QT Cross Compile

Cross-compiling QT on x86 platforms to rockchip arm64 platforms

On x86 platforms

Setting up the cross-compilation toolchain

```
sudo apt-get install build-essential cmake
wget https://releases.linaro.org/components/toolchain/binaries/7.3-2018.05/
aarch64-linux-gnu/gcc-linaro-7.3.1-2018.05-x86_64_aarch64-linux-gnu.tar.xz
sudo tar xvf gcc-linaro-7.3.1-2018.05-x86_64_aarch64-linux-gnu.tar.xz  -C /usr/
local/
```

Download QT source code

```
mkdir -p pro/qtSourceDir && cd pro/qtSourceDir
wget https://download.qt.io/archive/qt/5.12/5.12.2/single/qt-everywhere-
src-5.12.2.tar.xz
tar -xvf qt-everywhere-src-5.12.2.tar.xz
```

Other versions to download.

Installing the QT Environment

- Modify qmake.conf

before modification

```
#
# qmake configuration for building with aarch64-linux-gnu-g++
#
```

```
MAKEFILE_GENERATOR      = UNIX
CONFIG                 += incremental
QMAKE_INCREMENTAL_STYLE = sublib

include(../common/linux.conf)
include(../common/gcc-base-unix.conf)
include(../common/g++-unix.conf)

# modifications to g++.conf
QMAKE_CC                = aarch64-linux-gnu-gcc
QMAKE_CXX               = aarch64-linux-gnu-g++
QMAKE_LINK              = aarch64-linux-gnu-g++d
QMAKE_LINK_SHLIB        = aarch64-linux-gnu-g++

# modifications to linux.conf
QMAKE_AR                = aarch64-linux-gnu-ar cqs
QMAKE_OBJCOPY           = aarch64-linux-gnu-objcopy
QMAKE_NM                = aarch64-linux-gnu-nm -P
QMAKE_STRIP             = aarch64-linux-gnu-strip
load(qt_config)
```

after modification

```
#
# qmake configuration for building with aarch64-linux-gnu-g++
#

MAKEFILE_GENERATOR      = UNIX
CONFIG                 += incremental
QMAKE_INCREMENTAL_STYLE = sublib

QT_QPA_DEFAULT_PLATFORM = linuxfb
QMAKE_CFLAGS_RELEASE += -O2 -march=armv8-a -lts
QMAKE_CXXFLAGS_RELEASE += -O2 -march=armv8-a -lts


include(../common/linux.conf)
include(../common/gcc-base-unix.conf)
include(../common/g++-unix.conf)

# modifications to g++.conf
QMAKE_CC                = /usr/local/gcc-linaro-7.3.1-2018.05-x86_64_aarch64-
linux-gnu/bin/aarch64-linux-gnu-gcc
QMAKE_CXX               = /usr/local/gcc-linaro-7.3.1-2018.05-x86_64_aarch64
linux-gnu/bin/aarch64-linux-gnu-g++
```

```
QMAKE_LINK              = /usr/local/gcc-linaro-7.3.1-2018.05-x86_64_aarch64-
linux-gnu/bin/aarch64-linux-gnu-g++
QMAKE_LINK_SHLIB        = /usr/local/gcc-linaro-7.3.1-2018.05-x86_64_aarch64-
linux-gnu/bin/aarch64-linux-gnu-g++

# modifications to linux.conf
QMAKE_AR                = /usr/local/gcc-linaro-7.3.1-2018.05-x86_64_aarch64-
linux-gnu/bin/aarch64-linux-gnu-ar cqs
QMAKE_OBJCOPY           = /usr/local/gcc-linaro-7.3.1-2018.05-x86_64_aarch64-
linux-gnu/bin/aarch64-linux-gnu-objcopy
QMAKE_NM                = /usr/local/gcc-linaro-7.3.1-2018.05-x86_64_aarch64-
linux-gnu/bin/aarch64-linux-gnu-nm -P
QMAKE_STRIP             = /usr/local/gcc-linaro-7.3.1-2018.05-x86_64_aarch64-
linux-gnu/bin/aarch64-linux-gnu-strip
load(qt_config)
```

- Create a new autoconfiguration script file in the root QT source code root directory

vi auto.sh

```
#!/bin/sh

export CROSS_COMPILE=/usr/local/gcc-linaro-7.3.1-2018.05-x86_64_aarch64-linux-
gnu/bin/aarch64-linux-gnu-
export PATH=/usr/local/gcc-linaro-7.3.1-2018.05-x86_64_aarch64-linux-gnu/bin:
$PATH
export ARCH=arm64

./configure \
-prefix /opt/qtdir \
-confirm-license \
-opensource \
-release \
-make libs \
-xplatform linux-aarch64-gnu-g++ \
-pch \
-qt-libjpeg \
-qt-libpng \
-qt-zlib \
-no-opengl \
-no-sse2 \
-no-openssl \
-no-cups \
-no-glib \
-no-dbus \
-no-xcb \
-no-separate-debug-info \
```

```
-no-ssl \
-nomake tests \
-nomake examples \
-nomake tools \
-no-sql-sqlite \
-no-iconv \
-skip qt3d \
-skip qtactiveqt \
-skip qtcanvas3d \
-skip qtcharts \
-skip qtconnectivity \
-skip qtdatavis3d \
-skip qtdeclarative \
-skip qtgamepad \
-skip qtandroidextras \
-skip qtdoc \
-skip qtwebchannel \
-skip qtwebengine \
-skip qtwebglplugin \
-skip qtwebview \
-skip qtvirtualkeyboard \
-recheck

make  -j4
sudo make install
```

- Execute the autoconfiguration script

```
sudo chmod +x auto.sh
sudo ./auto.sh
```

If you encounter an error similar to the following

```
error: 'numeric_limits' is not a member of 'std'
```

Under the sol.hpp file, add

```
#include <limits>
```

- Check if the installation was successful

```
test@test-desktop:/opt$ ls /opt/qtInstallDir/
```

```
bin  doc  include  lib  mkspecs  plugins  translations

test@test-desktop:/opt$ ls /opt/qt5.9.4-arm/
 bin  lib  mkspecs
```

Install QTCreator

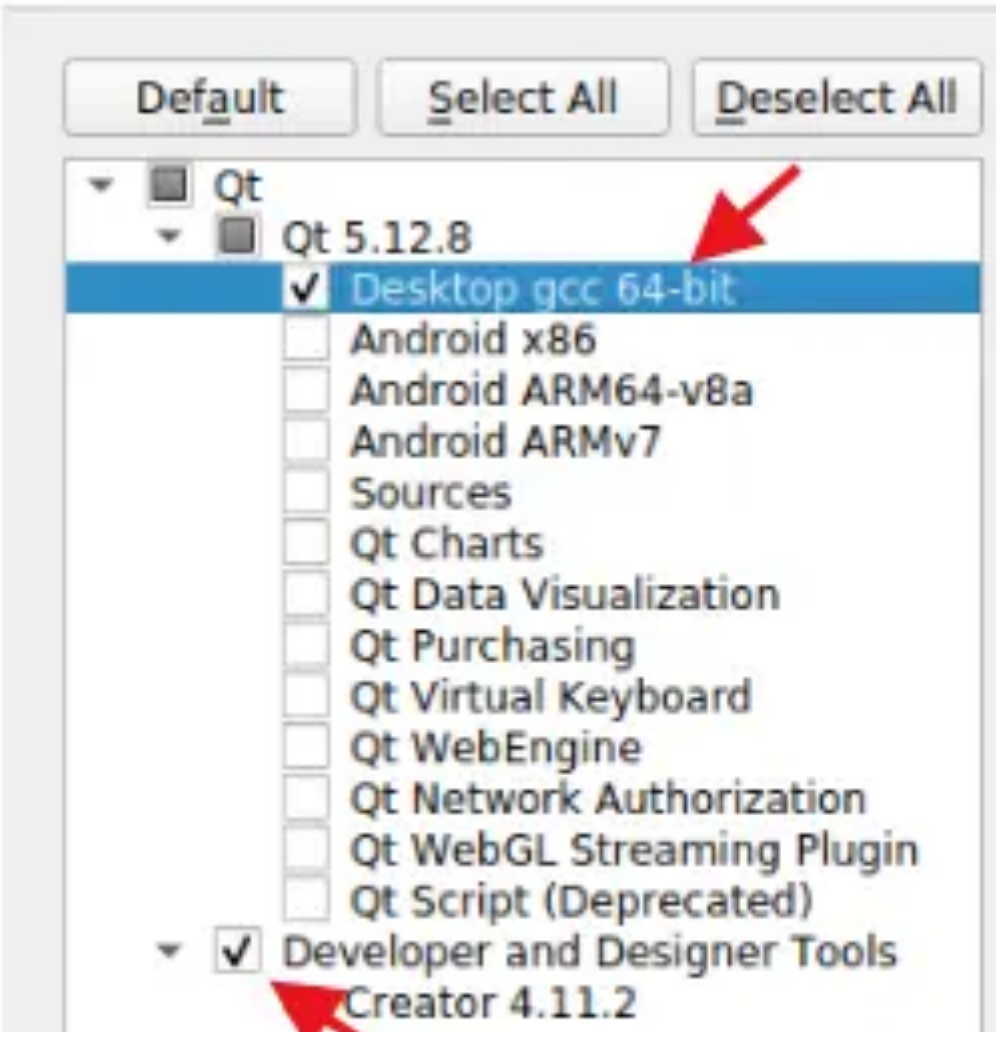
- Download and install tools

```
wget https://download.qt.io/archive/qt/5.12/5.12.2/qt-opensource-linux-x64-5.12.2.run
```

Or download it from here.

- installation

```
sudo chmod +x qt-opensource-linux-x64-5.12.2.run
sudo ./qt-opensource-linux-x64-5.12.2.run
```

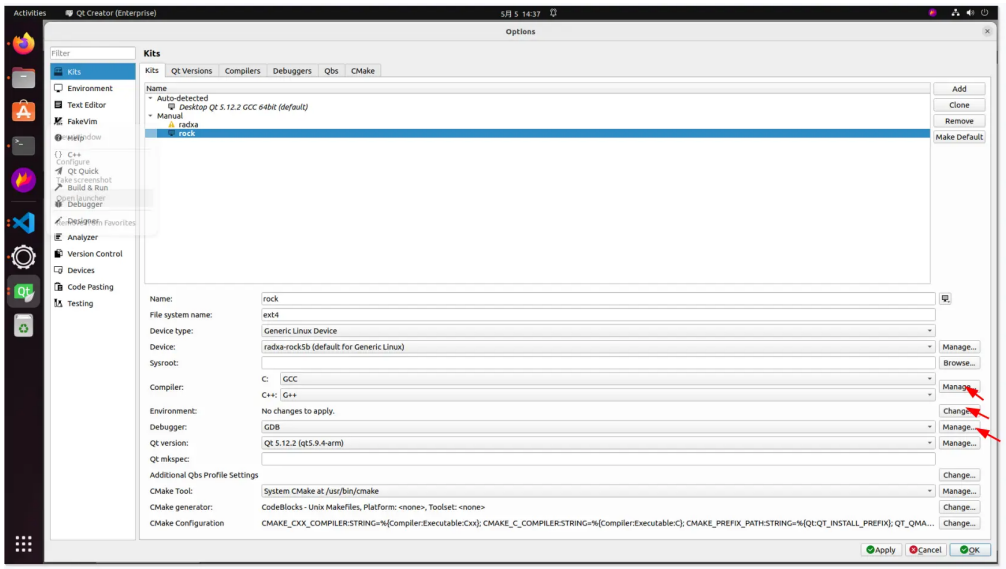


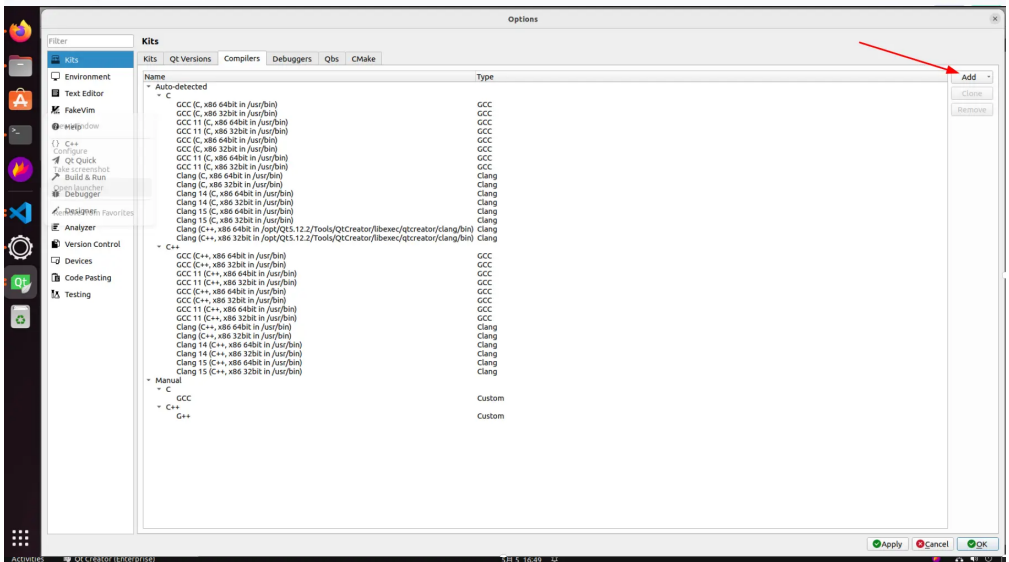
Demo Example Project

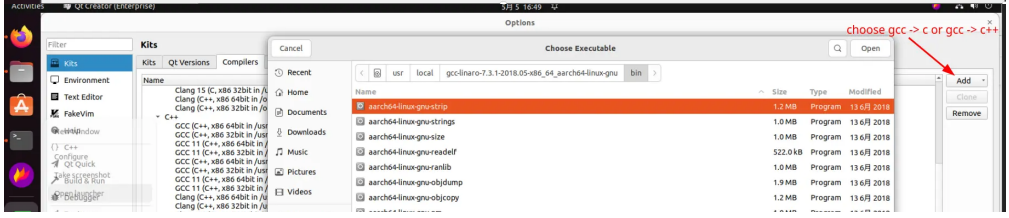
- Required environment

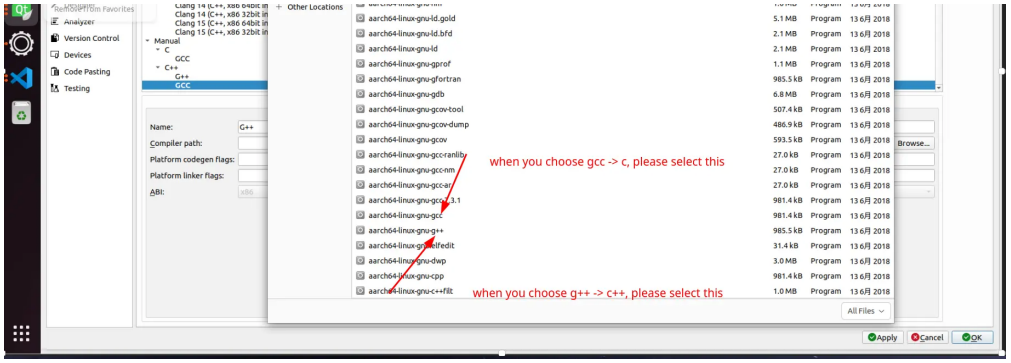
```
sudo apt-get install libgl1-mesa-dev libglu1-mesa-dev -y
```

- Add Tools



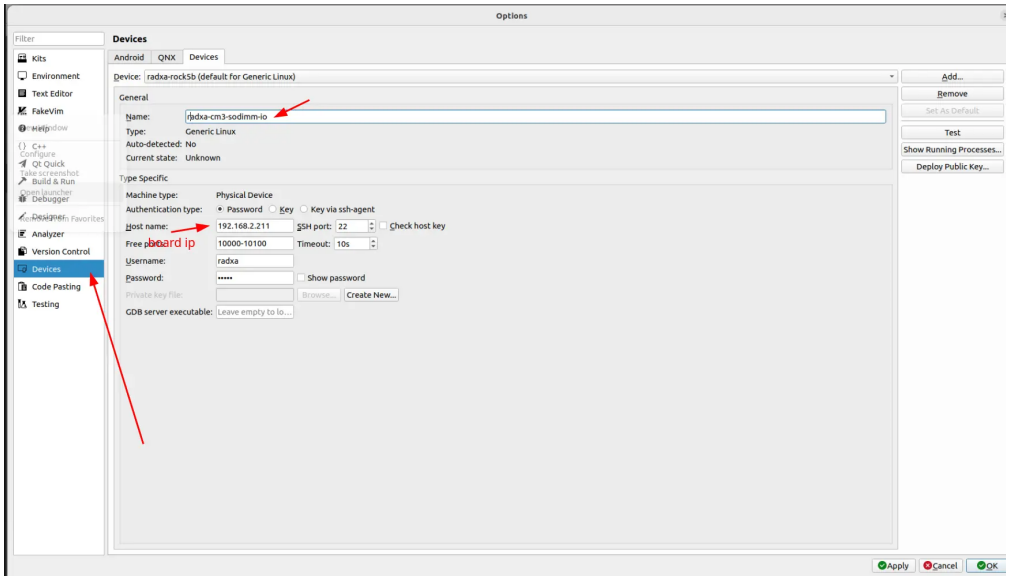




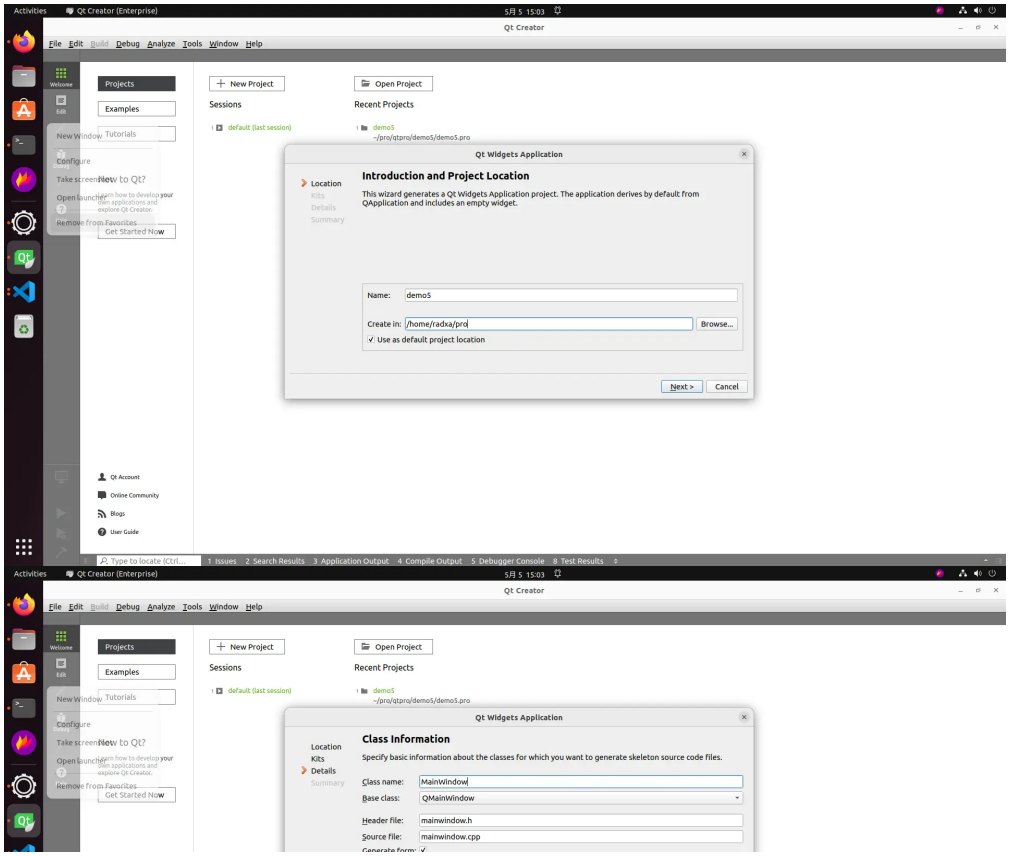


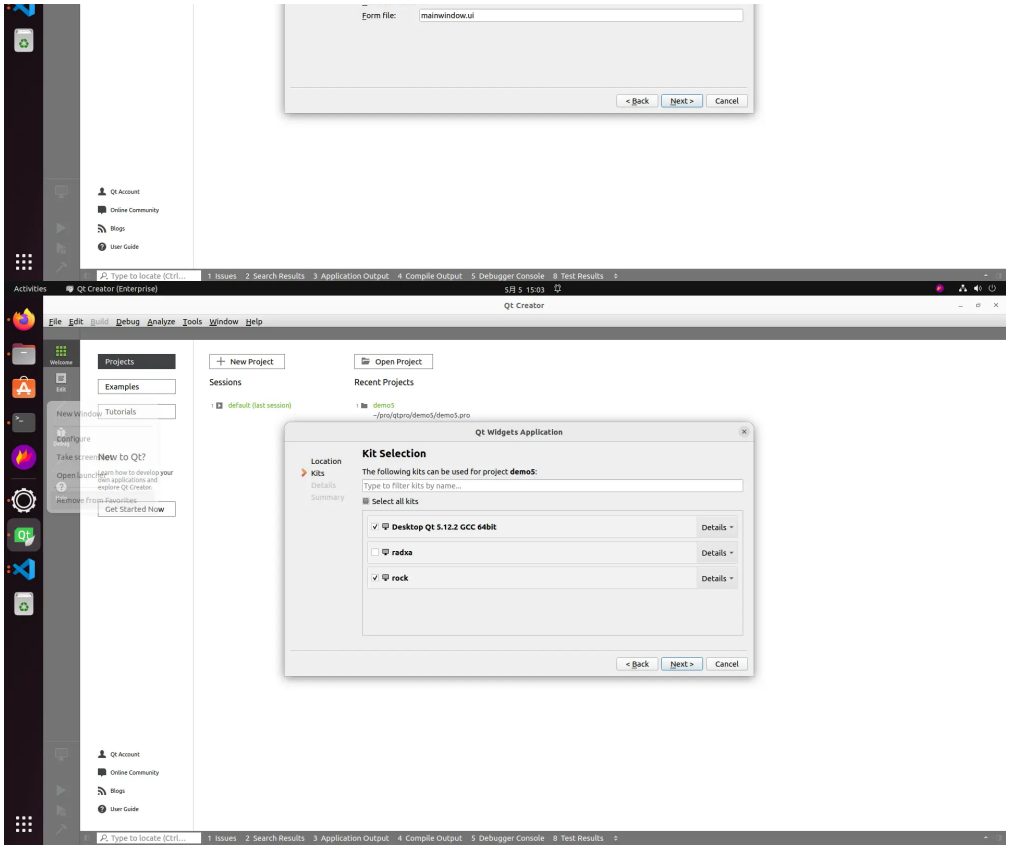
Then you can add gdb as you did with gcc/g++ above.

- Add Device

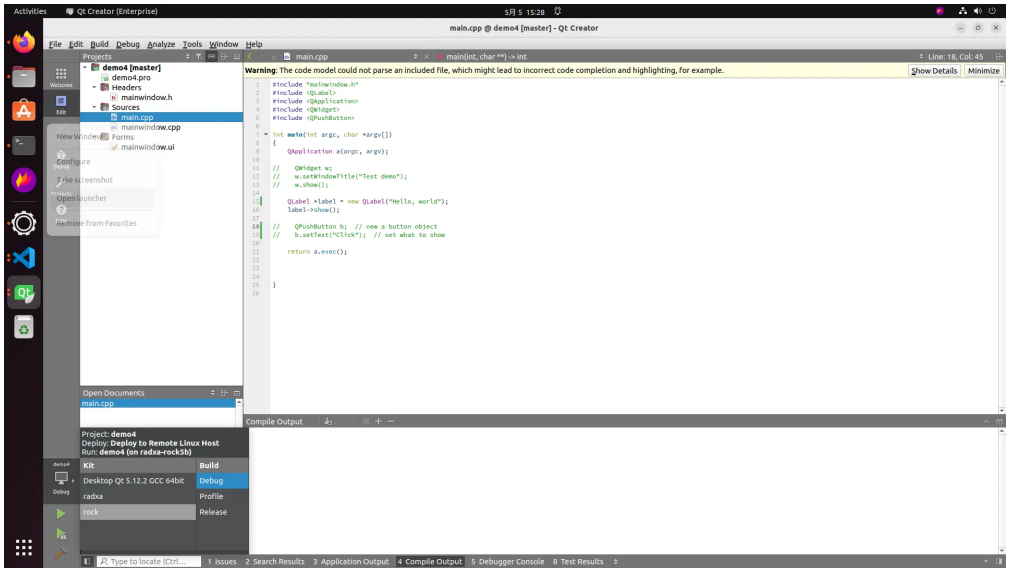


- New project





- sample code



- compiling

click Debugger->rock->Debug

- Copying the compiled project to the board

```
scp -r build-demo4-ext4-Debug radxa@192.168.2.xxx:~
```

On the development board

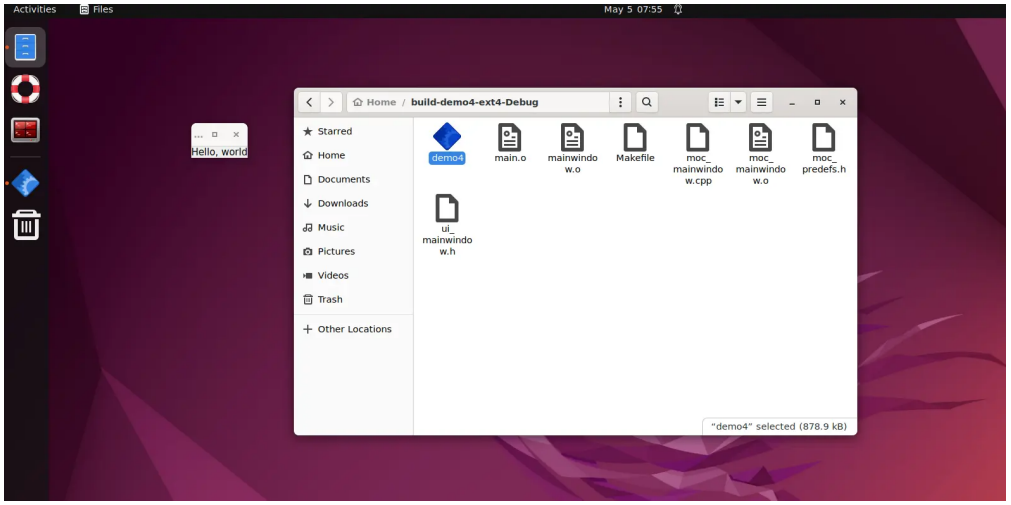
Env

```
sudo apt-get install libqt5gui5 -y
```

Execute the program

```
sudo chmod +x ./demo4
./demo4
```

Or click on the program directly on the desktop



Report Issue

Edit this page