SystemC Installation Guide

Written by	***Responsibility***
Vittoriano Muttillo	RTD-A
Verified by	
Luigi Pomante	Project Manager
Approved by	
Luigi Pomante	Project Manager

CHANGE RECORDS

ISSUE	DATE	§ CHANGE RECORDS	AUTHOR
1.0	12/08/2019	First issue.	V. Muttillo
2.0	12/08/2023	Second issue.	V. Muttillo
3.0	02/04/2024	Third issue.	V. Muttillo

TABLE OF CONTENTS

1. INTRODUCTION

1.1. SCOPE AND PURPOSE

The purpose of this document is to setting up SystemC library based on an edit of instructions that come with the source and configuring and running an eclipse-IDE C++ Project. As of now (August 2019), systemc-2.3.3 is the most recent SystemC version. It provides several improvements over systemc-2.3.2, including bug fixes and several new features that are disabled by default.

1.2. REFERENCE DOCUMENTS

Acronym	Reference	Title
REF1	http://karibe.co.ke/2014/02/setting-up-systemc-and-eclipse-for-c-hardware-simulation/	Setting up SystemC and Eclipse for C++ hardware simulation

1.3. DEFINITIONS AND ACRONYMS

2. MAY THE SOURCE BE WITH YOU

To get the source code from accellera systems, you need to head to the download page:

https://www.accellera.org/downloads/standards/systemc

and download the latest version of the library. Place and extract the source in your home folder and cd there.

cd systemc-2.3.2

3. PRE-REQUISITES

Most Linux installations have the required tools by default, just make sure you have make installed. If not, install it:

sudo apt-get install build-essential automake libtool

To check that you have these already, use the which command:

which g++

Then you have to set some environmental variables:

export CXX=g++

If you have a custom compiled gcc, you have the /usr/local/bin/gcc and its g++ as default compiler, probably an older version for building something, then you need to be more specific with the system installed gcc path like so:

export CXX=/usr/bin/g++

export CC=/usr/bin/gcc

Then create a working directory, where you will build the library before installing it

mkdir objdir

get into that directory

cd objdir

4. CONFIGURING

Create a folder where the installation will be made under /usr/local

sudo mkdir /usr/local/systemc-2.3.3

While at 'objdir' folder, run the command

../configure --prefix=/usr/local/systemc-2.3.3

This configures the installation folder and generates the makefiles, among other things…. Last few lines output:

```
configure: creating ./config.status
config.status: creating Makefile
config.status: creating src/Makefile
config.status: creating src/systemc.pc
config.status: creating src/tlm.pc
config.status: creating src/sysc/Makefile
config.status: creating src/sysc/packages/boost/Makefile
config.status: creating src/sysc/packages/qt/Makefile
config.status: creating src/tlm_core/Makefile
config.status: creating src/tlm_utils/Makefile
config.status: creating examples/Makefile
config.status: creating examples/sysc/Makefile
config.status: creating examples/tlm/Makefile
config.status: creating examples/tlm/common/Makefile
config.status: creating docs/Makefile
config.status: creating docs/sysc/doxygen/Doxyfile
config.status: creating docs/tlm/doxygen/Doxyfile
config.status: executing depfiles commands
config.status: executing libtool commands
------------------------------------------------------------------------
Configuration summary of SystemC 2.3.3 for x86_64-unknown-linux-gnu
------------------------------------------------------------------------

 Directory setup (based on classic layout):
   Installation prefix (aka SYSTEMC_HOME):
      /usr/local/systemc-2.3.3
   Header files  : /include
   Libraries     : /lib-linux64
   Documentation : /docs
   Examples      : /examples
```

```
Architecture    : linux64
Compiler        : /usr/bin/g++ (C/C++)

Build settings:
  Enable compiler optimizations  : yes
  Include debugging symbols      : no
  Coroutine package for processes: QuickThreads
  Enable VCD scopes by default   : yes
  Disable async_request_update   : no
  Phase callbacks (experimental) : no

------------------------------------------------------------------------
```

5. MAKE THE LIBRARY

Assuming configuration went on finishing smoothly, build the library

make

If you want it to run faster, make use of multiple threads with the -j option, I used

make -j4

6. INSTALL THE LIBRARY IN YOUR SYSTEM

sudo make install

After this command, you can check that the library was properly installed:

ls /usr/local/systemc-2.3.2/lib-linux64/

libsystemc-2.3.2.so libsystemc.a libsystemc.la libsystemc.so pkgconfig

Note: for a 32-bit system, the library folder is /usr/local/systemc-2.3.2/lib-linux/. you can type while using tab to auto-complete paths

To configure the library with the standard Linux library path, which will ease up linking stage of project build:

sudo ln -s /usr/local/systemc-2.3.2/lib-linux64/libsystemc-2.3.2.so /usr/lib/libsystemc-2.3.2.so

You can check the path is correct:

ls -l /usr/lib/libsystemc-2.3.2.so

lrwxrwxrwx 1 root root 56 Apr 26 12:31 /usr/lib/libsystemc-2.3.2.so -> /usr/local/systemc-2.3.2/lib-linux64/libsystemc-2.3.2.so

This library path problem has of late been persistent and the solution is to add a config file under /etc/ld.so.conf.d/ as follows:

sudo gedit /etc/ld.so.conf.d/systemc.conf

add the path information to that file, i.e "/usr/local/systemc-2.3.2/lib-linux/" or "/usr/local/systemc-2.3.2/lib-linux64/" depending on your system.

then run

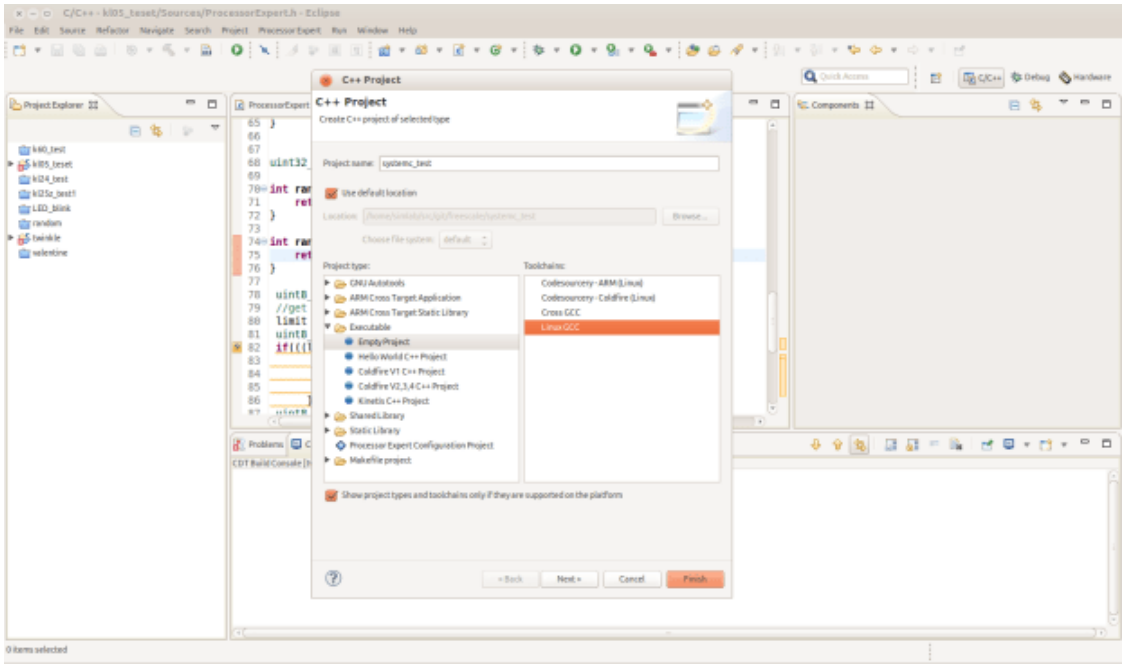
sudo ldconfig

7. GET ECLIPSE

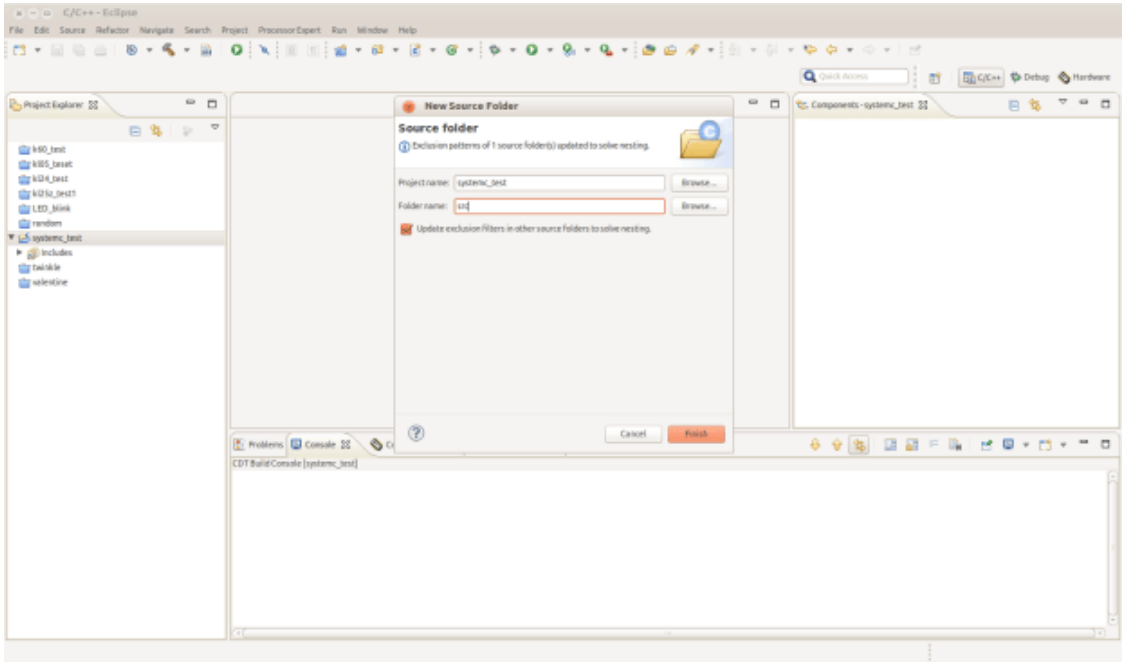
If you already have eclipse, go to step 7. If not, go to Eclipse.org and download the Eclipse IDE for C/C++ developers. Choose 32-bit/64-bit Linux version and the closest mirror… Extract under the home folder and optionally set up a .desktop shortcut.

8. CREATE A NEW ECLIPSE PROJECT

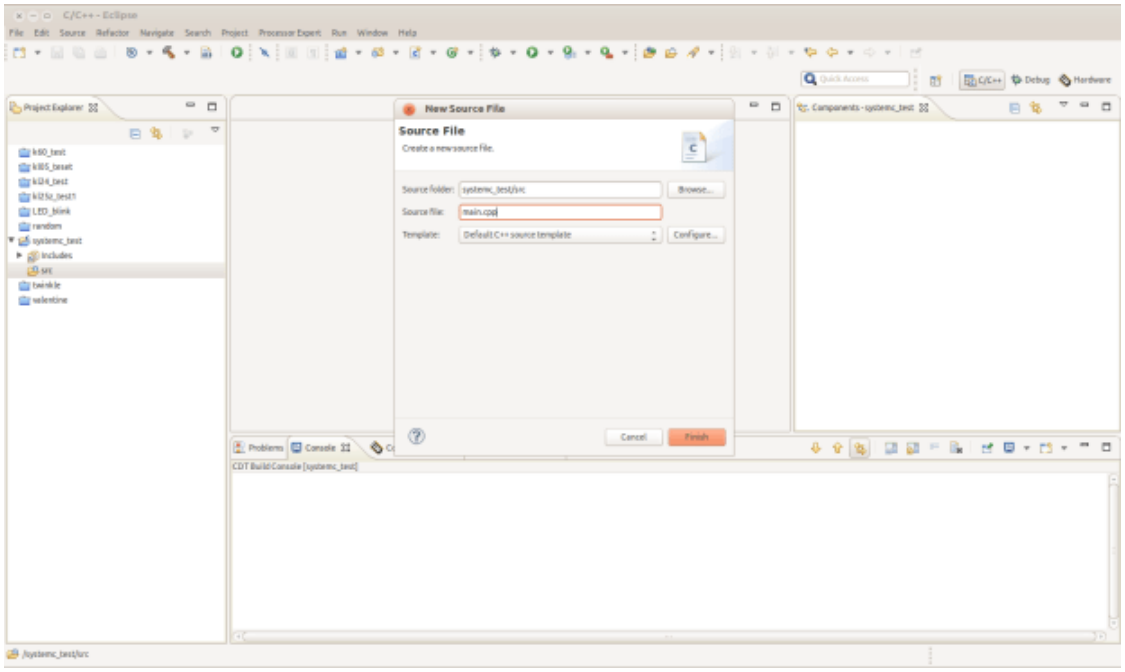
Open Eclipse and create a new C++ project, under New menu



Create a new source folder src in your project

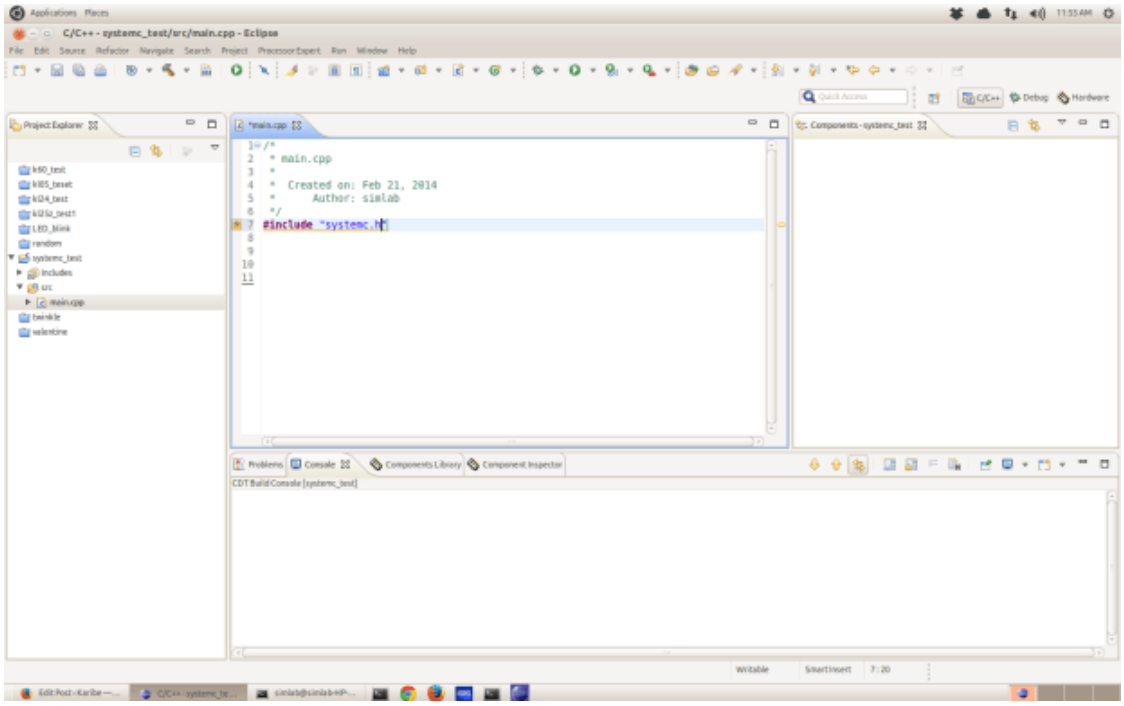


and inside it a new file main.cpp

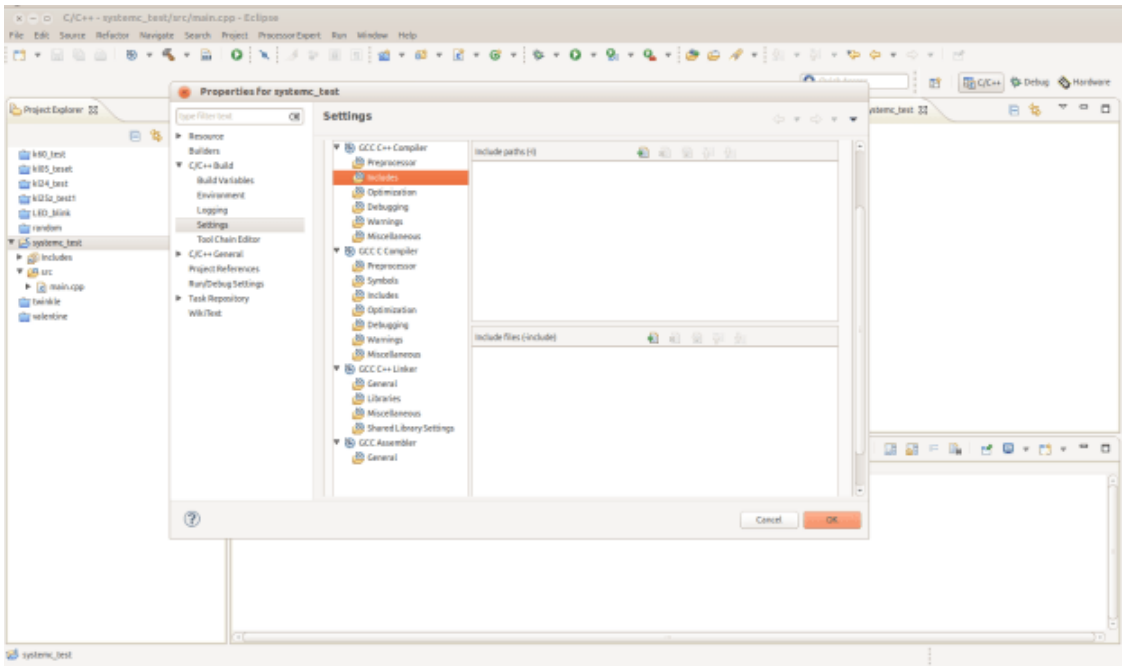


9. CONFIGURE SYSTEMC LIBRARY PATH FOR THE PROJECT

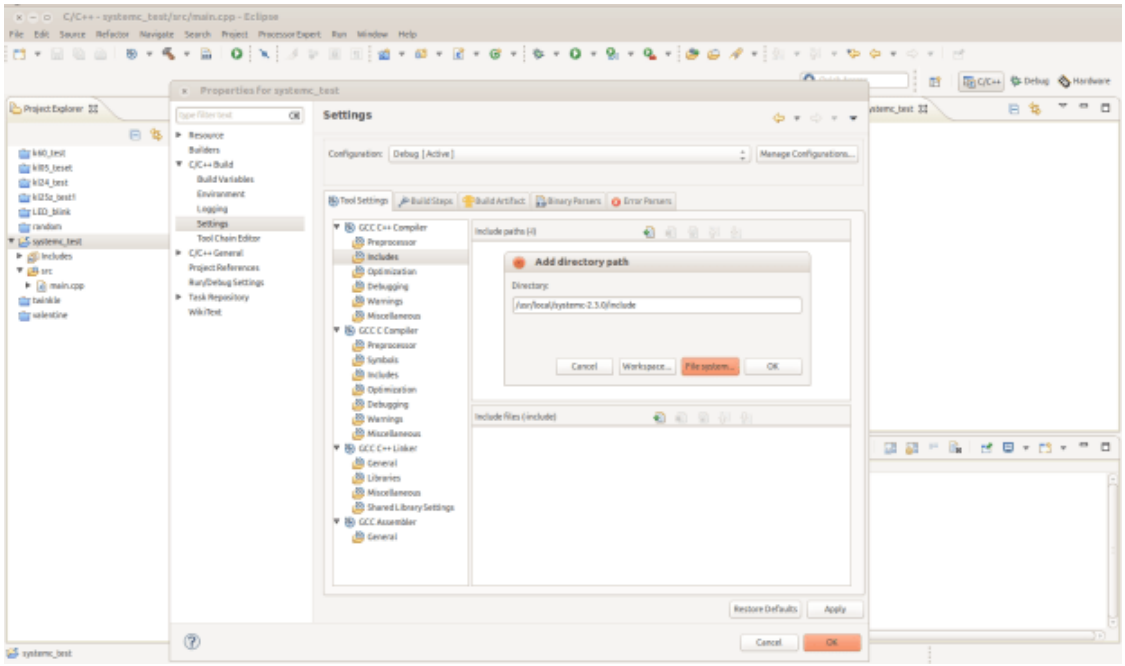
At this point, your project isn't configured with the library path:



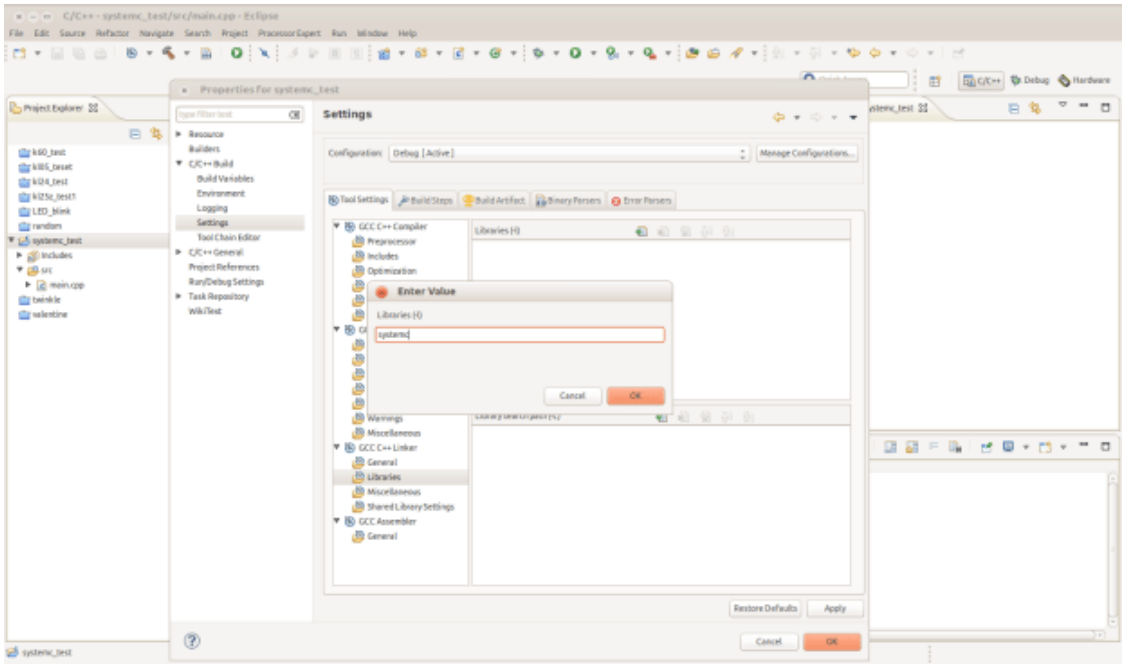
Right Click on the project root folder in the project explorer view and click on properties, alternatively, go to the Project menu and click properties.



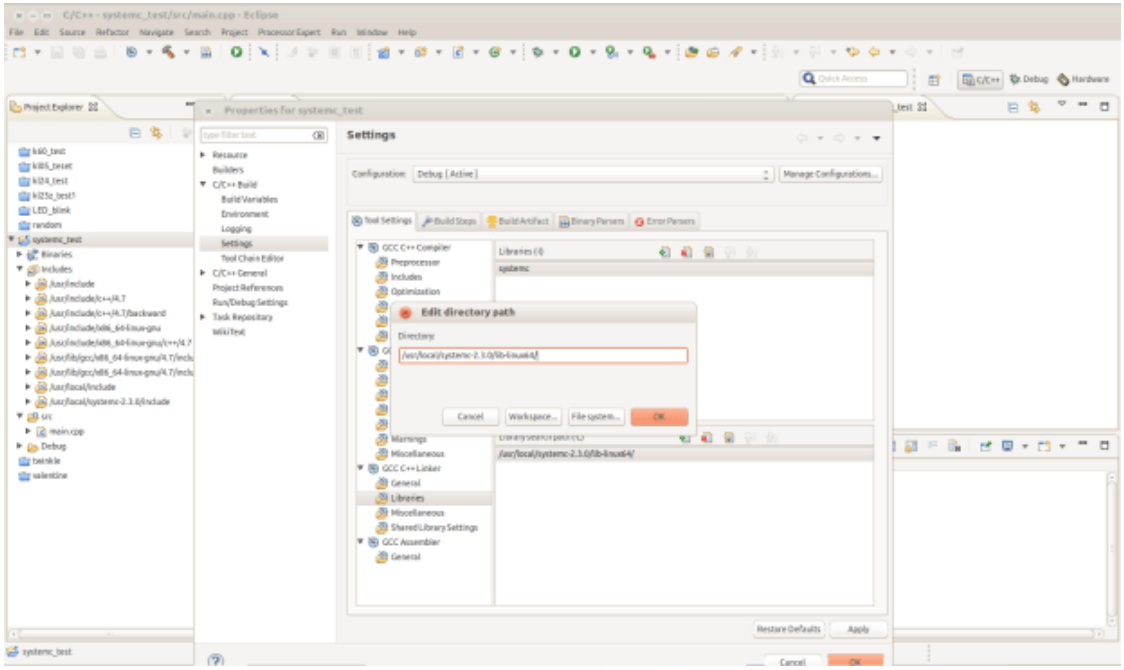
Expand the C/C++ Build entry, and click on settings. Under tool settings tab, expand GCC C++ Compiler and click on includes. On the right under include paths(-l), click the + icon and add a path to /usr/local/systemc-2.3.2/include/ by browsing File System. Note: The path in the image is actually different as it was added in 2014 when the latest version was still 2.3.0



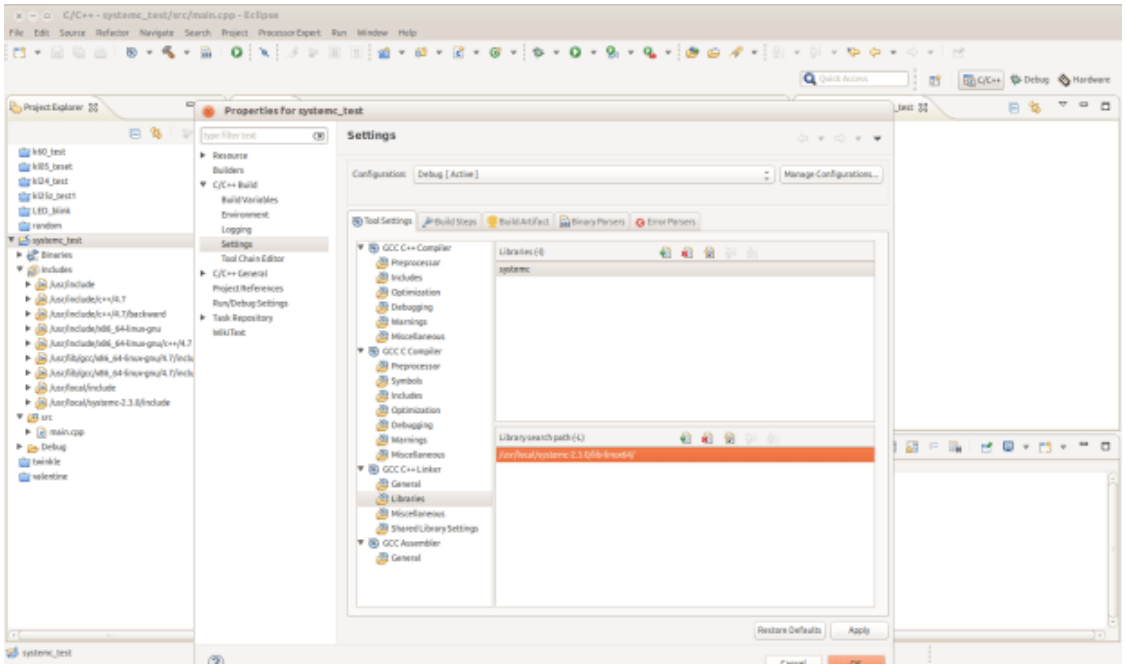
Now expand GCC C++ Linker and under Libraries, at the top Libraries(-l) entry, click on the + icon and add the library name “systemc”



then at the bottom Library search path, click on the + icon and on the pop-up, add the path to the installation directory /usr/local/systemc-2.3.2/lib-linux64 for 64-bit or /usr/local/systemc-2.3.2/lib-linux for 32-bit systems. Note: The path in the image is for systemc version 2.3.0, but we are using 2.3.2



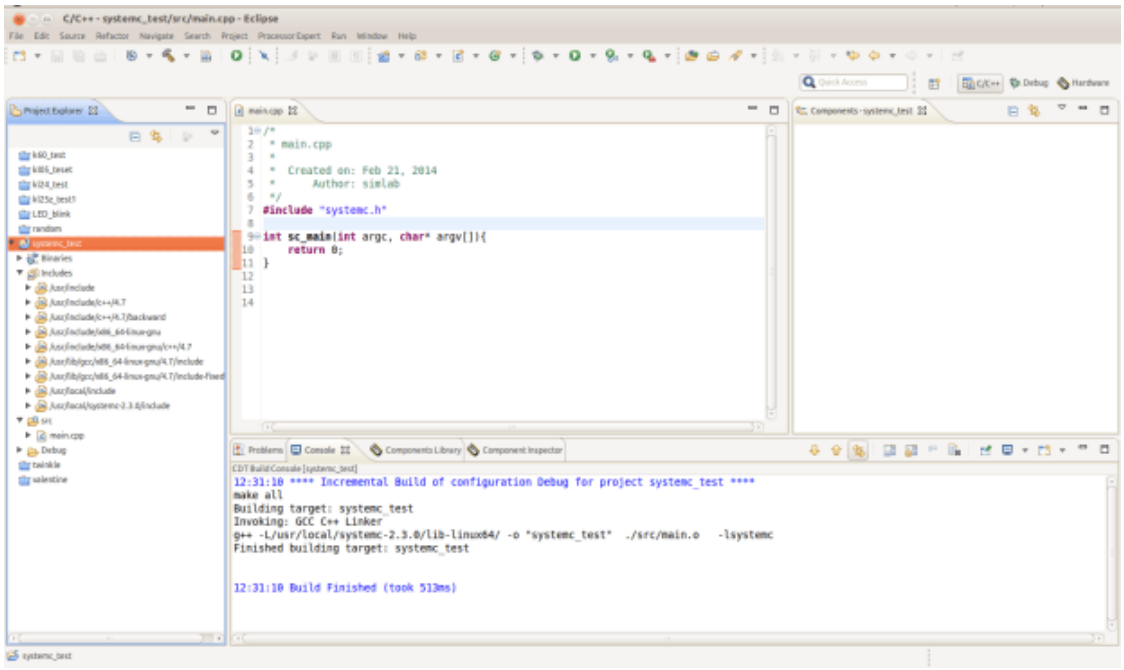
eclipse systemc library search path



eclipse systemc library configured

10. BUILD AND RUN

Click on the build symbol, the hummer, and if everything is ok in terminal: saying finished building target,



click on the Run icon(like play icon) the one to the right of the IDE, not the one on the left side. At this point its hard to tell what the output of the terminal will be ☺ depends on how much you have learnt systemc and implemented in your main.cpp file.

END OF DOCUMENT