



## How to compile, install and distribute QHY\_SDK\_CrossPlatform driver and test applications on 32/64 bit Linux Ubuntu 16.xx.

Version 1.2

2017-07-16

Written by: Jan Soldan

[astrosoft@email.cz](mailto:astrosoft@email.cz)

### FOR SDK DRIVER DEVELOPERS:

1. Note: change **/projects/astrosoft** directory to your one inside lines below.
2. Go to your working copy of QHY\_SDK\_CrossPlatform SVN repository.  
**cd /projects/astrosoft/QHYCCD\_SDK\_CrossPlatform**
3. Update your working copy.  
**svn update**
4. Go to your build directory or create one if it doesn't exist.  
**cd build**
5. Remove all files.  
**rm -rf \***
6. To compile on Linux machine, start with:  
**cmake -DPLATFORM=linux ..**  
  
For cross compilation for RPI3 ARM v8 machine, start with:  
**cmake -DPLATFORM=rpi3-gcc-4.9 -DWORKSPACE\_ROOT=/opt ..**
7. Run make command to compile driver.  
**make**
8. Install driver and all necessary files.  
**sudo make install**

9. Generate distribution package.

**sudo cpack --config CpackConfig.cmake**

The **cpack** command generates distribution packages inside the **release** directory. The package name is based on CMake's variables:

`${CPACK_PACKAGE_NAME}-${CPACK_PACKAGE_VERSION}-${PLATFORM}`

For example, if the selected `${PLATFORM}` is **linux**, then the generated package is:

**qhyccdsdk-0.1.8-linux.tar.gz**

## FOR APPLICATION USERS:

1. Select requested file and unpack it:

**tar xzvf qhyccdsdk-0.1.8-linux.tar.gz**

This will create a new directory named `qhyccdsdk-0.1.8-linux`

2. Go to the `/install_scripts` directory and run:

**sudo ./linux\_install\_qhyccd\_sdk\_driver.sh**

This will install static and shared libraries, all firmwares, udev files and so on.

3. You are done. Now, you can use QHYCCD shared or static libraries.

When you unpack your tar.gz file, you will find the following subdirectories inside the **qhyccdsdk-0.1.8-linux**:

- **cmake\_modules**  
Holds several cmake modules needed for compilation of test applications.
- **doc**  
Document files.
- **firmware**  
The firmware directory contains all firmware files (\*.HEX, \*.img) needed by QHYCCD's cameras. An appropriate file is uploaded to the camera on boot automatically. If you run an installation script above, those files will be copied to: `/lib/firmware/qhy` directory.
- **include**  
The include directory contains all necessary header files needed for your own application development.
- **install\_script**  
To install the full package, please run the next command:  
**sudo ./linux\_install\_qhyccd\_sdk\_driver.sh**

- **lib**

The lib directory contains static (libqhy.a) and shared (libqhy.so.0) libraries including symbolic links. Installation script copies those files to the /usr/local/lib directory.

- **testapp**

There are several directories, each containing simple test application and corresponding CMakeLists.txt file. To compile those test applications, the CMAKE has to be installed on that machine. Please visit <https://cmake.org> how to do that. Follow the next section how to compile and run SingleFrameMode test application.

There is also a common subdirectory which contains source files related to QHYCCD cameras. Please look there to see how to control different parts of the hardware.

- **udev**

The udev directory contains udev rule file named 85-qhyccd.rules. This file is copied to the /etc/udev/rules.d/ and /lib/udev/rules.d/ directories.

## How to compile and run SingleFrameMode test application.

1. Unpack distribution package:  
**tar xzvf qhyccdsdk-0.1.8-linux.tar.gz**
2. **cd testapp**
3. **mkdir build** create directory if doesn't exist
4. **cd build**
5. **rm -rf \***
6. **cmake ../SingleFrameMode**
7. **make**
8. Connect your QHYCCD camera to USB port and run test application:  
**./SingleFrameMode**

Listing from SingleFrameMode application and QHY5II-L camera:



```
jsoldan@asus:/projects/astrosoft/QHYCCD_SDK_CrossPlatform/  
release/qhyccdsdk-0.1.8-linux/build$ ./SingleFrameMode
```

```
2017-06-09 08:38:29.229 LOG_ALARM ----- log4z  
thread started! -----  
2017-06-09 08:38:29.229 LOG_ALARM logger id=0 key=Main  
name=SingleFrameMode path=/tmp/ level=1 display=1
```

```
SDK resources initialized.  
Number of QHYCCD cameras found: 1  
Application connected to the following camera from the list:  
Index: 1, cameraID = QHY5LII-M-60f7e7106f427a411  
Open QHYCCD success.  
SetQHYCCDStreamMode success.  
InitQHYCCD success.  
GetQHYCCDChipInfo success:  
CCD/CMOS chip information:  
Chip size: width 4.800 [mm], height 3.600 [mm]  
Pixel size: width 3.750 [um], height 3.750 [um]  
Image size: 1280 x 960, bpp: 8  
SetQHYCCDResolution success.  
SetQHYCCDBinMode success.  
ExpQHYCCDSingleFrame success.  
GetQHYCCDSingleFrame success.  
CancelQHYCCDExposingAndReadout success.  
Close QHYCCD failure, error: 0  
SDK resources released.
```

```
jsoldan@asus:/projects/astrosoft/QHYCCD_SDK_CrossPlatform/  
release/qhyccdsdk-0.1.8-linux/build$
```

**Listing from SingleFrameMode application and QHY163M camera:**



```
jsoldan@asus:/projects/astrosoft/QHYCCD_SDK_CrossPlatform/
```

```
release/qhyccdsdk-0.1.8-linux/build$ ./SingleFrameMode  
2017-06-09 08:42:44.132 LOG_ALARM ----- log4z  
thread started! -----  
2017-06-09 08:42:44.133 LOG_ALARM logger id=0 key=Main  
name=SingleFrameMode path=/tmp/ level=1 display=1
```

```
SDK resources initialized.  
Number of QHYCCD cameras found: 1  
Application connected to the following camera from the list:  
Index: 1, cameraID = QHY163M-189b984ba3fe674f9  
Open QHYCCD success.  
SetQHYCCDStreamMode success.  
InitQHYCCD success.  
GetQHYCCDChipInfo success:  
CCD/CMOS chip information:  
Chip size: width 17.647 [mm], height 13.323 [mm]  
Pixel size: width 3.800 [um], height 3.800 [um]  
Image size: 4656 x 3522, bpp: 16  
SetQHYCCDResolution success.  
SetQHYCCDBinMode success.  
ExpQHYCCDSingleFrame success.  
GetQHYCCDSingleFrame success.  
CancelQHYCCDExposingAndReadout success.  
Close QHYCCD failure, error: 0  
SDK resources released.
```

```
jsoldan@asus:/projects/astrosoft/QHYCCD_SDK_CrossPlatform/  
release/qhyccdsdk-0.1.8-linux/build$
```