

ADUVC - AN EPICS AREADETECTOR DRIVER FOR USB VIDEO CLASS DEVICES

J. Wlodek¹, K. J. Gofron[†], Brookhaven National Laboratory, Upton, USA
¹also at Stony Brook University, Stony Brook, USA

Abstract

Most devices supported by the Experimental Physics and Industrial Control System [1] (EPICS) areaDetector [2] project fall under one of two categories: detectors and cameras. Many of the cameras in this group can be classified as industrial cameras, and allow for fine control of exposure time, gain, framerate, and many other image acquisition parameters. This flexibility can come at a cost; however, with most such industrial cameras' prices starting near one thousand dollars, with the price rising for cameras with more features and better hardware. While these prices are justified for situations that require a large amount of control over the camera, for monitoring tasks, and some basic data acquisition, the use of consumer devices may be sufficient while being less cost-prohibitive. The solution we developed was to write an areaDetector driver for USB Video Class (UVC) [3] devices. This driver allows for a variety of cameras and webcams to be used through EPICS and areaDetector, with most costing under \$100.

INTRODUCTION

The EPICS areaDetector project enables universal camera and detector control for EPICS installations through the use of camera-specific driver implementations that all share an underlying base, called ADCore. Currently supported devices include many models of X-ray detectors which are used extensively for data collection, as well as a large array of industrial cameras such as Prosilica (Vimba) [4] devices, Flir (Point Grey) [5] devices, and more. These industrial cameras are used for some data collection, but are primarily used for monitoring purposes, allowing scientists to see a live feed of their experimental stage and experiment itself during its duration.

These industrial cameras often have expansive feature sets with various image acquisition and processing modes that allow for fine-tuned control of the feed shown to the scientists. This level of customizability is important in several use-cases (particularly for taking images with different exposure times); however, in some cases it may be unnecessary. In such situations, a cheaper more flexible alternative to these expensive industrial cameras would perhaps be more suitable.

To solve this problem, we looked for a set of cameras that were inexpensive, used a standardized communication interface, and had a readily available C/C++ library that could be integrated into an areaDetector driver. Taking into account these criteria, we settled on the UVC standard that is used for most consumer class webcams and USB video

devices. This standard encompasses a vast number of supported cameras ranging from under \$20 webcams to several hundred dollar industrial cameras that support many of the same features allowed by the more traditional ones already included with areaDetector. In this paper we will discuss how we added UVC device support to EPICS areaDetector, the construction of the ADUVC driver [6], and some supported cameras and deployment results and challenges.

THE UVC STANDARD

The UVC standard is utilized almost universally among USB cameras and video devices, most notably in consumer webcams. Most operating systems that allow for viewing camera images via a dedicated application (i.e. the Windows 10 "Camera" application) are internally using a custom UVC driver for camera control. The UVC standard focuses on defining camera settings and variables common to most cameras and abstracting them away with a high-level communication protocol that can be reused between most devices, much like how ADCore contains functionality common to all areaDetector cameras. This universality allows for UVC to be used to control and use a huge number of devices with varied output resolutions, framerates, and image formats. UVC enforces certain requirements that must be satisfied by the firmware of each device:

- Each UVC device must have a series of predefined formats within firmware that can be read according to the standard, supplying all required image information to the driver prior to streaming initialization, including the data type of the output, and the color mode of the resulting image (Table 1).
- Each UVC device must support standard commands sent over the USB interface for starting and stopping streaming.
- Each UVC device must send frames back to the driver in a standardized format.

Table 1: Some Supported UVC Frame Formats

Format	Data Type	Color Mode
MJPEG	UInt8	RGB
RGB	UInt8	RGB
Grayscale	UInt8/UInt16	Mono
YUYV	UInt8	RGB/Mono
Uncompressed	UInt8/UInt16	RGB/Mono

[†] kgofron@bnl.gov

The UVC protocol also contains standard commands for setting various camera parameters like saturation, brightness, sharpness, etc. One limitation of UVC and the devices that support it, is that not all devices are required to support all functions, meaning that in some cases a device will simply not react in any way to receiving a signal to edit a particular camera parameter. This issue is particularly prominent among the very inexpensive devices that often support a bare minimum implementation of the UVC standard, which can cause problems with usage. In many cases these trade-offs are worthwhile, as the cost of these cameras is so low that additional camera functionality is often not necessary.

LIBUVC

Prior to any EPICS integration being created for UVC devices, it was important to identify a readily available library that allows for communication with UVC devices using the standard's predefined protocols. The identified solution was libuvc [7], an open source implementation of said protocol that is available for public use under the BSD License.

The two key libraries used by libuvc to allow for communication with UVC devices are libusb [8], an open source USB connectivity library that is used universally for projects that involve USB devices, and libjpeg [9], an open source library that is used for jpeg compression/decompression. Libusb is used to manage the USB connection to the camera, and to send the standard UVC protocol commands for controlling the camera. When libuvc first connects to a device, it uses libusb to acquire a handle on the camera, locking it from being used by any other programs.

After control of the camera has been acquired, libuvc establishes supported acquisition modes, as defined in the UVC standard. These are the modes that are guaranteed to return frames from the camera when selected.

Next, once the user selects an acquisition mode and selects "Acquire", libuvc creates an image stream from the camera. It uses the device handle created by libusb, and ties it to a callback function that is fired every time a frame is received from the device. Within the callback function the user can do whatever processing he needs before the frame is freed, and the next frame is acquired from the camera. Libuvc also supports most of the non-required UVC camera parameters (saturation, brightness, etc.).

ADUVC

Once libuvc was identified as the solution for communicating with UVC devices, an areaDetector driver was written that built an EPICS shell around libuvc, allowing users to call library functions through write requests on EPICS process variables (PVs). The driver is based on other areaDetector drivers, most notably the ADProsilica and ADPointGrey drivers. It uses the previously mentioned callback function defined during the libuvc start streaming call to grab the frames from the UVC device, convert them into the areaDetector NDArray standard, and pass them back up the chain to the EPICS layer. All other supported camera parameters in libuvc are represented internally in

the driver as PVs, and, when writing a new value, the appropriate libuvc function is fired sending the new value for the appropriate parameter to the camera.

The driver also uses the pre-defined operating modes required by the UVC standard to provide the user with a drop-down menu listing selectable operating modes in the EPICS front end. This was an important addition in order to give users the ability to quickly see which modes are available to them, as unlike other more expensive industrial cameras that support almost any image size, data format, and framerate, UVC cameras are locked into these predefined formats. In addition, the driver has an auto-adjust function that attempts to match EPICS image format PVs to the image format received from the camera. This was added to the driver when we found that certain modes supported multiple data types and color modes, and thus made selecting valid acquisition values difficult.

As with all other areaDetector drivers, ADUVC is fully interoperable with all areaDetector plugins, as once the camera frame is converted into an NDArray the entire software stack does not depend on the UVC standard.

For connecting an ADUVC driver input output controller (IOC) to a device, the serial number or the product id code can be used. In some cases, particularly among very inexpensive devices from untested manufacturers, the serial number of the device cannot be read, and the product number must be used. No two devices with the same product number may be connected at the same time, as this will cause the ADUVC driver to attempt to connect twice to a single camera. Each IOC requires sole access to its assigned device, though as long as each device uses a different serial/product number to connect, as many devices as desired may be connected to a single host server.

Table 2: Some Tested UVC Camera Specifications and Costs

UVC Device	Max Resolution/Framerate	Cost with lens
E-Con Systems See 3 Cam	2592 x 1944 @ 60 fps	\$300
Webcamera IMX322	1920 x 1080 @ 30+fps	\$75
Opti-Tekscope microscope camera	1600 x 1200 @ 30 fps	\$28
Fantronics Snake Camera	1200 x 900 @ 30 fps	\$20

RESULTS

Using ADUVC, we have deployed a number of devices with varied roles in beamline operation. Most devices deployed were used primarily in a monitoring role with their low cost making them a strong alternative for this use case. Certain cameras have also been tested as data collection and beam viewing devices at the XF:17BM beamline [10] at NSLS2. Among the devices tested with the driver, we

found a huge variety in form factors, some of which have already proved very useful for experiments conducted by the beamlines. With costs ranging from \$20 to \$300 at the higher end of cameras tested, the price to performance ratio has thus far proven impressive (Table 2).

E-Con Systems See 3-Cam

The most expensive camera tested, at \$300, the E-Con Systems See 3-Cam also gave the most impressive results. Almost all UVC defined standard camera parameters are supported, along with high resolution and framerate acquisition up to 5MP and 60 frames per second, depending on the acquisition mode. The camera was also able to acquire in both 16 bit Mono and 8 bit RGB modes, both uncompressed. Being a board camera, we have created a 3D-printed enclosure for the device, and it is now mounted at the beamline as a primary monitoring camera (Fig. 1).

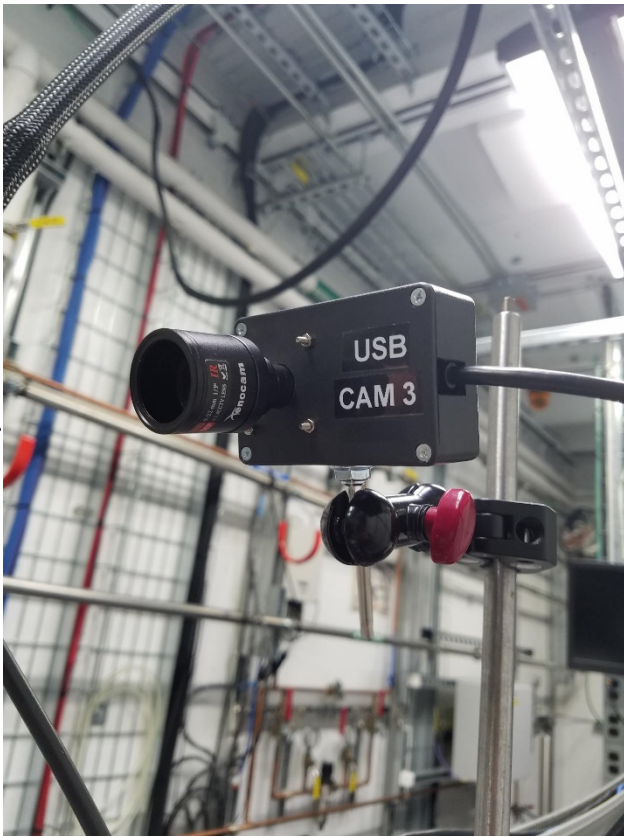


Figure 1: E-Con Systems See 3-Cam mounted at the XF:17BM beamline at NLS2 in the custom 3D printed enclosure.

Webcamera IMX322

Perhaps the second most impressive device we tested was the Webcamera IMX322, so named because of its use of the IMX322 camera sensor from Sony. At only \$75 on Amazon.com with an included lens, metal casing, and standard mount points, we tested this camera as a possible replacement for Prosilica cameras for viewing the beam (Figs. 2 & 3). The device supports images of a resolution of 1920 x 1080 at 30 frames per second, though it does not support as many UVC parameters as the E-Con Systems

device. We found that the hue, backlight compensation, and saturation values were not affected when writing to the matching PVs in the EPICS front end.

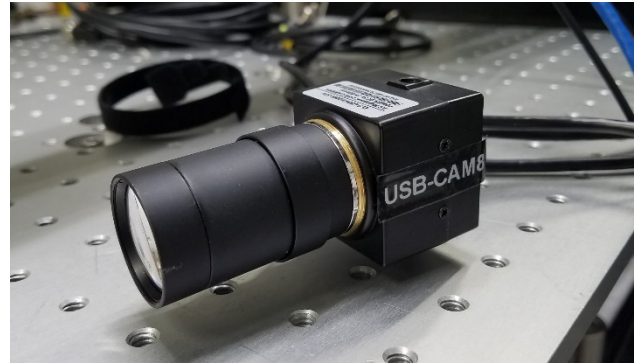


Figure 2: Webcamera IMX322 at the XF:17BM beamline, with included lens and metal casing.

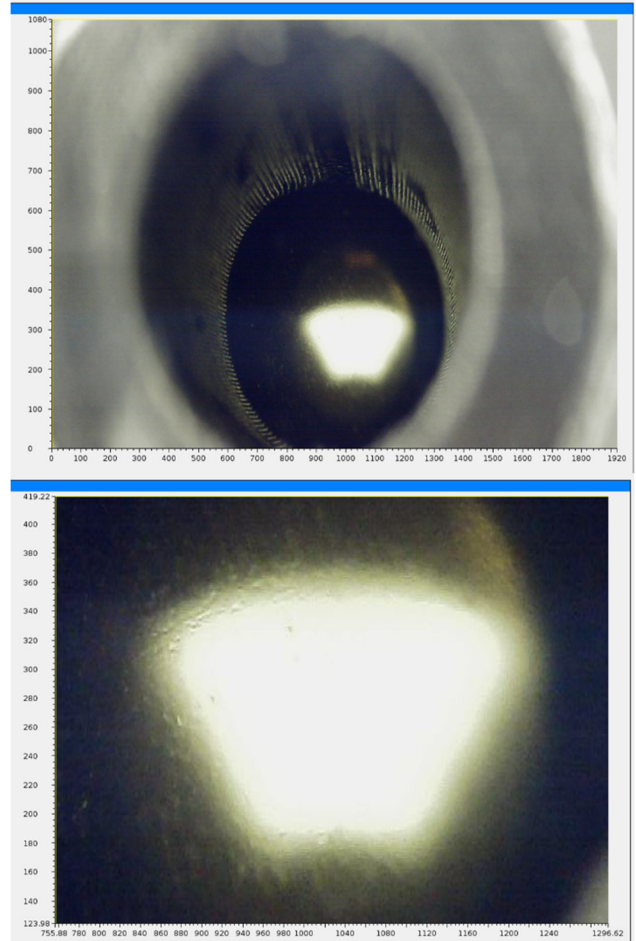


Figure 3: First images of the beam captured at XF:17BM with the IMX322 camera. The focus still needs adjustment, but as a proof of concept, the camera was functional in this use case.

Opti-Tekscope Microscope Camera

The Opti-Tekscope Microscope Camera is a pencil camera we tested with the ADUVC driver, currently available for \$28 from Amazon.com. This device was one for which we were unable to read the serial number, meaning that we

were unable to run multiple such devices on one server. We found the pencil-cam form factor was useful for monitoring areas that were too narrow or inaccessible for other devices (Fig. 4). In addition, using a simple clamp we were able to mount it at a high elevation and because of its infinite focal distance (auto-focus) we were able to have a monitoring device for the entire hutch.



Figure 4: The Opti-Tekscope Pencil Camera.

FANTRONICS SNAKE CAM

The most interesting camera form-factor wise that we tested was the Fantronics Snake Cam, which is essentially a tiny cell-phone style camera on the end of a stiff wire that can be bent into any shape required by the user. This allowed us to easily position the camera where the users wanted a set of eyes, a flexibility that proved useful time and again. The device is currently in production on the beamline, being used based on the requirements of each user.

Other Cameras

Aside from the ones listed above, we have tested ADUVC with several other cameras thus far. Each device varies in performance and cost, and each has found a use at the beamline. The variety of form factors allowed by UVC devices give the ADUVC driver a flexibility of use cases not often found with other camera standards. Some other cameras we tested included another board camera from E-Con Systems that allowed for a 90-degree angle viewport, a microscope camera with a very shallow depth of focus for magnification of samples, among others.

Issues Experienced

There were several issues we experienced and solved when testing the ADUVC areaDetector driver on the beamline.

- UVC devices may require root access by default to be accessed via libuvc. This can be circumvented by adding udev rules to your host machine.
- Some UVC devices do not seem to have a readable serial number. In this case use the product ID to connect.
- Two UVC devices with the same product ID cannot be running on one host machine at the same time.
- If using a USB hub, there are some limitations with running certain cameras at the same time. The extent of these has not yet been looked at; however, it seems this is only an issue with the very inexpensive devices.
- Not all UVC devices seem to support all the features provided by the UVC standard (saturation, brightness, sharpness, etc.). Again, this seems to be an issue with the particularly inexpensive devices.

CONCLUSION

While in some cases a highly specialized industrial camera with fine control of camera parameters is required for scientific experiments, in many instances these expensive devices can be substituted for a much less costly consumer class device. The versatility of UVC cameras, along with the vast amount of supported form factors give them unprecedented flexibility in use cases. With just the ADUVC driver, users may control an industrial-class device like the See 3-Cam, while also controlling a \$20 Snake Camera that can be used to easily to look at a sample from any angle or position.

ACKNOWLEDGMENT

We would like to acknowledge the XF:17BM and XF:10ID beamline staffs for their willingness to adopt the ADUVC project and to use certain supported cameras in production on the beamlines.

This research used the resources of the 17BM beamline at the National Synchrotron Light Source II, a U.S. Department of Energy (DOE) Office of Science User Facility operated for the DOE Office of Science by Brookhaven National Laboratory under Contract No. DE-SC0012704.

APPENDIX

Below is a list of cameras that were used for testing the ADUVC driver, and links from where each device was purchased.

- E-Con Systems See 3-Cam: <https://www.e-consys.com/5mp-low-noise-usb-camera.asp>
- Webcam w/ Sony IMX322 Camera Sensor: https://www.amazon.com/gp/product/B07M7JN595/ref=ppx_yo_dt_b_asin_title_002_s00?ie=UTF8&psc=1
- Opti-Tekscope Pencil Microscope Camera: https://www.amazon.com/gp/product/B0184CC0Y0/ref=ppx_yo_dt_b_asin_title_006_s01?ie=UTF8&psc=1

- Fantronics Flexible Snake Camera: https://www.amazon.com/gp/product/B071HYRPND/ref=ppx_yo_dt_b_asin_title_o09_s00?ie=UTF8&psc=1
- E-Con Systems CAMCU50 Board Camera: <https://www.e-consystems.com/M12-CMOS-USB-3-Camera.asp>
- Splaks USB Digital Microscope Camera: https://www.amazon.com/gp/product/B07L5LYZLJ/ref=ppx_yo_dt_b_asin_title_o09_s00?ie=UTF8&psc=1
- Teslong Ear Examination Camera: https://www.amazon.com/Teslong-Otoscope-Inspection-Waterproof-Endoscope/dp/B07TTBDDTV/ref=sr_1_3?keywords=ear+camera+teslong&qid=1569354490&s=electronics&sr=1-3
- Microsoft USB2.0 VGA UVC Webcam: Laptop Webcam for the Asus K501L Windows 10 laptop

REFERENCES

- [1] EPICS Collaboration, <https://epics.anl.gov/>
- [2] M. L. Rivers, “areaDetector: EPICS Software for 2-D Detectors”, in *Proc. 16th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'17)*, Barcelona, Spain, Oct. 2017, pp. 1245-1251. doi:10.18429/JACoW-ICALEPCS2017-THDPL03
- [3] USB Video Class v1.5, <https://www.usb.org/document-library/video-class-v15-document-set>
- [4] Allied Vision, <https://www.alliedvision.com/en/about-us/about-us.html>
- [5] Flir, <https://www.flir.com/about/about-flir/>
- [6] J. Wlodek, K. Gofron, <https://github.com/jwlodek/ADUVC>
- [7] K. Tossell, <https://github.com/libuvc/libuvc>
- [8] libusb developers, <https://github.com/libusb/libusb>
- [9] T. Lane, Independent JPEG Group (IJG), <http://libjpeg.sourceforge.net/>
- [10] Brookhaven National Laboratory, Photon Sciences Division, <https://www.bnl.gov/ps/beamlines/beamline.php?r=17-BM>