Portable Microscope

Austin Bryant, Cayla Gill,
Hannah Pierson

University of Central Florida College of Electrical Engineering and Computer Science in partnership with the College of Optics and Photonics, Orlando, FL.

***ABSTRACT* – The objective of this project is to build a microscope that is capable of wirelessly transmitting its video feed to a mobile device at frame rates exceeding thirty frames per second. Unlike most microscopes currently on the market, our aim was to design a microscope that was both wireless and portable, such that the user can move around and use the microscope freely without any restrictions.**

I. Introduction

Microscopes are a fundamental part of any science education because they are a useful tool in examining specimens that are too small for our own eyes to resolve. Students and researchers alike use them all the time, but traditional microscopes are tethered with a power supply, utilize an eyepiece that must be refocused for each individual user, and the specimen is usually encased in a slide. We have changed all of that by creating a portable, wireless, and digital microscope.

Our microscope is equipped with a base light for thin specimen and a top light for thick specimen. It boasts a tested 4X and 10X optical magnification for viewing a wide variety of samples. The microscope is also wireless by using Lithium-ion batteries that power both the Raspberry Pi and the light sources.

Figure 1 – Microscope Comparison

The Raspberry Pi is used with an OV5647 sensor to capture and transmit images to an application on your android device. This allows more people access to the microscope in a more efficient way so that more specimen can be studied in less time. An image comparing the layout of our microscope to a traditional microscope is shown above in Figure 1.

II. Components

The following section details the selection of our components and our justification for choosing them.

A. Optical

Objective Lens – For our microscope we chose to use a set of AmScope DIN standard lenses. We chose these lenses, instead of infinity corrected objective lenses, because they focus an image onto a plane without the need of an additional lens. They are also more economical at $20.00 per lens compared to $120.00 for an infinity corrected objective lens.

Sensor – The sensor we chose is an OV5647 made by OmniVision. It is a 5MP camera with a 1.4 um pixel size and it is full color. This sensor was chosen because it is already formatted to work with the Raspberry Pi module.

LEDs – We are using 5mm diffuse white LEDs for both the base light and the top light. We chose these for their brightness and their availability.

Diffuser – The base light utilizes a diffuser that was custom made using a ring of Delrin plastic and a piece of circular opaque plastic screwed into the ring. This was made from scrap pieces of Delrin and a recycled milk jug.

Focusing system – In order to focus the objective lens onto the specimen, a vertical translation system was required. To build this, we used scrap acrylic, steel rods, an ACME threaded rod, and a custom flange that matched the threaded rod. We chose the acrylic because it was readily available and graciously donated to us. We used steel rods for their sturdiness. We chose to use the ACME threaded rod with matching flange nut for their precision and durability.

B. ELECTRICAL

Easy EDA – This program was chosen for the creation of the printed circuit boards for this project. It was an easy to use, web-based EDA with design rule checking (DRC) error systems that verified that no trace or connection errors occurred.

JLCPCB – This website was used to order the printed circuits boards. This website was linked to Easy EDA, which made ordering boards easy. The boards were made quickly and from high quality materials.

Batteries – Two 3.7V Lithium-ion batteries were chosen to power this system. These would provide portability and compactness to the structure. A 9V Li-Ion battery was chosen for the base light.

Regulator Components - For the printed circuit board design of the regulator, small packaged, surface-mount components were used in order to reduce the size required for the regulator.

Lighting Components – Through-hole components were used for these printed circuit boards due to their availability and their ability to be soldered easily.

C. MOBILE APPLICATION

Android Studio – Android Studio is the integrated development environment (IDE) of choice for Android development. We chose to develop solely for Android because our initial design choice included using Wi-Fi direct, which iOS does not support.

SQLite – SQLite was selected because we wanted to use a database that can handle a significant amount of data without the overhead that remote databases come with. SQLite was the best option for this project because of vast array of documentation and easy-to-use software development kit (SDK) was available.

OpenCV – The OpenCV SDK was chosen as the image processing library for this project because it is one of the well-known image processing libraries currently available. Additionally, OpenCV is simple to set up and use out of the box and offers a wide variety of image processing options that can used.

Dnsmasq – Dnsmasq is a software library that provides network infrastructure for small networks: specifically, domain name system (DNS) and dynamic host configuration protocol (DHCP). It was designed to be lightweight and have a small memory footprint which is why it was selected for this project.

Hostapd – Hostapd is a user space daemon for creating access points and authentication servers. For this project, it was used to create our access point on the Raspberry Pi and was selected because of its simple configuration process and ample documentation.

Raspberry Pi 3 Model B+ – The Raspberry Pi was selected for this project because we were familiar with the device and had prior experience working with it. Additionally, this model of the Raspberry Pi has a 1.4GHz quad-core processor, supplying more than enough processing power for us to effectively capture and transmit images from the camera module.

III. OPTICAL DESIGN

Imaging System – To obtain an image we created a system consisting of an OmniVision OV5647 image sensor that is mounted to a 151 mm tube made from black Delrin. The other end of the tube has been threaded to allow for a DIN standard objective lens to be attached.

To provide adequate lighting we have installed two separate light sources. There is a top light that consists of 9 white diffused 5mm LEDs. This light source has two levels of brightness, one that uses 3 LEDs and one that uses 6 LEDs. These are controlled by a switch. The top light is used for samples that are thick and is best used with the 4X objective lens. The base light is a separate light source that is made of 7 white diffuse 5mm LEDs and a diffusion filter. The diffusion filter was custom made from black Delrin plastic and recycled opaque plastic. This system is shown in Figure 2.

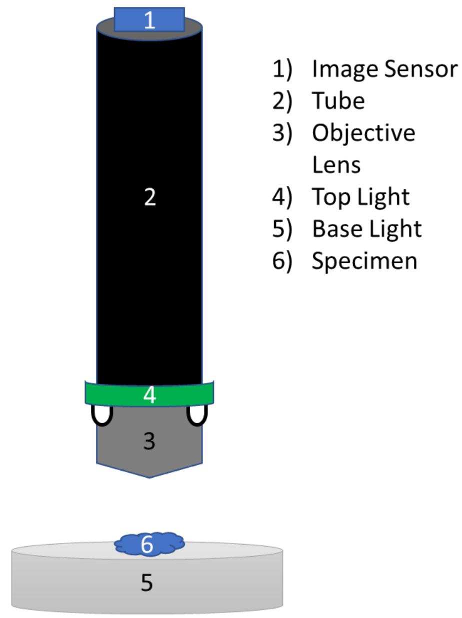

Figure 2 – Imaging System

The specimen is first illuminated by one the light sources. The light source is chosen based on the thickness of the sample. For thick specimens that light cannot pass through, the top light is used. It is important to note that when using the top light, the 4X objective lens should be used. If a larger objective lens is used, the length of the lens will cast a shadow and the specimen will have insufficient lighting to produce an image. For this system, the light within a cone of 5.74° will reflect off the sample and be collected into the objective lens. The objective lens will then focus the bundle of rays onto the image sensor. The final image is 560µm x 840µm regardless of specimen.

For thin specimen, use of the base light is recommended. The base light provides an evenly distributed illumination of the sample. This can be used with either objective lens. The speed of the light will change depending on the refractive index and thickness of the portion of the sample it's passing through. These minor variations will be collected by the objective lens and then focused onto the image sensor.

Focusing System – Our focusing system is custom made so that it can translate vertically. This was essential for our project so that specimen of various sizes could be studied. This also allows us to be able to change objective lenses so that different levels of magnification can be observed.

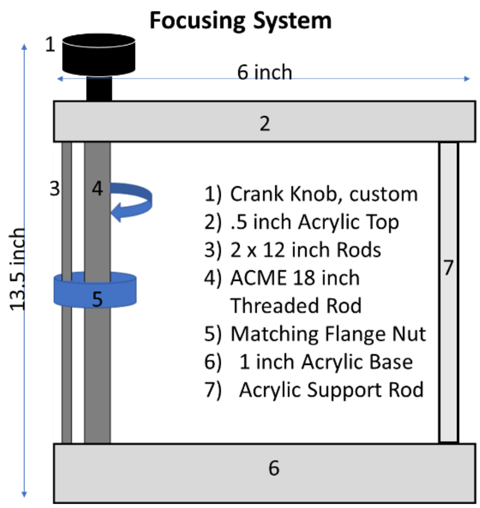

Figure 3 - Focusing System

We were able to build this system by attaching a knob to one end of a threaded rod. The other end of this rod was attached to a ball bearing, so that it can rotate. We then put a flange nut onto the threaded rod. One side of this flange nut is attached to two guide rails so that it will not rotate with the threaded rod, instead it will move up and down as the rod is being rotated. These components were all embedded into a base and top plate of clear acrylic. We then attached two acrylic rods to the opposite side to support the top plate. This system is shown in Figure 3.

IV. Electrical Design

Battery System – For this design we chose battery power in order to ensure portability, maintain a compact design and also to remain as lightweight as possible. In order to stay convenient, it was important that the batteries be rechargeable as well. Lithium Ion batteries were chosen due to their high current capacity, durability, compactness and low self-discharge. Two 3.7V Li-Ion batteries were chosen to have a 7.4V total output and 3000mAh current capacity. Along with the 3.7V Li-Ion batteries, a 9V Li-Ion battery was chosen for the base light system. This was chosen for portability and accessibility for the base LED system.

Voltage Regulator – The regulator chosen for this design was a step-down switching voltage regulator that was used to bring the battery input of 7.4V down to a 5V output. This 5V output was necessary to power the Raspberry Pi as well as the lighting system needed for this design. A switching regulator design was chosen in order to increase the battery life of the system as well as ensure a consistent power output.

Lighting Systems – Two lighting systems were needed to ensure that the sensor used was able to process the images being projected upon it. For this system, there was a top light present with options for both three and six diffused LEDs. Having these options would allow the user control over lighting of the sample. This top lighting system is powered by the 5V output from the voltage regulator that is present in the system.

There was also a base light present that was useful for viewing slide samples. This system included seven circularly arranged diffused LEDs that were powered by a 9V Li-Ion battery.

Power System – The Raspberry Pi that was chosen for this design required voltage input of 5V and a current input of 2.5A. These were important when considering the design for the voltage regulator. The power from the regulator was transferred to the Raspberry Pi via a USB to Micro USB cable. This posed a challenge in finding a cable that was able to deliver the 2.5A that were necessary for the Raspberry Pi to be powered correctly.

For the top light, the regulator output voltage was sent to an on-off-on switch that allows the user to control whether three or six diffused LEDs were illuminated.

The battery life of the Raspberry Pi and top lighting system, when considering the highest current drainage, was calculated to be one hour and ten minutes. However, when tested, the battery life was closer to three hours before the batteries needed replacement.

Because the LED system consumes such little power, the battery life of the base light system was calculated to be about ten hours in length. With this long of a calculated battery life and with all the testing that has been done with the system, we have still not used the base battery enough for it to need replacement.

Overall, most aspects of this design were low power systems. Battery power was an efficient decision, especially after seeing the actual battery life compared to the calculated one.

PCB Design – The circuit board design of the top light was designed circular in order to accommodate the tube used for the objective lens. Diffused LEDs encircle the top in an even layout in order to maintain an even distribution of light.

The base lighting circuit board was also designed in a circular fashion in order to fit within the base of the system. This circuit board was designed to have a central diffused LED surrounded by six additional diffused LEDs in order to ensure enough light for viewing individual sample slides.

V. SOFTWARE DESIGN

Image Acquisition – All images are obtained by interfacing with the Raspberry Pi's camera module using a Python script. The Raspberry Pi camera module comes complete with a set of commands that are used for performing various tasks with the camera, such as taking a single image, recording a video, or taking a continuous stream of images.

Our implementation involves utilizing the PiCamera Python module which allows us to interface directly with the camera without having to open an external shell and hardcode a startup command. The PiCamera module allows us to instantiate a PiCamera object which allows us to easily fine-tune specific parameters such as the image resolution, image orientation, and framerate. For our project, we've chosen an image resolution of 600x400 pixels, and set the desired framerate to be a theoretical maximum of 80 frames per second. The command to begin the capture sequence is shown in Figure 4.

```
camera.capture_sequence(outputs(), ‘jpeg’, use_video_port=true)
```

Figure 4 - Capture Sequence Code Sample

The method outputs() parses the IO buffer and sends the images to the mobile application. The use_video_port parameter is used to simulate the speed of video captures, but in a burst format with a series of still images. The use of this parameter in conjunction with setting the framerate of the camera to its theoretical maximum allows us to obtain images from the camera module at peak efficiency.

Wireless Access Point Setup – To configure the Raspberry Pi to serve as a wireless access point (WAP), we used the hostapd and dnsmaq packages to get everything up and running. Hostapd is the package that lets us create a wireless hotspot using the Raspberry Pi, and dnsmasq is an easy-to-use DHCP and DNS server, which allows us to distribute IP addresses to clients that connect to the hotspot.

To get everything set up, minor modifications were made to the dhcpd and dnsmasq configuration files that allowed us to convert the wlan0 wireless interface on the Raspberry Pi into an access point. We then configured the DHCP server to properly allocate IP addresses to clients that wished to connect. As our application currently only supports one-to-one connections, the DHCP server was configured to only be able to assign one IP address at a given time, to prevent other users from attempting to connect while there is already an active connection present. The IP address lease time is set to last for approximately two minutes, which is the shortest lease time allowed according to the dnsmasq documentation.

The last thing to do was to configure the hostapd access point software. This involved modifying a configuration file with some general network configuration variables, including the interface name, SSID, Wi-Fi Protected Access (WPA) version, and WPA passphrase. Once all these steps were completed, the Raspberry Pi simply needed to be rebooted and it was then prepared for a device to connect to it.

Data Transfer – To transfer images from the Raspberry Pi to the mobile application, the same Python script that was used for image acquisition also initializes a TCP socket connection using the same IP address that the DHCP server leases and a port number that was arbitrarily chosen. The outputs() method that was briefly mentioned earlier in the image acquisition section handles the process of creating a byte array from each image that is captured and then sending that data over the socket to the mobile application.

For the mobile application to know how many bytes to read from each image, a custom protocol was created that would start by initially sending over the size of the image in bytes, followed by a special character that served as a delimiter between the size and the actual image data. Once that delimiter is read, the previous bytes that were read are converted to an integer representation which means that we now know the size of the image that will be coming over the socket and we can read the bytes into a byte array, convert the byte array to a drawable image that can be displayed on the application, and repeat the process for every image that is sent.

Image Processing – OpenCV was exclusively used for all image processing on the application. OpenCV has an Android SDK that we were able to add to the project structure and immediately integrate into our application. To manage the different image processing methods that we wanted to include, a PreferenceScreen activity was created to hold the different options that we wished to add. Since there are a seemingly infinite number of image processing techniques that could be added to this application, the software needed to be written in a way that makes adding or removing techniques a simple task.

To do this, an interface was created to define some generic methods such as apply(), isEnabled(), etc. The apply() method is going to differ for each technique as there are different methods to call and conditions to check depending on what you would like to accomplish. Each time a new image processing technique needs to be added, all the developer needs to do is create a new class that implements that interface and then include those inherited methods with any custom tweaks that are needed. References to these objects are available in a helper class that stores all the implemented processing techniques in an array with the interface name as its type, meaning that we are easily able to iterate over all the different techniques and apply them by using a simple looping mechanism, such as a for loop.

VI. Overall Design

Shown in Figures 5 and 6 are images of the microscope in its functioning state with a legend detailing the different components that are included.

The device works by flipping a switch located on the electrical panel. This switch powers Raspberry Pi connects to the image sensor and grabs images from it. The mobile device then connects to the WAP hosted on the Raspberry Pi. The images are then sent over a socket connection to the mobile device where the application displays them. This allows the user to see the project image in real-time and adjust the focus as needed. On the application, the user can capture an image, record a video, and perform basic image processing on the received media.

The mobile application also allows for the user to store captured media files to the directory of their choosing. In addition, the user is able to tag the images and group them in a logical way for easier accessibility.

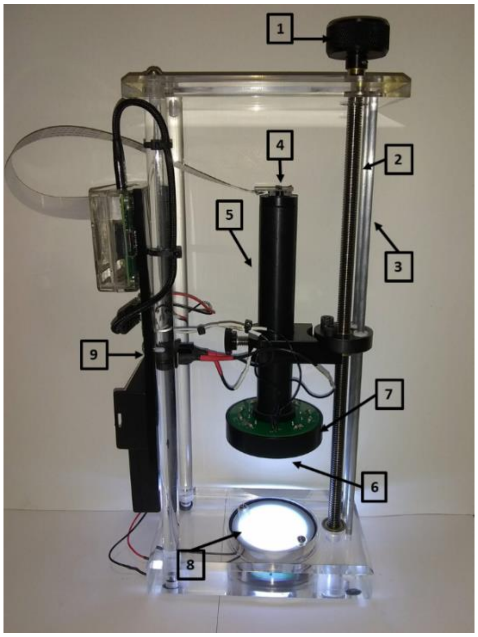

Figure 5 – Custom Knob (1), 12-inch ACME threaded rod (2), Two stainless rail guides (3), OV5647 Image Sensor (4), 150mm Delrin tube (5), DIN Standard Objective Lens (6), Top light with skirt (7), Base light with diffuser (8), Electrical panel (9)

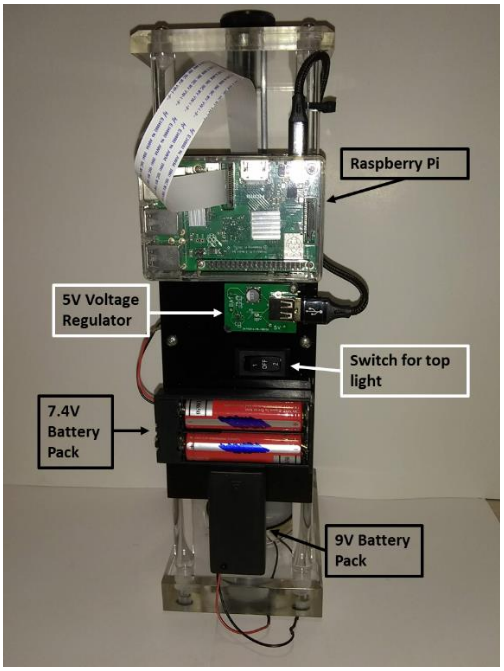

Figure 6 – Electrical Panel

.

References

[1] "Dnsmasq." *Dnsmasq - Network Services for Small Networks.*, www.thekelleys.org.uk/dnsmasq/doc.html.

[2] "Hostapd." *Hostapd: IEEE 802.11 AP, IEEE 802.1X/WPA/WPA2/EAP/RADIUS Authenticator*, w1.fi/hostapd/.

[3] "OpenCV 3.0.0." *Generated Documentation*, docs.opencv.org/java/3.0.0/.

Acknowedgements

We would like to specifically thank Richard Zotti, as without him, completion of this project would not have been possible.

Austin Bryant is a senior at the University of Central Florida and will receive his Bachelor's of Science in Computer Science. He will be starting a job with Amazon as a software development engineer in Austin, TX, working on the Alexa devices team.

Cayla Gill is a senior at the University of Central Florida and will receive her Bachelor's of Science in Electrical Engineering in May of 2019. She plans to move to the workforce after receiving her degree and will pursue a career in power systems.

Hannah Pierson is a LEAD scholar and a senior at the University of Central Florida. She will receive her Bachelor's of Science in Photonic Science and Engineering in May of 2019. She plans to move to the workforce after receiving her degree and will pursue a career in optics and photonics.