Intel Do-It-Yourself Challenge
Compile C/C++ for Galileo

Nicolas Vailliet
www.Intel-Software-Academic-Program.com
paul.guermonprez@intel.com
Intel Software
2014-02-01

Why ?

C/C++ code ?

Why C/C++ code ?

Arduino sketches and node.js scripts are great,
well known methods to develop smart objects.
But what if you want to write C/C++ code ?
Maybe you like it, or want better performance
or need to use certain C/C++ libraries.

CPU intensive code on a Galileo ?

After all the Quark SoC has a lot of power to offer,
processing data from sensors, sound or webcam can be
done on the Galileo board itself.
Why use a remote server if you can do it locally ?

intel Software

Native code ?

Processor compatibility

The core inside the Intel Quark System on Chip is ia32. The assembly is nearly the same as a Pentium processor. It means you could easily develop, compile and even execute on any regular Linux system running on an Intel machine. It's simple.

Libraries compatibility

Yocto has a specific set of libraries and Quark is 32bit, that's why you need to install a build environment compatible with Yocto and Quark on your workstation.

Compile on the Galileo

Compile on the Galileo itself

Add gcc to Yocto

You can add gcc and tools to the Galileo Yocto build and compile for Galileo on the Galileo itself. It works !

Performance

As the Galileo is not as fast as a workstation,
it's not convenient for large projects.

Using Yocto?

Compile your code faster on your computer and then, run it on Galileo!

Generate the cross compile toolchain

Yocto output files

Rebuild Yocto vs build for Yocto

Yocto can be used in two different ways. After a short configuration phase, you will choose between generating a Linux file system file or the cross compile development environment (or toolchain).

And then, it will take up to 8 hours to compile output files, on a notebook (depending on system configuration and network connection speed).

On a Sandy bridge workstation, it lasts less than 2 hours.

intel Software

Cross compilation

Generating a cross compiler

This course will cover how to generate the cross compile toolchain. But what is a cross compiler?

A cross compiler allows you to compile a code for Intel Quark SoC, on your computer, and then to move it to the Galileo board to execute it.

intel Software

You'll need

Board Support Package Sources for Intel Quark

An archive with all tools you need, available on Galileo drivers website.
https://communities.intel.com/docs/DOC-22226

Storage and internet bandwidth

Make sure you have 100Gb available on your hard drive.
We'll also need a good internet connectivity.

CPU

If you just want to compiling a small code, no problem.
You can even do that on the Galileo itself (if you have already installed the full Linux image, see our course about it).

But if you'd like to rebuild the full Yocto OS or generate the cross compile toolchain, it will take up to 8 hours on a workstation.

intel Software

You'll need

Linux

We'll use Ubuntu 12.04

Packages

"sudo apt-get install **build-essential** sed wget cvs subversion git-core coreutils unzip texi2html texinfo libsdl1.2-dev docbook-utils gawk python-pysqlite2 diffstat help2man make gcc g++ desktop-file-utils chrpath libgl1-mesa-dev libglu1-mesa-dev mercurial autoconf automake groff libtool xterm p7zip-full bitbake"

Procedure (1/6)

Uncompress
7z x Board_Support_Package_Sources*.7z

Rename your BSP folder with a shorter name
mv Board_Support_Package_*** BSP_Galileo

Unpack
tar xvzf meta-clanton_*.tar.gz

Go to the new Yocto home folder
cd meta-clanton_...

Download, compile and set up Poky, a tool used by Yocto
./setup.sh
source poky/oe-init-build-env yocto_build

intel Software

Procedure (2/6)

Setting up Yocto receipe

By default, Yocto is ready to compile a tiny Linux image or cross compile toolchain. But, we want to compile, debug and use libraries.

Asking for a full configuration:

Edit the conf/local.conf file.

Change “clanton-tiny” to “clanton-full”.

Set BB_NUMBER_THREADS and PARALLEL_THREADS to “number of cores your processor has multiply by 3”.

Save the file.

Procedure (3/6)

Disable uClibc

It will disable uClibc, and replace it by EGlibC, which have more features and is commonly used under Linux.

Edit "../meta-clanton-distro/recipes-multimedia/v4l2apps/v4l-utils_0.8.8.bbappend"

Comment these 3 lines:
#FILESEXTRAPATHS_prepend := "${THISDIR}/files:"
#SRC_URI += file://uclibc-enable.patch
#DEPENDS += "libiconv"

Procedure (4/6)

Get a default config

copy the full image configuration from a sample:
cp ../meta-clanton-distro/recipes-core/images/image-full.bb ../meta-clanton-distro/recipes-core/images/image-sdk.bb

Edit the default config

These options are set up for generating the associated Linux image later.
You can (un)comment features you do (not) want. The image size will be 3GB.

edit ../meta-clanton-distro/recipes-core/images/image-sdk.bb :

IMAGE_INSTALL = "packagegroup-core-boot ${ROOTFS_PKGMANAGE_BOOTSTRAP} ${CORE_IMAGE_EXTRA_INSTALL} packagegroup-core-basic packagegroup-core-lsb kernel-dev"

IMAGE_FEATURES += "package-management tools-sdk dev-pkgs tools-debug eclipse-debug tools-profile tools-testapps debug-tweaks"

IMAGE_ROOTFS_SIZE = "3072000"

intel Software

Procedure (4/6)

Get a default config

copy the full image configuration from a sample:
cp ../meta-clanton-distro/recipes-core/images/image-full.bb ../meta-clanton-distro/recipes-core/images/image-sdk.bb

Edit the default config

These options are set up for generating the associated Linux image later.
You can (un)comment features you do (not) want. The image size will be 3GB.

edit ../meta-clanton-distro/recipes-core/images/image-sdk.bb :

IMAGE_INSTALL = "packagegroup-core-boot ${ROOTFS_PKGMANAGE_BOOTSTRAP} ${CORE_IMAGE_EXTRA_INSTALL} packagegroup-core-basic packagegroup-core-lsb kernel-dev"

IMAGE_FEATURES += "package-management tools-sdk dev-pkgs tools-debug eclipse-debug tools-profile tools-testapps debug-tweaks"

IMAGE_ROOTFS_SIZE = "3072000"

intel Software

Procedure (5/6)

Last details

To have a Linux system that can support full Galileo connectivity (pins, pwm…), you have to apply the following patch:

Edit *../meta-clanton-bsp/recipes-kernel/linux/files/clanton.patch*

+static unsigned int i2c_std_mode = 1;

This is line #10722. By default, this variable is not initialized. You set it to 1.

intel Software

Procedure (6/6)

Launch the incredible machine
It will compile the cross compile toolchain.
This is the step that will take a while….

bitbake image-sdk –c populate_sdk

Install the environment
To use the new environment, execute the script:
./tmp/deploy/sdk/clanton-tiny-***-1.4.2.sh

Link your new headers and libs
source /opt/clanton-full/1.4.2/environment-******-linux

You're now ready to compile a program.

intel Software

Linking and compiling

${CC} and ${CXX} variables

To compile a C program, use CC, a environment variable referring to gcc for Intel Quark architecture (C compiler).

${CC} myfile.c –o myfile

And use ${CXX} instead of ${CC} to use g++ for Intel Quark architecture (C++ compiler).

Need a library?

If you want to use a library, just add :
`pkg-config LIBNAME --libs`

at the end of the C++ compile command.

Try it with opencv or libusb-1.0 (cf. other courses).

intel Software

Run your program

SSH and SCP

You are compiling on your workstation, but you need to execute on the Galileo board. Remember to send the binary over the network, scp is very convenient and secure (it uses SSH).

scp mybinary root@192.168.1.XXX:~

Execute

You can now connect over SSH and execute your binary.

ssh root@192.168.1.XXX

./mybinary

Tips

Reuse the cross compile toolchain

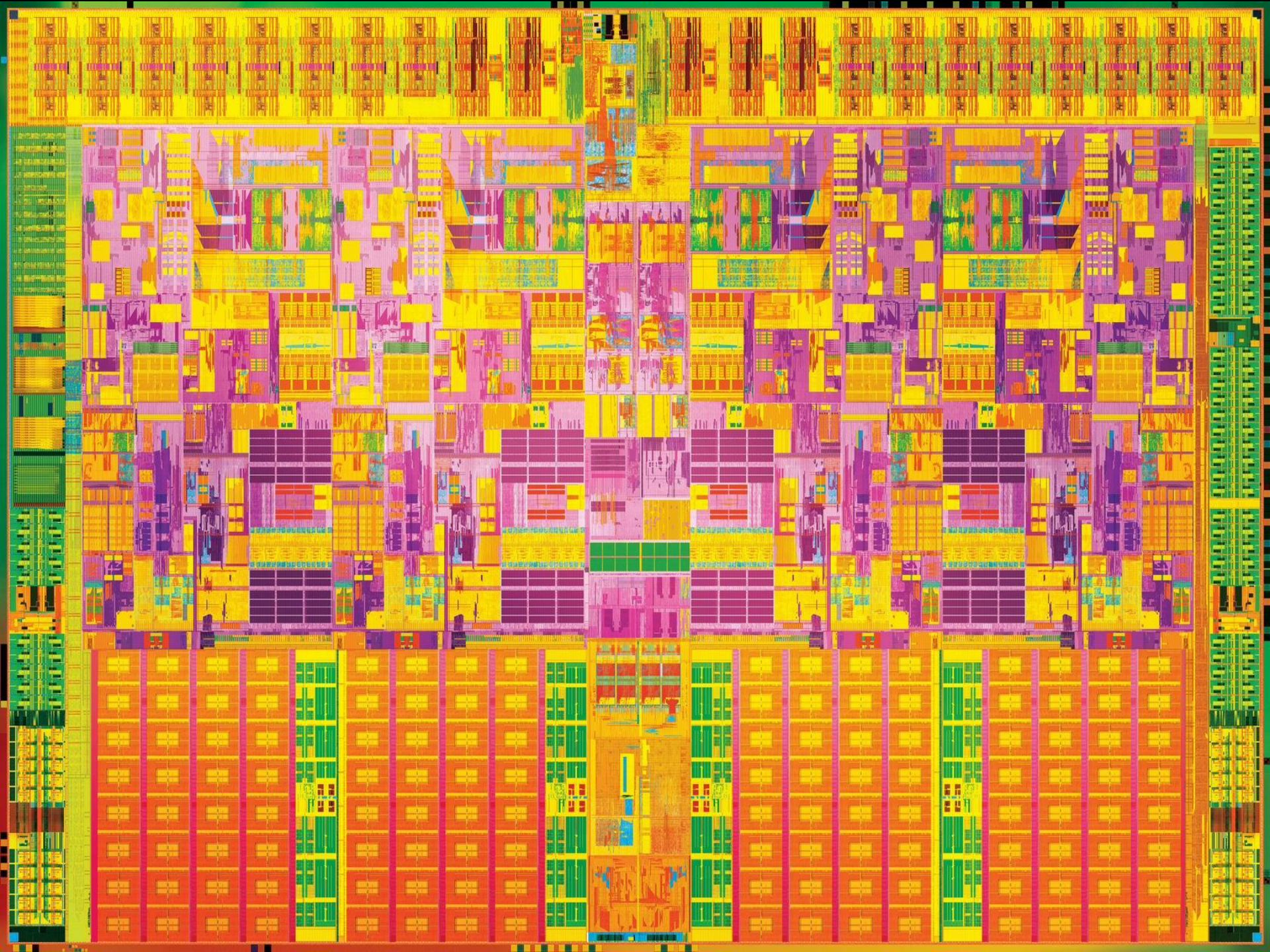
Once you have generated the cross compile toolchain, you will only need to execute this command:

>source /opt/clanton-full/1.4.2/environment-******-linux

You have to run this command every time you open a new terminal/shell.

Using gcc and g++ to compile a program for your computer is not recommended (it should not link the program properly).

Keep your Board Support folder if you plan to generate other output files with Yocto.



License Creative Commons – By 3.0

You are free:

- **to Share** — to copy, distribute and transmit the work
- **to Remix** — to adapt the work
- to make commercial use of the work

Under the following conditions:

- **Attribution** — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

With the understanding that:

- **Waiver** — Any of the above conditions can be waived if you get permission from the copyright holder.
- **Public Domain** — Where the work or any of its elements is in the public domain under applicable law, that status is in no way affected by the license.
- **Other Rights** — In no way are any of the following rights affected by the license:
  - Your fair dealing or fair use rights, or other applicable copyright exceptions and limitations;
  - The author's moral rights;
  - Rights other persons may have either in the work itself or in how the work is used, such as publicity or privacy rights.
- **Notice** — For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to this web page.

http://creativecommons.org/licenses/by/3.0/