



Program Listing for File v4l2_video_capture.hpp

[Return to documentation for file \(](#)

`include/holoscan/operators/v4l2_video_capture/v4l2_video_capture.hpp`)

```
/* * SPDX-FileCopyrightText: Copyright (c) 2023-2024 NVIDIA CORPORATION &
AFFILIATES. All rights reserved. * SPDX-License-Identifier: Apache-2.0 * * Licensed
under the Apache License, Version 2.0 (the "License"); * you may not use this file
except in compliance with the License. * You may obtain a copy of the License at * *
http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law
or agreed to in writing, software * distributed under the License is distributed on an
"AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express or implied. * See the License for the specific language governing
permissions and * limitations under the License. */ #ifndef
HOLOSCAN_OPERATORS_V4L2_VIDEO_CAPTURE_HPP #define
HOLOSCAN_OPERATORS_V4L2_VIDEO_CAPTURE_HPP #include <linux/videodev2.h>
#include <memory> #include <string> #include "holoscan/core/operator.hpp"
namespace holoscan::ops { class V4L2VideoCaptureOp : public Operator { public:
HOLOSCAN_OPERATOR_FORWARD_ARGS(V4L2VideoCaptureOp)
V4L2VideoCaptureOp() = default; void setup(OperatorSpec& spec) override; void
start() override; void initialize() override; void compute(InputContext& op_input,
OutputContext& op_output, ExecutionContext& context) override; void stop()
override; private: Parameter<holoscan::IOSpec*> signal_;
Parameter<std::shared_ptr<Allocator>> allocator_; Parameter<std::string> device_;
Parameter<uint32_t> width_; Parameter<uint32_t> height_; Parameter<uint32_t>
num_buffers_; Parameter<std::string> pixel_format_; Parameter<uint32_t>
exposure_time_; Parameter<uint32_t> gain_; void v4l2_initialize(); void
v4l2_requestbuffers(); void v4l2_check_formats(); void v4l2_set_mode(); void
v4l2_set_formats(); bool v4l2_camera_supports_control(int cid, const char*
control_name); void v4l2_set_camera_control(v4l2_control control, const char*
control_name, bool warn); void v4l2_set_camera_settings(); void v4l2_start(); void
v4l2_read_buffer(v4l2_buffer& buf); void YUYVToRGBA(const void* yuyv, void* rgba,
size_t width, size_t height); void MJPEGToRGBA(const void* mjpg, void* rgba, size_t
width, size_t height); struct Buffer { void* ptr; size_t length; }; Buffer* buffers_; int
fd_ = -1; uint32_t width_use_; uint32_t height_use_; uint32_t pixel_format_use_; }; } //
```

```
namespace holoscan::ops #endif/*
```

```
HOLOSCAN_OPERATORS_V4L2_VIDEO_CAPTURE_HPP */
```

© Copyright 2022-2024, NVIDIA.. PDF Generated on 06/06/2024