

1. Set Environment Variables

These commands tell the compiler where your tools are located.

```
export ANDROID_SDK_ROOT=/home/iot/Android/Sdk
export ANDROID_NDK_ROOT=/home/iot/Android/Sdk/ndk/21.3.6528147
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64
export PATH=$JAVA_HOME/bin:$PATH
```

2. Prepare the Build Directory

This clears old "Desktop" build junk and sets up a fresh Android configuration.

```
# Delete the old build folder
rm -rf /home/iot/finali_demo/build
```

```
# Create a fresh one
mkdir -p /home/iot/finali_demo/build
cd /home/iot/finali_demo/build
```

```
# Generate the Android Makefile using Qt's qmake
/home/iot/Qt/5.15.2/android/bin/qmake /home/iot/finali_demo/finali_demo.pro -spec android-clang
```

3. Compile the C++ Code

This builds the core logic into libraries for both 32-bit and 64-bit architectures.

```
# Compile using 4 CPU cores
make -j4
```

```
# Install compiled libraries into the local android-build folder
mkdir -p android-build
make install INSTALL_ROOT=/home/iot/finali_demo/build/android-build
```

4. Generate the Android APK

This uses the Qt deployment tool to wrap your code into a installable .apk file.

```
/home/iot/Qt/5.15.2/android/bin/androiddeployqt \
--input android-finali_demo-deployment-settings.json \
--output android-build \
--android-platform android-29 \
--jdk /usr/lib/jvm/java-1.8.0-openjdk-amd64 \
--gradle
```

5. Install and Launch

The final step to get the app onto the board hardware.

```
# Install to the board via USB OTG
adb install android-build/build/outputs/apk/debug/android-build-debug.apk
```

```
# (Optional) Launch the app immediately
```

```
adb shell am start -n  
org.qtproject.example.finali_demo/org.qtproject.qt5.android.bindings.QtActivity
```