Testing of Low Cost, Remote Data Acquisition Platform for In Situ Biological and Fluidic Measurements

Rachel Zaltz
Department of Computer Engineering
The George Washington University
Washington, DC,
United States of America
rachelzaltz@gwu.edu

Connor Kraft
Department of Computer Engineering
The George Washington University
Washington, DC,
United States of America
connorkraft3@gwu.edu

Phillip Jones
Department of Electrical Engineering
The George Washington University
Washington, DC,
United States of America
phillipjones116@gwu.edu

***Abstract*—A low-cost data acquisition platform will be designed and assembled to increase the ease of dynamic in situ data collection from confined isolated environments such as incubators. The aim of this project is to record humidity, temperature, PH, turbidity, time-stamped imaging, and various other sensor data at user-specified intervals. The device will remain physically independent of the space outside the incubator by using a modular power supply and network connection. The device will run independently of a power supply for the duration of the experiment and wirelessly send the desired data to a remote user. The device will advance research in cellular and polymetric interactions with indirect measurement and imaging in incubators.**

I. INTRODUCTION:

The low-cost data-acquisition platform will be designed to increase dynamic in situ data collection in confined and isolated environments. Currently, when an experiment is performed within an incubator for a long period of time, lab technicians are unable to open the confined environment until the experiment has been completed in its entirety. This results in a loss of crucial data points as the experiment is progressing within the incubator. Currently, the data cannot be obtained during the duration of the experiment and this results in the loss of many crucial data points. A low-cost data acquisition platform would increase data collection in in situ environments.

Currently, there are only a few products that solve the issue at hand. Some of the best products are from Dickson Data, such as the DicksonOne. These products only allow for a small set of sensors and lack more advanced sensor technology and cameras. They are also very expensive, with the cheapest model being $350. Each individual sensor costs over 100 dollars and requires a paid subscription to use some of the more advanced features. Overall, a product does not exist that meets all of the design requirements for this task.

This project looks to implement a system which can gather data within a closed incubator and send that data to a user on any remote computer. The data is gathered from multiple sensors including temperature, humidity, pH, turbidity, and time stamped images. This data is gathered at a time interval which can be specified by the user to allow for greater flexibility of the system. When data is not being gathered, sensors are turned off in order to save power and conserve energy. This data is then sent to a remote server using a Wifi connection so that it may be viewed by the user.

II. THEORY OF OPERATION

The goal of the project is to design a system which can gather data in a closed environment and send this data wirelessly to a remote user. The data gathered is specified by the sensors and the system is able to do this with only one input from the user, the data collection rate (every 30 seconds), and how long to gather data for (12 hours). These inputs and outputs are outlined in Figure I below.

At the current time, gathered data is collected and stored in labeled local directories on the Raspberry Pi. All relevant timestamps, sensor data, and image names are then queued by Redis and transferred to a SQL Server database running on the server. For the sake of redundancy, all relevant Raspberry Pi local files are then synced to a directory on this server.

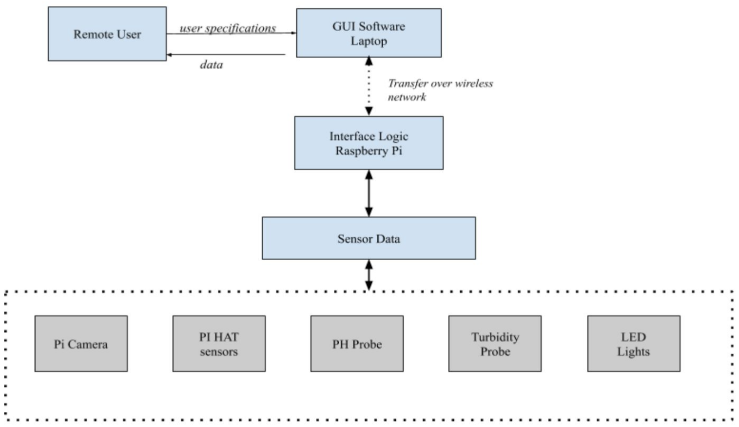


Figure I: Context Level Diagram

III. TESTING METHODS

This section outlines the testing methods that were used to verify the performance of the Remote Data Acquisition Platform in accordance with the set specifications. The purpose of these tests is to measure the end-to-end performance of the device, and to identify any constraints in operation. Due to the current Coronavirus outbreak, several of the designed tests were not able to be run; this will be discussed below in individual testing sections and in the breakdown of specifications. Please note, the result of this is that several aspects of the specifications were not able to be verified at the current time.

A. Raspberry Pi and Sensors

The most comprehensive testing option available to assess the behavior and operation of the project's data pipeline was to monitor system performance during fully-fledged application runs. Three tests were designed to exhibit system characteristics during what can be considered as a typical experimental operation.

The tests are as follows: 1 hour testing run with

360 samples (10 second poll interval), 2 hour 30 minute testing run with 50 samples (3 minute poll interval), and 8 hour 20 minute testing run with 100 samples (5 minute poll interval).

At each polling interval, system diagnostics will be gathered by recording the CPU usage, CPU temperature, and RAM usage. Due to the testing constraints, this test is largely qualitative, but provides key information about system health and operation. Each of the three tests will be run twice, to form an idea of system performance while idling and while running the designed application. The qualifications desired when looking at the overall comparison of dry versus loaded test runs are that the system stays under 85 degrees Celsius (particularly 80 degrees Celsius, as this is where thermal throttling begins), and that only marginal increases in average CPU and RAM usage occur going from dry to loaded test versions. Essentially, this testing acts as verification that the system performs as expected, and that the designed application does not overload the system in any way.

B. User Interface

The purpose of this test was to ensure the front end was properly connected to the server and there were no malfunctions. This test mimicked an experimental operation of a 1 hour testing run with 360 samples (10 second poll interval) and an 8 hour 20 minute testing run with 100 samples (5 minute poll interval). The purpose of the test was to ensure all data appeared in the front end GUI and no connection or data was Lost in the process.

C. Power Management

The purpose of this test was to estimate the power draw of the entire system to select a battery that would be able to last the entire length of an experiment. The test minicked an experiment lasting 4 weeks sampling once every minute in one experiment and sampling once every hour in the other.

TABLE I:
REQUIREMENTS AND SPECIFICATIONS

System Level Requirements:	System Level Specifications:	Test Results:
Operates with a lithium ion battery pack.	The device shall be able to run on a lithium ion battery pack.	Battery was not purchased due to COVID-19 impediments.
Have a long battery life so the device does not have to be removed from the incubator.	The device shall be able to last on the battery pack for the duration of the experiment (minimum 1 day - maximum 4 weeks).	Measurements were calculated (see results section in report).
Take pictures of the test environment.	A Pi camera must be used to take pictures of the test environment at user specified intervals.	This requirement was met (see code).
Have a light source for the test environment.	The device must conserve battery by only using the lights when needed (e.g., taking a picture) or when the user specifies.	This requirement was met (see code).
Conserve battery with 2 different modes: sleep and active.	The device shall be in sleep mode when measurements and images are not being taken.	Though this was not formalized, this was verified by testing in experiment-like circumstances.
Measure pH.	The device shall be able to measure the pH of the solution from 0 to 14 at +/- 0.001 increments using a pH probe, with an accuracy of +/- 0.002.	This is confirmed by the data sheet found here.
Measure humidity and	The device shall be able to	This is confirmed by the data

temperature.	measure humidity from 20 to 80% with an accuracy of +/- 5% and temperature from 0 to 50 degrees C with an accuracy of +/- 0.2 degrees C using an Adafruit DHT11/22 sensor.	sheet found here.
Measure turbidity.	A turbidity probe shall be used to measure the amount of light scattered by the solids in the solution.	Not applicable, this sensor was not added to the array.
Measure electrical conductivity.	A conductivity probe shall be used to measure the electrical conductivity of the solution from 5 to 200,000 micro-S/cm, with an accuracy of +/- 2%.	This is confirmed by the data sheet found here.
Be accessible to a remote user.	The device shall send data to a GUI that can be accessed remotely through a desktop and mobile application.	The data is available to a remote use if they have the code running on their laptop. The code was not deployed to the server (due to COVID-19).
Communication will be enabled via Wifi.	The device should be able to send data to a computer via the built in Wifi module on the Raspberry Pi 3+.	This requirement was met and demonstrated in the presentation.

IV. TEST RESULTS

I. Raspberry Pi and Sensors

At the very least, evidence in comparison between the dry and loaded run test qualitatively shows that the designed
application succeeds in its goal of being low impact. Through these tests, it can be definitively noted that the data pipeline performs exactly as suspected and that the application has little impact on system performance over time.
A short examination of the test results is as follows (all values are averages taken over the length of the experiment):

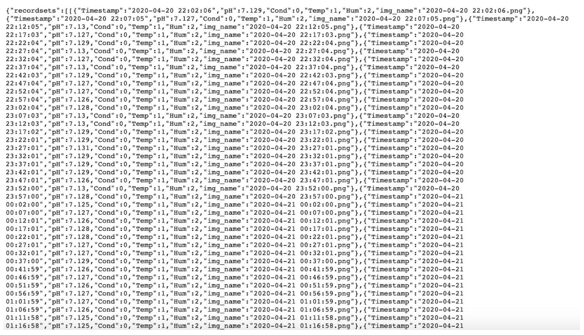
- CPU temperature - 53.9 degrees Celsius
- CPU usage - 24.5 %
- RAM usage - 52.3 %
- 2 hour, 30 minute test (50 samples)
  - CPU temperature - 54.9 degrees Celsius
  - CPU usage - 25.8 %
  - RAM usage - 52.6 %
- 8 hour, 20 minute test (100 samples)
  - CPU temperature - 54.5 degrees Celsius
  - CPU usage - 24.6 %

- Dry test results
  - 1 hour test (360 samples)
    - RAM usage - 52.4 %
- Loaded test results
  - 1 hour test
    - CPU temperature - 56.9 degrees Celsius
    - CPU usage - 26.6 %
    - RAM usage - 74.1 %
  - 2 hour, 30 minute test
    - CPU temperature - 56.6 degrees Celsius
    - CPU usage - 26.9 %
    - RAM usage - 71.7 %
  - 8 hour, 20 minute test
    - CPU temperature - 58.4 degrees Celsius
    - CPU usage - 28.2 %
    - RAM usage - 72.9 %

Overall, even between short and long-term testing periods, the system shows expected changes when going from the dry to loaded tests. Average CPU usage and temperature experience a marginal increase when the system is under load, but nowhere near alarming values. CPU temperatures stay well below operational limits and do not come close to thermal throttling, in either case. Of note is the 20 % or so increase in RAM usage while under load, which is expected due to the implementation of the Redis queue in this project. This data convincingly verifies appropriate system performance under experiment-like load.

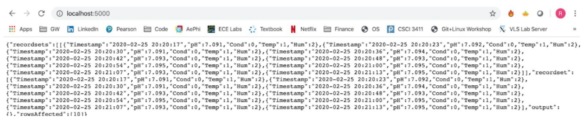
II. User Interface

The front end test was done for a 1 hour test and an 8 hour experiment. The results below show that data correctly flowed through the pipeline and was received on the front end side of the code.



```
{"recordsets":[[{"Timestamp":"2020-04-20 22:02:06","pH":7.129,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-20 22:02:06.png"},
{"Timestamp":"2020-04-20 22:07:05","pH":7.127,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-20 22:07:05.png"},{"Timestamp":"2020-04-20
22:12:05","pH":7.13,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-20 22:12:05.png"},{"Timestamp":"2020-04-20
22:17:03","pH":7.127,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-20 22:17:03.png"},{"Timestamp":"2020-04-20
22:22:04","pH":7.129,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-20 22:22:04.png"},{"Timestamp":"2020-04-20
22:27:04","pH":7.13,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-20 22:27:04.png"},{"Timestamp":"2020-04-20
22:32:04","pH":7.127,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-20 22:32:04.png"},{"Timestamp":"2020-04-20
22:37:04","pH":7.13,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-20 22:37:04.png"},{"Timestamp":"2020-04-20
22:42:03","pH":7.129,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-20 22:42:03.png"},{"Timestamp":"2020-04-20
22:47:04","pH":7.127,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-20 22:47:04.png"},{"Timestamp":"2020-04-20
22:52:04","pH":7.127,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-20 22:52:04.png"},{"Timestamp":"2020-04-20
22:57:04","pH":7.126,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-20 22:57:04.png"},{"Timestamp":"2020-04-20
23:02:04","pH":7.128,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-20 23:02:04.png"},{"Timestamp":"2020-04-20
23:07:03","pH":7.13,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-20 23:07:03.png"},{"Timestamp":"2020-04-20
23:12:03","pH":7.13,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-20 23:12:03.png"},{"Timestamp":"2020-04-20
23:17:02","pH":7.129,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-20 23:17:02.png"},{"Timestamp":"2020-04-20
23:22:01","pH":7.129,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-20 23:22:01.png"},{"Timestamp":"2020-04-20
23:27:01","pH":7.131,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-20 23:27:01.png"},{"Timestamp":"2020-04-20
23:32:01","pH":7.129,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-20 23:32:01.png"},{"Timestamp":"2020-04-20
23:37:01","pH":7.129,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-20 23:37:01.png"},{"Timestamp":"2020-04-20
23:42:01","pH":7.129,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-20 23:42:01.png"},{"Timestamp":"2020-04-20
23:47:01","pH":7.126,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-20 23:47:01.png"},{"Timestamp":"2020-04-20
23:52:00","pH":7.13,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-20 23:52:00.png"},{"Timestamp":"2020-04-20
23:57:00","pH":7.128,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-20 23:57:00.png"},{"Timestamp":"2020-04-21
00:02:00","pH":7.125,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-21 00:02:00.png"},{"Timestamp":"2020-04-21
00:07:00","pH":7.127,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-21 00:07:00.png"},{"Timestamp":"2020-04-21
00:12:01","pH":7.126,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-21 00:12:01.png"},{"Timestamp":"2020-04-21
00:17:01","pH":7.128,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-21 00:17:01.png"},{"Timestamp":"2020-04-21
00:22:01","pH":7.128,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-21 00:22:01.png"},{"Timestamp":"2020-04-21
00:27:01","pH":7.127,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-21 00:27:01.png"},{"Timestamp":"2020-04-21
00:32:01","pH":7.127,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-21 00:32:01.png"},{"Timestamp":"2020-04-21
00:37:00","pH":7.129,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-21 00:37:00.png"},{"Timestamp":"2020-04-21
00:41:59","pH":7.126,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-21 00:41:59.png"},{"Timestamp":"2020-04-21
00:46:59","pH":7.127,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-21 00:46:59.png"},{"Timestamp":"2020-04-21
00:51:59","pH":7.126,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-21 00:51:59.png"},{"Timestamp":"2020-04-21
00:56:59","pH":7.127,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-21 00:56:59.png"},{"Timestamp":"2020-04-21
01:01:59","pH":7.127,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-21 01:01:59.png"},{"Timestamp":"2020-04-21
01:06:59","pH":7.126,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-21 01:06:59.png"},{"Timestamp":"2020-04-21
01:11:58","pH":7.125,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-21 01:11:58.png"},{"Timestamp":"2020-04-21
01:16:58","pH":7.125,"Cond":0,"Temp":1,"Hum":2,"img_name":"2020-04-21 01:16:58.png"},{"Timestamp":"2020-04-21
```

Figure II: Front End of 8 Hour Test

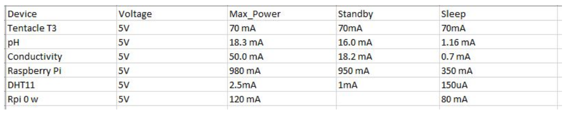


```
{"recordsets":[[{"Timestamp":"2020-02-25 20:20:17","pH":7.091,"Cond":0,"Temp":1,"Hum":2},{"Timestamp":"2020-02-25 20:20:23","pH":7.092,"Cond":0,"Temp":1,"Hum":2},
{"Timestamp":"2020-02-25 20:20:30","pH":7.091,"Cond":0,"Temp":1,"Hum":2},{"Timestamp":"2020-02-25 20:20:36","pH":7.094,"Cond":0,"Temp":1,"Hum":2},
{"Timestamp":"2020-02-25 20:20:42","pH":7.093,"Cond":0,"Temp":1,"Hum":2},{"Timestamp":"2020-02-25 20:20:48","pH":7.093,"Cond":0,"Temp":1,"Hum":2},
{"Timestamp":"2020-02-25 20:20:54","pH":7.095,"Cond":0,"Temp":1,"Hum":2},{"Timestamp":"2020-02-25 20:21:00","pH":7.095,"Cond":0,"Temp":1,"Hum":2},
{"Timestamp":"2020-02-25 20:21:07","pH":7.093,"Cond":0,"Temp":1,"Hum":2},{"Timestamp":"2020-02-25 20:21:13","pH":7.095,"Cond":0,"Temp":1,"Hum":2}]],"recordset":
[{"Timestamp":"2020-02-25 20:20:17","pH":7.091,"Cond":0,"Temp":1,"Hum":2},{"Timestamp":"2020-02-25 20:20:23","pH":7.092,"Cond":0,"Temp":1,"Hum":2},
{"Timestamp":"2020-02-25 20:20:30","pH":7.091,"Cond":0,"Temp":1,"Hum":2},{"Timestamp":"2020-02-25 20:20:36","pH":7.094,"Cond":0,"Temp":1,"Hum":2},
{"Timestamp":"2020-02-25 20:20:42","pH":7.093,"Cond":0,"Temp":1,"Hum":2},{"Timestamp":"2020-02-25 20:20:48","pH":7.093,"Cond":0,"Temp":1,"Hum":2},
{"Timestamp":"2020-02-25 20:20:54","pH":7.095,"Cond":0,"Temp":1,"Hum":2},{"Timestamp":"2020-02-25 20:21:00","pH":7.095,"Cond":0,"Temp":1,"Hum":2},
{"Timestamp":"2020-02-25 20:21:07","pH":7.093,"Cond":0,"Temp":1,"Hum":2},{"Timestamp":"2020-02-25 20:21:13","pH":7.095,"Cond":0,"Temp":1,"Hum":2}],"output":
{},"rowsAffected":[10]}
```

Figure III: Front End of 1 Hour Test

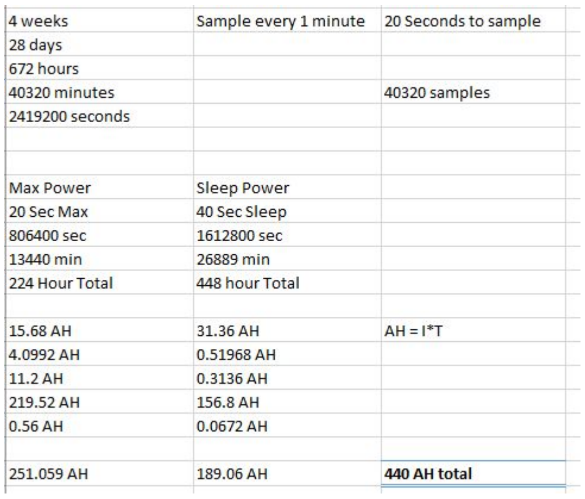
III. Power Management

The power draw test was estimated for 4 weeks taking data either every minute or every hour. The results show that a 440 Amp Hour battery is needed in order to last while collecting data every minute and a 290 Amp Hour battery is needed in order to last while collecting data every hour.



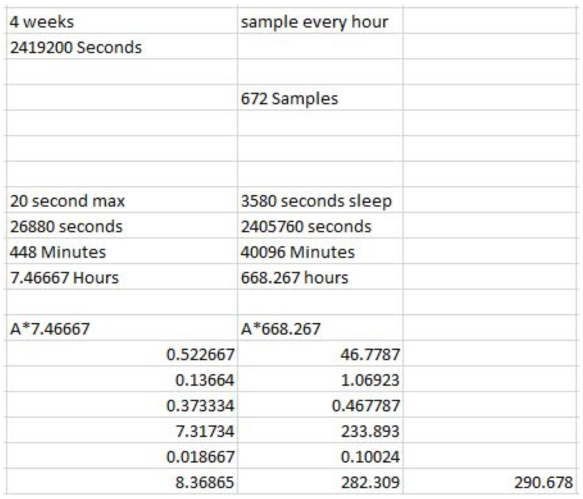
Device	Voltage	Max_Power	Standby	Sleep
Tentacle T3	5V	70 mA	70mA	70mA
pH	5V	18.3 mA	16.0 mA	1.16 mA
Conductivity	5V	50.0 mA	18.2 mA	0.7 mA
Raspberry Pi	5V	980 mA	950 mA	350 mA
DHT11	5V	2.5mA	1mA	150uA
Rpi 0 w	5V	120 mA		80 mA

Figure IV: Power Draw of Components



4 weeks	Sample every 1 minute	20 Seconds to sample
28 days		
672 hours		
40320 minutes		40320 samples
2419200 seconds		
Max Power	Sleep Power	
20 Sec Max	40 Sec Sleep	
806400 sec	1612800 sec	
13440 min	26889 min	
224 Hour Total	448 hour Total	
15.68 AH	31.36 AH	AH = I*T
4.0992 AH	0.51968 AH	
11.2 AH	0.3136 AH	
219.52 AH	156.8 AH	
0.56 AH	0.0672 AH	
251.059 AH	189.06 AH	**440 AH total**

Figure V: Sampling every minute



4 weeks	sample every hour	
2419200 Seconds		
	672 Samples	
20 second max	3580 seconds sleep	
26880 seconds	2405760 seconds	
448 Minutes	40096 Minutes	
7.46667 Hours	668.267 hours	
A*7.46667	A*668.267	
0.522667	46.7787	
0.13664	1.06923	
0.373334	0.467787	
7.31734	233.893	
0.018667	0.10024	
8.36865	282.309	290.678

Figure VI: Sampling every Hour

ACKNOWLEDGEMENT
This project would not be in its current state without the help of Professor Aslani and Professor Bulusuu
Additionally, the project was supported by the following departments: Electrical and Computer Engineering Department, Biofluid Dynamics Laboratory (PI: Prof. M. W. Plesniak), Department of Mechanical and Aerospace Engineering, and the Tissue Engineering Laboratory (PI: Prof. L. –G. Zhang), Department of Mechanical and Aerospace Engineering.

REFERENCES

Conductivity Probe K10 Datasheet. (n.d.). Retrieved from https://www.atlas-scientific.com/_files/_datasheets/_probe/EC_K_10_probe.pdf
Lab Grade pH Probe Datasheet. (n.d.). Retrieved from https://www.atlas-scientific.com/_files/_datasheets/_probe/pH_probe.pdf
Digital Output Relative Humidity & Temperature Sensor/Module Datasheet. (n.d.). Retrieved from https://cdn-shop.adafruit.com/datasheets/DHT22.pdf