**Fluorostat Operation**

Contents

[Overview](#_Overview)

[Parts](#_Parts)

[Setup](#_Setup)

[Software](#_Software)

[Calibrating the Device](#_Calibrating_the_Device)

[Preparing a Run](#_Preparing_a_Run)

[Starting a Run](#_Starting_a_Run)

[Killing a Run](#_Killing_a_Run)

[Clean Up](#_Clean_Up)

[Analysis](#_Data_and_Analysis)

# Overview

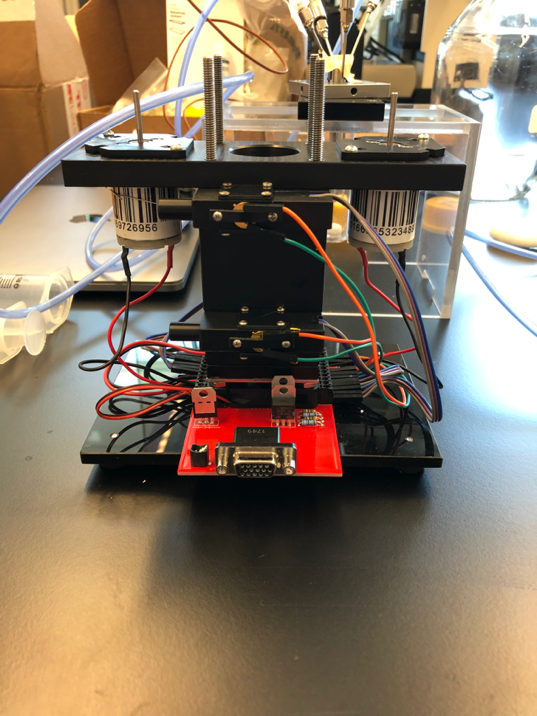
The fluorostat is a continuous culture device that has the capability of taking fluorescence measurements with two different colors. There are two different versions of the fluorostat: standard mode, which is appropriate for microbes that don’t require lighting or aeration, and cyano mode, which can bubble in air, and has a lighting set up for photosynthetic organisms such as *Synechococcus elongatus*.

|  |
| --- |
| **Standard Mode** |
| A screenshot of a video game  Description automatically generated |
| **Cyano Mode** |
| A screenshot of a cell phone  Description automatically generated |

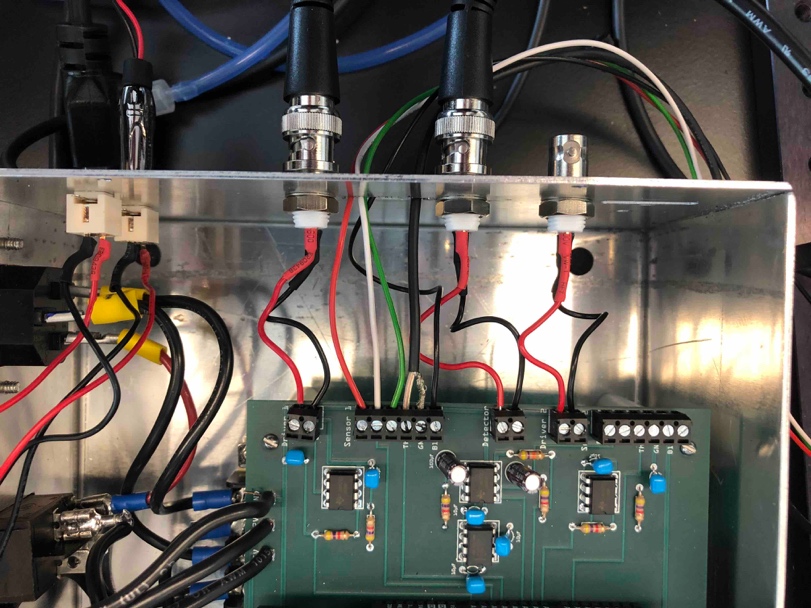
# 

# Parts

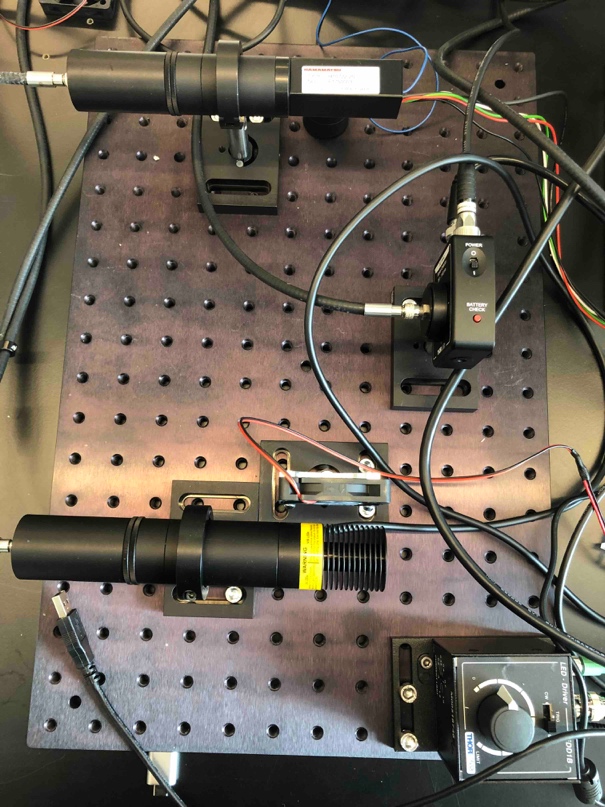
**Sample Holder**



**Motherboard**



**Optics Breadboard**



# Setup

Adjusting height of needles- use a small hex key to loosen them and you can then move them up and down.

Motors rotate counter clockwise

-the outer tubing goes to the bottle and the inner tubing goes to the vial

# Software

Choose your operating system:

The fluorostat runs successfully on Unix based OS, and has specifically been used on:

* MacOS
* Ubuntu
* Raspbian (Debian on the Raspberry Pi)

In theory, it can work on Windows, but differences in the USB communication protocol have prevented me from successfully getting it to work.

Installing Software:

The following programs need to be included:

* Python 3
  + Dependencies (most of these will be already included in python. You can see which ones you don’t have by attempting to run the code and seeing what modules are missing)

Sys

OS

Numpy

Matplotlib

Datetime

Time

Collections

Pyserial

To install

Pip3 install pyserial

* Arduino
  + Dependencies
    - The thermocouple needs to Adafruit\_MAX31855.h module
      * In Arduino, go to Tools>Manage Libraries
      * Search “Adafruit\_MAX31855”
      * Select Install

The following programs are optional but helpful:

* Dropbox (or some kind of file cloud service)
* Anaconda/Jupyter Notebook for Analysis (Not possible on a Raspberry Pi)

The fluorostat depends on two programs to run: Arduino and Python. The Arduino code uploads information to the fluorostat motherboard, and tells what functions it should do. The python code collects the data from the devices into a csv. In other words, if you run the Arduino code, but don’t run the python code, the fluorostat will function, BUT you will collect no data!

* 1. Arduino
     1. Make sure that the COM Port selected
     2. TwoColors
        1. Before uploading the code, scroll to the bottom and uncomment one of the following
           1. turbidostat(target\_OD);

Runs as a turbidostat diluting when OD passes threshhold

* + - * 1. fluorostat(Fluorostat\_target\_channel,Target\_fluoro[Fluorostat\_target\_channel]);//(channel (0 or 1), gain, target reading)

Clamp on fluorescence, channel tells you which color you are clamping on and target\_fluoro is a gain of the target reading. You will need to calibrate to get this number

* + - * 1. Pump\_for\_Exp\_Start();

Turns on the pumps

* + - * 1. rolling\_measure();
      1. To kill the code, comment out all of these fns and upload

OD values

Inverse voltage, high voltage means lower []

Target channel: 0 is channel 1, 1 is channel 2

Target gain

You scan over different gain to see if you get a signal

Scan over different gain tog et a higher dynamic range

Gain has 6 different channels you can hcoose to set threshold

Targe\_fluor[z] = (200, 200) this is the syntax for setting the threshold for 2 colors

* 1. Python
     1. Twocolors.py
        1. Run this code once the Arduino code is going.
        2. Once you’ve uploaded your code you want to run two\_colors.py
        3. Whenever you run this code it stores data in a folder with the date as its name.
     2. Analyze.py
        1. This code allows you to combine the data from several different runs (for example if you switched modes during a run)
        2. You will need to change these variables
           1. ﻿list\_of\_files – a list of strings containing the file names of the

# Calibrating the Device

# Preparing a Run

You may choose to perform a run in batch mode (no media flow or pumps) or in continuous culture mode.

**Batch Culture**

**Continuous Culture**

* 1. Prepare a bottle of your desired media
     1. If your media has elements that cannot be autoclaved, that is ok! You can use the input valve in the media cap to add these elements afterward
  2. Connect all the tubing before autoclaving
  3. attach the caps for the pumps to the wires
  4. Make sure all the inputs have a cap screwed on
  5. Tie off the media line with autoclave tap to keep the media from traveling through the lines during autoclaving

Put vial in the bottle to autoclave

Can use a syringe to flush out water after autoclaving

# Starting a Run

Arduino

Python

# Killing a Run

* 1. First kill the python code by hitting ctrl+c in the terminal
  2. In Arduino scroll to the bottom and comment out all the commands, then upload

# Clean Up

* 1. Pump the lines with ethanol, then DI water.
     1. Don’t use bleach, some of the parts are stainless steel and will rust.

Junctions for pump are asymmetrical

Put probe in alcohol then put it in a fresh tube

Cleaning- rinse with tap water then rinse with DI water once

Flush all the tubes

# Data and Analysis

Make sure the gasket in the cap is flat

Data stored in dropbox