MACHINE VISION CAMERAS

S200 | S210 | S641 | S991 | S711

Phantom Machine Vision Cameras: Hardware and Control Software GUIDE

When it's too fast to see, and too important not to.®

PHANTOM®

VISION RESEARCH / AMETEK®

CONTENTS

Written and produced by the Marketing Department at Vision Research.

The contents of this manual are subject to change without notification.

PN: ZDOC-64119-MA-0001, Rev 1
Last Updated: 5 Sept 2023

ABOUT THIS DOCUMENT

1

This document has been designed to introduce users to Phantom Machine Vision cameras and also to get users up and running in terms of backend hardware configuration and camera-control software.

For those who need specifics about the camera specifications, we encourage you to visit www.phantomhighspeed.com. For those looking for the technical details regarding specific cameras, we encourage you to explore the ***Set-up and Operation Guide*** provided on the bottom of the respective cameras' product page. A sample table of contents is provided for the S641 to give you an idea for the contents therein:

S641 Set-up and Operation Guide

CONTENTS

The contents of this guide are subject to change without notification.

Last Updated: June 2023

INTRODUCTION TO MACHINE VISION

2

Machine vision is the process of feeding image-based data into an electronic workflow and retrieving useful information back. This can range from:

- Image preprocessing – identify features of the raw image and enhance the features we would like to analyze (reduce noise, improve contrast, identify edges)
- Feature extraction/ pattern recognition – identify and extract significant characteristics of the image (template matching, face recognition, color variations)
- Measurement and quantitative analysis – assess physical properties, spatial relationships, and pass along real-time quantitative evaluations (DIC, pyrometry, spectroscopy)
- Motion analysis and temporal understanding – analyze the motion of objects or fluid flows (object tracking, rotational motion, dynamic mode decomposition)
- Segmentation and interpretation – dividing an image into meaningful parts and interpreting those parts (semantic segmentation, object segmentation, image registration)

Generally, the purpose of machine vision is to enhance human visual perception by performing image-based analysis quickly and efficiently, or as a support to the limits of human patience, as machine vision processes can automate mundane tasks that a person might lose focus on or could potentially make an error in. In Figure 1, instead of a person analyzing each object to look for defects, the image processing library takes care of it, rejecting any defective object off the conveyor belt.

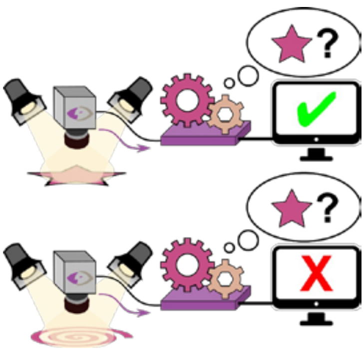


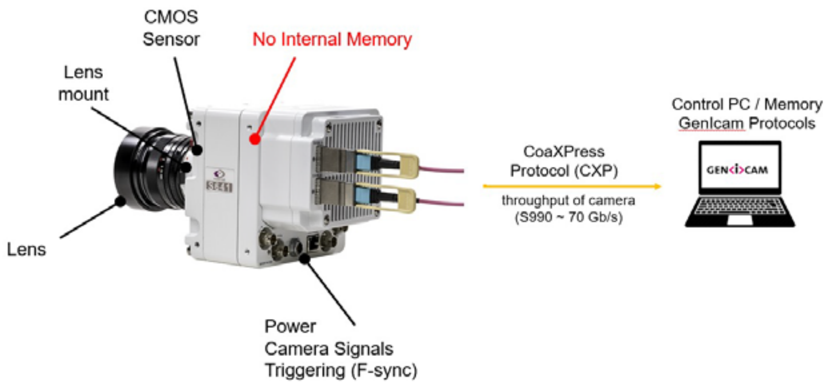
Figure 1 – An example of feature extraction. The backend searches for a star and if a star is found, let the user know with a checkmark. If not, display an X.

ADVANTAGE OF MACHINE VISION ARCHITECTURE

3

The paramount feature of the S-line Phantom cameras (where S stands for streaming) is the ability to stream image data directly off the camera as it is created (Figure 2a), where there is no intermediate volatile memory (i.e., a RAM buffer) on the camera before making it to a backend PC (like on the standard Phantom cameras, Figure 2b). In contrast, the data can be streamed immediately off the camera at speeds up to 9 GPx/sec (such as for the S990/991).

a.



b.

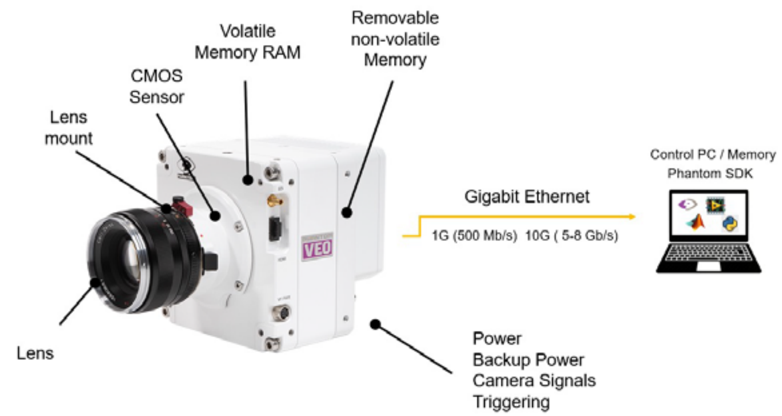


Figure 2 – (a) Phantom machine vision architecture. (b) Phantom standard architecture.

APPLICATIONS AND TECHNIQUES

4

From the standpoint of applications and techniques, the S-line cameras have the unique ability to enable the following tasks: (1) real-time image analysis, (2) long-record times, and (3) real-time saving the non-volatile memory.

Unlike standard Phantom cameras (Figure 2b), not only can you record as long as you desire (limited only by your PC's storage), you can process frame data in any way that you desire (your own PC's processing speed permitting). The S-line of cameras couple ultra-high speed sensors with all of the advantages that machine vision techniques provide.

A list of sample high-speed applications & techniques for the S-line cameras include, but are certainly not limited to:

APPLICATION AREAS

- Industrial Inspection
- Explosives Engineering
- Microscopy & Microfluidics
- Wind Tunnel Research
- Welding Imaging
- Image Cytometry
- Semiconductor Inspection
- Combustion Emission
- Ballistics & Range Testing
- Fluid Dynamics
- Materials Analysis
- Life Sciences (Animal Studies)
- Aerospace
- Automotive
- Fundamental Research

TECHNIQUE AREAS

- Digital Image Correlation
- Particle Image Velocimetry
- Spectroscopic Analysis
- Motion Analyses (2D/3D)
    - Displacement
    - Speed
    - Acceleration
    - Vibration
    - Rotation
- Intensified Imaging
- Schlieren & Shadowgraphy
- X-ray Imaging
- Electron Microscopy
- Sacrificial Camera
- Digital Holography
- Optical coherence tomography

RECOMMENDED HARDWARE

5

As mentioned in the preceding section, the camera's throughput can be on the order of 12 GB/s (without data processing). Therefore, it is imperative that hardware is used that can handle these throughputs for the imaging application. For the S641, S710, and S991 the hardware recommendations are generally the same with only the S210 and S200 requiring less high-end hardware. Here is a brief explanation of each category of hardware needed (a summary is shown in Figure 3):

1. **Frame grabber:** The frame grabber is a PCI-e card that is used to capture and process the individual frames of video coming from the camera. Unlike the standard cameras, the frame grabber is necessary for the S-line of cameras because they provide the absolute maximum throughput and speed to do real time analysis. Many frame grabbers also offer features to accelerate frame analysis, such as direct GPU transfer or programmable FPGAs. Importantly, some frame grabbers support custom IP cores, allowing you to implement specialized video processing or data handling algorithms directly on the FPGA for enhanced performance and flexibility.
2. **Data Transfer Cables:** The data transfer cables can be either CoaXPress (CXP) or CoaXPress over Fiber (CXPoF) depending on the camera model. These are the media in which the data is transferred from the camera to the frame grabber. For the case of CXPoF, QSFP+ transceivers will be slotted into the back of the camera, whereas standard CXP connections implement DIN 1.0/2.3 connections.
    a. **Single**- vs **Multi**- mode fibers: Single mode fibers enable one type of light mode at a time, whereas multimode fibers can propagate multiple modes simultaneously. The key differences between the fiber types are diameter, wavelength, bandwidth, color sheath, distance, and cost. Single-mode fibers are capable of greater transmission rates and distances, compared to multi-mode fibers. However, single-mode fibers are generally more expensive than multi-mode fibers.
3. **Fiber Extension**: If long data transfer cables are needed (i.e., >10 km), it is possible to introduce CoaXPress Fiber Extenders for 1 or more data channels.
4. **Lens**: The lens (with a respective aperture) will focus light onto the sensor.
5. **Lens Mount**: the lens mount adapts specific lenses (i.e., Canon vs Nikon) to the camera.
6. **Power**: the AC adapter necessary to use the camera at maximum capacity. The cameras can be pulled into standard wall power (i.e., 120 VAC 60 Hz).

7. **CPU**: The CPU is the Central Processing Unit and ultimately determines how fast your computer can process the frame data and how fast it can perform any real time analysis.
8. **Motherboard**: This is the main printed circuit board (PCB) that houses all the other backend components, including the CPU, RAM, storage, and a potential GPU. One of the most important specifications to look for in a motherboard is the PCI-e lane availability and lane generation. On every motherboard there is a certain allowable PCI-e bandwidth and it is important to use a motherboard that is capable of handling the frame grabber(s), storage devices, and potential GPU all in parallel without throttling any of the devices. For example, the S641, S710, and S991 require two frame grabbers to run at max throughput and if you are also using a GPU, you would need a motherboard that can handle an x16/x8/x8 (16 lanes for the GPU, 8 for the frame grabber, 8 for the second frame grabber) all in parallel. The storage devices also use PCI-e lanes as well (x4), so it also becomes important to check if the motherboard is capable of writing to the storage device in parallel without throttling the other devices.
9. **RAM Memory**: RAM (random access memory) where frame data can be temporarily written to for easy and ultra-fast access. The frame grabber's DMA (direct memory access) unit writes the frame data directly to here so the user can access and process the data immediately upon arrival.
10. **Non-volatile Memory**: The M.2 NVMe is a type of SSD which interfaces with the motherboard through PCI-e. These are typically the fastest storage devices available and if you choose to write the raw frame data to storage, it is necessary to select storage devices which can write as fast as possible to keep up with the throughput of the camera.

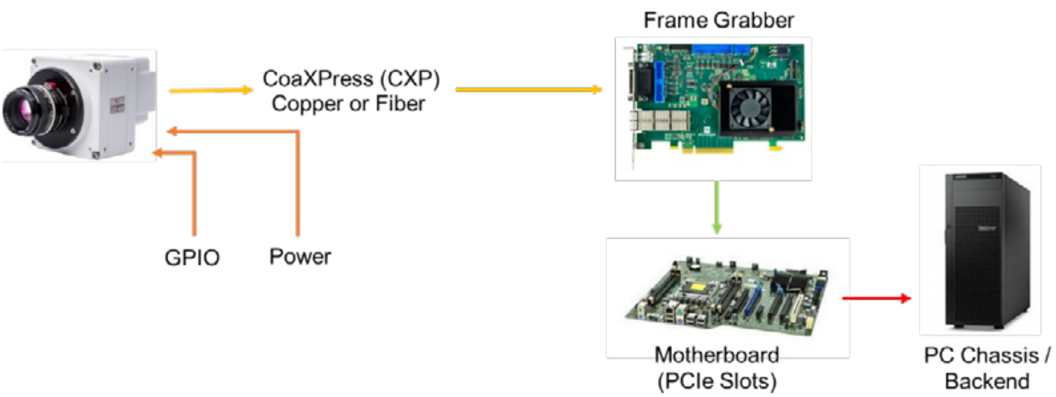


Figure 3 – Schematic illustration of the fundamental components in a Machine Vision configuration, where a camera is connected to one or more frame grabbers that are slotted inside a motherboard within a backend chassis.

S200, S210 Sample Backend Hardware

Frame Grabber(s)	1x 4-port CXP-6 PCI-e card
Cables	4 CXP-6 cables
Lens	C-Mount, F-mount (*need C-F adaptor)
Power	N/A
Other Accessories	Fiber extenders for distances >50m
CPU	Intel Core i5-12500 (or equivalent)
Motherboard	Capable of at least x8/x4 PCI-e 3.0+[1]
Memory	16+ GB DDR4/5
Storage	M.2 NVMe with >3000 MB/s write speed

The S200 and S210 (pictured above) do not require much in terms of backend hardware, relative to the S991, for example. An industrious laptop would be able to handle the throughput of the S210 with any sort of real time processing if a portable system is desired with a PCI-e extension board as well. The S210 is also the only camera capable of power over CXP (PoCXP) where the frame grabber will supply the power to the camera with no AC adapters or power cabling required.

S641, S991 Sample Backend Hardware

Frame Grabber(s)	2x CXPoF QSFP+ compatible PCI-e card
Cables	2x MTO/MTP fibers
Lens	F-, Canon Mount
Power	80 W Power Supply
Other Accessories	4x QSFP+ Transceivers
CPU	Intel Core i7-13700KF (or equivalent)
Motherboard	Intel Core i7-13700KF (or equivalent)
Memory	64+ GB DDR4/5
Storage	2x M.2 NVMe with >5000 MB/s write speed

The S641 (pictured above) and S991 require higher end hardware since both cameras are capable of >10 GB/s throughput. These are both CXPoF based cameras so the fiber can be drawn out as far as the transceiver is rated.

[1] Must be a compatible chipset with your processor.

S710 Sample Backend Hardware (no GPU)	
Frame Grabber(s)	2x 8-port CXP-6 PCI-e card
Cables	16x CXP-6 cables
Lens	F-, G-, C-, Canon
Power	160 W Power Supply
Other Accessories	Fiber extenders for distances >50m
CPU	Intel Core i7-13700KF (or equivalent)
Motherboard	Capable of at least x8/x8/x4 PCI-e 3.0+1
Memory	64+ GB DDR4/5
Storage	2x M.2 NVMe with >5000 MB/s write speed

The S710 (pictured above) requires similar backend specifications as the S641 and S991, but the data transport is over CXP instead of fiber.

Server Architecture (with GPU)	
Chassis	4U Rack mountable 8 Bay chassis
Motherboard	Intel C422 platform motherboard with 48 PCIe lanes
CPU	Intel Xeon W-Series 64-bit LGA2066 processor
RAM	32 – 1024 GB DDR4-2933 RDIMM (8 slots)
Hard drive	1 TB Intel SSD
OS	Win 10/11 64 PRO
Internet	Dual Intel X-550-AT2 10G Ethernet
GPU	NVIDIA Quadro GPU processor
Chassis	4U Rack mountable 8 Bay chassis

A note on GPUs

GPUs are not necessary for most applications and certainly not necessary at all if you do not require real time processing. The necessity of a GPU for parallel processing is considered on a case-by-case basis and it would change the specifications required here – specifically, we would recommend a backend with server architecture as opposed to the desktop class architecture above. The major difference is that the server architecture and server processors are capable of handling more PCI-e lanes compared to the desktop class computers so you can easily run all devices at full throughput on your backend, such as a PCI-e structure that looks like x16/x8/x8/x4 at least (one GPU, two frame grabbers, one M.2 NVMe all operating in parallel through the CPU).

HARDWARE INSTALLATION

6

The installation for each camera is very similar, and described below:

1. Align and insert the frame grabber PCI-e cards into your motherboard.
2. Cable connections:
    a. CXP: Connect the CXP ports with the CXP cables from the frame grabber (see below) to the camera. Make sure they are in the proper order – the top CXP cable on the frame grabber (labeled A) should be inserted in port A on the camera.
    b. CXPoF: Insert the multimode fiber into the QSFP+ transceiver on both ends. Connect one end into the frame grabber by sliding the transceiver all the way through until you hear a click. Do the same for the camera by sliding the transceiver all the way through until you hear a click.
3. Plug the power cable into the camera (S200 and S210 has power over CXP).
4. Mount your lens, remove lens cover.

Figure 4 – CXP-6 cables plugged into the Euresys Coaxlink Octo.

7

SOFTWARE INSTALLATION (EURESYS & PYTHON)

Download the Euresys eGrabber software and drivers for the Coaxlink frame grabber. It will be located under the Setup Files subsection under EGRABBER FOR COAXLINK AND GRABLINK DUO.

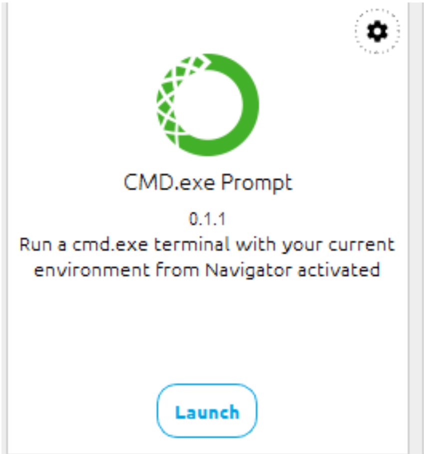
EGRABBER FOR COAXLINK AND GRABLINK DUO 22.10 ▾ −

	Download	File size	Operating system
Release Notes	egrabber-release-notes-22.10.1.2172.pdf	0.9 MB	Windows, Linux, macOS
Documentation	View eGrabber 22.10 online documentation (including PDFs)		Windows, Linux, macOS
	egrabber-linux-offline-documentation-en-22.10.1.2172.tar.gz	0.1 GB	Linux
	egrabber-win-offline-documentation-en-22.10.1.2172.exe	94 MB	Windows
Setup Files	egrabber-linux-aarch64-22.10.1.50.tar.gz	0.3 GB	Linux
	egrabber-linux-x86_64-22.10.1.45.tar.gz	0.3 GB	Linux
	egrabber-macos-aarch64-22.10.1.42.pkg	0.4 GB	macOS
	egrabber-macos-x86_64-22.10.1.46.pkg	0.4 GB	macOS
	egrabber-win10-x86_64-22.10.1.47.exe	0.3 GB	Windows
	egrabber-win7-x86_64-22.10.1.47.exe	0.3 GB	Windows
	Intermediate versions		

Select for your desired operating system and install.

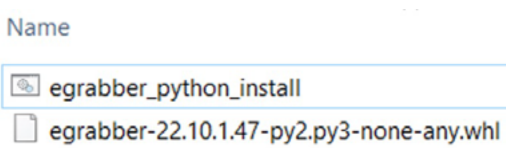
1. Download the Anaconda Python distribution platform. This will provide most of everything you need in the Python programming language in order to interface with the camera. https://www.anaconda.com/products/distribution

2. Once installed, click the CMD.exe Prompt.



3. In the command prompt, type 'pip install opencv-python'. This will install the Open Source Machine Vision Python package to your Anaconda.

4. Navigate to the folder where you installed the Euresys drivers and software. For example, the folder for me is C:\Program Files\Euresys. From here, navigate to ...\eGrabber\python. You should see two files:

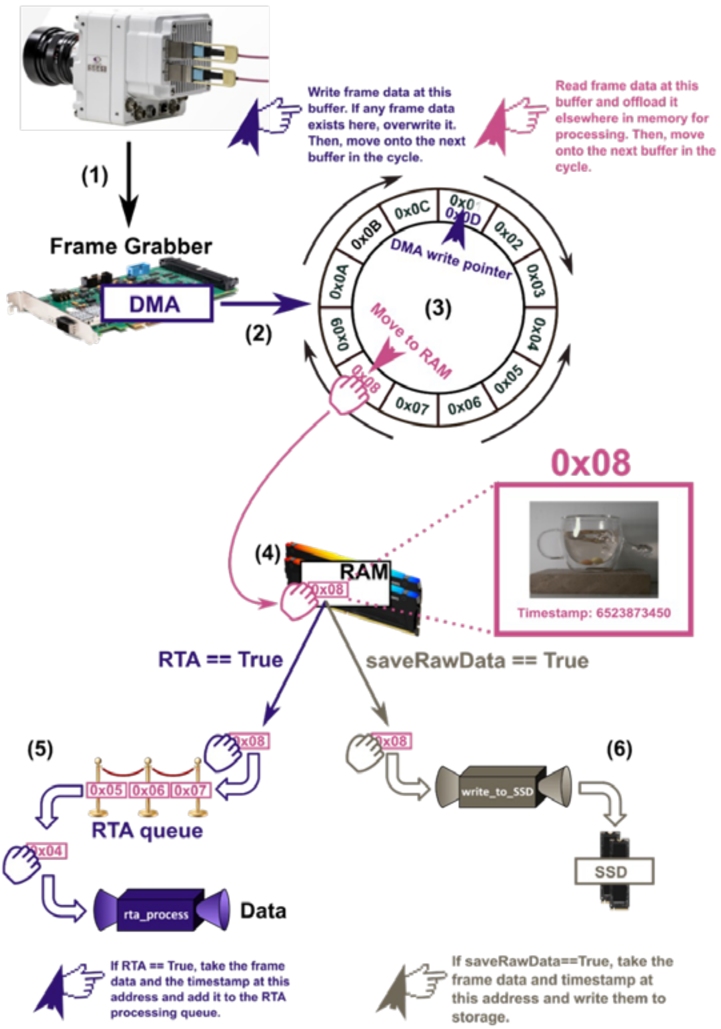


You will want to do a pip install of this wheel file (.whl file). In the command prompt, you would type 'pip install <directory>\Euresys\eGrabber\python\egrabber-22.10.1.47-py2.py3-none-any.whl' where <directory> is the location of the Euresys folder on your computer.

5. Here are the rest of the modules needed to run the camera. For any libraries/modules that you do not already have, pip install them with the same process above.

IMAGE-DATA FLOW

8



The figure above features a graphical depiction of the flow of data in the typical program.

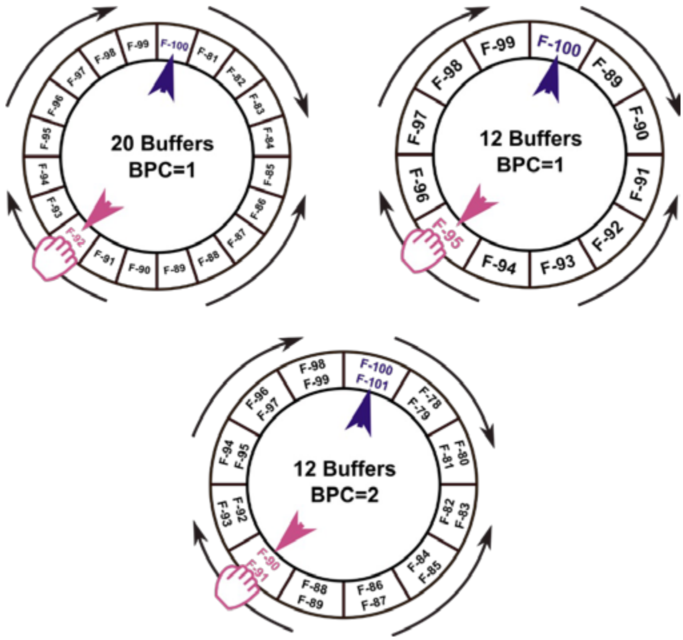
1. The camera sends the readout data from the sensor to the frame grabber.
2. The frame grabber receives the readout data and "grabs" individual, digital still images based on the image parameters set. The frame grabber through its DMA unit slots it over into a pre-allocated memory pool on your RAM and signals that the frame data with any associated metadata is available to interface with.

3. This is the memory pool, also known as a buffer pool. This type of buffer pool is a circular buffer with a FIFO scheduling routine.
   a. **Circular buffer**: also known as a ring buffer or cyclic buffer, a circular buffer is a data structure that uses a single, fixed-size buffer as if it were connected end-to-end. When the buffer is filled, the next frame data will overwrite the earliest frame data in the buffer, effectively wrapping around to the beginning (such as frame 0x0D is overwriting 0x01). This structure is particularly useful for holding streams of data such that old data can be continuously overwritten by newer data once the buffer is full.
   b. **FIFO:** First In, First Out (FIFO) is a method for organizing and manipulating data in a data structure, particularly in data storage and retrieval operations. As the name implies, the first frame data that enters the structure is the first one to be extracted and processed. In the context of a buffer pool or memory management, FIFO scheduling ensures that data is processed in the same order it was added, helping maintain the temporal consistency of the processed frames.

   In the context of the program, the frame grabber's DMA operates according to the purple pointer's instructions while the demo script operates according to the pink pointer's instructions. The goal here is to offload the frame from the buffer as soon as possible such that the purple pointer's instructions are simultaneously never interrupted and that the DMA never outpaces the program. For example, if your CPU is too slow, the DMA's purple pointer will overwrite frame data before the program's pink pointer can process it, causing dropped frames. This is why proper hardware is important.

   Additionally, there are two parameters that will assist you in making sure frames are not dropped: the number of buffers in the pool and buffer part count (BPC). These will be discussed in the next section.
4. The program offloads the frame data from the circular buffer and is ready to be processed. Here, we show exactly what is inside the memory chunk that was grabbed from the buffer: it's the frame data and a timestamp.
5. If you choose to use real time analysis (RTA), the frame data is sent to the RTA queue where it waits to be processed by your rta_process function, however you choose to define it. The example provided simply averages the brightness of each pixel and prints the value. If the brightness is too low or high, it will print a warning – an example of a feedback loop.
6. If you choose to save the raw frames, it will write the frame and the timestamp to storage.

9 BUFFER CONTROL DISCUSSION



Let's analyze the buffer pool a bit more closely. The figure above shows three examples of the two buffer parameters you can change: the number of buffers and the buffer part count (BPC). Here, instead of showing the memory addresses where the buffer is located, we show what is inside each buffer, in this case a frame and its index (F-100 is frame 100, or the 100th frame).

- **Number of buffers**: This controls how many buffers will be allocated to the pool. For the hardware specifications above, generally any number above 100 is sufficient for the RAM offloading routine to keep up with the DMA onloading routine. You might consider increasing this further if you have RTA processes running that can occasionally take up CPU cycles to complete, causing the offloading routine to lag behind the DMA onloading routine.
- **Buffer part count**: This is vital for low resolution, high FPS applications! BPC adds more partitions (hence, "part count") to an individual buffer. In other words, the DMA onloading routine will slot in more frames per buffer. In the example above when BPC=2, there are now two frames per buffer instead of one frame per buffer. If you are finding that your hardware cannot keep up with the frame rate of the camera simply by reading the data in time (even before any offloading routine), very typically the answer is to increase the BPC – it is more efficient and faster for the DMA to send over 10 512x512 frames at once (and for your computer to read in 512*512*10=2.62 MB one time) than it is to send over 1 512x512 frame 10 times (0.262 MB ten times within the same time span).

PYTHON SAMPLE SCRIPT

10

PYTHON SCRIPT:

https://phantomhighspeed.my.site.com/?section=PCCSoftware

SERVICE AND SUPPORT

11

If you experience any technical support issues, please do not hesitate to reach out to us at Phantom.Support@ametek.com.

In terms of support for real-time analysis, open-source software, and Euresys eGrabber libraries, please feel free to reach out to our applications expert at matthew.vayner@ametek.com, or contact your local sales representative.

GLOBAL HEADQUARTERS

Vision Research, Inc. - Wayne, New Jersey
100 Dey Road
Wayne, New Jersey 07470 USA
T: +1.973.696.4500

For answers to most questions, please visit us at: ***www.phantomhighspeed.com*** and search the camera product pages, tutorials, support knowledgebase and FAQs.

SUBMITTING A SUPPORT TICKET

For technical product support, operation and application information or to request an RMA, please submit a ticket by filling out a form at ***www.phantomhighspeed-service.force.com*** or by emailing us at **phantom-support@amtek.com**.

LIVE CUSTOMER AND TECHNICAL SUPPORT

Serving the Americas:
M-F 8:00 AM to 5:00 PM EST (GMT -4:00)
T: +1.973.696.4500
Customer Support, ext. 4002
Technical Support, ext. 4003

Serving Europe, the Middle East and Africa:
M-F 9:00 AM to 6:00 PM GMT +3:00

Vision Research, Inc. - Bucharest, Romania
T: +40 21 210 8587

Serving Asia Pacific:
M-F 8:00 AM to 5:00 PM GMT +8:00

Vision Research, Inc. - Shanghai, China
T: 86-21-58685111, ext. 141

Written and produced by the Marketing Department at Vision Research.

The contents of this manual are subject to change without notification.

PN: ZDOC-64110-MA-0001 Rev 4
Last Updated: September 5, 2023

WWW.**PHANTOM**HIGHSPEED.COM

100 Dey Road
Wayne, NJ 07470 USA
+1.973.696.4500