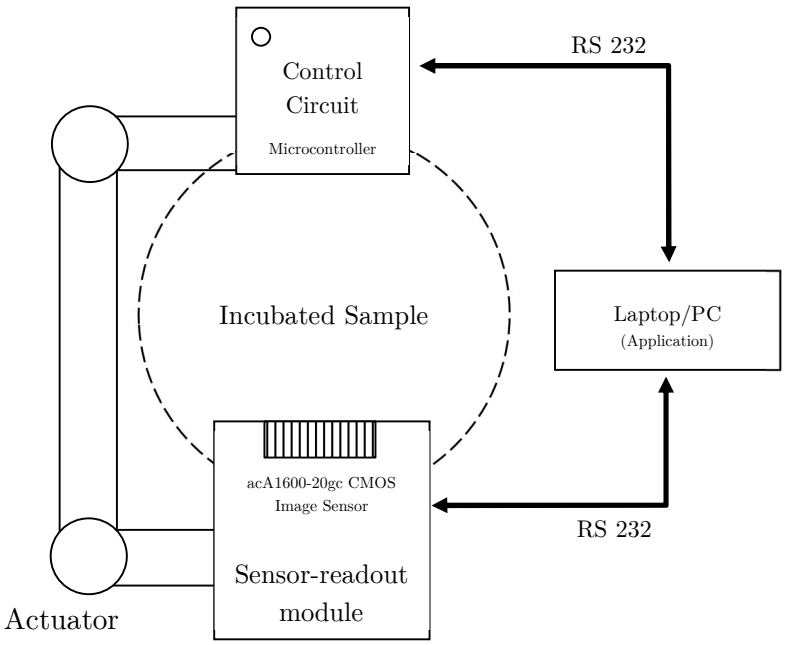
Visual Sepsis Detector

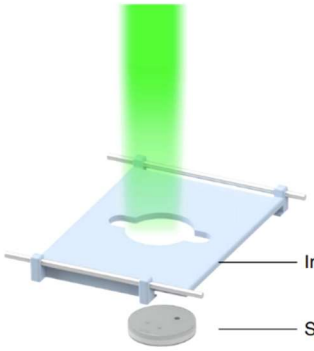
Microscope Computer System

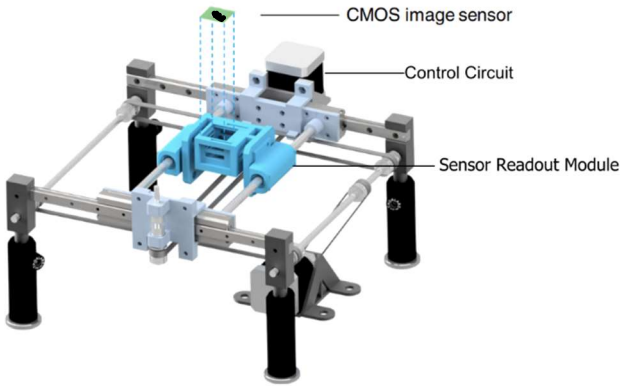
Our microscope has the following components:

- Sensor-readout module (Electrical team).
    - acA1600-20gc CMOS image sensor with a 4x objective lens attached.
    - Microcontroller PIC18F4450 for reading image sensor data.
- Translation stage powered by an actuator, either Accumini 2AD10AAAHL or any Zaber product along with linear bearings. Movement powered by two-stepper motors, one for each direction with a controller chip. May need to use 3D printing (Mechanical team).
- RS 232 USB communication between sensor-readout module, translation stage and external PC (Electrical & Mechanical team).
- Control circuit (Electrical team)
    - Houses the central microcontroller.
    - Stepper motor driver carrier to control translation stage.
    - Transistor-based digital switch (SUP75P03-07) to control the image sensor.
- Microscope incubator housed by a 3D-printed frame (Mechanical team).
- External Laptop/PC running application (Software team).
    - Image processing & image stitching.
    - Deep learning model.

This is a bench-microscope with a single image sensor with a microcontroller connected to it. It will use one-dimensional (1D) array CCD/CMOS image sensor inside a sensor readout module. The two-axis translation stage allows us to change where the microscope is looking at. We will use acA1600-20gc from Basler as our image sensor, which has 1624 × 1234 pixels resolution meaning each pixel is around 4.4 μm × 4.4 μm.

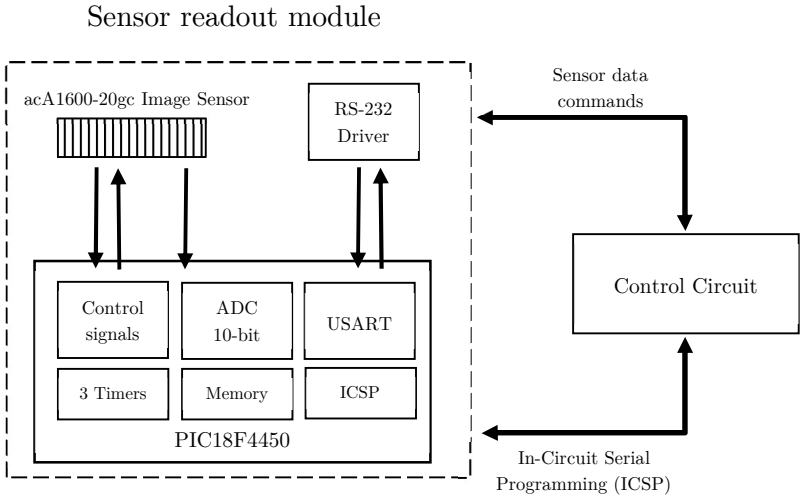






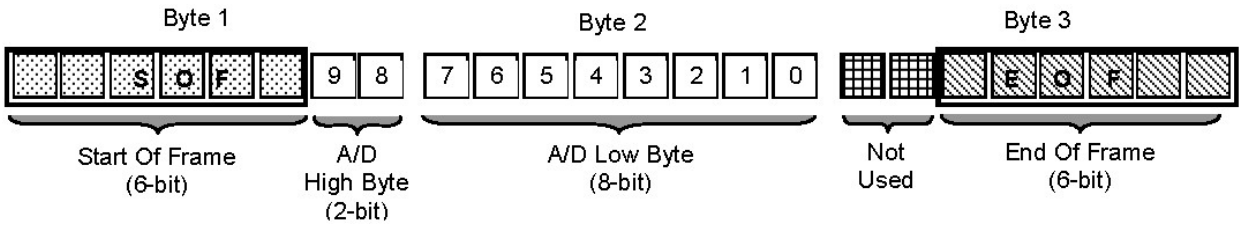
Sensor readout module

An SRM is responsible for reading out the signals from all the pixels composing each row and each column of the array to assemble an image from the photodiode charge accumulation data. The signals from all the pixels must be accurately detected and measured. We will use MATLAB's Instrument Control toolbox to handle the RS232 communications. We will use a PIC18F4450 microcontroller inside the sensor readout module. Our image sensor is a cheap one and does not have exactly have full computer-like capabilities as a regular full camera might have.



ICSP (In-Circuit Serial Programming) connection allows us to program the microcontroller to allow control of the image sensor from the Control Circuit outside.

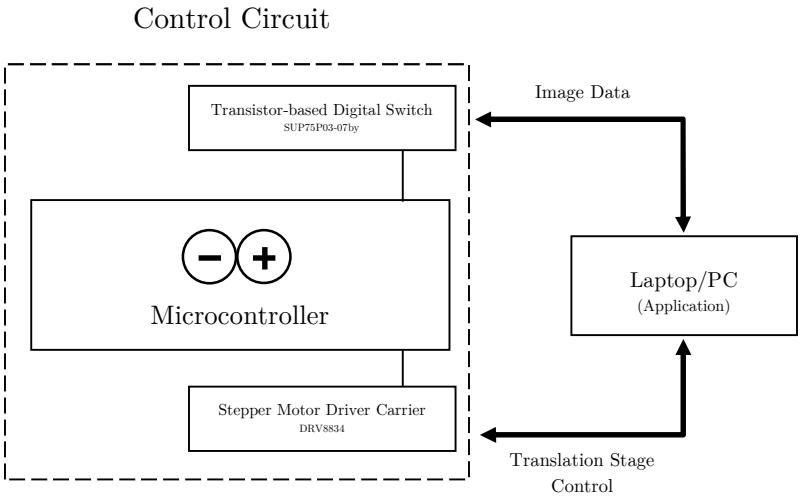
The RS-232 is a serial port used to communicate between the sensor readout module and the PC. Uses a baud rate of 59200 bps. Timeslot width (time to transfer one unit of data) is 900 µs. We will say $T_{\text{cycle}} = 900$ µs. Therefore, the acquisition of all 1024 by 1024 sensor pixels takes approximately one second. We'll try the RS-232 but it's highly subject to change.



There is some areas that the image sensor covers that are not necessary. We can use the ICSP functionality to specify a region of interest (ROI). ROI is the number of pixels that are being recorded. With a lower ROI, the acquisition rate is increased. We should be able to control the ROI through our MATLAB application.

Control Circuit

The control circuit will be used to get data from the image sensor to out MATLAB application and control the translation stage. It will house the microcontroller (a good choice would be an Arduino Micro but to be decided by software team), control chips for the motor drive (an example would be DRV8834 from Pololu) and a digital switch for controlling the CMOS image sensor connection (we will use SUP75P03-07 Vishay Siliconix).



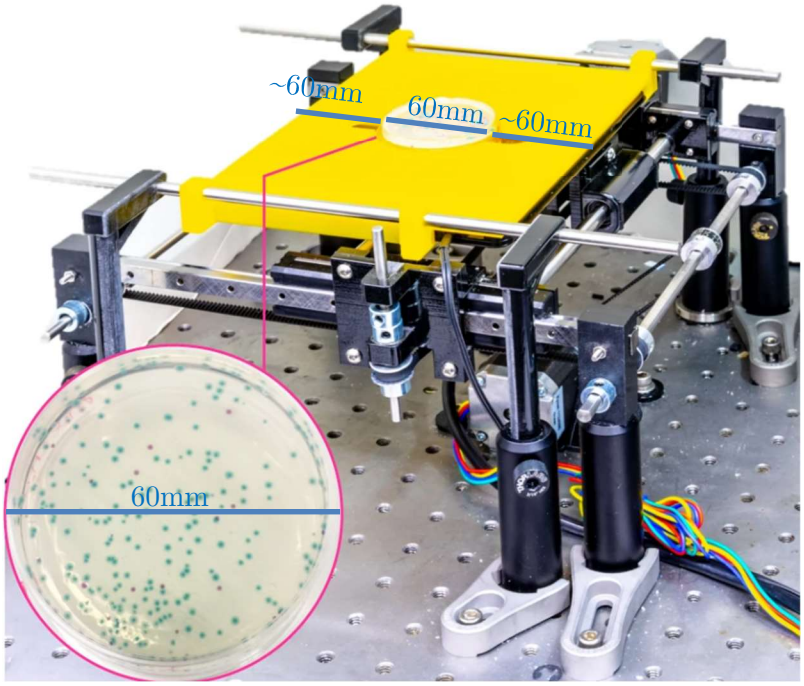
Translation Stage (by Mechanical team)

We will use actuators that move the microscope lens around the sample. This approach can make our device cheaper while still generating a high-resolution image. We can use actuators from Zaber (https://www.zaber.com/), up to the mechanical team to decide which component they'd like to use. We call this XY axis area the translation stage. Zaber even appears to already have a component specific for scanning microscopes such as this one (https://www.zaber.com/products/scanning-microscope-stages). Both electrical and mechanical engineers may need to work together to set up the translation stage.

Incubator (by Mechanical team)

The incubator will be within the translation stage. It keeps the agar plate sealed away from outside interference/particles. In the UCLA study (https://www.nature.com/articles/s41377-020-00358-9), they've used a special microscope incubator (Tokai Hit Stage Top® Incubator) housed by a 3D-printed frame. The frame appears to be around 15-20 mm thick, which is as thick as the agar plate.

The agar plate that the lens will rotate around is 60 mm in diameter and around 15 mm thick. The area of the translation stage may depend on the actuator/movement component used, but should be approximately 180 mm by 180 mm. This size is not specifically required for anything, but the agar plate must be 60 mm in diameter.



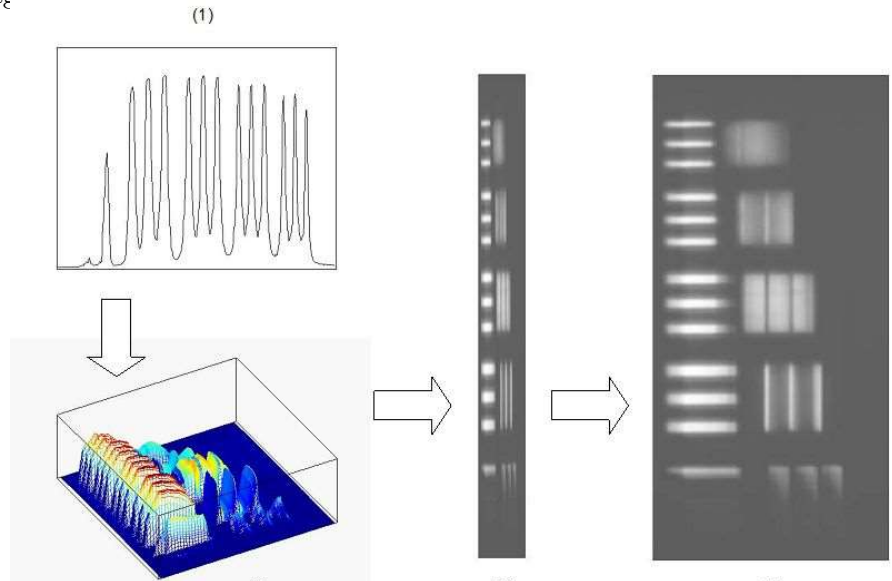
Translation Stage (by Electrical team)

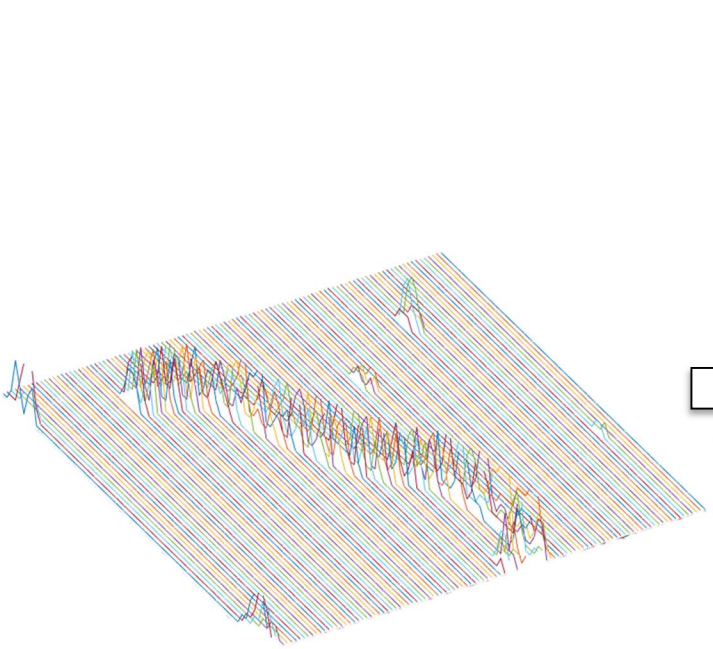
Communications settings for the translation stage through the RS232 will use 9600 baud, no hand shaking, no parity, one stop bit. Instructions will consist of a group of 6 bytes.

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Unit #	Command #	Data (LSB)	Data	Data	Data (MSB)

Bytes 3, 4, 5 and 6 are data in long integer, 2's complement format with the least significant byte transmitted first. How the data bytes are interpreted depends on the command byte (Byte 2). Essentially each byte will receive data from a single pixel and use it to build a coherent 2D image.

In our application, we will have several samples of different microorganisms/bacteria. They will be 3D wave objects of certain bacteria which will look like (2) in the figure below. The device will process the image (3) into what we see on (4). (1) reflects the data from one sensor readout. What we're seeing here is what the sensor accepts as input and therefore what we need to feed into it to get an image





(a) the image sensor creates a 3D graph out of a series of 2D graphs that each of its photon cells detects. (Oversimplified)

(b) 3D graph gets converted into an unprocessed 2D image through image processing.

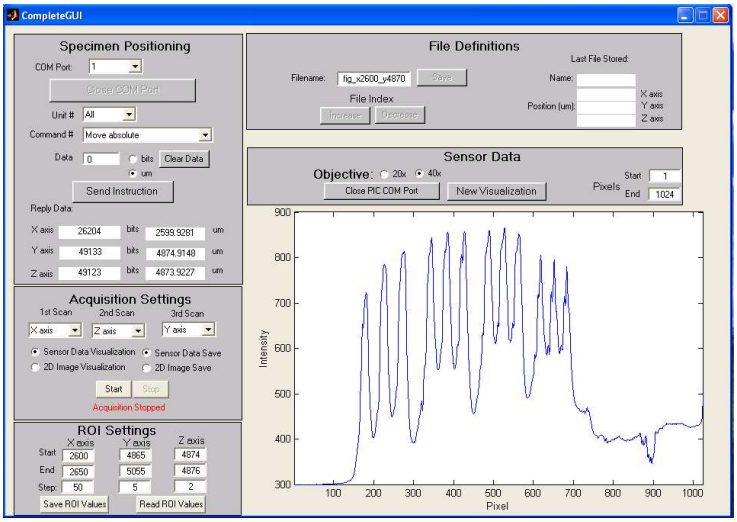
(c) We can use more image processing to turn it into a coherent image.

Proposed GUI

We will need a GUI that allows us to see what the device is recording. This GUI will also allow us to control the translation stage. In general, our GUI will:

- Control the translation stage with actuators.
    - User can control the order of scanning in the three axes.
    - Use can control the region-of-interest (ROI) in the three axes.
    - Sends commands to the actuators through the RS232 serial port.
- Sensor data readout
    - Send a command to the sensor readout module to signal acquisition start through the RS232 serial port.
    - Receive the sensor data through the RS232.
- Image
    - Real-time visualization of each sensor readout, i.e., 1D image that shows sensor raw data.
    - Visualization of each 2D image as soon as the acquisition of the set of sensor data is completed.
- Data-files creation
    - Store data files (.csv, Excel, image files) in computer hard disk.
- Deep learning/Computer vision component
    - Functionality that begins the thorough scanning of a sample and applies the computer vision model. It will allow us to see the results of the model.
- Allow alternating between Simulation Mode and Device Mode

Any GUIs we make for the sake of simulation will then be easily used to control the actual fabricated device. The figure below is an example of such a GUI:



Simulation Mode and Device Mode

We will have two modes that we can switch in between through the GUI:

1) Simulation mode – the GUI will control/monitor a simulated 3D device on Simulink.
2) Device mode – the GUI will control/monitor actual fabricated device.

We can use both modes to develop our device.

Image Processing & Image Stitching

https://www.nature.com/articles/s41377-020-00358-9 is an excellent resource for the image processing and analysis required for this device. See the "Image processing and analysis" section to understand how they managed to stitch together their images. Essentially the translation stage moves the image sensor around the sample and records multiple images and then stiches them together into one coherent image. The following is an oversimplification of what they've done:

Five major image processing steps were used for classification and counting of colonies.
1. Image stitching to obtain the image of the entire plate area.
2. Colony candidate selection by differential analysis
3. Deep neural network-enabled detection of growing bacterial colonies
4. Colony counting
5. Calculating of the imaging throughput

Important size information about image stitching

Our image sensor is 1624 px × 1234 px. Each pixel is 4.4 μm × 4.4 μm.
That means each image that the sensor takes covers 7145.6 μm × 5429.6 μm.
Meaning a single image of the sensor covers 7.15 mm × 5.43 mm of the 60 mm agar plate.
We will need approximately ~80 pictures to cover the entire agar plate. Keep that in mind.

Image Quality Assessment

There are two approaches to assess the quality of an image, quantitative and qualitative. The quantitative approach involves using two special functions called the Modulation Transfer Function (MTF) which measures contrast (clarity) and the Point Spread Function (PSF) which measures resolution. MTF is most useful for microscopy applications:

$$M = \frac{I_{\max} - I_{\min}}{I_{\max} + I_{\min}}$$
$I_{\max}$ = maximum intensity value
$I_{\min}$ = minimum intensity value

Or

$$MTF(\xi) = \frac{M_{\text{output}}(\xi)}{M_{\text{sinusoidal}}(\xi)}$$

There may be existing Python code/functions that already implement MTF.

Preparation of samples

We will leave this section to be decided by either a medical student or microbiologist student. One approach we could reverse engineer is what https://www.nature.com/articles/s41377-020-00358-9 have done. We won't do exactly what they've done, but this is just to understand some of the measures taken to record data. Here's the specifics for how they prepared samples:

Studied organisms

E. coli (Migula) Castellani and Chalmers (ATCC® 25922™) (risk level 1), K. aerogenes Tindall et al. (ATCC® 49701™) (risk level 1), and K. pneumoniae subsp. pneumoniae (Schroeter) Trevisan (ATCC®13883™) (risk level 2).

Preparation of the poured agar plates

They used CHROMagar™ ECC (product no. EF322, DRG International, Inc., Springfield, NJ, USA) chromogenic substrate mixture as the solid growth medium for the detection of E. coli and total coliform colonies. CHROMagar™ ECC (8.2 g) was mixed with 250 mL of reagent grade water (product no. 23-249-581, Fisher Scientific, Hampton, NH, USA). The mixture was then heated to 100 °C on a hot plate while being stirred regularly. After cooling the mixture to ~50 °C, 10 mL of the mixture was dispensed into Petri dishes (60 mm × 15 mm) (product no. FB0875713A, Fisher Scientific, Hampton, NH, USA). The agar plates were allowed to solidify, were sealed using parafilm (product no. 13-374-16, Fisher Scientific, Hampton, NH, USA), and were covered with

aluminium foil to keep them in the dark before use. The plates were stored at 4 °C and were used within two weeks of preparation.

Preparation of the melted agar plates

CHROMagar™ ECC (3.28 g) was mixed with 100 mL of reagent grade water using a magnetic stirrer bar, and the mixture was heated to 100 °C. After the mixture cooled to ~40 °C, 1 mL of the bacterial suspension was mixed with the agar and dispensed into Petri dishes. The plates were either incubated in a benchtop incubator (product no. 51030400, ThermoFisher Scientific, Waltham, MA, USA) or the imaging platform (for monitoring the bacterial growth digitally).

Tryptic soy agar was used to culture E. coli at 37 °C and K. aerogenes at 35 °C and nutrient agar to culture K. pneumoniae at 37 °C. Twenty grams of tryptic soy agar (product no. DF0369-17-6, Fisher Scientific, Hampton, NH, USA) or 11.5 g of nutrient agar (product no. DF0001-17-0, Fisher Scientific, Hampton, NH, USA) were suspended in 500 mL of reagent grade water using a magnetic stirrer bar. The mixture was boiled on a hot plate and then autoclaved at 121 °C for 15 min. After the mixture cooled to ~50 °C, 15 mL of the mixture was dispensed into Petri dishes (100 mm × 15 mm) (product no. FB0875713, Fisher Scientific, Hampton, NH, USA), which were then sealed with parafilm and covered with aluminium foil to keep them in the dark before use. The Petri dishes were stored at 4 °C until use.

Preparation of the chlorine-stressed E. coli samples

E. coli was grown on tryptic soy agar plates and incubated for 48 h at 37 °C. Disposable centrifuge tubes (50 mL) were used as a sample container, and the sample size was 50 mL. Five hundred millilitres of reagent grade water were filtered for sterilization using a disposable vacuum filtration unit (product no. FB12566504, Fisher Scientific, Hampton, NH, USA). A fresh chlorine suspension was prepared in a 50 mL disposable centrifuge tube to a final concentration of 0.2 mg/mL using sodium hypochlorite (product no. 425044, Sigma Aldrich, St. Louis, MO, USA), mixed vigorously, and covered with aluminium foil41. Sodium thiosulfate (10% [w/v]) (product no. 217263, Sigma Aldrich, St. Louis, MO, USA) in reagent grade water was prepared, and 1 mL of the solution was filtered using a sterile disposable syringe and a syringe filter membrane (product no. SLGV004SL, Fisher Scientific, Hampton, NH, USA) for sterilization. Water suspensions were prepared by spiking E. coli into filtered water samples. Fifty microlitres of the chlorine suspension (i.e., 0.2 ppm) was added to the test water sample, and a timer counted the chlorine exposure time.

The reaction was stopped at 10 min of chlorine exposure by adding 50 µL sodium thiosulfate into the test water sample and vigorously mixing the solution to immediately stop the chlorination reaction. CHROMagar™ ECC plates were inoculated with 200 µL of the chlorine-stressed suspension, were dried in the biosafety cabinet for at most 30 min and then were placed on the setup for lens-free imaging. In addition, three TSA plates and one ECC ChromoSelect Selective Agar plate (product no. 85927, Sigma Aldrich, St. Louis, MO, USA) were inoculated with 1 mL of the control sample (not exposed to chlorine) and 0.2 ppm of the chlorine-stressed E. coli water sample and dried under a biosafety cabinet for approximately 1–2 h with the gentle mixing of Petri dishes at some time intervals. After drying, the plates were sealed with parafilm and incubated at 37 °C for 24 h. After incubation, the bacterial colonies grown on the agar plates were counted, and the E. coli concentrations of the control samples and chlorine-stressed E. coli samples were compared. If the achieved reduction in colony count was between 2.0 and 4.0 log, then the images of CHROMagar™ ECC plates captured using the lens-free imaging platform were used for further analysis.

Preparation of the culture plates for lens-free imaging

A bacterial suspension in a phosphate-buffered solution (PBS) (product no. 20-012-027, Fisher Scientific, Hampton, NH, USA) was prepared every day from a solid agar plate incubated for 24 h. The concentration of the suspension was measured using a spectrophotometer (model no. ND-ONE-W, Thermo Fisher), and the suspension was then diluted in PBS to a final concentration of 1–200 CFU per 0.1 mL. One hundred microlitres of the diluted suspension was spread on a CHROMagar™ ECC plate using an L-shaped spreader (product no. 14-665-230, Fisher Scientific, Hampton, NH, USA). The plate was covered with its lid, inverted, and incubated at 37 °C in the optical platform (Fig. 2).

Preparation of a concentrated broth

A total of 180 g of tryptic soy broth (product no. R455054, Fisher Scientific, Hampton, NH, USA) was added to 1 L reagent grade water and heated to 100 °C by continuously mixing using a stirrer bar. The suspension was then cooled to 50 °C and filter sterilized using a disposable filtration unit. The broth concentrate was stored at 4 °C and used within 1 week after preparation.

Plan of Action

Our main goal is to:

- Simulate the entire device in Simulink with all realistic electrical, mechanical and software components.
- Fabricate device/order necessary parts.

Here is a proposed roadmap that we will follow for now:

<table>
<tr><th>Electrical team</th><th>Mechanical team</th><th>Software team</th></tr>
<tr><td>1) Work on the sensor-readout module and control circuit in Simulink.
The sensor-readout module is connected to the control circuit via USB or RS-232 (can be either).
The control circuit is connected to the external PC via USB or RS-232.
The sensor readout module contains the image sensor which sends its data to the control circuit.
Control circuit will only contain microcontroller and a digital switch for now.</td><td>1) Work on the CAD design of the translation stage that incubates the agar plate. Transfer Solidworks/Inventor model into the Simulink project.
It has a base with legs.
Has an incubator that covers a 60-mm diameter agar plate that is 15 mm thick.
Area of the stage should be approximately 180 mm by 180 mm, may depend on actuators used.
Create the frame that will house the sensor readout module. It will fit our image sensor which is 7.2 mm $\times$ 5.4 mm, but account for up to three 2-3 mm thick wires that will need to come out of it.</td><td>1) Figure out a workflow/code for training the model and then transferring the model to a microcontroller (decide on which microcontroller to use and let Electrical team know).
Decide on a microcontroller and let electrical team know the microcontroller model.
Decide on programming language, ML framework (likely Python + libraries).
Begin a codebase with a simple foundation for performing computer vision with deep neural networks.</td></tr>
<tr><td colspan="3">Integrate everything together by project lead.</td></tr>
<tr><td>2) Begin work on the control circuit and its communication between the sensor-readout module, translation stage and external PC.</td><td>2) Simulate the translation stage; allowing the image sensor to move around the sample. Work closely with the electrical team.</td><td>2) Create the GUI. Will allow us to control the simulated device and ultimately fabricated device. See section “Proposed GUI”.</td></tr>
<tr><td colspan="3">Integrate everything together by project lead; add MATLAB + Simulink unit tests.</td></tr>
<tr><td>3) Order necessary parts and fabricate device. Improve if failing unit tests.</td><td>3) Order necessary parts and fabricate device. Improve if failing unit tests.</td><td>3) Work on the image processing and image stitching.</td></tr>
<tr><td></td><td></td><td>4) Begin work on model. Start the data collection process and reiterating the model.</td></tr>
</table>

Different methods and techniques

It is completely fine to divert from the original design and components and to use components or a different design that you deem better fit for the purpose of our device. Even if it diverts from the specifications and the output we're expecting, please do it if you feel it's a better choice. Just let us know about it through Discord or ClickUp.

Possible Resources

Zaber (https://www.zaber.com/) is a Vancouver/UBC-based engineering company that makes XY and XYZ movement-based actuators as well as entire automated scanning microscopes such as the ones we're making here. It may be worthwhile to reach out to them and ask for any recommendations of what components we should use or if they have any discounts for student projects and/or research projects. Here are some interesting components that Zaber provides:

- Motorized Microscope stages (https://www.zaber.com/products/scanning-microscope-stages)
- Simple linear stages (https://www.zaber.com/products/linear-stages)
- Automated microscope: this one is very expensive, one of our goals is to make sepsis detection as cheap as possible such that it's favorable to use in medical/hospital settings. But it does have a very interesting design (see picture below) that we could take inspiration from for future reiterations of the device (https://www.zaber.com/products/microscopes).

References

[1] H. Wang et al., "Early detection and classification of live bacteria using time-lapse coherent imaging and Deep Learning," Light: Science & Applications, vol. 9, no. 1, 2020. doi:10.1038/s41377-020-00358-9

[2] M. P. Macedo, "A MATLAB-based microscope," MATLAB Applications for the Practical Engineer, 2014. doi:10.5772/58532

[3] T.-C. Poon and T. Kim, Engineering Optics with MATLAB. World Scientific, 2018.