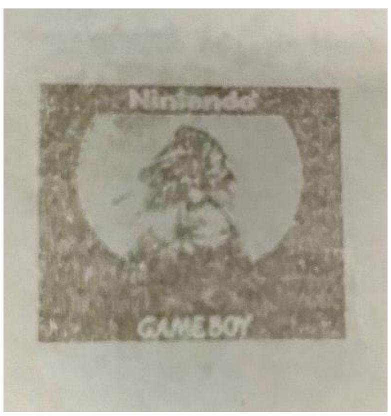
Raspberry Pi Microscopy – My Experience

By Jim Harper, Amateur Microscopist

As a former professional, but now an amateur microscopist, I've used many different camera combinations to record images. In the early 1980s I found success with film cameras, in spite of the significant cost and time to process 35 mm films. Polaroid instant prints and 35 mm instant slides achieved fast results for many applications, but the high cost was a factor. Analog video cameras and three-color video printers gave quick results but poor quality—yet enough to tell a story.

Digital cameras, with ever-increasing resolution, provided the tools to expertly document my observations at a low cost. I've tried web cameras, fixed lens digital cameras, and SLR digital cameras. I've even managed to photograph a flea using a Nintendo Game Boy camera (but only once!). All these attempts involved finding a mount to couple the camera to the microscope, optics to get the image from the microscope to the camera sensor, and a way to transfer the digital files to a computer all whilst keeping magnification, field of view, and image quality under control in a reproducible fashion. My professional interests were primarily materials and fibers, in particular. Dimensional calibration of the microscope is a primary concern since the microscope is a measuring tool for me, in addition to generating images to tell a story.



Flea as taken with a Nintendo Gameboy / Leitz Dialux Pol
and Motic X1 camera and Nikon H Field Microscope

The purpose of this article is to describe my favorite solution to date which uses a Raspberry Pi High Quality Camera in my home lab. At the onset, I should say I know very little about Raspberry Pi programming. At the Raspberry Pi official site, I was able to learn how to load the operating system for a Raspberry Pi 4 and utilize the Pi HQ camera. The current US price for the camera is about $50 and the Pi 4 is about $50. I use the standard Pi operating software, and I type the camera commands at the command prompt to view and record images. Video or still photos can be created. I have three primary microscopes for different purposes, and I interchangeably use the camera arrangement for each scope.

The Lab

I have an Olympus Vanox-T AH2 that was an Ebay "accident". A ridiculously low bid unexpectedly secured a scope that has been fun to explore and offers future challenges. The Pi camera and screen normally resides on this scope and a wireless USB keyboard for the Pi 4, attached with a bit of Velcro, has taken the place of the rather sophisticated camera control that these scopes were known for.

The second microscope is a Leitz Dialux Pol that I purchased from my employer after a corporate buy out. It is my fiber "work horse" scope that I use for measuring fiber birefringence and refractive index. I use it to measure fiber size and thermal properties using a hot stage, as well.

The third microscope is a Heerbrugg Wild EPI reflected light microscope. This was part of an estate sale that gave me the capability to measure packaging film cross-sections and understand the construction of multilayer films.

The Raspberry Pi Camera System

I use a 7" touchscreen monitor with Raspberry Pi 4 mounted to the back of the monitor. There is also a HDMI video-out from the computer that will attach to my 40" TV/monitor. There are four USB connections, and I use an USB-HDMI adapter to transmit a live image to my desktop computer. A 3D-printed holder secures the little monitor and computer to the top of the microscope.

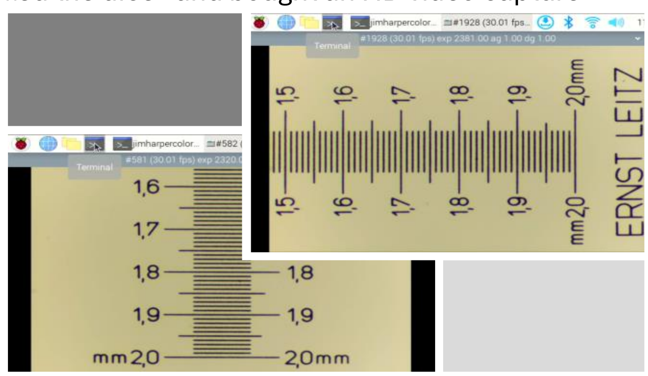
The Raspberry Pi HQ camera has a standard C-Mount adaptor which I was conveniently able to screw onto the video camera port of the Olympus Microscope. The Olympus came with an Olympus NFK 2.5x LD projection lens to project the image directly onto the CCD of the Pi Camera. I added a small 3Dprinted cover to project the electronics of the camera. A 1-meter ribbon cable connects the Pi Camera to the computer—long enough to reach the other scopes. 3D Printed adapters to the Leitz and the Wild microscope let me move the Olympus mount from scope to scope.

A Logitech USB wireless keyboard is used to interact with the computer. It has a touchpad and keyboard. Velcro secures the keyboard where the camera controls for the Olympus scope reside, Velcro allows me to still get to the camera controls if I need them.

Since I know nothing about Raspberry Pi programming, I just followed the instructions from the official Raspberry Pi website to download the operating system and installed it on the SD card of the Raspberry Pi. A web search led me to the camera controls of the operating system that can be entered from the command prompt. Some examples of the commands are included in the sidebar at the end of the article.

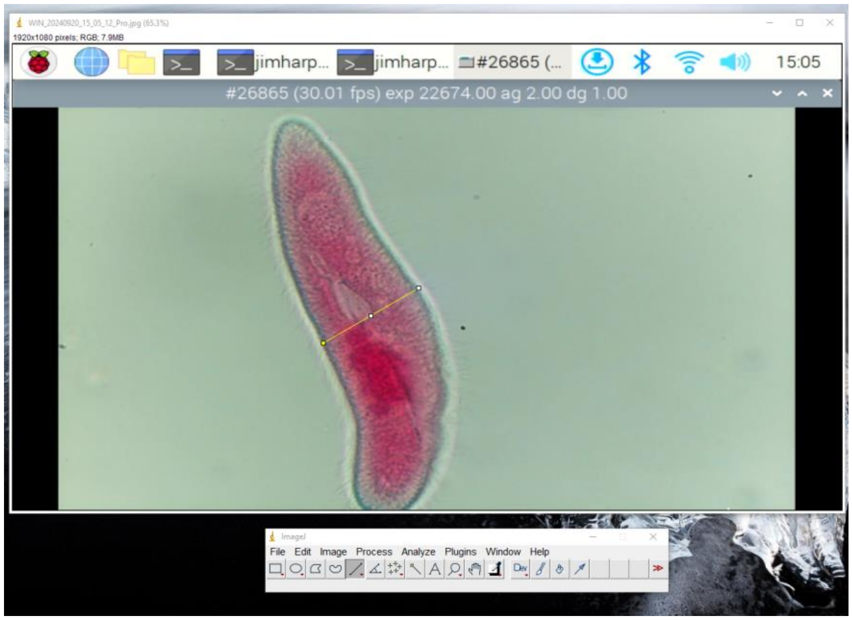
This arrangement is complete for making photographs that are stored on the SD card and can be transferred manually to my desktop computer by copying to a USB thumb drive. I use ImageJ software to analyze images on my desktop, mainly to calibrate and make measurements. There is a version of ImageJ that runs under the Pi Operating system directly on the Raspberry Pi, but I have not investigated this ability yet. You can find the Windows version at the ImageJ.net website. There are also instructions to install ImageJ directly on a Raspberry Pi here.

I wanted to have an interactive connection between the Raspberry Pi HDMI output and software on my computer which would allow me to more easily measure images, add images to documents, or even use the scope for live ZOOM meetings. I "rolled the dice" and bought an HD video capture adaptor (HDMI out to USB input) and a 10-foot USB cable to reach my desktop. Low and behold, it worked. Windows includes a "Camera" application in the "Start Menu" button. On a lark, I ran the Camera application and was able to select the USB camera (the Pi Camera HDMI output) and voila, the connection worked. I opened ImageJ and selected the USB camera. Once again, everything worked.



The calibration process uses an image of a scale calibration slide in both horizontal and vertical directions. The camera magnification factor for the Pi Camera is significantly different in the x and y directions. Fortunately, calibrating the ImageJ software automatically adjusts for the camera magnification factor once it is calibrated for a particular objective / projection lens and camera position on the microscope.

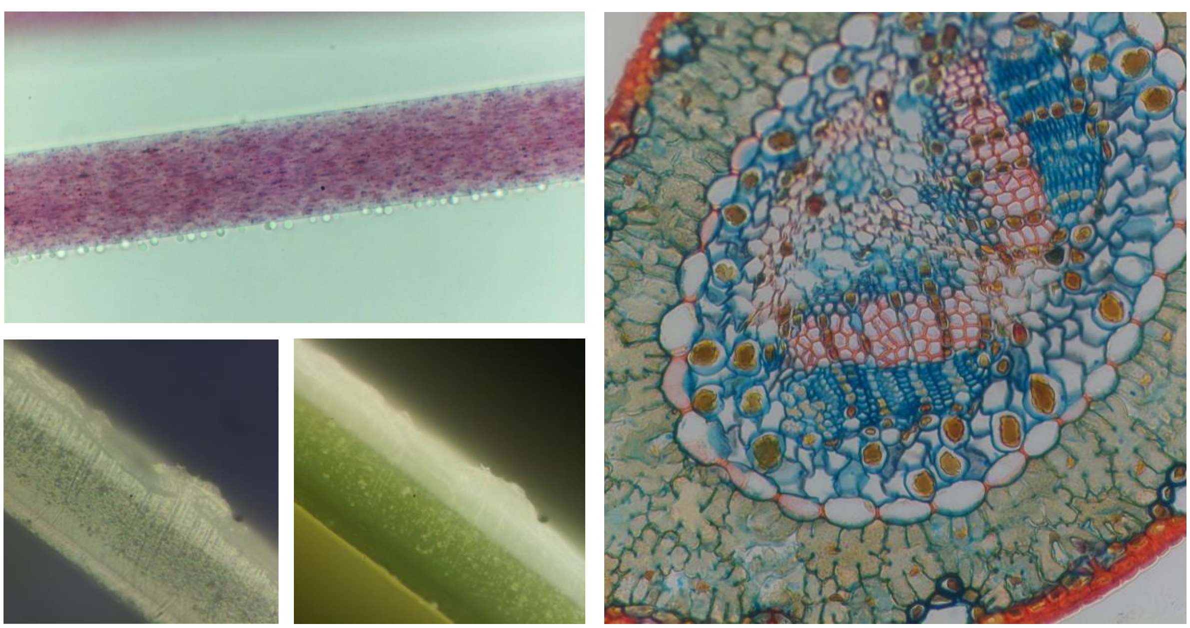
It is beyond the scope of this article (and the author!) to explain all the functionality of the ImageJ software. I just use the basic functions that I need. This software is open source and has been around forever and there are many on-line articles about it.



Conclusion

After many attempts to adapt a versatile camera to multiple microscopes, I've settled on the Raspberry Pi for now. I can easily move it to my work horse scopes and I have free software to do the simple measurement tasks I need. The cost is very reasonable and even without a knowledge of Pi programming, I can make it work. My future ambition for the set-up would be to find or create a visual, front-end software tool that controls the Pi camera. I would also like to install ImageJ on the Pi Computer directly.

I've included a my "cheat sheet" for camera commands to be entered at the command prompt of the Raspberry Pi operating system. I've also included a list of items I've used to build this camera system.



Raspberry Pi Camera Images
Upper Left: PP pigmented (Pigment Red 254) fiber – Leitz Dialux Pol. Right: Pine cone cross-section – Olympus Vanox.
Lower Left: Wild Dark Field Epi view 3-layer film cross-section (clear / green pigment / white). Lower Right: Bright Field

Addendum

PiOS Camera Commands Examples
(Type these at the command prompt.)

libcamera-hello (this command will verify that the camera is working and briefly show you a continuous image from the microscope)

libcamera-hello -t 100000 (this command will verify that the camera is working and show you a continuous image from the microscope for 100 seconds, increase or decrease number to adjust time)

libcamera-jpeg -o test.jpg (this command takes a photo from the microscope camera and saves a file named "test.jpg".)

libcamera-jpeg -o test.jpg -t 5000 –width 1920 –height 1080 (this command shows a 5 second preview the takes a photo with and image size of 1920 x 1080 and saves to "test1080.jpg".)

libcamera-still -o still-test.jpg (this command takes a still picture and saves to "still-test.jpg"

libcamera-still -e png -o still-test.png (this command saves a png file to "still-test.png".

libcamera-still –datetime (this command captures and image at a certain time and saves as a jpg file using the MMDDhhmmss date format as a filename

libcamera-vid -t 100000 -o test.h264 (this command takes a 10 sec video and saves to test.h264 format)

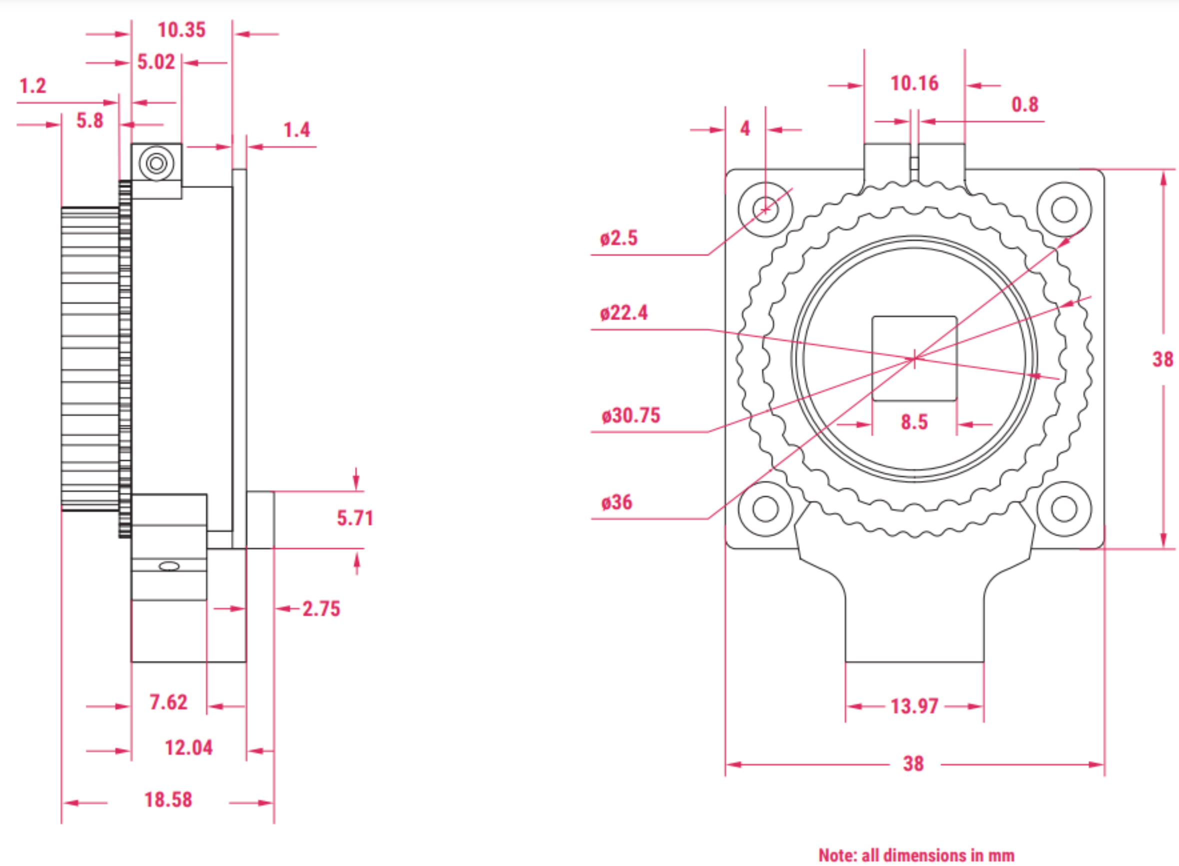
Source: https://www.tomshardware.com

Raspberry Pi Microscopy

- Raspberry Pi 4
- Raspberry Pi Camera
- 7-inch touch screen monitor (top of microscope)
- USB Keyboard
- 128 GB Micro SD card
- 1 meter camera cable
- Olympus NFK 2.5 x LD (125) projection lens
- Simple 3dp camera electronics dust protector
- Power supply for Raspberry Pi
- HDMI cable to connect to large screen monitor
- Long USB cable
- HD video capture adaptor (converts HDMI video to a USB output for separate computer)
- Windows 10

Raspberry Pi HQ Camera Specifications
https://www.raspberrypi.com/documentation/accessories/camera.html#hardware-specification

- Sensor: 12.3 megapixel Sony IMX477R stacked, back-illuminated sensor
- Image size: 6.287 mm x 4.712 mm and 7.9 mm diagonal
- Sensor resolution: 4056 x 3040 pixels
- Video Modes: 2028 x 1080p50, 2028 x 1520p40, 1332 x 990p120
- Pixel size: 1.55 μm × 1.55 μm
- Optical size: 1 / 2.3 in
- Focus: Adjustable
- Max allowable exposure time: 670.74 seconds
- Lens mount: C / CS or M12 Mount
- Output: RAW12/10/8, COMP8
- Back focus: Adjustable (12.5 mm–22.4 mm)
- Lens standards: C-mount and CS-mount
- IR cut filter: Integrated, but can be removed to enable IR sensitivity
- Ribbon cable length: 200 mm
- Tripod mount: 1/4”-20
- Size: 38 × 38 × 18.4 mm (excluding lens)
- Weight: 30.4 g



Comments to the author welcomed, email - jimharpercolor AT aol DOT com