

Version 1.0.10 Beta
October 30, 2012

sQuid



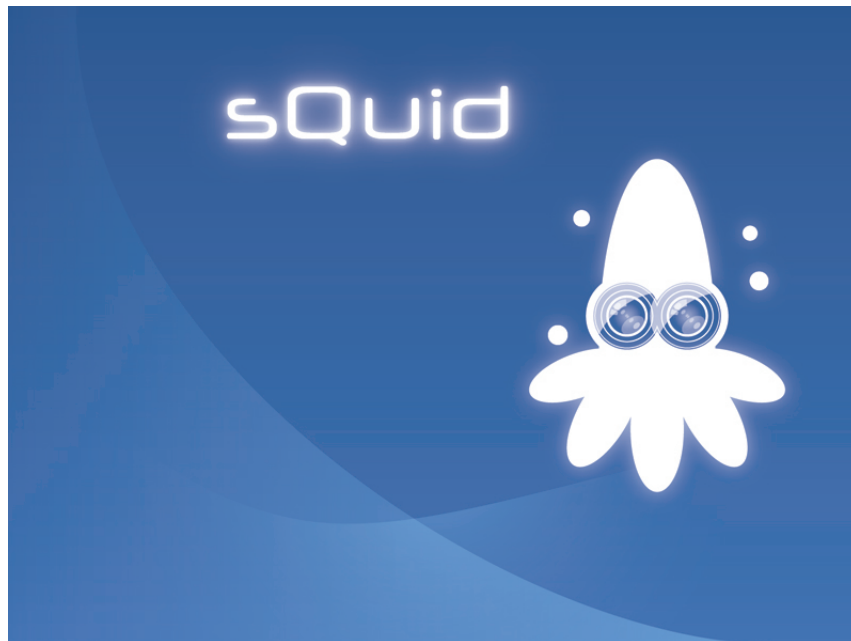
sQuid Installation Guide

Observation and interaction in experimental environments

Thomas Schaffter

sQuid Installation Guide

Observation and interaction in experimental environments



Version 1.0.10 Beta
October 30, 2012

Thomas Schaffter

thomas.schaffter@gmail.com

Laboratory of Intelligent Systems, Ecole Polytechnique Fédérale de
Lausanne (EPFL), Lausanne, Switzerland

lis.epfl.ch/squid



Table of Contents

1	Introduction	1
1.1	Overview	1
1.2	License / How to cite sQuid	1
1.3	Contact	1
2	Installation	3
2.1	Requirements	3
2.2	User and root commands	3
2.3	Vim editor	3
2.4	Install openSUSE 12.1	3
2.5	Download sQuid	4
2.6	Install third-party components	4
2.6.1	Qt-Creator	4
2.6.2	QLed library	4
2.6.3	Google logging library (glog)	5
2.6.4	Boost libraries	5
2.6.5	FireWire modules (IEEE 1394, dc1394)	5
2.6.6	Rtkit	6
2.6.7	Parallel port (IEEE 1284)	7
2.6.8	Sending emails with <i>Postfix</i>	7
2.6.9	Update environment variables	7
2.7	Install sQuid	8

1

Introduction

1.1 Overview

sQuid is an open-source C++/Qt software application designed to observe environments using FireWire cameras (IEEE 1394), while interacting with these environments through a port interface (e.g. parallel port (IEEE 1284)). We envision that this application will be very useful in scientific research as well as robotics.

As an example, we control a system of LEDs and valves while multiple cameras capture visual information from an experiment. Specifically, fruit flies (*Drosophila melanogaster*) are placed within a transparent chamber to visualize their behavior. While capturing images from two cameras, we simultaneously control a series of LEDs to excite fluorescence proteins and generate gas flow by controlling a series of valves.

1.2 License / How to cite sQuid

sQuid is an open-source software application released under a Creative Commons Attribution-NonCommercial 3.0 Unported License. A brief description of the license is available at <http://creativecommons.org/licenses/by-nc/3.0/> and the full license at <http://creativecommons.org/licenses/by-nc/3.0/legalcode>. Please cite the papers listed on lis.epfl.ch/squid when using sQuid in your publication. For commercial use, please contact the author.

1.3 Contact

The sQuid project can be contacted via its website at lis.epfl.ch/squid. Please feel free to contact the author with bug reports and information on related projects.

2

Installation

2.1 Requirements

- Operating system supporting POSIX threads (pthread) and timerfd (kernel 2.6.27 or above)
- Computer equipped with FireWire (IEEE 1394) host adapter card(s)
- FireWire camera(s) 400 (IEEE 1394a) or 800 (IEEE 1394b, faster transfer speed)
- FireWire cable(s) with alpha 6-conductor connector (IEEE 1394a) and/or beta 9-conductor connector (IEEE 1394b)¹
- Parallel port (IEEE 1284)

2.2 User and root commands

In the next sections, we use the sign '\$' and '#' to distinct commands that must be executed as *user* or *root*. This sign must not be entered in the console.

2.3 Vim editor

We use the command *vim* to edit text files. If you don't know how to use it, you can edit the files with any graphical text editors.

2.4 Install openSUSE 12.1

1. Download the ISO image of 32-bit openSUSE 12.1 or above from opensuse.org.
2. Burn the ISO image to a DVD and start a standard installation.
3. During the installation, select

¹The transfer speed is limited to FireWire 400 if a cable has at least one alpha connector.

- Software > Add *C/C++ Development* and *Qt 4 Development*
4. Once openSUSE is installed, add or remove the following software repositories using *YaST* or the command-line interface *zypper*.²
 - Add *Pacman*.
zypper ar -f -n packman http://packman.inode.at/suse/openSUSE_12.1 packman
 - *YaST* > Software Repositories > Disable the URL starting with *cd:////?devices=*
 5. Update openSUSE
zypper up

2.5 Download sQuid

1. Download the latest release of sQuid at lis.epfl.ch/squid.
2. Extract the folder *squid* from the archive (*version* is to be replaced by the current version of sQuid).
\$ tar xjvf squid-version.tar.bz2 .

Since now on, the absolute path the for folder sQuid is denoted by *SQUID_DIR*. This folder contains two other folders named *src* and *lib*. The first one contains the source code of sQuid and its absolute path is denoted by *SQUID_SRC_DIR* below. The second folder contains thrid-part libraries that are required to run sQuid. Its absolute path is *SQUID_LIB_DIR*.

Example:

SQUID_DIR: /home/tschaffter/applications/squid

SQUID_SRC_DIR: /home/tschaffter/applications/squid/src

SQUID_LIB_DIR: /home/tschaffter/applications/squid/lib

2.6 Install third-party components

2.6.1 Qt-Creator

```
# zypper in qt-creator
```

2.6.2 QLed library

1. Move to *SQUID_LIB_DIR*/qled-0.6.2/
\$ cd /home/tschaffter/applications/squid/lib/qled-0.6.2
2. Compile and install
\$ qmake qledplugin.pro
\$ make
make install

²http://en.opensuse.org/Additional_package_repositories

3. Copy *qled.h* to *QtCore include*.
cp qled.h /usr/include/QtCore/.

Reference: <http://qt-apps.org/content/show.php?content=72482>

2.6.3 Google logging library (glog)

1. Move to *SQUID_LIB_DIR/glog-0.3.2/*
2. Compile and install
\$./configure
\$ make
make install

2.6.4 Boost libraries

Install the following *Boost* libraries (version 1.46.1 or above) if not yet installed.

```
# zypper in boost-devel libboost_filesystem1_46_1 libboost_program_options1_46_1
```

If zypper doesn't find them, use the following command to list all *Boost* libraries available for installation.

```
# zypper se boost
```

Reference: <http://code.google.com/p/google-glog/>

2.6.5 FireWire modules (IEEE 1394, dc1394)

Start by checking whether the dc1394 host adapter card is available. If nothing is printed, this means that there is either no FireWire interface installed on that computer or it is not functioning properly.

```
# lspci | grep 1394
05:00.0 FireWire (IEEE 1394): Texas Instruments TSB82AA2 IEEE-1394b Link Layer Controller (rev 01)
```

Check whether *libraw1394* and *libdc1394* are installed.

```
# zypper se libraw1394
# zypper se libdc1394
```

The letter 'i' in front of a line indicates that the associated package is installed. For each package *libraw1394* and *libdc1394*, both the library (e.g. *libraw1394*) and the sources/headers (e.g. *libraw1394-devel*) must be installed. If they are not:

1. Install the library *libraw1394*
zypper in libraw1394 libraw1394-devel

2. Install the library *libdc1394*

```
# zypper in libdc1394 libdc1394-devel libavc1394
```

Check that the modules *firewire-core*, *firewire-ohci* and *firewire-sbp2* are at least loaded (the option -l below is a lowercase L).

```
# modprobe -l | grep firewire
kernel/drivers/firewire/firewire-core.ko
kernel/drivers/firewire/firewire-ohci.ko
kernel/drivers/firewire/firewire-sbp2.ko
kernel/drivers/firewire/firewire-net.ko
kernel/drivers/firewire/nosy.ko
kernel/drivers/media/dvb/firewire/firedtv.ko
kernel/sound/firewire/snd-firewire-lib.ko
kernel/sound/firewire/snd-firewire-speakers.ko
kernel/sound/firewire/snd-isight.ko
```

2.6.6 Rtkit

Rtkit is a mechanism used to allow real-time scheduling to be used by normal user processes. It is used by sQuid to generate software camera triggers with precision below one millisecond. *Rtkit* may already be installed, however a version below the required version 0.10 (dot ten or above) may be present. Get the installed version of *rtkit*:

```
# rtkitctl --version
rtkitctl 0.5
```

Here the version 0.5 is below 0.10 (dot ten). To update *rtkit* to version 0.10:

1. Move to *SQUID_LIB_DIR/rtkit/*

2. Update *rtkit* to version 0.10.

```
# rpm -Uvh rtkit-0.10_git201107141740-8.3.i586.rpm rtkit-0.10_git201107141740-8.3.src.rpm
OR
# rpm -Uvh *.rpm
```

To enable normal user processes to promote themselves to realtime priority, open the following text file:

```
# vim /etc/polkit-default-privs.local
```

Press the key “i” to enable edition, then add the next line to the end of the file (replace \TAB by a tabulation):

```
org.freedesktop.RealtimeKit1.acquire-real-time \TAB yes:yes:yes
```

Press Esc followed by “:x” to save and close the document. To apply the changes, use the command below. If a warning message is displayed, just ignore it.

```
# /sbin/set_polkit_default_privs
```

Reference: <http://git.0pointer.de/?p=rtkit.git>

2.6.7 Parallel port (IEEE 1284)

1. Check if the parallel port is available.

```
# ls -al /dev/parport*  
crw-rw-r-- 1 root lp 99, 0 Feb 24 05:10 /dev/parport0
```

2. Set the correct rights.

```
# chmod 660 /dev/parport0
```

3. Add the parallel port to the group "lp".

```
# chgrp lp /dev/parport0
```

4. Add the current user to the group "lp" (replaced *USERNAME* by your own system username).

```
# usermod -G lp USERNAME
```

5. Log out and log back to apply the changes (or wait until the end of the next section).

2.6.8 Sending emails with *Postfix*

sQuid can be configured to send experiment reports and/or major errors to specified email addresses. To enable *Postfix*:

```
# chkconfig postfix --level 345 on  
# /etc/init.d/postfix start
```

2.6.9 Update environment variables

1. Open the file `/home/USERNAME/.profile`

```
$ vim ~/.profile
```

2. Add the following lines at the end of the document (replace *SQUID_SRC_DIR* by the correct absolute path).

```
mylibs=/usr/lib/qt4/plugins/designer:SQUID_SRC_DIR/libportplayer:SQUID_SRC_DIR/libsquid  
export LIBRARY_PATH=$LIBRARY_PATH:$mylibs  
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$mylibs
```

3. Log out and log back to apply the changes.

2.7 Install sQuid

sQuid is distributed as four packages:

- **squid** The application sQuid.
- **libsquid (used by sQuid)** This library implements methods to communicate with FireWire (IEEE 1394) cameras and save frames grabbed as images in TIFF and PGM formats.
- **qportplayer** The application *QPortPlayer* allows to communicate with a port interface. *QPortPlayer* is integrated in sQuid but can also be used as a standalone application. The current version enables to control the seven data pins of the parallel port (IEEE 1284).
- **libportplayer (used by QPortPlayer)** This library implements a framework to communicate with port interface. As an example, an implementation to control the pins of the parallel port is provided.

Finally, you need to compile each package as user.

1. Move to *SQUID_SRC_DIR/libportplayer/*.
\$ qmake
\$ make
2. Move to *SQUID_SRC_DIR/qportplayer/*.
\$ qmake
\$ make
3. Move to *SQUID_SRC_DIR/libsquid/*.
\$ qmake
\$ make
4. Move to *SQUID_SRC_DIR/squid/*.
\$ qmake
\$ make

Finally, you can create command-line shortcuts for sQuid and *QPortPlayer*:

```
$ ln -s SQUID_SRC_DIR/squid/squid /home/USERNAME/bin/.  
$ ln -s SQUID_SRC_DIR/qportplayer/qportplayer /home/USERNAME/bin/.
```

sQuid and *QPortPlayer* can now be run with the command lines:

```
$ squid OR $ squid -s settings.txt  
$ qportplayer OR $ qportplayer -s settings.txt
```

where *settings.txt* are specific settings files that can be found in *SQUID_SRC_DIR/squid/* and *SQUID_SRC_DIR/qportplayer/*.