Camera Array Guide

Written by: Steven Stuber

What is this?

This is a guide on how to use the ArduinoCameraController, ArduinoStrobeController, and USBDriveController for a capture using the camera array. The camera array can consist of up to 36 Sony Handycam HD camcorders. The controllers in this guide have separate documentation that can be consulted for any further questions or concerns. The following system works on Windows, Linux and Mac. The following sections will describe the steps needed to set up the devices, take a capture and download that capture.

Setting up Computer

Before any camera setup is done, we should make sure that the computer is fully operational and capable of running the controller applications. The following MUST be completed or the applications will simply not function.

For Windows:

Install the FTDI drivers for the virtual com port downloadable here.

If using a 64 bit version of windows you may need to recompile the software.

For Linux:

Install the FTDI drivers for the virtual com port (if needed) here.

Get added to the user group *UUCP*.

For Mac:

Install the FTDI drivers for the virtual com port downloadable here.

Look at notes for Mac side application in ArduinoCameraController documentation.

For any of the three systems the serial port may have problems functioning properly. If in Windows the COM port is past COM9 it will not auto detect so it is advisable to force the new port to be from COM0 to COM9. In Linux or Mac your distribution may not be laid out the same way; so far, OpenSUSE and Ubuntu have been tested to work.

For any problems that may occur there is a troubleshooting section in the documentation for each program.

Setting up Cameras

The first matter at hand is to make sure the cables, cameras, strobes and controllers are properly plugged in and positioned.

Position your cameras in the general area they will be pointing in. Do not worry about perfect alignment at this stage as more set up is required.

Plug the power, USB, and LANC cable (2.5mm to 3.5mm converter or cable) into the cameras.

Take every power cable and plug them in. Take every USB cable and plug it into the USB hubs – order does not matter here. Take every LANC cable and extend it with any length 3.5mm to 3.5mm extension and/or plug it in to the respective position on the Camera Controller box. Connect such that CameraXX is in port XX on the Camera Controller box.

. Plug the Strobe Controller box into the other strobes via the DMX XLR cable and ensure that all strobes are daisy chained with the DMX XLR cables so the signal propagates.

Plug both the Camera Controller box and Strobe Controller into separate ports on the computer or their own hub that is separate from the camera USB hub.

Plug the DC power supplies into the Camera Controller, USB hubs and optionally the Strobe Controller and plug the alternate end into a power bar.

Also ensure that all strobes are plugged in with power and that all power bars are turned on.

Notes about Cameras

The cameras have an “easy” button. This button will reset most of the cameras current configuration to a predefined state. This includes changing it to auto focus, auto exposure, auto white balance and others. It is a useful way for getting the cameras to a known state where further configuration can then proceed from. Consider using this as part of the calibration step for your individual captures.

The cameras also must have the power knob to On in order to let the Camera Controller to turn them on remotely. If the position is Off the camera will come on for several seconds and then turn off automatically. On the newer cameras the viewfinder being opened or eyepiece being retracted, serve the same purpose as the On knob for the older models.

Optional Step – Clearing Cameras before Capture

This step should have been taken by those preceding you; however, if it was not, we should perform it.

Open a terminal to operate the Camera Controller. Run “ArduinoCameraController.exe –d find” to find the port the Camera Controller is on. For the rest of this guide I will assume the result was COM4.

Now we run “ArduinoCameraController.exe –d COM4 turnon” this will turn every camera currently connected on. To double check every camera is on, a quick test can be performed: Run “ArduinoCameraController.exe –d COM4 report” and make sure the total at the end reads the number of cameras you are actually working with.

When the cameras have turned on we can then plug the USB hub into the computer.

This will bring up a menu on each camera that was connected with a USB cable. We now need to press the top left button labeled Computer. This will allow the drives to be detected and mounted.

Once every drive is mounted open a terminal to operate the USB Drive Controller application. Run “USBDriveController.exe report” and make sure this returns a total number of cameras equal to the number that you are working with. If it does not, some troubleshooting must be done (consult the individual application documents).

When we have all drives mounted we can now type the clear command to erase the cameras of their previous capture information. If it is desired to clear even the index values a “-ri” flag should be added to the following statement: “USBDriveController.exe –e –v –p”. This will erase all of the pictures and videos from the cameras.

Note that by resetting the index we are required to later physically press the screen to allow the camera to rebuild its file system. This should have no harmful effects other than taking up time.

To get the cameras back to an idle state complete the following: run “ArduinoCameraController.exe –d COM4 turnoff”, unplug the USB hub from the computer and then run “ArduinoCameraController.exe –d COM4 turnon”. This should remove the connecting/connected screen they previously had and return the cameras to the state they were in before we plugged in the USB hub.

Starting a Capture

Taking a capture is best done when the cameras are empty and all of the previously described setup procedures have been followed. If this is the case, continue with this guide.

Open a terminal to operate the Camera Controller. Run “ArduinoCameraController.exe –d find” to find the port it is connected to. For the rest of this guide I will assume the result was COM4.

Run “ArduinoCameraController.exe –d COM4 turnon” to turn the cameras on and then type “ArduinoCameraController.exe –d COM4 report” to make sure the right number was detected.

Run “ArduinoCameraController.exe –d COM4 video” to get every camera into the video mode.

Set up camera positions and settings to the exact values you desire for your capture. It is recommended to manually set the cameras to have autofocus on during this step because later we can then disable it once the object has been focused on, allowing it to stay focused even if lower light conditions occur later on.

Turn off the room lights

Open a terminal and run “ArduinoStrobeController.exe –d find” to find the Strobe Controller. For the rest of this guide I will assume this previous command returned COM5.

To set up focus on the object we run “ArduinoStrobeController.exe –d COM5 on” and place the object in the capture field of view. Then, we wait for the cameras to autofocus and then issue the command “ArduinoCameraController.exe –d COM4 ftog” to lock the focus level.

Now run “ArduinoStrobeController.exe –d COM5 –f NTSC –w 6600” to make the strobes flash at NTSC (29.97Hz). Any –w setting can be used but 6600 microseconds is a fairly nice one. “–w” represents what width in microseconds we want the strobes to be on for, lower or raise as needed.

Run “ArduinoCameraController.exe –d COM4 record” to start recording and when finished run “ArduinoCameraController.exe –d COM4 pause”. Repeat as necessary.

Now we turn off the strobes: run “ArduinoStrobeController.exe –d COM5 off”

Turn on the room lights

Downloading the Capture

Downloading a Capture requires you to physically press the camera screen at one point so before proceeding make sure all desired captures have been taken with the current alignment and configuration.

With the cameras on in some idle state we now plug in the USB hub to the computer.

A screen will come up on every cameras viewfinder; press the Computer button on every screen and wait for them to say connected.

To double check we can run “USBDriveController.exe report” to see if the operating system has found all of the cameras yet. Make sure this number returned coincides with the number we are actually using.

Now we are ready to download. A path is required to download to and for this guide we will pretend it is “C:\Temp\capture\”; the trailing “\” is not required and will be generated if needed later. To download all the videos to this directory, we run “USBDriveController.exe –v C:\Temp\capture\”. To also get the picture files, we could simply add –p. Other options are available and outlined in the USBDriveController documentation.

After some time the program will return and the operations will be complete. At this point it is a good idea to double check that the right number of files was downloaded and no interruption occurred.

If the data is safely stored we can now proceed to erase the camera data to give the next user a clean set up.

We can erase all the camera files by simply running “USBDriveController.exe –v –e”. If we desire to also delete the pictures or to reset the index we can add “-p” or “-ri” respectively. Again, more options and information about them are available in the USBDriveController documentation.

Now that the cameras are clean we can put them back into an idle state again by unplugging the USB hub and then running “ArduinoCameraController.exe –d COM4 turnoff turnon” to reset them and return them to whatever mode they were in before we entered this download step.

If no further captures are needed the cameras can be turned off by running “ArduinoCameraController.exe –d COM4 turnoff”.

Appendix

Scripting Notes

When creating scripts to automate the use of this application it is important to utilize the report command available for the ArduinoCameraController and USBDriveController applications. These will report back the total number of cameras detected at that point in time, this should be compared to the actual amount being used. If the amount is not found then either more time is needed or user intervention is required.

Autofocus notes

During a capture we noticed that the autofocus on the cameras was very unreliable under low light conditions. To help focus I recommend using the strobes full on, placing the desired capture object in the field of view, allowing them to come to focus (by having autofocus on) and then disabling the autofocus so that it locks the focus settings. This will ensure that even under lower light conditions you will still retain a proper focus on the object in question.

Doing the following guide in Linux or Mac

Using this guide in Linux or Mac will require a few slight changes to the terminal calls I made earlier.

“ArduinoCameraController.exe” needs to be “./CameraController”
“ArduinoStrobeController.exe” needs to be “./StrobeController”
“USBDriveController.exe” needs to be “./DriveController”

Also when using “./CameraController –d find” for example instead of receiving results like COM4 or COM5, results such as “/dev/ttyUSB2” or “/dev/tty.usbserial” will be found instead. And instead of downloading to a path like “C:\Temp\capture\” a path like “/tmp/capture/” is more appropriate.

On Mac, the ability to open a serial port without restarting the Arduino was not discovered. This means that each call will take roughly 7 seconds of overhead to complete. The only way around this would be to find settings specific to your Mac OS and attempting to turn off Hang UP on CLose (HUPCL) on the ports connected to the Arduino.

Other than these slight changes the rest of the application should function exactly the same way, except for the return messages from the operating system that are displayed during the file copy, move or delete commands.