

# Utilizing social media and video games to control #DIY microscopes

Maxime Leblanc-Latour<sup>1</sup>, Craig Bryan<sup>1</sup> and Andrew E. Pelling<sup>1,2,3,4</sup>

<sup>1</sup>Department of Physics, University of Ottawa, Ottawa, Ontario, Canada

<sup>2</sup>Department of Biology, University of Ottawa, Ottawa, Ontario, Canada

<sup>3</sup>Institute for Science, Society and Policy, University of Ottawa, Ottawa, Ontario, Canada

<sup>4</sup>SymbioticA and the School of Anatomy, Physiology and Human Biology, University of Western Australia, Perth, Western Australia, Australia

## ABSTRACT

Open-source lab equipment is becoming more widespread with the popularization of fabrication tools such as 3D printers, laser cutters, CNC machines, open source microcontrollers and open source software. Although many pieces of common laboratory equipment have been developed, software control of these items is sometimes lacking. Specifically, control software that can be easily implemented and enable user-input and control over multiple platforms (PC, smartphone, web, etc.). The aim of this proof-of principle study was to develop and implement software for the control of a low-cost, 3D printed microscope. Here, we present two approaches which enable microscope control by exploiting the functionality of the social media platform Twitter or player actions inside of the videogame Minecraft. The microscope was constructed from a modified web-camera and implemented on a Raspberry Pi computer. Three aspects of microscope control were tested, including single image capture, focus control and time-lapse imaging. The Twitter embodiment enabled users to send ‘tweets’ directly to the microscope. Image data acquired by the microscope was then returned to the user through a Twitter reply and stored permanently on the photo-sharing platform Flickr, along with any relevant metadata. Local control of the microscope was also implemented by utilizing the video game Minecraft, in situations where Internet connectivity is not present or stable. A virtual laboratory was constructed inside the Minecraft world and player actions inside the laboratory were linked to specific microscope functions. Here, we present the methodology and results of these experiments and discuss possible limitations and future extensions of this work.

Submitted 28 February 2017

Accepted 9 November 2017

Published 11 December 2017

Corresponding author

Andrew E. Pelling, a@pellinglab.net,  
apelling@uottawa.ca

Academic editor

Feng Xia

Additional Information and  
Declarations can be found on  
page 14

DOI 10.7717/peerj-cs.139

© Copyright

2017 Leblanc-Latour et al.

Distributed under

Creative Commons CC-BY 4.0

OPEN ACCESS

**Subjects** Network Science and Online Social Networks, Social Computing

**Keywords** Microscope, Do-It-Yourself, Open source, Raspberry Pi, Twitter, Flickr

## INTRODUCTION

The general interest in using and developing low cost, open source labware is gaining considerable traction in garages, academic labs and commercial spaces (*Baden et al., 2015; Keulartz & Van den Belt, 2016*). This is largely being driven by the so-called “maker movement” in which people are now exploiting the widespread popularity and accessibility of fabrication tools (3D printers, laser cutters, CNC machines, etc.) and open source electronics (Arduino, Raspberry Pi, etc) to build simple and advanced scientific equipment

for a diversity of applications (Pearce, 2012). Designs and instructions are shared freely, typically under some form of open source license, and generally undergo several rounds of improvement as a result of the contributions from other users. Importantly, such innovations have matured to a level that it is possible to setup a basic, functional, laboratory space at extremely low cost. In addition, the spirit of “frugal science” has led to several innovations in low cost diagnostic tools, often built around cell phone platforms. These approaches have the important potential for lowering the cost of diagnosis and treatment of diseases in both developed and developing countries.

In the context of the biological sciences, the “do-it-yourself biology (DIYBio) movement” has driven the development of several tools that are critical in any cell/molecular biology laboratory (Landrain et al., 2013). This includes open source pipettes (Baden, 2014), centrifuges (Wong et al., 2008), water baths (Garvey, 2012), stirrers and hot plates (Watts, 2011), shakers (Miller, 2010), electrophoresis kits (Long, 2011), incubators (Pelling, 2015), PCR (Chai Biotechnologies Inc., 2014a; Wong et al., 2015) and qPCR (Chai Biotechnologies Inc., 2014b) machines, and low cost kits for manipulating DNA or transforming bacteria (Synbiota, 2017). One other key piece of equipment is a light microscope. Several designs and approaches have been developed for creating low cost light microscopes with reasonable magnification (Cybulski, Clements & Prakash, 2014). Such designs have resulted in microscopes that can operate in a variety of modalities including bright field, dark field and fluorescence (Cybulski, Clements & Prakash, 2014; McLeod & Ozcan, 2016). Cellphone based microscopes have been developed in which the phone’s camera is simply employed as the imaging sensor (Contreras-Naranjo, Wei & Ozcan, 2016). These approaches either mount a low cost set of optics directly to the cellphone camera (Zhu et al., 2011) or mount the cell phone onto a simplified microscope stand that employs microscope objectives (Skandarajah et al., 2014). Alternatively, discarded webcams can be converted into the microscope by taking apart the camera and simply flipping the lens in front of the imaging sensor (Switz, D’Ambrosio & Fletcher, 2014), or placing a low cost ball lens in front of an imaging sensor or the eye (Smith et al., 2011). Lenses can also be sourced from a discarded CD-ROM drive or an optical mouse (Cavanihac, 2006; Ibanez, 2012). The ability to generate functional, low cost microscopes is made more attractive as they can be simply produced from discarded electronics. Indeed, the simplest embodiment of a DIY Microscope can be achieved by simply placing a water drop on the cellphone front facing camera. These various approaches are not only important for educational purposes, but also have a significant role to play in developing low cost diagnostic tools for the lab or the field (Landrain et al., 2013; Baden, 2014; Wong et al., 2008; Garvey, 2012; Watts, 2011; Miller, 2010; Long, 2011; Pelling, 2015; Chai Biotechnologies Inc., 2014a; Wong et al., 2015; Chai Biotechnologies Inc., 2014b; Synbiota, 2017; Cybulski, Clements & Prakash, 2014; McLeod & Ozcan, 2016; Contreras-Naranjo, Wei & Ozcan, 2016; Zhu et al., 2011; Skandarajah et al., 2014; Switz, D’Ambrosio & Fletcher, 2014; Smith et al., 2011; Cavanihac, 2006; Ibanez, 2012; Kim, Gerber & Riedel-Kruse, 2016).

As low-cost imaging tools (and general labware) become more prevalent there is an increasing need for the development of software control and monitoring solutions. In order to employ a DIY Microscope in a research setting, one may desire the ability to conduct

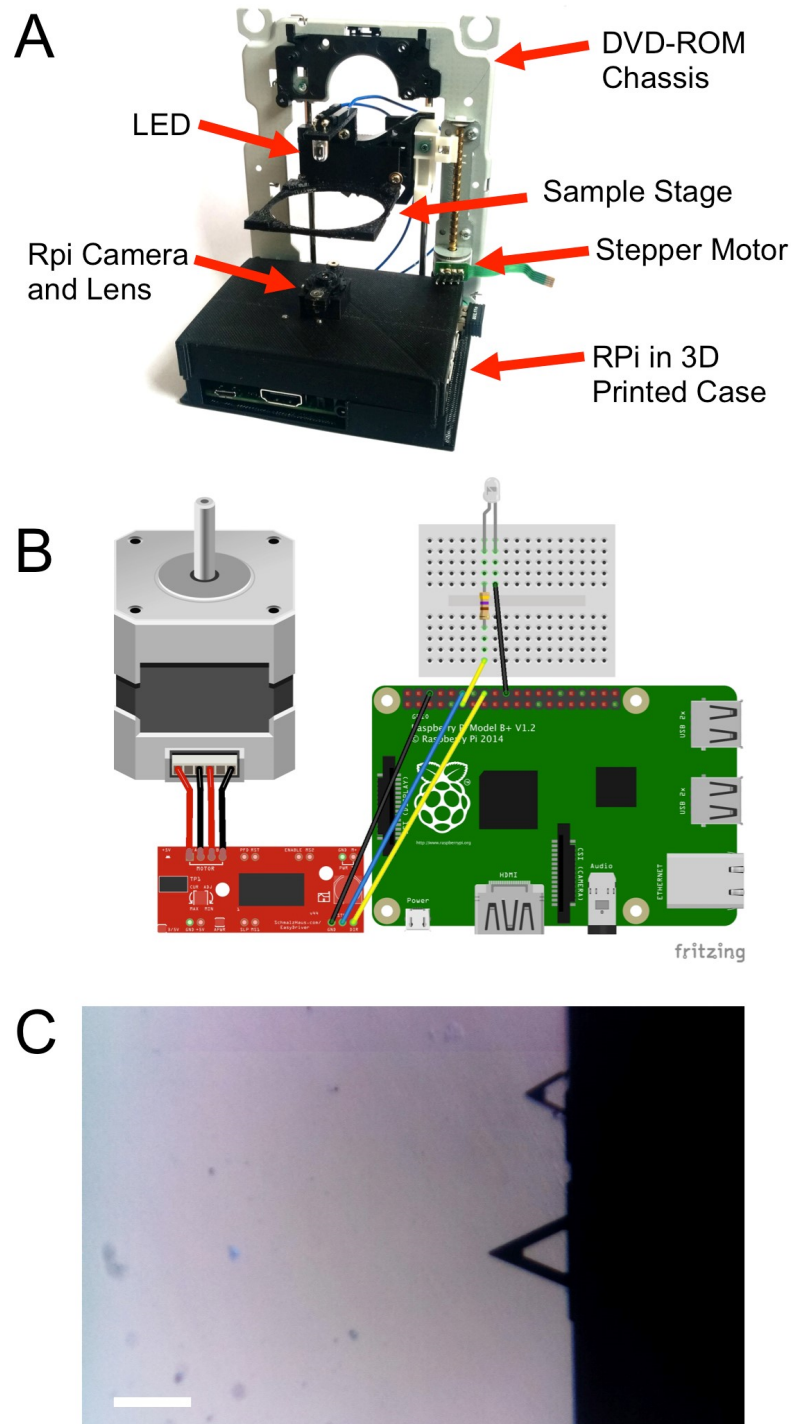
imaging experiments without having to be physically present at the microscope in order to initiate image capture. For example, conducting time-lapse experiments of cell growth on a microscope has been placed inside of a sterile, temperature and atmospherically controlled incubator. Therefore, the purpose and objective of this study was to develop a general proof-of-principle approach and physical embodiment of DIY Microscope control that relies on freely available programs that can be installed on a PC or cellphone. In order to achieve these goals, we first constructed a basic DIY Microscope by modifying the popular Raspberry Pi (RPI) camera to act as an objective (Switz, D'Ambrosio & Fletcher, 2014). A 3D printed case for the RPi computer and camera was designed and constructed to form a microscope stand. Finally, a DVD-ROM drive was used to create a moveable sample stage, allowing for focus control. Sample positioning along the optical axis of the microscope and image capture was controlled by a Python script.

Once the DIY microscope was constructed, we developed user interfaces by exploiting three popular existing applications and their available application program interfaces (APIs). Here, we demonstrate the ability to use the popular Twitter interface to send commands to an Internet connected DIY microscope. We also implemented online data storage by uploading captured images, along with important metadata, to the photo-sharing network Flickr. In this scenario, the DIY Microscope was assigned a Twitter account (@DIYMicroscope), which monitored for simple commands sent by any other Twitter user. Simple message syntax was developed in order to allow other user to adjust microscope focus, capture single images or initiate time-lapse recordings. Upon image capture, all data was stored on a publically accessible Flickr account. In a second embodiment, we developed an approach to control a local DIY Microscope by exploiting the API of the popular video game Minecraft. Here, we first constructed a virtual lab inside of the Minecraft world. In this scenario, one is able to “play” within this virtual world and use gaming actions to control a physical DIY microscope. Simple actions inside of the videogame allowed one to again adjust microscope focus, capture single images or initiate time-lapse recordings. In this case, the images are stored locally on the RPi hard drive. In this embodiment, there is no need for a consistent or reliable Internet connection. To our knowledge, this is the only social media and video game controlled microscope.

## MATERIALS AND METHODS

### DIY Microscope

A basic DIY Microscope was constructed by employing strategies that have been previously demonstrated (Cybulski, Clements & Prakash, 2014; Switz, D'Ambrosio & Fletcher, 2014; Cavanihac, 2006). Briefly, the microscope was constructed using a Raspberry Pi (RPI) model B+ as the control computer, an RPi camera module (Raspberry Pi Foundation, Cambridge, UK), a discarded computer DVD-ROM drive and a 3D printed frame (MakerBot Replicator 2; MakerBot, Brooklyn, NY, USA) (Fig. 1A). All 3D printer files are available online at <http://www.thingiverse.com/pellinglab>. The original lens from the RPi camera module was removed prior to installation, leaving only the image sensor (OmniVision OV5647-5MPx; OmniVision, Santa Clara, CA, USA). A web camera lens (Logitech c310) was then inverted



**Figure 1** **DIY microscope construction.** (A) A case was 3D printed in order to mount the RPi computer, camera and lens assembly and DVD-ROM drive chassis. A sample stage was also 3D printed and mounted to the laser pickup assembly. (B) A simple motor driver was then employed to control the stepper motor with the GPIO pins of the RPi. (C) Calibration of the microscope was achieved by acquiring images of microfabricated atomic force microscopy cantilevers. The lower cantilever has a known length of  $200\ \mu\text{m}$ , corresponding to 190 pixels in the image (scale bar =  $200\ \mu\text{m}$ ).

Full-size DOI: [10.7717/peerjcs.139/fig-1](https://doi.org/10.7717/peerjcs.139/fig-1)

and installed on top of the image sensor. The RPi camera module and the lens were maintained in fixed positions relative to one another by mounting in a 3D printed mount.

The DVD-Rom drive was then disassembled leaving only the stepper motor, the laser pickup assembly and the frame intact. The drive was then mounted perpendicular to the RPi camera assembly in order to create a sample positioning stage that could be moved along the optical axis of the microscope in an inverted configuration. In order to fix the positions of each component of the entire assembly, a 3D printed frame was produced to which all components could be mounted. Finally, a sample stage was 3D printed and mounted to the laser pickup assembly. This configuration allowed us to easily adjust sample position and focus under computer control. The movement of the sample tray was achieved by controlling the movement of the stepper motor by employing the Easy Driver Stepper Motor Driver V4.4 (<http://www.schmalzhaus.com/EasyDriver>) in 8-step micro stepping mode. The driver and a standard 5 mm white LED were connected to RPi via GPIO pins in order to control sample positioning and illumination (Fig. 1B).

To calibrate the DIY microscope, we acquired images of standard microfabricated cantilevers commonly employed used in atomic force microscopy. The cantilevers have known dimensions. The larger cantilever in the image is 200  $\mu\text{m}$ , corresponding to 190 pixels in a 1,600 by 900 pixel image (Fig. 1C).

### Online control of the DIY microscope

A Python script was written that allows any Twitter subscriber to remotely interact with the microscope via the Twitter app or website. The code we developed is available online at <https://github.com/pellinglab>. The open source Python library Tweepy (<http://www.tweepy.org>) was employed to facilitate communication with Twitter's application programming interface (API). The microscope was assigned the Twitter account @DIYMicroscope in order to facilitate user interaction. The Python script running on the RPi monitors 'tweets' sent to the account @DIYMicroscope and examines them for simple key words. For example, the Twitter user can capture single images, control sample positioning and focus, and initiate time-lapse imaging (details are presented in the 'Results and Discussion' section).

### Online image capture and storage

The RPi storage capacity can be limited as it is defined by the capacity of the SD card the user has employed. In order to prevent memory issues associated with a large number of image acquisitions, we utilized the photo-sharing website Flickr ([flickr.com](http://www.flickr.com)), as an image hosting platform. To remotely and automatically interact with the Flickr API, we implemented the Beej Flickr API Python library (<http://stuvel.eu/flickrapi>). Each time an image is acquired by the script (i.e., a single frame or a concatenate of frames), a copy is uploaded to the DIY microscope's Flickr account. Then the original image is removed then from the RPi hard drive to save space.

### Offline control of the DIY microscope

To locally control the microscope, we designed a Python script that will create a user interface in the videogame universe of Minecraft (Mojang). The code we developed is

available online at <https://github.com/pellinglab>. We constructed a virtual lab in which user actions during game play can be used to initiate specific microscope functions. To achieve this, we employed the Minecraft Python API library (<http://www.stuffaboutcode.com/p/minecraft-api-reference.html>). Inside the virtual laboratory, the sword-equipped player can control the microscope by performing specific actions with ‘control blocks’ inside the virtual laboratory. Block actions were designed to allow the user to generate a live preview, capture an image and adjust the stage position for focus control. Captured images are stored in a specific local folder for future use.

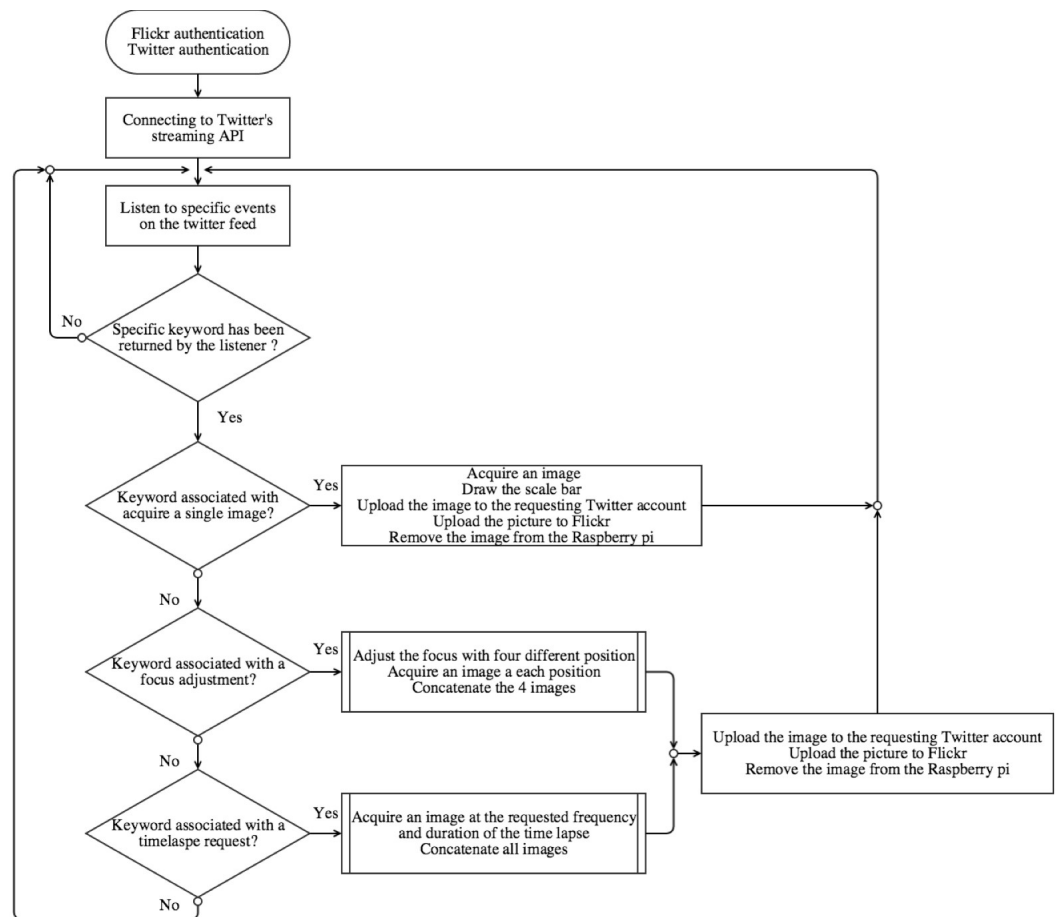
## RESULTS AND DISCUSSION

### Software design for twitter control

When the Python script is launched, an authentication procedure to *Flickr* is initiated (a *Yahoo!* account, API key and API secret are required), followed by an authentication to twitter (a *Twitter* account, API key and API secret, token and secret token are required). Both accounts must be established and verified by the system administrator in advance. The Flickr account is required for storage of images acquired by the DIY Microscope. The DIY Microscope will be addressed and controlled by sending ‘tweets’ that mention the system specified Twitter account. When the authentication between Flickr and Twitter is correctly established, the script connects to Twitter’s streaming API with specific keywords. This allows the program to obtain real-time tweets from the social network. When a user sends a tweet to the system-specified Twitter account containing single, or multiple, keywords, a JavaScript Object Notation (JSON) object is returned by the streaming API, containing parameters such as the user name, screen name, location, *tweet* content and the time. The JSON object is then examined and conditional actions are undertaken depending on the keywords identified in the tweet (Fig. 2). Importantly, to avoid any unwanted interactions with the DIY Microscope (i.e., a random user has one of the keywords in their tweets), the requesting user must include the system-specified twitter handle (e.g., @example) in their tweet.

In this embodiment, we designed four types of user interaction, such as taking a single image, initiating a timelapse, adjusting the focus and obtaining a ‘focus group’ image (Figs. 3 and 4). When the tweet contains the keyword ‘singleimage’ (Fig. 3A), the current image frame captured by the RPi camera is temporarily stored on the RPi, a scale bar is drawn on the image and returned to the requesting user in a Twitter message (Fig. 3B). In addition to the returned image, the reply tweet also includes the message ‘@user scale bar is X\_UNITS’, where @user is the Twitter handle of the requesting user and X\_UNITS = value determined by the user after microscope calibration. Finally, the temporarily stored local copy of the image is then uploaded to the user specified Flickr account and permanently removed from the RPi. In Fig. 3, a single image is requested by the user @pellinglab and an image of cellulose derived from apples (*Modulevsky et al., 2014*) is acquired and returned by the microscope.

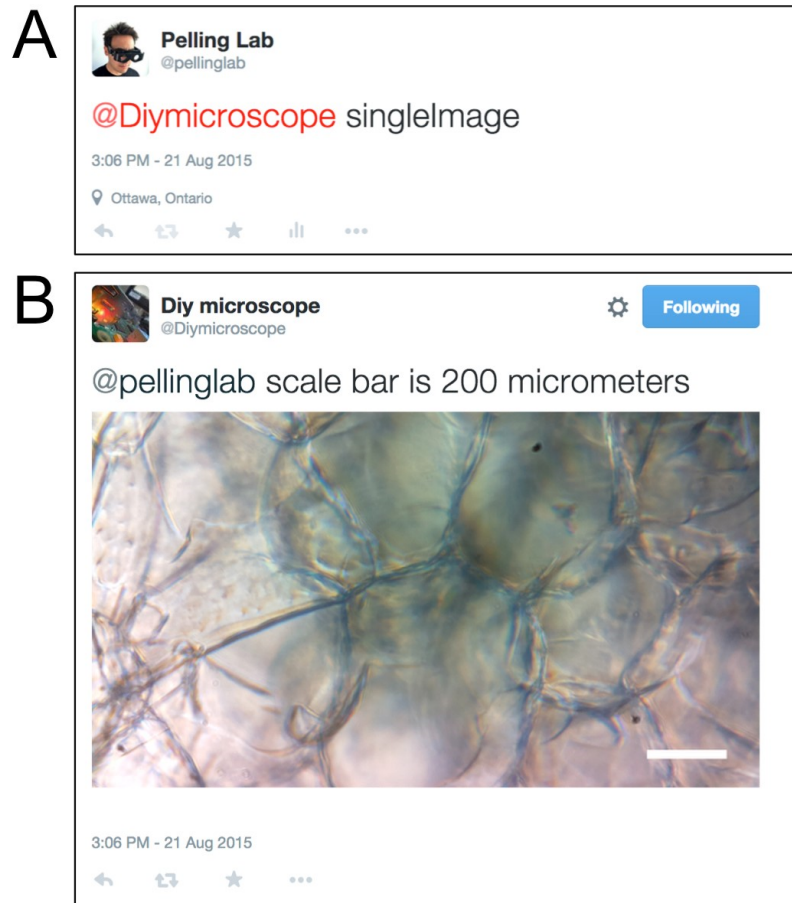
To move the stage in order to focus the sample, the user sends a tweet to the DIY Microscope that includes the keywords ‘diyfocus further N’ or ‘diyfocus closer N’ (N =



**Figure 2** Flow-chart representing our Python script that enables Twitter-based user interaction with the DIY Microscope.

Full-size DOI: 10.7717/peerjcs.139/fig-2

integer value). The sample stage will then move further away, or closer to, the RPi camera by  $N$  micropixels. An image will be acquired after moving to the new position, sent back to the user through twitter and stored on Flickr as above. If the user wants to sample multiple focus positions, a tweet is constructed which includes the keywords 'diyfocus dofocus  $N$ ' ( $N = \text{integer value}$ ). In this scenario, the sample platform is moved away from the RPi camera by  $N$  micropixels and an image is acquired. The sample then moves another  $N$  micropixels away from the camera a second image is obtained before moving back to the original position. The process is then repeated in the opposite direction in order to acquire image number 3 and 4. The four locally stored images are uploaded to Flickr, along with their metadata (the twitter handle of the requesting user, the original message sent by the user,  $N$  and the corresponding frame number), and then concatenated into a single 900 by 900 pixel image (Fig. 4A). The concatenated image is then returned to the user with the corresponding frame numbers printed on each sub-image along with the message '@user your  $N$  micropixel dofocus sequence is complete' (where @user and  $N$  are determined from the original user message). The user can now adjust the stage position to the desired



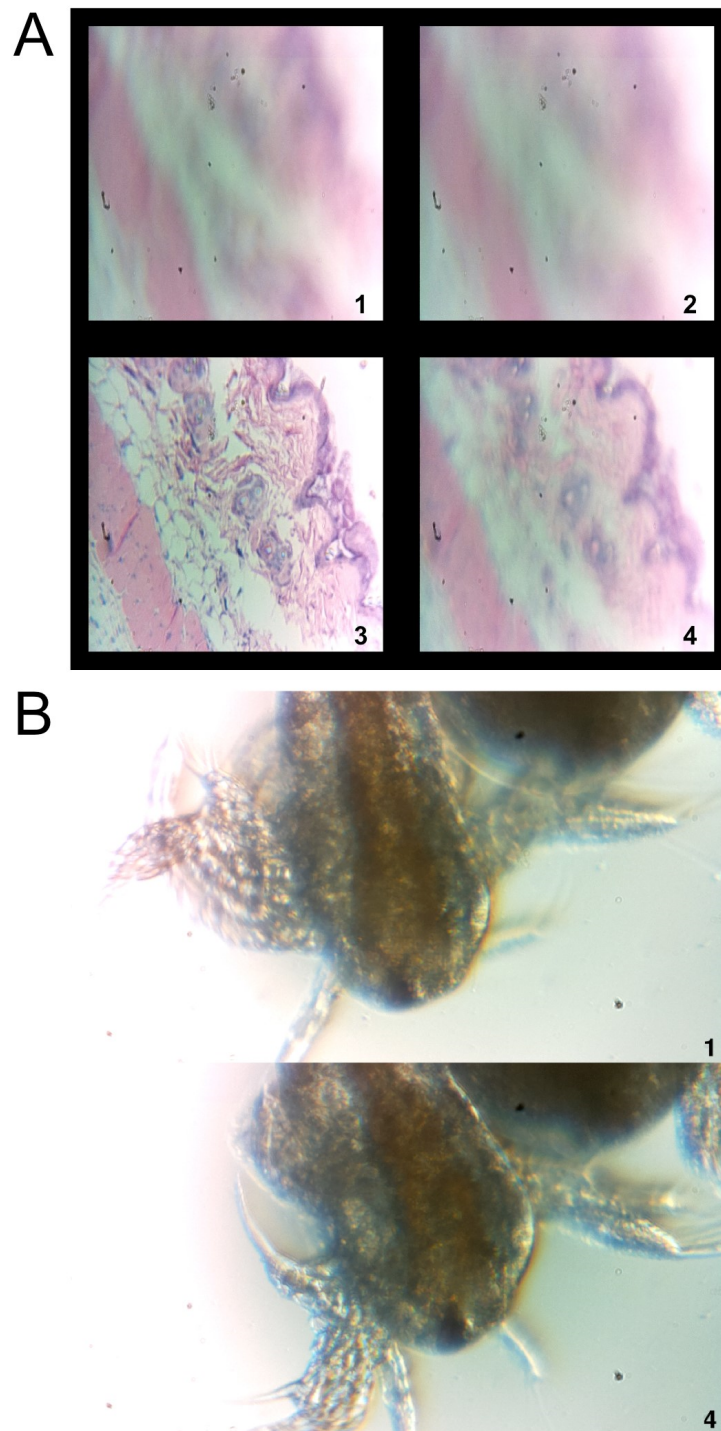
**Figure 3** Twitter acquisition of a single image from the DIY Microscope. (A) The account @pellinglab initiates image acquisition by posting a tweet that mentions the account @DIYMicroscope and includes the keyword 'singleImage'. Other text can be included in the message but the script will ignore them. (B) The DIY microscope acquires an image and responds to the resting account.

Full-size DOI: 10.7717/peerjcs.139/fig-3

location (using the 'diyfocus further/closer' keywords, followed by an appropriate integer value). In Fig. 4A, the concatenated image of a hematoxylin and eosin stained histology sample being moved in and out of focus is shown.

Finally, the user can initiate a time lapse using the 'diytimelapse duration D frequency F' keywords, replacing the 'D' and 'F' with integers for duration and frequency, respectively. The D and F, values assume units of minutes and frames/minute, respectively. The timelapse is carried out as requested and each acquired image is uploaded to Flickr along with its corresponding metadata (the twitter handle of the requesting user, the original message sent by the user, D, F and the corresponding frame number). Upon completion of the timelapse, a concatenated image containing the first and last frame of the time lapse is constructed and returned to the user with the frame numbers printed on each sub-image (Fig. 4B). The returned image also includes the message '@user your timelapse of D minutes at F frames/minute is now complete' (where @user, D and F are determined





**Figure 4** Focus and time-lapse control through Twitter interactions. (A) The command 'diyfocus dofocus N' initiates image capture of the sample when moved to four specific positions. The user is then provided with a concatenated image containing the four acquired, and indexed, images. This routine allows the user to determine the optimal sample position in order to set the focus. (continued on next page...)

Full-size  DOI: 10.7717/peerjcs.139/fig-4

**Figure 4 (...continued)**

In this case, a sample histology slide was employed for imaging. As these images are only for stage positioning purposes no scale bar is included; however, each image is  $0.95 \times 0.95$  mm. (B) It is also possible to initiate time-lapse imaging using the keywords 'diytimelapse duration D frequency F', D and F, correspond to integer values with units of minutes and frames/minute, respectively. In this case, brine shrimp (*Artemia*—commonly grown as live feed for fish larva) were imaged with the microscope. A final image is returned to the user that only contains the first and last image acquired during the timelapse interval. All images from the sequence are stored on Flickr along with any relevant metadata. No scale bar is provided in this example; however, the scale is known after user calibration and relevant details are included in the Flickr metadata. In this case each image is  $1.70 \times 0.95$  mm.

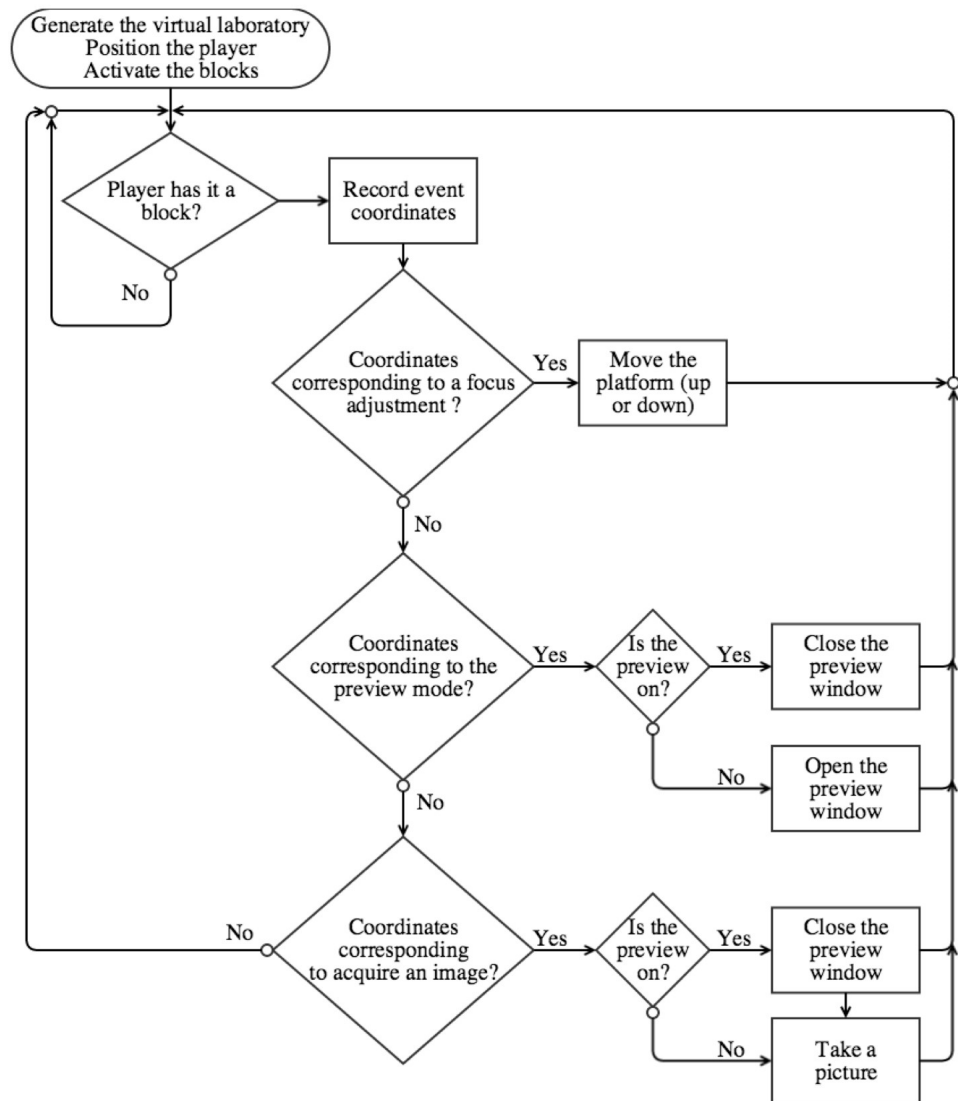
from the original user message). In Fig. 4B, time-lapse microscopy was conducted on a sample of *Artemia* (brine shrimp) and the image sent back to the user through twitter is shown. Four images were acquired (duration of one minute at a frequency of four images per minute) and as described above, only the first and last images are sent to the user through twitter. All images in the sequence are stored on Flickr for later analysis by the user.

Importantly, whenever an image is uploaded to Flickr (irrespective of the keyword(s) employed), specific metadata is included with the image in order to allow for identification and filtering. The metadata associated with each image includes the twitter handle of the requesting user, the keyword(s), a timestamp and the frame number.

**Software design for minecraft control**

In situations where a user may lack Internet access, we designed an approach for local, offline, control of the DIY Microscope. In this case, we designed a Python script that allows for user interaction through the videogame Minecraft (Fig. 5). Conveniently, Minecraft is already included in the freely available Raspbian distribution of Linux for the RPi. When our Python script is launched, the player position is immediately updated to be facing the virtual 'laboratory' (Fig. 6).

Upon entering the laboratory, a sword-equipped player can now interact with the DIY microscope 'control blocks' (Fig. 6A). When standing in front of the desired control block, the right mouse button can be used to initiate a specific action. Such actions will return an event object, containing the coordinates of the specific block as a tuple. The coordinate tuple is then used to initiate conditional actions that are used to control the 'real-life' DIY Microscope. When the player activates one the extremity blocks (black and grey blocks), the stepper motor will perform N microsteps clockwise (or counterclockwise). The user can specify N in the Python script. The yellow block will display a live preview of the sample, on a 640 by 480 pixels window (Fig. 6B). The blue blocks will take a single image when activated, and save the image as a ".png" file on the RPi. To avoid over usage of the GPU memory, the program will close live preview (if open) prior to acquiring an image. A message on the minecraft user-interface is displayed when the image is successfully stored (Fig. 6C).



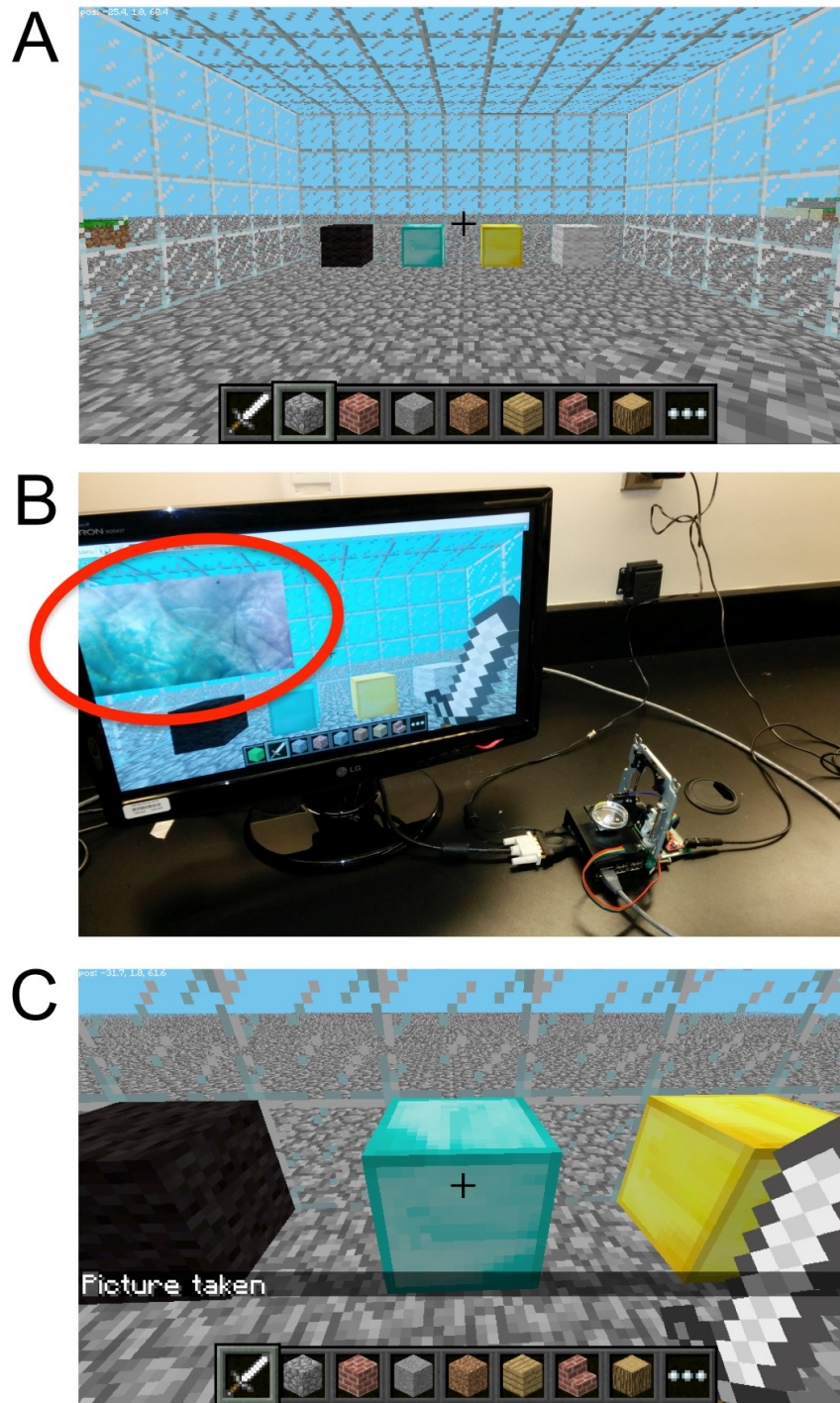
**Figure 5** Flow-chart representing our Python script that enables Minecraft-based user interaction with the DIY Microscope.

Full-size  DOI: [10.7717/peerjcs.139/fig-5](https://doi.org/10.7717/peerjcs.139/fig-5)

## CONCLUSIONS

### Possible limitations

The Twitter public streaming API limits the application to a fixed number of keyword filters per application (*Twitter, Inc, 2015*). The current application only requires 3 keywords (“singleImage”, “diyfocus” and “diytimelapse”) to initiate user-interaction, however, more complex interactions may require many more keywords. As well, user-interaction is also limited by the number of allowed connections to the API per hour (*Twitter, Inc, 2015*) but the exact number is not currently publically available. Importantly, a user will also receive an error message from Twitter if they post the exact same tweet multiple times



**Figure 6** The Minecraft environment. (A) The user simply uses the sword to control a locally connected DIY Microscope by interacting with the four control blocks inside of a virtual 'laboratory'. The blocks allow the user to adjust the sample stage up and down (black/gray 6 blocks), (B) obtain a live preview (yellow block) and (C) acquire an image (cyan block).

Full-size  DOI: [10.7717/peerjcs.139/fig-6](https://doi.org/10.7717/peerjcs.139/fig-6)

over a short time period. Therefore, Twitter will not allow a single user to post the tweet '@DIYMicrosocpe singleImage' more than once. The user can overcome this limitation by adding more text to the tweet or deleting the original tweet. Currently, the Flickr API also limits the application to 3,600 requests per hour (*Yahoo!, 2015*).

Some other limitations can also arise if multiple users are attempting to interact with the microscope simultaneously. For instance, if a time-lapse sequence is initiated, the script will complete the image acquisition before returning to the stream listener. In this case, a user sending a request will not get an immediate response from the microscope.

Finally, the microscope is inherently limited to the presence and stability of Internet connectivity available to the RPi. In the case of a lost connection, the script will have to be re-executed in order to establish a connection to the Twitter streaming API and Flickr API. To overcome the issue with Internet connectivity, we also implemented a control interface within the Minecraft universe. Of course, electrical power is always required as with any modern microscope. However, it is possible to power the RPi using solar panels and batteries in cases where microscopic imagery is required but electrical connections are not easily accessed. As the Minecraft implementation of the DIY Microscope stores pictures locally, the maximum storage space will depend on the SD card capacity. Cloud storage can be implemented with the *Minecraft* program, but will require an Internet connection. However, if storage space becomes a limitation, the user can move the files to an alternative storage device (usb stick or via local file transfer) or employ a larger SD card.

### Possible extensions

Future versions of both the *Twitter* and *Minecraft* interfaces could include image analysis features, such as cell counting, cell tracking for time lapses, image redundancy protections and thresholding by implementing OpenCV or other methods (*Itseez, 2017*). Such integration could allow the user to obtain qualitative and quantitative from the sample, in a remote or local way. An automatic focusing library could also be implemented to both program to enhance image quality acquisition rapidity. In its current implementation, our approach does not maintain privacy as all micro blogging posts and pictures on *Twitter* and *Flickr* are publicly available. In order to overcome this potential limitation, a smart phone based application could easily be developed to interact with the current version of the microscope, either via a direct connection (*wifidirect* or *bluethooth*) or a web-based server. Such a configuration could allow the user to use customize features, independent from *Twitter*, *Flickr* and *Minecraft APIs*, consequently avoiding the rate limitation from the third-parties. This will also be important for the implementation of a potential multi-user platform. Although the purpose of this study was to present a proof-of-concept implementation of using social media and video games to control our microscope, one can imagine extending this platform to allowing multi-user experiments. Future work will require rigorous testing to ensure the robustness of our embodiment for such applications. Of potential interest is that both of our approaches can also be exploited on other types of laboratory equipment. One can easily interface the multiple outputs of the RPi to an existing or "hand-made" equipment, or extends existing programs (*Baden et al., 2015; Keulartz & Van den Belt, 2016; Pearce, 2012; Landrain et al., 2013*). Future use of

the *Minecraft* program can let access to multiple player in the “virtual laboratory”, either by a local or internet connection. This let the possibility for the “players” to cooperatively interact and controls “real” physical laboratory equipment inside a virtual world.

## ACKNOWLEDGEMENTS

The authors would like to thank Daniel Modulevsky and Dr. Charles Cuerrier for providing the histological sample for imaging.

## ADDITIONAL INFORMATION AND DECLARATIONS

### Funding

This work was supported by the National Sciences and Engineering Research Council Discovery Grant. Andrew E. Pelling is supported by the Canada Research Chairs Program. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

### Grant Disclosures

The following grant information was disclosed by the authors:  
National Sciences and Engineering Research Council Discovery Grant.  
Canada Research Chairs Program.

### Competing Interests

The authors declare there are no competing interests.

### Author Contributions

- Maxime Leblanc-Latour conceived and designed the experiments, performed the experiments, analyzed the data, contributed reagents/materials/analysis tools, wrote the paper, prepared figures and/or tables, performed the computation work, reviewed drafts of the paper.
- Craig Bryan conceived and designed the experiments, contributed reagents/materials/-analysis tools, wrote the paper, performed the computation work, reviewed drafts of the paper.
- Andrew E. Pelling performed the experiments, analyzed the data, wrote the paper, prepared figures and/or tables, reviewed drafts of the paper.

### Data Availability

The following information was supplied regarding data availability:  
Github: [https://github.com/pellinglab/minerscope\\_and\\_twittercope](https://github.com/pellinglab/minerscope_and_twittercope).

## REFERENCES

**Baden T.** 2014. Biropette: customisable, high precision pipette. Thingiverse. Available at <http://www.thingiverse.com/thing:255519>.

- Baden T, Chagas AM, Gage G, Marzullo T, Prieto-Godino LL, Euler T. 2015.** Open Labware: 3-D printing your own lab equipment. *PLOS Biology* **13(3)**:e1002086 DOI [10.1371/journal.pbio.1002086](https://doi.org/10.1371/journal.pbio.1002086).
- Cavanilhac J-M. 2006.** DIY: an inverted microscope with electronic focusing. *Micscape Magazine*. Available at <http://www.microscopy-uk.org.uk/mag/indexmag.html?http://www.microscopy-uk.org.uk/mag/artaug06/jmc-constr1.html>.
- Chai Biotechnologies Inc. 2014a.** OpenPCR—the \$649 open source pcr machine/thermal cycler. Available at <http://openpcr.org/>.
- Chai Biotechnologies Inc. 2014b.** Open qPCR: the \$2499 real-time PCR machine. Available at <https://www.chaibio.com/openqpcr>.
- Contreras-Naranjo JC, Wei Q, Ozcan A. 2016.** Mobile phone-based microscopy, sensing, and diagnostics. *IEEE Journal of Selected Topics in Quantum Electronics* **22(3)**:392–405 DOI [10.1109/JSTQE.2016.2545685](https://doi.org/10.1109/JSTQE.2016.2545685).
- Cybulski JS, Clements J, Prakash M. 2014.** Foldscope: origami-based paper microscope. *PLOS ONE* **9(6)**:e98781 DOI [10.1371/journal.pone.0098781](https://doi.org/10.1371/journal.pone.0098781).
- Garvey C. 2012.** Arduino water bath for DIYbio. Available at <https://github.com/cathalgarvey/KettleKontroller>.
- Ibanez L. 2012.** The kitware blog—DIY microscopy: optical mouse lens + iPad. Available at <http://www.kitware.com/blog/home/post/256>.
- Itseez. 2017.** OpenCV. Available at <http://opencv.org/>.
- Keulartz J, Van den Belt H. 2016.** DIY-Bio—economic epistemological and ethical implications and ambivalences. *Life Sciences, Society and Policy* **12(1)**:1–19 DOI [10.1186/s40504-016-0034-6](https://doi.org/10.1186/s40504-016-0034-6).
- Kim H, Gerber LC, Riedel-Kruse IH. 2016.** Interactive biotechnology: building your own biotic game setup to play with living microorganisms. In: *Proceedings of the 2016 CHI conference extended abstracts on human factors in computing systems*. New York: ACM, 1000–1002.
- Landrain T, Meyer M, Perez AM, Sussan R. 2013.** Do-it-yourself biology: challenges and promises for an open science and technology movement. *Systems and Synthetic Biology* **7(3)**:115–126 DOI [10.1007/s11693-013-9116-4](https://doi.org/10.1007/s11693-013-9116-4).
- Long J. 2011.** Gel electrophoresis system (mini). Available at <http://www.instructables.com/id/Gel-electrophoresis-system-mini/>.
- McLeod E, Ozcan A. 2016.** Unconventional methods of imaging: computational microscopy and compact implementations. *Reports on Progress in Physics* **79(7)**:076001 DOI [10.1088/0034-4885/79/7/076001](https://doi.org/10.1088/0034-4885/79/7/076001).
- Miller J. 2010.** Open source orbital shaker. Available at <https://www.thingiverse.com/thing:2633507>.
- Modulevsky DJ, Lefebvre C, Haase K, Al-Rekabi Z, Pelling AE. 2014.** Apple derived cellulose scaffolds for 3D mammalian cell culture. *PLOS ONE* **9(5)**:e97835 DOI [10.1371/journal.pone.0097835](https://doi.org/10.1371/journal.pone.0097835).
- Pearce JM. 2012.** Materials science. Building research equipment with free, open-source hardware. *Science* **337(6100)**:1303–1304 DOI [10.1126/science.1228183](https://doi.org/10.1126/science.1228183).

- Pelling AE. 2015.** DIY CO<sub>2</sub> incubator bioreactor for mammalian cell culture. Available at [https://www.pellinglab.net/single-post/diy/DIY-CO<sub>2</sub>-Incubator-Bioreactor-for-Mammalian-Cell-Culture](https://www.pellinglab.net/single-post/diy/DIY-CO2-Incubator-Bioreactor-for-Mammalian-Cell-Culture).
- Skandarajah A, Reber CD, Switz NA, Fletcher DA. 2014.** Quantitative imaging with a mobile phone microscope. *PLOS ONE* **9**(5):e96906  
DOI 10.1371/journal.pone.0096906.
- Smith ZJ, Chu K, Espenson AR, Rahimzadeh M, Gryshuk A, Molinaro M, Dwyre DM, Lane S, Matthews D, Wachsmann-Hogiu S. 2011.** Cell-phone-based platform for biomedical device development and education applications. *PLOS ONE* **6**(3):e17150  
DOI 10.1371/journal.pone.0017150.
- Switz NA, D'Ambrosio MV, Fletcher DA. 2014.** Low-cost mobile phone microscopy with a reversed mobile phone camera lens. *PLOS ONE* **9**(5):e95330  
DOI 10.1371/journal.pone.0095330.
- Synbiota. 2017.** Synbiota homepage. Available at <https://synbiota.com/>.
- Twitter, Inc. 2015.** Connecting to a streaming endpoint. Available at <https://dev.twitter.com/streaming/overview/connecting>.
- Watts M. 2011.** Magnetic Stirrer. Available at <http://www.teklalabs.org/magnetic-stirrer/>.
- Wong AP, Gupta M, Shevkoplyas SS, Whitesides GM. 2008.** Egg beater as centrifuge: isolating human blood plasma from whole blood in resource-poor settings. *Lab Chip* **8**(12):2032–2037 DOI 10.1039/b809830c.
- Wong G, Wong I, Chan K, Hsieh Y, Wong S. 2015.** A rapid and low-cost PCR thermal cycler for low resource settings. *PLOS ONE* **10**(7):e0131701  
DOI 10.1371/journal.pone.0131701.
- Yahoo!. 2015.** Flickr: the flickr developer guide—API. Available at <https://www.flickr.com/services/developer/api/>.
- Zhu H, Mavandadi S, Coskun AF, Yaglidere O, Ozcan A. 2011.** Optofluidic fluorescent imaging cytometry on a cell phone. *Analytical Chemistry* **83**(17):6641–6647  
DOI 10.1021/ac201587a.