THE UNIVERSITY OF QUEENSLAND

MULTIFUNCTIONAL REMOTE MICROSCOPE

By

TIMOTHY SPITZER

School of Information Technology and Electrical Engineering (ITEE),

University of Queensland

Submitted for the degree of

BACHELOR OF ENGINEERING

In the division of Mechatronics

NOVEMBER 2012

Timothy Spitzer
Student No. 42046035
10 Warenda Street
Carina Heights, 4152

November 5, 2012

Professor Paul Strooper
Head of School
School of Information Technology and Electrical Engineering
University of Queensland
St Lucia, QLD 4072

Dear Professor Paul Strooper,

In accordance with the requirements of the degree of Bachelor of Engineering in the division of Mechatronic Engineering, I present the following thesis entitled

"Multifunctional Remote Microscope"

This work was performed under the supervision of Dr Mark Schulz. I declare that the work submitted in this thesis is my own, except as acknowledged in the text and footnotes, and has not been previously submitted for a degree at the University of Queensland or any other institution.

Yours sincerely,

TIMOTHY SPITZER

Acknowledgements

First of all, I would like to express my sincere thanks to my supervisor, Dr Mark Schulz, who has consistently provided me with support and guidance throughout my thesis. His valuable advice has saved me many hours in the design and construction of my project and he has kept my progress on track throughout.

Additionally I extend my gratitude to the EAIT mechanical workshop staff, particularly Mr Keith Lane, who helped me simplify my design and always had patience despite his busy schedule.

I would also like to thank CEIT for providing great facilities to work in and useful staff input on my project design.

Abstract

Rising student numbers and the development of new microscopy technologies has had an adverse affect on the quality of students' educations. To combat a reduction in practical learning opportunities, several Internet-based systems have been developed to give students additional resources outside the classroom.

The remote microscope system was developed on the proof-of-concept system created by Mr Yilun Fan in 2011. This project set out to prove that a similar system could simulate all the controls of a traditional microscope and also feature the advanced optical features required of university-level microscopes.

The design made use of a USB microscope, compact actuation control scheme and Arduino controller to minimise the project cost. It also utilised the MQTT messaging protocol to facilitate fast and reliable communication between student clients and the microscope.

The project successfully produced a prototype microscope system that was capable of remotely simulating all of the functionality of a traditional microscope as well as transitioning between multiple slides. However, due to the poor optical quality of the microscope, the investigation showed that a university level system could not be implemented without significantly greater expenditure.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Project Aim

The goal of this project was to develop a low-cost, robust, computer controlled, optical microscope capable of being operated remotely over the Internet. The project aimed to redesign and expand on the USB-microscope-based system created by Mr Yilun Fan [1]. Specifically, the project aimed to investigate whether a similar system could be adapted to university use by adding the functionality required of a microscope at that level, as well as additional functionality found in specialized microscopes.

1.2 Project Motivation

The motivation for this project came from three different sources. The first was the problem identified through discussion with Dr Mark Schulz and Dr Kirsten Zimbardi: namely that a shortage of microscopes and an increase in student numbers meant that university students are missing out on hands on time with microscopes. Background theory of histology is taking the place of the practical experience that is still required in the workplace [2].

The second motivation was the computer-controlled microscope created by Mr Yilun Fan [1]. This proof of concept system, shown in Fig. 1, successfully demonstrated that a remote microscope system could be developed for a high school environment by making use of low cost USB microscopes. However, due to its limited functionality, poor specimen illumination and a lack of robustness, it was not suitable for use at a tertiary level.

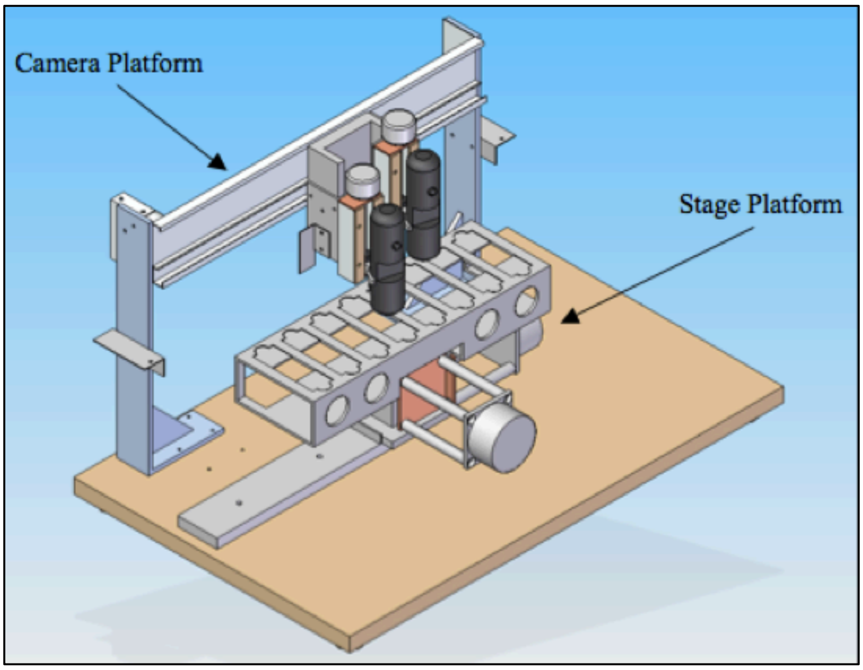


Figure 1: Mr Yilun Fan's computer controlled microscope. [1]

The final motivation came from the iLab initiative first developed at MIT. This program established a unified, scalable, web-based architecture for linking remotely controlled experiments to thousands of clients all over the world [3]. Clients were able to connect to a service broker that could schedule access to a variety of independently owned experiments, allowing them to learn and share results with others outside of the classroom.

1.3 Project Scope and Functional Aims

In order for the product to be a viable substitute for hands-on experience it had to replicate all of the core functionality of a university level microscope. These were the minimum requirements of the system:

- Slide manipulation (slide translation in the x-y frame)
- Adjustable focus (adjustable slide to camera distance)
- Compatible with both reflected and transmitted light
- Source and transmitted light filtering
- Adjustable light intensity
- Adjustable magnification

For it to be a useful remote teaching aid it had to have some additional functionality:

- Transition between multiple slides
- Remotely controllable

Because this project was focused on the design and implementation of the electromechanical system, the creation of a client interface and usage scheduler were deemed to be outside the scope of this investigation.

1.4 Report Overview

This report aims to describe and review the methods and reasoning behind the design of the system and its demonstrated outcomes. The next chapter provides some background information on optical microscopy as well as a summary of the prior art that assisted in the design of the system. This is followed by a breakdown of the design and implementation of the microscope, the methods used to test it and a discussion of the results and conclusions that can be drawn from this investigation.

Chapter 2

Background

2.1 Optical Microscopy

There are three main classes of microscope (optical, electron and scanning probe), all of which are used to view objects not visible to the naked eye. This project focused on the development of a bright field optical microscope. These are optical microscopes that use a condensed light source to illuminate the specimen from underneath.

2.1.1 The Basics

Fig. 2 shows a typical modern bright field microscope system.

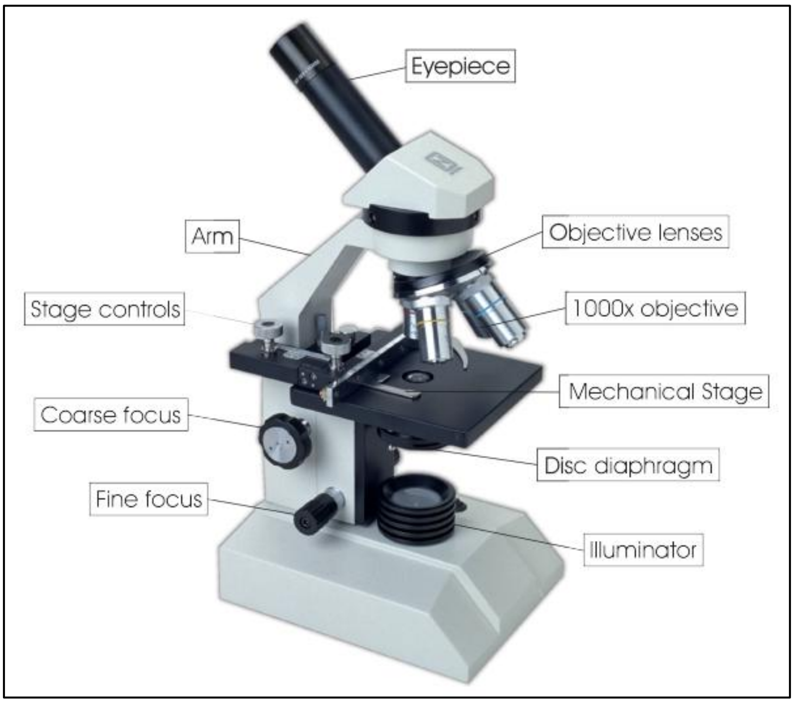


Figure 2: A typical bright field microscope. [4]

The microscope has several features to facilitate observation of the various parts of the specimen slide. Because the project aimed to mimic the operation of a laboratory microscope all of these features were included in the user's control interface:

- Stage controls to move the specimen relative to the objective lens
- Multiple magnifications
- A course and fine focus knob
- A disk diaphragm to adjust the level of light reaching the specimen.

The image is often improved by staining: a process that involves adding dyes which bind to certain parts of the specimen. This increases the amount of light absorbed in those parts and hence improves the contrast of the overall specimen [5]. Two slides with magenta staining are shown in Fig. 3.

Figure 3: Two slide specimens stained with magenta ink to enhance contrast.

2.1.2 Advanced Illumination Techniques

In addition to staining there are other ways of improving contrast and highlighting specific elements of a specimen. The multifunctional remote microscope was designed to facilitate three of these methods:

Dark Field Illumination

Dark field illumination is a technique whereby only inconsistencies, like cell walls, are illuminated as shown in Fig. 4.

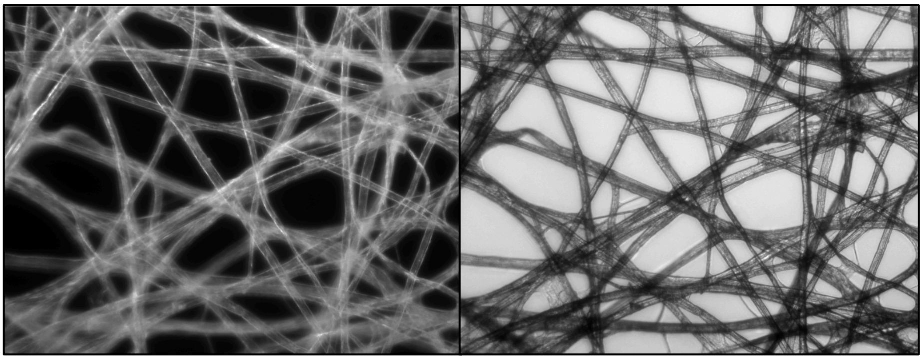


Figure 4: Hair viewed under dark field illumination (left) vs. bright field illumination (right). [6]

This is achieved by preventing light coming linearly from the source to the objective lens and only allowing the specimen to be illuminated by incidental light, as shown in Fig. 5. The result is that only the light scattered by boundaries in the specimen reach the observer.

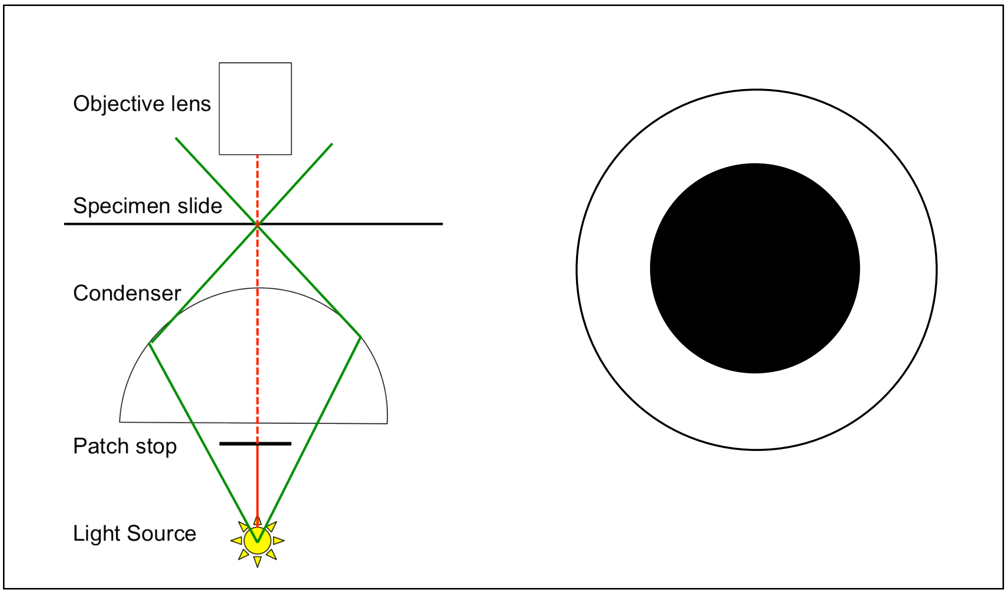


Figure 5: An illustration of how dark field illumination works (left). If the central light rays (red) are blocked by a patch stop (right) only the incidental light (green) scattered by the specimen will reach the observer.

This method of illumination is particularly useful for viewing live specimens as often the dyeing process is lethal and without contrast enhancement very little can be seen.

Florescence

Florescence microscopy is used to highlight specific parts of a specimen. It involves using fluorescent substances known as fluorochromes that mark certain structures in cells. Illuminating the specimen at the wavelength that the fluorochrome is sensitive to causes it to emit light at a different wavelength. By appropriately filtering the light before and after it reaches the specimen the cell structures are clearly highlighted as shown in Fig. 6.

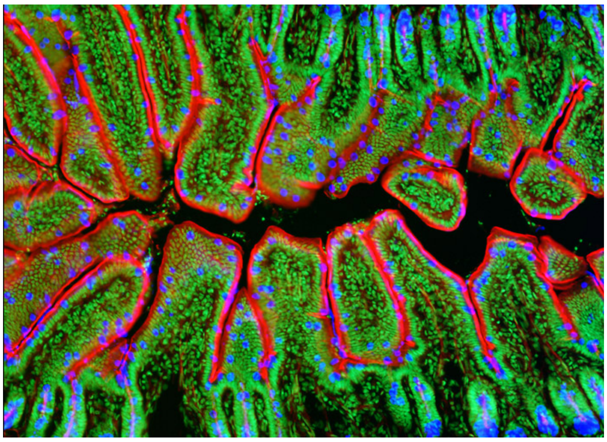


Figure 6: Mouse intestine tissue tagged with several different fluorochromes. Emission light has been recoloured to a different RGB value for each fluorophore. [7]

Polarisation

This form of microscopy makes use of the polarising effect of materials to study crystal structure and determine information such as refractive indices. The effect is achieved by changing the angle between two polarised lenses, one before and one after the specimen, until the desired effects are observed (Fig. 7).

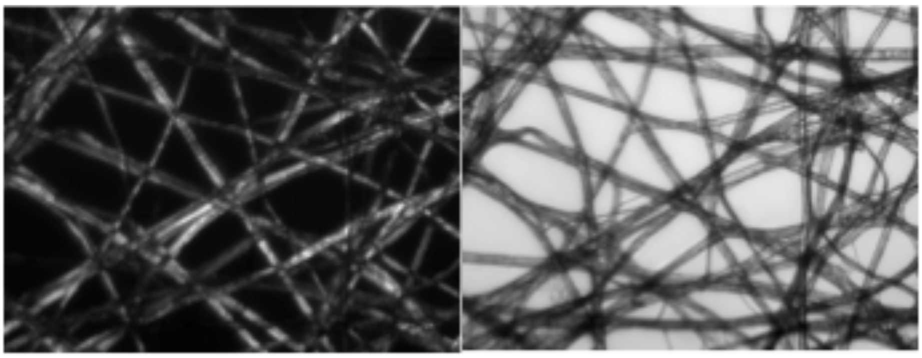


Figure 7: Hair viewed under a polarisation microscope (left) vs. a bright field microscope (right). [6]

2.1.3 The USB microscope

The USB microscope used in this project (Fig. 8) consists of a webcam, magnifying lenses and a ring of LEDs for reflective illumination. Magnification is adjusted by shifting the magnifying lenses up and down using the silver drum wheel. Raising and lowering the microscope adjusts the focus.

Figure 8: A $70 USB microscope capable of magnifying 20-200x [8]

The microscopes range in price from $50 to $500 depending on the quality of the camera and the level of magnification [9]. This project utilised a $70 model, as it was a low-cost proof-of-concept system.

2.2 Prior Art

Two existing designs for a computer-controlled microscope were reviewed as well as a virtual microscopy system. A breakdown and analysis of each system was performed in sections 2.2.1 - 2.2.3. A summary of the good and bad aspects of each system can be found in section 2.2.4.

2.2.1 Computer Controlled Microscope - Mr Yilun Fan [1]

Mr Yilun Fan's project focused on designing the proof-of-concept, remotely controlled microscope system shown in Fig 1 (Section 1.2).

The system consisted of a stage, holding multiple slides in a row, which was capable of actuating horizontally (in both the X and Y directions) to manipulate the slide position. Mounted above the stage, on a rail, were two USB microscopes that were pre-focused for magnifications of 20x and 200x. The microscopes could move along the rail to change slides as well as magnifications. Both the camera rail and the slide platform actuators were recycled systems that were used in order to lower the cost of the system.

The electrical system was largely structured around the use of the Arduino platform to reduce time in the prototyping stage of the project. Communication between the system and a client was facilitated through the use of an Ethernet shield and the MQTT protocol: a protocol that allowed for text based communication between the client and the system. The motors were driven using a motor shield capable of driving the two stepper motors and the rail servo.

Review

The project successfully proved that a low-cost computer controlled microscope could be implemented. The system demonstrated that the MQTT protocol could be used effectively to control the microscope over the Internet. It also demonstrated an innovative use of micrometers and stepper motors to perform low-cost, high-precision slide manipulation.

The downfalls of this design were a lack of robustness and functionality. The camera rail for the system (a repurposed printer head carriage) lacked strength, which both greatly reduced the lifetime of the product and caused problems during development. The system was also restricted to only two magnifications and the operator was unable to adjust the focus. These features could not be implemented because the camera rail could not take additional weight.

2.2.2 Control of a Remote Microscope Over the Internet: Merck Research Labs [10]

This project was designed to fulfil an increasing need for researchers to collaborate over the Internet by allowing them to view and control a microscope remotely. The project was divided into three levels of control to meet different staff needs.

Phase I was to augment a Leitz Laborlux 12 Microscope with a camera so that remote clients could watch through the microscope. Because of bandwidth restrictions, the design used a timed capture system that transmitted photos every few seconds to a webpage that constantly refreshed. This refresh rate was maximized at once every four seconds for local clients and once every seven seconds for overseas clients.

Phase II implemented remote control of the microscope over the Internet. The user had control over the slide platform position and magnification. Because of the significant image delay experienced by the user the system was designed to either move to specific coordinates or to increase/decrease the current position by 100um (at 400x magnification). The user also had access to an auto focus function.

Phase III added a slide loading mechanism capable of switching between 15 different slides (Fig. 9). The system consisted of a slide rack that was driven by a DC motor and pinion gear. Optical sensors worked as an interlock to ensure slides were centred correctly.

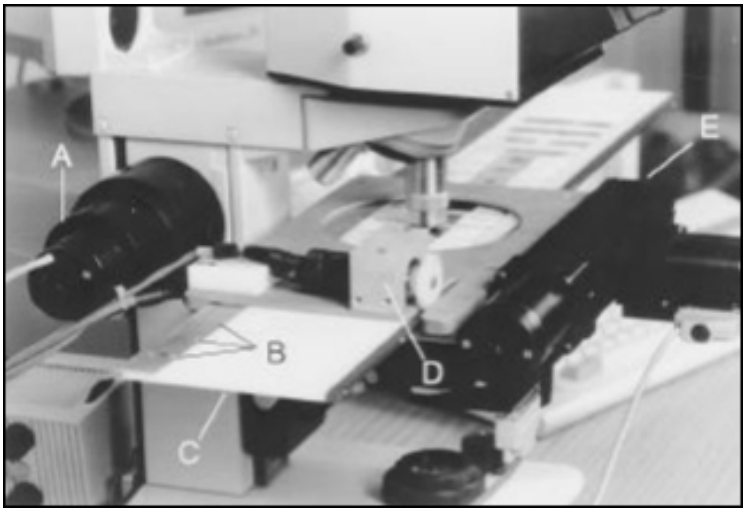


Figure 9: ***Custom built slide loader. (A) Ludl autofocus controller motor, (B) optical sensors, (C) 15-slide microscope tray, (D) spring-loaded, geared motor unit and (E) Ludl x-axis/y-axis motorized stage." [10]***

Review

Merck Research Laboratories successfully instrumented an existing microscope to be controlled over the Internet. They managed to create a low-cost slide changing mechanism that could be augmented onto any system without other changes having to be made. Despite bandwidth limitations they also managed to create a system that would function satisfactorily in collaborative research when a researcher was working at the station.

The main drawback of this system was slow rate of transfer of images to remote observers. Because of the large delay this system would function very poorly when remote operators were attempting to locate objects on slides. In addition, the high cost (~$34000), while acceptable in this instance, does not satisfy the project goal of building a low-cost system.

2.2.3 Virtual Microscopy: M. Triola and W. Holloway [2]

This project took an alternate approach to teaching microscopy to students. Rather than giving students the opportunity to use a microscope, over 1000 slides were scanned and stored in a database as Google Maps formatted images. A basic GUI (figure 10) was created for students to access and explore the slides.

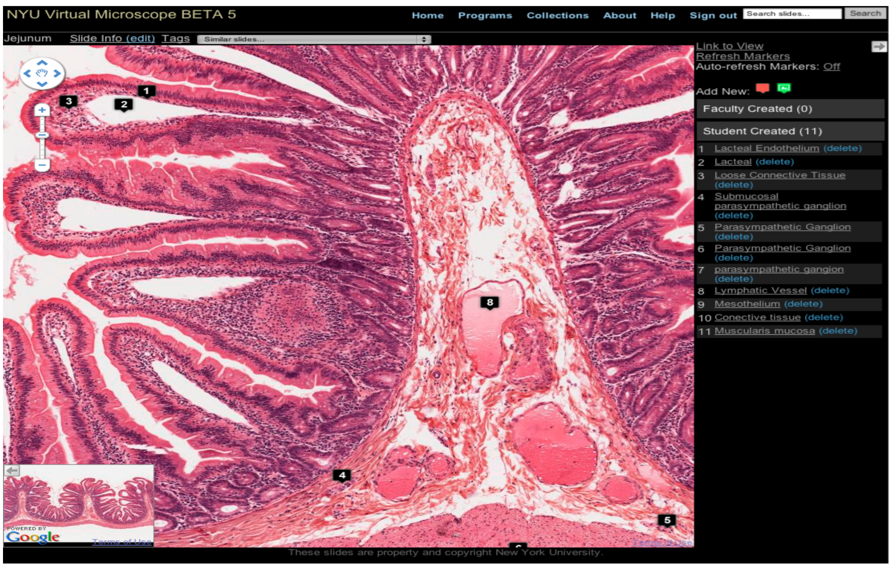


Figure 10: A virtual microscope GUI. [2]

Additional functionality included the ability for students and staff to annotate the slide with a number, name and short description. All students could see these tags and up or down vote them based on whether or not they were correctly labelled (and with sufficient down- votes tags would be removed). An exam mode was included as well, whereby students were given a set of slides that they were required to label appropriately in the time given.

Review

Virtual microscopy is a very powerful tool in teaching students identification techniques and better than physical microscopy in many ways. It gives students a chance to work at their own pace rather than being pressed to share a limited number of microscopes within designated class hours. It also gives students a chance to collaborate more effectively as they can all view identical slides and quickly move between tagged elements on the slides.

The system did have limitations. Students were limited to viewing slides in one focal plane: an obstacle that cannot be overcome without vastly increasing the database's memory space. Students were also only able to view static images and not dynamic processes that can be viewed with a physical microscope. In addition students miss out learning how to use a physical microscope and these skills are still considered important in the workplace [2]. Furthermore, slide scanners are still prohibitively expensive for many institutions, meaning that students are not able to view their own slides.

Chapter 3

Design & Implementation

The following chapter is broken into three main sections: the mechanical, electrical and software design and implementation. These sections outline the reasoning behind the various design decisions that were made in order to implement the final multifunctional remote microscope shown in Fig. 11.

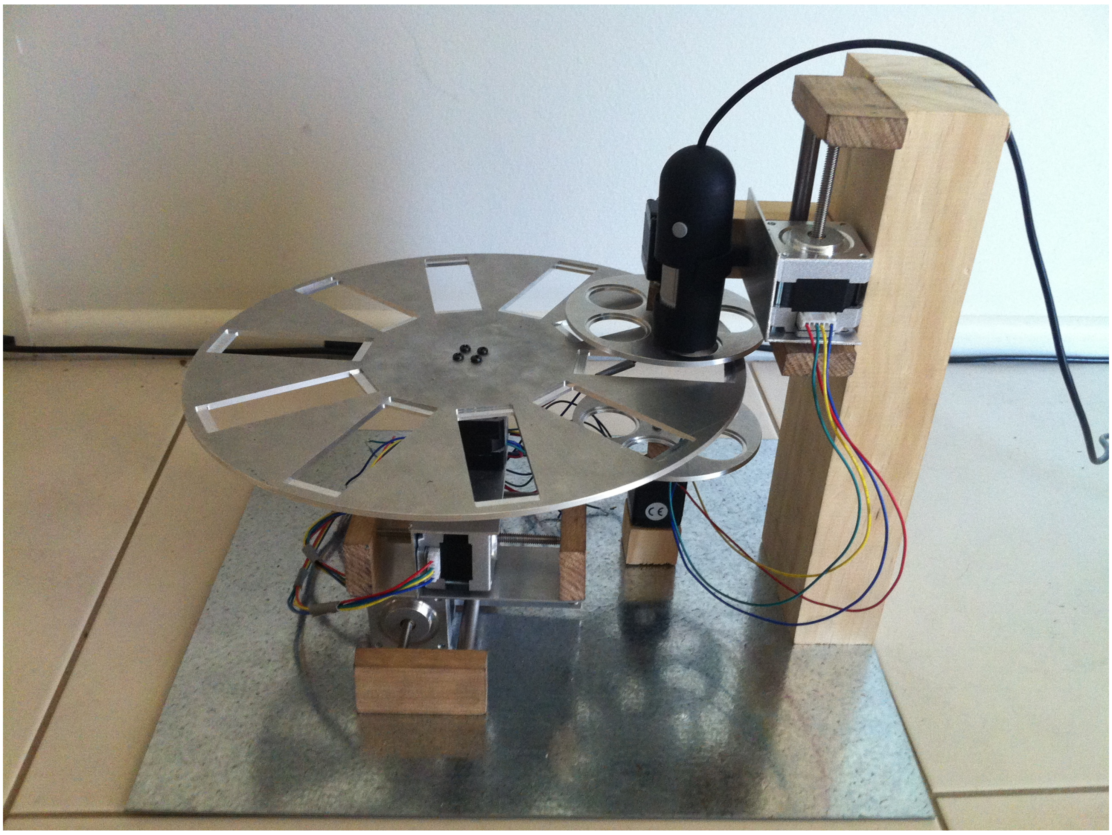


Figure 11: The multifunctional remote microscope

3.1 Mechanical

3.1.1 Overall Design

In order to meet the functional aims of the project (listed in Section 1.3) the microscope had to be able to do the following:

- Move a slide relative to the microscope camera in the X, Y and Z direction (for manipulation and focus adjustments)
- Switch between multiple slides
- Rotate the magnification wheel of the microscope
- Switch between filter lenses both above and below the specimen

The main problem with adapting this functionality to Yilun's computer controlled microscope [1] was that the additional subsystems – lens holders, light source, magnification adjuster and focus adjuster – would have had to move on the same rail as the microscope, or on a second rail underneath. These rails would have had to been redesigned completely to support the extra weight and would have added an additional motor to the final design. Alternatively if the linear slide holder had been moved instead of the optical equipment then the high precision slide manipulation, which was a hallmark of the design, would have had to be replaced with a less precise system in order to make transitions between slides acceptably fast.

By switching to the rotating slide holder design, shown in figure 11, instead of a linear one the optical path could remain fixed. This simplified the problem by making the separate subsystems independent of each other. It also meant that the high precision X-Y actuation system from Yilun's computer controlled microscope [1] could be kept.

3.1.2 X-Y-Z Linear Stages

Unfortunately a low-cost X-Y platform, similar to the one borrowed by Mr Yilun Fan from the laser laboratory at UQ, could not be found. The solution was to develop a comparable system using the low-cost, hybrid stepper motors shown in figure 12.

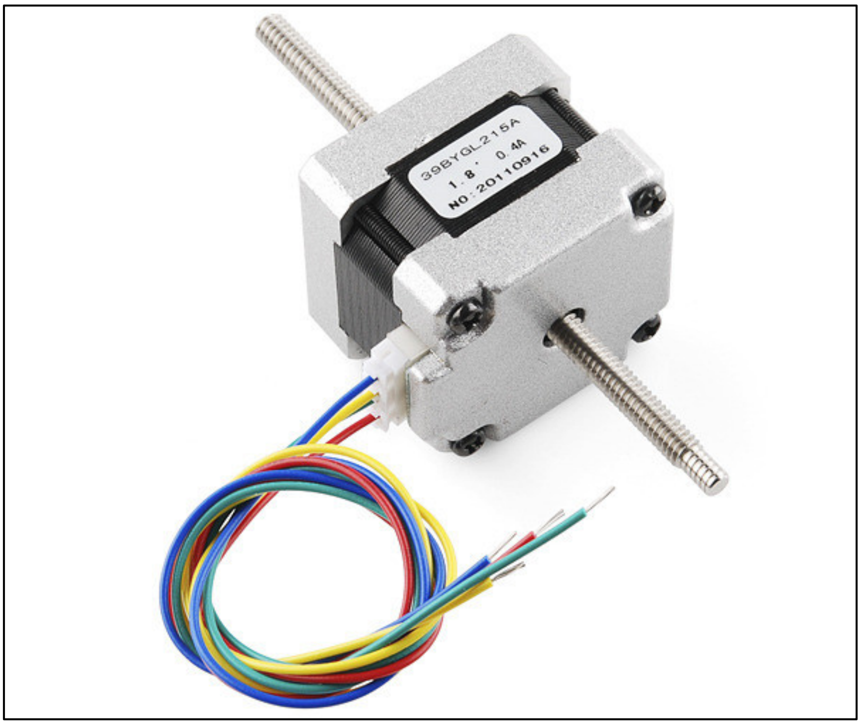


Figure 12: A hybrid stepper motor that uses a leadscrew based design for linear actuation. [11]

These motors utilise a fixed, 10cm threaded shaft and a rotor with a threaded bore to move linearly at 0.01mm/step, providing a very high level of precision at the cost of a slow actuation speed. They also had an advantage over other linear actuators, like belt-drive systems, in that they held position even when powered down. Their low cost ($35 [11]) compared to linear actuators with similar precision (like the Concentric International linear actuator series starting at $87 [12]) and convenient shaft length (they were able to traverse the full length of a slide 75mm long slide) made them a good choice for this proof-of-concept system.

The housing for these motors, shown in figure 13, was designed with modularity and simplicity in mind. The X, Y and Z-axis housings all were identical, simplifying the manufacturing process and allowing them to be easily replaced if broken. They consisted of a silvered steel rod running parallel to the motor shaft that supported the majority of the static load and prevented the motor case from rotating around the shaft. Both the rod and the shaft were fixed into lightweight, plastic blocks at either end. The aluminium brackets attached to the motors served three purposes:

- They connected the motor to the silvered steel rod
- They provided a mounting point for the subsystem they were attached to
- They overlapped the end blocks when the motors were fully extended allowing for a photointerrupter to be used as a limit switch

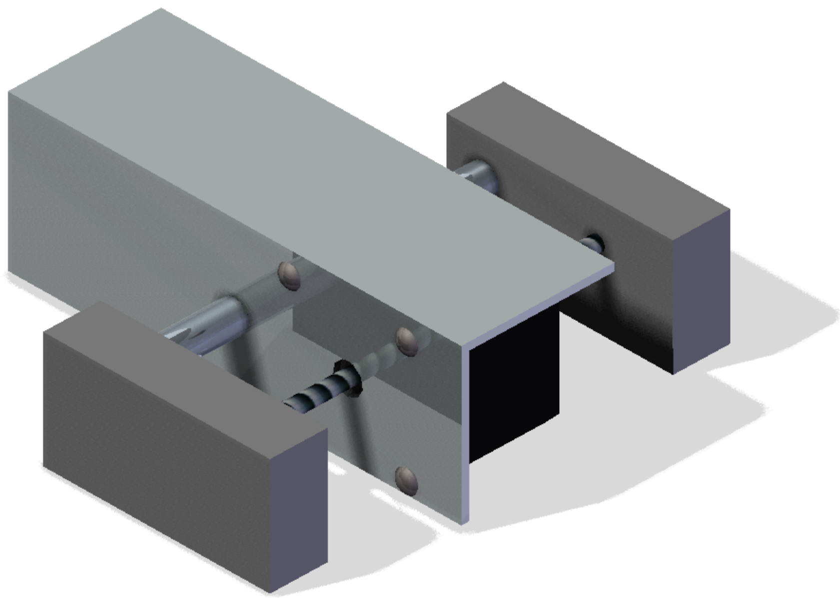


Figure 13: The X/Y/Z stage linear actuator for the multifunctional remote microscope

Bearings on the steel rod were deemed an unnecessary expenditure as the motors had more than enough torque to overcome the additional friction and the small amount of clearance between the shaft diameter and the bracket hole prevented the stage from rattling around.

3.1.3 Slide Holder

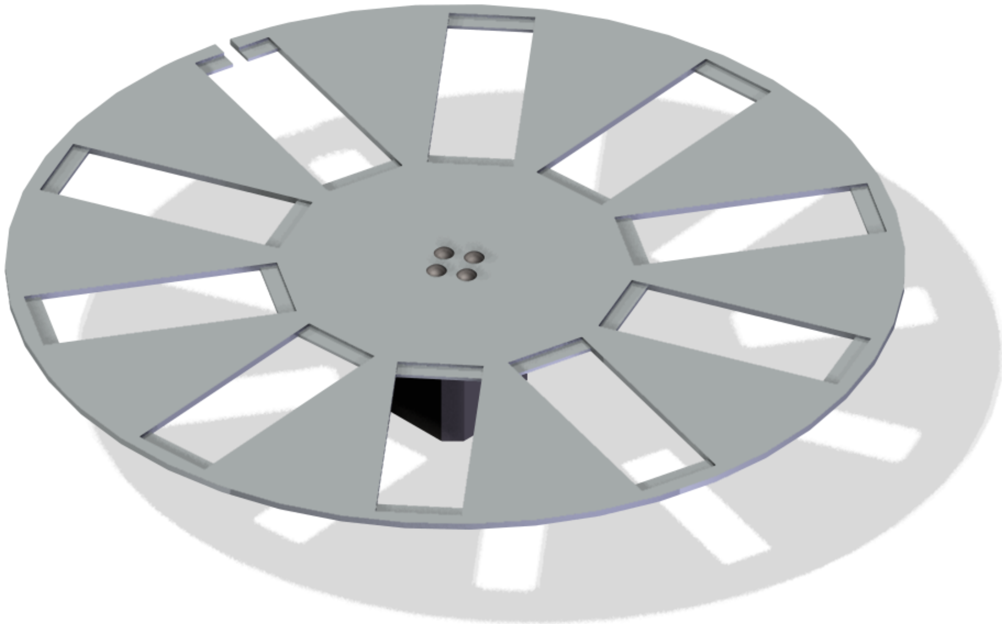


Figure 14: The slide holder for the multifunctional remote microscope

The slide holder (figure 14) was a 26cm diameter disk of 2mm thick aluminium with ten separate, metric (75mm x 25mm) slide cavities milled 1mm into the surface. Inside these cavities were smaller (65mm x 25mm) rectangular cutouts to allow light through. These cutouts were larger in size than on Mr Yilun Fan's slide loader (30mm x 25mm) [1], as they had to work for the multi-specimen slides more commonly used at a university level. The slide holder was mounted onto a motor shaft using a pre-made motor coupler [13], which in turn was mounted onto the Y stage aluminium bracket. It also had a small cutout on the perimeter so that a photointerrupter could zero the system and the correct slides could be navigated to.

The diameter and capacity of the slide disk were reasonably arbitrary. The minimum diameter that was required to prevent interference between the x and y stages and the bottom lens holder was 25cm so a slightly greater diameter was chosen to allow for some clearance. It is possible to increase the slide capacity by increasing the disk diameter, however, it would result in a larger mass moment of inertia and less rigidity meaning that both a motor with more torque and a thicker gauge of aluminium would be required.

3.1.4 Lens Holders

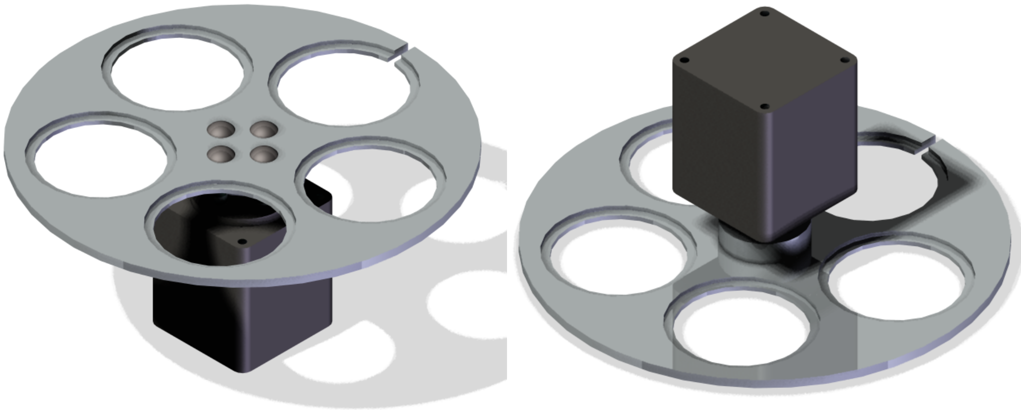


Figure 15: The bottom (left) and top (right) filter lens holders for the multifunctional remote microscope.

The two lens holders (figure 15) were constructed in the same way as the slide holder. They consisted of five 32mm diameter cavities with 28mm diameter holes for holding large filter lenses. They were also capable of holding smaller lenses by using an adapter ring illustrated in figure 16.

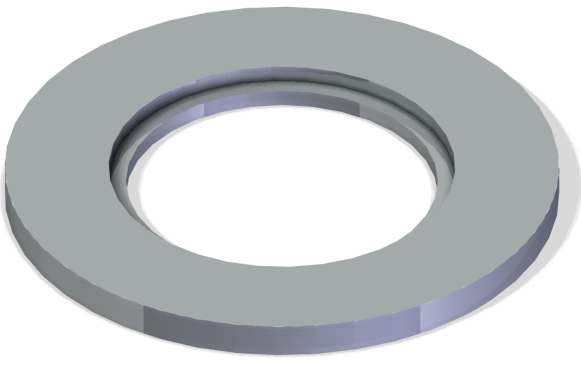


Figure 16: An adapter ring for holding 20mm filter lenses.

3.1.5 Microscope Mount

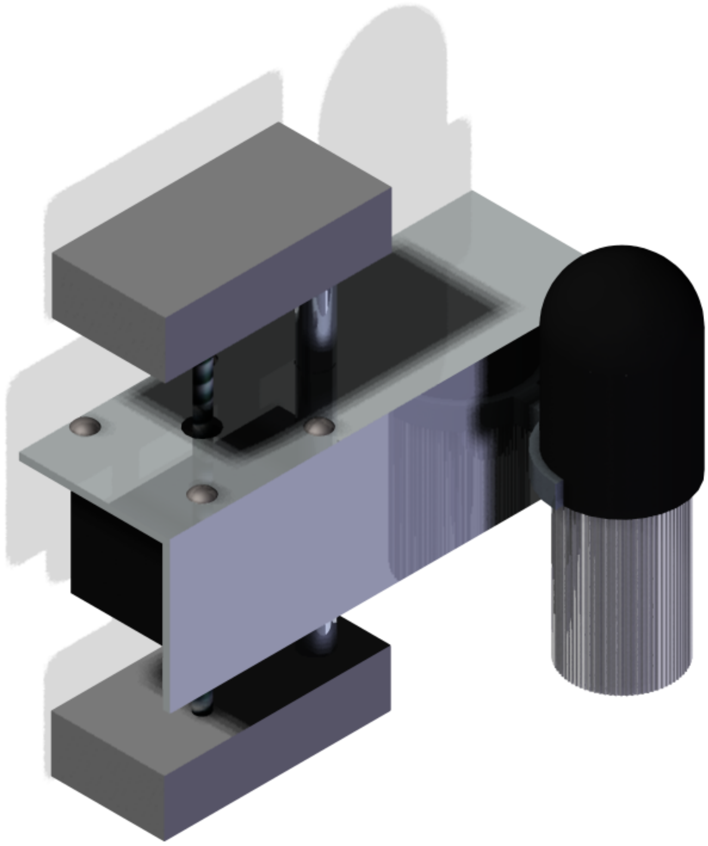


Figure 17: A USB microscope mounted to the Z stage (focus stage) of the multifunctional remote microscope.

The microscope mount design (figure 17) made use of a clip provided in the package to mount the scope to the aluminium bracket of the Z stage. Two different methods, where investigated for the control of the magnification wheel.

The first method was to use a belt between a motor and the wheel. The advantage of this method was that the motor could be placed closer to the Z stage, reducing the moment acting on the Z stage motor. The downside was that a significant part of the microscope housing would have to be removed, potentially opening it up to debris.

The second method was to use a cylindrical silicon drum mounted next to the magnification wheel and driven by a motor. The advantage of this method was that it would be compatible with any of the USB microscopes without having to modify them. The microscope could have been easily upgraded and the lens would not be exposed to dust and debris. The downside was that the drum had to be mounted further from the Z stage, creating a larger strain on the Z stage motor.

Unfortunately, due to unforseen problems with the implementation of the electrical design and insufficient reserved time neither of these magnification methods could be implemented in the final project construction.

3.1.6 Frame

The frame was constructed from scrap materials. It utilised a 1.5mm thick steel sheet as a sturdy base and a 30x60x300mm piece of pinewood for mounting the top lens holder and microscope mount.

3.1.7 Motor Selection

Seven actuators were required to automate all of the elements of the microscope: the top lens disk, bottom lens disk, slide disk, x-axis, y-axis, z-axis and magnification adjuster. All of these motors had to be able to accurately turn specific angles on command. The reason for choosing stepper motors for the x/y/z stages is described in section 3.1.2. Stepper motors were chosen for the other four actuators because they greatly simplified the electrical and software design. Additionally, they were considerably cheaper than encoder coupled dc motors ($20 vs. $40 for two products with similar specifications [14][15]) and because they didn't require PWM signals (like servos) a cheaper Arduino with fewer PWM channels could be used.

Because both the lens disks and slide disk were lightweight the torque required by the motors was comparably low compared to the torque supplied by available hobby stepper motors. The lens disk motors [16] were chosen for their small form factor (preventing them from interrupting the optical path) and their low cost relative to smaller motors [14]. The slide disk motor [17] was chosen for its greater torque – allowing it to move faster without missing steps – and its compatible operation voltage.

3.2 Electrical

3.2.1 Electrical overview

The electrical design centred on the use of an Arduino, which had proven to be an effective controller for Yilun's microscope [1]. The primary concern was the powering and control of seven stepper motors, which each required four control inputs. Off-the-shelf motor control solutions such as the EasyDriver stepper motor driver [18] were considered but their high cost and large number of required connections made them a poor solution. Instead, a custom controller scheme was designed to minimise the number of required connections, lower the cost and reduce the power usage of the system. This scheme is illustrated in the flowchart below (figure 18).

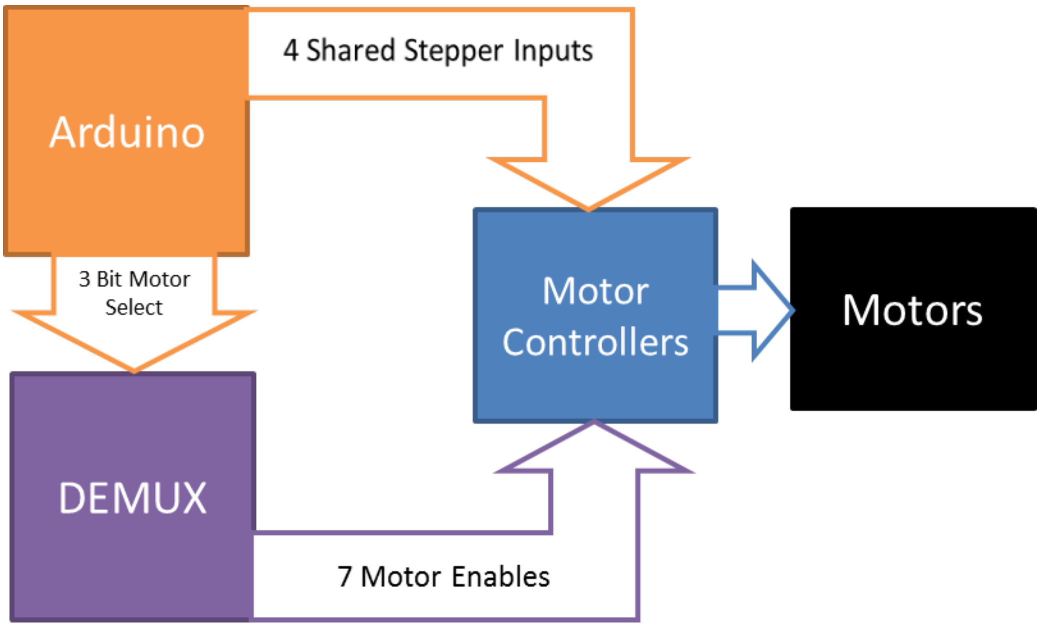


Figure 18: A flow chart illustrating the motor control scheme used in this project.

By using a demultiplexer to toggle the enable pins of the motor controllers they could all be connected to the same four stepper control inputs. This meant that only one motor could run at a time, limiting the current required from the power supply circuit to one Ampere. The only limitation of this design was that the microscope could not move diagonally without a more complex, rapid toggle between x-y motors in the software implementation.

3.2.2 Arduino Board and Ethernet Shield

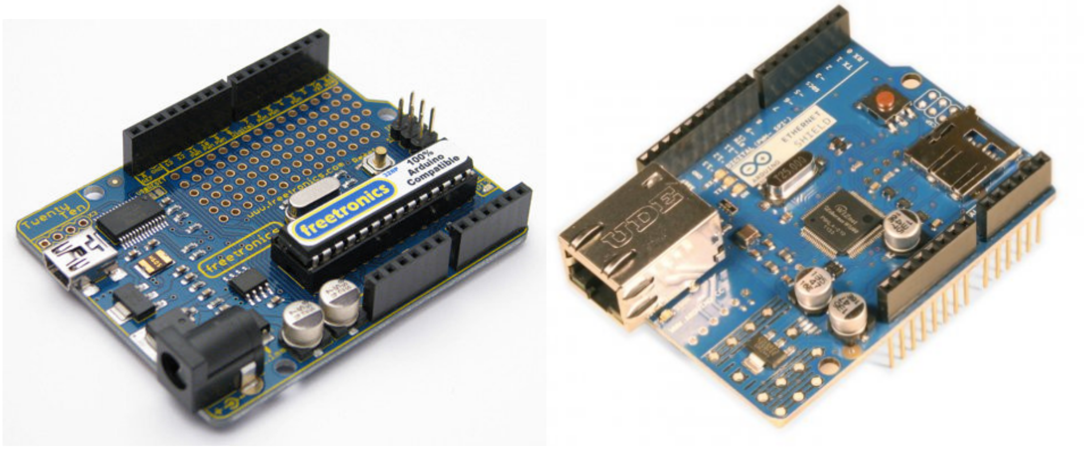


Figure 19: An Arduino-compatible TwentyTen board [20] (left) and an Ethernet shield [21] (right).

The Arduino board selected was the Freetronics TwentyTen shown in figure 19. It was chosen primarily because Yilun had effectively demonstrated it in his microscope system [1] and one was already available from Dr Mark Schulz. As a low end Arduino board it still had more than sufficient functionality including:

- Twice the required flash memory used in the final program (32KB)
- 1KB of EEPROM for storing motor positions for start-up.
- 14 digital I/O pins - two more than the required 12.
- 5V logical supply voltage for powering the demultiplexer and motor controller chips.

Coupled with the TwentyTen was an Arduino Ethernet shield [20] (shown in figure 19). The Ethernet shield was required for communication with the MQTT service broker that facilitated remote control of the microscope.

Although these two boards functioned satisfactorily, they could have been replaced with an Arduino Uno board [21] that had the same features as the TwentyTen along with on-board Ethernet and a reduced total price.

3.2.3 Demultiplexer

The demultiplexer chosen was the Texas Instruments CD4051B CMOS analogue multiplexer/demultiplexer [22]. It featured three channel selection pins, eight output channels, an output source pin and an enable pin. The wiring schematic used is shown in figure 20.

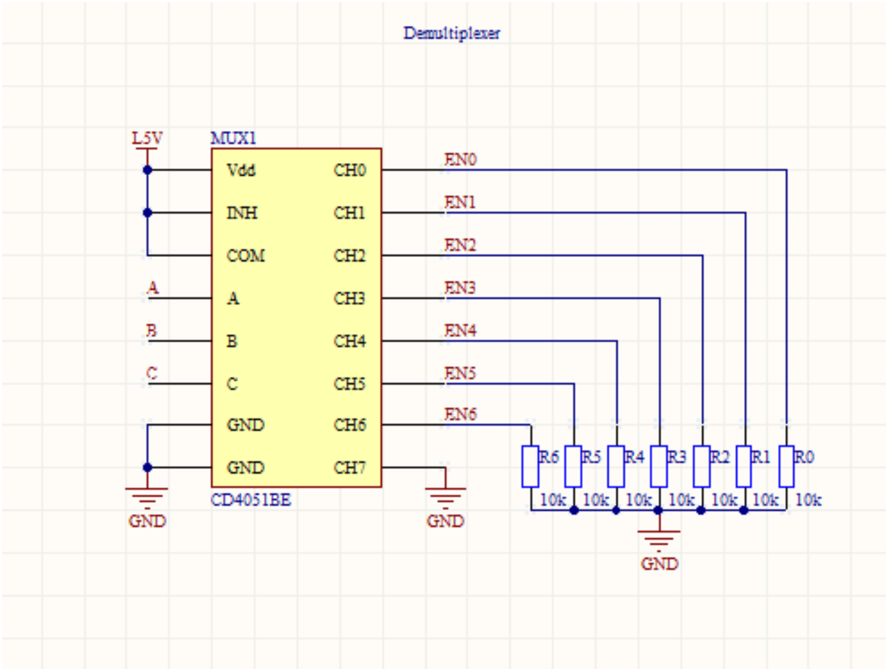


Figure 20: A schematic for the CD4051B demultiplexer

The resistors shown in figure 20 were connected in parallel with the motor controllers and were used to pull down inactive channels to the ground voltage. The source voltage (COM) as well as the enable voltage (INH) where kept at a constant 5V. These did not have to be connected to digital outputs on the Arduino, as there was an eighth output channel to switch to when disabling all motors.

This demultiplexer was chosen specifically because it was cheap, had the required eight outputs and it was contained in a DIP package (which made soldering easy). The Texas Instruments SN74AHCT138D logical demultiplexer [23] would have been a better chip, as it didn't require pull-down resistors to function; however, it only came in a surface mount package so it was not used.

3.2.4 Motor Drivers

The Texas Instruments SN754410 Quadruple Half-H Driver [24] was the motor controller used in the multifunctional remote microscope. It featured a maximum output current of 1A, a 5V logical supply voltage and a motor supply voltage ranging from 4.5-36V, fitting within the specifications for all seven motors. It was the cheapest DIP package motor driver to fit within the system specifications and was substantially cheaper than the EasyDriver board ($4 [24] compared to $12 [18]). The schematic for one of the seven controllers is illustrated in figure 21.

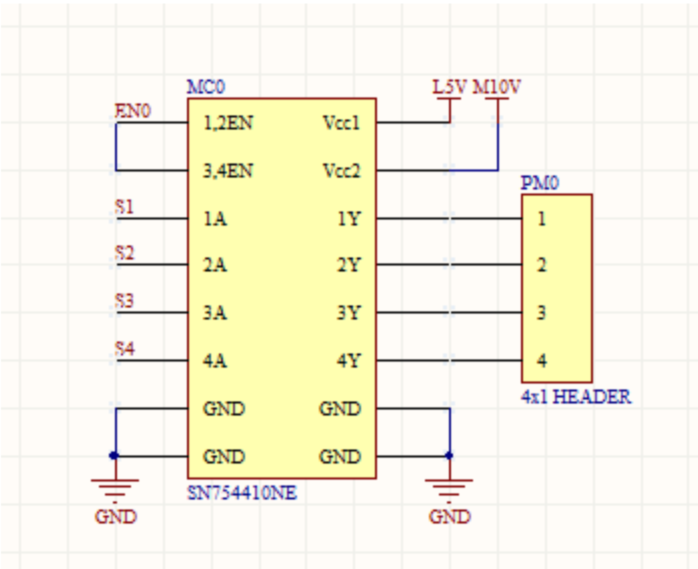


Figure 21: A schematic diagram showing how the SN754410 motor driver was connected in the project.

The inputs (1A-4A) were connected to four digital IO pins on the Arduino and shared with the other six motor controllers. The logical voltage Vcc2 was connected to either a 10V regulator (for the X/Y/Z and slide motor controllers) or a 5V regulator (for the lens and magnification motors). The enable pins (1,2EN and 3,4EN) were connected to the seven different output channels of the demultiplexer described in section 3.2.3.

3.2.5 Power Supply

The Arduino and Ethernet shield were powered using a standard 12V AC adapter. The voltage was regulated down to 10V and 5V for the motor controllers using a standard voltage regulator circuit shown in figure 22.

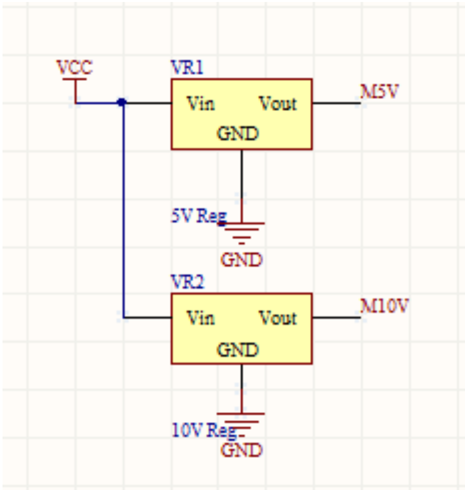


Figure 22: The shared voltage regulating circuit for the motor drivers used in the microscope system.

The voltage regulators chosen (LM2940T [25] and L4940V5 [26]) were selected for their suitable current rating, input voltage and low cost. Smoothing capacitors coupled either side of the regulators were recommended but deemed unnecessary as the signal noise was not significant enough to affect the motors.

3.2.6 Light Source

The light source was required to cover a broad spectrum, including visible light as well as ultraviolet and infrared for fluorescence microscopy. Three different LED packages were used to fully cover the spectrum required. They were the Cree C503C [27], VCC VAOL-3GUV8Y4 [28] and the OSRAM SFH4350 [29]. These were chosen to be cheap and have similar intensities. Figure 23 illustrates how they were connected.

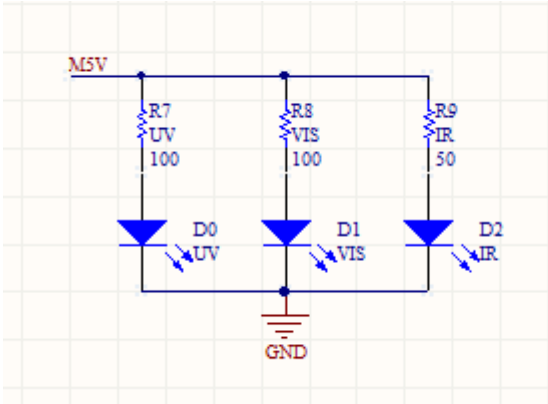


Figure 23: The LED circuit used for the light source.

The resistors were required to prevent the LEDs from drawing too much current and burning out. Filter lenses were used to reduce intensity, rather than a PWM output, in order to better replicate the disk diaphragm of a traditional microscope. This included turning the LED off by covering it with an opaque lens (removing the need to connect to the Arduino at all).

3.3 Software

3.3.1 Software Overview

The software component of the project was simple thanks to the uniformity of the hardware design and the availability of premade libraries on the Arduino platform. The project made use of the MQTT library created by Mr Nick O'Leary [30] and the generic stepper motor library [31] to speed up development. The basic control flow is illustrated in figure 24.

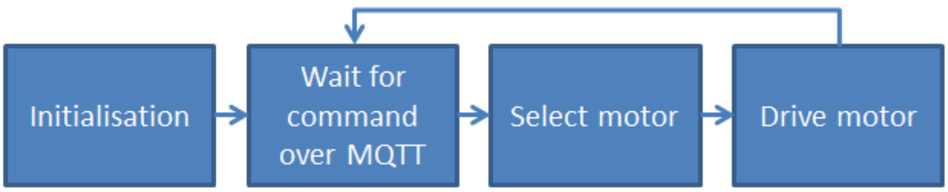


Figure 24: A flow diagram illustrating the basic structure of the Arduino code.3.3.2 Initialisation

The full software implementation can be found in Appendix A or on the supporting CD.

3.3.3 MQTT Basics

The MQTT protocol is a lightweight machine-to-machine messaging protocol [32]. It uses a system of publishing and subscribing to transmit messages between an arbitrary number of clients. Messages can be published to a message broker under a topic name. Any clients subscribed to the same topic receive these messages. Figure 25 illustrates the infrastructure behind a basic MQTT network.

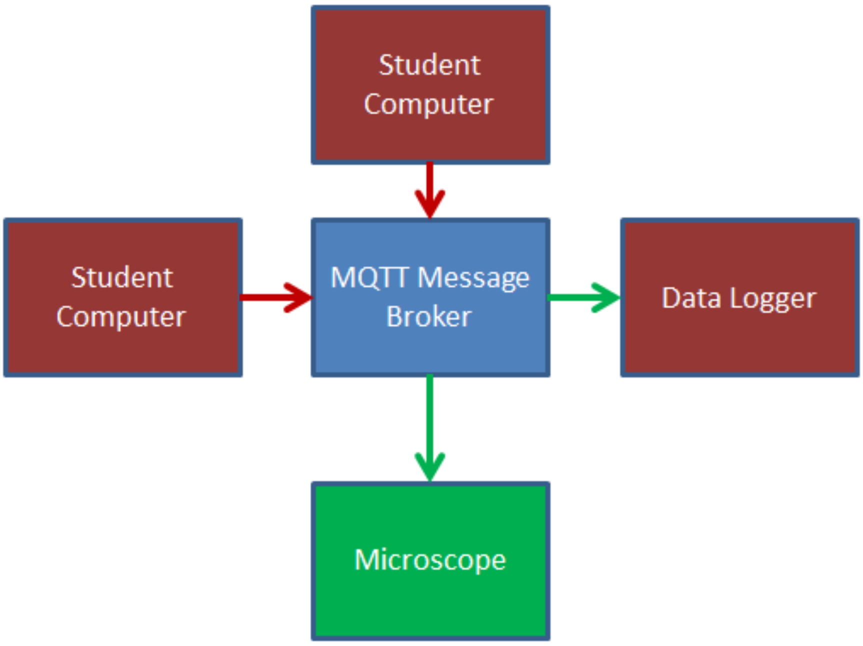


Figure 25: A graphical representation of the infrastructure behind a basic MQTT network. Red arrows indicate the flow of published messages and green arrows indicate those messages being red by subscribers.

Students publish messages to the broker under the */Microscope* topic. The microscope, which is subscribed to the same topic, reads the messages and executes the appropriate command. Additional devices, such as the "data logger" block in figure 25, can subscribe to the same topic and do whatever they want to the information without interrupting communication with the microscope.

In the project implementation, subscription is handled during setup with the code block shown in figure 26.

```
#include <PubSubClient.h>
#include <Ethernet.h>

byte mac[] = {0x90, 0xA2, 0xDA, 0x00, 0x01, 0x1B};
byte ip[] = {192, 168, 0, 77};
byte server[] = {192, 168, 0, 3};

void setup() {
	...
	// initialise Ethernet communication
	Ethernet.begin(mac, ip);
	if (client.connect("microscope")) {
		// subscribe to the desired channel if connected
		client.subscribe("/Microscope");
	}
}
```

Figure 26: Setup code for the MQTT protocol used in the project.

The microscope client checks for incoming messages every time the client.loop() function is called. If this is not called regularly then the system can disconnect and hence a check was added to reconnect the system if required. Messages received on the subscribed topic are handled by the callback function. This code is shown in figure 27.

```
PubSubClient client(server, 1885, callback);

void callback(char* topic, byte* payload, unsigned int length) {
	// messages are handled in here
}

void loop() {
	if(!client.loop()) {
		client.connect("microscope");
		client.subscribe("/Microscope");
	}
	...
}
```

Figure 27: Code for checking for incoming messages and handling them.

3.3.4 MQTT Commands

In order to select and drive motors using plain text messages, a message format was devised. Messages were formatted in two different ways:

- mds
  - m = letter unique to each motor
  - d = direction, either “f” for forward or “b” for backward
  - s = the speed in RPM for the motor to rotate
- mn
  - m = letter unique to each motor
  - n = number to rotate to (for slides/filter lenses)
- k = kill all motors

Table 1 lists the implemented message formats and their associated commands.

Table 1: Implemented message formats and their associated commands.

Message	Command
s#	Move to slide #
t#	Move to top filter #
b#	Move to bottom filter #
xf##	Move x stage forward at ## rpm
xb##	Move x stage backward at ## rpm
yf##	Mode y stage forward at ## rpm
yb##	Move y stage backward at ## rpm
zf##	Move z stage forward at ## rpm
zb##	Move z stage backward at ## rpm
k	Kill all motors

These commands were implemented because they facilitate the basic operation of a pushbutton-based GUI. When a user wanted to switch between slides or filter lenses they would type in the desired number and press the appropriate button, which triggered the publishing of the “mn” format message. To navigate around the slides, the client held down an arrow in the direction that they wanted the camera to move until they reached their desired location. The initial button press triggered the transmission of the “mds” format message and the release publishes the “k” message. By allowing the speed of the motors to be adjusted by the client a coarse and fine navigation speed could be implemented, further replicating the controls of a traditional microscope.

3.3.4 Motor Selection and Control

The motor control code utilises the stepper library. A stepper instance is created by calling the "Stepper" function with the number of steps of the motor and the four control pins as arguments. After this setup the "step" method was called to move the motor forward an appropriate distance, blocking the program until it finished.

Motors were selected using the "selectMotor" function (see Appendix A). This function used a basic switch/case format to set and clear the demultiplexer control pins for the appropriate motor defined in the header.

Slide and filter lens motors each had a drive function so that slide/lens numbers could be used instead of steps. The number of steps in-between each slide was calculated as the number of steps per revolution divided by the number of slides or lenses. This was multiplied by the difference between the desired lens number and the current lens number in order to move the correct distance.

In order to implement the potentially endless forward-until-kill scheme, outlined in the previous section, without disrupting the MQTT communications the motors were moved in intervals of 20 steps (in-between checking for new communications). This reduced the resolution of the X, Y and Z motors by ten times. Although it wasn't ideal, it proved to have a negligible effect due to the wide viewing angle of the USB microscope used. If a more expensive USB microscope with a smaller viewing angle was used it would have been advisable to reduce this interval.

Chapter 4

Testing Apparatus and Results

4.1 Testing Networking

Communications were tested by connecting a laptop, running both a MQTT message broker and a publishing client, to the Arduino. The MQTT message broker used was an open source broker for mac called "Mosquito" [33]. Command line publishing, facilitated by the same package, was used to publish messages to the */Microscope* topic. The system performed well, responding almost instantly to any published commands.

4.2 Testing Motor Performance

4.2.1 Testing Slide Transitions

To verify that the slide changer was accurately returning to slides after switching a test was devised using a printed "+" symbol with dotted lines (with a spacing of 0.5mm). This was put in the place of a slide and manually centred using slow x and y commands. The microscope was set to 200x magnification and put into focus. The system was commanded to move randomly between slides for a certain number of times before returning to the + slide. The results are shown in table 2.

Table 2: The captured camera position after a certain number of slide transitions.

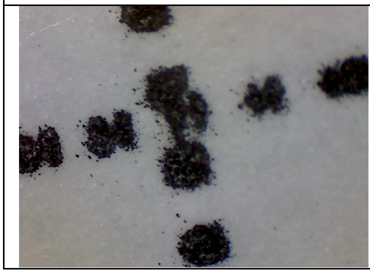
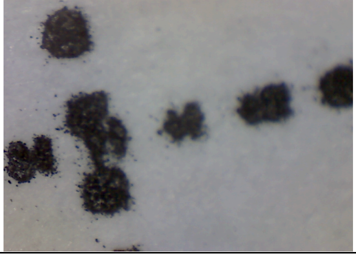
Initial Position	One change and return	Ten changes
		

Table 2 indicates that, while the error in slide changing was negligible over two transitions, it increased to a position error of approximately 0.5mm after ten transitions.

4.2.2 Measuring X/Y/Z Motor Wobble

During testing of the x and y stages a pronounced cyclical wobble in the linear actuation was observed. The magnitude of this effect was measured by using the same dotted + used in the previous section. The magnification was reduced to 20x so that origin of the + remained in frame and the effects over one full cycle of the wobble could be measured. Videos were taken so that the maximum amplitude of the effect could be easily identified. The amplitude of the wobble was observed to be similar in both cases. Figure 28 is a compiled image showing the maximum amplitude of the wobble in the x stage.

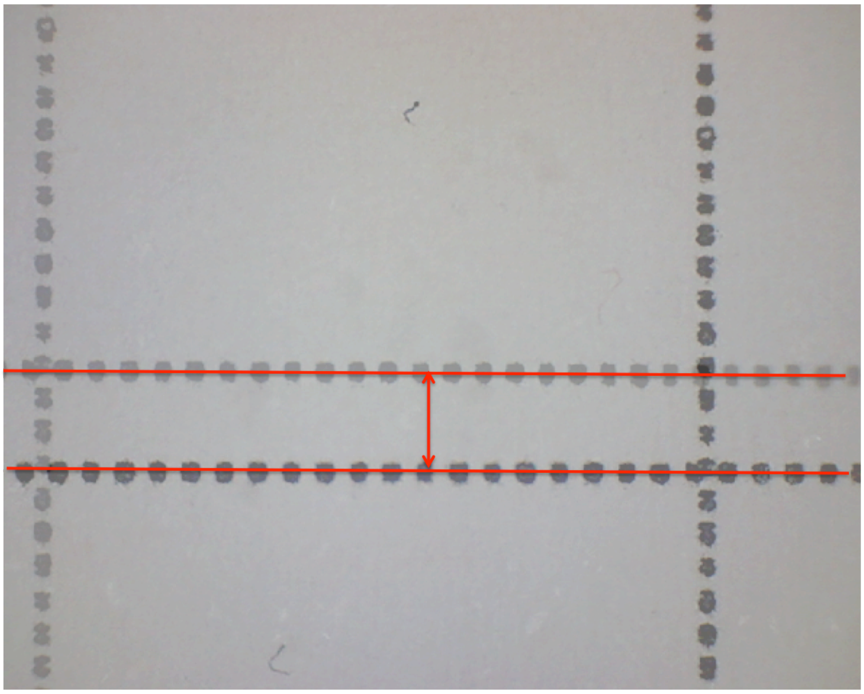


Figure 28: A composite image showing the maximum amplitude of the wobble observed in the x stage.

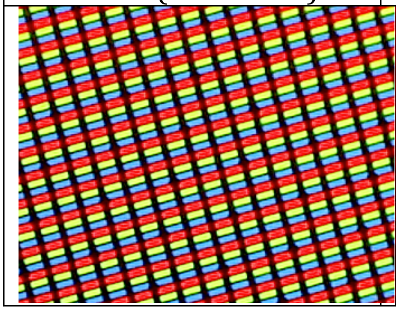
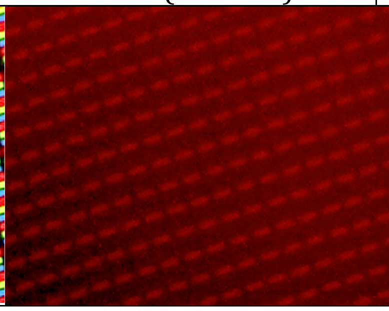
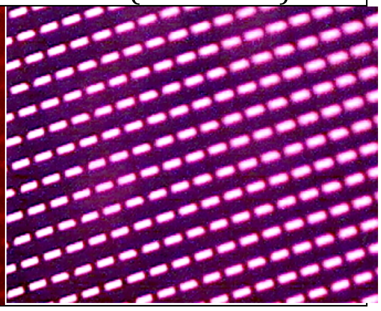
The maximum wobble amplitude was measured as 2mm over a 10mm linear movement. It was also noted that if the motor was moved back the same distance it would arrive within 0.2mm of the original starting position.

4.3 Testing Camera Performance

4.3.1 Specimen Emission Filtering

In order to assess the transmission filtering system an LCD display was used. By placing the display under the microscope and magnifying 200x the individual red, green and blue sub pixels could be seen. An image was captured before and after placing a red filter lens in between the camera and the display. A third image - an unfiltered picture of the LCD screen with only red pixels active - was taken for comparison. The three captures are shown in table 3.

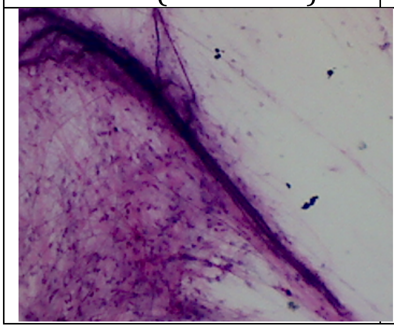
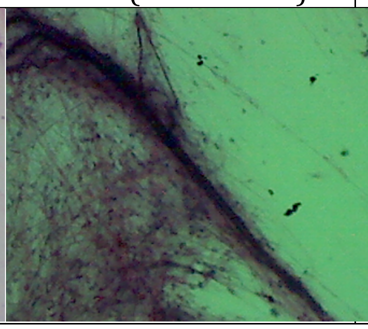
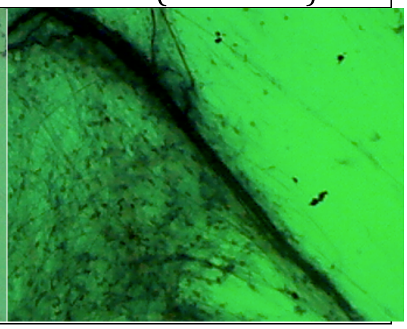
Table 3: A comparison of the effects of filter lenses on a RGB LCD screen viewed at 200x magnification.

White (Unfiltered)	White (Filtered)	Red (Unfiltered)
		

4.3.2 Light Source Filtering

A magenta stained specimen was used to assess the quality of the bottom filter lenses. The specimen was lit using the transmission light source both unfiltered and filtered with a green lens with the aim of improving contrast in the image (as green is the opposite colour to magenta). The white light source was replaced with a green LED so that the effectiveness of the filter could be measured. The resulting images are shown in table 4.

Table 4: A comparison of a magenta stained specimen viewed under filtered light and green light

White (Unfiltered)	White (Green filter)	Green (Unfiltered)
		

Although it is difficult to see in the images presented in the table, the unfiltered green light greatly improved the contrast of the image. The improvement in the filtered image was not as substantial as it failed to filter all of the magenta light.

4.3.3 Testing of Advanced Illumination Techniques

Unfortunately, due to a delay in oversees shipping, neither the fluorescence filters nor the polarised filters arrived in time for testing. The dark-field illumination technique was tested unsuccessfully on a specimen slide. Figure 29 shows the result of the test.

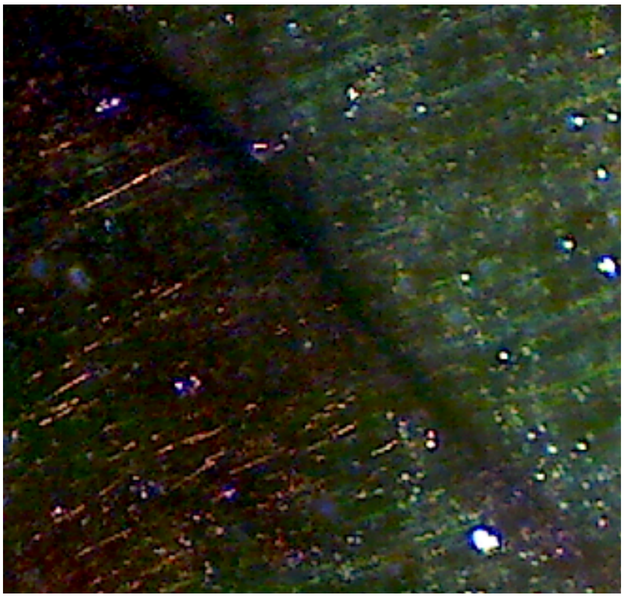


Figure 29: The results of the dark-field illumination test on the same specimen shown in table 3.

It was speculated that the failure of this illumination technique was due to aberrations in the slide glass (seen as an arcing pattern across the image). These aberrations were lit by the scattering effect more than the specimen edges causing them to be washed out. The experiment was repeated with two freestanding pieces of hair. Figure 30 showed that the illumination technique did work but that camera artefacts in the black portions of the image reduced the image quality.

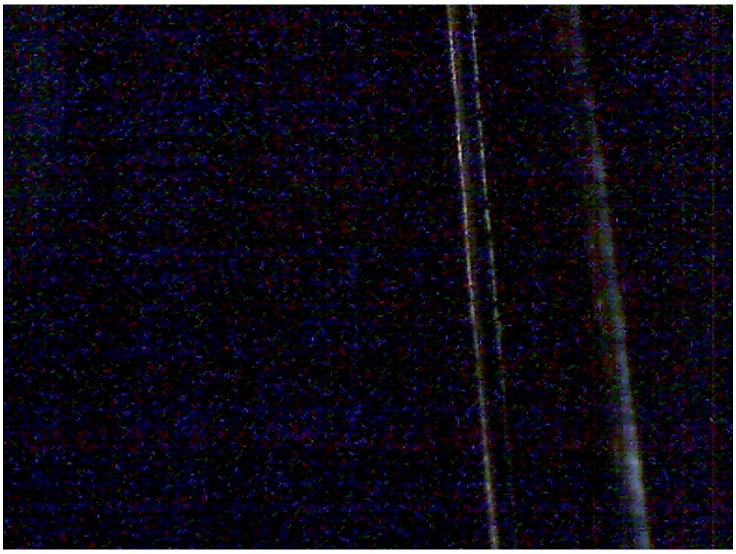


Figure 30: The results of the dark-field illumination test on two pieces of freestanding hair (without low quality slide glass)

Chapter 5

Discussion

5.1 Communication Speed and Reliability

It was concluded, from the testing outlined in section 4.1, that the MQTT protocol, and its associated implementation in the project, worked very well and satisfied the project requirements. Although the rapid response time can be attributed to the low ping between the client and microscope in the test, it can be inferred that the program itself did not slow down the communication significantly.

The reliability of the connection was very good with no observed disconnections between the microscope and the service broker. MQTT's quality of service feature ensured that messages sent from the client were received and only received once by the microscope. The downside of the design was that the system lacked any form of confirmation to the client other than the observable response of the microscope motors. This wasn't required in a proof-of-concept, locally controlled system but would be required if the microscope was controlled over the Internet (disconnection is more likely). This could be implemented by programming the microscope to publish confirmation messages to another topic.

5.2 Hardware Performance and Robustness

5.2.1 Lens and Slide Holder Assembly's

The lens and slide motors performed well under testing (section 4.2.1). The measured shift in absolute position on the slide was small enough (0.5mm) that it would not overly affect the system over a short to medium number of transitions. Using the photointerrupter to reset the position after a certain number of transitions would have compensated for the shift.

The only significant problem with the slide and lens assemblies was a slight tilt that caused them to change height as they rotated (~1mm in lens holders and ~3mm in slide holder). However, there was sufficient clearance to prevent interference between the disks. The only appreciable affect was that transitioning between slides would put the microscope out of focus. This tilt was likely due to elastic deformation in the motor supports, and could be fixed by increasing the aluminium thickness.

5.2.2 Linear Stages

The linear stages had the high level of precision and adequate speed to fit the project specification. The performance, however, was less than desirable due to the wobble analysed in section 4.2.2. While the stages on their own were accurately moving the desired distance, the wobble was producing an error in the perpendicular direction of up to 20%. It was hypothesised that the wobble was caused by elastic deformation of the threaded shaft under loading, as the effect was not present when the stages were removed from the assembly. In order to combat this, additional support rods could be added. Alternatively, the movement could have been compensated for in the programming as the error was not random but in a measurable cyclical pattern.

5.2.3 Motor Controller Performance

The project electronics performed well under testing. The only problem observed was an occasional overheating of the voltage regulators when the motors were run for an extended period of time. Placing additional voltage regulators in parallel temporarily solved the problem by allowing them to take turns powering the circuit. When one regulator got too hot and the voltage started to drop another would take over, allowing the first to cool down. This was not a good solution in the long term, as it would damage the regulators. Larger heat sinks on the regulators and adequate airflow would solve this problem properly.

5.2.4 Transmission Light Performance

The transmission light performed adequately under the tests performed in section 4.3. One noticeable problem was that the blue light, generated by both the white led and the UV led, was stronger than the rest of the spectrum. Fortunately, the camera on the USB microscope compensated for the colour shift so the effect was not noticeable in the magnified images.

5.3 Imaging Quality

5.3.1 Overall Camera Quality

The quality of the camera was too low to meet a university level. The number of visual artefacts in the dark sections of the images was too great and would make it difficult to see the effects of visible fluorescence. The wide viewing angle meant that the resolution of the image was relatively low and would not provide the level of definition required to study smaller specimens in much detail. The camera functioned satisfactorily for a proof-of-concept system, but if the multifunctional remote microscope were going to be used by university students a more expensive microscope module, with higher resolution and magnification, would be required.

The inability to control the microscope's colour adjustment was also a significant problem and is shown clearly in table 3. The unfiltered red image was shifted towards blue, producing a pink image rather than the red that is clearly seen in the unfiltered white image. The automatic colour adjustment would make it difficult to do any post-processing in an advanced client interface.

5.3.2 Transmission Filter Quality

The printed transmission filters proved to be quite effective at their job. Not only did the red filter completely remove both the green and blue sub-pixels but it also supressed the automatic colour adjustment of the camera. This was due to a significant decrease in intensity but was easily overcome by increasing the source light intensity.

5.3.3 Source Filter Quality

Coloured source filter performance was average compared to the transmission filters. They caused a significant drop in intensity for very little gain in contrast. (table 4). In comparison, the coloured light source performed much better. A controllable RGB light source would be a good substitute for coloured filtering of white light.

5.3.4 Advanced Illumination Performance

Advanced illumination performance was unsatisfactory. Although neither the fluorescence or polarisation techniques were tested, it can be assumed from the dark-field performance that the camera artefacts would ruin both of these low-light illumination methods. The dark-field technique was shown to work to some extent but, for any of these methods to be viable, a more expensive USB microscope would have to be purchased.

Chapter 6

Conclusions

6.1 General Conclusions

The project successfully delivered a proof-of-concept, remotely controlled microscope for university-level mineralogical and electrical applications. However, there were several significant shortcomings in the optical quality that would prevent it from being a low-cost system for biology experiments. The most significant problem was that the camera resolution and contrast ratio were simply too low to meet the required standards. The camera would have to be replaced with a more expensive model, doubling the cost of the system. Additionally, the quality of the filter lenses left a lot to be desired. The bandwidth achievable in the bottom filter lenses was not small enough to be useful. These lenses would have to be replaced with expensive optical quality glass lenses, adding significantly to the cost.

Positive outcomes were achieved from both the actuation scheme and the network communication scheme. It was successfully demonstrated that a system with a large number of stepper motors could be controlled by a low-cost motor driver setup and with only a few digital I/O's. The MQTT communication protocol was also validated as one of the best methods of controlling an iLab device.

6.2 Future Work

6.2.1 Improving the Optical System

Improvement of the optical system is the key change required to make this microscope a viable tool for biologists. Upgrading from the $70 USB microscope module to a $300 one [9] with greatly improved contrast and resolution would let it to perform the low-light experiments it was designed for. Additionally, purchasing optical grade filters and an off-the-shelf light source would open up additional optical illumination methods like phase contrast illumination.

6.2.2 Changing to a New Platform

An alternative approach to the problem would be to instrument a laboratory grade microscope. The advantages this would have over improving on the current design would be an increase in robustness and greater optical quality. The downside would be a much greater cost.

Elements of the current design could be brought across to reduce some of the initial cost. The motor control and networking scheme have both been proven to work well and could be implemented in an instrumented microscope. The filter system has also demonstrated potential, although higher quality lenses would be required.

Making the microscope accessible to a larger audience could further offset this cost. By adding additional command functionality, such as moving to specific coordinates, the microscope could be used more easily by people with slower Internet connections. Increasing the slide capacity would also facilitate a greater number of student users, reducing the number of microscope systems required.

6.2.2 Client-Side Interface

A user friendly, well-designed, client interface is another key component that will have to be implemented for the remote microscope to function as a useful external learning tool. A GUI with a representation of a traditional microscope, along with all the appropriate control buttons, would help to cement the students' understanding of how the microscope works. Additional features, like the ability to tag structures for everyone to see and a tour mode that guides students around slides by replaying MQTT commands, would further increase the collaborative learning potential of a remotely controlled microscope.

Appendix A. Complete Arduino Sketch

```
#include <Stepper.h>
#include <PubSubClient.h>
#include <SPI.h>
#include <Ethernet.h>
#include <String.h>

byte mac[] = {0x90, 0xA2, 0xDA, 0x00, 0x01, 0x1B};
byte ip[] = {192, 168, 0, 77};
byte server[] = {192, 168, 0, 3};

// number of stepper motor steps:
#define MOTOR_STEPS 200

// number of slides and lenses
#define SLIDES 10
#define LENSES 5

// control pins for the stepper motor:
#define MOTOR_PIN1 6
#define MOTOR_PIN2 7
#define MOTOR_PIN3 8
#define MOTOR_PIN4 9

// control pins for the demux
#define DEMUX_A 4
#define DEMUX_B 3
#define DEMUX_C 2

// motor demux numbers
#define SLIDE_MOTOR          2
#define TOP_LENSE_MOTOR      6
#define BOT_LENSE_MOTOR      4
#define X_MOTOR              0
#define Y_MOTOR              1
#define Z_MOTOR              3


// define the no motor selection
#define OFF -1

// define motor speed (rpm)
#define M_SPEED 60
```

```
// create the stepper instance:
Stepper MyStepper = Stepper(MOTOR_STEPS, MOTOR_PIN1, MOTOR_PIN2,
MOTOR_PIN3, MOTOR_PIN4);

PubSubClient client(server, 1885, callback);

//global variables
int currentSlide = 0;           // current slide selected
int currentTopLense = 0;        // current top lens selected
int currentBotLense = 0;        // current bottom lens selected
int currentMotor;               // current motor running
int currentDirection;           // 1 or -1 (multiplier)
int kill = 1;                   // 1 when no motor running, 0 otherwise

// Initialise the demux
void initSelector() {
	pinMode(DEMUX_A, OUTPUT);
	pinMode(DEMUX_B, OUTPUT);
	pinMode(DEMUX_C, OUTPUT);
}

// Select the motor to run by specifying the demux number
void selectMotor(int motor) {
	switch (motor) {
		case 0:
			digitalWrite(DEMUX_A, LOW);
			digitalWrite(DEMUX_B, LOW);
			digitalWrite(DEMUX_C, LOW);
			break;
		case 1:
			digitalWrite(DEMUX_A, HIGH);
			digitalWrite(DEMUX_B, LOW);
			digitalWrite(DEMUX_C, LOW);
			break;
		case 2:
			digitalWrite(DEMUX_A, LOW);
			digitalWrite(DEMUX_B, HIGH);
			digitalWrite(DEMUX_C, LOW);
			break;
		case 3:
			digitalWrite(DEMUX_A, HIGH);
			digitalWrite(DEMUX_B, HIGH);
			digitalWrite(DEMUX_C, LOW);
			break;
		case 4:
			digitalWrite(DEMUX_A, LOW);
			digitalWrite(DEMUX_B, LOW);
			digitalWrite(DEMUX_C, HIGH);
			break;
```

```
                case 5:
                        digitalWrite(DEMUX_A, HIGH);
                        digitalWrite(DEMUX_B, LOW);
                        digitalWrite(DEMUX_C, HIGH);
                        break;
                case 6:
                        digitalWrite(DEMUX_A, LOW);
                        digitalWrite(DEMUX_B, HIGH);
                        digitalWrite(DEMUX_C, HIGH);
                        break;
                default:
                        digitalWrite(DEMUX_A, HIGH);
                        digitalWrite(DEMUX_B, HIGH);
                        digitalWrite(DEMUX_C, HIGH);
        }
}

// Drive the specified motor for the given steps.
// Delays until the motor has finished running
void driveMotor(int motor, int steps) {
        selectMotor(motor);
        MyStepper.step(steps);
}

// turn to the selected slide
void setSlide(int slideNum) {
        MyStepper.setSpeed(M_SPEED);
        int steps = (slideNum - currentSlide) * (MOTOR_STEPS/SLIDES);
        driveMotor(SLIDE_MOTOR, steps);
        currentSlide = slideNum;
}

// turn to the selected top lens
void setTopLense(int lenseNum) {
        MyStepper.setSpeed(M_SPEED);
        int steps = (lenseNum - currentTopLense) * (MOTOR_STEPS/LENSES);
        driveMotor(TOP_LENSE_MOTOR, steps);
        currentTopLense = lenseNum;
}

// Turn to the selected bot lens
void setBotLense(int lenseNum) {
        MyStepper.setSpeed(M_SPEED);
        int steps = (lenseNum - currentBotLense) * (MOTOR_STEPS/LENSES);
        driveMotor(BOT_LENSE_MOTOR, steps);
        currentBotLense = lenseNum;
}
```

```
// payload format: mds || m#
// m = motor:
//     - s = slide motor
//     - t = top lense motor
//     - b = bottom lense motor
//     - x = x motor
//     - y = y motor
//     - z = z motor
//     - k = kill motors
// d = direction
//     - f = forward
//     - b = backward
// s = speed (RPM)
// # = select slide/lense number
void callback(char* topic, byte* payload, unsigned int length) {
	int i;
	// check if its a kill command
	if (payload[0] == 'k') {
		kill = 1;
		Serial.println("kill motors");
		return;
	} else {
		kill = 0;
	}
	// determine what motor to operate and which direction
	switch ((char) payload[0]) {
		case ('s'):
			kill = 1;
			i = payload[1] - '0';
			Serial.print("Change to slide ");
			Serial.println(i);
			setSlide(i);
			return;
		case ('t'):
			kill = 1;
			i = payload[1] - '0';
			Serial.print("Change to top lense ");
			Serial.println(i);
			setTopLense(i);
			return;
		case ('b'):
			kill = 1;
			i = payload[1] - '0';
			Serial.print("Change to bottom lense ");
			Serial.println(i);
			setBotLense(i);
			return;
		case ('x'):
			currentMotor = X_MOTOR;
```

```
			Serial.print("drive x motor ");
			if ((char) payload[1] == 'f') {
				currentDirection = 1;
				Serial.print("forward at ");
				sscanf((char*)payload, "xf%d", &i);
				Serial.print(i);
				Serial.println("RPM");
				MyStepper.setSpeed(i);
			} else if ((char) payload[1] == 'b') {
				currentDirection = -1;
				 Serial.print("backward at ");
				sscanf((char*)payload, "xb%d", &i);
				Serial.print(i);
				Serial.println("RPM");
				MyStepper.setSpeed(i);
			} else {
				kill = 1;
			}
			break;
		case ('y'):
			currentMotor = Y_MOTOR;
			Serial.print("drive y motor ");
			if ((char) payload[1] == 'f') {
				currentDirection = 1;
				Serial.print("forward at ");
				sscanf((char*)payload, "yf%d", &i);
				Serial.print(i);
				Serial.println("RPM");
				MyStepper.setSpeed(i);
			} else if ((char) payload[1] == 'b') {
				currentDirection = -1;
				Serial.print("backward at ");
				sscanf((char*)payload, "yb%d", &i);
				Serial.print(i);
				Serial.println("RPM");
				MyStepper.setSpeed(i);
			} else {
				kill = 1;
			}
			break;
		case ('z'):
			currentMotor = Z_MOTOR;
			Serial.print("drive z motor ");
			if ((char) payload[1] == 'f') {
				currentDirection = 1;
				Serial.print("forward at ");
				sscanf((char*)payload, "zf%d", &i);
				Serial.print(i);
				 Serial.println("RPM");
```

```
                                        MyStepper.setSpeed(i);
                                } else if ((char) payload[1] == 'b') {
                                        currentDirection = -1;
                                        Serial.print("backward at ");
                                        sscanf((char*)payload, "zb%d", &i);
                                        Serial.print(i);
                                        Serial.println("RPM");
                                        MyStepper.setSpeed(i);
                                } else {
                                        kill = 1;
                                }
                                break;
                        default:
                                kill = 1;
                                return;
        }
}

void setup() {
        // set the motor speed to 60RPMs (1 per sec):
        MyStepper.setSpeed(M_SPEED);
        // initialise the demux
        initSelector();
        // Initialise the serial port
        Ethernet.begin(mac, ip);
        Serial.begin(9600);
        if (client.connect("microscope")) {
                client.subscribe("/Microscope");
                Serial.println("connected");
        }
}

void loop() {
        if(!client.loop()) {
                client.connect("microscope");
                client.subscribe("/Microscope");
                Serial.println("reconnected");
        }
        if (!kill) {
                driveMotor(currentMotor, currentDirection*20);
                Serial.print('.');
        } else {
                selectMotor(OFF);
        }
}
```

Bibliography

[1] Yilun, F, 2011, ‘Computer Controlled Microscope’, BE thesis, University of Queensland, Brisbane.

[2] Triola, MM & Holloway, WJ 2011, ‘Enhanced virtual microscopy for collaborative education’, *BMC Medical Education*, vol. 11, no. 1, pp. 4.

[3] Harward, V J, del Alamo, J A, Lerman, S R et al. 2008, ‘The iLab Shared Architecture: A Web Services Infrastructure to Build Communities of Internet Accessible Laboratories’, *Proceedings of the IEEE* , vol.96, no.6, pp.931-950.

[4] Introduction to Microscopy 2007, Viewed March 2012, <http://www.freeinfosociety.com/article.php?id=118>

[5] Quercioli, F 2011, ‘Fundamentals of Optical Microscopy’, *Optical Fluorescence Microscopy,* Springer, Berlin.

[6] Dark-Field Microscopy 2012, Viewed March 2012, <http://en.wikipedia.org/wiki/Dark_field_microscopy>

[7] Fluorescence Microscopy Digital Image Gallery 2012, Viewed March 2012, <http://www.microscopyu.com/galleries/fluorescence/index.html>

[8] Veho 400x USB Microscope 2011, Viewed March 2012, <http://walyou.com/usb-microscope-ces/>

[9] Digital Microscopes 2012, Viewed October 2012, <http://www.microscope-shop.com.au/digital.aspx>

[10] R. Maturo, G.Kath, R. Zeigler and P. Meechan, 1997, ‘Control of a Remote Microscope Over the Internet’, BioTechniques, vol. 22, No. 6, p1154.

[11] Stepper motor – 29Oz.In ROB-10848 2012, Viewed March 2012, <http://littlebirdelectronics.com/products/stepper-motor-29-oz-in-200-steps-rev-threaded-shaft>

[12] Pololu 2012, *Linear Actuators,* Viewed October 2012, <http://www.pololu.com/catalog/category/127>

[13] Pololu 2012, *Pololu Universal Aluminium Mounting Hub for 5mm Shaft Pair,* Viewed March 2012, <http://www.pololu.com/catalog/product/1203>

[14] Pololu 2012, *Stepper Motors,* Viewed March 2012, <http://www.pololu.com/catalog/category/87>

[15] Pololu 2012, *37D mm Gearmotors*, Viewed March 2012, <http://www.pololu.com/catalog/category/116>

[16] Pololu 2012, *Stepper Motor: Bipolar, 00 Steps/Rev, 28x32mm, 3.8V, 670mA*, Viewed March 2012, <http://www.pololu.com/catalog/product/1205>

[17] Pololu 2012, *Stepper Motor: Biplar, 200 Steps/Rev, 35x28mm, 10V, 500mA*, Viewed March 2012, <http://www.pololu.com/catalog/product/1208>

[18] Sparkfun 2012, *EasyDriver Stepper Motor Driver,* Viewed June 2012, <https://www.sparkfun.com/products/10267>

[19] Freetronics 2012, *TwentyTen,* Viewed April 2012, <http://www.freetronics.com/products/twentyten>

[20] Arduino 2012, *Arduino Ethernet Shield*, Viewed April 2012, <http://www.arduino.cc/en/Main/ArduinoEthernetShield>

[21] Arduino 2012, *Arduino Board Uno,* Viewed April 2012, <http://arduino.cc/en/Main/ArduinoBoardUno>

[22] Texas Instruments 2003, *CD4051B*, Viewed July 2012, <http://www.farnell.com/datasheets/1633807.pdf>

[23] Texas Instruments 2003, *SN54AHCT138*, Viewed July 2012, <http://www.ti.com/lit/ds/symlink/sn74ahct138.pdf>

[24] Texas Instruments 1995, *SN754410,* Viewed April 2012, <http://www.farnell.com/datasheets/75370.pdf>

[25] National Semiconductor 2005, *LM2940,* Viewed April 2012, <http://www.farnell.com/datasheets/7427.pdf>

[26] L4940XX5, Viewed April 2012, <http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/CD00000442.pdf>

[27] Cree 2010, *Cree 5mm Round LED C503C-WAS/WAN Data Sheet,* Viewed August 2012, <http://www.farnell.com/datasheets/1640468.pdf>

[28] UV LED LAMP VAOL-3GUV8Y4, Viewed August 2012, <http://vcclite.com/wp-content/files/VAOL-3GUV8Y4-LED-3mm-UV.pdf>

[29] IR-Lumineszenzdiode (850mm) SFH4350, Viewed August 2012, <http://www.farnell.com/datasheets/43569.pdf>

[30] O'Leary, K 2011, *Arduino Client for MQTT,* Viewed April 2012, <http://knolleary.net/arduino-client-for-mqtt/>

[31] Arduino 2012, *Stepper Library,* Viewed April 2012, <http://www.arduino.cc/en/Reference/Stepper>

[32] MQTT.ORG 2012, *MQ Telemetry Transport*, Viewed April 2012, <http://mqtt.org/>

[33] Mosquitto: An Open Source MQTT v3.1 Broker, Viewed October 2012, <http://mosquitto.org/>