**McClean Lab manual for the design and use of an automated illumination, culturing, and sampling system for use with optogentic induction systems in *Saccharomyces cerevisiae*.**

**AUTHORS:**

*Stewart, Cameron J.*

*Biomedical Engineering*

*University of Wisconsin-Madison*

*Madison, U.S.A*

[cstewart7@wisc.edu](mailto:cstewart7@wisc.edu)

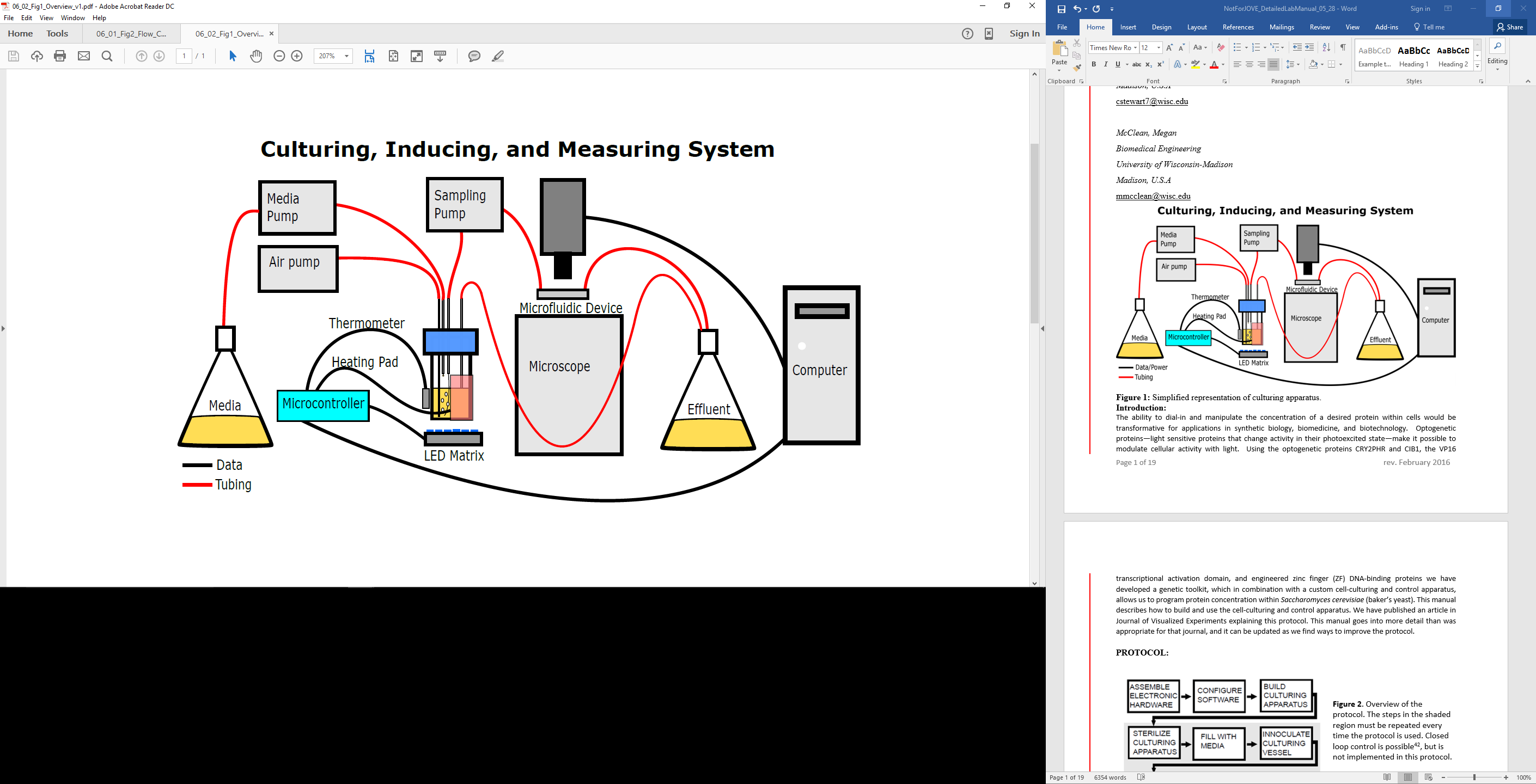
*McClean, Megan*

*Biomedical Engineering*

*University of Wisconsin-Madison*

*Madison, U.S.A*

[mmcclean@wisc.edu](mailto:mmcclean@wisc.edu)

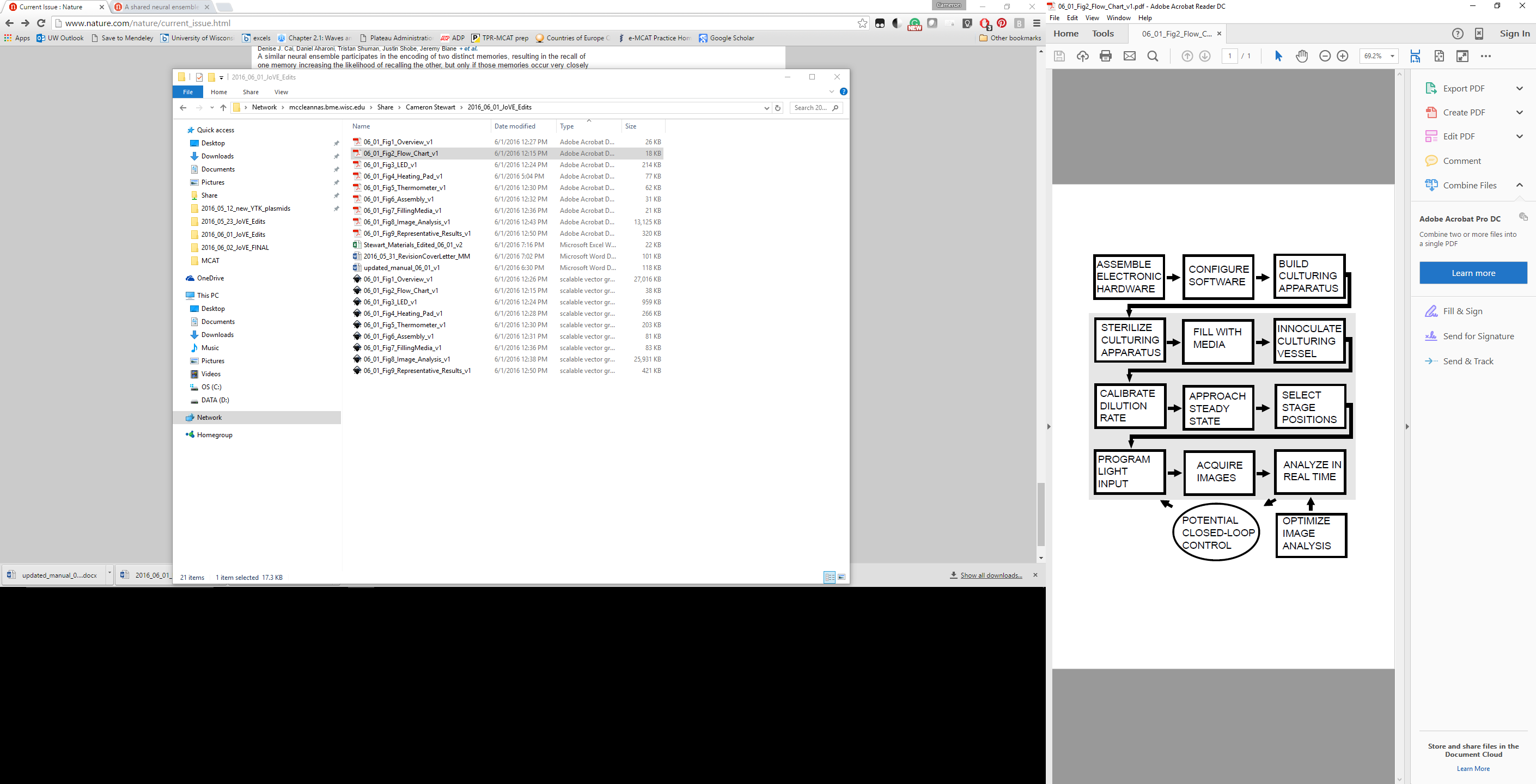


**Figure 1:** Simplified representation of culturing apparatus.

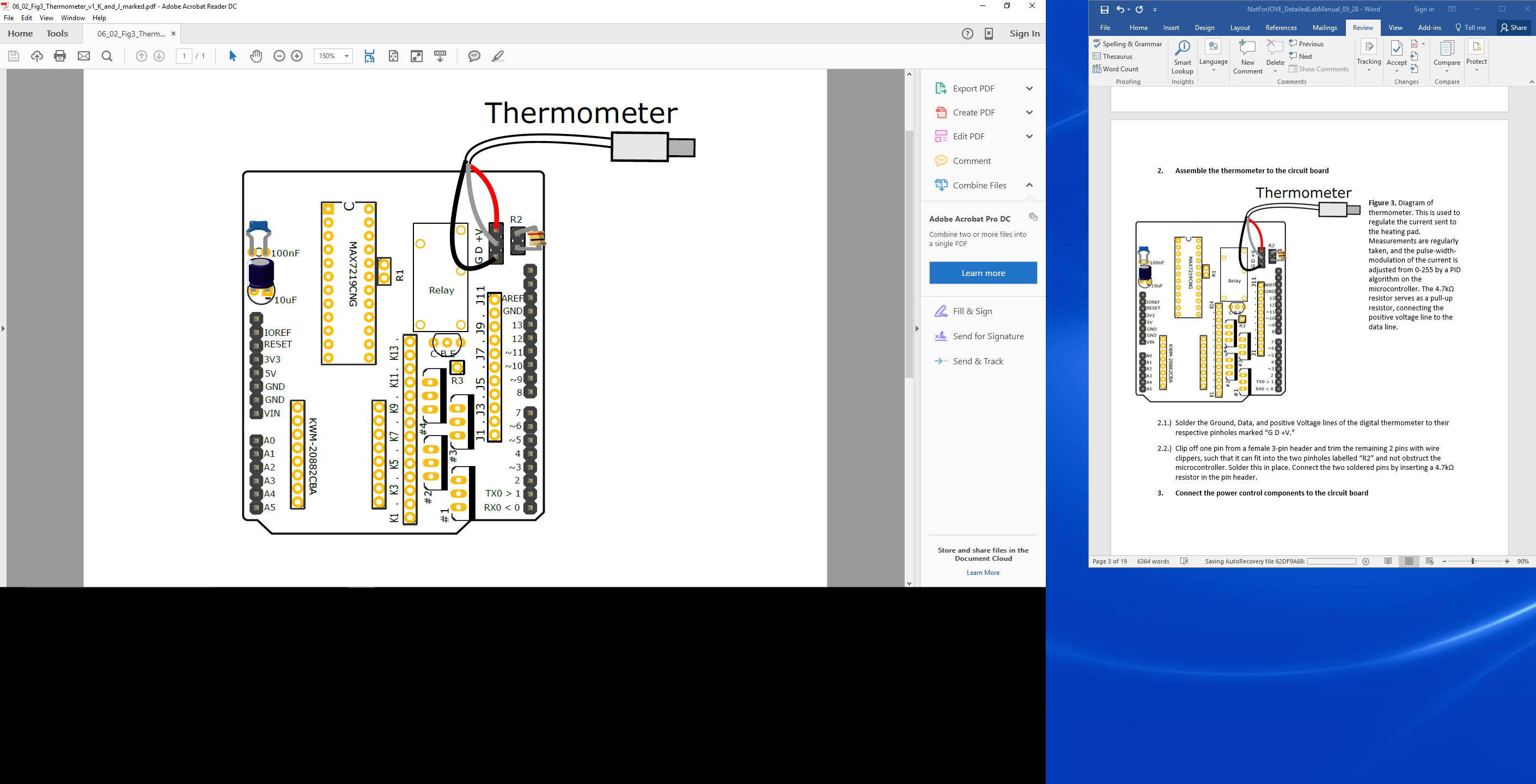
**Introduction:**

The ability to dial-in and manipulate the concentration of a desired protein within cells would be transformative for applications in synthetic biology, biomedicine, and biotechnology. Optogenetic proteins—light sensitive proteins that change activity in their photoexcited state—make it possible to modulate cellular activity with light. Using the optogenetic proteins CRY2PHR and CIB1, the VP16 transcriptional activation domain, and engineered zinc finger (ZF) DNA-binding proteins we have developed a genetic toolkit, which in combination with a custom cell-culturing and control apparatus, allows us to program protein concentration within *Saccharomyces cerevisiae* (baker’s yeast). This manual describes how to build and use the cell-culturing and control apparatus. We have published an article in Journal of Visualized Experiments explaining this protocol. This manual goes into more detail than was appropriate for that journal, and it can be updated as we find ways to improve the protocol.

**PROTOCOL:**



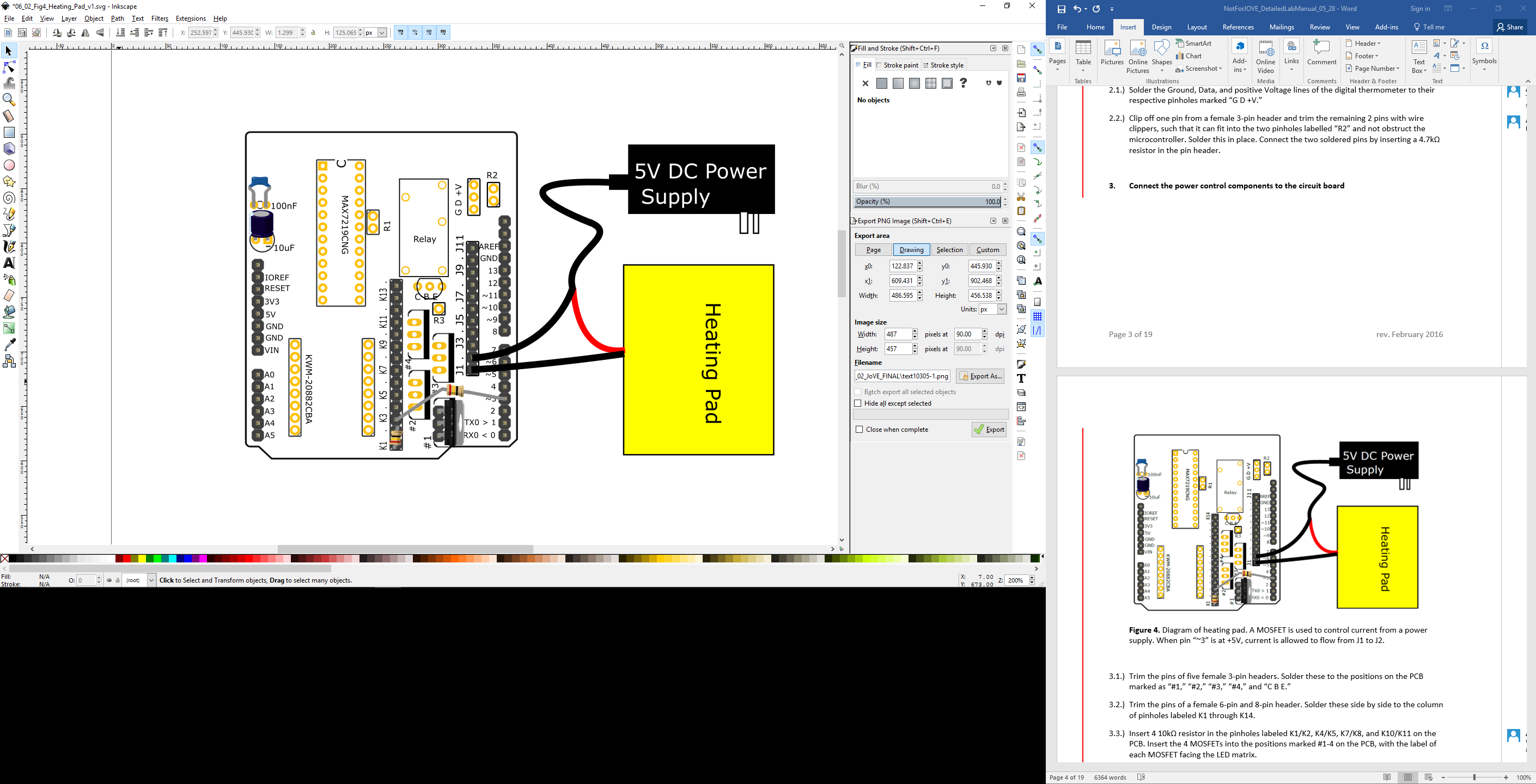
**Figure 2**. Overview of the protocol. The steps in the shaded region must be repeated every time the protocol is used. Closed loop control is possible42, but is not implemented in this protocol.

1. **Download Printed Circuit Board (PCB) design, and order it from Fritzing**
   1. Download the custom microcontroller code, Beanshell scripts, circuit board design, software testing script, and circuit board assembly instructions from the “Optogenetic Chemostat Files” GitHub repository (<https://github.com/McCleanResearch/Optogenetic-Chemostat-Files>).
   2. Order the PCB design described in the “.fzz” file from Fritzing Services. Other vendors may also make the PCB if it is converted to a more general PCB design format File>Export>for Production> Extended Gerber (RS-274X).
2. **Assemble the thermometer to the circuit board**

**Figure 3.** Diagram of thermometer. This is used to regulate the current sent to the heating pad. Measurements are regularly taken, and the pulse-width-modulation of the current is adjusted from 0-255 by a PID algorithm on the microcontroller. The 4.7kΩ resistor serves as a pull-up resistor, connecting the positive voltage line to the data line.

* 1. Solder the Ground, Data, and positive Voltage lines of the digital thermometer to their respective pinholes marked “G D +V.”
  2. Clip off one pin from a female 3-pin header and trim the remaining 2 pins with wire clippers, such that it can fit into the two pinholes labelled “R2” and not obstruct the microcontroller. Solder this in place. Connect the two soldered pins by inserting a 4.7kΩ resistor in the pin header.

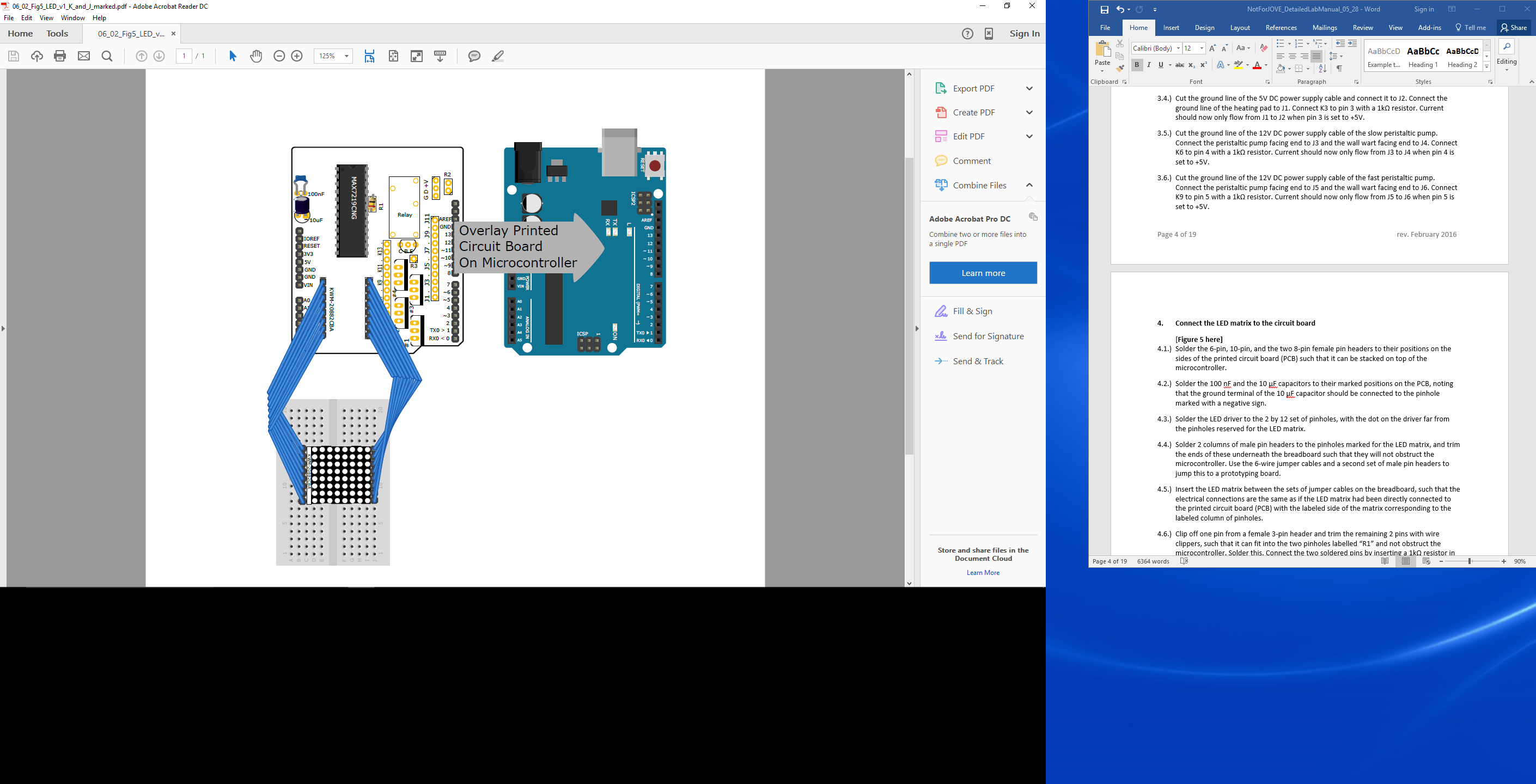
1. **Connect the power control components to the circuit board**



**Figure 4.** Diagram of a controlled heating pad. A MOSFET is used to control current from a power supply. When pin “~3” is at +5V, current is allowed to flow from J1 to J2.

* 1. Trim the pins of five female 3-pin headers. Solder these to the positions on the PCB marked as “#1,” “#2,” “#3,” “#4,” and “C B E.”
  2. Trim the pins of a female 6-pin and 8-pin header. Solder these side by side to the column of pinholes labeled K1 through K14.
  3. Insert 4 10kΩ resistor in the pinholes labeled K1/K2, K4/K5, K7/K8, and K10/K11 on the PCB. Insert the 4 MOSFETs into the positions marked #1-4 on the PCB, with the label of each MOSFET facing the LED matrix.
  4. Cut the ground line of the 5V DC power supply cable and connect it to J2. Connect the ground line of the heating pad to J1. Connect K3 to pin 3 with a 1kΩ resistor. Current should now only flow from J1 to J2 when pin 3 is set to +5V.
  5. Cut the ground line of the 12V DC power supply cable of the slow peristaltic pump. Connect the peristaltic pump facing end to J3 and the wall wart facing end to J4. Connect K6 to pin 4 with a 1kΩ resistor. Current should now only flow from J3 to J4 when pin 4 is set to +5V.
  6. Cut the ground line of the 12V DC power supply cable of the fast peristaltic pump. Connect the peristaltic pump facing end to J5 and the wall wart facing end to J6. Connect K9 to pin 5 with a 1kΩ resistor. Current should now only flow from J5 to J6 when pin 5 is set to +5V.

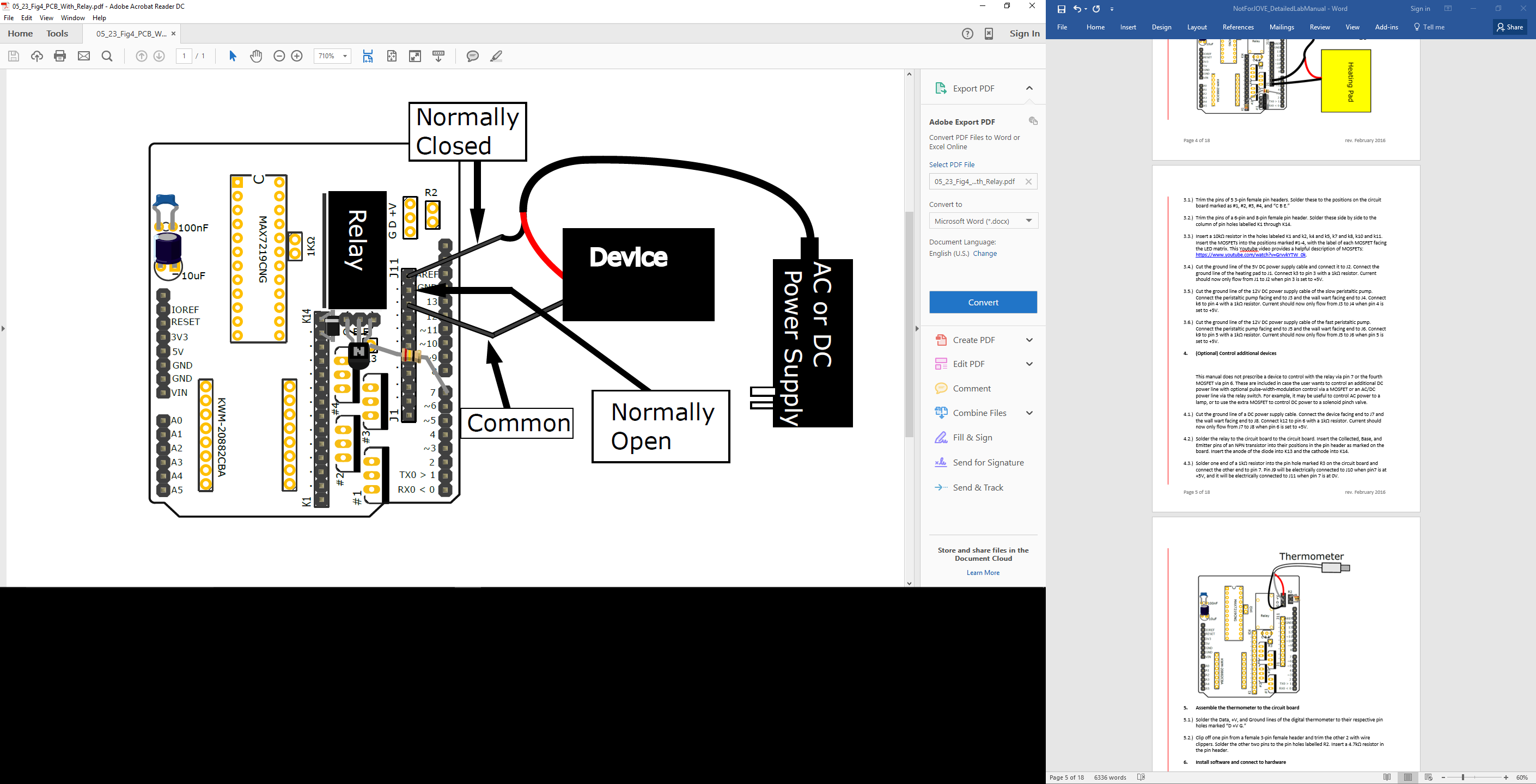
1. **Connect the LED matrix to the circuit board**



**Figure 5.** Diagram of the PCB, microcontroller, and parts that are necessary to control the LED matrix. Note that the components for the thermometer and for controlling DC power to other devices have been removed for clarity. Outputs 2, 12, and 13 from the microcontroller are connected to the DIN, LOAD (CS), and CLOCK pins of the LED driver, respectively. The outputs of the LED driver are connected to the 2 columns of 6 pins marked by “KWM-20882CBA.” The LED matrix would work if it was attached directly to the board, but it is more convenient to have it one a breadboard for this application. The number of LEDs on (0-64) and the pwm current to them (0-255) is controlled by the microcontroller.

* 1. Solder the 6-pin, 10-pin, and the two 8-pin female pin headers to their positions on the sides of the printed circuit board (PCB) such that it can be stacked on top of the microcontroller.
  2. Solder the 100 nF and the 10 μF capacitors to their marked positions on the PCB, noting that the ground terminal of the 10 μF capacitor should be connected to the pinhole marked with a negative sign.
  3. Solder the LED driver to the 2 by 12 set of pinholes, with the dot on the driver far from the pinholes reserved for the LED matrix.
  4. Solder 2 columns of male pin headers to the pinholes marked for the LED matrix, and trim the ends of these underneath the breadboard such that they will not obstruct the microcontroller. Use the 6-wire jumper cables and a second set of male pin headers to jump this to a prototyping board.
  5. Insert the LED matrix between the sets of jumper cables on the breadboard, such that the electrical connections are the same as if the LED matrix had been directly connected to the printed circuit board (PCB) with the labeled side of the matrix corresponding to the labeled column of pinholes.
  6. Clip off one pin from a female 3-pin header and trim the remaining 2 pins with wire clippers, such that it can fit into the two pinholes labelled “R1” and not obstruct the microcontroller. Solder this. Connect the two soldered pins by inserting a 1kΩ resistor in the pin header.

1. **(Optional) Control additional devices**



**Figure 4.** Diagram of the PCB and parts that are necessary to control power to an unspecified device with a relay. This is optional and intended to facilitate the customization of this hardware beyond the uses strictly described in this protocol. Note that other electronic components have been removed for clarity.

This manual does not prescribe a device to control with the relay via pin 7 or the fourth MOSFET via pin 6. These are included in case the user wants to control an additional DC power line with optional pulse-width-modulation control via a MOSFET or an AC/DC power line via the relay switch. For example, it may be useful to control AC power to a lamp, or to use the extra MOSFET to control DC power to a solenoid pinch valve.

* 1. Cut the ground line of a DC power supply cable. Connect the device facing end to J7 and the wall wart facing end to J8. Connect k12 to pin 6 with a 1kΩ resistor. Current should now only flow from J7 to J8 when pin 6 is set to +5V.
  2. Solder the relay to the circuit board to the circuit board. Insert the Collected, Base, and Emitter pins of an NPN transistor into their positions in the pin header as marked on the board. Insert the anode of the diode into K13 and the cathode into K14.
  3. Solder one end of a 1kΩ resistor into the pin hole marked R3 on the circuit board and connect the other end to pin 7. Pin J9 will be electrically connected to J10 when pin7 is at +5V, and it will be electrically connected to J11 when pin 7 is at 0V.

1. **Install the microcontroller software and troubleshoot**
   1. Download the free, open-source Arduino Integrated Development Environment (IDE).
   2. In the IDE, go to File> Open and select the chemostatController.ino file found in the “Arduino\_Chemostat\_Code\_and\_Supporting\_Library” subfolder of the “Optogenetic-Chemostat-Files” folder downloaded from GitHub.
   3. Compile this file. If the compiler cannot access the libraries for the file, go to Sketch>Include libraries and manually link the libraries in the “library” folder in the “Arduino\_Chemostat\_Code\_and\_Supporting\_Library” folder to the chemostatController.ino file.
   4. Stack the PCB on the microcontroller and connect the microcontroller to the computer via an A-B USB cable. Upload the code to the microcontroller.
   5. The simplest way to test the hardware configuration is to hard-code different settings for the devices.

**Note about the PID control algorithm\*** The microcontroller will adjust the pulse-width-modulated current to the heating pad based on the thermometer readings and their relationship to the desired temperature (setpoint). This is performed by a Proportional-Integral-Derivative (PID) algorithm. The setpoint is hard-coded to be 30°C, but can be changed and then recompiled. The weights of the PID constants are much lower than what would be optimal. This means that it will adapt the power output very slowly in response to perturbations. However, if the culturing vessel is not thermodynamically perturbed, then the very low constants are acceptable.

1. **(Optional)- Test that the Microcontroller can receive messages to dynamically control devices**

This step is optional because it is abstracted away so that the user should never need to know how to send messages in this arbitrary format. However, these instructions may be useful for troubleshooting or for improving the microcontroller code.

* 1. In the Arduino IDE, click Tools> Serial Monitor.
  2. Send “immeC,0.25,0.5, 255,e.”
  3. “immeC” is simply the password, enabling the **imme**diate **C**ommand to follow and be accepted by the microscontroller.
  4. The second number indicates what the fraction of every 30 seconds for which power should be sent to pin MEDIA\_PUMP of the microcontroller. By default, this is pin 6. This is done to control the flow rate of the peristaltic pump. In this example, the pump would be ON for 7.5 seconds and OFF for 22.5 seconds, repeatedly.
  5. Similar to the prior number, but it controls power to SAMPLING\_PUMP. By default, this is pin 9.
  6. “e” is the arbitrarily chosen termination character, indicating the **e**nd of a message.
  7. The Arduino will respond with “start<time>,<temperature>,<heating pwm>,<LEDs>,<LED pwm>,e”.
     1. <Time> is in milliseconds since it started or received time course instructions>.
     2. <Temperature> is the temperature sensed by the digital thermometer in Celsius.
     3. <heating pwm> is the pulse-width-modulated of power to the heating pad on a scale of 0-255>.
     4. <LEDs> is the number of LEDs that should be on from 0-64.
     5. <LED pwm> is the pulse-width-modulated brightness of those LEDs on a scale of 0-15 LEDs.
  8. If the Arduino is not connected to anything, the returned values will be meaningless.

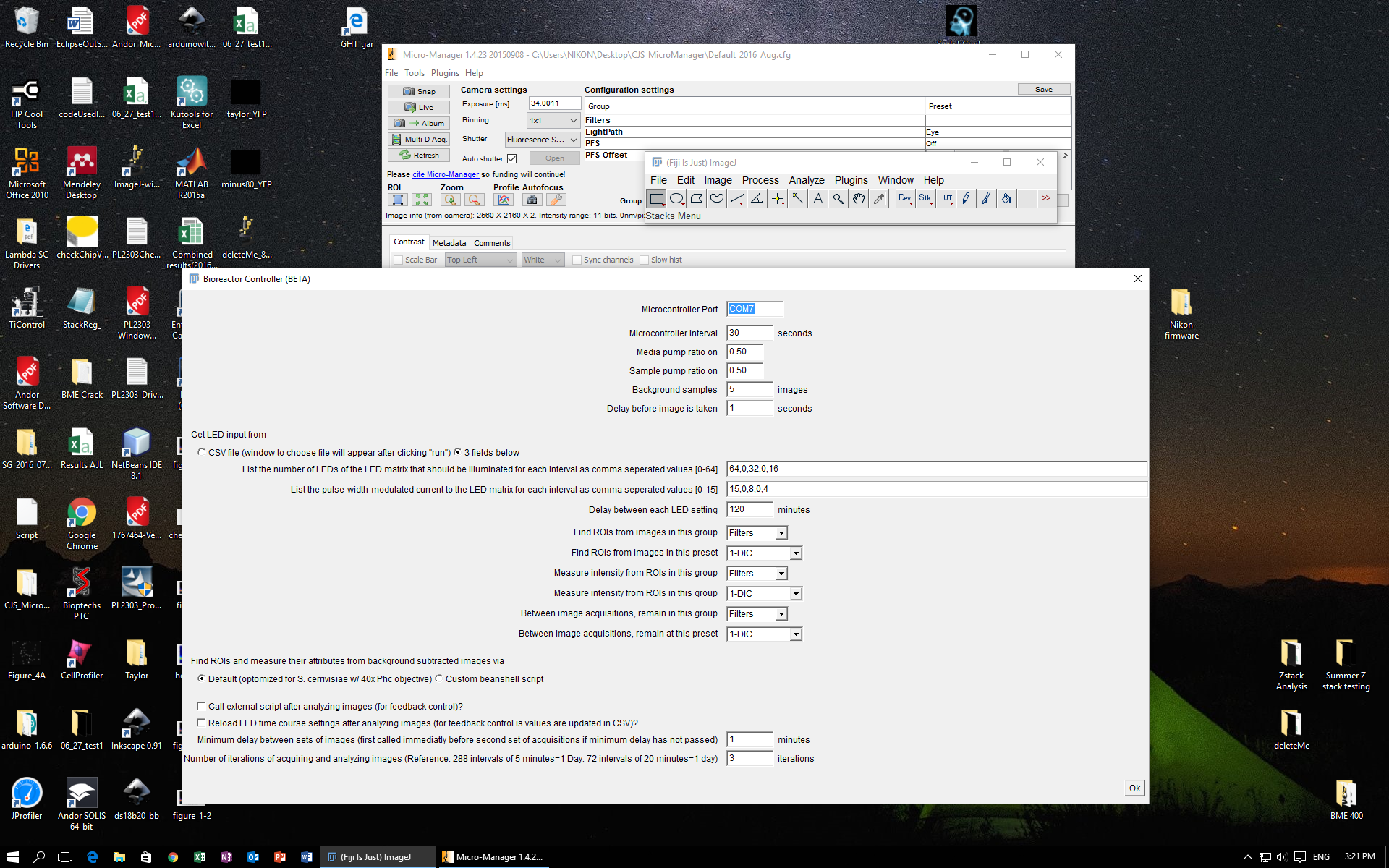
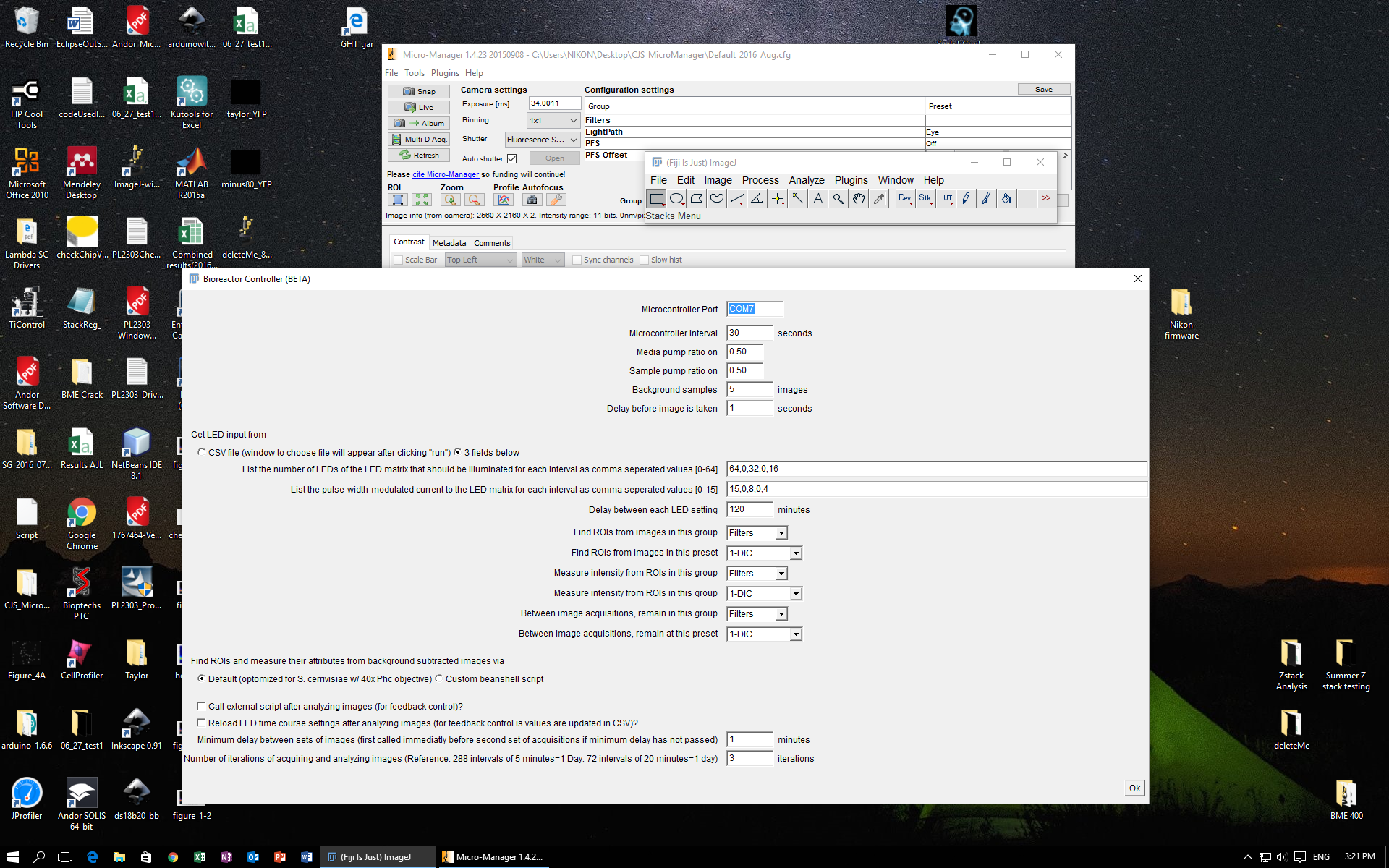
1. **(Optional) – Test that the microcontroller can receive messages to dynamically adjust the LED settings**

This step is optional for the same reasons as are mentioned above.

* 1. In the Arduino IDE, click Tools> Serial Monitor.
  2. Send this arbitrary command: “timeC,0,5,1,8,e,timeC,1,10,5,8,e,timeC,2,45,5,8”.
  3. The microcontroller should turn on 1 LED with a pwm brightness of 8 after 5 seconds. Then it should turn on 5 LEDs with pwm brightness of 8 after 10 seconds. After 45 seconds, it should go dark.
  4. “timeC” starts every command because it sets the time-Course for the LED matrix to follow
  5. The microcontroller saves these instructions in an array, and then continually checks if it is time to follow the next instruction. The first number in the message is the Row that the instructions should correspond to in that array.
  6. At 45 seconds the matrix will go dark because, immediately after turning on 5 LEDs with pwm brightness of 8, it will see that the following instruction in row 3 says to have 0 LEDs with pwm brightness of 0 after 0 seconds. Therefore, it is best for the last row to be a time far in the future, if the lights should stay on. Similarly, the first “timeC” message should specify a time far enough in the future for the second message to have already arrived, for the same reason.
  7. Note that the microcontroller processes these messages slowly (a few “timeC” instructions per second), and that the serial buffer is only 64 characters long. Thus, to send a longer chain of commands, each “timeC” message should be delayed by a second.

1. **Install software to control microscope and analyze images**
   1. Download and install FIJI31.
   2. Download and configure Micro-Manager29,30 as a FIJI plugin31.
      1. From the directory into which micromanager was downloaded, copy all of the “libmmgr\_dal” files and paste them into the Fiji.app directory. Similarly, copy the “mmplugins” directory and the “mmautofocus” directory from the micromanager directory to the Fiji.app directory.
      2. Copy the “plugins/Micro-Manager” directory from the micromanager main directory to the “plugins” directory in the Fiji.app directory.
   3. Ensure that the Java Virtual Machine (JVM) has access to most of the computer’s available memory. For Windows 7+, go to Control Panel>Programs>Java>Java and enter “–Xms<RAM HERE>m”, where <RAM HERE> is the maximum number of megabytes that the JVM can use. Note that 1024 MB are equivalent to 1 GB.
   4. Go to FIJI>Edit>Options>Memory & Threads. Ensure that FIJI has access to most of the computer’s available RAM. Set the number of parallel threads equal to the number of logical processors on your computer.
   5. From FIJI click Plugins>Micro-Manager> Micro-Manager Studio.
   6. A pop-up may ask which configuration option you would like to use. Select “none” until you have created a configuration file.
   7. Go to Tools> Hardware Configuration Wizard. Follow the instructions to enable micromanager to communicate to all of the devices which are used to control the microscope. Include a “FreeSerialPort” device that can be used to communicate to the microcontroller.
      1. Go to the “Devices” tab of the micromanager website for more detailed instructions.
   8. Create a configuration group by clicking the “+” icon next to “Group:.” Give it a name and select the devices which will form the set of devices that need to be controlled by preset settings.
   9. Create a configuration preset by clicking the “+” icon next to “Preset:.” Each configuration preset consists of a list of states in which the set of devices in the configuration group should be. These presets are useful for defining the channel states for multichannel image acquisition.
   10. Test the configuration.
       * 1. Click the “Snap” icon to test that micromanager can take a picture.
         2. Go to Tools>Stage Position List. Mark a few positions and test that micromanager can move the microscope stage to each position.
2. **Install the custom code for this protocol**

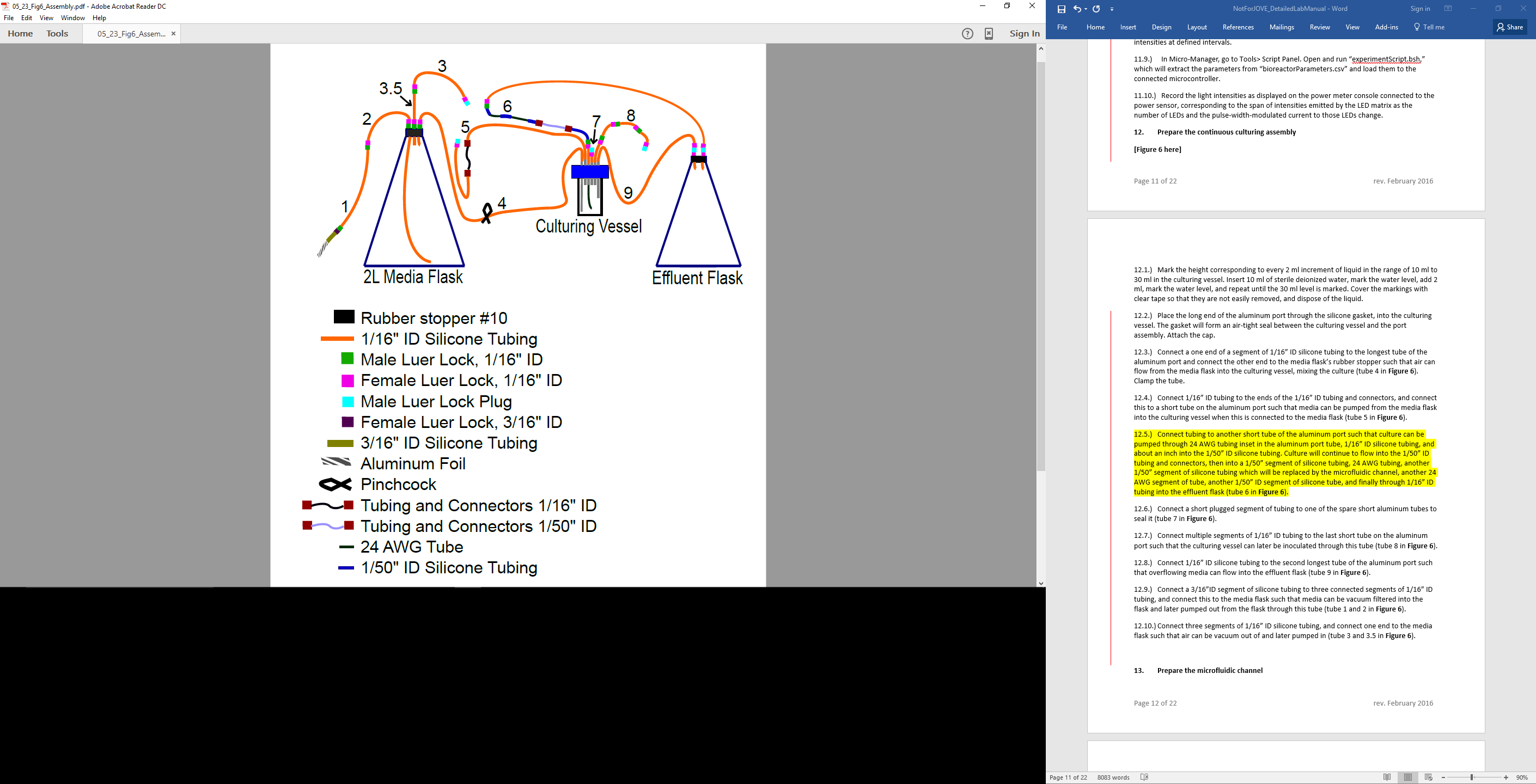
**Overview of custom code:**



When an experiment is about to begin, the script searches for the “bioreactorParameters.csv” and the “bioreactorTimecourse.csv” files in the current working directory. The latter file contains the instructions for what the LED matrix settings should be at any given time. The former contains a list of other variables, described in the folder, that are intended to accommodate a wide variety of scenarios. In theory, a user will only need to change the parameters in these files. The main exception would be to edit the image analysis routine used in the script. After values are extracted from these files, the loadTimeCourse() method is called which sends the time course data to the microcontroller, how many LEDs should be on and with what pwm intensity at each time. This method is slow because the microcontroller is slow to read messages from its message buffer. For this reason, delays are incorporated to send the messages slowly and query the microcontroller to verify that they were recieved. Data from an experiment is then collected through a continuous iteration of the following instructions. The computer sends a message to the Arduino telling to stop the sampling pump. Yeast settle to the bottom of the sampling pump. The acquireImages() method is called, which makes the microscope take a fluorescent and non-fluorescent image is take at every stage position specified in the stagePositionList. The sampling pump is turned on. The generateBG() method is called which makes the microscope take an image at each stage position is taken either three, five, or seven times, as specified by the bgSamples variable. For each pixel location, the median pixel intensity of the acquired set of images is used to create a composite image. The composite image is an image of the PDMS channel without any flowing yeast, because it is assumed that a for the majority of the images acquired there was not a yeast at that location. This assumption is most valid when the cell density is low and the value of bgSamples is high. The getData() method is called which analyzes those images. This method calls the backgroundArithmetic() method which subtracts the background images from the initially acquired images, so that the yeast from the initially acquired images are all that is left. Inside the getData() method, the alignImages() method is then called which translates the fluorescent image to ensure that it most closely matches the non-fluorescent image. This is especially useful when the yeast move slightly between the acquisition of the two types of images. Next, getData() calls the processAndThreshold() method. This method finds the regions of interest in the non-fluorescent image which correspond to cells, as described in step 19. Next, the getData() method calls the analyzeROIs() method, which maps each of the regions of interest from the non-fluorescent image to the fluorescent image, and then saves data about the pixels within the ROIs to the Results Table, a table that can be automatically generated when FIJI’s Analyze Particles function is called. Finally, getData() calls the saveResults() method. Data summary data from the results table is saved in the summary comma separated value file. The image analysis routine is optimized to find *S. cerevisiae* in a phase image acquired under Köhler illumination with a Plan Apo λ 40x/0.95 OFN25 Ph2 DM Nikon objective.

* 1. Download the “bioreactorController.jar” file into the “mmplugins,” directory.
  2. From Micromanager, go to “Plugins> Bioreactor Controller,” click the “Open” icon, and load the “ExperimentScript.bsh”

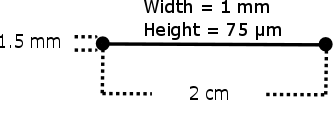
1. **Make and characterize the light-proof enclosure for the bioreactor**
   1. Stack the 8 pin IC sockets as building blocks on the breadboard adjacent to the LED matrix such that the culturing vessel can rest on them above the matrix.
   2. Fasten a portion of diffusion paper under the top layer of the sockets, such that the light striking the culturing vessel from the LED matrix is diffuse.
   3. Add layers of black foam with insets cut out until the black foam lies even with and surrounds the diffusion paper over the LED matrix. Cut out two 6-inch by 6-inch or larger rectangles of black foam and cut an inner rectangle to surround the prototyping board in each, then stack them. Add a similar layer, with the inner rectangle cut to surround the LED and diffusion paper, about 1-inch by 1-inch.
   4. Cut out the foam pads and tape the ends together to make a hollow spiral column of black foam that the culturing vessel can fit into with room for tubes and wires.
   5. Center the hollow column of black foam over the LED matrix, and mark the boundary of the column. This boundary will mark where it should be centered in the future.
   6. Cut out a portion of foam that can be taped to the top of the column so that all external light is blocked out.
   7. Attach the photodiode power sensor to the culturing vessel’s cap with tape, attach the cap, and insert the culturing vessel in the enclosure.
   8. In “bioreactorParameters.csv” set the “controlMicroscope” variable to 0 and set the LED matrix time-course variables such that the matrix will incrementally span a range of intensities at defined intervals.
   9. In Micro-Manager, go to Tools> Script Panel. Open and run “experimentScript.bsh,” which will extract the parameters from “bioreactorParameters.csv” and load them to the connected microcontroller.
   10. Record the light intensities as displayed on the power meter console connected to the power sensor, corresponding to the span of intensities emitted by the LED matrix as the number of LEDs and the pulse-width-modulated current to those LEDs change.
2. **Prepare the continuous culturing assembly**



**Figure 6**. Diagram of continuous culturing vessels and tubing prior to being autoclaved. Label 1 as "Inoculation port." Label 2 as "Overflow effluent." Label 3 as "Sampled effluent." Label 4 as "Replace with microfluidic." Label 5 as "Media into culturing vessel." Label 6 as "1) Attach to vacuum, 2) Clamp shut." Label 7 as "1) Media in from filter." Label 8 as "2) Media out, to culturing vessel."

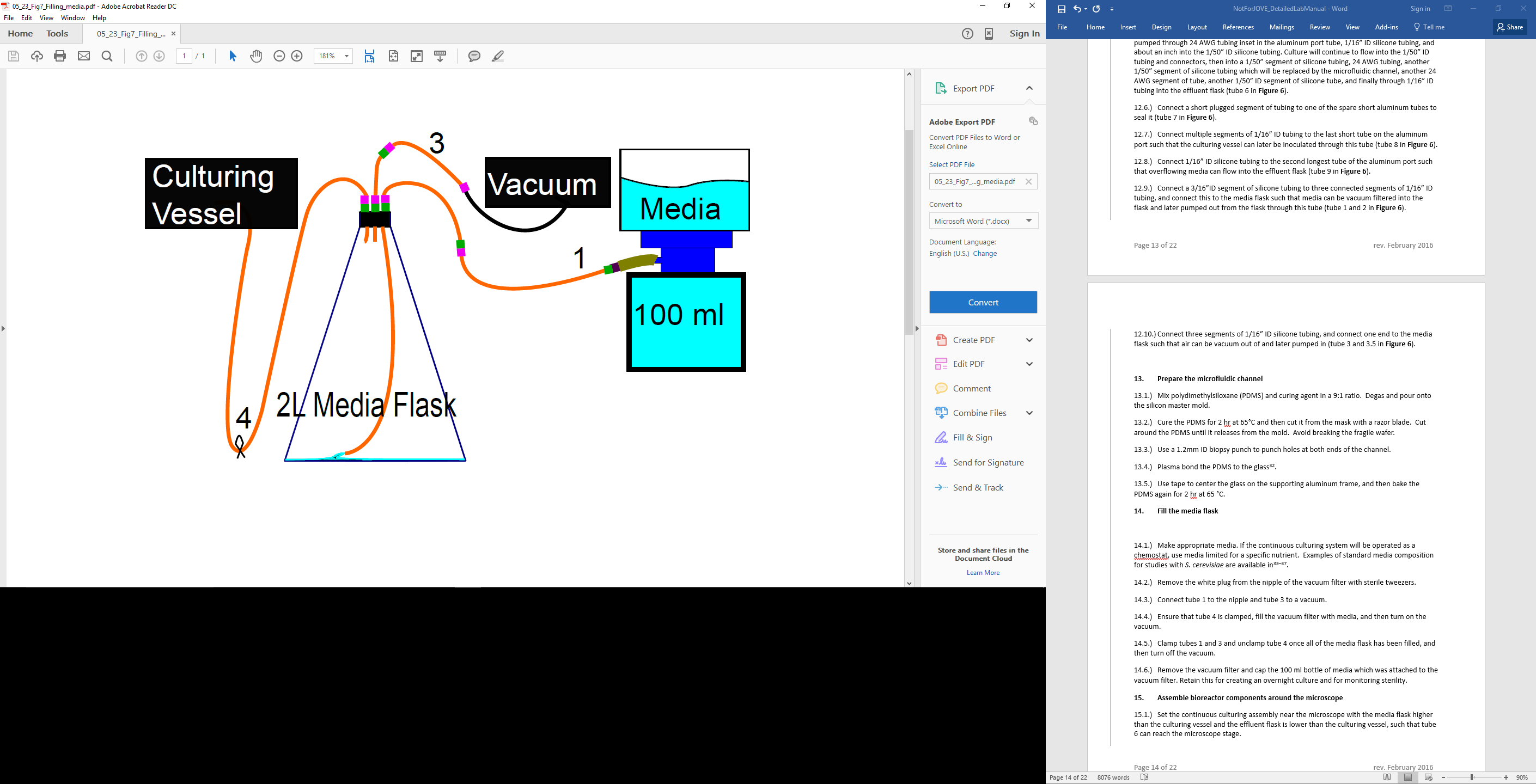
* 1. Mark the height corresponding to every 2 ml increment of liquid in the range of 10 ml to 30 ml in the culturing vessel. Insert 10 ml of sterile deionized water, mark the water level, add 2 ml, mark the water level, and repeat until the 30 ml level is marked. Cover the markings with clear tape so that they are not easily removed, and dispose of the liquid.
  2. Place the long end of the aluminum port through the silicone gasket, into the culturing vessel. The gasket will form an air-tight seal between the culturing vessel and the port assembly. Attach the cap.
  3. Connect a one end of a segment of 1/16” ID silicone tubing to the longest tube of the aluminum port and connect the other end to the media flask’s rubber stopper such that air can flow from the media flask into the culturing vessel, mixing the culture (tube 4 in **Figure 6**). Clamp the tube.
  4. Connect 1/16” ID tubing to the ends of the 1/16” ID tubing and connectors, and connect this to a short tube on the aluminum port such that media can be pumped from the media flask into the culturing vessel when this is connected to the media flask (tube 5 in **Figure 6**).
  5. Connect tubing to another short tube of the aluminum port such that culture can be pumped through 24 AWG tubing inset in the aluminum port tube, 1/16” ID silicone tubing, and about an inch into the 1/50” ID silicone tubing. Culture will continue to flow into the 1/50” ID tubing and connectors, then into a 1/50” segment of silicone tubing, 24 AWG tubing, another 1/50” segment of silicone tubing which will be replaced by the microfluidic channel, another 24 AWG segment of tube, another 1/50” ID segment of silicone tube, and finally through 1/16” ID tubing into the effluent flask (tube 6 in **Figure 6**).
  6. Connect a short plugged segment of tubing to one of the spare short aluminum tubes to seal it (tube 7 in **Figure 6**).
  7. Connect multiple segments of 1/16” ID tubing to the last short tube on the aluminum port such that the culturing vessel can later be inoculated through this tube (tube 8 in **Figure 6**).
  8. Connect 1/16” ID silicone tubing to the second longest tube of the aluminum port such that overflowing media can flow into the effluent flask (tube 9 in **Figure 6**).
  9. Connect a 3/16”ID segment of silicone tubing to three connected segments of 1/16” ID tubing, and connect this to the media flask such that media can be vacuum filtered into the flask and later pumped out from the flask through this tube (tube 1 and 2 in **Figure 6**).
  10. Connect three segments of 1/16” ID silicone tubing, and connect one end to the media flask such that air can be vacuum out of and later pumped in (tube 3 and 3.5 in **Figure 6**).

1. **Prepare the microfluidic channel**
   1. Create a mold for the Mix polydimethylsiloxane (PDMS) channel with a width of 500 µm, a height of 75 µm, and a length of several centimeters. The CAD file specifying the dimensions of the transparency mask used to make a 5cm long channel can be found in the “CAD\_for\_PDMS” subfolder of the “Optogenetic-Chemostat-Files” folder downloaded from the GitHub repository. The mold can be made with the transparency mask using standard photolithography techniques and SU8-50 photoresist.
   2. Pour PDMS onto the mold using a 1: 9 curing agent to monomer ratio.
   3. Cure the PDMS for 2 hours, or overnight, at 650C and then cut it from the mask with a razor blade. Be careful not to break the fragile mold.
   4. Use a 1.2mm ID hole puncher to punch holes at both ends of the channel. Press the hole puncher against the channel side, and insert it through to the other end.
   5. Plasma bond the PDMS to the glass32. This step is important because incorrect plasma bonding can cause the PDMS to delaminate from the glass. The McClean lab’s plasma bonding instructions are available on openwetware (<http://openwetware.org/wiki/McClean:_Flow_Cells>.)
   6. Tape the glass to the aluminum frame, and then bake the PDMS again for 2 hours or overnight at 650C.



**Diagram of microfluidic channel.** These are the dimensions of the microfluidic channel that we used to collect representative results.

1. **Autoclave the continuous culturing system**
   1. Wrap the top of the media flask with a roughly 4” by 10” rectangle of aluminum foil. In the autoclave, the rubber stopper must be on the flask sideways so that it does not form an air-tight seal. The curtain of aluminum foil will make it easy to press the cap down into the media flask after it has been autoclaved without contaminating the flask, and will remain useful as a precautionary layer to keep contaminants away from the lid of the media flask.
   2. Wrap a similar curtain around the top of the bioreactor’s lid. The lid should be on loosely before it is autoclaved. The curtain will make it easy to tighten the lid without contaminating the bioreactor, and will remain useful as an additional precautionary layer to keep contaminates away from the lid of the bioreactor.
   3. Tape the end of the tube 5 in figure 6 onto the media flask. It will be connected to the media flask’s tube after the media flask has been filled with media, and taping it here keeps it where it will need to be.
   4. Attach autoclave tape to one of the containers, and autoclave them according to your laboratories autoclave procedure. The chemostat assembly has not been tested to see if it can withstand temperatures above 2510F, so do not autoclave above this temperature unless necessary.
   5. After the autoclave cycle has finished, ensure that the autoclave tape indicates that the autoclave indeed sterilized the containers.
   6. Wearing gloves to protect your hands, press down on the media flasks cap so that it forms an airtight seal with the media flask. Screw down the cap on the bioreactor so that it also forms an airtight seal.
   7. Move the chemostat to a clean work area where it can be filled with media.
2. **Prepare phosphate limited media (or use media with a different limitation)**
   1. Make 1 L of stock 1000x vitamin solution: mix together 2 mg of biotin, 400 mg of D-calcium pantothenate, 2 mg of folic acid, 2000 mg of inositol, 400 mg of nicotinic acid, 200mg of 4-aminobenzoic acid, 400mg of pyridoxine hydrochloride, 20 0mg of (-)-riboflavin, and 400 mg of thiamine hydrochloride. Bring this solution up to 1 L with DI water, stir the solution until the vitamins dissolve, and then vacuum filter.
      1. Aliquot the sterile solution into 50 mL containers and store them in a -20 0C freezer. The solution will remain viable in this condition for years.
      2. Keep a working solution in a dark 4 0C fridge. The working solution can last for a few weeks. Shake the container until the vitamins are well mixed before withdrawing from the solution.
   2. Make 1 L of stock 1000x mineral solution: mix together 500 mg of boric acid, 40 mg of copper sulfate pentahydrate, 100 mg of potassium iodide, 200 mg of iron(III) chloride hexahydrate, 400 mg of manganese(II) sulfate monohydrate, 200 mg of sodium molybdate dehydrate, and 400 mg of zinc sulfate heptahydrate. Bring this solution up to 1 L with DI water, stir the solution until the minerals dissolve, and then vacuum filter or autoclave it.
      1. Store at room temperature in the dark. This solution can last for over a year.
      2. Mix the container well to ensure that the minerals are fully dissolved before withdrawing from it.
   3. Make 1 L of stock 10x salt solution: mix together 2g of calcium chloride dehydrate, 2g of sodium chloride, 5g of anhydrous magnesium sulfate, 100g of ammonium sulfate, and 20g of potassium chloride. Bring this solution up to 1 L with DI water, stir the solution until the salts dissolve, and then vacuum filter or autoclave.
      1. This solution can be stored at room temperature for over a year.
   4. Make 1 L of stock 10x phosphate solution: fill a container with 1g of potassium phosphate dibasic and bring this solution up to 1 L with DI water. Stir the solution until the phosphate dissolves and then vacuum filter or autoclave it.
      1. This solution can be stored at room temperature for over a year.
   5. Make 1 L of stock 20x dextrose solution: fill a container with 400g of dextrose and bring the total volume to 1 L with DI water. Vacuum filter it.
      1. This can be stored at room temperature and will last several months.
   6. Make 2 L of the phosphate limited media: mix together 2mL of the 1000x vitamins solution, 2 mL of the 1000x minerals solution, 20 mL of the 10x phosphate solution, 200mL of the 10x other salts solution, 100mL of the 20x dextrose solution, 100 mL of the 20x amino acids solution, and 1576 mL of sterile DI water. Vacuum filter the media. Note that only 20 mL of the 10x phosphate solution are added so that the media is phosphate limited.
      1. This can be stored at room temperature and will last several months.
3. **Fill the media flask**



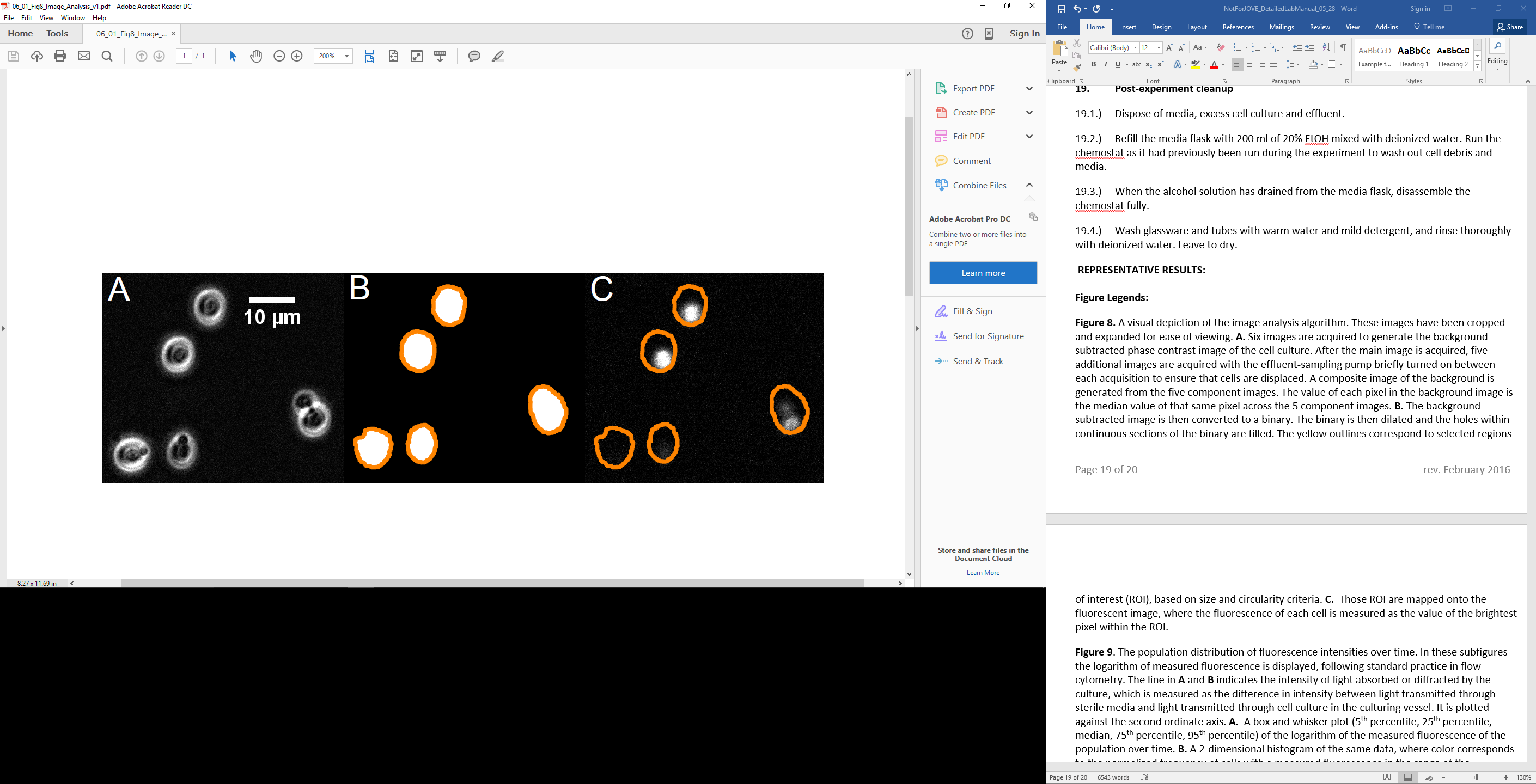
**Figure 7.** Diagram of how to vacuum filter media into the media flask. With the cap and inner plug removed from the nipple of the vacuum filter, media is sucked through the nipple to the media flask. Once filled, connect the tube media flask’s tube labeled “2) Media out, to culturing vessel” to the culturing vessel’s tube labeled “Media into culturing vessel.”

* 1. Make appropriate media. If the continuous culturing system will be operated as a chemostat, use media limited for a specific nutrient. Our labs phosphate limited media recipe, borrowed from the Dunham lab, is at the end of this manual.
  2. Remove the white plug from the nipple of the vacuum filter with sterile tweezers.
  3. Connect tube 1 to the nipple and tube 3 to a vacuum.
  4. Ensure that tube 4 is clamped, fill the vacuum filter with media, and then turn on the vacuum.
  5. Clamp tubes 1 and 3 and unclamp tube 4 once all of the media flask has been filled, and then turn off the vacuum.
  6. Remove the vacuum filter and cap the 100 ml bottle of media which was attached to the vacuum filter. Retain this for creating an overnight culture and for monitoring sterility.

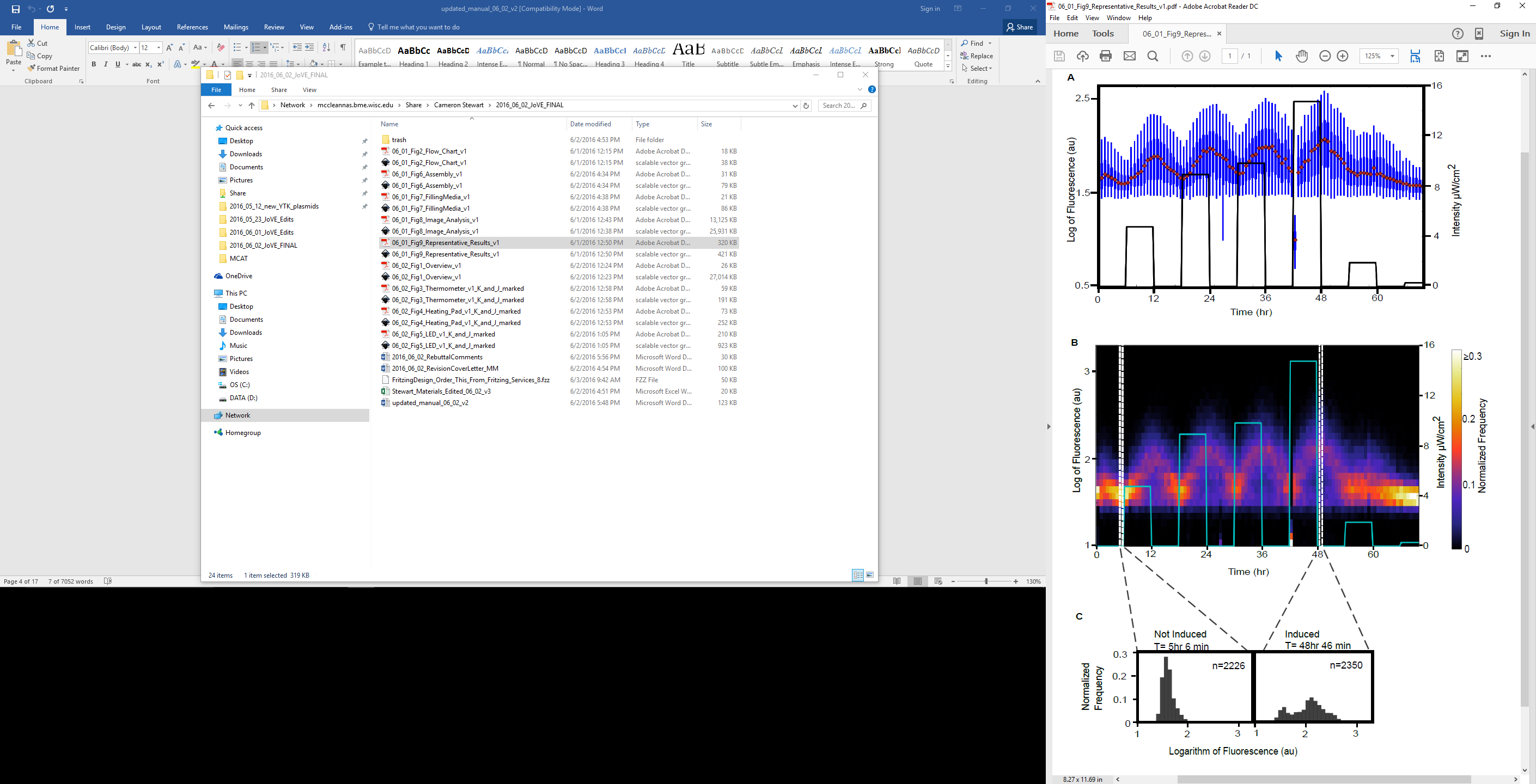
1. **Assemble bioreactor components around the microscope**
   1. Set the continuous culturing assembly near the microscope with the media flask higher than the culturing vessel and the effluent flask is lower than the culturing vessel, such that tube 6 can reach the microscope stage.
   2. Tape the heating pad to the base of the bioreactor. Tape the thermometer to the base of the bioreactor.
   3. Coil the media pump around the culturing vessel so that the entering media will be at the same temperature as the vessel.
   4. Insert the bioreactor into the black foam enclosure over the LED matrix, and ensure that none of the tubes are pinched. Cover the enclosure with a foam lid so that no light enters.
   5. Securely tape the lid of the media flask and effluent container down.
   6. Connect the white end of the syringe air filter to the aquarium pump and connect the blue end to the media flask’s tube labeled “2) Air into media flask. Blue side here.”
   7. Clamp the air tube between the media flask and the bioreactor and turn on the aquarium pump to force media into the culturing vessel. Unclamp the air tube when the culturing vessel is full.
   8. Connect the peristaltic pump tube segments to their corresponding peristaltic pumps.
   9. Unplug the ends of the 24 AWG tubing from the 1/50” ID tube labeled “replace w/ channel” and plug these ends into the inlet and outlet of the microfluidic device.
   10. Disconnect the effluent tubes from the effluent flask and collect effluent in test tubes chilled on ice. Calculate the flow rate of media into culturing vessel from the volume collected in a test tube and the time to collect it. Measure the optical density of the chilled effluent, which indicates the average population density during the period that the sample of effluent was collected.
   11. Ensure that the media in the media flask looks the same as it did when it was first made. If it is cloudy, then it has been contaminated.
   12. Turn off the pumps. Inoculate the culturing vessel with 2 ml of culture through the inoculation port. Let it grow overnight and then begin diluting the culture the next morning.
   13. With the value of the “controlMicroscope” at zero, iteratively adjust the parameters in the “bioreactorParameters.csv” file until the flow rate from the sampling pump is just lower than the flow rate of media into the culturing vessel, calculated as the sum of the sampling flow rate and the effluent flow rate. This iterative calibration process can take about a day.
2. **Program light pulses and data collection**
   1. In Micromanager, go to Tools> Stage Position List and select a set of stage positions where images will be acquired.
   2. Open and run the “bioreactor controller” plugin to begin the experiment.
   3. When finished, open the “microcontrollerRecords.csv” file to review the temperature and LED matrix status over the course of the experiment, the “Summary.csv” file to review the summary data from each set of images and the “Results<n>.csv” files to review data summarizing each ROI from each time period, where n is the nth data set.
3. **Post-experiment cleanup**
   1. Dispose of media, excess cell culture and effluent.
   2. Refill the media flask with 200 ml of 20% EtOH mixed with deionized water. Run the chemostat as it had previously been run during the experiment to wash out cell debris and media.
   3. When the alcohol solution has drained from the media flask, disassemble the chemostat fully.
   4. Wash glassware and tubes with warm water and mild detergent, and rinse thoroughly with deionized water. Leave to dry.

**REPRESENTATIVE RESULTS:**

**Figure Legends:**



**Figure 8.** A visual depiction of the image analysis algorithm. These images have been cropped and expanded for ease of viewing. **A.** Six images are acquired to generate the background-subtracted phase contrast image of the cell culture. After the main image is acquired, five additional images are acquired with the effluent-sampling pump briefly turned on between each acquisition to ensure that cells are displaced. A composite image of the background is generated from the five component images. The value of each pixel in the background image is the median value of that same pixel across the 5 component images. **B.** The background-subtracted image is then converted to a binary. The binary is then dilated and the holes within continuous sections of the binary are filled. The yellow outlines correspond to selected regions of interest (ROI), based on size and circularity criteria. **C.** Those ROI are mapped onto the fluorescent image, where the fluorescence of each cell is measured as the value of the brightest pixel within the ROI.



**Figure 9**. The population distribution of fluorescence intensities over time. In these subfigures the logarithm of measured fluorescence is displayed, following standard practice in flow cytometry. The line in **A** and **B** indicates the intensity of light absorbed or diffracted by the culture, which is measured as the difference in intensity between light transmitted through sterile media and light transmitted through cell culture in the culturing vessel. It is plotted against the second ordinate axis. **A.**  A box and whisker plot (5th percentile, 25th percentile, median, 75th percentile, 95th percentile) of the logarithm of the measured fluorescence of the population over time. **B.** A 2-dimensional histogram of the same data, where color corresponds to the normalized frequency of cells with a measured fluorescence in the range of the corresponding bin. The colors were scaled to the range of data without the outlier at 42 hr 26 min included. The normalized frequency of the outlier in the lowest fluorescence bin is 0.7. **C.** One dimensional histograms of the logarithm of the measured fluorescence of the population before the first recorded exposure to light and after the greatest exposure to light. It demonstrates that the light-induced expression of YFP in this strain is bimodal.

**Note\*** In the GitHub repository, we have included a python script and our sample data so that others can recreate **figure 9**.

**ACKNOWLEDGMENTS:**

We would like to acknowledge Molly Lazar for assistance in testing the protocol, Kieran Sweeney for helpful discussions and editing, and Taylor Scott, My An-adirekkun, and Stephanie Geller for critical reading of the manuscript. Megan Nicole McClean, Ph.D. holds a Career Award at the Scientific Interface from the Burroughs Wellcome Fund.