

# Camera Capture Package

---

## Camera Capture Package

---

### Description:

This page discusses the *camera\_capture\_package* in detail. The package uses the Arena SDK and OpenCV libraries to stream images and allows the user to manually capture frames.

Note that requires only one camera to be connected to your machine. Edits must be made to incorporate multiple cameras!

---

### Requirements:

- Ubuntu Focal Fossa
  - ROS2 Foxy Fitzroy
  - Arena SDK for Linux
  - C++17 or higher
- 

### Configuration and Launch Files

Before using this package, make sure ALL paths in the configuration and launch files are set correctly. This will especially cause issues when cloning the repository to a new machine as the paths cloned from the remote repo are (most likely) not valid for the new local repo.

**Note:** All paths are set globally. Paths are also rarely set outside of configuration/launch files. This allows the user to point to different files in a more automated manner. Editing the configuration file DOES NOT require you to rebuild the package, but editing the launch file DOES!

#### **Configuration File:**

`camera_capture_config.yaml` is shown below:

```

src > camera_capture_package > config > ! camera_capture_config.yaml
1  #TRI028S-CC
2  camera1:
3    camera_capture:
4      ros_parameters:
5        serial: "223600392"
6        pixelformat: "rgb8"
7        width: 1280 #1440 #1280 #1920
8        height: 720 #1080 #720 #1080
9        gain: 40.0
10       exposure_time: 4000.0
11       stream_auto_negotiate_packet_size: true
12       stream_packet_resend_enable: true
13       image_save_path: "/home/tahnt/T3_Repos/camera_packages/ros2_ws/src/camera_capture_package/images/test_images"
14       image_save_name: "/test"
15       image_save_ext: ".jpg"
16
17  #TRI054S-CC
18  camera2:
19    camera_capture:
20      ros_parameters:
21        serial: "223302308"
22        pixelformat: "rgb8"
23        width: 1936 #1440 #1280 #1920
24        height: 1464 #1080 #720 #1080
25        gain: 42.0
26        exposure_time: 4000.0
27        stream_auto_negotiate_packet_size: true
28        stream_packet_resend_enable: true
29        #specify path directly to calibration images folder
30        image_save_path: "/home/tahnt/Documents/camera_calibration/ros2_ws/src/camera_calibration_package/images/TRI054S_CC"
31        image_save_name: "/calib_img"
32        image_save_ext: ".jpg"
33
34  #TRI028S-MC
35  camera3:
36    camera_capture:
37      ros_parameters:
38        serial: "223301667"
39        pixelformat: "mono8"
40        width: 1936 #1440 #1280 #1920
41        height: 1464 #1080 #720 #1080
42        gain: 0.0
43        exposure_time: 4000.0
44        stream_auto_negotiate_packet_size: true
45        stream_packet_resend_enable: true
46        #specify path directly to calibration images folder
47        image_save_path: "/home/tahnt/Documents/camera_calibration/ros2_ws/src/camera_calibration_package/images/TRI028S_MC"
48        image_save_name: "/calib_img"

```

There are three namespaces included in this configuration file (camera1, camera2, and camera3) but note that only one can be launched at a time.

#### ROS2 Parameters Configured:

- **serial:** Serial number of the camera
- **pixelformat:** Pixel format of the camera
- **width:** Desired image width resolution
- **height:** Desired image height resolution
- **gain:** Desired gain
- **exposure\_time:** Desired exposure time (4000 recommended for faster fps)
- **stream\_auto\_negotiate\_packet\_size:** A node for setting dynamic packet sizes (true recommended for faster fps)
- **stream\_packet\_resend\_enable:** Another node to handle packets - not actually sure what it does (true recommended for faster fps)
- **image\_save\_path:** Desired path to save images
- **image\_save\_name:** Desired name for saved images (each new image has the same name but is appended with an iterating count)

- **image\_save\_ext**: Desired extension for saved images

### Launch File:

`camera_capture_launch.py` is shown below:

```
src > camera_capture_package > launch > camera_capture_launch.py > generate_launch_description
1  from launch import LaunchDescription
2  from launch_ros.actions import Node
3
4  def generate_launch_description():
5
6      config_path = '/home/tahnt/T3_Repos/camera_packages/ros2_ws/src/camera_capture_package/config/camera_capture_config.yaml'
7
8      return LaunchDescription([
9          Node(
10             package="camera_capture_package",
11             executable="capture",
12             namespace="camera1",
13             name="camera_capture",
14             output="screen",
15             parameters=[config_path]
16         ),
17     ])
```

Typically, the launch file does not need to be edited often. Make sure **config\_path** correctly points to `camera_capture_config.yaml`. Under the **LaunchDescription**, the **namespace** can be changed when working with different cameras.

---

## Using The Package

This package is used to start/display an image stream and also allows the user to manually save frames. This is useful when needing the generate new calibration images, or in situations where capturing specific frames over recording an entire ROS bag is desired.

To use, follow the steps listed (also listed in the `README.md`)

### Before Use:

- Make sure **ALL PATHS ARE SET CORRECTLY** in the launch and config files before use!
- These steps assume you have already created a workspace folder and a `/src` directory within it!

### Steps:

1. Navigate into the `/src` directory of your workspace and clone the repo using `git clone`
2. Navigate back into the workspace directory and source `$ source /opt/ros/foxy/setup.bash`
3. Build package `$ colcon build` or `$ colcon build --packages-select <package_name>`
4. Open a new terminal and source it `$ . install/setup.bash`
5. Run launch file `$ ros2 launch <package_name> <launch_file_name>` in this case it is `$ ros2 launch camera_capture_package camera_capture_launch.py`

If executed correctly, the stream window should open and the terminal should output the following (or similar):

```
tahnt@pelican-glide: ~/T3_Repos/camera_packages/ros2_ws
tahnt@pelican-glide: ~/T3_Repos/camera_packages/r... x tahnt@pelican-glide: ~/T3_Repos/camera_packages/r... x
tahnt@pelican-glide:~/T3_Repos/camera_packages/ros2_ws$ ros2 launch camera_capture_package camera_capture_
launch.py
[INFO] [launch]: All log files can be found below /home/tahnt/.ros/log/2023-12-05-17-42-48-889103-pelican-
glide-62553
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [capture-1]: process started with pid [62646]
[capture-1] [INFO] [1701826969.388257573] [camera1.camera_capture]:
[capture-1] Device 1:
[capture-1] Lucid Vision Labs
[capture-1] Model: TRI028S-C
[capture-1] Serial: 223600392
[capture-1] Mac: 1c:0f:af:04:33:5b
[capture-1] IP: 169.254.92.51
[capture-1] [INFO] [1701826970.044284524] [camera1.camera_capture]: Device created
[capture-1] [INFO] [1701826970.130124213] [camera1.camera_capture]: Default profile is loaded
[capture-1] [INFO] [1701826970.130302144] [camera1.camera_capture]: Setting nodes ...
[capture-1] [INFO] [1701826970.239033200] [camera1.camera_capture]: Image settings:
[capture-1] [INFO] [1701826970.239192843] [camera1.camera_capture]: pixelformat: rgb8
[capture-1] [INFO] [1701826970.239260724] [camera1.camera_capture]: width: 1280
[capture-1] [INFO] [1701826970.239312537] [camera1.camera_capture]: height: 720
[capture-1] [INFO] [1701826970.239359462] [camera1.camera_capture]: OffsetX: 328
[capture-1] [INFO] [1701826970.239406606] [camera1.camera_capture]: OffsetY: 372
[capture-1] [INFO] [1701826970.239449848] [camera1.camera_capture]: gain: 40.000000
[capture-1] [INFO] [1701826970.239509753] [camera1.camera_capture]: exposure_time: 4000.000000
[capture-1] [INFO] [1701826970.239570853] [camera1.camera_capture]: stream_auto_negotiate_packet_size: 1
[capture-1] [INFO] [1701826970.239622533] [camera1.camera_capture]: stream_packet_resend_enable: 1
[capture-1] [INFO] [1701826971.040468250] [camera1.camera_capture]: Streaming with OpenCV ...
[capture-1] press 's' to save image
[capture-1] press 'esc' to exit
```

The terminal displays the enumerated device info and the camera node values set in the config file, and prompts the user to begin saving images by pressing 's' or exit by pressing 'esc'.