries); type "make -j6 install" to rebuild the APK first run "make clean"

Task 5: Prepare Buildsystem for APK

for apk creation some additional steps are neccessary.

1. setup ant build system to be in PATH
2. copy * from `trunk/docs/apk` to trunk.
3. additionally copy `trunk/docs/apk/android/AndroidManifest.xml` to trunk.
4. edit android.apk.cmake to match your device regarding `ANDROID_APK_API_LEVEL`

your apk will be build in `trunk/bin_cmake/Apk/bin/`

Task 6: Run tests

Your freshly build binarys are located in ssi roots `bin_cmake` directory:

`trunk/cmake_bin/Android`

(keep in mind where your SDK is placed; if it is not in PATH adapt commands below!)

1. create and start your emulator:

   ```
   android create avd -t 2 -n name
   ```

   where -t names the correct target from android list targets
2. copy files to emulator:

   ```
   adb -e push trunk/cmake_bin/Android /data/test
   adb -e shell chmod 777 /data/test/*
   adb -e shell
   ```
3. Set the `LD_LIBRARA_PATH`

   `export LD_LIBRARY_PATH=".:/system/lib/"` or fixed to the SSI root path, so shared libraries can be found.
4. To test a simple pipeline: then run `./ssiandroid_test` from `data/test/`. The correct working directory is needed, so the shared plugins can be found.
5. Apart from c++ code one can as well run xml pipelines via:

   ```
   ./xmlpipe test.pipeline
   ```
6. The same steps are valid for rooted Android devices. Use `adb -d` to connect to your device.

Task 7: Choose IDE

To further develop SSI on Linux QtCreator is recommended. Do not forget to install the GNU debugger GDB.

For better Android Integration Android-Studio might be a wise choise.

Setting up remote GDB follows..

Have fun.