

# SLS Detectors software installation

Anna Bergamaschi

August 22, 2017

## Contents

<b>1</b>	<b>The software package</b>	<b>2</b>
<b>2</b>	<b>Requirements</b>	<b>2</b>
2.1	Qt4 installation . . . . .	3
2.2	Qwt installation . . . . .	3
2.3	Installation with YUM . . . . .	4
2.4	Root installation . . . . .	4
<b>3</b>	<b>Compilation</b>	<b>5</b>
<b>4</b>	<b>Detector upgrade</b>	<b>6</b>
4.1	MYTHEN upgrade . . . . .	7
4.1.1	Firmware upgrade . . . . .	7
4.1.2	Software upgrade . . . . .	7

The SLS detectors software is intended to control the detectors developed by the SLS Detectors group.

It provides a command line interface (text client), a graphical user interface (GUI) as well as an API that can be embedded in your acquisitions system, some tools for detector calibration and the software to receive the data from detector with high data throughput (e.g. GOTTHARD, EIGER).

## 1 The software package

The complete software package is composed of several programs which can be installed (or locally compiled) depending on the needs:

- The **slsDetector shared and static libraries** which are necessary for all user interfaces.  
The class `slsDetectorUsers` can be used as API from your acquisition software (see separate documentation).
- The **command line interfaces** (`sls_detector_put`, `sls_detector_get`, `sls_detector_acquire`, `sls_detector_help`), which are provided to communicate with the detectors using the command line and eventually to the data receiver
- The **data receiver** (`slsReceiver`), which can be run on a different machine, receives the data from the detector and interfaces to the control software via TCP/IP for defining e.g. the file name, output path and return status and progress of the acquisition
- The **graphical user interface** (`slsDetectorGUI`) which provides a user friendly way of operating the detectors with online data preview
- The **calibration wizards** (`energyCalibrationWizard`, `angularCalibrationWizard`) to analyze the data and produce the energy or angular calibration files
- The **GOTTHARD and MYTHEN virtual servers** to simulate the detectors behavior (however only control commands work, not the data acquisition itself)

Please refer to the SLS Detectors FAQ for additional documentation.

## 2 Requirements

The software is written in C/C++.

It needs to be able to access the shared memory of the control PC and communicate to the detectors over TCP/IP. Therefore the detector should receive a proper IP address (either DHCP or static) and no firewall should be present between the control PC and the detector.

For installing the slsDetector shared and static libraries and the slsDetectorClient software, any Linux installation with a working gcc should be fine.

The slsDetectorGUI is based on Qt4 with Qwt libraries.

The calibration wizards are based on the CERN Root data analysis framework.

To compile the software you will need the whole Qt4, Qwt and Root installation, including the header files.

To run the software, it is enough to have the Qt4, Qwt or Root libraries appended to the LD\_LIBRARY\_PATH.

## 2.1 Qt4 installation

A Qt version equal or higher than 4.6 is required.

You can retrieve the Qt4 libraries using YUM or download the open source version from e.g. <ftp://ftp.qt.nokia.com/qt/source/qt-everywhere-opensource-src-4.8.1.tar.gz>

To install:

```
> gunzip qt-everywhere-opensource-src-4.6.2.tar.gz
> tar xvf qt-everywhere-opensource-src-4.6.2.tar
> ./configure
> make
> make install
```

By default Qt4 will be installed in /usr/local/Trolltech/Qt-4.8.1/ Edit your .bashrc to define the environment variable QTDIR and add the libraries and executables to your LD\_LIBRARY\_PATH and PATH:

```
export QTDIR=/usr/local/Trolltech/Qt-4.8.1
export PATH=$QTDIR/bin:$PATH
export LD_LIBRARY_PATH=$QTDIR/lib:$LD_LIBRARY_PATH
```

If your system also has Qt3 installed, make sure that QTDIR, PATH and LD\_LIBRARY\_PATH point to Qt4 before installing Qwt (and of course compiling and running the GUI).

## 2.2 Qwt installation

A Qwt version equal or higher than 5 is required.

Before installing it, make sure that your QTDIR, LD\_LIBRARY\_PATH and PATH point to the correct Qt4 version.

You can retrieve the Qwt libraries using YUM or download the open source version via svn:

```
> svn co https://qwt.svn.sourceforge.net/svnroot/qwt/branches/qwt-6.0
```

To install:

```
> cd qwt-6.0
> qmake
> make
> make install
```

By default Qwt will be installed in /usr/local/qwt-6.0 Edit your .bashrc to define the environment variable QWTDIR and add the libraries to the LD\_LIBRARY\_PATH:

```
export QWTDIR=/usr/local/qwt-6.0-svn/
export LD_LIBRARY_PATH=$QWTDIR/lib:$LD_LIBRARY_PATH
```

## 2.3 Installation with YUM

You must install the qt4 and qwt development package i.e.

```
> yum install qt-devel qwt-devel
```

and then edit your .bashrc as follows

```
export ROOTSYS=/opt/root/5.28.00
export QTDIR=/usr/lib64/qt4
export QWTDIR=/usr/include/qwt
export PATH=$QTDIR:bin:$PATH
```

You should then continue with the root installation.

To compile, you should edit the file slsDetectorGui/slsDetectorGui.pro as follows.

All lines containing \$QTDIR and \$QWTDIR should be commented, except in the INCLUDEPATH (\$QWTDIR \).

Moreover the LIBS line should be changed -L\$(QWTDIR)/lib to -L\$(QWTDIR)/lib64.

## 2.4 Root installation

The software has been developed and tested with root 5.20, but any version should work.

Download the sources via svn:

```
> svn co https://root.cern.ch/svn/root/trunk root
```

To install:

```
> cd root
> ./configure --enable-qt
> make
> make install
```

Edit your .bashrc to define the ROOTSYS environment variable and add the libraries and executables to the LD\_LIBRARY\_PATH and PATH:

```
export ROOTSYS=/usr/local/root
export PATH=$ROOTSYS/bin:$PATH
export LD_LIBRARY_PATH=$ROOTSYS/lib:$LD_LIBRARY_PATH
```

You can also download the binaries, assuming that your linux and gcc versions match:

<http://root.cern.ch/drupal/content/production-version-534>

### 3 Compilation

If you simply want to install the software in the working directory you can:

`make` compile the library, the command line interface and the receiver

`make lib` compile only the library

`make textclient` compile the command line interface (and the library, since it is required)

`make stextclient` compile the command line interface statically linking the library (and the library, since it is required)

`make receiver` compile the data receiver (and the library, since it is required)

`make sreceiver` compile the data receiver statically linking the library (and the library, since it is required)

`make gui` compile slsDetectorGUI - requires a working Qt4 and Qwt installation

`make calWiz` compile the calibration wizards - requires a working root installation

`make doc` compile documentation in pdf format

`make htmldoc` compile documentation in html format

`make install_lib` installs the libraries, the text clients, the documentation and the includes for the API

`make install` installs all software, including the gui, the cal wizards and the includes for the API

`make confinstall` installs all software, including the gui, the cal wizards and the includes for the API, prompting for the install paths

`make clean` remove object files and executables

`make help` lists possible targets

`make mythen_virtual` compile a virtual MYTHEN detector server (works for control commands, not for data taking)

make gotthard\_virtual compile a virtual GOTTHARD detector server (works for control commands, not for data taking)

The path where the files binaries, libraries, documentation and includes will be installed can either be defined interactively by sourcing the `configure` script (not executing!) or during compilation using `make confinstall` or defined on the command line defining one (or all) the following variables (normally `INSTALLROOT` is enough:

`INSTALLROOT` Directory where you want to install the software. Defaults to `PWD`

`BINDIR` Directory where you want to install the binaries. Defaults to `bin/`

`INCDIR` Directory where you want to put the header files. Defaults to `include`

`LIBDIR` Directory where you want to install the libraries. Defaults to `bin/`

`DOCDIR` Directory where you want to copy the documentation. Defaults to `doc/`

To be able to run the executables, append the `BINDIR` directory to your `PATH` and `LIBDIR` to the `LD_LIBRARY_PATH`.

To run the GUI, you also need to add to your `LD_LIBRARY_PATH` the Qt4 and Qwt libraries, without the need to install the whole Qt and Qwt developer package:

- `libqwt.so.6`
- `libQtGui.so.4`
- `libQtCore.so.4`
- `libQtSvg.so.4`

To run the calibration wizards it is preferable to have a complete Root installation (binaries), with `ROOTSYS` defined and the libraries added to the `LD_LIBRARY_PATH`.

## 4 Detector upgrade

Sometimes the upgrade of the communication software, can require an upgrade of the software and/or firmware running on the detector as well.

In these cases, the users are not expected to compile the software themselves (which would require dedicated softwares) but only to download on the detector board the programming files and/or communication program provided by the SLS Detectors group.

## 4.1 MYTHEN upgrade

### 4.1.1 Firmware upgrade

To upgrade the firmware you need either a working version of the Altera Quartus software or of the Quartus programmer, which can easily be downloaded from <https://www.altera.com/download/programming/quartus2/pq2-index.jsp>. Normally installation of the software and of the driver for the USB-Blaster (provided together with the MYTHEN detector) are simpler under Windows.

Under Windows, the first time that you connect the USB-Blaster to one of your USB ports, you will be asked to install new hardware. Set the path to search for the driver to: `C:\altera\80sp1\qprogrammer\drivers\usb-blasterp` (where `C:\altera\80sp1\qprogrammer\` is assumed to be the path where your Quartus version is installed).

1. After starting the Quartus programmer, click on Hardware Setup and in the "Currently selected hardware" window select USB-Blaster.
2. In the Mode combo box select "Active Serial Programming".
3. Plug the end of your USB-Blaster WITH THE ADAPTER PROVIDED in the connector ASMI on the MCS board taking care that pin1 corresponds to the one indexed and with the rectangular pad.
4. Click on add file and from select the programming file provided when the upgrade has been recommended.
5. Check "Program/Configure" and "Verify".
6. Push the start button and wait until the programming process is finished (progress bar top left).
7. In case the programmer gives you error messages, check the polarity of your cable (pin1 corresponds) and that you have selected the correct programming connector.

### 4.1.2 Software upgrade

First telnet to the board:

```
telnet mymcs.mydomain.com
username: root
password: pass
killall mythenDetectorServer
ls /mnt/flash/root
#if the directory does not exist mkdir /mnt/flash/root
```

To upgrade the software on the detector board transfer the provided software by ftp to the MCS:

```
ftp mymcs.mydomain.com
username: root
password: pass
cd /mnt/flash/root
put mythenDetectorServer
quit
```

After pressing reset on the board, the board should reboot.

If the program does not correctly start either check by using the http interface that it is started by the inittab (check that the file `/mnt/etc/inittab` ends with the line `myid2:3:once:/mnt/flash/root/mythenDetectorServer`).

Otherwise make the program executable by telnetting to the MCS and executing: `chmod a+xrw /mnt/flash/root/mythenDetectorServer`

After pressing reset on the board, the board should reboot and the acquisition program correctly start.