APPLICATION NOTE

```
// Create an instant camera object with the firs
Camera_t camera( CTlFactory::GetInstance().Creat

// Register an image event handler that accesses
camera.RegisterImageEventHandler( new CSampleIma
Ownership_TakeOwnership);

// Open the camera.
camera.Open();
```

Interfacing Basler Cameras with ROS

Applicable only to cameras that can be accessed by the Basler pylon Camera Software Suite

Document Number: AW001491
Version: 01 Language: 000 (English)
Release Date: 17 May 2018

BASLER
the power of sight

Contacting Basler Support Worldwide

Europe, Middle East, Africa

Basler AG
An der Strusbek 60–62
22926 Ahrensburg
Germany

Tel. +49 4102 463 515
Fax +49 4102 463 599

support.europe@baslerweb.com

The Americas

Basler, Inc.
855 Springdale Drive, Suite 203
Exton, PA 19341
USA

Tel. +1 610 280 0171
Fax +1 610 280 7608

support.usa@baslerweb.com

Asia-Pacific

Basler Asia Pte. Ltd.
35 Marsiling Industrial Estate Road 3
#05–06
Singapore 739257

Tel. +65 6367 1355
Fax +65 6367 1255

support.asia@baslerweb.com

www.baslerweb.com

Table of Contents

1 Introduction

This document explains exemplarily how to interface Basler GigE Vision and USB3 Vision cameras with the Robot Operating System (ROS) using the 'pylon_camera' ROS package.

ROS is an all open source framework of software libraries and tools. The framework supports the building of various robot applications. ROS provides the developing tools, algorithms and drivers for a variety of robotics platform projects.

The procedures described in this document were evaluated with the following software:

- Ubuntu 16.04.3 LTS (Xenial Xerus)
- ROS (Kinetic Kame)

ROS is able to run a large number of executables (nodes) in parallel and let them exchange data synchronously (service) or asynchronously (subscribed/published topics). In practice, the data are generally sensor queries whose result data are processed to cause robot actions.

Sensors used in robotics are single information and array detectors. In addition, cameras are also increasingly used for robotics applications. To interface cameras for robotics the ROS user community is providing camera driver wrappers and processing nodes.

ℹ	To help you with entering commands this document provides commands ready to copy and paste. The commands are given in orange after the $ prompt.

Legal Notice

Basler does not assume any liability for the functionality and suitability of any recommended open source products referenced in this Application Note. This is just a presentation of a sample use case. The readers of this Application Note are fully responsible to conduct their own testing procedures to assess the suitability of the mentioned open source products for their own applications.

2 Installing Software

Installation of the following software is described below:

- Operating system
- Basler pylon Camera Software Suite for Linux x86
- ROS Robot Operating System
- 'pylon_camera' ROS package

2.1 Operating System Installation

This document focuses on the ROS use on natively installed Linux x86 OSs.

It is reportedly possible to get the latest ROS Lunar release running on the Windows Subsystem for Linux provided by Windows 10 (1703) and newer. However, the limitations for GUI and HW support are too serious to consider this constellation.

Thus use or create a new operating system installation using a Linux ISO image. In our case an Ubuntu 16.04.3 LTS x64 image has been used. Make sure you have an internet connection on your Linux machine available. In case of difficulties check if any proxy server settings are necessary or must be adjusted.

2.2 Basler pylon Camera Software Suite for Linux x86 Installation

Download the current version of the Basler pylon Camera Software Suite for Linux x86. The procedures described in this document assume that you are using pylon v 5.0.11 or newer. Please check pylon version compatibilities when creating or using further ROS nodes.

To install the pylon Camera Software Suite for Linux, there are two installer packages available:

- tar.gz (for all Linux distributions)
- .deb (for Ubuntu and related Linux distributions)

You can download the installer packages from: http://www.baslerweb.com/

Installation Applicable to All Linux Distributions

Install the pylon SDK from the tar.gz installer package. Details about installation and configuration are available from the included INSTALL and README files.

NOTICE

Make sure to carry out the necessary adjustments as described in the INSTALL file:

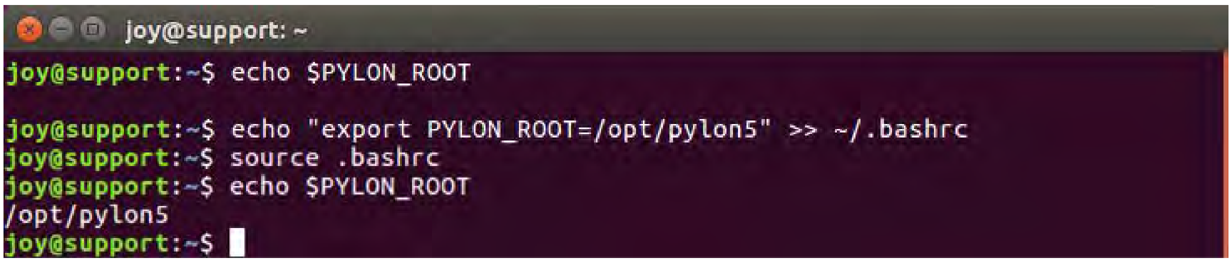
1. Run the pylon-setup-env.sh script to set the environment variable PYLON_ROOT.
2. If you want to use Basler USB3 Vision cameras, run the included setup-usb.sh script.

Installation Applicable to Ubuntu and Related Linux Distributions

Install the Basler pylon Camera Software Suite for Linux on Debian and related Linux distributions (e.g. Ubuntu) from the .deb installer package that suits your platform. You must install from the .deb installer package using the ‘dpkg’ command line tool:

```
$ dpkg –i <package.deb>
```

Set the pylon location environment variable and optionally make sure that it is persistent by adding variable creation to the ~/.bashrc file:



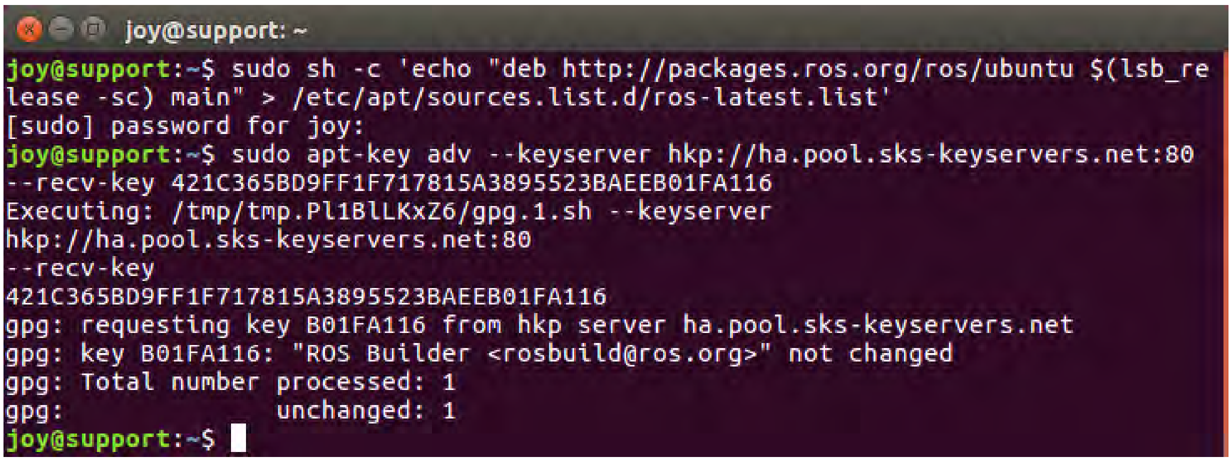
```
joy@support:~$ echo $PYLON_ROOT

joy@support:~$ echo "export PYLON_ROOT=/opt/pylon5" >> ~/.bashrc
joy@support:~$ source .bashrc
joy@support:~$ echo $PYLON_ROOT
/opt/pylon5
joy@support:~$
```

The PYLON_ROOT environment variable is necessary for pylon path identification related to development and ‘pylon_camera’ ROS package use. See below for more information about ‘pylon_camera’, designed for use with cameras supported by pylon.

2.3 ROS Robot Operating System Installation

This documentation presents a straightforward installation. The installation steps are listed without comment. For additional information, see the ROS wiki.

Preparatory Steps



```
joy@support:~$ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_re
lease -sc) main" > /etc/apt/sources.list.d/ros-latest.list'
[sudo] password for joy:
joy@support:~$ sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80
--recv-key 421C365BD9FF1F717815A3895523BAEEB01FA116
Executing: /tmp/tmp.Pl1BlLKxZ6/gpg.1.sh --keyserver
hkp://ha.pool.sks-keyservers.net:80
--recv-key
421C365BD9FF1F717815A3895523BAEEB01FA116
gpg: requesting key B01FA116 from hkp server ha.pool.sks-keyservers.net
gpg: key B01FA116: "ROS Builder <rosbuild@ros.org>" not changed
gpg: Total number processed: 1
gpg:              unchanged: 1
joy@support:~$
```

$ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'

$ sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-key 421C365BD9FF1F717815A3895523BAEEB01FA116

Installation of ROS

Below is described the installation of ROS Kinetic Kame, which is the LTS version up to 2021. For more details and possible alternative installation steps visit the Ubuntu ROS Kinetic Installation site.

This application note also applies to ROS Indigo Igloo, which has LTS up to 2019. The installation for ROS Indigo Igloo is analogous to the installation of ROS Kinetic Kame.

Install ROS Kinetic Kame:

```
$ sudo apt-get update

$ sudo apt-get install ros-kinetic-desktop-full
```

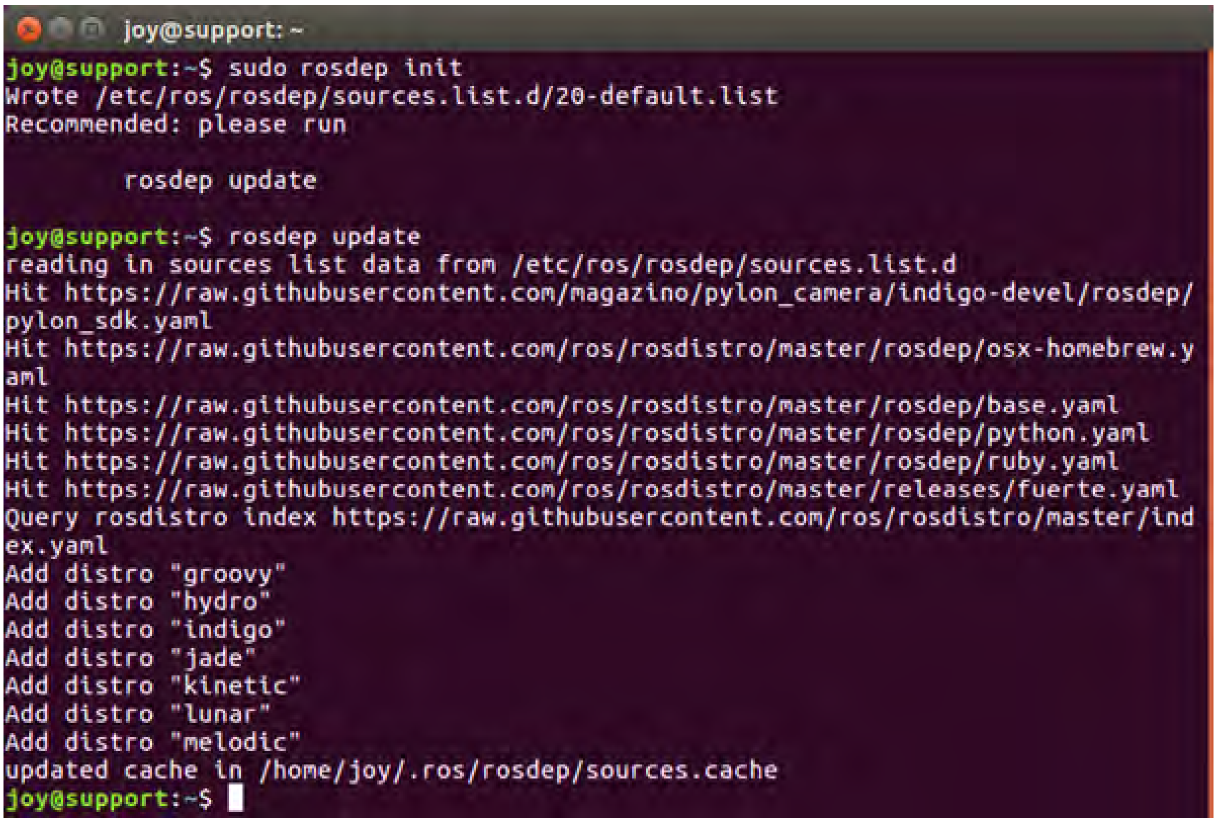
Initialization of rosdep

In the following, rosdep initialization is described for two different paths:

- A) without an error occurring
- B) with an error occurring

Do not run a rosdep update with sudo. This will later result in permission errors.

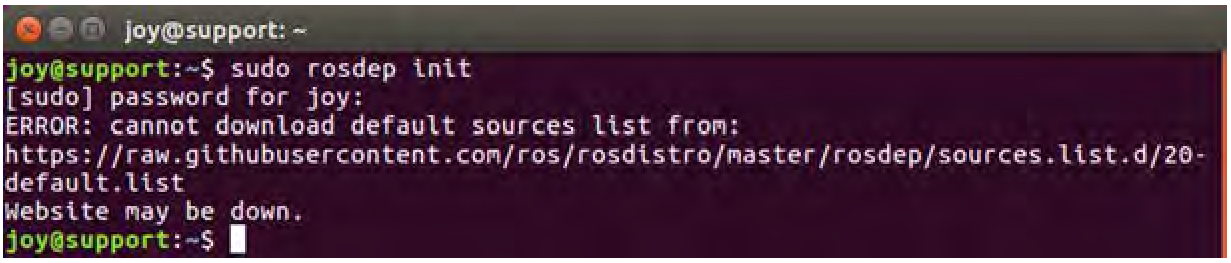
A) Initialize 'rosdep' (it is assumed that no error will occur):



```
joy@support:~$ sudo rosdep init
Wrote /etc/ros/rosdep/sources.list.d/20-default.list
Recommended: please run

        rosdep update

joy@support:~$ rosdep update
reading in sources list data from /etc/ros/rosdep/sources.list.d
Hit https://raw.githubusercontent.com/magazino/pylon_camera/indigo-devel/rosdep/
pylon_sdk.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/osx-homebrew.y
aml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/base.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/python.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/ruby.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/releases/fuerte.yaml
Query rosdistro index https://raw.githubusercontent.com/ros/rosdistro/master/ind
ex.yaml
Add distro "groovy"
Add distro "hydro"
Add distro "indigo"
Add distro "jade"
Add distro "kinetic"
Add distro "lunar"
Add distro "melodic"
updated cache in /home/joy/.ros/rosdep/sources.cache
joy@support:~$
```

```
$ sudo rosdep init

$ rosdep update
```

B) Initialize 'rosdep' (the following assumes that an error will occur):

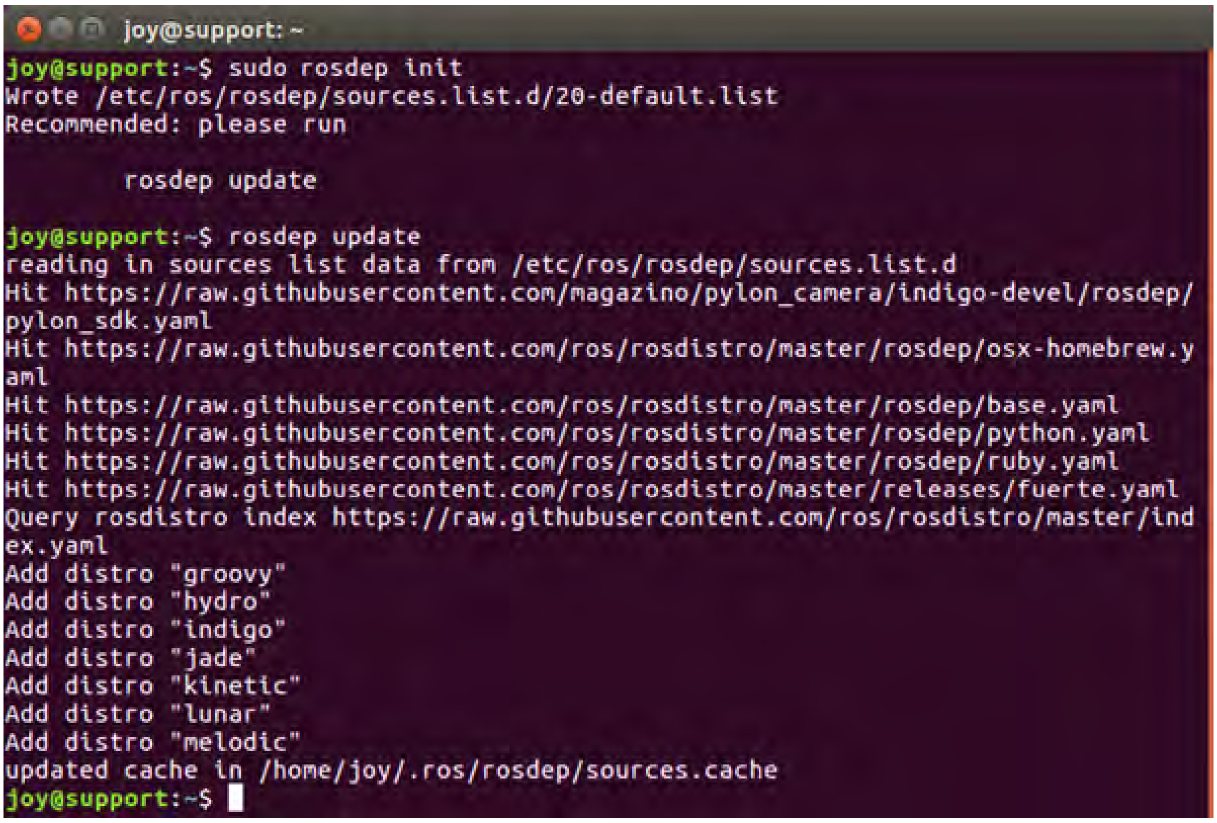


```
joy@support:~$ sudo rosdep init
[sudo] password for joy:
ERROR: cannot download default sources list from:
https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/sources.list.d/20-
default.list
Website may be down.
joy@support:~$ 
```

NOTICE

If $ rosdep init signals an error caused by problems with finding the server and if you're using a proxy server:

1. Find the correct setup of the proxy server – possibly with the help of the administrator.
2. You may try $ sudo –E rosdep init
3. Use an internet connection without involving a proxy server.

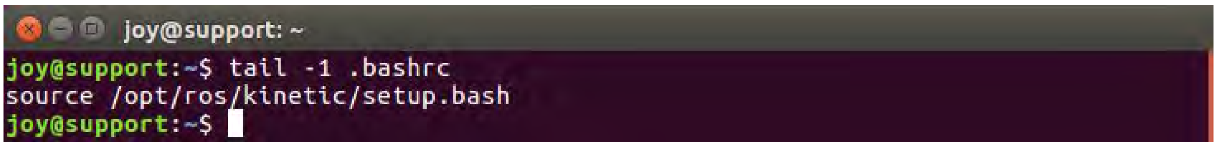


```
joy@support:~$ sudo rosdep init
Wrote /etc/ros/rosdep/sources.list.d/20-default.list
Recommended: please run

        rosdep update

joy@support:~$ rosdep update
reading in sources list data from /etc/ros/rosdep/sources.list.d
Hit https://raw.githubusercontent.com/magazino/pylon_camera/indigo-devel/rosdep/
pylon_sdk.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/osx-homebrew.y
aml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/base.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/python.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/ruby.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/releases/fuerte.yaml
Query rosdistro index https://raw.githubusercontent.com/ros/rosdistro/master/ind
ex.yaml
Add distro "groovy"
Add distro "hydro"
Add distro "indigo"
Add distro "jade"
Add distro "kinetic"
Add distro "lunar"
Add distro "melodic"
updated cache in /home/joy/.ros/rosdep/sources.cache
joy@support:~$ 
```

Establishment of Environment Settings

$ echo "source /opt/ros/kinetic/setup.bash" >> ~/.bashrc

You may check the correct establishment:

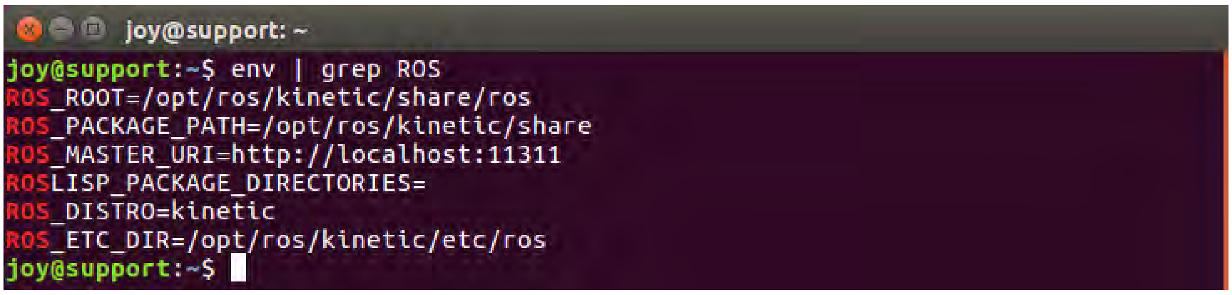


```
joy@support:~$ tail -1 .bashrc
source /opt/ros/kinetic/setup.bash
joy@support:~$
```

Source the .bashrc file to apply the modification:

$ source ~/.bashrc

You can check whether the ROS environment variables were successfully set:

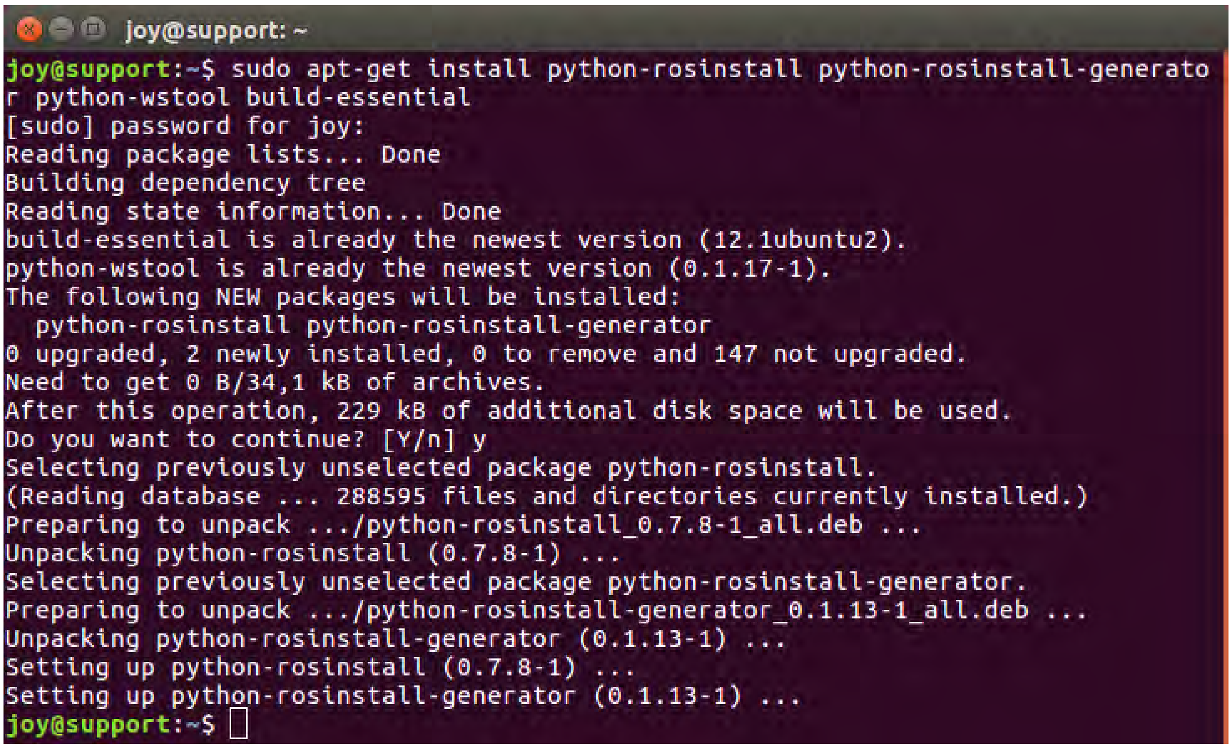


```
joy@support:~$ env | grep ROS
ROS_ROOT=/opt/ros/kinetic/share/ros
ROS_PACKAGE_PATH=/opt/ros/kinetic/share
ROS_MASTER_URI=http://localhost:11311
ROSLISP_PACKAGE_DIRECTORIES=
ROS_DISTRO=kinetic
ROS_ETC_DIR=/opt/ros/kinetic/etc/ros
joy@support:~$
```

Installation of Tools

Having completed the ROS installation, it is useful to add a couple of tools to create and manage ROS workspaces of your own. Those tools are not distributed with ROS.

Install useful tools and requirements:



```
joy@support: ~
joy@support:~$ sudo apt-get install python-rosinstall python-rosinstall-generato
r python-wstool build-essential
[sudo] password for joy:
Reading package lists... Done
Building dependency tree
Reading state information... Done
build-essential is already the newest version (12.1ubuntu2).
python-wstool is already the newest version (0.1.17-1).
The following NEW packages will be installed:
  python-rosinstall python-rosinstall-generator
0 upgraded, 2 newly installed, 0 to remove and 147 not upgraded.
Need to get 0 B/34,1 kB of archives.
After this operation, 229 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Selecting previously unselected package python-rosinstall.
(Reading database ... 288595 files and directories currently installed.)
Preparing to unpack .../python-rosinstall_0.7.8-1_all.deb ...
Unpacking python-rosinstall (0.7.8-1) ...
Selecting previously unselected package python-rosinstall-generator.
Preparing to unpack .../python-rosinstall-generator_0.1.13-1_all.deb ...
Unpacking python-rosinstall-generator (0.1.13-1) ...
Setting up python-rosinstall (0.7.8-1) ...
Setting up python-rosinstall-generator (0.1.13-1) ...
joy@support:~$
```

```
$ sudo apt-get install python-rosinstall python-rosinstall-generator python-wstool build-essential
```

2.4 Middleware Installation

The descriptions given so far do not consider the intermediary (“wrapper”) between the powerful pylon and ROS software structures. Such wrapper is usually created by the ROS developers community.

The installation of a wrapper is illustrated here using the ‘pylon_camera’ ROS driver package as the wrapper. The installation assumes that operating system, Basler pylon Camera Software Suite for Linux x86, and ROS Robot Operating System are already installed, as described above.

‘pylon_camera’ was developed by Magazino GmbH for interfacing their products. ‘pylon_camera’ is a good example for how to connect Basler cameras to ROS via the pylon API.

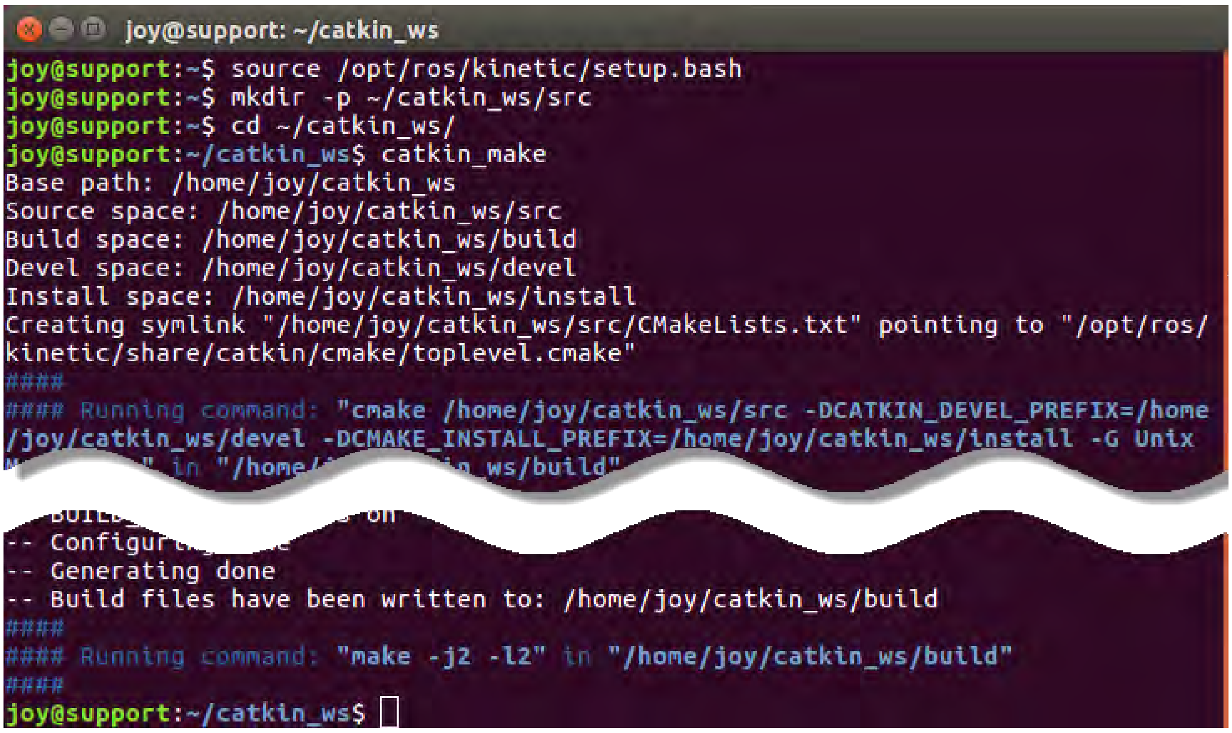
2.4.1 Details About the pylon_camera ROS Driver Package

The ‘pylon_camera’ ROS driver package provides a range of the pylon API features that allow interactive camera operation. Images are published into ROS. All recent USB3 Vision and GigE Vision Basler cameras are supported. ‘pylon_camera’ is made to meet certain application tasks. Hence, ‘pylon_camera’ is not a complete wrapper for all pylon API methods. However, adhering to the open source concept ‘pylon_camera’ can be studied, copied or modified, observing the related Copyright and the BSD license model.

2.4.2 Preparation of a ROS Build Workspace

When ROS is installed, ‘catkin’ (a build system workspace) is included that provides low level build system macros and infrastructure. catkin is necessary to build code projects like for example ‘pylon_camera’.

A workspace must be created where single or multiple packages can be built. Therefore, the following example illustrates creating subfolders ‘src’, ‘devel’ and ‘build’ for the ‘catkin_ws’ workspace folder.

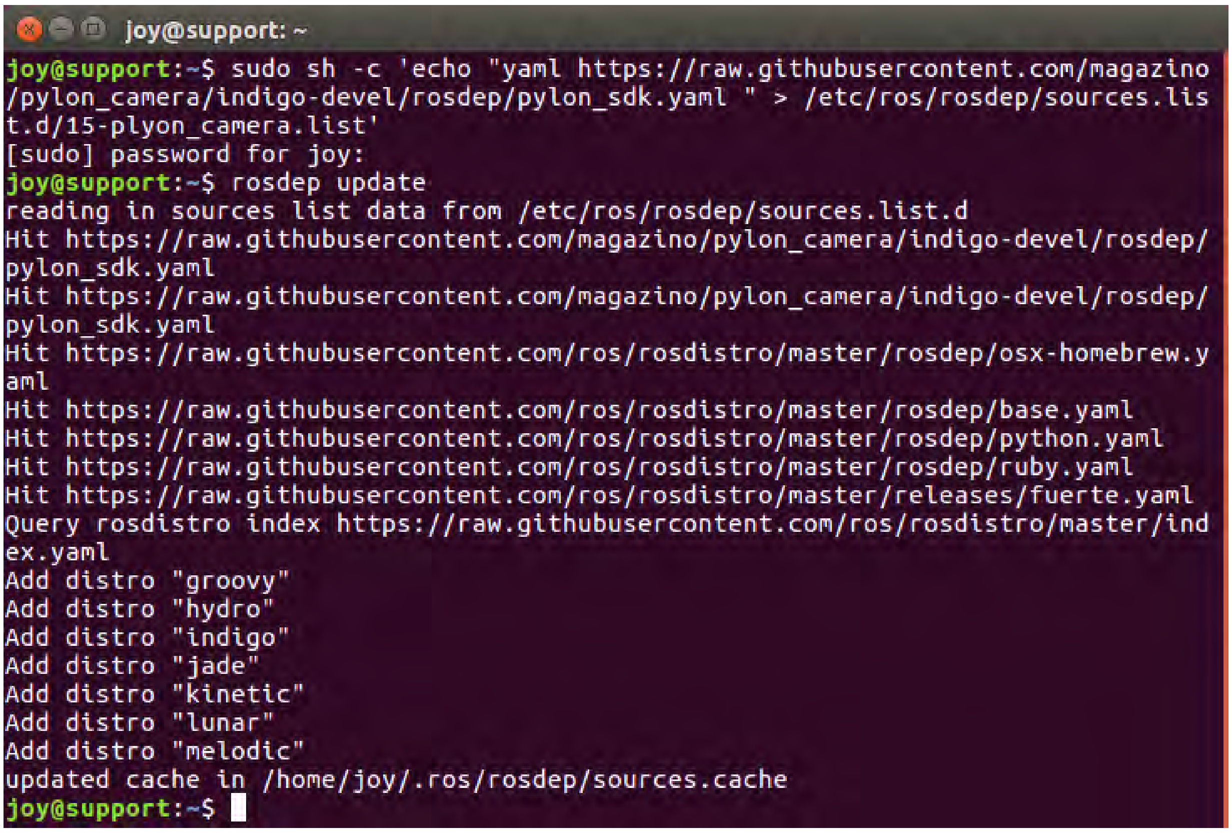

joy@support: ~/catkin_ws

```
joy@support:~$ source /opt/ros/kinetic/setup.bash
joy@support:~$ mkdir -p ~/catkin_ws/src
joy@support:~$ cd ~/catkin_ws/
joy@support:~/catkin_ws$ catkin_make
Base path: /home/joy/catkin_ws
Source space: /home/joy/catkin_ws/src
Build space: /home/joy/catkin_ws/build
Devel space: /home/joy/catkin_ws/devel
Install space: /home/joy/catkin_ws/install
Creating symlink "/home/joy/catkin_ws/src/CMakeLists.txt" pointing to "/opt/ros/
kinetic/share/catkin/cmake/toplevel.cmake"
####
#### Running command: "cmake /home/joy/catkin_ws/src -DCATKIN_DEVEL_PREFIX=/home
/joy/catkin_ws/devel -DCMAKE_INSTALL_PREFIX=/home/joy/catkin_ws/install -G Unix
N       " in "/home/        n_ws/build"

  BUILD_               on
-- Configur
-- Generating done
-- Build files have been written to: /home/joy/catkin_ws/build
####
#### Running command: "make -j2 -l2" in "/home/joy/catkin_ws/build"
####
joy@support:~/catkin_ws$ 
```

2.4.3 The Driver Employment

Make sure the pylon Camera Software Suite for Linux version is installed as described above and the PYLON_ROOT environment variable is properly set.

To install 'pylon_camera' it is just necessary to configure 'rosdep' that is the ROS command-line tool for adding system dependencies. This creates a '15-pylon_camera.list' file. This file is scanned with all current files in the same folder during the following update.

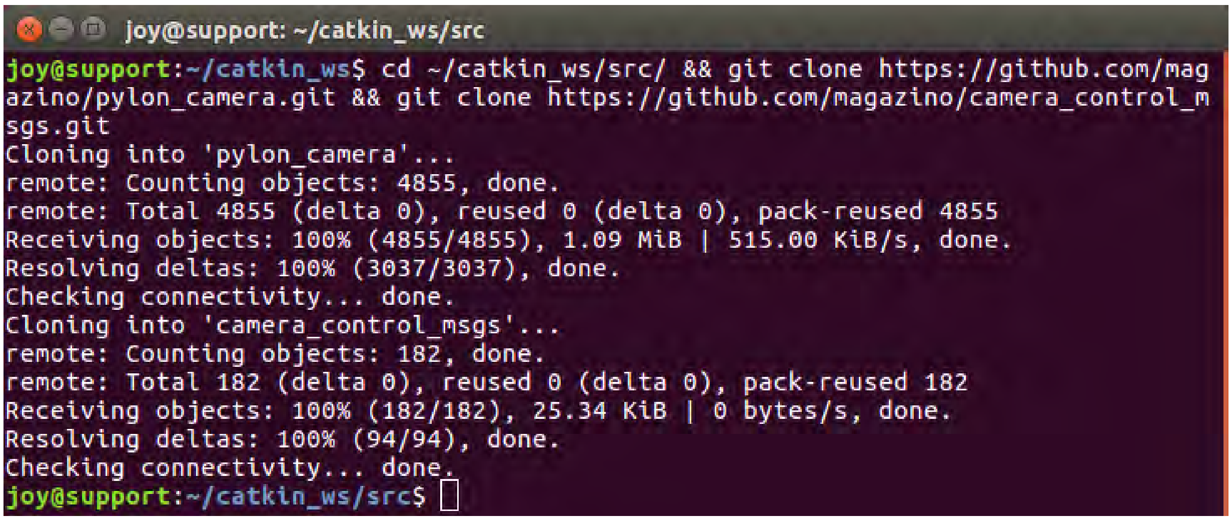


```
joy@support:~$ sudo sh -c 'echo "yaml https://raw.githubusercontent.com/magazino
/pylon_camera/indigo-devel/rosdep/pylon_sdk.yaml " > /etc/ros/rosdep/sources.lis
t.d/15-plyon_camera.list'
[sudo] password for joy:
joy@support:~$ rosdep update
reading in sources list data from /etc/ros/rosdep/sources.list.d
Hit https://raw.githubusercontent.com/magazino/pylon_camera/indigo-devel/rosdep/
pylon_sdk.yaml
Hit https://raw.githubusercontent.com/magazino/pylon_camera/indigo-devel/rosdep/
pylon_sdk.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/osx-homebrew.y
aml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/base.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/python.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/ruby.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/releases/fuerte.yaml
Query rosdistro index https://raw.githubusercontent.com/ros/rosdistro/master/ind
ex.yaml
Add distro "groovy"
Add distro "hydro"
Add distro "indigo"
Add distro "jade"
Add distro "kinetic"
Add distro "lunar"
Add distro "melodic"
updated cache in /home/joy/.ros/rosdep/sources.cache
joy@support:~$
```

$ sudo sh -c 'echo "yaml https://raw.githubusercontent.com/magazino/pylon_camera/indigo-devel/rosdep/pylon_sdk.yaml " > /etc/ros/rosdep/sources.list.d/15-plyon_camera.list'

$ rosdep update

Clone the necessary packages to the 'catkin' build system workspace:


joy@support: ~/catkin_ws/src

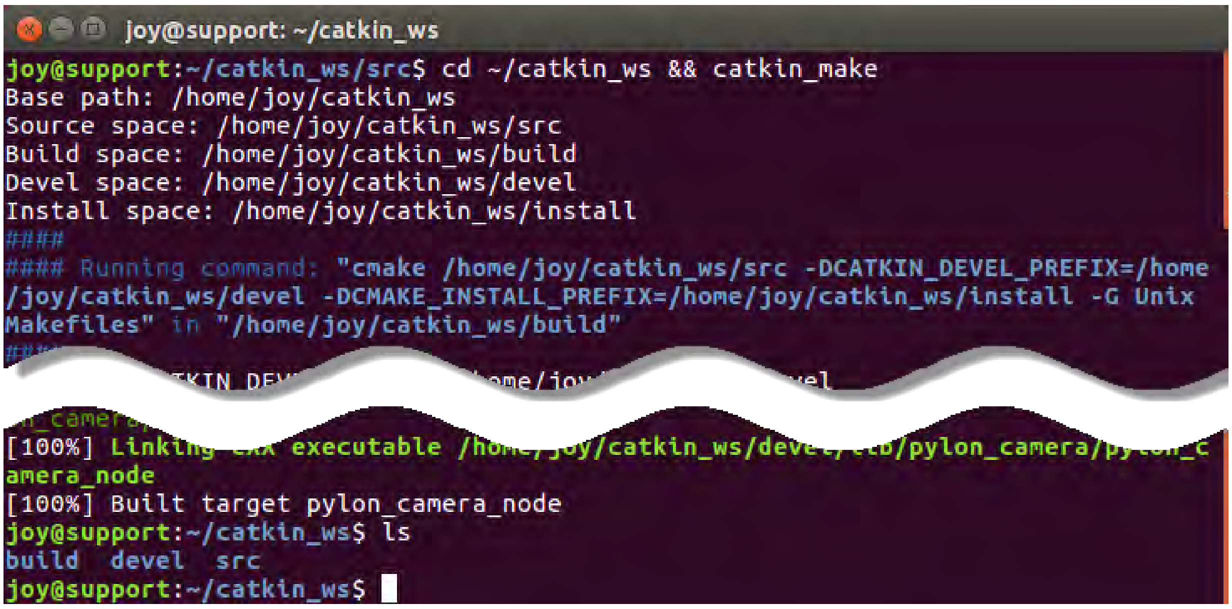
```
joy@support:~/catkin_ws$ cd ~/catkin_ws/src/ && git clone https://github.com/mag
azino/pylon_camera.git && git clone https://github.com/magazino/camera_control_m
sgs.git
Cloning into 'pylon_camera'...
remote: Counting objects: 4855, done.
remote: Total 4855 (delta 0), reused 0 (delta 0), pack-reused 4855
Receiving objects: 100% (4855/4855), 1.09 MiB | 515.00 KiB/s, done.
Resolving deltas: 100% (3037/3037), done.
Checking connectivity... done.
Cloning into 'camera_control_msgs'...
remote: Counting objects: 182, done.
remote: Total 182 (delta 0), reused 0 (delta 0), pack-reused 182
Receiving objects: 100% (182/182), 25.34 KiB | 0 bytes/s, done.
Resolving deltas: 100% (94/94), done.
Checking connectivity... done.
joy@support:~/catkin_ws/src$ 
```

$ cd ~/catkin_ws/src/ && git clone https://github.com/magazino/pylon_camera.git && git clone https://github.com/magazino/camera_control_msgs.git

Install mandatory dependencies:

```
$ rosdep install --from-paths . --ignore-src --rosdistro=$ROS_DISTRO –y
```

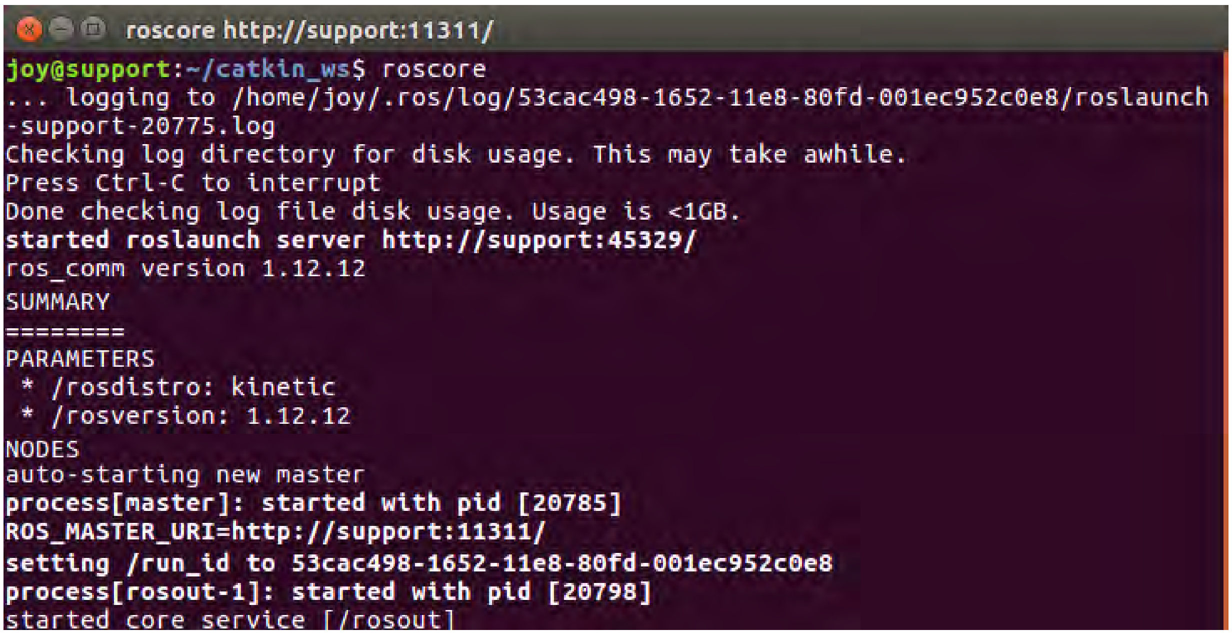
Build 'pylon_camera':


joy@support: ~/catkin_ws

```
joy@support:~/catkin_ws/src$ cd ~/catkin_ws && catkin_make
Base path: /home/joy/catkin_ws
Source space: /home/joy/catkin_ws/src
Build space: /home/joy/catkin_ws/build
Devel space: /home/joy/catkin_ws/devel
Install space: /home/joy/catkin_ws/install
####
#### Running command: "cmake /home/joy/catkin_ws/src -DCATKIN_DEVEL_PREFIX=/home
/joy/catkin_ws/devel -DCMAKE_INSTALL_PREFIX=/home/joy/catkin_ws/install -G Unix
Makefiles" in "/home/joy/catkin_ws/build"
####
      TKIN_DE            ome/jo                    el

n_camera
[100%] Linking CXX executable /home/joy/catkin_ws/devel/lib/pylon_camera/pylon_c
amera_node
[100%] Built target pylon_camera_node
joy@support:~/catkin_ws$ ls
build  devel  src
joy@support:~/catkin_ws$ 
```

$ cd ~/catkin_ws && catkin_make

Run ‘roscore’:

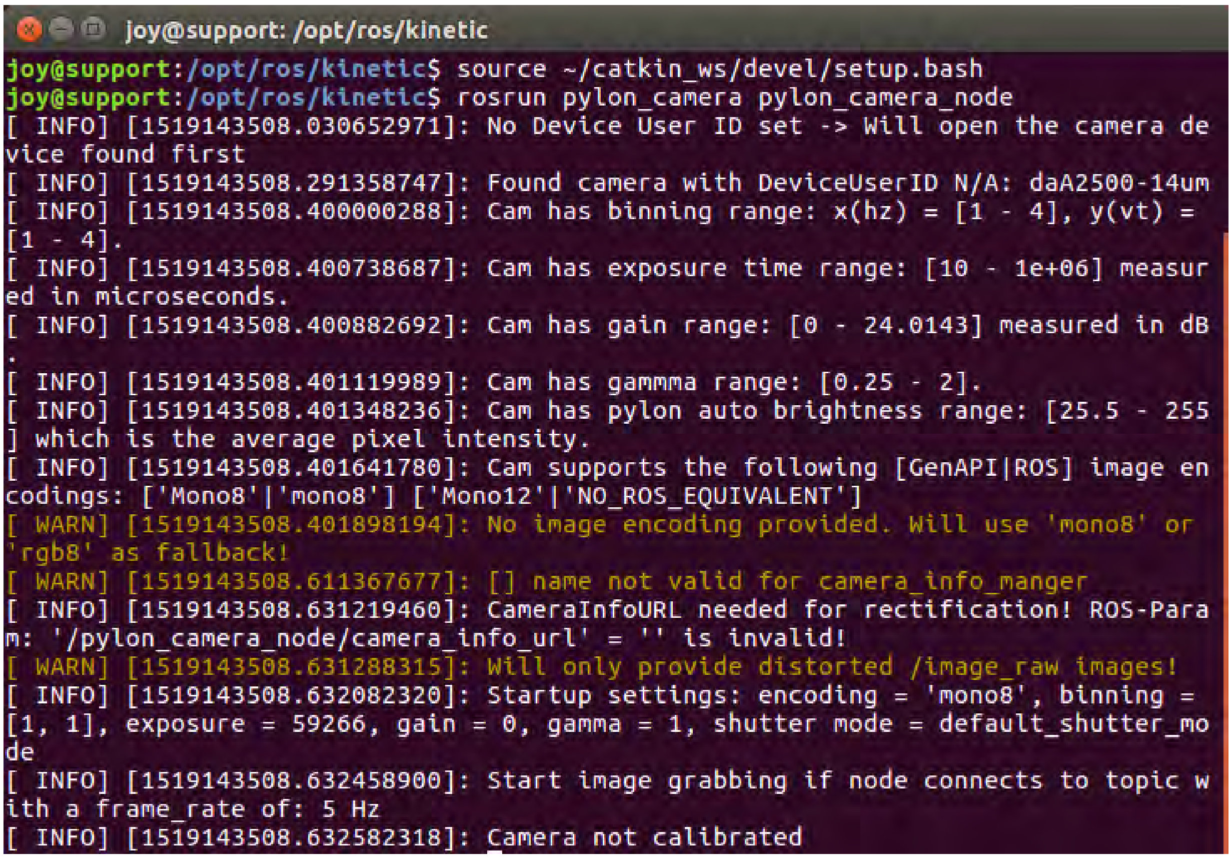


```
joy@support:~/catkin_ws$ roscore
... logging to /home/joy/.ros/log/53cac498-1652-11e8-80fd-001ec952c0e8/roslaunch
-support-20775.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.
started roslaunch server http://support:45329/
ros_comm version 1.12.12
SUMMARY
========
PARAMETERS
 * /rosdistro: kinetic
 * /rosversion: 1.12.12
NODES
auto-starting new master
process[master]: started with pid [20785]
ROS_MASTER_URI=http://support:11311/
setting /run_id to 53cac498-1652-11e8-80fd-001ec952c0e8
process[rosout-1]: started with pid [20798]
started core service [/rosout]
```

$ roscore

Open another terminal instance and run the ‘pylon_camera’ package as a node. It will occupy this newly opened terminal as well.

The workspace may still be assigned to the original default. Run source setup.bash from ‘devel’ folder to overlay the default assignment:



```
joy@support: /opt/ros/kinetic
joy@support:/opt/ros/kinetic$ source ~/catkin_ws/devel/setup.bash
joy@support:/opt/ros/kinetic$ rosrun pylon_camera pylon_camera_node
[ INFO] [1519143508.030652971]: No Device User ID set -> Will open the camera de
vice found first
[ INFO] [1519143508.291358747]: Found camera with DeviceUserID N/A: daA2500-14um
[ INFO] [1519143508.400000288]: Cam has binning range: x(hz) = [1 - 4], y(vt) =
[1 - 4].
[ INFO] [1519143508.400738687]: Cam has exposure time range: [10 - 1e+06] measur
ed in microseconds.
[ INFO] [1519143508.400882692]: Cam has gain range: [0 - 24.0143] measured in dB
.
[ INFO] [1519143508.401119989]: Cam has gammma range: [0.25 - 2].
[ INFO] [1519143508.401348236]: Cam has pylon auto brightness range: [25.5 - 255
] which is the average pixel intensity.
[ INFO] [1519143508.401641780]: Cam supports the following [GenAPI|ROS] image en
codings: ['Mono8'|'mono8'] ['Mono12'|'NO_ROS_EQUIVALENT']
[ WARN] [1519143508.401898194]: No image encoding provided. Will use 'mono8' or
'rgb8' as fallback!
[ WARN] [1519143508.611367677]: [] name not valid for camera_info_manger
[ INFO] [1519143508.631219460]: CameraInfoURL needed for rectification! ROS-Para
m: '/pylon_camera_node/camera_info_url' = '' is invalid!
[ WARN] [1519143508.631288315]: Will only provide distorted /image_raw images!
[ INFO] [1519143508.632082320]: Startup settings: encoding = 'mono8', binning =
[1, 1], exposure = 59266, gain = 0, gamma = 1, shutter mode = default_shutter_mo
de
[ INFO] [1519143508.632458900]: Start image grabbing if node connects to topic w
ith a frame_rate of: 5 Hz
[ INFO] [1519143508.632582318]: Camera not calibrated
```

$ source ~/catkin_ws/devel/setup.bash

$ rosrun pylon_camera pylon_camera_node

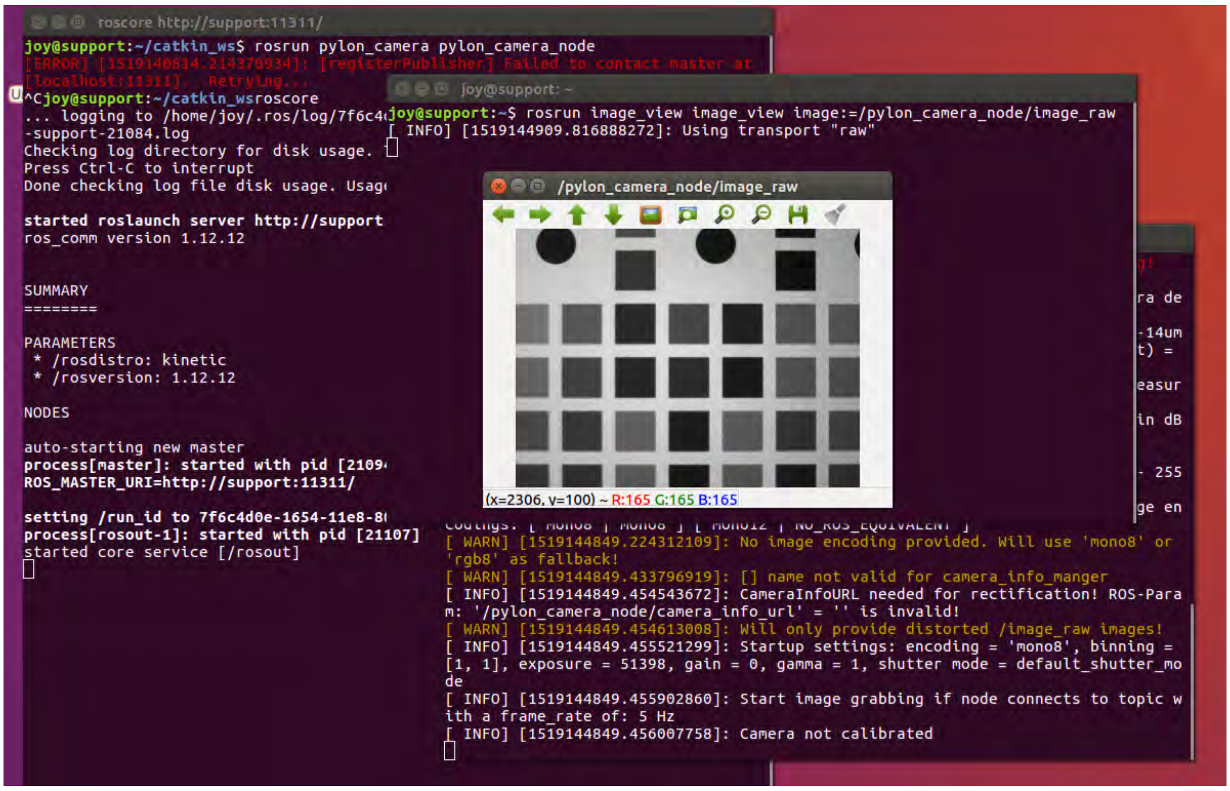
This node is now operating with the camera and provides received images via topics.

To view the images you need to use the ‘image_view’ node. This node subscribes to the provided topics. The ‘image_node’ is included in the ‘image_pipeline’ node stack.

Open a third terminal and execute the following command line:

```
$ rosrun image_view image_view image:=/pylon_camera_node/image_raw
```

An image viewer control opens where the camera's live images can be seen, zoomed, and saved.



```
joy@support:~/catkin_ws$ rosrun pylon_camera pylon_camera_node
[ERROR] [1519140814.214370934]: [registerPublisher] Failed to contact master at
[localhost:11311].  Retrying...
^Cjoy@support:~/catkin_wsroscore
... logging to /home/joy/.ros/log/7f6c4
-support-21084.log
Checking log directory for disk usage.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usag

started roslaunch server http://support
ros_comm version 1.12.12


SUMMARY
========

PARAMETERS
 * /rosdistro: kinetic
 * /rosversion: 1.12.12

NODES

auto-starting new master
process[master]: started with pid [2109
ROS_MASTER_URI=http://support:11311/

setting /run_id to 7f6c4d0e-1654-11e8-8
process[rosout-1]: started with pid [21107]
started core service [/rosout]
```

```
joy@support:~$ rosrun image_view image_view image:=/pylon_camera_node/image_raw
[ INFO] [1519144909.816888272]: Using transport "raw"
```

/pylon_camera_node/image_raw
(x=2306, v=100) ~ R:165 G:165 B:165

```
[ WARN] [1519144849.224312109]: No image encoding provided. Will use 'mono8' or
'rgb8' as fallback!
[ WARN] [1519144849.433796919]: [] name not valid for camera_info_manger
[ INFO] [1519144849.454543672]: CameraInfoURL needed for rectification! ROS-Para
m: '/pylon_camera_node/camera_info_url' = '' is invalid!
[ WARN] [1519144849.454613008]: Will only provide distorted /image_raw images!
[ INFO] [1519144849.455521299]: Startup settings: encoding = 'mono8', binning =
[1, 1], exposure = 51398, gain = 0, gamma = 1, shutter mode = default_shutter_mo
de
[ INFO] [1519144849.455902860]: Start image grabbing if node connects to topic w
ith a frame_rate of: 5 Hz
[ INFO] [1519144849.456007758]: Camera not calibrated
```

The camera interfacing is complete.

3 Driver Adjustment

ROS packages are open source projects. The ROS driver package, presented in this document serves as an example. You can program your own ROS driver package according to your needs.

To get informed of latest developments about 'pylon_camera', access the issue tracker on the 'pylon_camera' GitHub.

Revision History

Document Number	Date	Changes
AW00149101000	17 May 2018	Initial release version of this document.