IntelRealSense / **librealsense**

Cross-platform API for Intel® RealSense™ devices https://communities.intel.com/communi…

#librealsense #developer-kits #sdk #library #camera-api #computer-vision #hardware

**2,463** commits | **4** branches | **25** releases | **57** contributors | Apache-2.0

Branch: **master** ▾ | **New pull request** | **Find file** | **Clone or download** ▾

**dorodnic** committed on **GitHub** Merge pull request #575 from ev-mp/fourcc_fix … Latest commit `66e4206` 24 days ago

| | | |
|---|---|---|
| .github | Adding links to RealSense community and support sites | 3 months ago |
| CMake | clean up on frame metadata | 11 months ago |
| config | scripts: removed bashisms. | 7 months ago |
| doc | Fixed table formatting | 2 months ago |
| examples | examples: control building of the graphical examples. | 7 months ago |
| include/librealsense | Adjust SR300 4CC codes for INVZ and INVI formats | 28 days ago |
| librealsense.qtcreator | Provide DSInfo; Enable symbols generation; Handle klockwork issues; F… | a year ago |
| librealsense.vc12 | Replacing SolutionDir macro with relative paths #468 | 5 months ago |
| librealsense.vc14 | Replacing SolutionDir macro with relative paths #468 | 5 months ago |
| librealsense.xc | The hard coded XCode project should be removed now that CMake works c… | 11 months ago |
| scripts | Adjust patched modules generation with the local vermagic for LTS ker… | a month ago |
| src | Fix FourCC parsing logic for windows backend | 24 days ago |
| unit-tests | adding case for version change in patch | a year ago |
| .gitignore | Add CMake's \`build/\` directory to .gitignore, update .def file CMake … | a year ago |
| .travis.yml | Update Travis Build for CMake | a year ago |
| AUTHORS | updating authors | a year ago |
| CLA.md | Bring CLA.md, CONTRIBUTING.md, and ISSUE_TEMPLATE.md into master branch. | 2 years ago |
| CMakeLists.txt | Generate cmake package config files | 7 months ago |
| CONTRIBUTING.md | proper naming | a year ago |
| COPYING | Replace stb_truetype and the Roboto Bold font with stb_easy_font. | 2 years ago |
| LICENSE | change license to Apache 2.0 | 2 years ago |
| appveyor.yml | initial untested appveyor config | 2 years ago |
| cmake_uninstall.cmake | Modify cmake uninstall for robustness | 10 months ago |
| package.xml | Update list of maintainers | a month ago |
| readme.md | Updating AppVeyor badge status | 24 days ago |
| realsenseConfig.cmake.in | Generate cmake package config files | 7 months ago |

**readme.md**

# Intel® RealSense™ Cross Platform API

| Platform | Build Status |
|---|---|
| Linux and OS X | build passing |
| Windows | build passing |

This project is a cross-platform library (Linux, Windows, Mac) for capturing data from the Intel® RealSense™ F200, SR300, R200, LR200 and the ZR300 cameras. This effort was initiated to better support researchers, creative coders, and app developers in domains such as robotics, virtual reality, and the internet of things. Several often-requested features of RealSense™ devices are implemented in this project, including multi-camera capture.

Developer kits containing the necessary hardware to use this library are available for purchase at this link. This project is separate from the production software stack available in the Intel® RealSense™ SDK, namely that this library only encompasses camera capture functionality without additional computer vision algorithms.

The Intel® RealSense™ Cross Platform API is experimental and not an official Intel product. It is subject to incompatible API changes in future updates.

## Table of Contents

## Useful Links

- Intel RealSense Community - Official support, Q&A and other useful content
- Support Site - Contains content and web ticket capability for 1:1 interaction
- SDK Design Guidelines - Guidelines and tips for designing applications using RealSense cameras
- R200 Datasheet - In-depth information about the R200 camera hardware.
- Intel RealSense Stereoscopic Depth Cameras - A technical paper describing the R200, LR200, SR300 and RS400 in detail. Includes theoretical background, performance expectations, post-processing suggestions, etc.

## Documentation

A comprehensive suite of sample and tutorial applications are provided in the `/examples` subdirectory. For new users, it is best to review the tutorial series of apps which are designed to progressively introduce API features.

- C API - With doxygen-style API comments
- To build documentation locally from sources, on Ubuntu run the following commands:
  - `sudo apt-get install doxygen`
  - `doxygen doc/Doxygen_API/Doxyfile`
- What's New?
- Projection APIs - A guide on coordinate systems, calibration information, and projection
- Camera Spec Sheet - A brief overview of R200, F200 and SR300
- Developer Notes - Several informal notes gathered during internal releases
- OpenCV Tutorial - Getting started with librealsense using OpenCV

- Stream Formats - A list of available stream resolutions and pixel formats provided by the supported devices.
- Branching and Releases - Overview of live branches and major releases

## Functionality

1. Native streams: depth, color, infrared and fisheye.
2. Synthetic streams: rectified images, depth aligned to color and vice versa, etc.
3. Intrinsic/extrinsic calibration information.
4. Majority of hardware-specific functionality for individual camera generations (UVC XU controls).
5. Multi-camera capture across heterogeneous camera architectures (e.g. mix R200 and F200 in same application)
6. Motion-tracking sensors acquisition (ZR300 only)

## Compatible Devices

1. RealSense R200
2. RealSense F200
3. RealSense SR300
4. RealSense LR200
5. RealSense ZR300

## Compatible Platforms

The library is written in standards-conforming C++11 and relies only on the C89 ABI for its public interface. It is developed and tested on the following platforms:

1. Ubuntu 14.04 and 16.04 LTS (GCC 4.9 toolchain)
2. Windows 8.1 and Windows 10 (Visual Studio 2015 Update 2)
3. Mac OS X 10.7+ (Clang toolchain)
4. Ostro

## Hardware Requirements

Developer kits containing the necessary hardware to use this library are available for purchase at this link. In addition, several consumer tablets and laptops with integrated cameras may also function, such as the HP Spectre x2 with R200.

Developer kits **require** USB 3.0. RealSense™ cameras do not provide backwards compatibility with USB 2.0. Not all USB host chipsets are compatible with this library, although it has been validated with recent generations of the Intel Host Controller chipset. An exhaustive list of incompatible hardware is not presently provided. On x86, a Haswell or newer architecture is recommended.

For small-form factor usages, this library has been demonstrated to work on the following boards:

- Intel Compute Stick, BOXSTK1AW32SCR
- MinnowBoard Max
- Kangaroo MD2B
- UP Board
- Intel Joule

## Integrations

The library has been integrated with a number of third-party components and operating systems. While most of these projects are not directly supported by the team, they are useful resources for users of this library.

- Robotic Operating System (Intel Supported; R200, F200, SR300 all supported)
- Yocto / WindRiver Linux
- Arch Linux

## Community Contributions

Additional language bindings (experimental, community maintained):

- Python
- Java (generated by JavaCPP)

Excellent blog-series by @teknotus covering the fundamentals of working with RealSense on Linux:

- Intel RealSense camera on Linux
- 3d Camera Controls
- Infrared, calibration, point clouds
- The long road to ubiquitous 3d cameras

## License