Department of Computer Science and Engineering

Seattle Pacific University

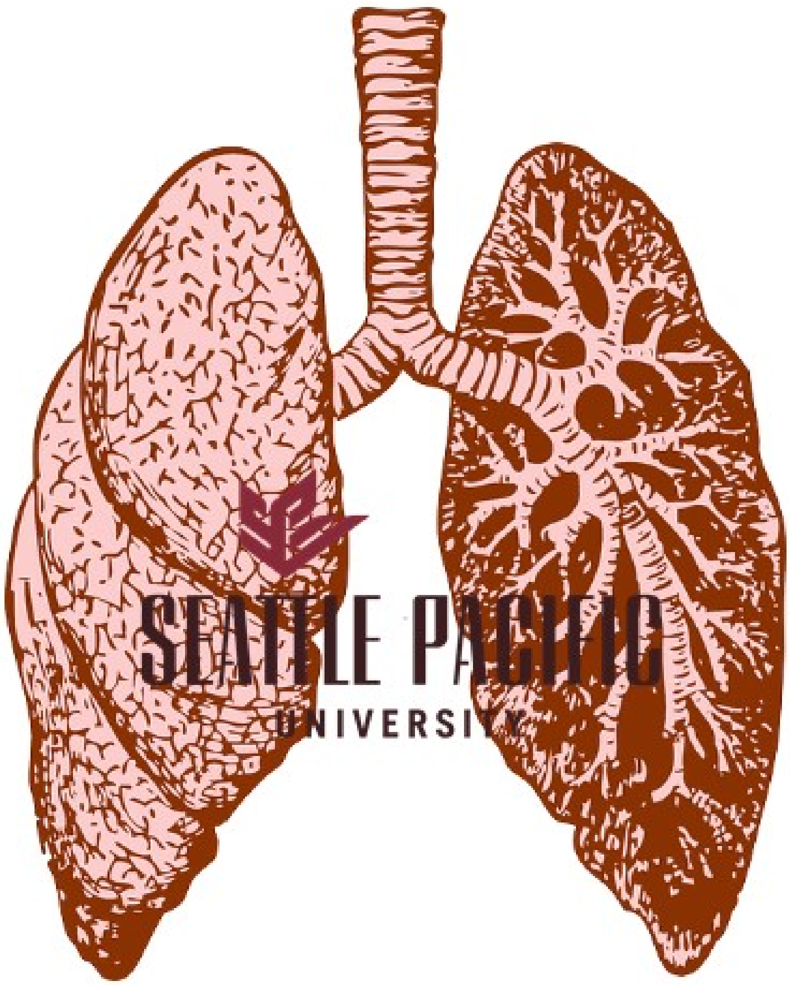
U-Catalyst: Senior Design Project

Mission and Vision Statement

Project – The Second Wind

Authors:
Nate Geller
Jason Smith
Makana Dang
Omar Garibay
Saihou Jobe

Supervisors:
Dr. James Walker
Dr. Lin Liu
Customer:
Kerry M. Curran – *Lung Technologies, LLC*



October 10th, 2021

TABLE OF CONTENTS

Table of Contents:

EXECUTIVE SUMMARY

The Incentive Spirometer 2.0 is a re-imagination of the current line of incentive spirometers, with the ability to record whether the user/patient is successfully following their prescribed therapy. With ergonomic and aesthetic redesigns, as well as optional OPEP integration, the Incentive Spirometer 2.0 will be a more comfortable and versatile device than the current models.

Target Customer and Market:

Team Second Wind is currently working with Lung Technologies LLC. to develop this spirometer. Our target customers are medical supply companies and health care providers. The target market would be Hospitals who want to ensure that their patients' lungs are fully recovering.

Risks Mitigated and Remaining:

The risks that Team Second Wind outlined at the start of this quarter have mostly been successfully reduced through their Risk Reduction Prototypes. These risks included: the ability to record when the piston in the diaphragm crosses the target volume; the integration of the OPEP device by having the ability to switch the air-flow direction within the device; the ability the audibly indicate/remind the patient to use to the device; and finally, to implement the previous changes without notably increasing the size of the incentive spirometer. Based on the RRP analysis conducted by the team, most of these risks have been greatly reduced. The only remaining risk that has not been fully addressed is the audible reminder for the patient to use the device.

Recommended Path Forward:

In the next two quarters Team Second Winds project intends to continue developing the Incentive Spirometer 2.0 as outlined in DR 1.1 with only some minor changes. The first main difference between our current path and the one outlined in DR 1.1 is that we will no longer be relying on SOLIDWORKS flow simulation to evaluate the OPEP and Incentive Spirometer, we will instead be exploring other options for flow. The second main difference is that we will be deciding between a laser, and proximity detection method in order to sense if the patient has reached their prescribed volume.

Items to address the first week of winter quarter:

The key items to address in the first week of the winter quarter are to make a final decision on which sensor we will be moving forward with, which method of flow analysis we intend to use, and submit SLS purchasing proposal.

UPDATED QUAD CHART

Restoring Lung Capacity

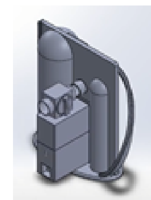
Team Second Wind: Jason Smith (ME), Saihou Jobe (CPE), Makana Dang (EE),
Nathaniel Geller (ME), Omar Garibay (ME)

Objective:

- To create a more incentivized and engaging device that allows doctors to track a patient's progress while also providing a secondary function that eliminates the need for a second device.
- Provide Lung Technologies in collaboration with Northwest Orthopedics in Spokane with a new, improved incentive spirometers.

Concept:

- Find optimized arrangement of components and features so that the product can be more beneficial and reliable, while also being cost competitive.



Approach:

- Redesign aesthetic features to improve the look
- Study existing OPEP technology and integrate into system
- Create a valve that directs air flow between the OPEP and incentive spirometer

<u>Projected Schedule and Budget</u>

Schedule:

- February - DR2.1: Subsystem Demonstration
- March - DR2.2: Prototype Demonstration
- May – DR3.1: Revised Prototype Demo and Test Results
- June – Final Product Demonstration.

Budget:

- Estimated Materials: $300
- SLS printer: $12,000

Total: $12,300

Risk Reduction Prototype

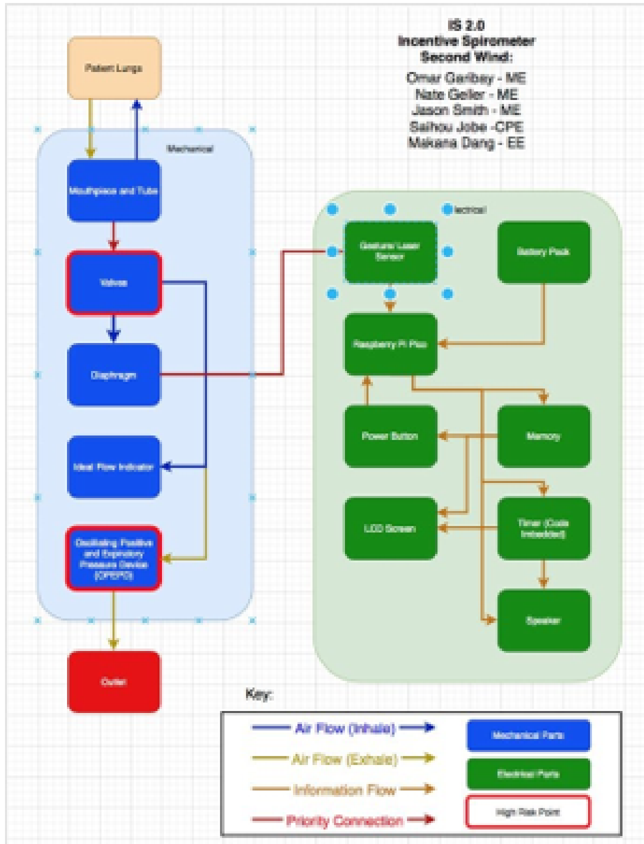


Figure 1: Block diagram of our incentive spirometer

At the start of this quarter team Second Wind identified multiple risks that would need to be mitigated in order for the project to succeed. These risks required mechanical, electrical and computer engineering solutions, which team Second Wind set out to resolve by the end of this quarter.

For the mechanical aspects, team Second Wind identified two critical paths crucial to the success of the Incentive Spirometer 2.0. These paths are reliant on two main components: the method and design for achieving Oscillating Positive Expiratory Pressure, and the seamless integration of the OPEP device into the spirometer. Kerry Curran, with Lung Technologies, also expressed an interest in both aesthetic and ergonomic redesign for the incentive spirometer.

On the electrical and computer engineering side, the most important aspect of the Incentive Spirometer 2.0 is the ability to detect and indicate when the piston has passed the prescribed volume. Mr. Curran also identified the importance of having a display system that shows the user and doctors how many completed inhalations were accomplished.

WHAT WE BUILT

For our RRP, we have completed 4 separate systems:

1. Incentive Spirometer and OPEP Housing – We designed a more aesthetic We researched and purchased two different OPEP devices with different mechanical methods of achieving oscillation of pressure. We will take the principles learned from these designs and tailor then to fit our needs. In terms of size and oscillation of pressure.
2. Inhalation and exhalation two-way valve – It is necessary for the user to be able to switch between using the incentive spirometer and the OPEP device. We wanted to minimize interference that these opposing systems would have on each other, so we designed a 2-way valve that met our needs. The valve is a simple construction of two parts that change the direction of air flow from the spirometer to the OPEP device.
3. Gesture sensor system / laser sensor system – we decided to make two different systems to capture the successful inhalation, each with their own benefits, to present to Mr. Curran for the final design decision. The first system connects a gesture/proximity sensor which utilizes IR detection to a microcontroller. The second system utilizes a laser emitter and light dependent resistor to a microcontroller.
4. Inhalation Display System – the microcontroller will record the successful inhalation and display it to the user through an LCD display. It will also send a signal to a speaker to sound in order to alert the user real time if they have successfully inhaled the prescribed volume.

On the mechanical side, the objective of our RRP was to familiarize ourselves with the complex feature of SOLIDWORKS and understand the concerns of plastic manufacturing design. The valve was our way of reconciling two different systems that rely on opposite directions of air flow. The mechanical engineers on the team have become more skillful and adept at modelling in SOLIDWORKS and new modelling/editing previous iterations will have a faster turnover rate.

On the electrical side, the objective of the RRP was to teach the electrical and computer-engineering group how to work with and integrate sensors with the microcontroller.

SPECIFICATION 1

Spec ID	Requirement	Threshold	Objective	Observed
RRP001	Housing for Spirometer and OPEP Device	Mr. Currans approval	Ergonomic and Aesthetic improvements to Incentive Spirometer	Pass

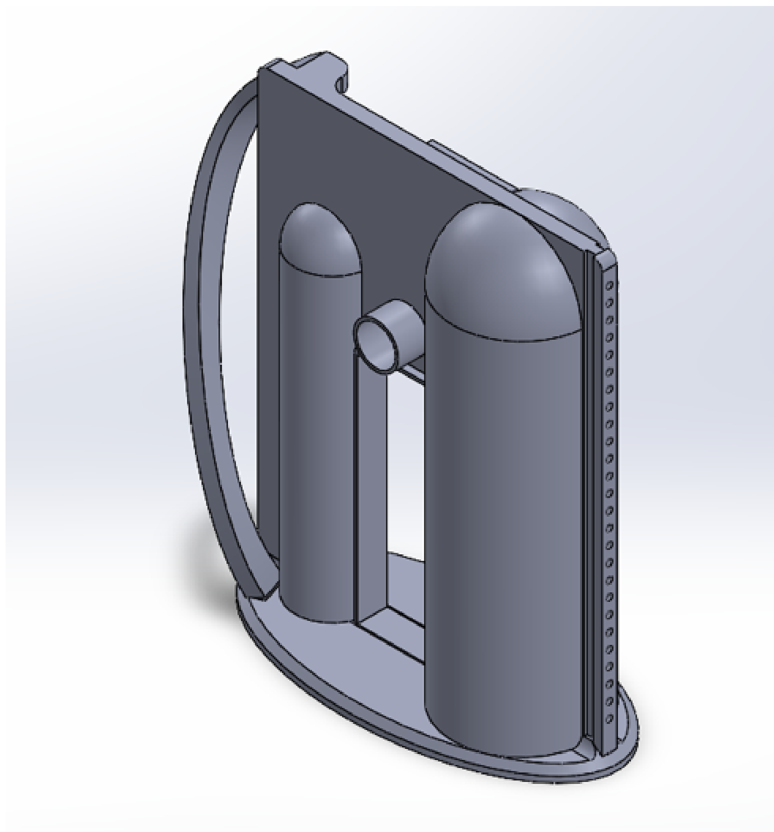


Figure 2: Incentive spirometer model

Our first RRP is one that we added at the behest of Mr. Curran, who expressed interest in having us redesign the incentive spirometer so that it could not only integrate an OPEP device and have good ergonomics, but also have improved aesthetics with a more modern design. We accomplished the first incentive spirometer design portion of this RRP by introducing a more modern design, and having the handle angled so that the user/patient would not tilt the device while completing their prescribed therapy. We accomplished the OPEP portion of this RRP by creating a space in-between the diaphragm and ideal indicator, where the OPEP housing can sit. This figure is the model that we presented to Mr. Curran and that he approved of.

SPECIFICATION 2

Spec ID	Requirement	Threshold	Objective	Observed
RRP002	Simple transition from spirometer to OPEP	N/A	Demonstrates ability to switch direction air flow within the device	Successfully channeled airflow away from the spirometer through 3D printed

Our second Risk Reduction Prototype was to develop a valve system that would allow for the patient/user to control the direction of flow. The importance of this RRP is that we need to ensure that integrating and implementing the OPEP device does not affect the flow or volumetric measurements of the incentive spirometer. We approached this RRP by developing a simple two-way valve with a rotating door that can close and prevent airflow to either the Spirometer or the OPEP device. We tested this RRP by 3-D printing our model and attaching it to our spirometer. We then tested how accurately we could control the flow of air by rotating the door and performing an inhalation and exhalation as seeing there was any airflow leakage when the door was open or closed.

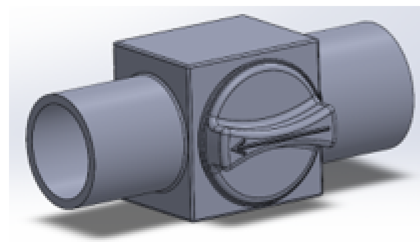


Figure 3: OPEP valve integration

This figure depicts our team testing the valves' ability to control airflow. The white piece between the two blue tubes is our valve. One of our mechanical engineers, Nate Geller, is holding a piece of paper as another mechanical engineer, Jason Smith, performs both an inhalation and exhalation with door, and then with the door closed. The test showed that we could control the flow of air but lost some air flow to the environment. We believe that this is most likely due to the roughness from 3-D printing.

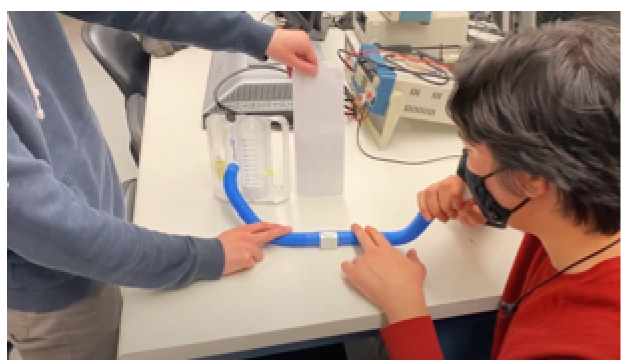


Figure 4: OPEP valve RRP verification

SPECIFICATION 3

Spec ID	Requirement	Threshold	Objective	Observed
RRP003.1	Readings off our Gesture/Proximity sensor	Pass/Fail	Accurately record successful inhalations	Readings from the gesture/proximity sensor displayed on code from microcontroller

The third specification for our RRP involved a sensor to tell whether or not the user achieved their prescribed volume. In order to collect this data to the best of our ability, we decided to test two different sensors with different benefits for both. This final decision of the sensor selection will be done by Mr. Curran after these tests. The two different sensors we used were a gesture/proximity sensor and a laser sensor.

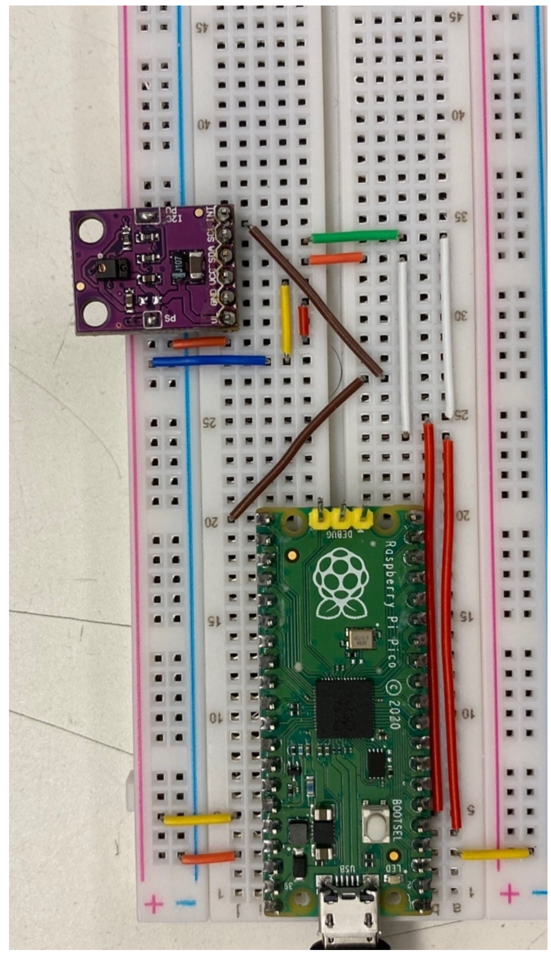


Figure 5: Gesture sensor connected to Raspberry Pi Pico

The gesture/proximity sensor shall sense the piston rising in the diaphragm of the spirometer by the user's inhalation. The APDS-9960, the sensor selected, utilizes an infrared-emitting diode with four different photodiodes that interpret the amount of IR light received in order to indicate from which the direction of light was intensified. We decided to use three different tests to test this specification. Since we weren't too familiar with how the sensor worked, the first test was just moving an object in front of the sensor from different directions, interpreting the directional output of the sensor through the microcontroller. In order to simulate the sensor's situation with the spirometer and it's rising piston, we secondly tested the sensor with the frosted plastic from the spirometer. This was to tell whether or not the sensor would be able to sense the direction of the piston through the plastic, testing our worries of the light bouncing off the plastic rather than the actual rising piston from the user's inhalation. Lastly, we put our sensor to the test with the actual spirometer, checking whether or not the APDS-9960 would be able to read the rising of the pin by the inhalation of an actual user. All of our tests were deemed successful or unsuccessful through code output on Thonny from the microcontroller. The electrical configuration of the sensor and the microcontroller can be seen in **Figure 5**.

The threshold for this specification was mostly successful, as we were accurately able to sense objects passing the sensor with and without the frosted glass from the housing, but not able to sense the actual piston of the spirometer while a user was inhaling. The tests and their results are shown in **Figures 6, 7, and 8.** While we are currently not able to sense the piston rising, through manipulation of the library file of the gesture sensor module provided by Adafruit, along with the insight gained through both successful and unsuccessful testing, we can confidently say we will be able to implement this system of a gesture sensor and Raspberry Pi Pico for our inhalation detection.

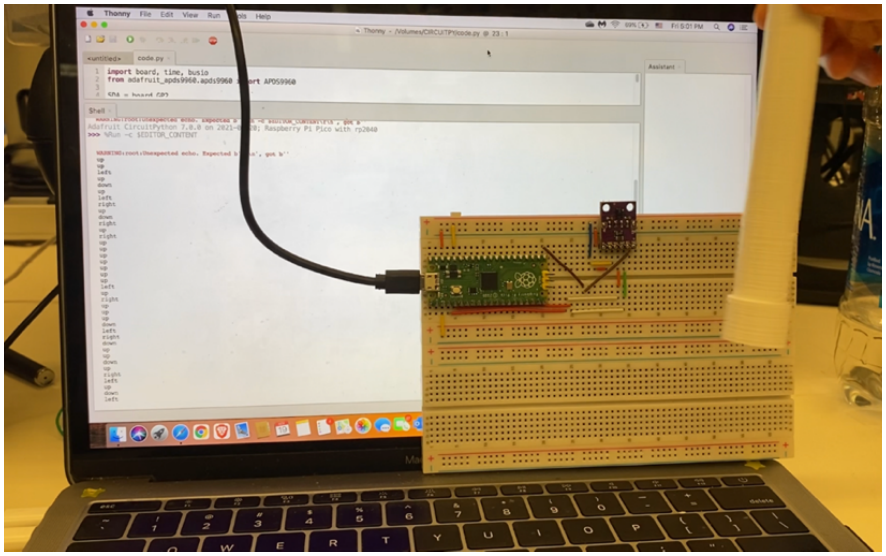


Figure 6: Testing and readings from first test

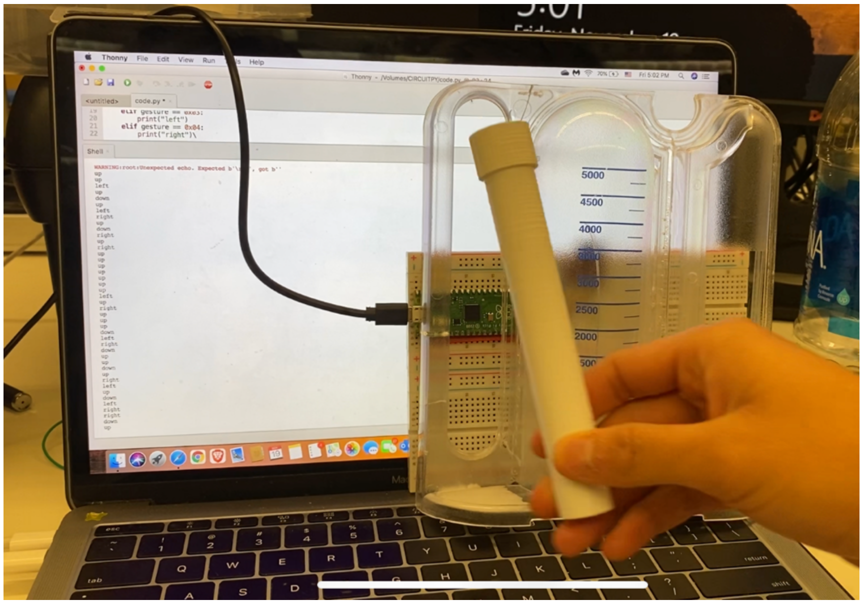


Figure 7: Testing and readings from second test

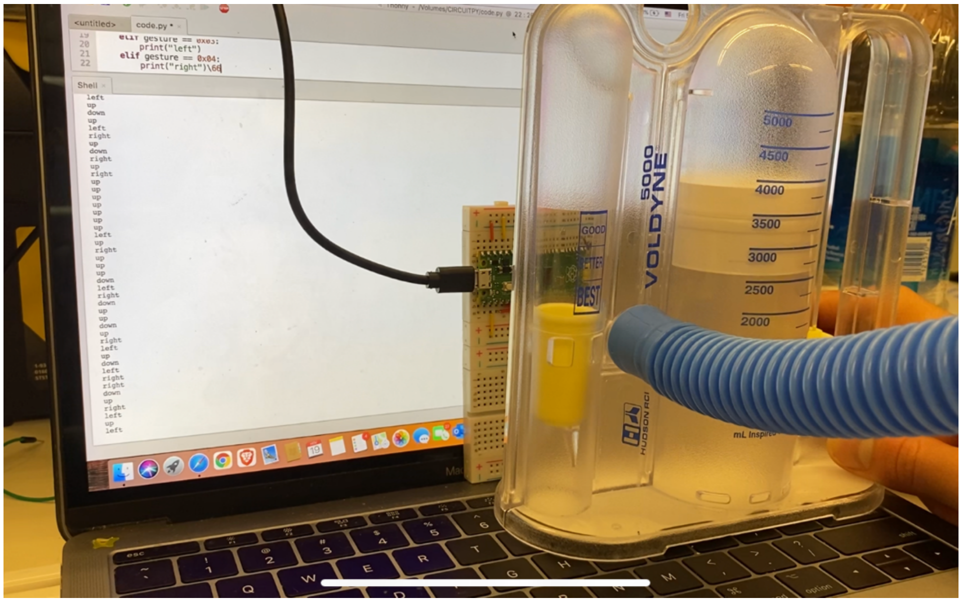


Figure 8: Testing and readings from third test

Spec ID	Requirement	Threshold	Objective	Observed
RRP003.2	Readings of our Light Dependent Resistor (LDR)	LDR can sense the laser shining on it or not	Determine LDR values to output message and trigger buzzer	Readings from the LDR displaying on screen and an audible alert by the buzzer

The third specification involved the LDR. The LDR shall sense whether the laser is shining on it or not. We tested it by outputting values of the LDR. The values change based on the resistance of the LDR. The resistance changes depending on the amount of light the LDR is exposed to. The LDR values are dependent on the LDR voltage. More lighting exposed to the LDR decreases resistance, and hence decreases LDR value. With less lighting, LDR resistance increases and so does the LDR value.

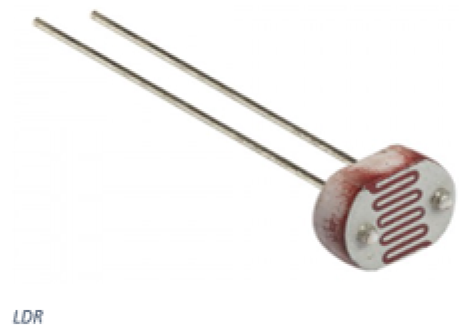


Figure 9: Light dependent resistor sensor

The threshold of specification was because once the laser was interrupted, simulating a successful inhalation exercise, the buzzer goes off, and the number of successful inhalations gets recorded and displayed on our LCD display.

For the laser sensor system, when the laser was interrupted by the diaphragm rising, the LCD shows a successful inhalation counting accompanied by an audible alert for confirmation as well.

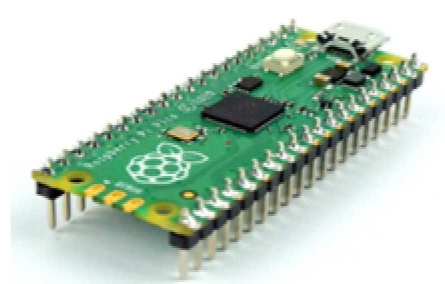


Figure 11: Raspberry PI Pico

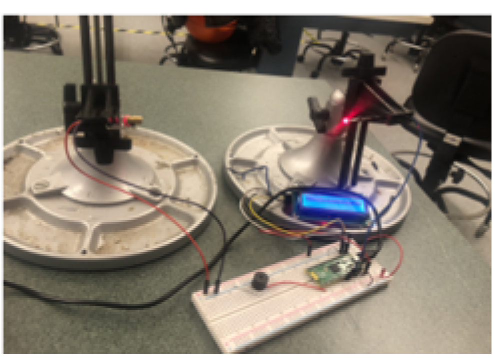


Figure 12: LDR laser sensor setup #1

The circuit was tested to verify that the LDR was not affected by ambient lighting of any room. The circuit successfully took accurate data in a dark as well as in a well-lit room. No ambient light interference was observed. The later goal is to make the LDR or code react quicker to the laser interruption. This is to mitigate a split of a second delay observed between the laser being interrupted to the Pico translating that data to the LCD display.

SPECIFICATION 4

Spec ID	Requirement	Threshold	Objective	Observed
RRP004	Inhalation Display System	Interpret data for patients and pulmonologists	Validate diaphragm level reached by patient	Interpret data to LCD display and an audible alert

The last specification for the RRP dealt with the interpretations of the data input from the sensors and how we would successfully display this to the user. The electrical system that we are adding to the spirometer shall be able to display to the user and the doctor the number of successful inhalations the user was able to achieve. In order to do so, we connected an LCD display to one of the GPIO pins of the Pico, which read the data that the microcontroller interpreted from the gesture and laser sensors. We also connected a small speaker that also indicated a successful inhalation with a slight buzzing sound. To test this, we connected the display and speaker to the Pico in the circuit configuration seen in **Figure 12; Figure 13**.

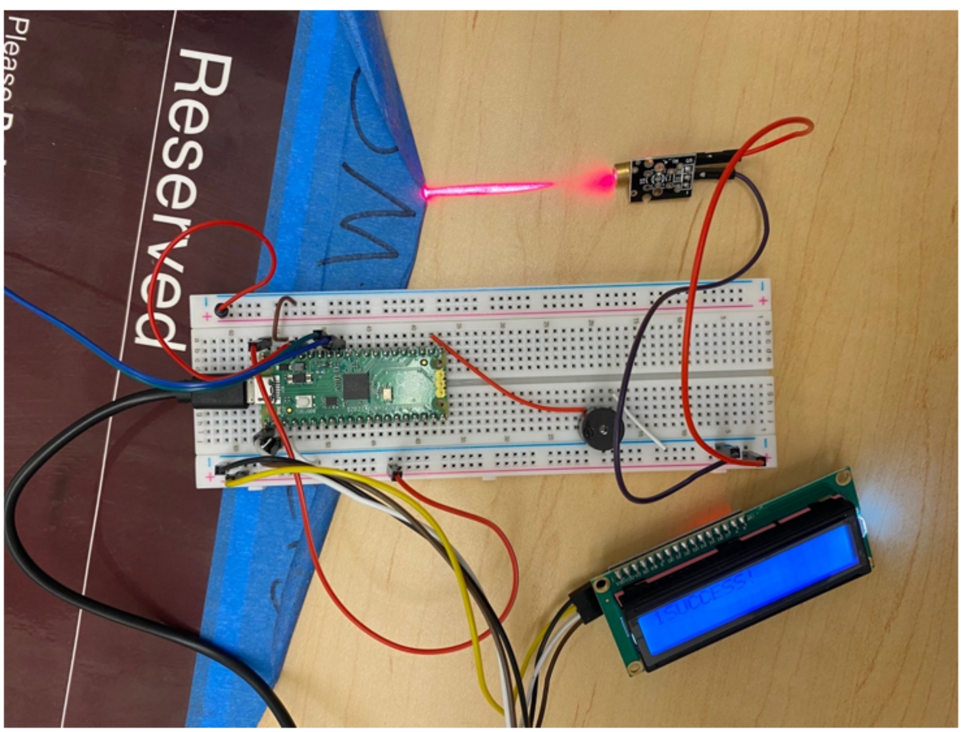


Figure 13: Laser Circuit Layout

While testing the laser sensor, we were successfully able to display the readings on the LCD display, shown in **Figure 14**. Since our testing of our LCD display code with the laser sensor was successful, we know that this can also be implemented to the gesture sensor, since the codes would be fairly similar. The only difference would be the language, since the laser circuit currently runs on Micro Python while the gesture sensor runs with the new Micro python variation, Circuit Python.

Figure 14: LDR Circuit Screen Output

While all the testing of the microcontroller was done with a connection to our laptops, we will need some sort of separate power source to power the microcontroller, based on the calculated load of the controller with varying microcontroller modes. This analysis can be seen later in the report.

MECHANICAL ENGINEERING ANALYSES

Per Mr. Currans' request, most of the mechanical aspects of the Incentive Spirometer 2.0 were based completely on previous iterations of incentive spirometers, meaning not much mechanical analysis was necessary, but now this will allow us to focus our attention to the OPEP module of the device. This new type of technology has studies supporting its effectiveness, but the mechanisms that allow for its beneficial effects are of little

SOLIDWORKS Flow Simulation: We had intended to use SOLIDWORKS flow simulation to ensure that our designs remained faithful to the airflow of other spirometers, and we had also planned to use this flow simulation to obtain data regarding the pressure and resistances in the OPEP device. However, SOLIDWORKS flow simulation ended up not being able to simulate what we needed. We cannot simulate the movement of parts that are driven by the fluid flow. The software is extremely capable of giving data on internal steady flow or the aerodynamics of shapes with their external flow studies, but they cannot facilitate the oscillatory flow that is essential for the validity of the OPEP device. This winter break, Team Second Wind will conduct further research into other forms of flow simulation as well as more experimental methods of obtaining data.

Design: A majority of the mechanical engineering's time was focused on developing aesthetic and ergonomic improvements. In addition to having a 'modern' look, Mr. Curran also requested that the be redesigned for ergonomics. The first step we took towards improving the ergonomics of the incentive spirometer was that we angled the handle towards the user. We noticed that when using the spirometer people would tend to tilt the device away from them, which made using the device more comfortable but also impacted the accuracy of the volume measurements. By angling the handle towards the user, the spirometer can be used comfortably without angling the diaphragm. In order to get a design that Mr. Curran was happy with, we discussed our design choices with him and implemented changes based on his Input. To add usability to the device, we raised the inhalation tube to have the ability to fit the OPEP module within the spirometer.

To begin designing the layout of the spirometer we first had to model the general layout of the current spirometer provided to us by Lung Technologies. Using this model, we could then explore other potential layouts and see the benefits and weaknesses of each one. The Overall dimensions of the spirometer are as follow; 7.5 in tall, 6.5 in long, 3.5 in wide, diaphragm diameter of 2.25 in, inhalation diameter of 0.25 in, and wall thickness ~1/8 in. On the right we see our first layout is very similar to the spirometer provided to us except for the inclusion of the pin's holes on the left of the device. These holes are meant to hold the adjustable electrical module which we will work on next quarter. The variety of holes allows us to move the sensor up and down depending on the patient's target inhalation.

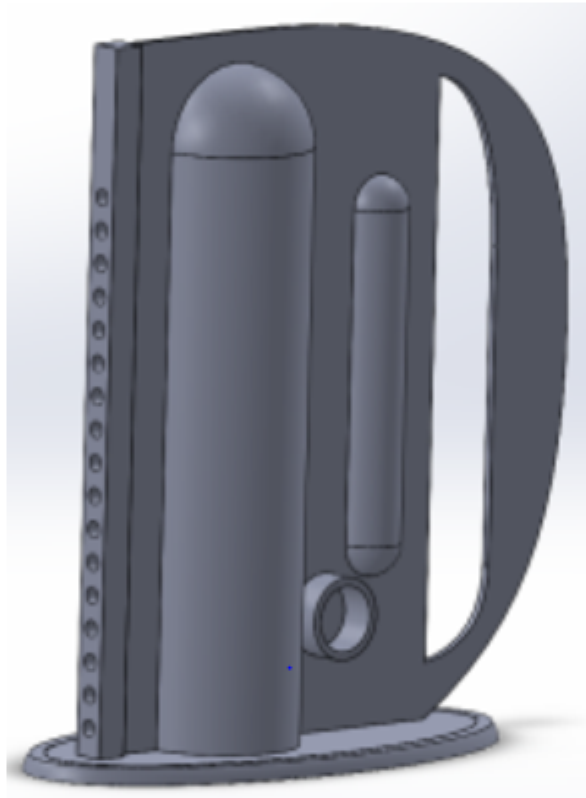


Figure 15: Reference spirometer model

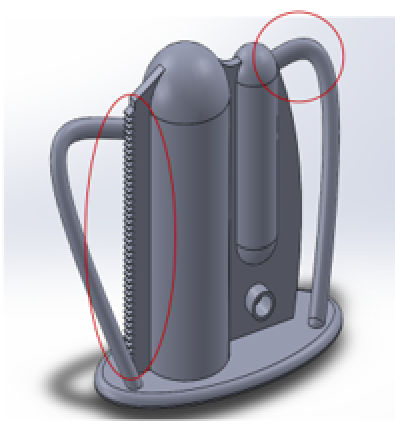


Figure 16: Initial ergonomic handle design

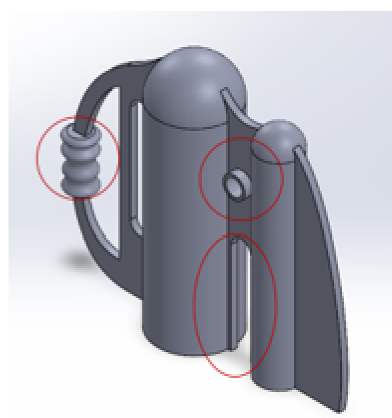


Figure 17: Additional layout of spirometer

We worked through multiple iterations to find a design that met what Lung technologies wanted us to deliver. Mr. Curran had mentioned to us that he would like to see a spiral shaped handle integrated into our design to modernize the spirometer. The implementation of curves, let alone 3 dimensional curves, are difficult to model and know their exact translation when fully fabricated, so some trial and error may be in order. The spiral handle was also difficult to integrate as we had to account for the fabrication of the device which involves molding the spirometer in halves and then putting them together.

Another major design aspect we took into consideration when making these models was the location of the electrical components and how to adjust them. The pin system was an idea given to us by Mr. Curran, this worked well for adjusting the sensor and was sturdy so we wouldn't have to worry about the safety of our electrical components. The weakness of the pin system is that it takes a little more effort than our group would like for the patient to adjust the sensor. To combat this, we have contemplated using a linear ratcheting mechanism to allow for more fluid motion. The weakness of this design is that it involves more moving parts which could increase the overall price of the device. Since the circuits that we will be implementing are still in the process of being finalized and we aren't sure of the overall size, we have set this part of the design to the side until we those dimensions.

OPEP Research: Extensive research also went into selecting a method of obtaining Oscillating Positive Expiratory Pressure. We mainly looked at two distinct types of OPEP designs, gravity-dependent OPEP devices, and gravity-independent OPEP devices. The two main gravity-dependent devices we researched were the RC-Cornet and Flutter OPEPS. The two types of gravity-independent devices are the Aerobika and Acapela models. After purchasing and using the gravity-dependent devices, we came to the conclusion that this type of device was not the optimal choice for having a seamless integration of the OPEP device into the spirometer. The main reasons for this decision were that the gravitational devices could not easily have their resistances changed and that these devices required that the spirometer be held perfectly straight up. The main difference between the Acapela and the Aerobika OPEP devices is the ability to change resistance levels. The fact that the Aerobika device can have its resistance changed by simply flipping a switch was the deciding factor when we were choosing between these two devices. The lever that controls the Aerobika's resistances is attached to the oscillating flap inside the OPEP housing. When switching the lever to a different resistance, the oscillating flaps angle is changed, thus increasing or decreasing the resistance. This means that as the angle increases, the resistance of the OPEP decreases.

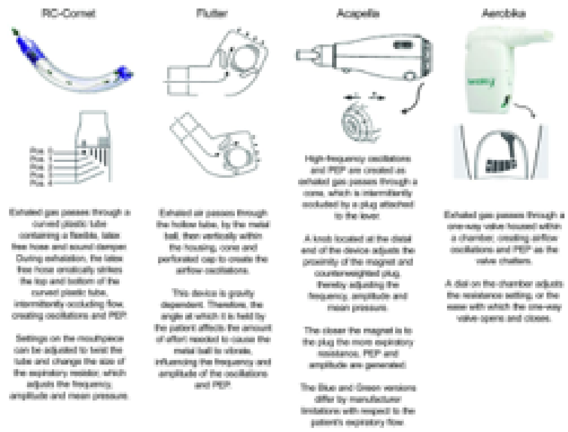


Figure 18: Different types of OPEP designs

Pressure Calculations: Although we are basing our model of the incentive spirometer off previous iterations of spirometers, we felt that it would still be beneficial to have a rough estimate of the pressure required to lift the piston in the diaphragm. The purpose of this analysis is to have a rough estimate to compare our future simulations with. Since these calculations ignore friction and air pressure loss, this estimate should be slightly higher than the actual pressure required to lift the piston. As we explore both experimental and simulations methods of getting flow data, having a rough estimate of pressure will help us to identify whether our simulations and experiments are accurate.

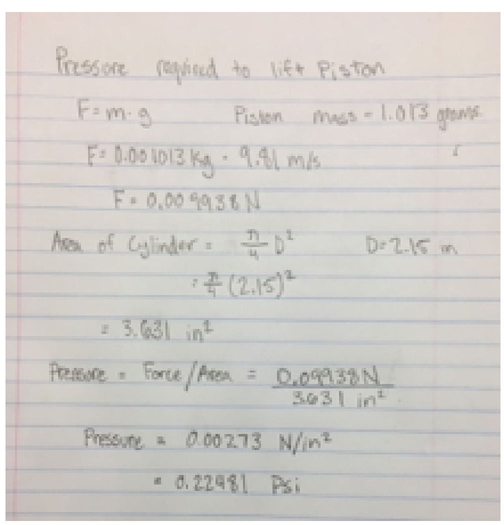


Figure 19: Pressure calculations to raise piston

To calculate this number, we used basic fluid mechanics equations. Using the equation: Force = Mass * gravitational acceleration. Using the piston mass of 1.013 grams, we were able to determine the force piston on the spirometer. After determining the force, we were able to use the equation: Pressure = Force/Area. From this equation we were able to determine that the minimum pressure required to lift the piston was .225 PSI.

ELECTRICAL AND COMPUTER ENGINEERING ANALYSES

APDS-9960 Module Analysis: Determine the circuit configuration of the gesture sensor and Raspberry Pi Pico together

The purpose of this analysis is to correctly connect the pins of the APDS-9960 with the pins of the Raspberry Pi Pico in order to power the module correctly and gather accurate readings from it. The gesture sensor module has 6 pins (Appendix - 1). The first pin from the left is the VL pin, which is used to provide optional power to the IR LED if the PS jumper is not connected. Since the Pico is already providing sufficient power to the sensor's IR LED, we do not need to use this. Another pin we do not need to utilize is the external interrupt pin, or the pin furthest to the right. Through different circuit examples, we noticed this pin was also currently unnecessary for the function of the sensor. While we were able to sense objects through the frosted glass, this pin may be utilized to help provide a steady interrupt signal for the provided speed of the piston we are trying to accommodate for. The other four pins were absolutely necessary for our gesture sensor to work. The ground pin, the second pin from the left, is connected to the ground bar of the circuit, which is also connected to the ground pin, pin 33. The Vcc pin of the module, the third pin from the left, is necessary to power the module and sensor. This is connected to the 3.3 V pin of the microcontroller, pin 36. The next two pins, the SDA and SCL pins, send a modulated pulse width that allows for the microcontroller to interpret the reading from the sensor. These are connected to pins 4 and 5 of the Pico.

GPIO pin Analysis: Determine the type of input/output pins on the Pico and interfaces needed to translate data into a meaningful message for the patient and pulmonologist.

The purpose of this analysis is to determine the type of pins required to collect and analyze data from the sensors to the microcontroller. The Raspberry Pi Pico has 40 different pins with varying input and output functions (Appendix - 2).

For the gesture/proximity sensor system, the APDS-9960 utilizes 12C communication to send data to the microcontroller. I2C communication is a serial communication protocol where data is transferred bit by bit along a single wire known as the signal data line (, where a master, in our case, the microcontroller, drives and reads the information from the slave, the sensor, through a serial clock pin. The raspberry Pi Pico has 2 I2C communications that can be utilized through 24 different pins. The pins we decided to use were GPIO pins 2 and 3. The settings of the clock and data line are already optimized by the sensor manufacturers (Appendix - 3). Another pin of the microcontroller that we are utilizing are the 3.3 V bus pin which is pin 36. This is necessary because the gesture sensor module, along with our display components, run off of 3.3 volts. In order to connect the grounds of the module, component, and the Pico, we used ground pin 33.

For the laser sensor system, values obtained from the LDR are analog values. Since the Pico comes with the majority of its pins configured only to process digital values, this analysis consisted of figuring out which pins on the Pico can convert analog values to digital values. The result produced that out of the 36 GPIO pins on the Pico, only 4 pins have analog to digital converters (ADC). And out of those 4 ADC pins, only 3 are usable. This analysis affirmed that 1 ADC GPIO pin will be required, and 3 other regular GPIO pins, to run the following devices: a Light Dependent Resistor (LDR), an LCD display, a laser, and a buzzer.

Software IDE Analysis: Determine the optimal integrated development environment (IDE).

The purpose of this analysis is to decide on the optimal IDE necessary to run the raspberry Pi Pico based on the different sensors we are using. The principal function of this IDE is to interpret the data from the sensors correctly and systematically display the data.

Since our laser sensor system only utilizes a laser emitter and a light dependent resistor, we chose to use Micro Python. Micro Python, our programming language of choice through Python interpreter (pip-installable), can be installed via operating-system package manager directly on the Raspberry Pi Pico. Our analysis produced that Thonny IDE is a great IDE to program the Pico using micropython. Because of its performance, Thonny IDE has been included by default in the Raspberry Pi's official operating system distribution. We concluded Thonny IDE far exceeds what we require to write an optimal code for the Pico.

Through analysis of the APDS-9960 and codes that incorporate it by the sensor's manufacturer Adafruit, we found that the necessary programming language needed is Circuit Python, an open-source derivative of Micro Python. This language interpreter and the library holding different Adafruit files essential for the sensor integration with the Pico can be installed via the Adafruit website. This package also had to be installed and changed in Thonny.

Sensor Distance and Orientation Analysis: Determine the distance within our electrical system sizing specifications that our sensor functions optimally.

The purpose of this analysis is to set the best conditions for the gesture sensor to read motion passing through the sensor's axis. While researching the sensor's capabilities, we noticed through the datasheet that the sensor is capable of sensing distances from around 1 to 8 centimeters. We decided to test this through experimental analysis. For the purpose of sensing the piston in our spirometer, we decided to that the distances between the distance of our spirometer and sensor would have to be under 3 centimeters. In order to test the optimal desired distance, we tested the reading of the sensor from four different directions, motion from the left, right, up, and down from 3 different distances, 1, 2, and 3 centimeters. We tested each direction 25 times for each distance. The testing circumstances with a set-up of a ruler reaching out from the bread board are seen in Appendix - 4.

From our results seen in Appendix - 5, the outcome of this analysis is that the optimal distance the sensor should be placed away from the diaphragm of the spirometer is 2 cm. The results show that the sensor was able to detect the 4 different directional motions to an accuracy of 88%, while only performing 80% accurately 1 centimeter away and 86% accurately 2 centimeters away. While some of this inaccuracy accounted for times when the sensor failed to sense any direction at all, some of the failed tests still output a direction of motion, just the wrong direction. We plan to mitigate these failed tests by optimizing the code of the APDS-9960 header file.

Through our testing, we were also able to tell the orientation of the module that would supply us with the most accurate sensor readings. From the results, we see that for 2/3 tests, the sensor outputs the most accurate data when the motion impeded from below the module moving upward past the sensor. With this information, we decided that the sensor should be placed with the pins on the bottom while integrated into our electrical system. If our results yielded any other direction as the most accurate, we would have oriented the module with that photodiode that senses that direction on bottom to indicate if the SMI was successful.

Power Consumption Analysis: Determine the average power draw of the system with the goal of limiting power drain, while still maintaining our sensor's full functionality.

The purpose of this analysis was to determine the power consumption of the system to be used for selecting a battery. In our system, the only component that will be depending on the power of the battery is the Raspberry Pi Pico. The other components such as the LCD display, the gesture/laser sensor, and small speaker are seen by their datasheets are all powered by 3 volts. These 3 volts are provided by the Pico itself. While these components draw current from the controller, they are all within the range of 10 – 20 micrometers, making them small enough that they do not affect the overall power consumption of the system.

While analyzing the load of the microcontroller, we noticed that the Pico runs on different modes that require varying currents. The Four different modes of the Pico are the popcorn mode, bootsel mode, dormant mode, and sleep mode. The average currents drawn from the Pico in these modes are shown in Appendix - 6, 7, 8, 9. The difference between these modes is the run-time and signal speeds and are decided based on the function of the code implemented. In our case, we decided that the microcontroller would be in two different modes. Most of the time, the controller is in popcorn mode while powering the sensor and collecting its data at a high rate. It will then switch to the bootsel mode while only displaying the number of successful inhalations since this would require a lot less power. With the average currents in these modes on the data sheet with an operating voltage of 5 volts for the Pico, the outcome of this analysis is that the system will require 0.4325 W of power when reading the output of the sensor and alerting the user and a much lower power of 0.04325 W when displaying the results only. These calculations can be seen in Appendix - 10. With these calculations, we can determine the size of the battery to its optimal capacity.

Battery Size Analysis: Calculate the battery capacity needed to successfully run the system for 70 hrs of operational time in a two-week span.

The purpose of this analysis is to calculate the optimal battery capacity and select a battery based on this analysis. This analysis depended on the power consumption analysis and 70 hour run time to calculate the battery size that would be needed. The calculated power needed to energize the electrical system is dependent on the mode of the Pico. While in a popcorn mode to power and read the data from the sensor, the Pico requires 0.4325 W. While the Pico is only displaying the results to the user and the doctor, it will be in a bootsel mode, which requires 0.04325W. To calculate the battery capacity, we need to fulfill a life cycle of 2 weeks in Amp hours, we divide each power by the battery voltage of 5 V while taking into consideration an efficiency of the battery of around 70%. This will be the current that the system will draw from the battery. We then multiply these currents to the number of hours we expect the Pico to be in this mode over the 70 hours of usage in 2 weeks. We decided as a team through estimations verified by Mr. Curran that the Pico would act in this high-signal mode for 60 of the 70 hours, while acting in the bootsel mode for the rest of those 10 hours of functionality. With these parameters, the result of the analysis was that a battery of around 10 Ah is needed to run the electrical system for 70 hours. These calculations are shown in Appendix - 11.

Based on these calculations of a desired 10 Ah, we narrowed down our battery to two different options. The first battery we decided upon is the CR 2032, which is a series of watch batteries that add up to a voltage of 3.6 V. While the Pico usually runs off a 5 V power source, its datasheet says its functional anywhere between 3-5.2 V. This lithium ion battery is also rated lower than the desired Amp hours of 10 at around 220 mAh. While this is significantly lower than the desired Amp hours for the 70 hours of functionality, it will be able to fit in the desired system well. We are also still considering this battery because we believe we may be able to run the Pico at lower current modes. The design and the product descriptions of this battery are shown in Appendix - 12.

The other battery we have decided to choose from is the USB Li-Ion Power Bank. This battery decision was brought up with the vision of making the electrical system rechargeable, providing with a longer lifespan which will be more reliable for the doctor's usage over time. This battery has 2 5 V outputs and can hold up to 5000 mAh. While this is still under the desired Amp hours of around 10, the rechargeability aspect allows for charging overnight, mitigating the need to hold the full capacity of power for the 2 weeks at one time. This battery option is also very admirable due to its already built-in USB ports. This design aspect allows us to connect the Pico to the power source in a very easy way, providing the necessary power to run the Pico and interpret the sensor's data. The design of this battery can be seen in Appendix - 13.

RISKS REDUCED OR REMAINING

During our discussions with Mr. Curran, we found out the true parameters of the device and how the device will be used day to day in hospitals.

Patient use/hospital distribution:

The device is considered a disposable single-use product. The plastic housing of the incentive spirometer can be produced cheaply, but due to the complex and narrow channels within the unit, it is not worth the time it would take to clean and sanitize the product for use among multiple patients. The hospital will distribute the unit to the patient post-surgery and will instruct them to use it during their stay and after they recover at home. The hospital wants the data of the patient's success and progress while they are in recovery at the hospital. Once a patient has gone home the doctor will not follow up and monitor the patient's spirometer usage. The electronic portion, therefore, should reflect that level of usage. The electronic modules should be removable from the spirometer housing. The cost of the electronic recorder drives the price far above the competition that exists on the market. The cost will be reduced through these electronics being removable. The cost of the electronic module will be spread out over dozens or hundreds of spirometers. The spirometers will be designed so that the patient can use the unit without the electronic portion and/or the OPEP module.

Internal Data Memory/Timer:

The initial team design considered the capture of data over several days with dozens of data logs each day. The amount of memory could be limited through simplification of the data, but this concern was soon dispelled. A more realistic estimation of the devices use is one of hourly use and memory wipe. The common case of use is a patient being prescribed a performance goal that should be reached 8-10 times within an hour. The device will be left with the patient for an hour and the attending physician will return to check on the progress made. If there is a need for more sets of inhalation therapy, then the attendant will reset the device and the hour-long reminders and data recording will resume for an hour. The only data that needs to be recorded is a count of how many times the patient met the performance goal set by the doctor. The number of unsuccessful attempts will be disregarded. Once the hour is over, the device will no longer collect data and store the number of successful attempts.

Battery life:

We were concerned about the hours of operation of the electronics during use. The main components of the electronics consist of the microcontroller, speaker, sensor module and LCD screen. Based off a specification sheet on the microcontroller cited earlier in the document which can be seen in the appendix, we noticed that the Raspberry Pi Pico ran on four different modes, all with varying required currents. One of the risks that still remains is the accurate sizing of our battery. In our battery sizing analysis, we decided that the Pico would be running on in the popcorn mode while reading the values from the sensor while immediately displaying the success of said inhalation to the user via the LCD display and in the bootsel mode while solely displaying the number of successful inhalations. With these assumptions and the amount of hours we decided we want our system to work, we calculated a desired capacity of around 10 Amp hours. We believe that this calculation may be slightly misleading by the mode we believe the Pico to be running on. There is a very high chance that our Pico will not be in a popcorn mode but in a lower power mode that requires a lower current. This will greatly decrease the capacity of our calculated battery.

Sensor Calibration:

While most of our tests were successful with the gesture sensor, the APDS-9960 was not able to indicate if the piston in the spirometer was able to reach the testing volume of 3500 ml. Since we know the sensor is able to detect an object through the frosted glass, we have concluded that the sensor is unable to detect the slow speed at which the piston is breaking the sensor's plain. In order to mitigate this risk, research will be done on the specifications of the actual sensor, along with how to manipulate the header file provided by the module's manufacturer Adafruit in order to account for lower speeds. Since the IR light is also a lot less intense than the laser sensor, it might be harder for this sensor to detect this motion compared to the sensor with such a high intensity. We can also manipulate the code and maybe add power to the VL pin of the module to intensify the IR light to gather more accurate readings.

Size Specification:

The size of the device needs to be kept at a reasonable size to where it takes as much room as preexisting premium spirometers on the market. We will allow the increase the overall footprint to accommodate for the addition of an optional OPEP module that interfaces with the spirometer, but if the device becomes too large, then it will be the same as if there were two separate devices.

Manufacturing Method:

The widespread practice for making plastic products is by injection molding. The spirometer housing needs to be able to be constructed by bringing two halves together and melting them together. This method of manufacturing keeps costs down, but the process of designing the mold can become complex and troublesome due to the complex geometry we are attempting to incorporate into the housing design. We are consulting with manufacturing engineers that are in contact with Mr. Curran. If there are issues or improvements to be made, they will be providing feedback on the models we produce. If we want to produce a model ourselves, we need to either obtain a SLA printer or seek out a service that will be willing to make one for us. Both avenues are being explored over the break.

UPDATED PROJECT

Figure 20: Updated block diagram

Memory: We initially wanted the electrical module to have some sort of memory storage to record a patient's therapy results over time. After discussion with Mr. Curran, we were told that this was not necessary as this device is aimed for patients who have just come out of surgery and the doctor would only like to know if the patient has completed their therapies.

Micro controller: We change adjusted our micro controller to be compatible with our sensors

ON/OFF button: through discussion with Mr. Curran, we simplified our user interface to only include a power button. We previously had thought about having additional buttons to adjust time intervals and look at other types of data that we could potentially provide to the doctor or patient.

WINTER BREAK AND FIRST WEEK SCHEDULE

While our customer, Kerry Curran, encouraged us to enjoy the break, Team 2nd Wind will be researching different elements of the project.

Task	Owner	Comments
Research how to speed up micro-python code on thonny IDE.	Saihou Jobe	Need to capture data quicker.
Research flow analysis methods (flow simulation Software/flow measuring instruments)	Nate Geller	Conduct research into manometers, and various flow simulation programs.
Will research potential options for producing/ molding plastic to produce our design	Omar Garibay	I will need to grab sheets of plastic and try different molding methods
Will compose an SLS purchasing proposal with Kinnon McPeak	Jason Smith	Finding price points, warranties on products and price to benefit ratios.
Research calibration specs and settings for APDS-9960 (gesture sensor) and start making website	Makana Dang	I will periodically insert information into the website over the break. The calibration specifications to read the piston will be discussed with a contact from Gonzaga who has experience with the APDS-9960.

First Week of the Quarter

Task	Owner	Comments
Continue researching to optimize code and narrow down how to capture data from spirometer, laser or proximity switch.	Saihou Jobe	Need to research how to make code run faster or use a different LDR
Work with Jason on developing experimental methods of measuring flow for both the OPEP and Incentive Spirometer device.	Nate Geller	Make decision on which flow simulation program to use
Continue drafting the drafting OPEP internals to fit onto the valve	Omar Garibay	Aiming to reduce size of OPEP device
Find test any software that claims to do flow simulations. Plan out experimental test set ups.	Jason Smith	Find the best way to verify that our OPEP design is valid and on par with market options
Finalize specific sensor (laser vs. proximity) and start circuit building and analysis.	Makana Dang	We need to select a sensor to focus on in order to maximize our efforts.

APPENDIX

The following pages contain all supporting documentation for the engineering analyses.

1.) Figures Referenced

2.) Supporting documents and figures

Referenced Figures:

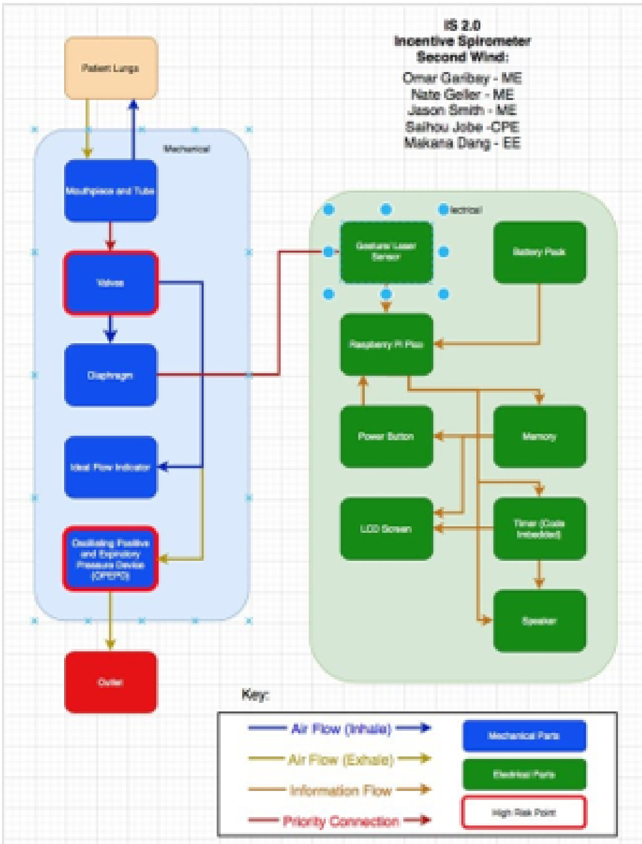


Figure 1: Block diagram of our incentive spirometer

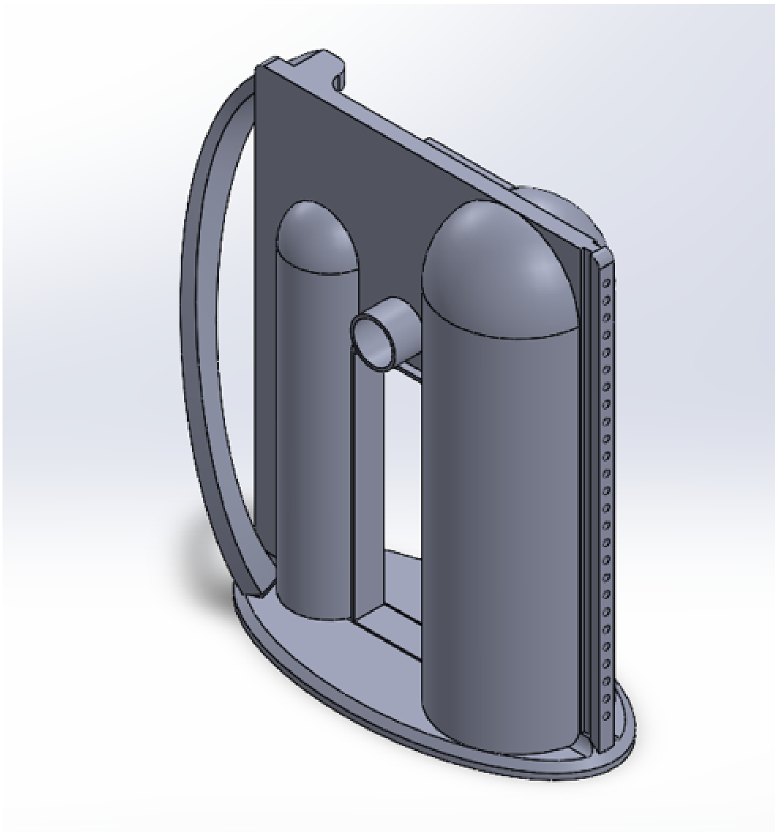


Figure 2: Incentive spirometer model

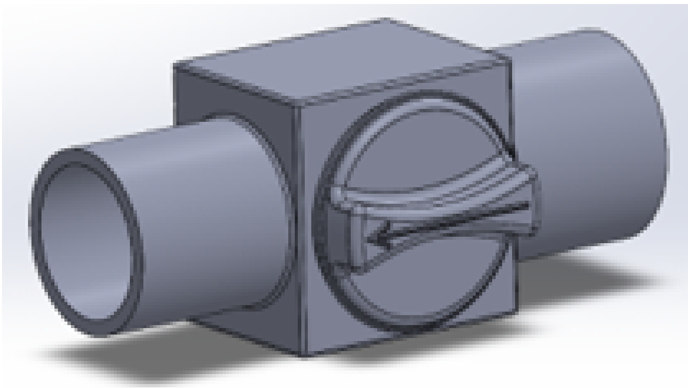


Figure 3: OPEP valve integration

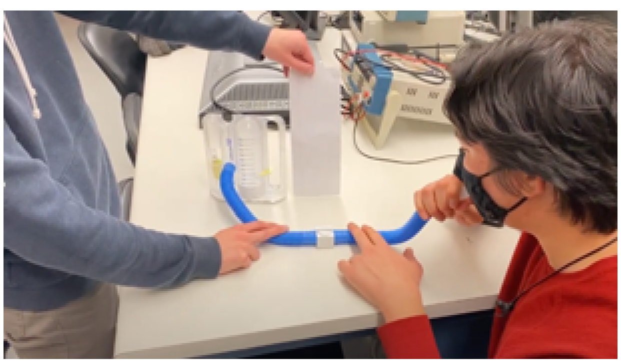


Figure 4: OPEP valve RRP verification

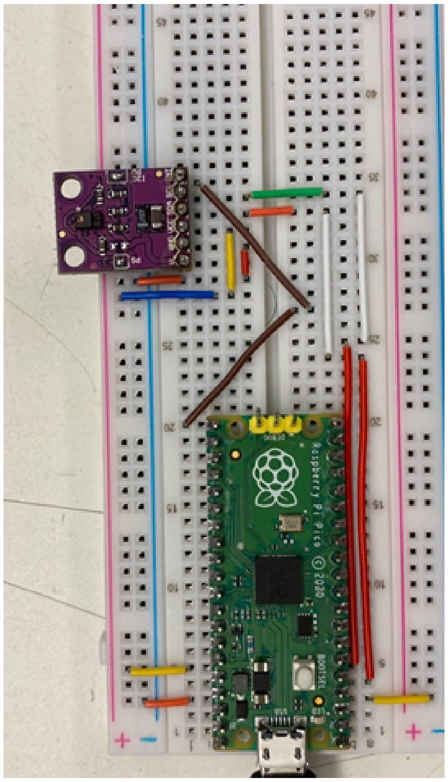


Figure 5: Gesture sensor connected to Raspberry Pi Pico

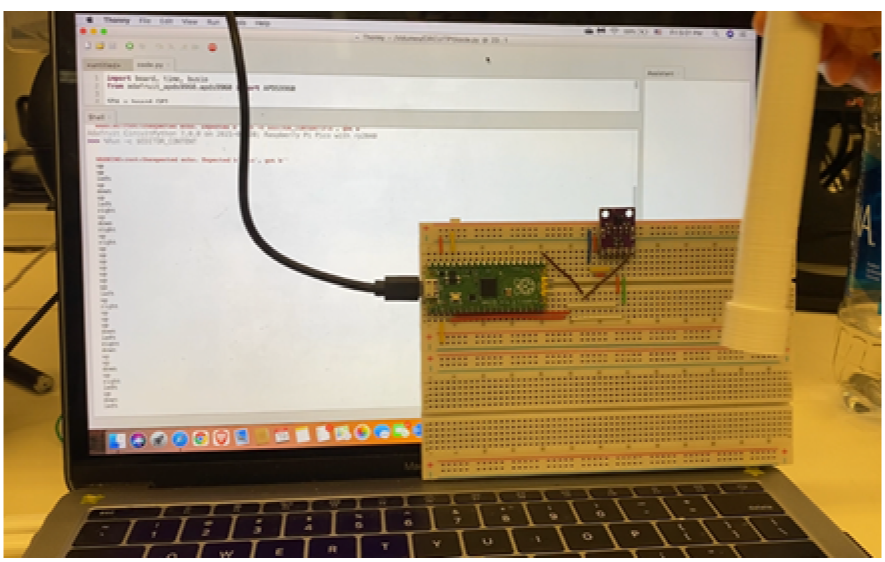


Figure 6: Testing and readings from first test

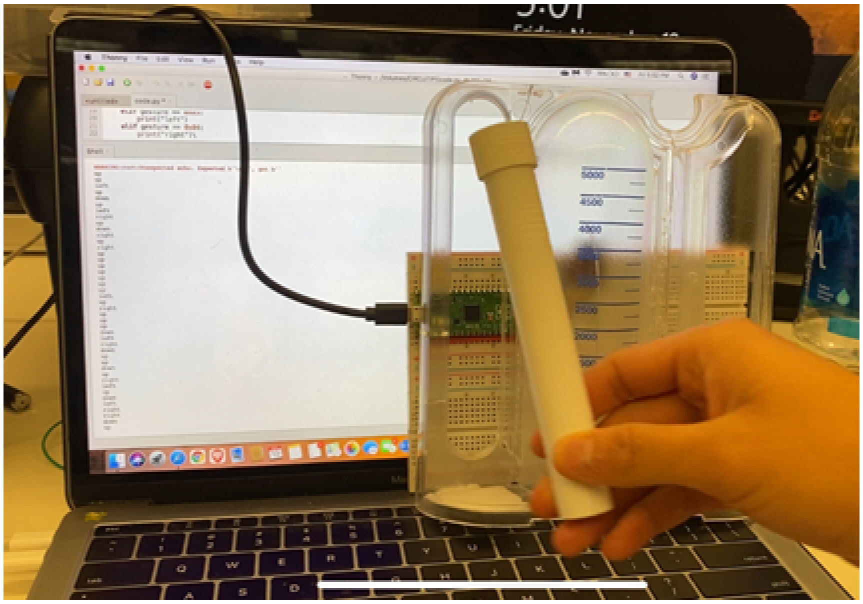


Figure 7: Testing and readings from second test

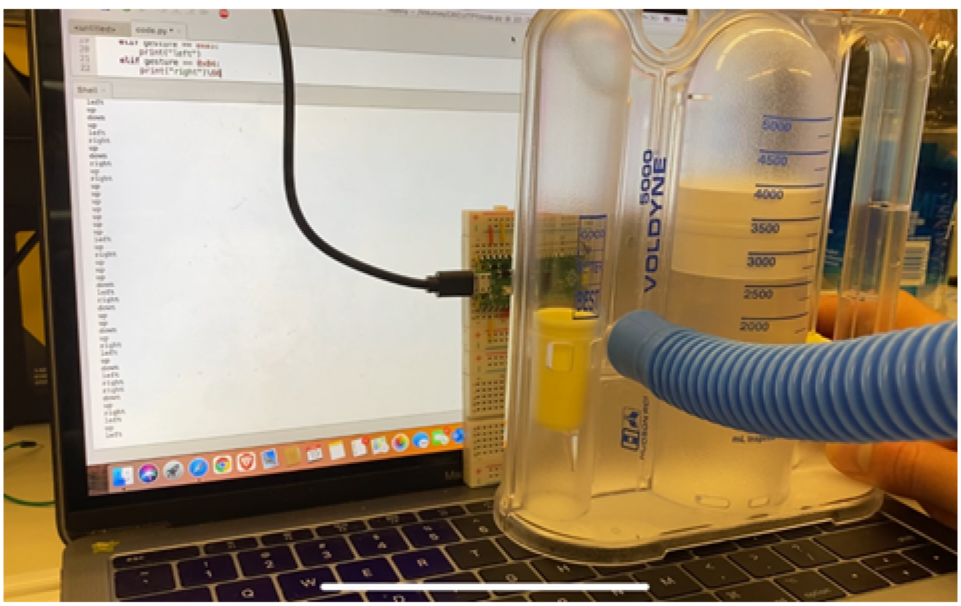


Figure 8: Testing and readings from third test

LDR

Figure 9: Light dependent resistor sensor

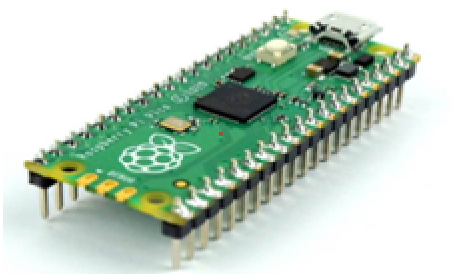


Figure 11: Raspberry PI Pico

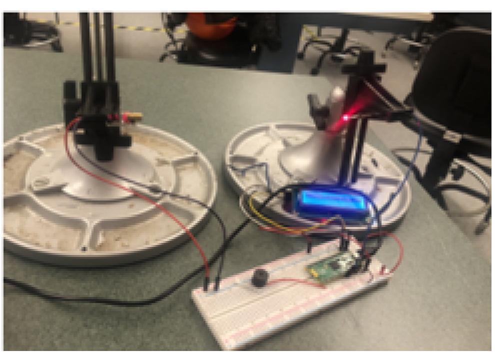


Figure 12: LDR laser sensor setup #1

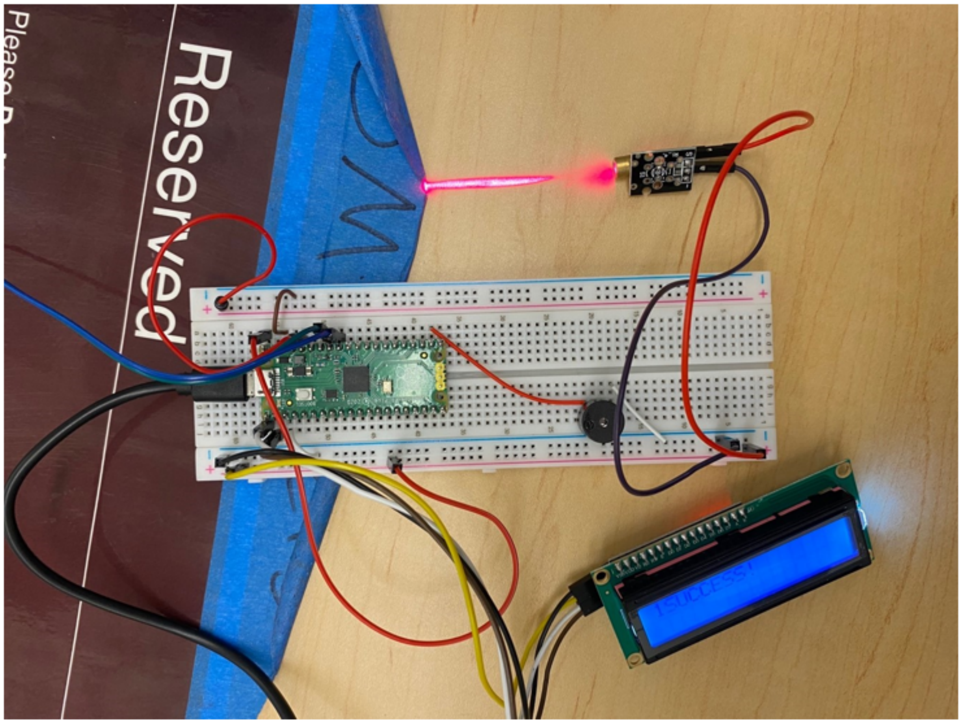
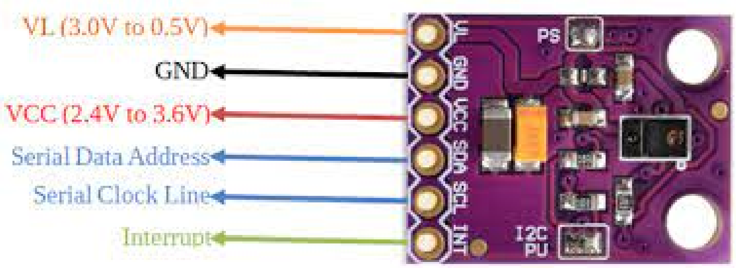


Figure 13: Laser Circuit Layout

Figure 14: LDR Circuit Screen Output

Appendix #	Figure/Table	Comments
1		<u>APDS-9960 Module Analysis</u>

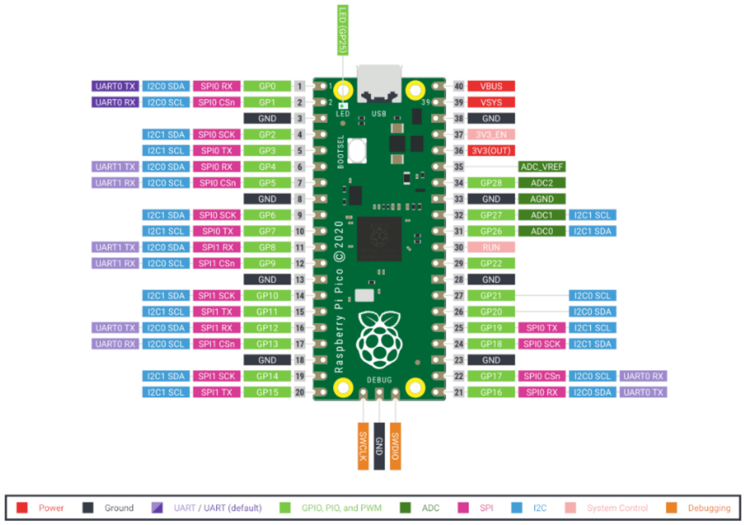
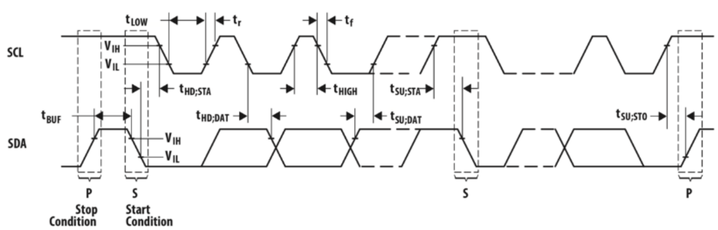
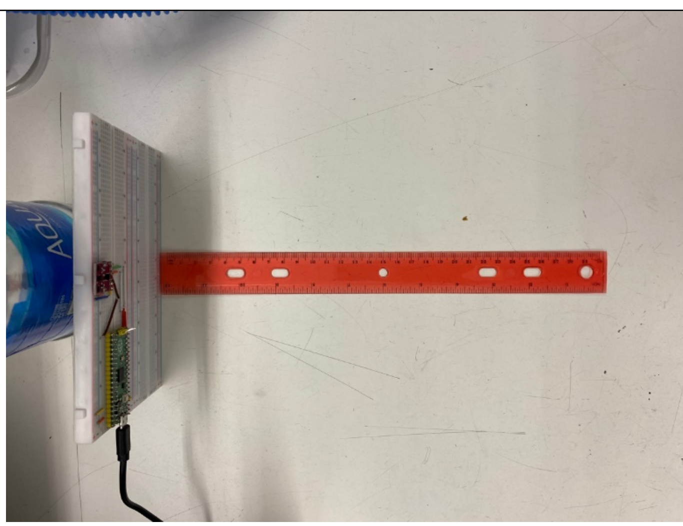
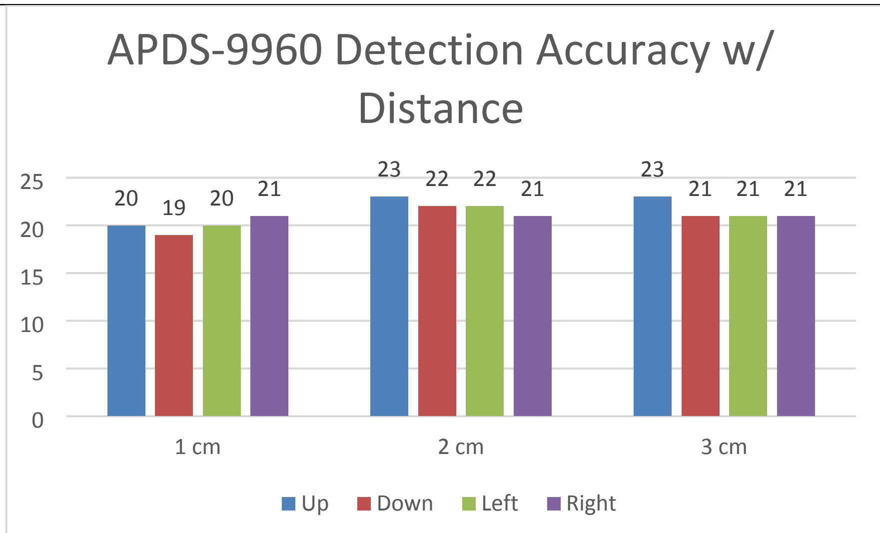
<table>
<tr><td>2</td><td>
</td><td><u>GPIO Pin Analysis</u></td></tr>
<tr><td>3</td><td>

Figure 1. Timing Diagrams</td><td>Timing diagram of</td></tr>
<tr><td>4</td><td></td><td><u>Sensor Distancing Analysis Setup</u></td></tr>
</table>

5



Sensor Distancing Analysis Results

6

Pico Board	Average VBUS Current @ 5V (mA)			Maximum VBUS Current @5V (mA)		
	Temperature (°C)			Temperature (°C)		
	-25	25	85	-25	25	85
#1	84.8	83.4	87.3	90.9	87.8	93.1
#2	87.5	89.9	89.8	93.4	94.1	94.0
#3	84.4	86.2	86.9	90.6	92.9	91.3
Mean	**85.6**	**86.5**	**88.0**	**91.6**	**91.6**	**92.8**

Power Consumption Analysis - Popcorn Mode

7

Pico Board	USB Idle VBUS Current @ 5V (mA)			USB Active VBUS Current @5V (mA)		
	Temperature (°C)			Temperature (°C)		
	-25	25	85	-25	25	85
#1	9.05	8.50	8.87	10.19	9.66	10.00
#2	9.58	9.12	9.20	10.64	10.35	10.36
#3	9.45	8.63	8.99	10.75	9.81	10.08
Mean	**9.4**	**8.7**	**9.0**	**10.5**	**9.9**	**10.1**

Power Consumption Analysis - Bootsel Mode

8

Power Consumption Analysis - Dormant Mode

Pico Board	VBUS Current @ 5V (mA)		
	Temperature (°C)		
	-25	25	85
#1	1.16	0.78	1.34
#2	1.19	0.92	1.40
Mean	1.2	0.8	1.4
#3	1.29	0.85	1.33

9

Power Consumption Analysis - Sleep Mode

Pico Board	VBUS Current @ 5V (mA)		
	Temperature (°C)		
	-25	25	85
#1	1.35	1.30	1.81
#2	1.53	1.39	1.92
#3	1.40	1.32	1.92
Mean	**1.4**	**1.3**	**1.9**

10

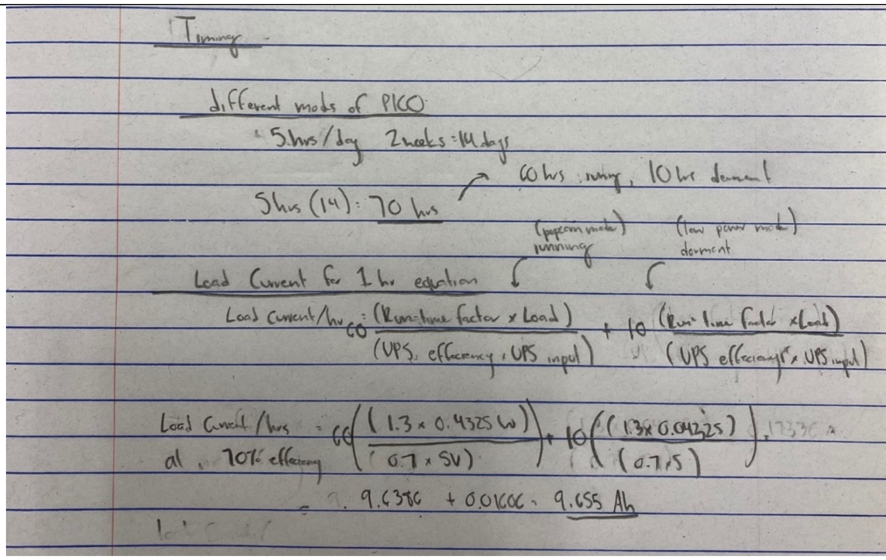
Load Calculations

Load Analysis & Battery Sizing

- always 5V - current changes based on pico mode

popcorn mode @ 25°C
current: 86.5mA
load = (86.5 mA)(5V) = 0.4325W
↳ plan on running for 60 hrs in 2 weeks

Bootsel mode @ 25°C when USB not attached
current: 8.63 mA
load = (8.63 mA)(5V) = 0.04325W
↳ plan on running in this mode for 10 hrs in 2 weeks

11		Battery Sizing Calculations
12	 representative photo click to zoom Weight: 0.05 lbs **Estimated Value - $15.00** Save as much as 58 % Last Value Update: 9/30/2020 Have you seen this for less? **Product Description** **Warranty** **OEM Part Numbers** • CR2032WRSUB • SUBSTITUTE FOR CR2032-WR • BATTERY • LITHIUM • 3V • 220MAH • COIN CELL STYLE • WIRE LEADS & CONNECTOR	Watch cell battery case

13		Lithium ion Power Bank

Other Models and Sketches