GENIVI®

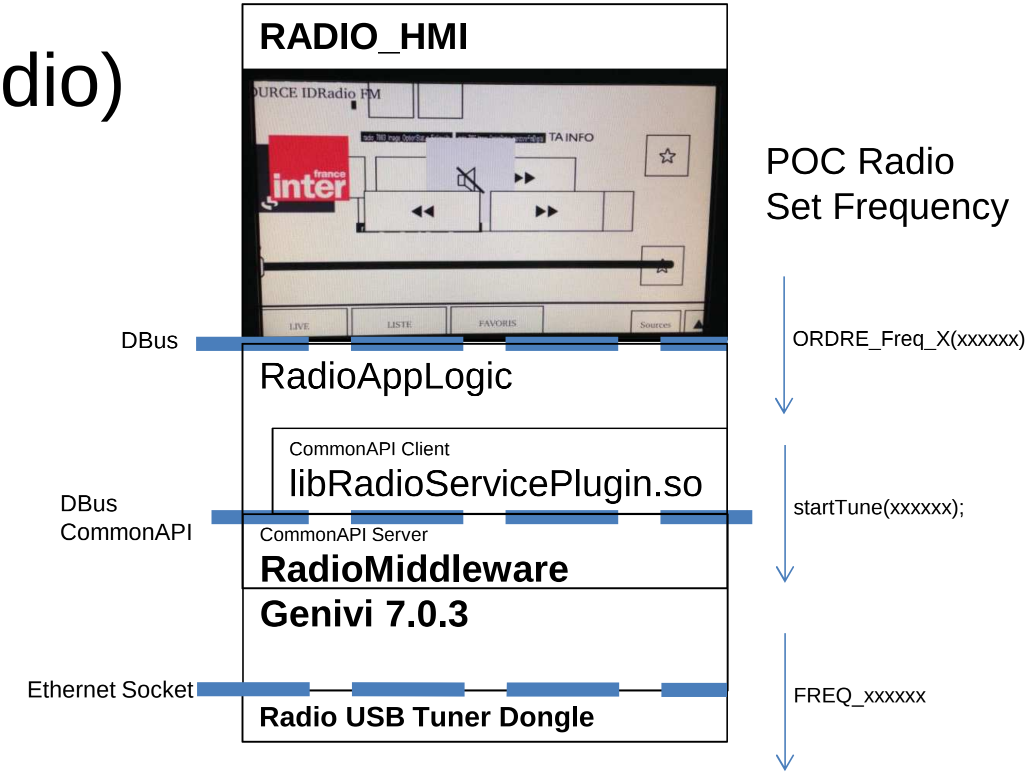
Genivi Demo Platform Hands-on Yocto™ Baseline Renesas™ & Intel™

Matthias Bloch
Multimedia Software Architect
PSA PEUGEOT CITROËN

Why we used GDP in PSA Peugeot Citroen ?

- Proof of concept (such as Radio)

  – Official Genivi IVI-Radio (2 binaries)
  – Update IVI-Radio to CommonAPI 2.1.6
  – Build in the Yocto™ Framework (Recipe Creation)
  – Convert client binary to library by the way to build



RADIO_HMI

POC Radio
Set Frequency

DBus

ORDRE_Freq_X(xxxxxx)

RadioAppLogic

CommonAPI Client
libRadioServicePlugin.so

DBus
CommonAPI

startTune(xxxxxx);

CommonAPI Server
RadioMiddleware

Genivi 7.0.3

Ethernet Socket

FREQ_xxxxxx

Radio USB Tuner Dongle

- Fuel Stop Advisor

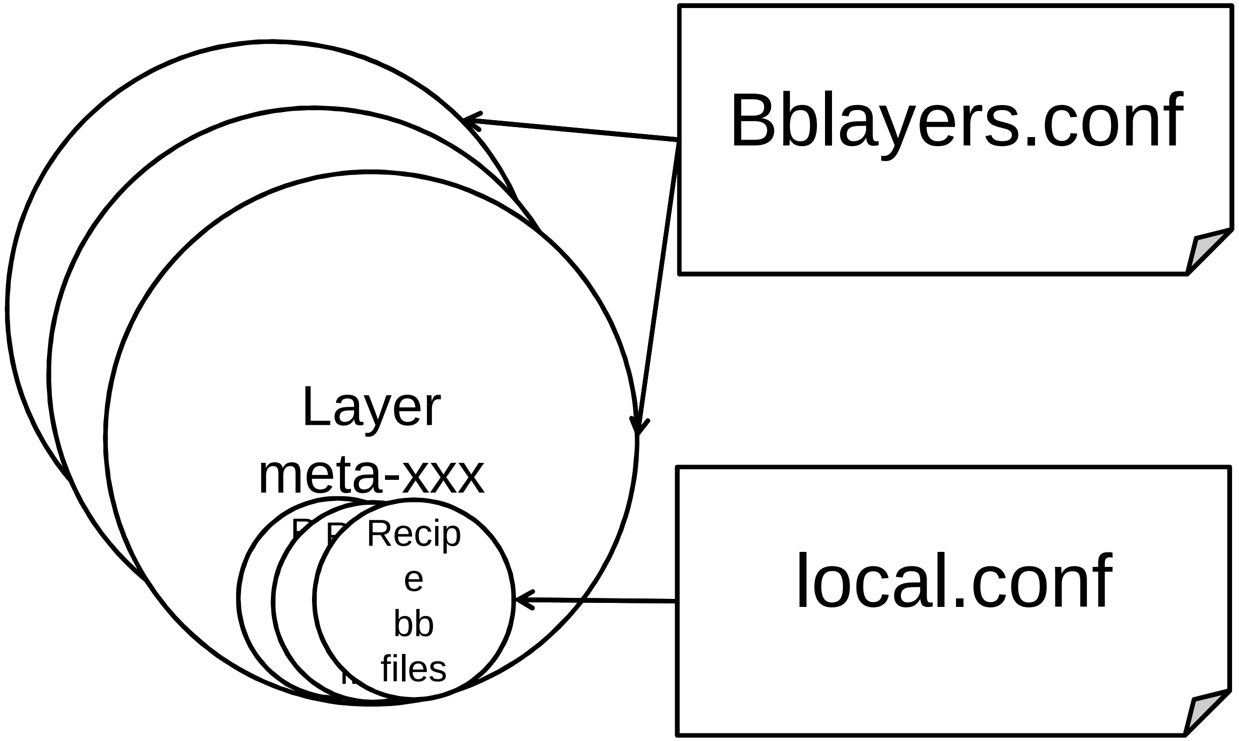
What we will cover

- Understand Yocto™ Build Concept
- Build Genivi Demo for target
- Understand SDK
- Build, Install and Configure SDK and third party tools.
- Build and Execute applications & module on target

Theory 😐

- Understand Yocto™ Build Concept

Important Concept



- What layers to look into for recipes

- Distribution: poky-ivi-systemd
- Machine: porter, intel-corei7-64, …
- Package Extra Configuration
- Add more Packages in the build image

What is a recipe

- Where to find to source code (git, svn, tar.gz, .c)
- What version
- Apply patches on it ?
- Special commands.

```
SUMMARY = “aaa"
DESCRIPTION = “bbb"
HOMEPGAE = “ccc"
LICENSE = "LGPL-2.1"
LIC_FILES_CHKSUM = "file://ddd;md5=597c8d49137513c98683e1d73158292f"

inherit cmake

PV = “hhh+git${SRCPV}"

DEPENDS = “eee fff ggg"

SRC_URI = “iii.jjjj.kkk"
SRC_URI += "file://lll.patch"
SRCREV = "955972390d16ca275159891cad29c2166217094d"

S = "${WORKDIR}/mmm"
do_install_append() {
   mv ${D}/usr/include/nnn/* ${D}/usr/include
}
INSANE_SKIP_${PN} = "dev-deps"
```

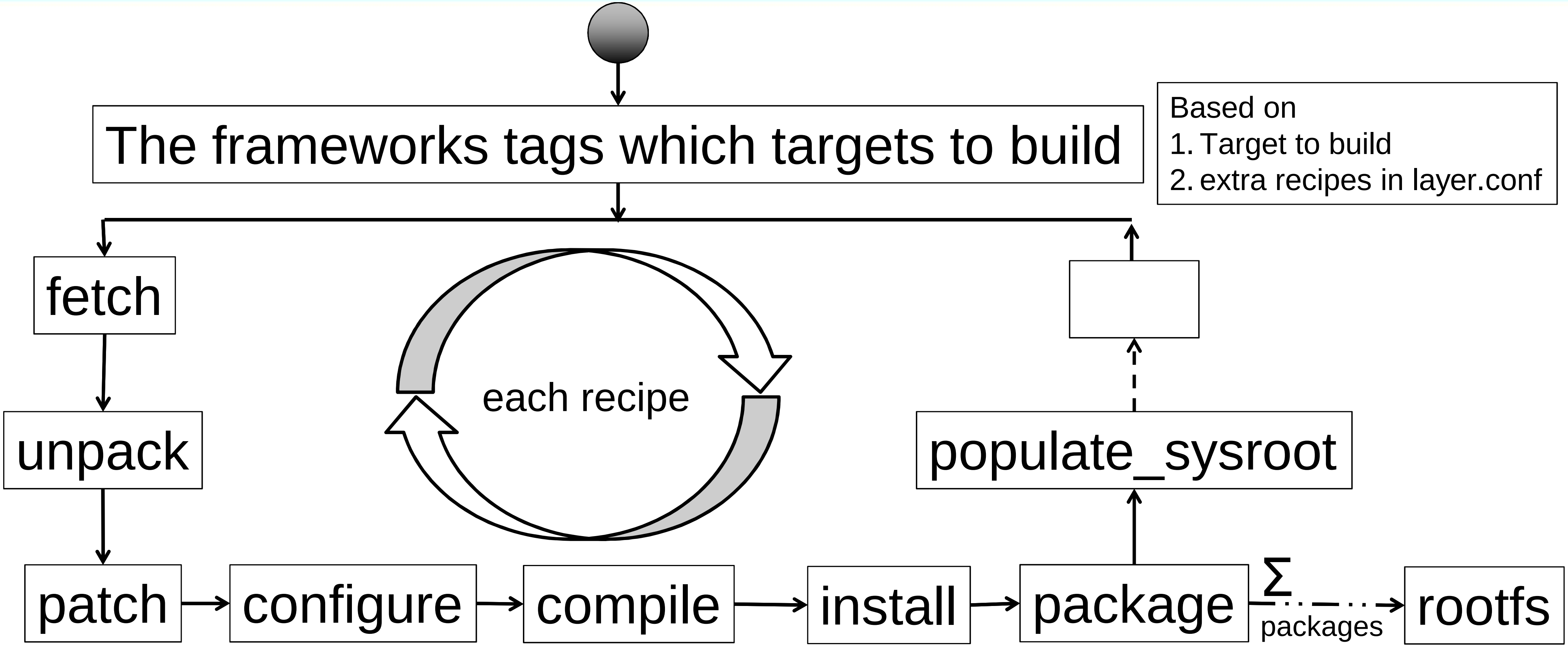
Build Command

- One python script controls everything: bitbake

- Build all target
  bitbake <name_of_the_target_to_build>
  (e.g. bitbake genivi-demo-platform)

- Build one special recipe (extra useful command)
  bitbake -c <command> -f <recipe_without_version_without_bb_extension>

What will happen during the build process



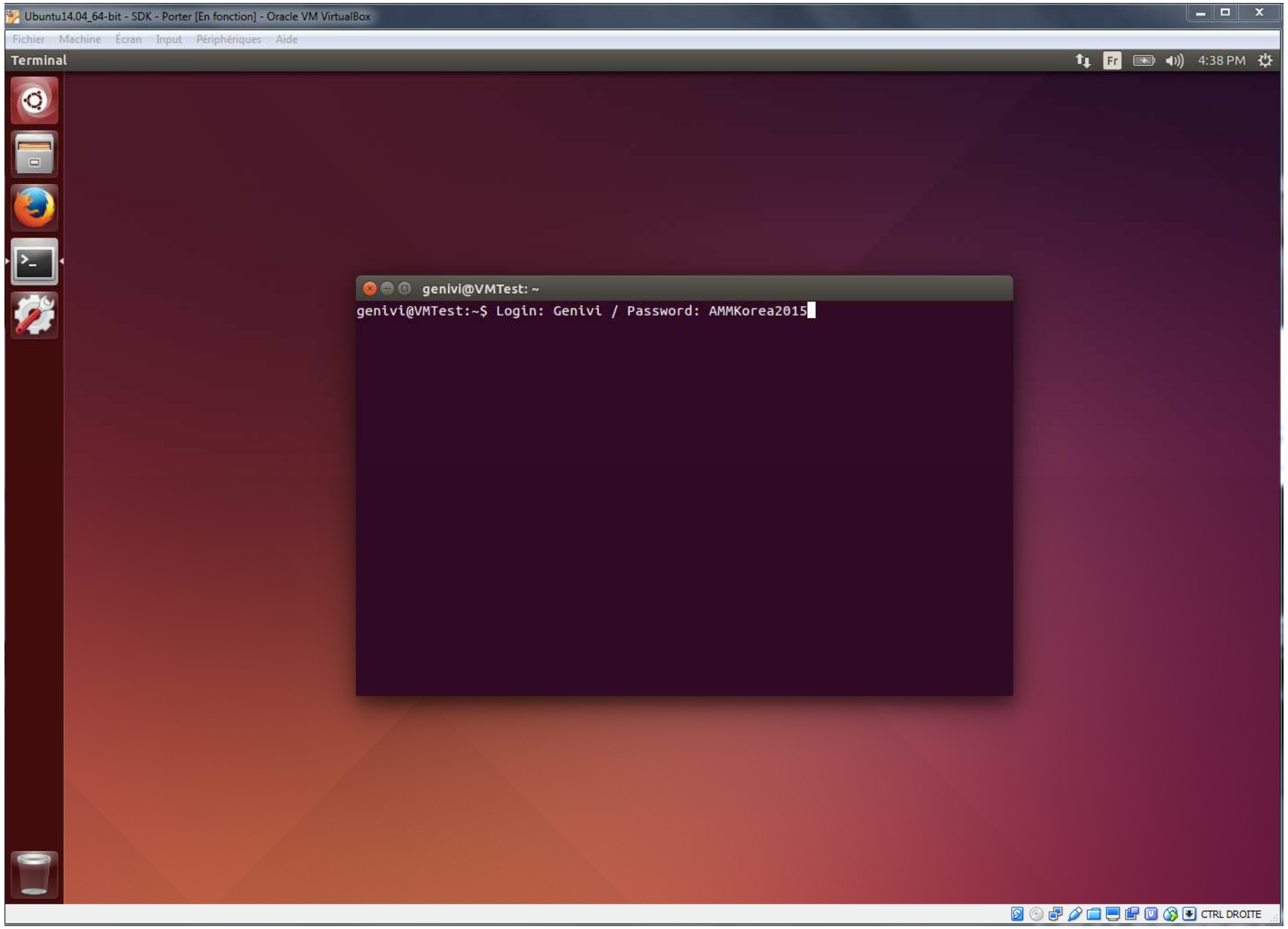
The frameworks tags which targets to build

Based on
1. Target to build
2. extra recipes in layer.conf

Hands on ☺

- Install the Virtual Machine files given

Virtual Machine



```
genivi@VMTest:~$ Login: Genivi / Password: AMMKorea2015
```

Hands on ☺

- Install the Virtual Machine files given
- How to build Image target

How to build target image

- The main pages are http://wiki.projects.genivi.org/index.php/GENIVI_Demo_Platform#Releases

```
sudo apt-get install gawk wget git-core diffstat unzip texinfo gcc-multilib build-essential chrpath socat libsdl1.2-dev xterm
mkdir <root_build_dir>/GDP
cd <root_build_dir>/GDP
git clone --recursive http://git.projects.genivi.org/genivi-demo-platform.git -b <target>
cd <root_build_dir>/GDP/genivi-demo-platform
source init.sh
bitbake genivi-demo-platform
```

- At the end of the build, 100GB will be allocated in the VM

Theory 😐

- Understand Yocto Build Concept
- SDK, SDK, why SDK ?

SDK, SDK, why SDK ?

- Deliver only useful (condensed) data to contributors
- Contributors may don’t know anything about Yocto™
- Known environment Eclipse™ and/or QTCreator™ for contributors to develop applications.

- WARNING Yocto main build modified => new SDK build

Hands on ☺

- Install the Virtual Machine files given
- How to build Image target
- How to build SDK

How to build SDK

- The main page is
  http://wiki.projects.genivi.org/index.php/Intrepid_-_Yocto_GDP_SDK_build_and_setup

```
cd <root_build_dir>/GDP/genivi-demo-platform
source init.sh
bitbake genivi-demo-platform-sdk -c populate_sdk
```

- At the end of the build, 2GB for the SDK in the VM

Hands on ☺

- Install the Virtual Machine files given
- How to build Image target
- How to build SDK
- How to install SDK

How to install SDK

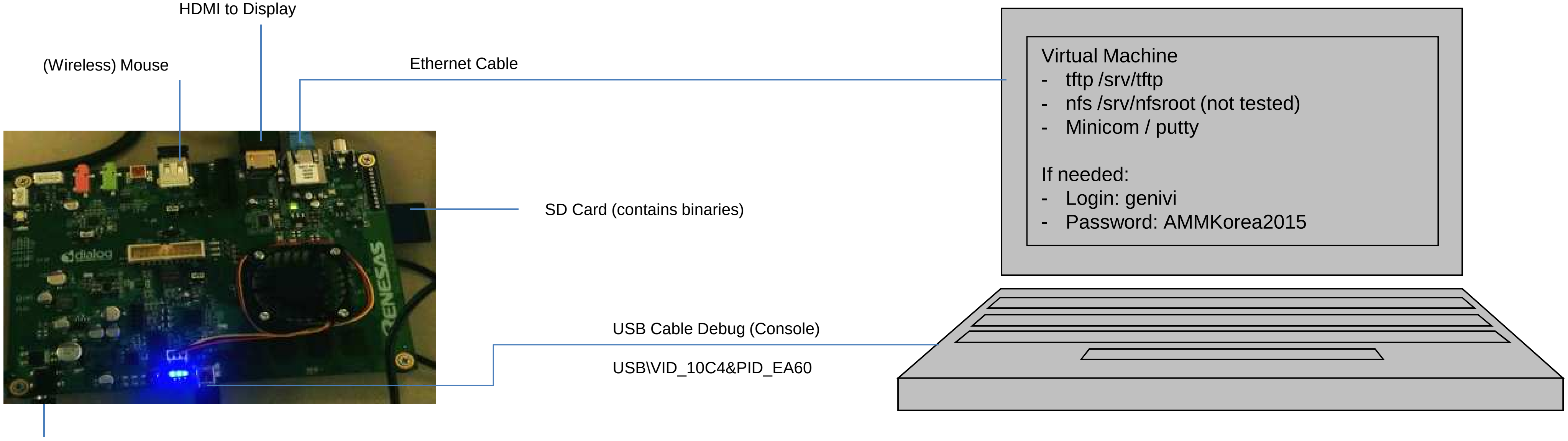
- The main page is http://wiki.projects.genivi.org/index.php/Intrepid_-_Yocto_GDP_SDK_build_and_setup

```
<sdk_build_dir>/oecore-x86_64-*.sh -d ./<sdk_dir>
You are about to install the SDK to “<sdk_dir>". Proceed[Y/n]?
Extracting SDK...done
Setting it up...done
SDK has been successfully set up and is ready to be used.
```

Hands on ☺

- Install the Virtual Machine files given
- How to build Image target
- How to build SDK
- How to install SDK
- Setup the targets

Setup Porter



HDMI to Display

(Wireless) Mouse

Ethernet Cable

SD Card (contains binaries)

USB Cable Debug (Console)

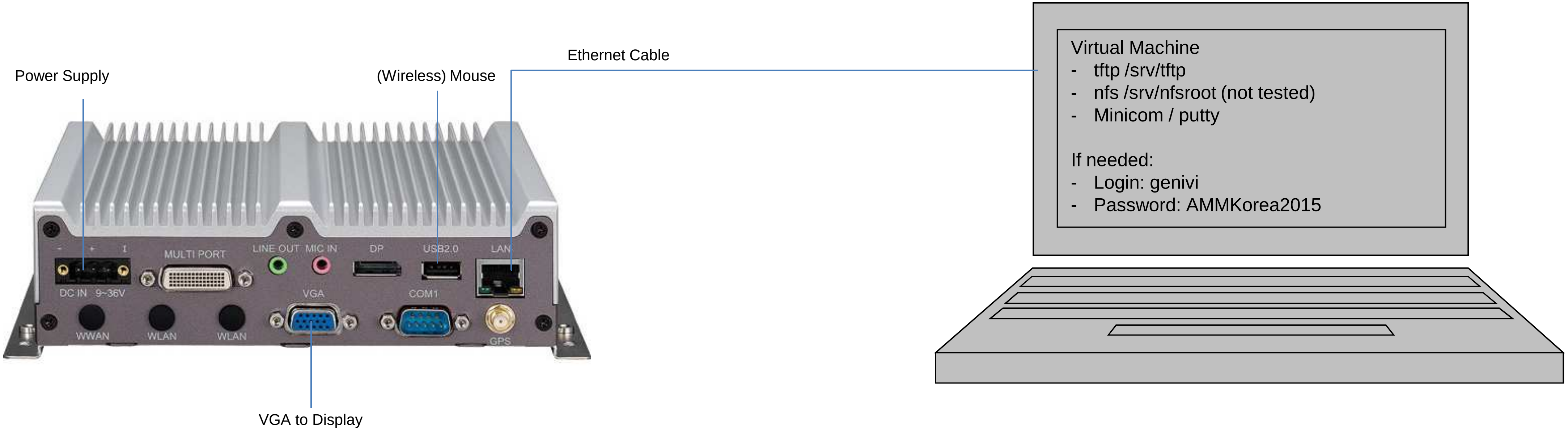
USB\VID_10C4&PID_EA60

Power Supply

Virtual Machine
- tftp /srv/tftp
- nfs /srv/nfsroot (not tested)
- Minicom / putty

If needed:
- Login: genivi
- Password: AMMKorea2015

Setup VTC-1010



Power Supply

(Wireless) Mouse

Ethernet Cable

VGA to Display

Virtual Machine
- tftp /srv/tftp
- nfs /srv/nfsroot (not tested)
- Minicom / putty

If needed:
- Login: genivi
- Password: AMMKorea2015

Hands on ☺

- Install the Virtual Machine files given
- How to build Image target
- How to build SDK
- How to install SDK
- Setup the targets
- How to install Eclipse™

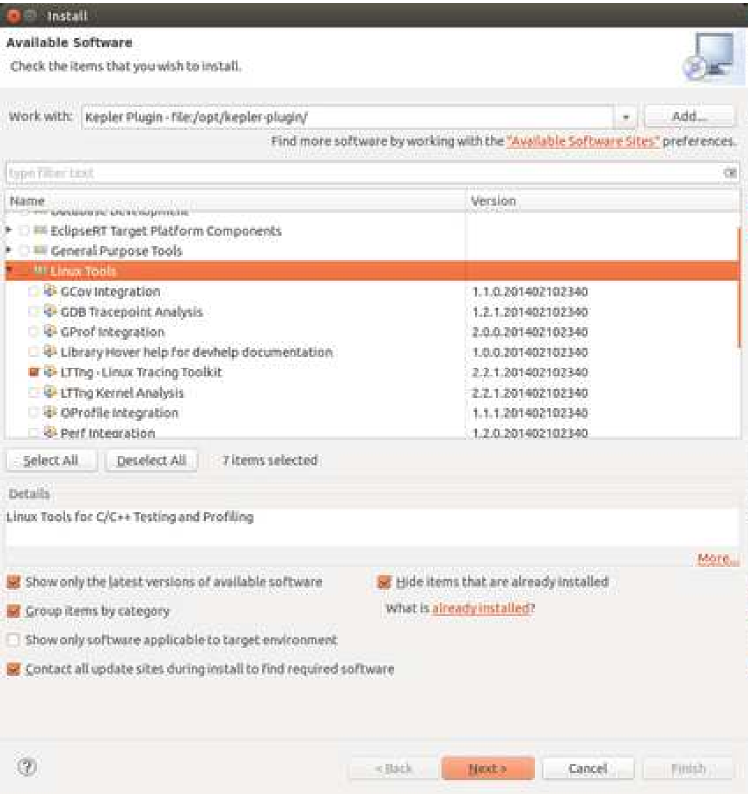
How to install Eclipse™

- The main page is
  http://wiki.projects.genivi.org/index.php/Intrepid_-_Yocto_GDP_SDK_build_and_setup

```
sudo apt-get install openjdk-7-jdk
tar -xvzf <dwn_dir>/eclipse-standard-kepler-SR2-linux-gtk-x86_64.tar.gz
source <sdk_dir>/environment-setup-*-poky-linux
<eclipse_dir>/eclipse &
```

- Help Menu ⟶Install New Software

How to install Eclipse™



Available Software

Check the items that you wish to install.

Work with: Kepler Plugin - file:/opt/kepler-plugin/

Find more software by working with the "Available Software Sites" preferences.

type filter text

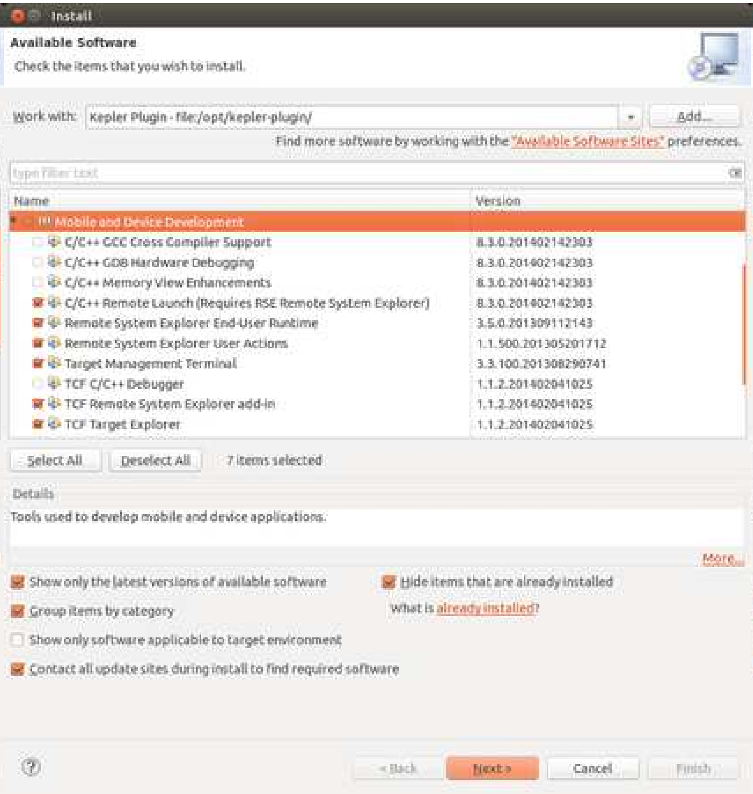
Name	Version
▸ □ EclipseRT Target Platform Components	
▸ □ General Purpose Tools	
▾ Linux Tools	
□ GCov Integration	1.1.0.201402102340
□ GDB Tracepoint Analysis	1.2.1.201402102340
□ GProf Integration	2.0.0.201402102340
□ Library Hover help for devhelp documentation	1.0.0.201402102340
☑ LTTng - Linux Tracing Toolkit	2.2.1.201402102340
□ LTTng Kernel Analysis	2.2.1.201402102340
□ OProfile Integration	1.1.1.201402102340
□ Perf Integration	1.2.0.201402102340

7 items selected

Details

Linux Tools for C/C++ Testing and Profiling

☑ Show only the latest versions of available software
☑ Hide items that are already installed
☑ Group items by category
What is already installed?
□ Show only software applicable to target environment
☑ Contact all update sites during install to find required software



Available Software

Check the items that you wish to install.

Work with: Kepler Plugin - file:/opt/kepler-plugin/

Find more software by working with the "Available Software Sites" preferences.

type filter text

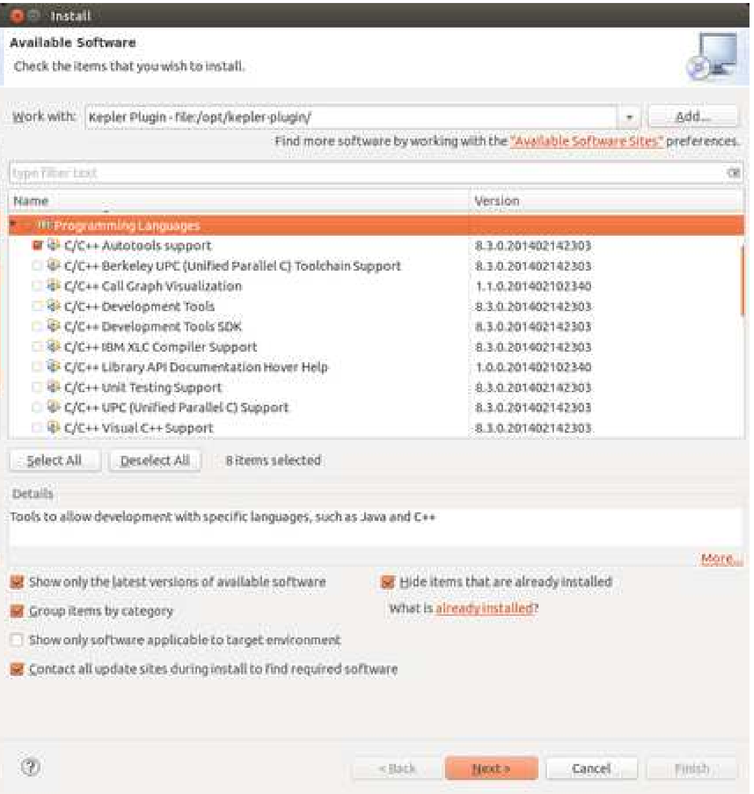
Name	Version
▾ Mobile and Device Development	
□ C/C++ GCC Cross Compiler Support	8.3.0.201402142303
□ C/C++ GDB Hardware Debugging	8.3.0.201402142303
□ C/C++ Memory View Enhancements	8.3.0.201402142303
☑ C/C++ Remote Launch (Requires RSE Remote System Explorer)	8.3.0.201402142303
☑ Remote System Explorer End-User Runtime	3.5.0.201309112143
☑ Remote System Explorer User Actions	1.1.500.201305201712
☑ Target Management Terminal	3.3.100.201308290741
□ TCF C/C++ Debugger	1.1.2.201402041025
☑ TCF Remote System Explorer add-in	1.1.2.201402041025
☑ TCF Target Explorer	1.1.2.201402041025

7 items selected

Details

Tools used to develop mobile and device applications.

☑ Show only the latest versions of available software
☑ Hide items that are already installed
☑ Group items by category
What is already installed?
□ Show only software applicable to target environment
☑ Contact all update sites during install to find required software



Available Software

Check the items that you wish to install.

Work with: Kepler Plugin - file:/opt/kepler-plugin/

Find more software by working with the "Available Software Sites" preferences.

type filter text

Name	Version
▾ Programming Languages	
☑ C/C++ Autotools support	8.3.0.201402142303
□ C/C++ Berkeley UPC (Unified Parallel C) Toolchain Support	8.3.0.201402142303
□ C/C++ Call Graph Visualization	1.1.0.201402102340
□ C/C++ Development Tools	8.3.0.201402142303
□ C/C++ Development Tools SDK	8.3.0.201402142303
□ C/C++ IBM XLC Compiler Support	8.3.0.201402142303
□ C/C++ Library API Documentation Hover Help	1.0.0.201402102340
□ C/C++ Unit Testing Support	8.3.0.201402142303
□ C/C++ UPC (Unified Parallel C) Support	8.3.0.201402142303
□ C/C++ Visual C++ Support	8.3.0.201402142303

8 items selected

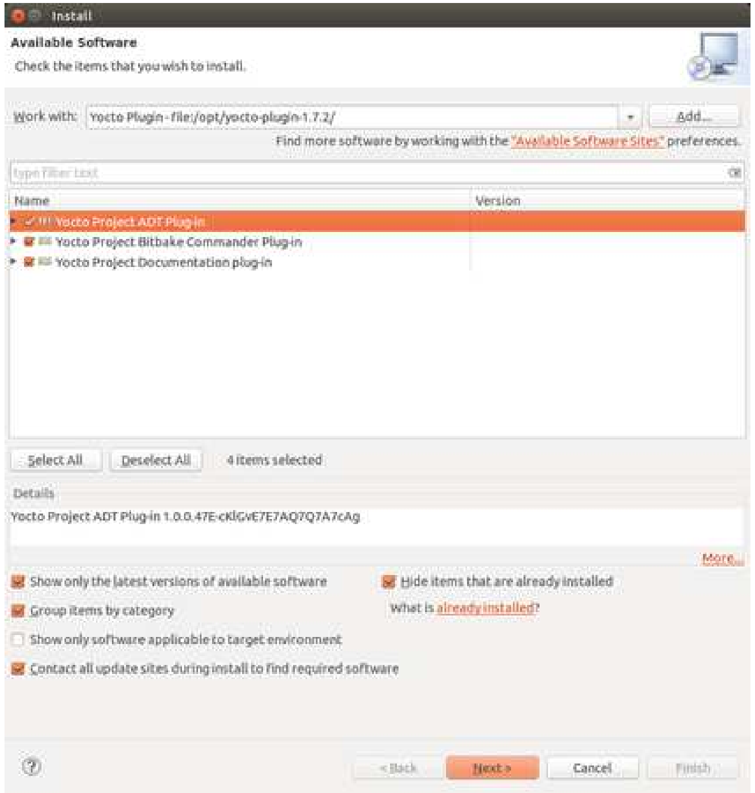
Details

Tools to allow development with specific languages, such as Java and C++

☑ Show only the latest versions of available software
☑ Hide items that are already installed
☑ Group items by category
What is already installed?
□ Show only software applicable to target environment
☑ Contact all update sites during install to find required software

How to install Eclipse™

- Yocto plug-in: http://downloads.yoctoproject.org/releases/eclipse-plugin/1.6.1/kepler/



Available Software

Check the items that you wish to install.

Work with: Yocto Plugin - file:/opt/yocto-plugin-1.7.2/

Find more software by working with the "Available Software Sites" preferences.

Name	Version
☑ Yocto Project ADT Plug-in	
☑ Yocto Project Bitbake Commander Plug-in	
☑ Yocto Project Documentation plug-in	

4 items selected

Details

Yocto Project ADT Plug-in 1.0.0.47E-cKlGvE7E7AQ7Q7A7cAg

☑ Show only the latest versions of available software

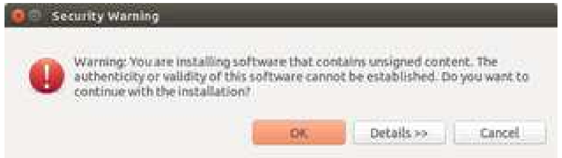
☑ Hide items that are already installed

What is already installed?

☑ Group items by category

☐ Show only software applicable to target environment

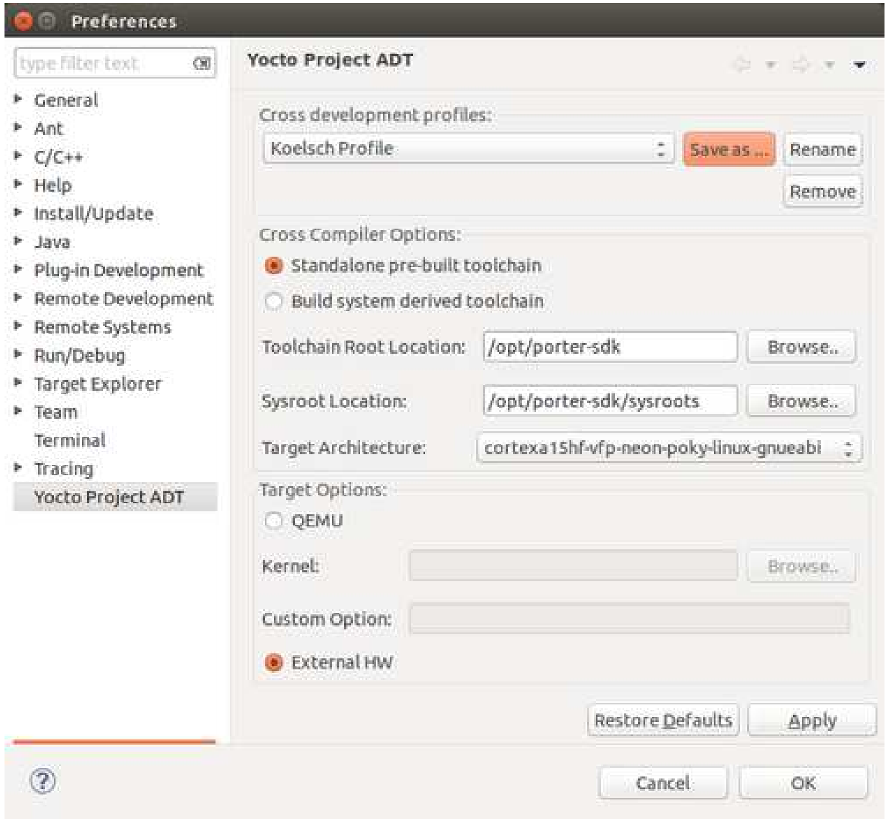
☑ Contact all update sites during install to find required software



Warning: You are installing software that contains unsigned content. The authenticity or validity of this software cannot be established. Do you want to continue with the installation?

How to install Eclipse™

- Setup for the target
  Window ⟶ Preferences



Yocto Project ADT

Cross development profiles: Koelsch Profile

Cross Compiler Options:
● Standalone pre-built toolchain
○ Build system derived toolchain

Toolchain Root Location: /opt/porter-sdk

Sysroot Location: /opt/porter-sdk/sysroots

Target Architecture: cortexa15hf-vfp-neon-poky-linux-gnueabi

Target Options:
○ QEMU
Kernel:
Custom Option:
● External HW

Hands on ☺

- Install the Virtual Machine files given
- How to build Image target
- How to build SDK
- How to install SDK
- How to install Eclipse™
- Setup the targets
- How to install QtCreator™

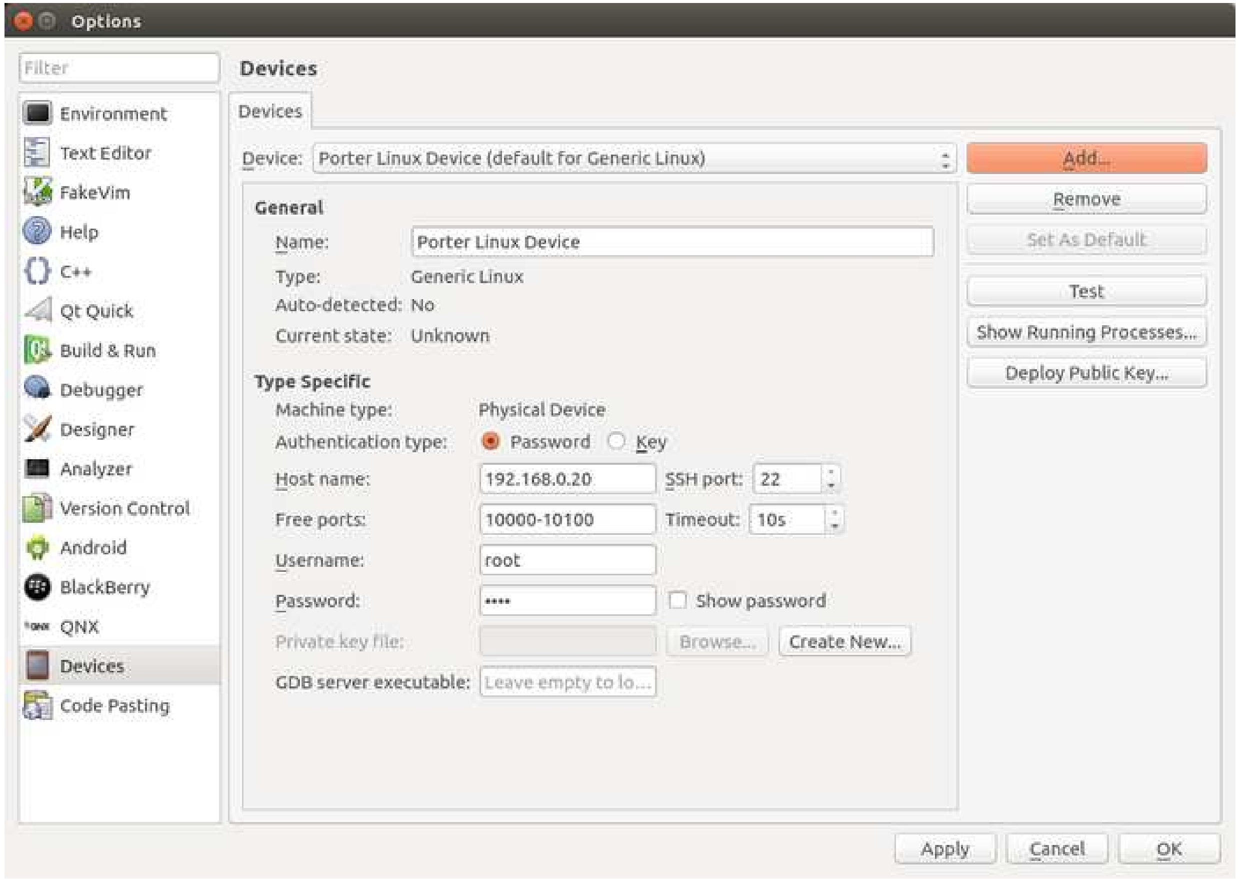
How to install QtCreator™

- The main page is http://wiki.projects.genivi.org/index.php/Intrepid_-_Yocto_GDP_SDK_build_and_setup

```
<dwnl_dir>/ qt-creator-opensource-linux-x86_64-3.2.2.run
source <sdk_dir>/environment-setup-*-poky-linux
export OE_QMAKE_CC="$CC"
export OE_QMAKE_CXX="$CXX"
export OE_QMAKE_CFLAGS="$CFLAGS"
export OE_QMAKE_CXXFLAGS="$CXXFLAGS"
export OE_QMAKE_LINK="$CXX"
export OE_QMAKE_LDFLAGS="$LDFLAGS"
export OE_QMAKE_AR="$AR"
export OE_QMAKE_STRIP="$STRIP"
<qtcreator_dir>/bin/qtcreator &
```

- Tools Menu⟶Option

How to install QtCreator™



Devices

Device: Porter Linux Device (default for Generic Linux)

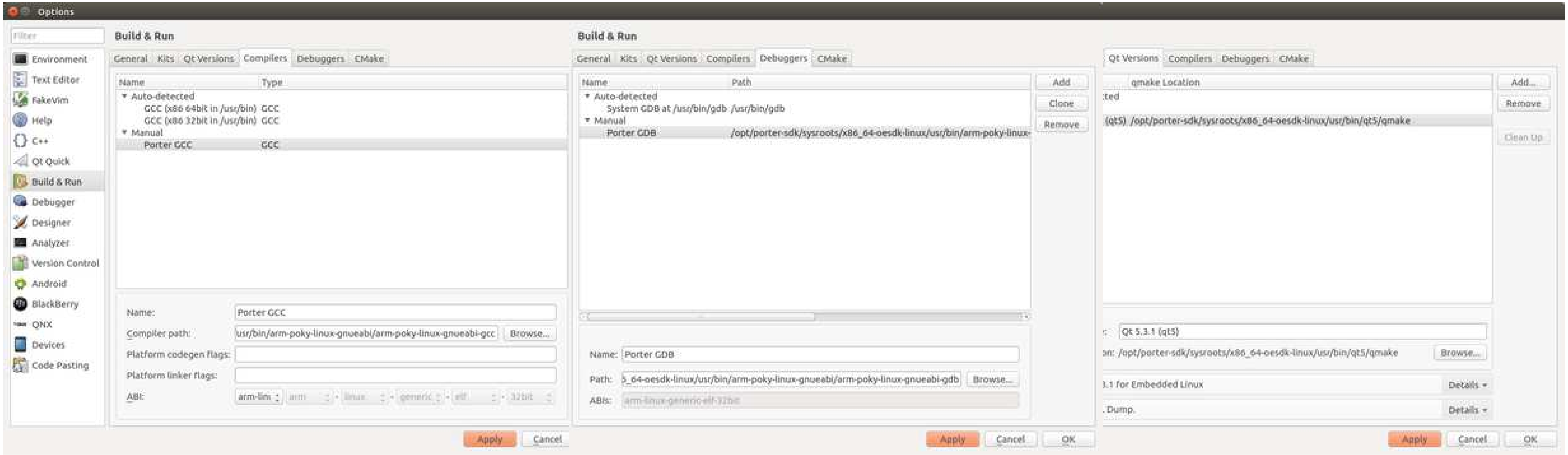
General

Name: Porter Linux Device
Type: Generic Linux
Auto-detected: No
Current state: Unknown

Type Specific

Machine type: Physical Device
Authentication type: ● Password ○ Key
Host name: 192.168.0.20 SSH port: 22
Free ports: 10000-10100 Timeout: 10s
Username: root
Password: **** □ Show password
Private key file:
GDB server executable: Leave empty to lo...

How to install QtCreator™



Build & Run — Compilers

Name	Type
▾ Auto-detected	
GCC (x86 64bit in /usr/bin)	GCC
GCC (x86 32bit in /usr/bin)	GCC
▾ Manual	
Porter GCC	GCC

Name: Porter GCC

Compiler path: usr/bin/arm-poky-linux-gnueabi/arm-poky-linux-gnueabi-gcc

Platform codegen flags:

Platform linker flags:

ABI: arm-linı - arm - linux - generic - elf - 32bit

Build & Run — Debuggers

Name	Path
▾ Auto-detected	
System GDB at /usr/bin/gdb	/usr/bin/gdb
▾ Manual	
Porter GDB	/opt/porter-sdk/sysroots/x86_64-oesdk-linux/usr/bin/arm-poky-linux-

Name: Porter GDB

Path: 5_64-oesdk-linux/usr/bin/arm-poky-linux-gnueabi/arm-poky-linux-gnueabi-gdb

ABIs: arm-linux-generic-elf-32bit

Qt Versions

	qmake Location
:ted	
(qt5)	/opt/porter-sdk/sysroots/x86_64-oesdk-linux/usr/bin/qt5/qmake

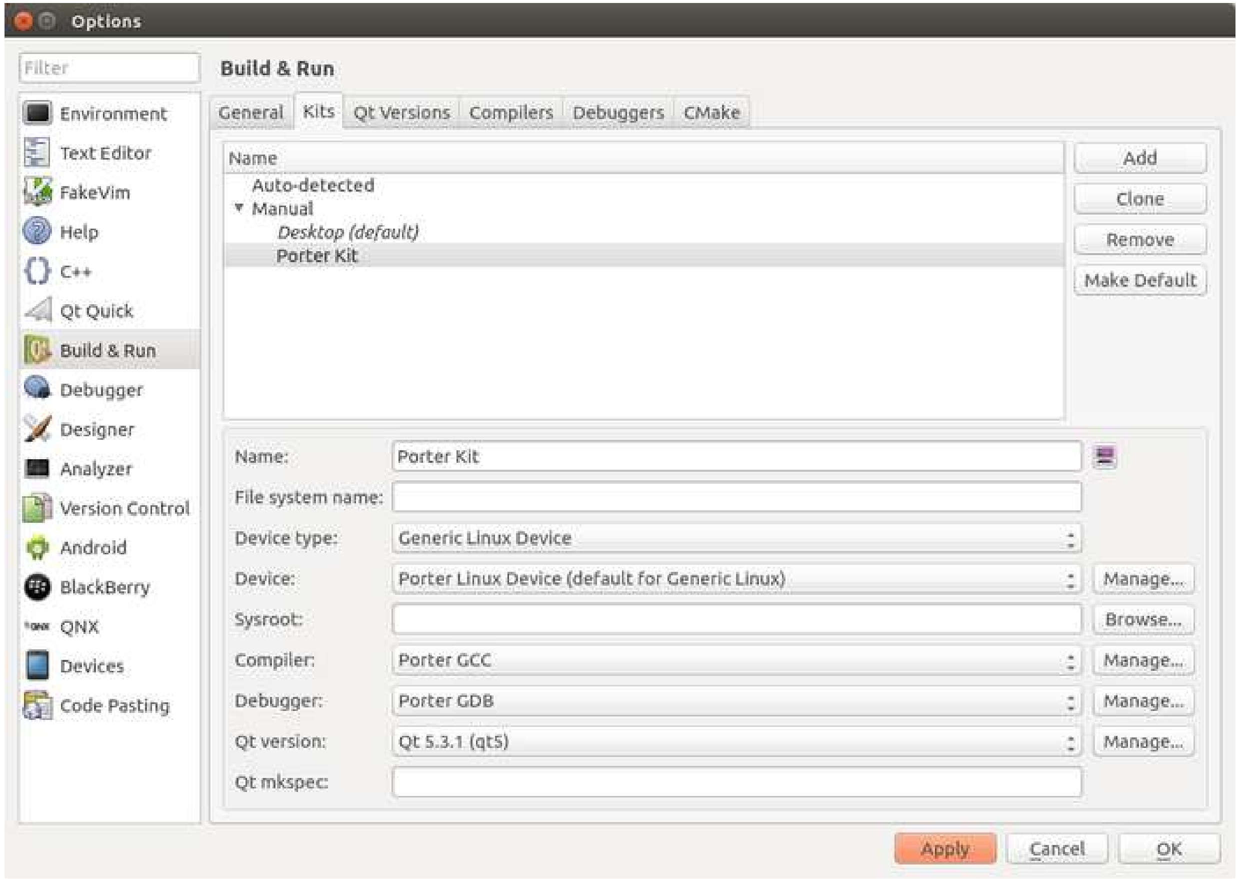
:: Qt 5.3.1 (qt5)

on: /opt/porter-sdk/sysroots/x86_64-oesdk-linux/usr/bin/qt5/qmake

3.1 for Embedded Linux

. Dump.

How to install QtCreator™



Build & Run

Name
- Auto-detected
- Manual
  - *Desktop (default)*
  - Porter Kit

Name: Porter Kit

File system name:

Device type: Generic Linux Device

Device: Porter Linux Device (default for Generic Linux)

Sysroot:

Compiler: Porter GCC

Debugger: Porter GDB

Qt version: Qt 5.3.1 (qt5)

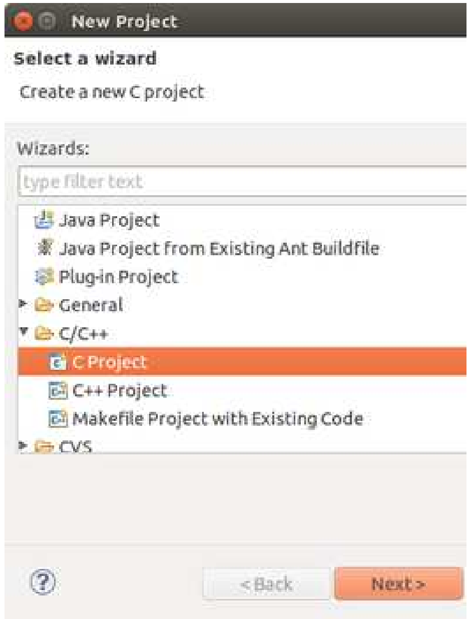
Qt mkspec:

Hands on ☺

- Install the Virtual Machine files given
- How to build Image target
- How to Build SDK
- How to install SDK
- Setup the targets
- How to install Eclipse™
- How to install QtCreator™
- Build HelloWorld applications and module
- Execute HelloWorld applications and module on target

Build HelloWorld application using Eclipse™

- The main pages are
  http://wiki.projects.genivi.org/index.php/Intrepid_-_Yocto_GDP_application_development_tutorial



New Project

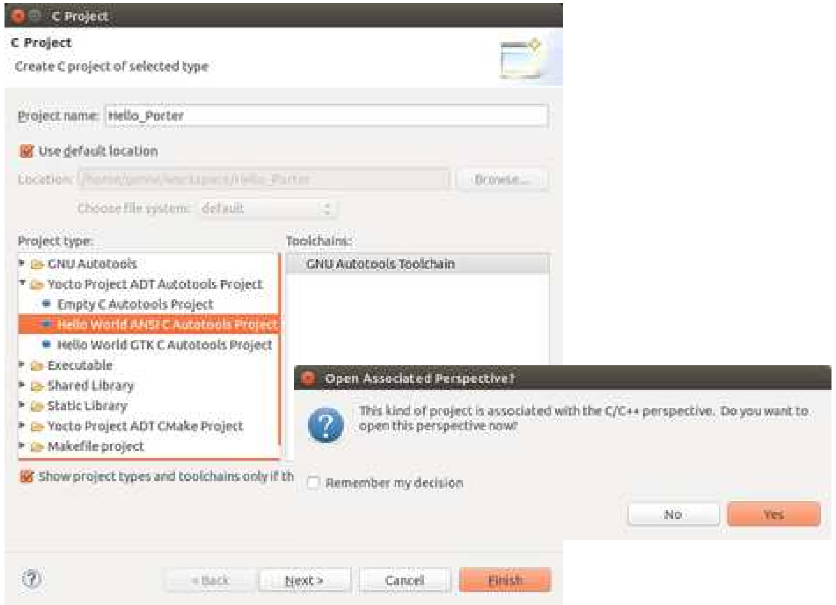
Select a wizard

Create a new C project

Wizards:

type filter text

Java Project
Java Project from Existing Ant Buildfile
Plug-in Project
General
C/C++
C Project
C++ Project
Makefile Project with Existing Code
CVS
< Back
Next >



C Project

C Project

Create C project of selected type

Project name: Hello_Porter

☑ Use default location

Location: /home/genivi/workspace/Hello_Porter

Choose file system: default

Project type:

GNU Autotools
Yocto Project ADT Autotools Project
Empty C Autotools Project
Hello World ANSI C Autotools Project
Hello World GTK C Autotools Project
Executable
Shared Library
Static Library
Yocto Project ADT CMake Project
Makefile project

Toolchains:

GNU Autotools Toolchain

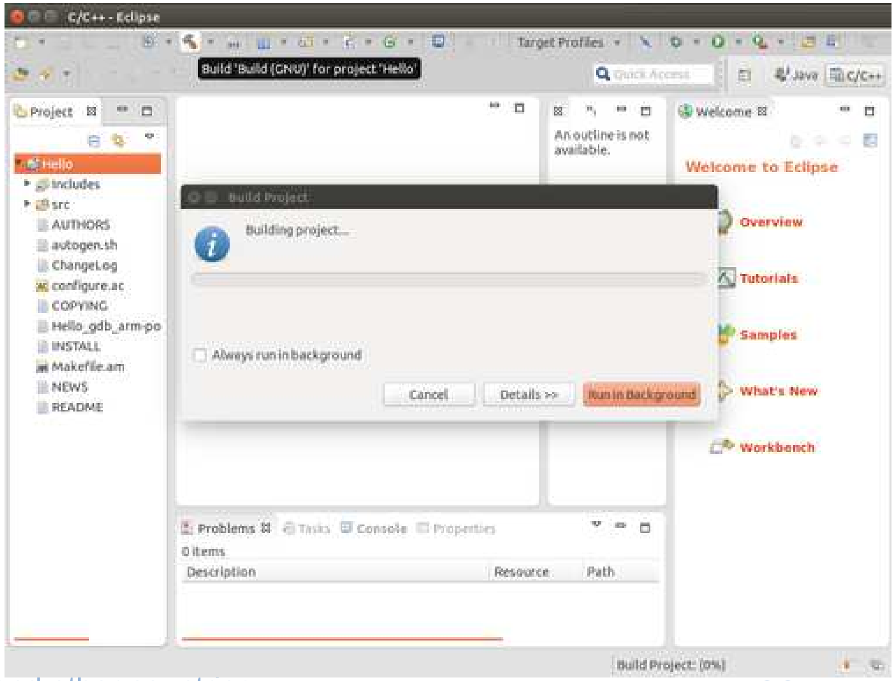
☑ Show project types and toolchains only if th

Browse...
< Back
Next >
Cancel
Finish

Open Associated Perspective?

This kind of project is associated with the C/C++ perspective. Do you want to open this perspective now?

□ Remember my decision



C/C++ - Eclipse

Build 'Build (GNU)' for project 'Hello'

An outline is not available.

Build Project

Building project...

□ Always run in background

0 items

Description	Resource	Path

Build Project: (0%)

Execute HelloWorld application using Eclipse™

- Then, To check module

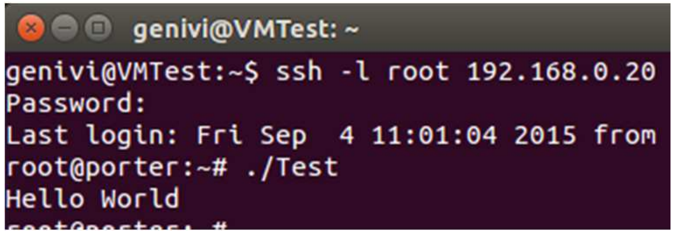
  file Test # check that everything went well
  Test: ELF 32-bit LSB executable, **ARM**, EABI5 version 1 (SYSV), dynamically linked (uses shared libs), for GNU/Linux 2.6.32, BuildID[sha1]=40ee7cdd514b8ec830680e6004bbcd2824469c95, not stripped

- For deploy, copy (scp) to whatever target directory.

  scp <source_file> <login>@<ip>:<dst_dir>/<dst_file>

- Execute like every Linux binary

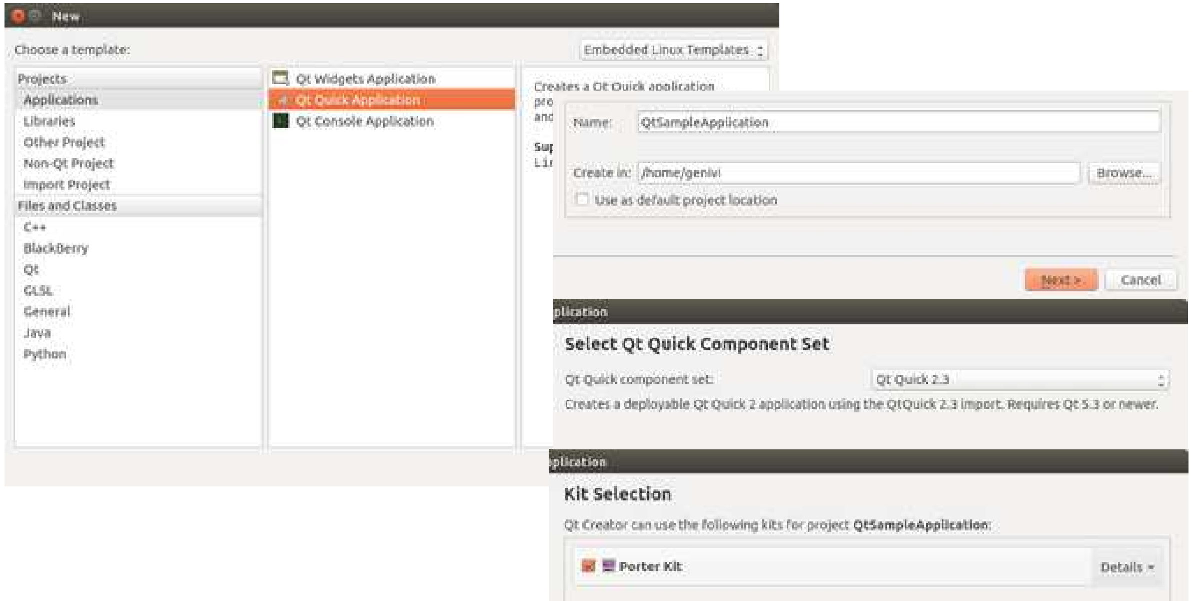
  <dst_dir>/<dst_file>



```
genivi@VMTest:~$ ssh -l root 192.168.0.20
Password:
Last login: Fri Sep  4 11:01:04 2015 from
root@porter:~# ./Test
Hello World
root@porter:~#
```

Build HelloWorld using QTCreator™

- The main pages are
  http://wiki.projects.genivi.org/index.php/Intrepid_-_Yocto_GDP_application_development_tutorial



Choose a template: Embedded Linux Templates

Creates a Qt Quick application

Name: QtSampleApplication

Create in: /home/genivi

□ Use as default project location

Select Qt Quick Component Set

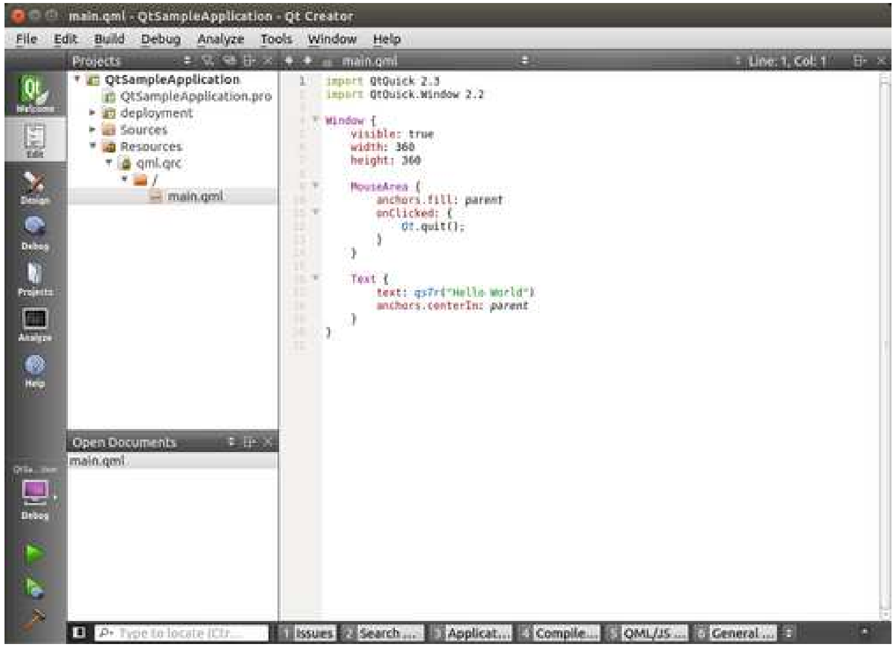
Qt Quick component set: Qt Quick 2.3

Creates a deployable Qt Quick 2 application using the QtQuick 2.3 import. Requires Qt 5.3 or newer.

Kit Selection

Qt Creator can use the following kits for project **QtSampleApplication**:

☑ Porter Kit



```
import QtQuick 2.3
import QtQuick.Window 2.2

Window {
    visible: true
    width: 360
    height: 360

    MouseArea {
        anchors.fill: parent
        onClicked: {
            Qt.quit();
        }
    }

    Text {
        text: qsTr("Hello World")
        anchors.centerIn: parent
    }
}
```

Execute HelloWorld using QTCreator™

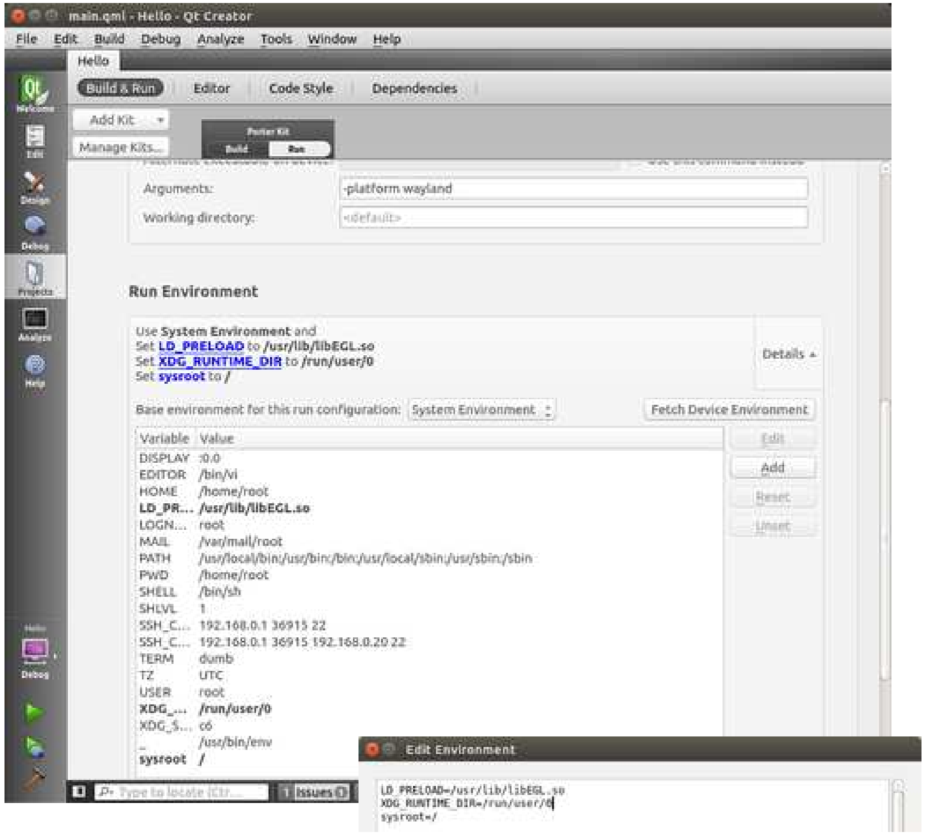
- For QTCreator™ deploy, sftp should be present in the board.

```
scp sftp <login>@<ip>:/usr/bin/sftp
```

- In /etc/ssh/sshd_config, Change to

```
Subsystem      sftp     internal-sftp
```

Execute HelloWorld using QTCreator™



Arguments: -platform wayland

Working directory: <default>

Run Environment

Use **System Environment** and
Set **LD_PRELOAD** to **/usr/lib/libEGL.so**
Set **XDG_RUNTIME_DIR** to **/run/user/0**
Set **sysroot** to **/**

Details ▴

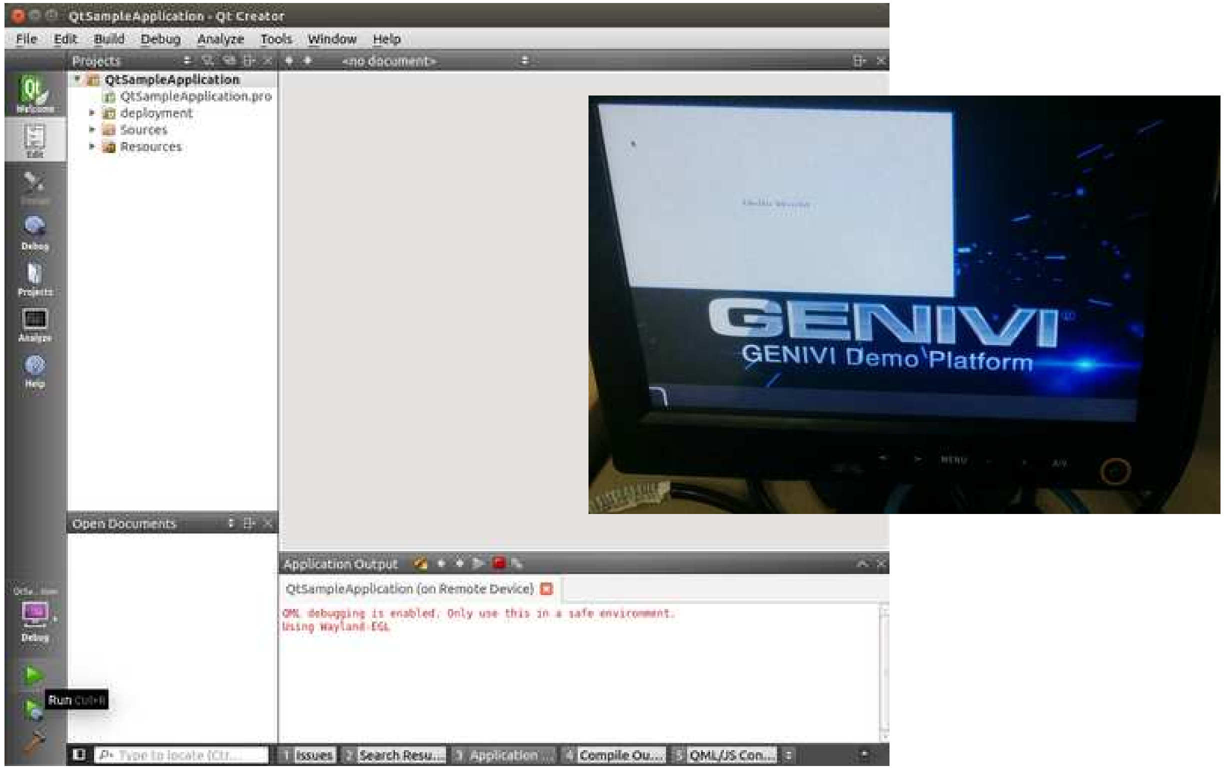
Base environment for this run configuration: System Environment

Fetch Device Environment

Variable	Value
DISPLAY	:0.0
EDITOR	/bin/vi
HOME	/home/root
LD_PR...	**/usr/lib/libEGL.so**
LOGN...	root
MAIL	/var/mail/root
PATH	/usr/local/bin:/usr/bin:/bin:/usr/local/sbin:/usr/sbin:/sbin
PWD	/home/root
SHELL	/bin/sh
SHLVL	1
SSH_C...	192.168.0.1 36915 22
SSH_C...	192.168.0.1 36915 192.168.0.20 22
TERM	dumb
TZ	UTC
USER	root
XDG_...	**/run/user/0**
XDG_S...	c6
_	/usr/bin/env
sysroot	**/**

Edit Environment

```
LD_PRELOAD=/usr/lib/libEGL.so
XDG_RUNTIME_DIR=/run/user/0
sysroot=/
```



Application Output

QtSampleApplication (on Remote Device)

```
QML debugging is enabled. Only use this in a safe environment.
Using Wayland-EGL
```

GENIVI is a registered trademark of the GENIVI Alliance in the USA and other countries

Build HelloWorld module

- The main pages are
  http://wiki.projects.genivi.org/index.php/Intrepid_-_Yocto_GDP_platform_development_tutorial

- To build module, need to build kernel scripts

```
source <sdk_dir>/environment-setup-*-poky-linux
cd <sdk_dir>/sysroots/cortexa15hf-vfp-neon-poky-linux-gnueabi/usr/src/kernel/
make scripts
```

- Then, to build module

```
source <sdk_dir>/environment-setup-*-poky-linux
cd <module_dir>/hello_mod/files
export KERNEL_SRC=<sdk_dir>/sysroots/cortexa15hf-vfp-neon-poky-linux-gnueabi/usr/src/kernel
make -C $KERNEL_SRC M=`pwd`
```

Execute HelloWorld module

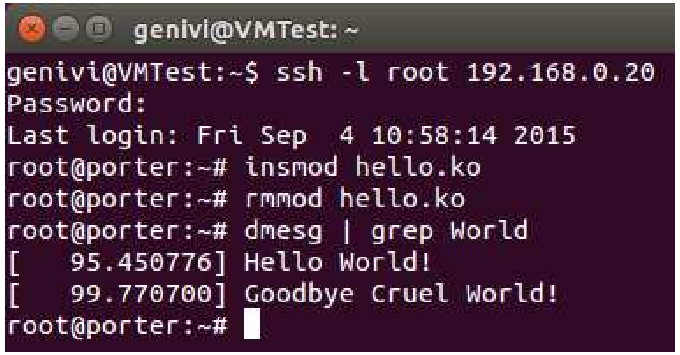
- Then, To check module

  file hello.ko # check that everything went well
  hello.ko: ELF 32-bit LSB relocatable, **ARM**, EABI5 version 1 (SYSV),
  buildID[sha1]=3f90ecff4b57a02770a705f654653bc0e2890251, not stripped

- Finally, put (scp) in the target and insmod/rmmod

  scp hello.ko <login>@<ip>:<dst_dir>/ hello.ko

  On target: insmod Hello.ko
                      rmmod Hello.ko



```
genivi@VMTest:~$ ssh -l root 192.168.0.20
Password:
Last login: Fri Sep  4 10:58:14 2015
root@porter:~# insmod hello.ko
root@porter:~# rmmod hello.ko
root@porter:~# dmesg | grep World
[   95.450776] Hello World!
[   99.770700] Goodbye Cruel World!
root@porter:~#
```

Thank you !!