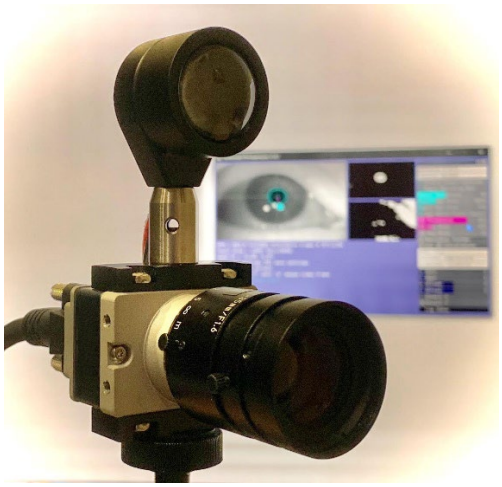


Copyright 2023 by Neuro-Software Developers, Inc. Jacksonville Florida. USA.

Contact:

Email : support@myeyetracker.com

Minimum computer hardware:

1. MS Windows 10, 64 Bit
2. Intel Core i5 7th Gen 2.2 GHz CPU w/8 GB RAM
3. Gigabit Ethernet Port *or* USB3.0 to Ethernet Dongle for connecting camera to computer.
4. Nvidia GTX 950 or Intel Iris Plus Graphics 640 or better graphics card (GPU) (VSync Off/Fast)
5. 128 GB Solid State Drive (SSD)

Software Download:

Oculomatic Program *Updates Only :*

https://www.myeyetracker.com/oculomatic_versions.html

FULL INSTALLTION (includes all drivers)
https://www.myeyetracker.com/downloads/hairball/Hairball_Installation.exe

3rd Party Analytics and Stimulus Presentation Software – "Hairball"

https://www.myeyetracker.com/downloads/hairball/Hairball_Installation.exe

In the Box (hardware components) 3

Installation & Setup 4

Using Software

1. Software first impression 7
2. Camera Configuration 8
3. Thresholding the image 9
4. Calibration 10
5. Pupil Size 14
6. Analog Output 15
7. Data Sharing 16
8. Saving and playing back video 17
9. Logging data 18
10. Troubleshooting 19

In The Box:

Hardware Item List:

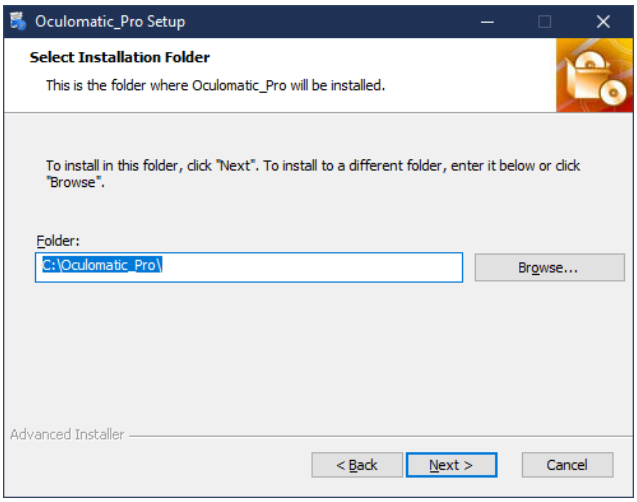
1. Camera and Lens – packaged as one unit.
2. IR Emitter – for illuminating the scene. Camera cannot see the eye without this. Attached to Camera body.
3. POE – (power over ethernet) connects camera cable to computer and supplies power. Includes all cables.
4. USB License key dongle – software won't run without it.
5. USB DAQ Device – for outputting the X,Y gaze analog signal to external devices.
6. Mounting arms for camera and/or IR Emitter.

   OPTIONAL ACCESSORIES:

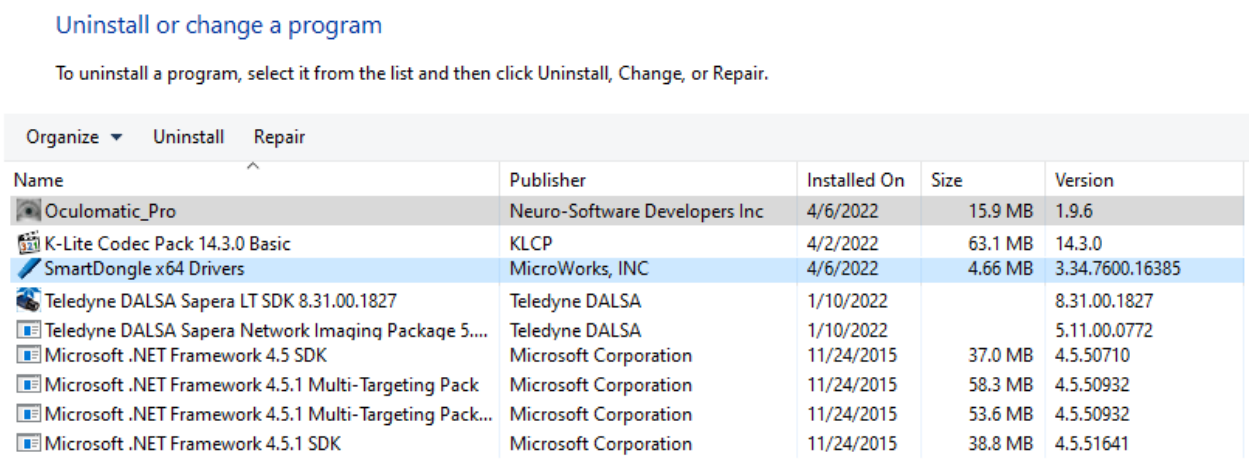
7. Ethernet to USB dongle – for connecting camera to a USB port instead of ethernet.
8. NUC PC (preconfigured)

Installation and setup:

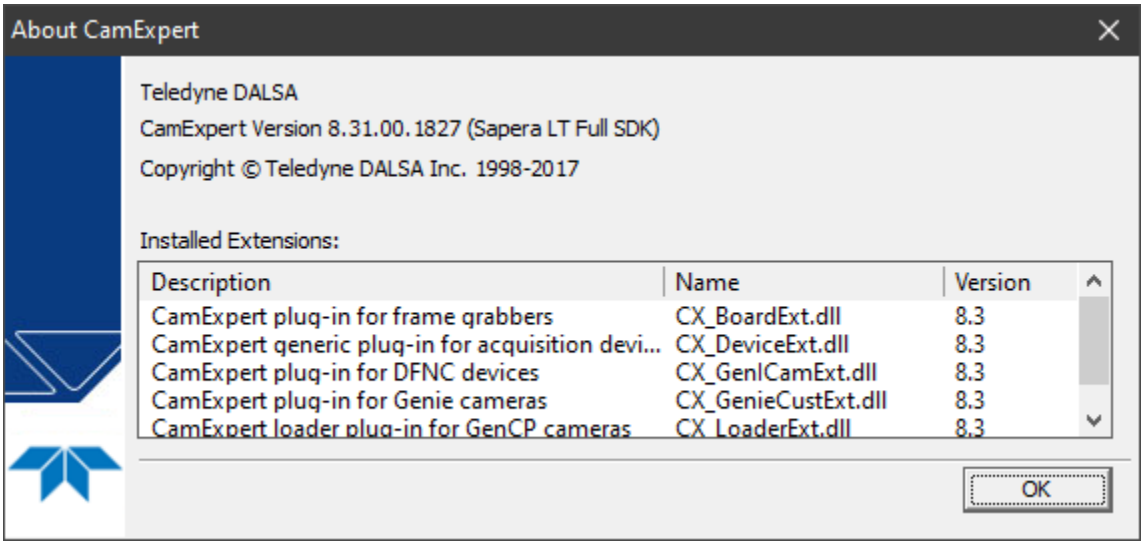
We provide a single click installation program that automatically performs all the below steps. However, when necessary, you can follow the step by step instructions below.



The following software components should be first installed for Oculomatic to work properly. These are all included automatically when installing the FULL Driver. Here is a screenshot from Windows Control Panel:



a) Teledyne DALSA Sapera LT SDK camera driver.

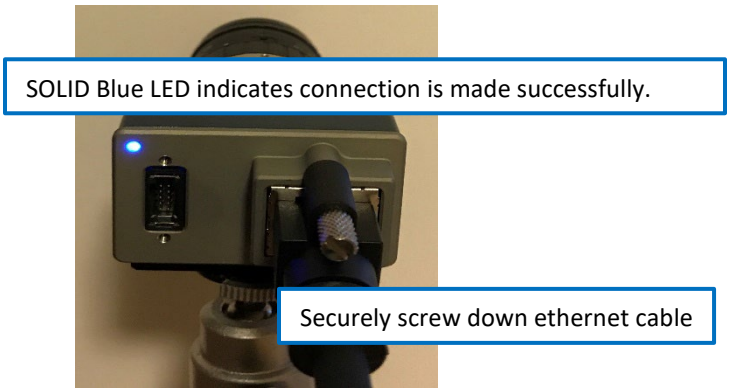


b) Microsoft .Net Framework 4.5
c) Smart Dongle (USB license key)
d) Microsoft 2013 *and* 2015 x64 Redistributable
e) K-Lite Codec Pack, will allow Oculomatic to playback MP4 and some AVI video files. Native Oculomatic video file playback (.vid files) do not require this codec.
f) Optional Data Acquisitions drivers:
    i) Measurement Computing (MCCDAQ / Instacal) and National Instruments (NIDAQmx) are both supported. MCCDAQ is installed automatically because it is our preferred vendor. Those wishing to use national instruments devices need to install the NIDAQmx driver manually, and specify the correct NIDAQ board in the settings.xml file. See the "Analog Output" section of this manual for more information on how to configure data acquisition devices.

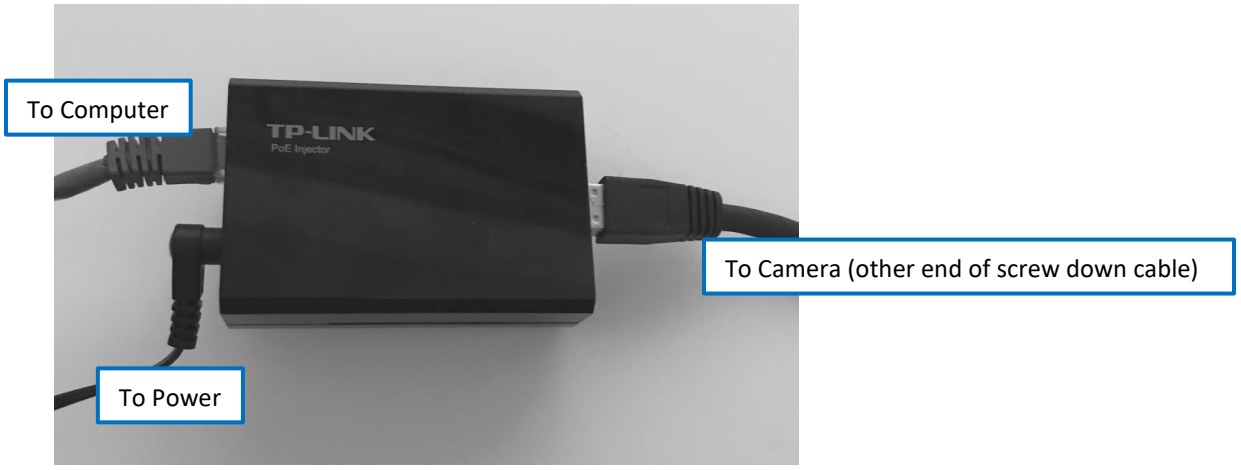
When all the above components are installed, Oculomatic Pro Software can then run smoothly with full functionality.

Connecting hardware:

a) Plug in the USB *License* Dongle (back of the computer into a USB(2.0) port works best).
b) Attach camera body and securely mount on a camera tripod or arm.
    i) Connect the supplied ethernet cable, with screw down end on the back of the camera and the other end into the POE Injector. MAKE SURE YOU SCREW DOWN ALL THE WAY.



ii) Make sure you connect the ethernet cable into the port labeled “output” on the POE injector.

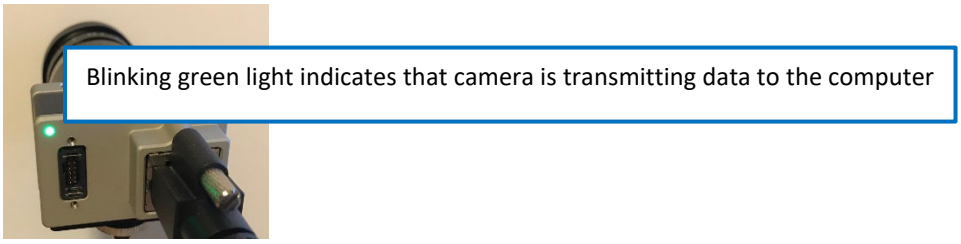


iii) Connect a regular ethernet network cable from your computer’s ethernet port (Gigabit) to the POE Injector’s “input”.

iv) Connect power cable to the POE injector and plug it into the electrical outlet.

v) You may need to restart your computer to get the network card to reset its TCP/IP

c) You can start Oculomatic software now.



d) Make sure IR Emitter is mounted and power is connected. A faint red light will glow from the emitter.

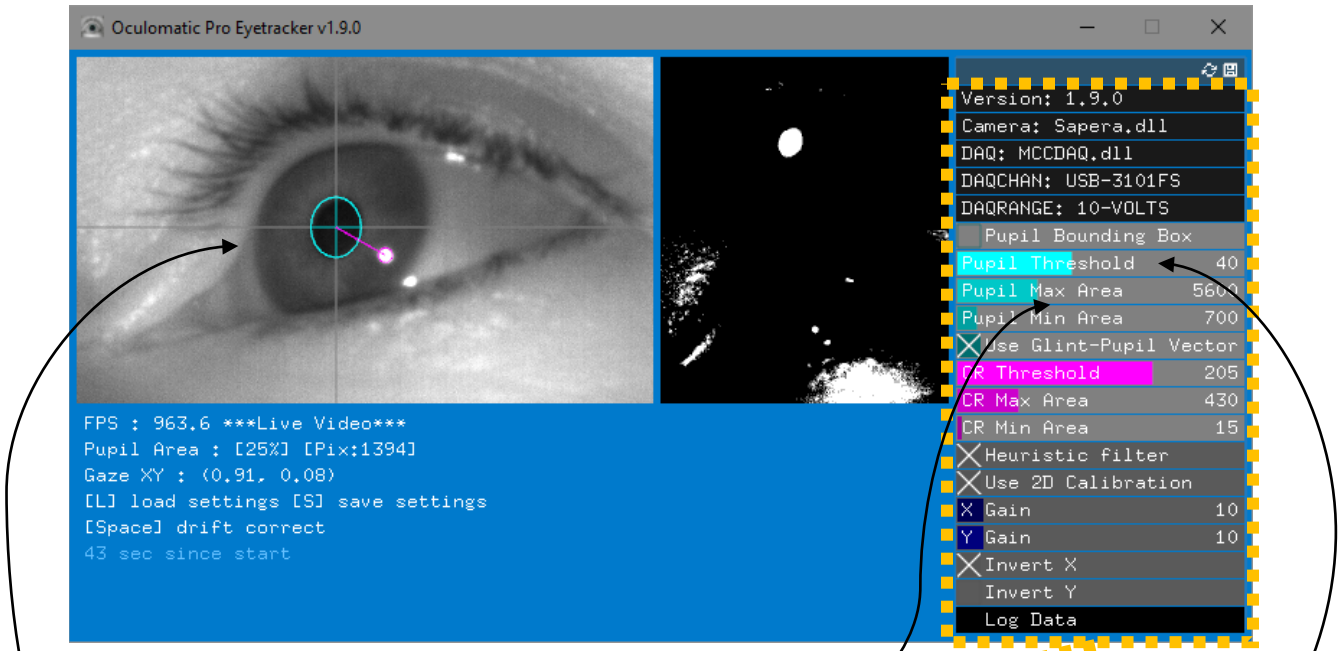
e) Position the camera and IR Emitter so that they are approximately 60 cm away from the eye (typical range is 30-70cm which is about arm’s length or less). You should see a clear image in the Oculomatic software, but you will likely need to adjust the focus, aperture, and direction the camera is pointing to achieve an image as shown below in the next section.

f) Make sure lens cap is removed, and check that aperture is open all the way.

g) Optionally, connect the DAQ Device to the USB Port and run BNC cables from the analog output 0,1 and 2 channels, to your data acquisition device.

Using the software

1) **First Look**



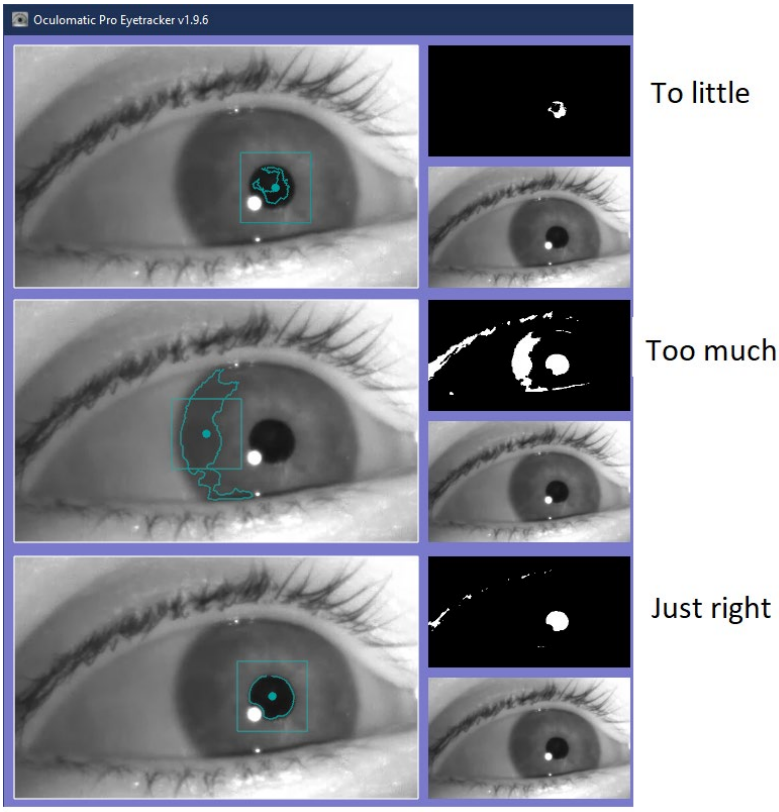
a) The software will start with default settings similar to those shown above.
b) Position the camera and IR emitter, and adjust the focus on the lens, so that you have a clear image of the eye as shown.
c) Adjust the Pupil Threshold so that it is clearly discriminated from the sclera, iris, and eye lashes.
d) Adjust the Pupil's max and min area to optimize tracking performance and properly normalize the pupil's diameter analysis.
e) Any changes to the settings shown in user interface require you press the 'S' key. This will overwrite the previous data\settings.xml file. Starting Oculomatic Pro will automatically load the previous settings.
f) You are now ready to start eye tracking.

2) **Camera Configuration**
   a) No configuration is necessary, the default camera is set to Sapera.dll however users have some options to choose from.
   b) Users can configure the camera type by editing the bin/data/settings.xml file
      Look for the following tags:

```
Option 1 (default)
<Source>Sapera.dll</Source>

Option 2
<Source>PTGrey.dll</Source>

Option 3 Videos
<Source>oculomatic_video_#n.vid</Source> (full frame rate recorded by Oculomatic)
Or
<Source>file name.mp4</Source>
Or
<Source>file name.avi</Source> (some files may not work)

If using a video instead of a camera
    1. Specify the file's location here :
           <Folder>data\videos\</Folder> (don't forget the '\' at the end)
    2. Install the K-Lite_Codec (included in the full driver distribution)
```

3) **Thresholding Adjustment**

a) Pupil threshold as well as Min and Max Area of the Pupil are the most important parameters. Make sure that the threshold is set correctly and optimally.



b) Once you have confirmed that set a range for acceptable Pupil dimensions (Min and Max Area) that works for your subject. Be sure to keep in mind the pupil diameter will change due to luminance changes of your stimulus display! Don't make this range too small and initially try what works best for you. When the contour perfectly encloses the pupil, gaze estimation tracking starts and analog outputs are sent.

4) **Gaze Calibration**

a) Because the gaze midpoint location is arbitrary in regard to the analog output range of the DAQ device, it is essential to perform an initial midpoint calibration. This requires the subject to fixate on a central fixation point while the experimenter presses the 'X' or Spacebar key. This is called "Drift Correction".

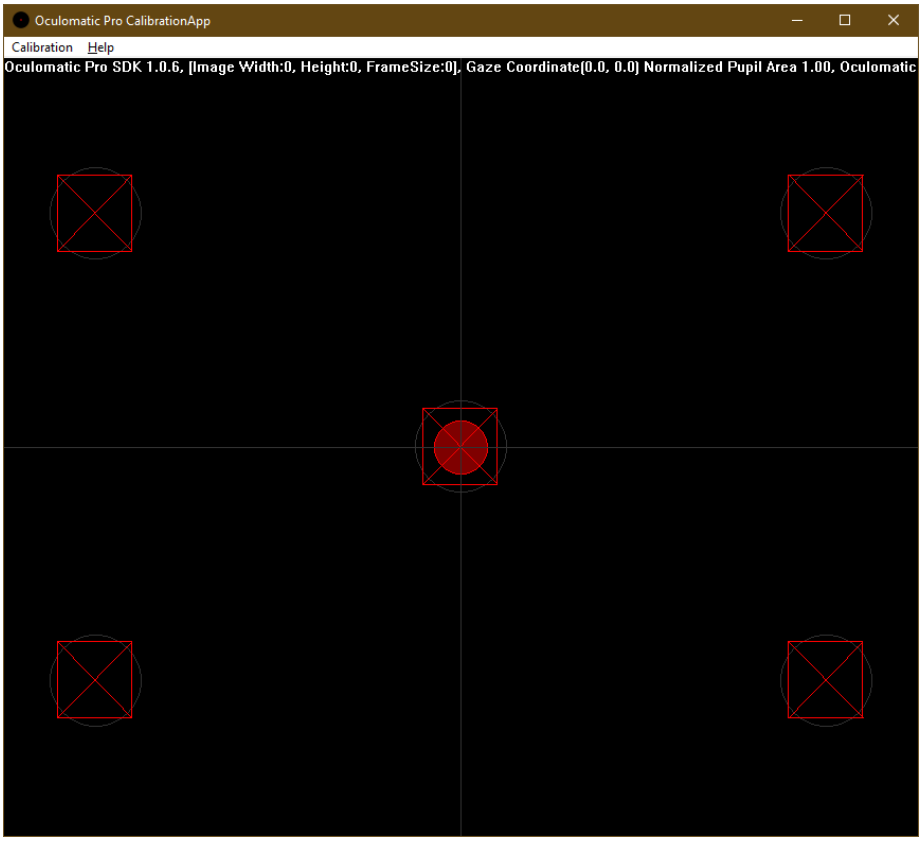
b) Gain can be independently controlled for the horizontal and vertical position if needed. If the gain settings are altered a NEW midpoint (drift correction) must be calibrated as to not introduce nonlinear shifts. If you require a smaller voltage range (such as +/- 5V you can reduce the gain from 1.0 to 0.5)

c) The usual approach is to keep the gain constant between experimental sessions and rather perform a 5 or 9-point 2D affine calibration in your experimental control software. This is very simple to do after you try it a few times.

d) By default, there is no 2D transformation applied; the raw gaze estimation signal is output on the horizontal and vertical channels.

e) To perform a 2D calibration, you must implement your own stimulus presentation sequence, capture the gaze raw data, and either save it to this file or send it to Oculomatic via the SDK.

f) We include a sample Matlab script *and* C++ program that demonstrates this. The C++ program is called **Calibration App**. You may edit the target locations and recompile the app as needed.



i) The 2D transformation matrix is stored in a file called bin\calibration.txt, see the appendix below for more info on the file format. You should not need to ever edit this file, or even worry about it. However, if you may wish to save separate calibration files per subject and use those files in future sessions. This will obviate the need to re-run the calibration sequence every time you change subjects. You can copy the saved calibration file into the Oculomatic\bin Folder. This feature will further improve in the future, but right now this is the only way to do it.

ii) To calibrate the eye tracker and improve spatial accuracy and precision you must ask the participant to look at some reference points (sometimes called control points). Typically, five 5 or 9 (recommended) locations are shown on the screen, then the participant is asked to look at each one individually. When their gaze is stable, record this average gaze coordinate (normalized) for each reference point. Oculomatic Pro uses these reference values to transform the raw data.

iii) Users should calibrate at least four (5) reference locations (oblique corners of the visual field and center).

g) There are two ways to transfer 2D calibration values to Oculomatic Pro

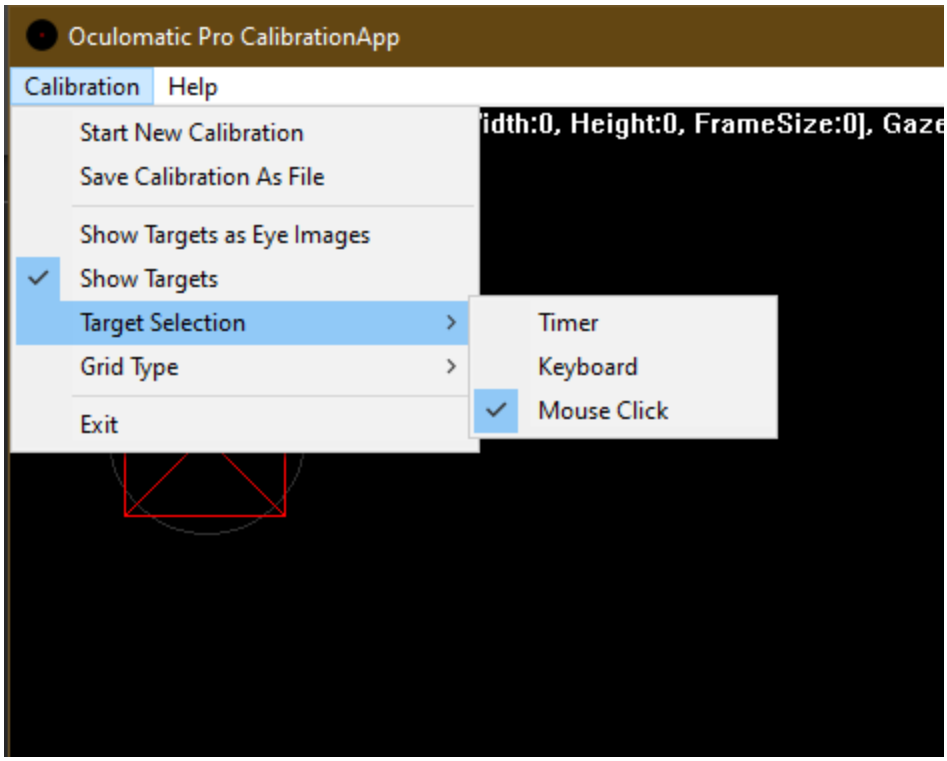
i) Using the format indicated below, create a file called “calibration.txt” that contains all of these values and place it into the Oculomatic Pro “bin” directory. For more information on the file format see below. The Calibration App included with Oculomatic does this automatically for you and allows saving a file. You can look at the code for more information.

ii) Use the SDK to transmit the data to Oculomatic. The Calibration App, also implements this for you automatically. As soon as a calibration procedure completes, the new values are transmitted to Oculomatic automatically, and the 2D transformation is enabled.

(1) Using The C Programming language, first `startCalibration(), then addCalibrationPoint(), and finally finishCalibration().` An example is given in the CalibrationApp.cpp file which comes included with the Oculomatic Pro software. The CalibrationApp is a complete open source example demonstrating how to render control points on the screen while sampling gaze position, recording and then transmitting to Oculomatic.

(2) Using Matlab, the script : “testCalibrateOculomatic.m” demonstrates how to display the control points on the screen, sample from the eye tracker and record the gaze value, construct a calibration matrix and send it to Oculomatic.

h) Using the Calibration App



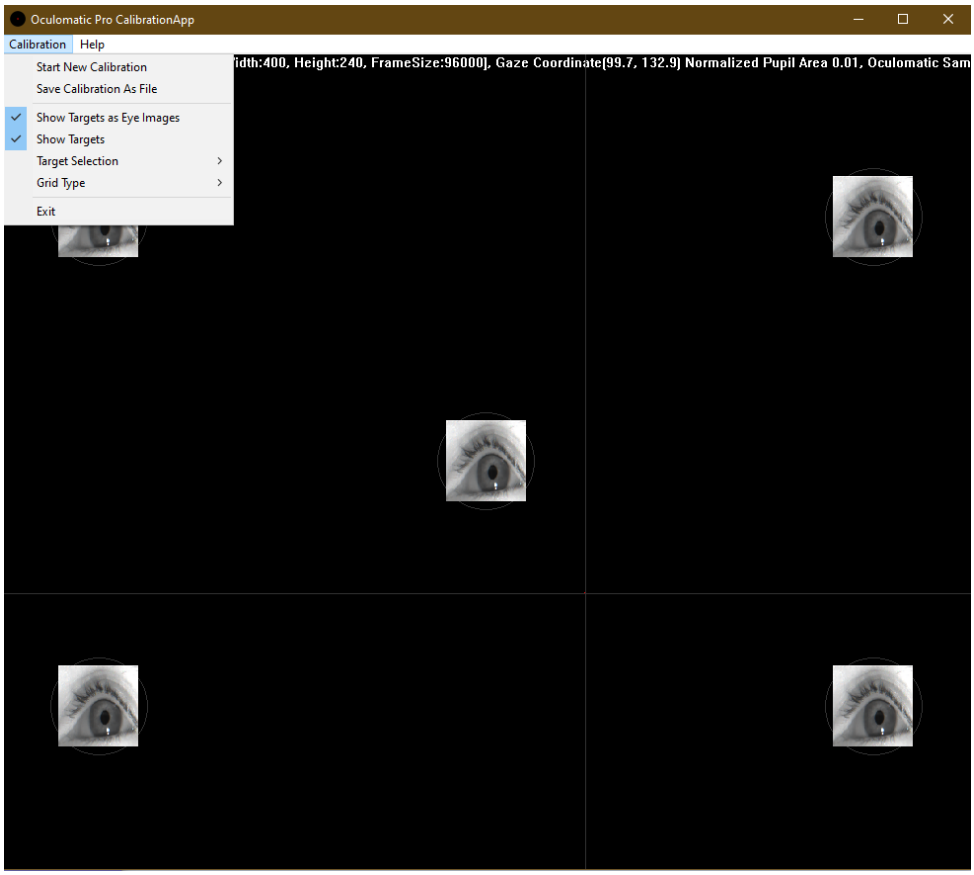
i) There are currently (3) Three target selection choices.

ii) The default is set to Mouse Click, which allows users to select any target, at any time, to calibrate. You may recalibrate the same location repeatedly as many times as necessary. **The calibration procedure does not end until the user pressed the escape key on the keyboard.**

iii) Choosing the Keyboard option will allow users to move to the next calibration target by pressing the space bar. This puts users in control of target timing, but the software selects the next target location (either sequentially or randomly depending on your choice). If you wish to control the choice of the next target, it is better to use the Mouse Click option instead. However, when choosing Keyboard , the program will automatically save and use the new calibration values after the last target is selected.

iv) The Timer selection method will increment the target location after ~1 second. This is ideal for hands free operation but limits the user's control. As with the Keyboard method, calibration automatically completes after the last target was presented.

v) Calibration App has a unique feature called Show Targets as Eye Images. This is can be a convenient way to verify that the eye is clearly visible for each target location.

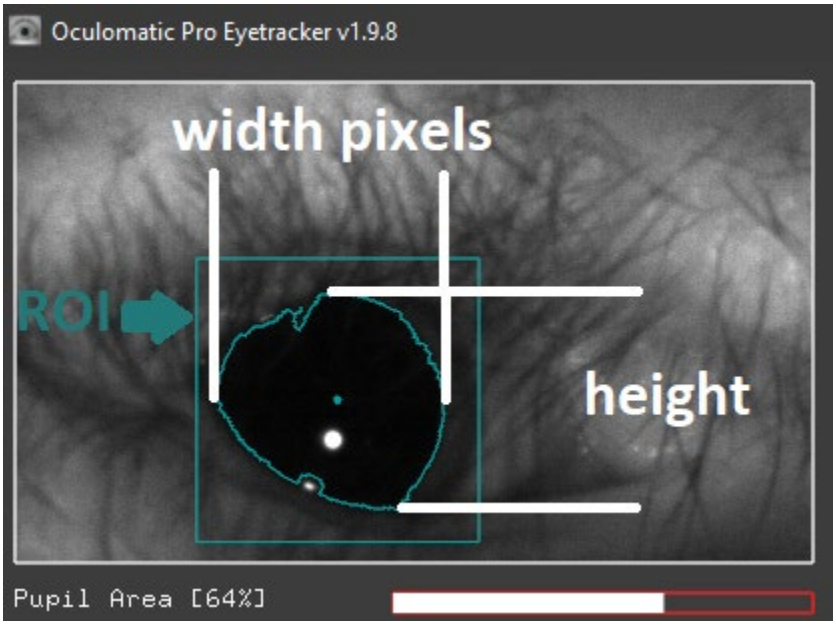


i) Calibration.txt file format
   i) The first line is the number of calibration points
   ii) The following lines each have 4 columns:
      (1) X,Y location of calibration point, followed by X,Y raw gaze when looking at the calibration point.

```
/**********************************

    9
    1.0     1.0     1.0     1.0
    1.0     0.0     1.0     0.0
    1.0     -1.0    1.0     -1.0
    0.0      1.0    0.0     1.0
    0.0     0.0     0.0     0.0
    0.0     -1.0    0.0     -1.0
    -1.0    1.0     -1.0    1.0
    -1.0    0.0     -1.0    0.0
    -1.0    -1.0    -1.0    -1.0

**********************************/
```

   iii) The “Save Calibration As File” option in the Calibration App is only useful if you wish to retain a distinct calibration file for different subjects.
   iv) The settings.xml allows specifying `<Calibration_File>calibration.txt</Calibration_File>`

5) **Pupil Size**



Calculation:
While tracking, Oculomatic Pro computes the maximum width & height of the rectangle bounding the pupil. The area of the pupil is computed as the product of these two dimensions. These calculations are in pixels. The camera samples a 400x240-pixel wide image frame; the pupil may fall within ¼ - ½ this size or approximately 100 x 100 pixel (10K pixels in area). Furthermore, Oculomatic Pro will report the normalized value, i.e., 10K / 96K (100*100/400*240). In later versions of Oculomatic Pro (>= 1.9.7), where the pupil ROI size is adjustable, the denominator will be bounded by the ROI's rectangle area rather than the entire image frame. This increases dynamic range which would otherwise be wasted. The pupil size calculation is performed on each frame along with the gaze estimate (~1KHz).

Best Practices:
In the example shown above, the area of the pupil is reported to be 64% the area of the ROI. Therefore, Oculomatic Pro reports the relative, not absolute, change in pupil size. Since many factors influence pupil dilation, best practices are to:
a) fix the distance between your subject and camera
b) fix the luminance of the environment during the measurement window
c) take measurements when the subject's eye is fixating at the same static target (i.e. center of the visual field)

Obtaining Data:
Oculomatic shares data using Analog Output and through its SDK.

The 4 channel A/D device has a voltage range of -10… 10V, where negative 10V indicates no pupil found and positive 10V indicates 100% of the ROI is filled by pupil area. Therefore, a 50% pupil area outputs Zero 0V and 64% shown in the example above will output +2.8V ((0.64 * 20)/10). Currently, the individual height and width dimensions are not output, but may be added as a future option.

The included DAQ device has a 4-channel output bank, where channel 3 is used for pupil size. The SDK simply outputs the percent area of the normalized pupil as shown in the user interface.

6) **Analog Output**

Oculomatic currently supports both Measurement Computing and National Instruments DAQ devices. We typically supply USB models, but you can use PCIe cards as well. The driver will automatically communicate with most any model.

a) Users can configure the DAQ device by editing the bin/data/settings.xml file, which contains all the startup parameters. Look for the following tags:

```
Option 1 (preferred)
<DAQ>MCCDAQ.dll</DAQ>              // Use Measurement Computing driver
<DAQCHAN>USB-3101FS</DAQCHAN>     // The name of the Measurement Computing device
<DAQRANGE>10-VOLTS</DAQRANGE>     // The voltage range can be +/- "10-VOLTS" or "5-VOLTS"

Option 2
<DAQ>NIDAQMX.dll</DAQ>            // Use National Instruments driver
<DAQCHAN>Dev1/ao0:3</DAQCHAN>     // The National Instruments Device Number and Channels
<DAQRANGE>10-VOLTS</DAQRANGE>     // The voltage range can be +/- "10-VOLTS" or "5-VOLTS"

Option 3
<DAQ></DAQ>                       // Do not load any DAQ device
<DAQCHAN></DAQCHAN>               // Leave this field blank
<DAQRANGE></DAQRANGE>             // Leave this field blank
```

b) The latest version of Oculomatic allows to disable the DAQ in the user interface by clicking on a check mark. This option to edit the settings.xml file remains as a legacy feature.

c) The DAQ Device will automatically stream the X,Y gaze signal through the device's first and second output channels. If you have a 4 channel output DAQ, then the pupil size will streamed on the 3rd channel. The 4th channel is currently reserved for future use. It does not output any signal at this time and should read 0 Volts.

d) The analog output dynamic range is either -10 to +10 Volts (preferred) or -5 to +5 Volts.

e) The instantaneous analog output level is displayed on the Oculomatic user interface along with normalized gaze estimate values.

f) Pupil size is normalized from 0-1 and then dynamically scaled to the voltage range, where no pupil detected corresponds to -10V and a fully open pupil is +10V. **Pupil normalization is computed by using the max pupil area bounded by the ROI. See pupil size section above.**

g) Any 2D Affine transformation will be applied to the analog output voltage as well as shared memory accessible via the SDK.

h) The gain and invert signal controls can be adjusted by the user but only affects the analog output voltage, and not shared memory. Therefore, 2D calibration must be performed if using the shared memory SDK.

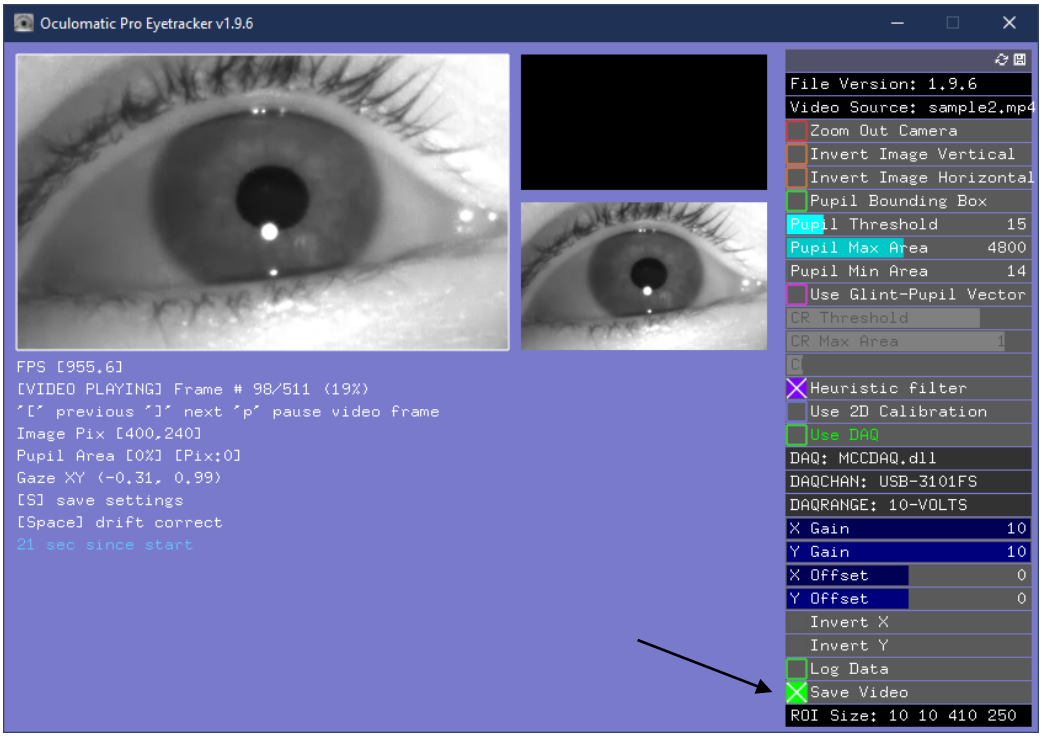
i) All GUI settings can be saved by pressing the 'S" key.

7) **Data sharing**
   a) Data is concomitantly output to shared memory. Any other program running on the same computer can read the x,y gaze and pupil estimates by implementing the Oculomatic Pro SDK. Matlab, C#, and C++ examples are provided in the SDK subdirectory of the distribution.
   b) Sharing data across computers (over the network) is currently not implemented.
   c) Unlike analog output, digital data is supplied as normalized values in the range of 0.0 to 1.0, where 0.5 is the center of the visual display. Analog output however, -10V indicates looking far left, +10V far right, and 0V is dead center. These are estimates.
   d) The current digital gaze signal is also displayed in the user interface.
   e) As with analog output, any 2D Affine transformation, including gain and invert, will be applied to the digital data output.
   f) Gaze x,y, pupil size, and time stamp are logged to file if the option is checked in the GUI. Each time Oculomatic starts up, the log file is over written.
      i) For sophisticated data saving, we leave it up to the user to implement their own functionality. The supplied C++ and MATLAB sample programs demonstrate how to do get data sample by sample, allowing you to process and record this information however you wish.

8) **Saving and playing back video**

a) Oculomatic can save video recording at its native frame rate in a proprietary format. However, videos can only be played back using Oculomatic. They are not compatible with any other video player.

b) To save videos simply press the save video button.



c) Videos are recorded to the directory specified under the <Folder></Folder> value in settings.xml and each file name is automatically generated (such as oculomatic_video_#1.vid). Feel free to move these files to another directory or rename them later.

d) When playing back videos, be sure to enter the correct file path; the final character must be a back slash as shown:

i) `<Folder>data\</Folder>`

ii) `<Folder>data\videos\</Folder>`

iii) `<Folder>data\videos\subject1\</Folder>`

a missing backslash will cause the video to not be found.

e) To play back a video, simply enter its name in the settings.xml file's camera property such as this:

i) `<Source>` **`oculomatic_video_#1.vid`** `</Source>`

ii) The video will be played back in place of the live camera stream.

iii) You may adjust all the threshold settings as if it were a live camera feed.

f) To pause the video at any time, press the 'P' key, or use the "[" or "]" keys to advance or reverse by a single frame.

9) **Logging data**
   a) Pressing the log data button in the Oculomatic UI will cause a file by the name of oculomatic_data.log to be saved in the Bin folder.
   b) File header reads:
      i) xNorm yNorm pupilAreaNorm ABS_TimeStamp_MICROSEC FPS

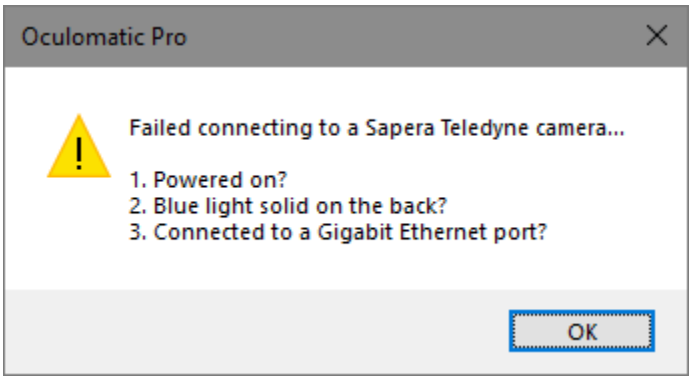
   c) Data may look something like this:
      i) 0.949485846860 0.234902760801 0.590400000000 305231609841 999.04
      ii) 0.949485846860 0.234902760801 0.590400000000 305231609841 999.04

   d) Users can examine this data in MS Excel or any text reader

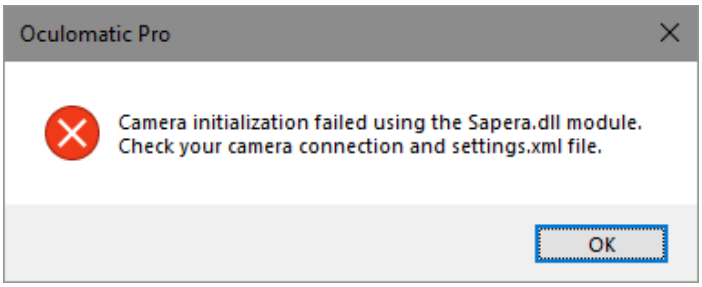
10) **Troubleshooting**

a) Camera disconnects or software freezes when connecting Calibration App or Matlab scripts.
   i) Make sure your computer meets the minimum requirements (graphics card and ethernet port)

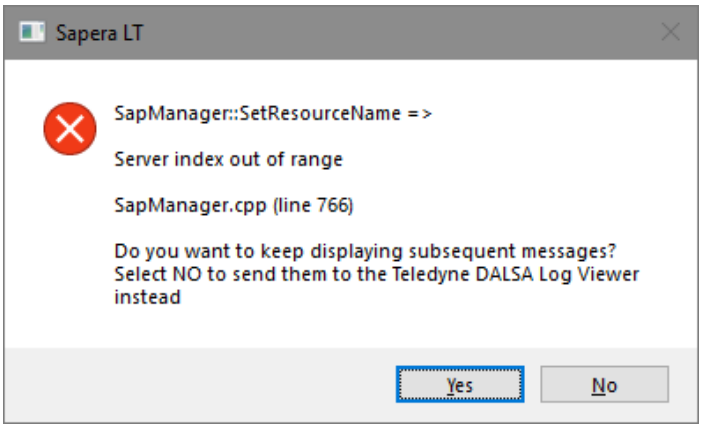
b) Error message(s) stating that it cannot connect to a camera:



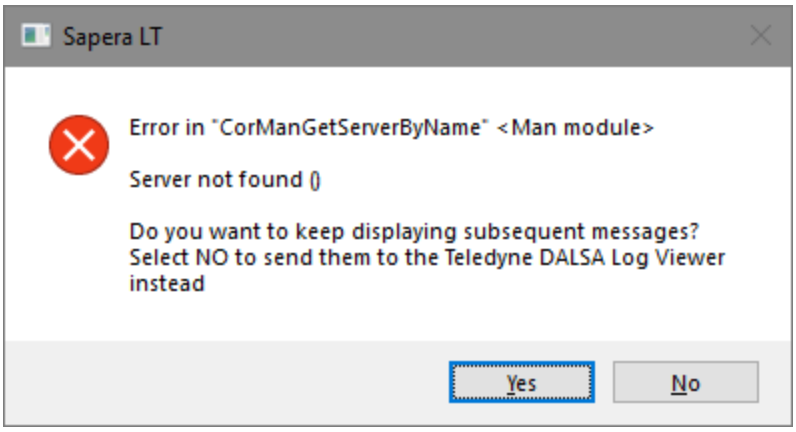
Most likely because 1) the camera's power is off, 2) cable is not securely connected, or 3) you are not using a Gigabit ethernet port.



Most likely because the camera's driver is not installed.
Otherwise, make certain that the settings.xml file indicates <Camera>**Sapera.dll**</ Camera >

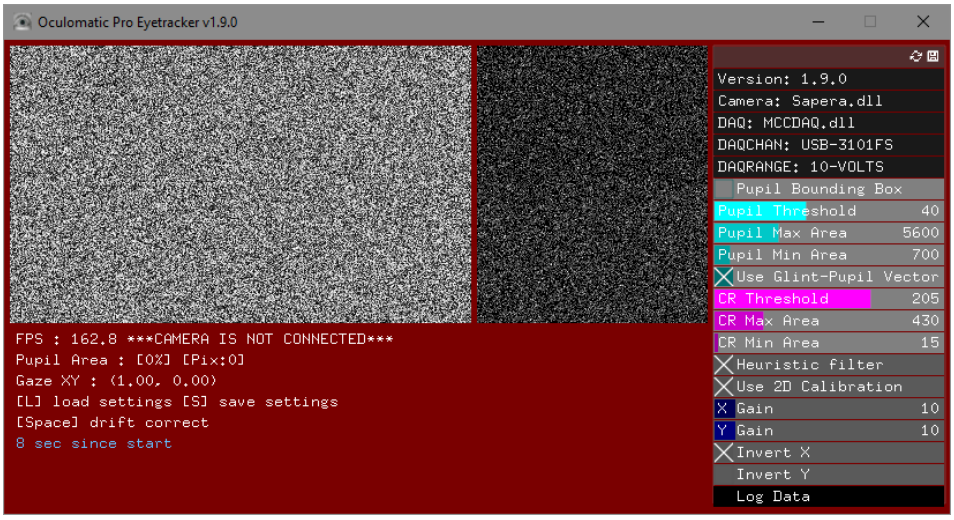


This message is reported by the camera's driver and we can't suppress it. It usually means 1) the camera's power is off, 2) cable is not securely connected, or 3) you are not using a Gigabit ethernet port.



This message always follow's the previous message.

Any Camera error message will likely result in the following window showing noise instead of a video



All these error messages are related to the system not detecting the camera. This can result from an incorrect **physical connection**, **improper software driver** installation, or **incompatible network card**.

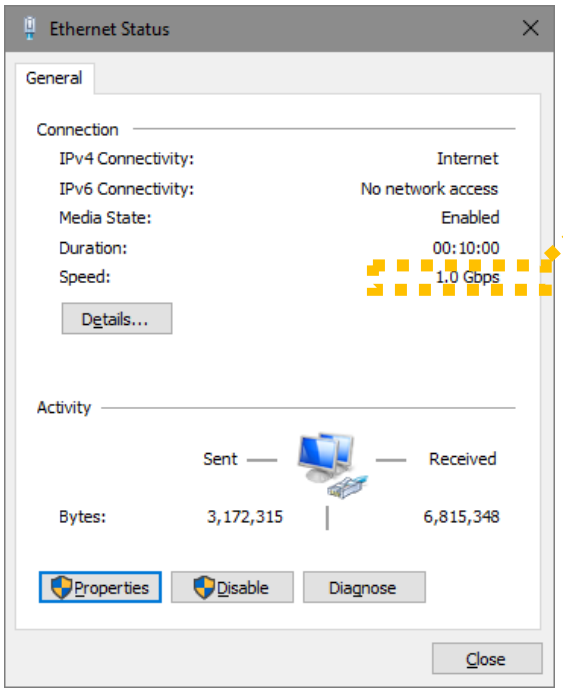
i) Check to make sure the cable is screwed down and the blue light is SOLID as shown in the following figure. This indicates that the camera is getting power and is connected to your computer correctly. Remember it must be connected to a gigabit ethernet card (or a USB3.0 to gigabit ethernet dongle).

FLASHING BLUE light suggests the camera is NOT properly connected. Most likely because it is *not* hooked to a gigabit ethernet connection, or the cables are not plugged in properly.

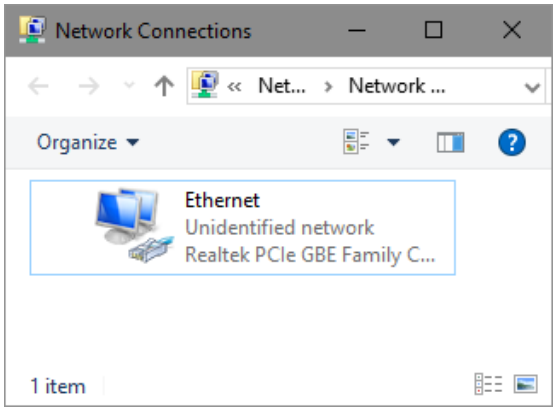
ii) After you start Oculomatic, this LED light will then switch to a green color and blink rapidly.

FLASHING GREEN light indicates Oculomatic Pro is receiving data from the camera

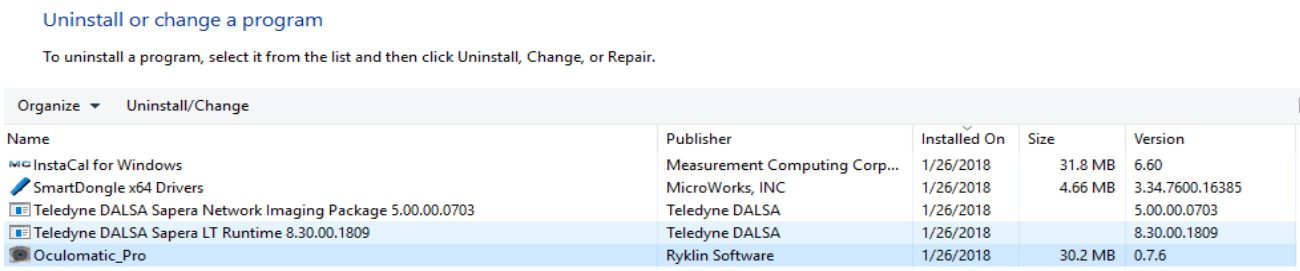
iii) Check the ethernet status, make sure that it is enabled. The **Speed must indicate 1.0 Gbps** as shown below**.**



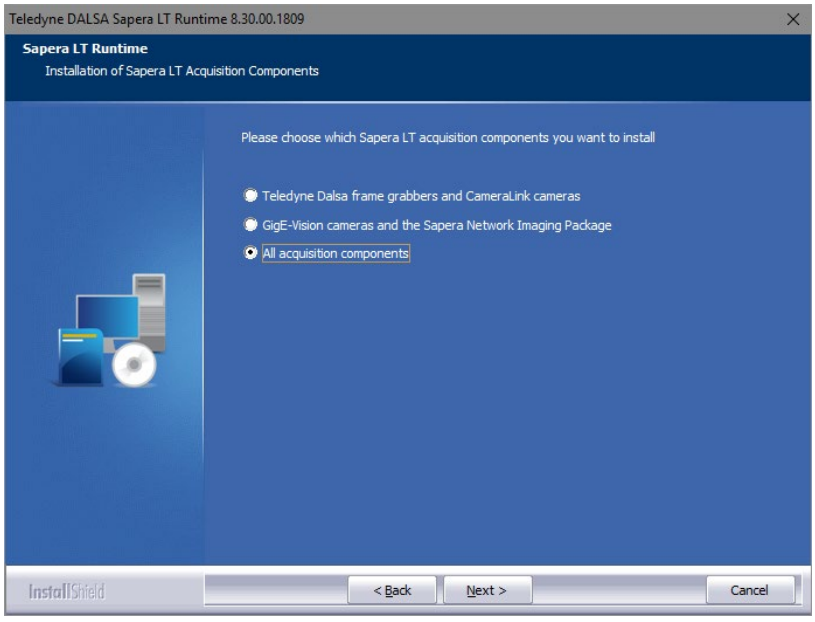
iv) Final confirmation that the cable is connected



v) Check in the MS Windows Control panel that the Sapera Driver is installed (as well as all other software as shown in the following screen shot). There are two components, the LT Runtime and Network Imaging Package. If they are missing, you will need to reinstall them. This might happen if you did not select "All acquisition components" as shown in the next step below.

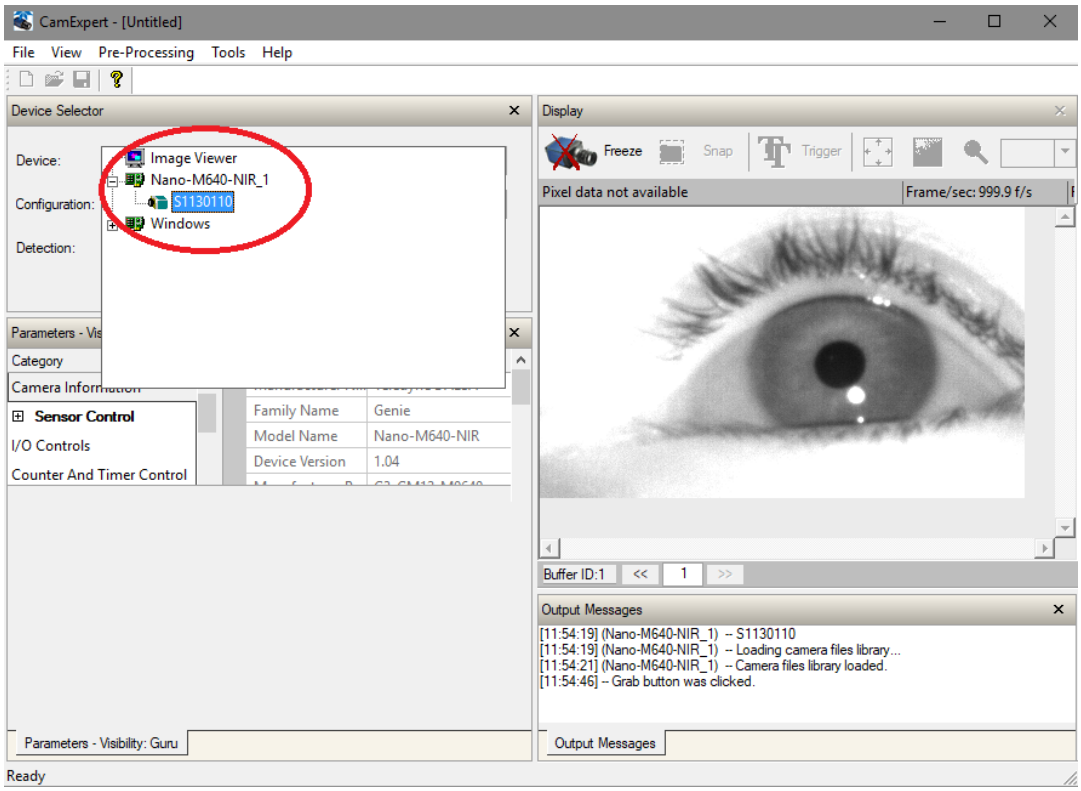


vi) If re/installing the camera driver, you must select "All acquisition components" when running the Sapera driver as shown below and restart your computer.

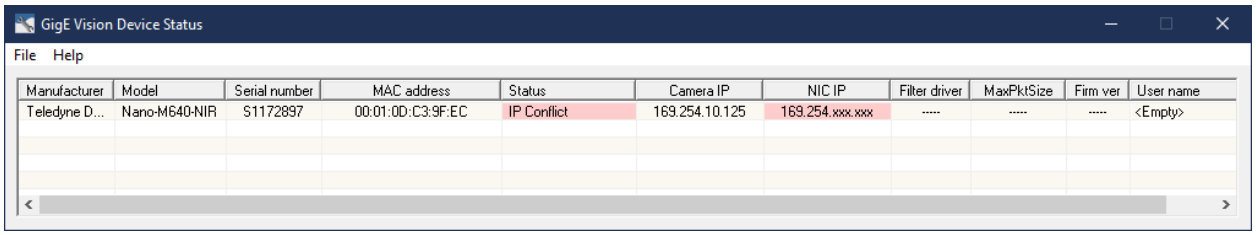


vii) Run the Sapera Cam Expert utility to check if the camera is detected, as shown below. The camera model is displayed once connected (NANO-M640-NIR_1) and below it the serial number. If this is not shown, then repeat steps above in section 10.b

viii)



ix) Do you have an IP conflict?



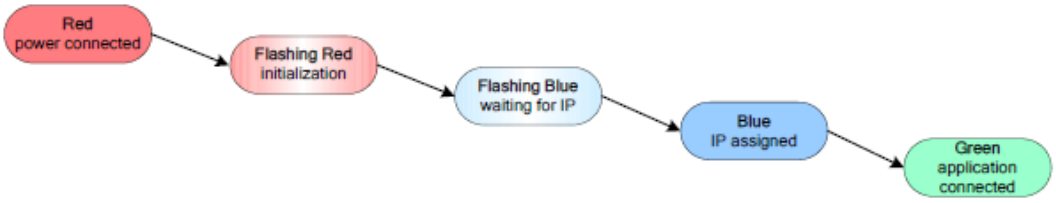
This happens if connecting an ethernet printer or other ethernet device directly to your computer. Please disconnect this device from your computer and use a proper router instead.

LED State	Definition
LED is off	No power to the camera
Steady Red	Initial state on power up before flashing. Remains as steady Red only if there is a fatal error. Camera is not initialized **
Flashing Red **	Initialization sequence in progress Wait less than a minute for the Nano to reboot itself.
Steady Red + Flashing Blue	Fatal Error. If the Genie Nano does not reboot itself contact Technical Support.
Slow Flashing Blue	Ethernet cable disconnected. The camera continuously attempts to assign itself an IP address.
Fast Flashing Blue	File Access Feature is transferring data such as a firmware update, etc.
Steady Blue	IP address assigned; no application connected to the camera
Steady Green	Application connected
Flashing Green	Acquisition in progress. Flashing occurs on frame acquisition but does not exceed a rate of 100ms for faster frame rates.

Note: Even if the Nano has obtained an IP address, it might be on a different subnet than the NIC it is attached to. Therefore, if the Nano LED is blue but an application cannot see it, this indicates a network configuration problem. Review troubleshooting suggestions in the Network Imaging manual.

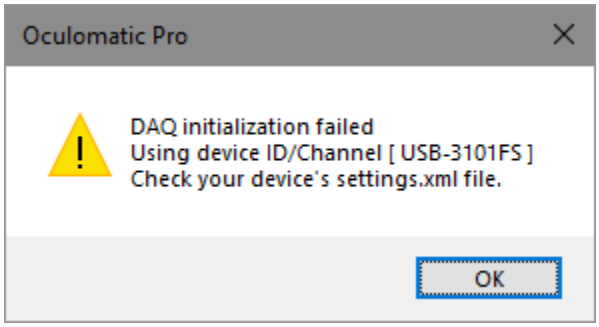
LED States on Power Up

The following LED sequence occurs when the Genie Nano is powered up connected to a network.



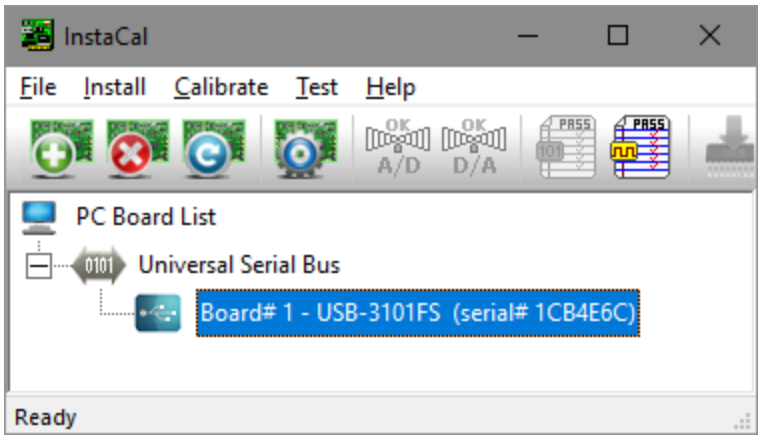
c) **Analog output does not work.**

i) If you get an error message such as:

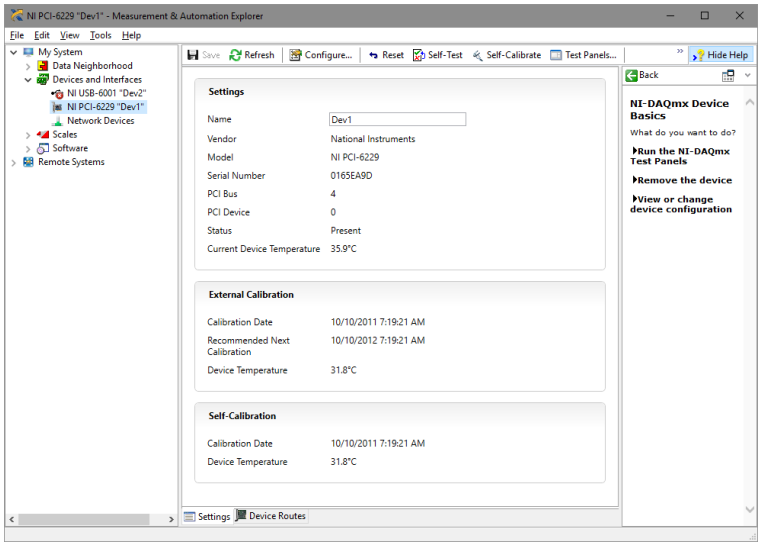


(1) Make sure the corresponding DAQ driver is installed.

(a) "MCCDAQ.dll" is for Measurement computing. You only need the "InstaCal" component of the driver to be installed.



(b) "NIDAQMX.dll" is for National Instruments. You MUST have "Measurement and Automation Explorer" installed as part of the driver.



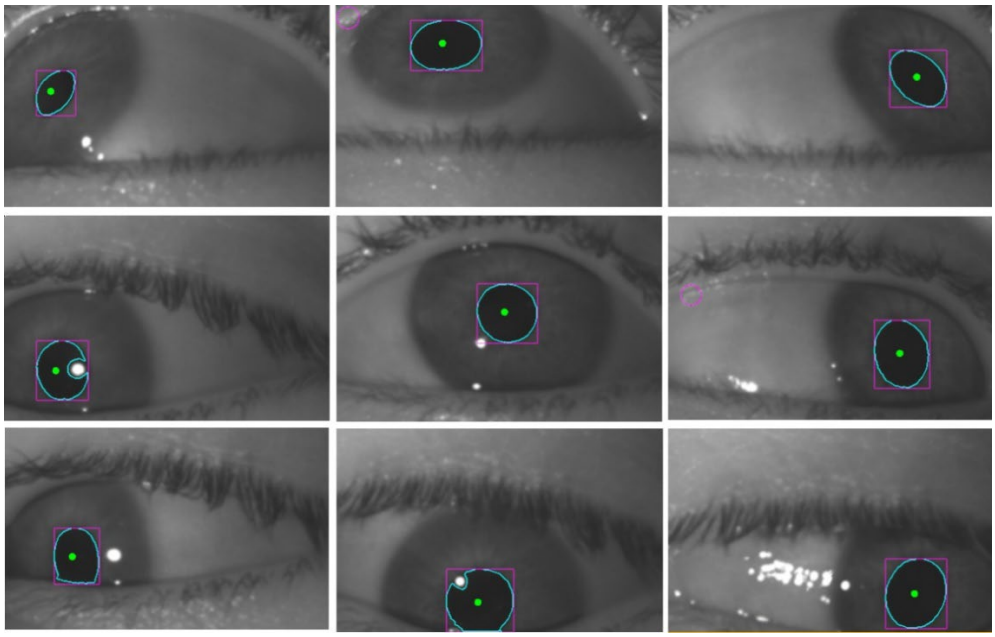
(2) Do you have the correct configuration in the settings.xml file? See above in SECTION 8, for the various DAQ devices you can configure to use with Oculomatic.

ii) Check that the correct BNC wires are connected and grounded.
(1) Horizontal/Ground
(2) Vertical/Ground
(3) Pupil Size /Ground
(4) Reserved/Ground

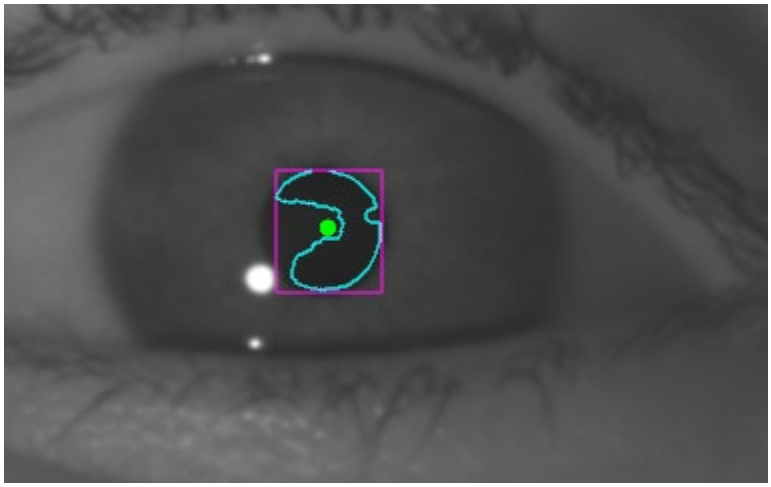
d) **Noisy or poor-quality gaze signal**:

i) A video demonstration on focusing the camera and lighting is available on youtube.com:
(1) https://www.youtube.com/watch?v=3e5iF1TzWAw

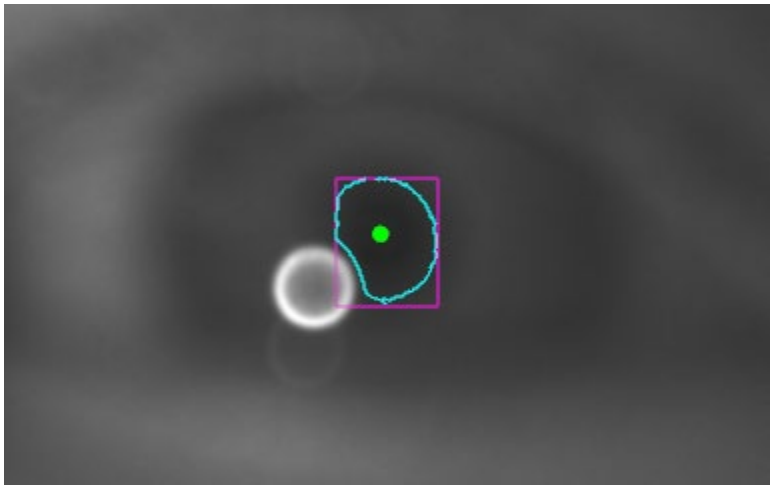
ii) Shown are 9 cardinal viewing angles. The pupil is clearly thresholded in each direction *BUT* the Glint is missing in the top right, top center and right middle images. The most likely causes are a) the IR emitter is not optimally positioned relative to the eye, and b) the eye is looking to far to the side (rotated too obliquely). This will cause bad tracking when using the 2D calibration feature.



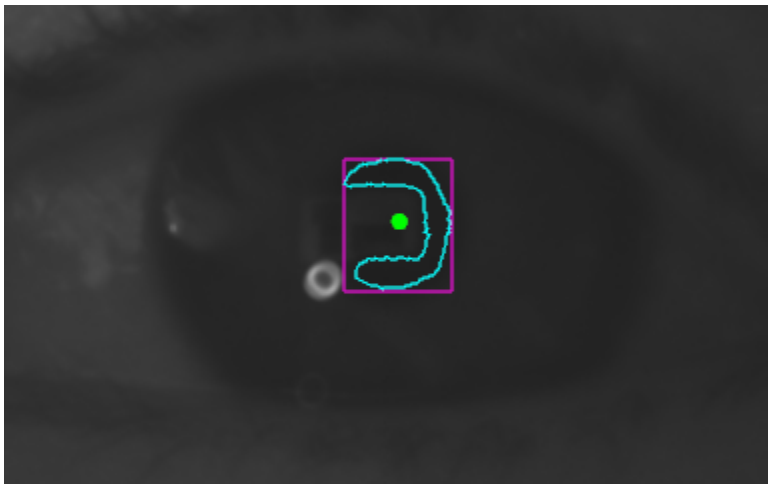
iii) Example of bad pupil threshold. Adjust the pupil threshold parameter. Otherwise, the lighting and focus are within tolerances.



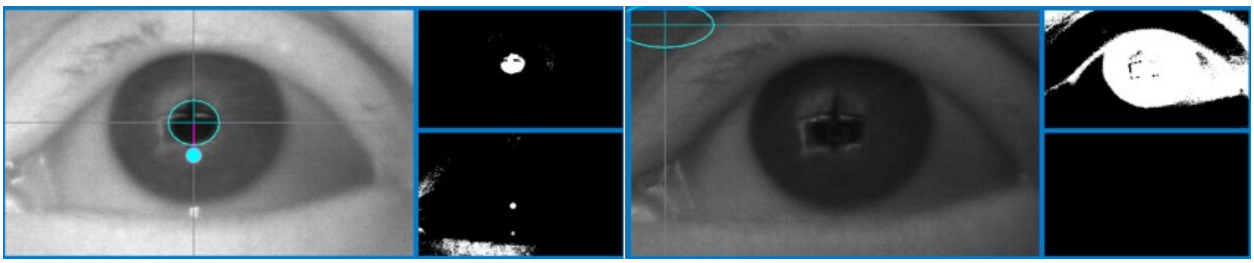
iv) Example of bad focus. Adjust focus on the camera lens.



v) Example of bad lighting. Is your IR Emitter plugged in and facing the eye? Is the camera's aperture opened?

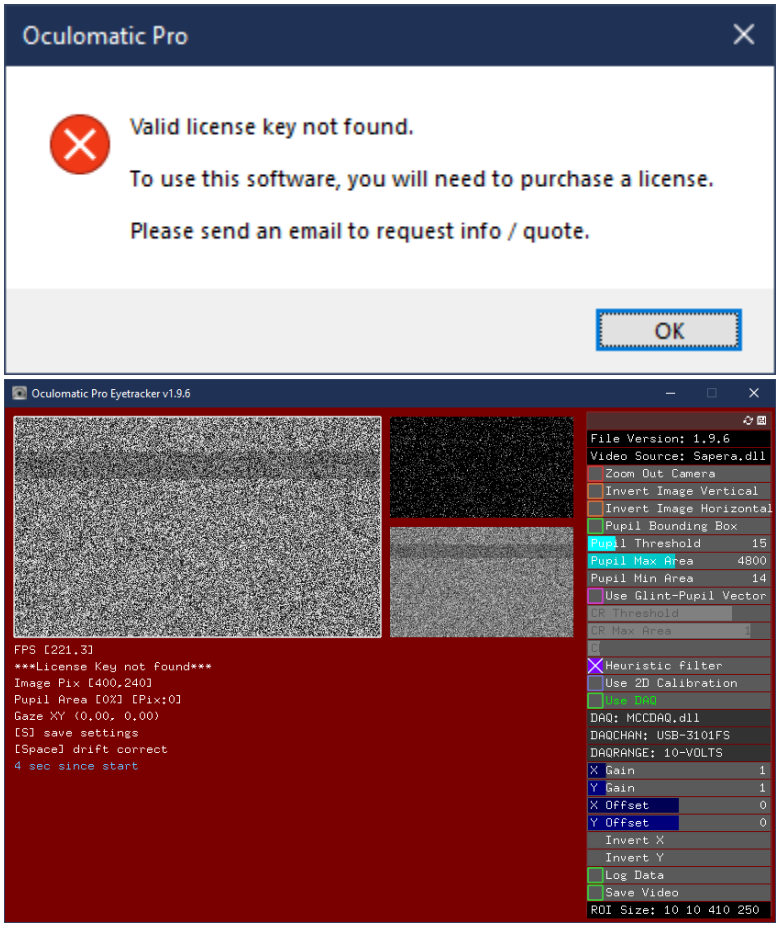


vi) Left image : Good IR Light
Right Image is the same frame, BUT no IR Light. Make sure the IR emitter is turned on. Is the lens aperature opened?



vii) Bad glints, when using the PC-CR vector

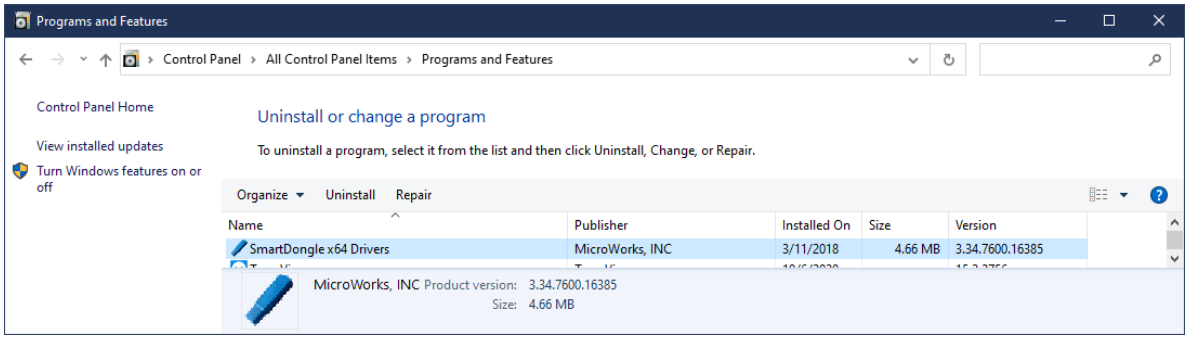
e) **Valid license key not found message.**



This results in the above noisy screen instead of a camera image.

i) Check that the USB Key is plugged in, and the LED light should glow green.

ii) Check in control panel, that the smart dongle license key driver is installed

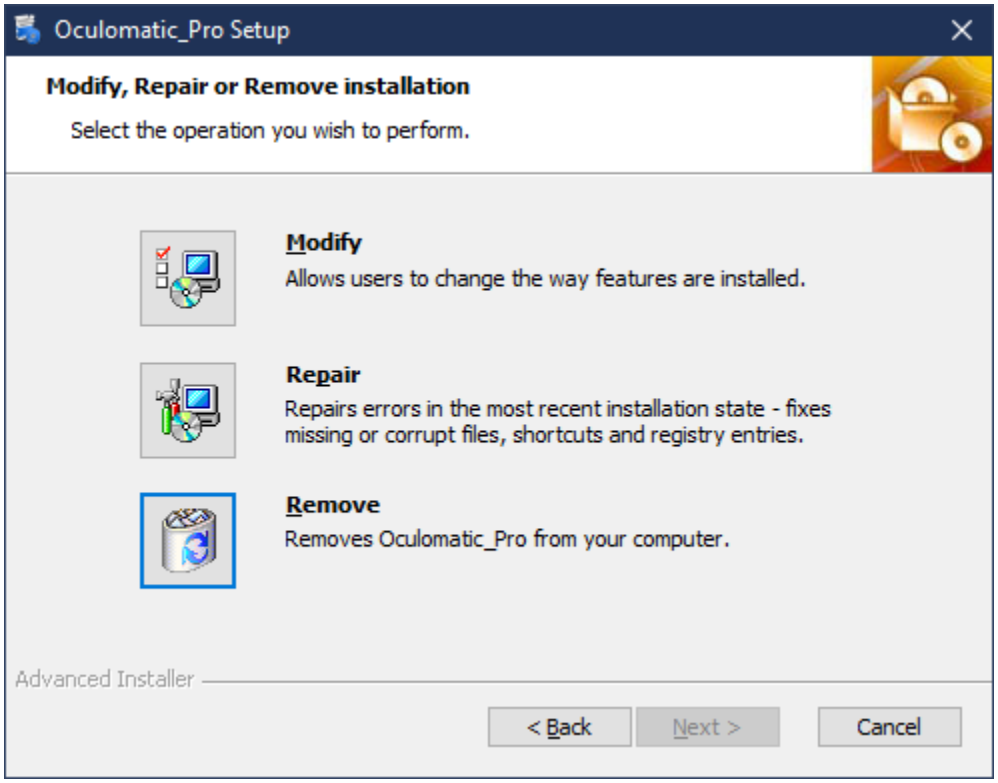


f) **Revert to previous software version**

i) Run the installation program that corresponds to your software version and choose the remove option. This may take a several minutes.

or Run the

and choose “Remove” as shown below.



ii) After removing the software, run a different installation program to revert to a previous software version. Note, you do not need to reinstall any drivers or run the “full” installation program, just the update installer.
iii) You do not need to restart the computer.