BME 405L: Senior Projects: Measurements and Instrumentation

Critical Design Review Report

Group 8

Mya Cohen, Boon Le, Karishma Rohatgi

Table of Contents

Product Overview

Presenting GlucoPress: one press, better life. GlucoPress is a three-step alert system for non-invasive glucose monitoring that is set up with a simple plug-in to your computer. A product like this will ease the minds of those managing diabetes. GlucoPress incorporates photoplethysmogram (PPG) signals to obtain a patient's heart rate. This heart rate value is typed into the patient-specified Machine Learning (ML) model to acquire their blood glucose level. The blood glucose value is then used for the LED signalling system to alert the user of hyperglycemia, hypoglycemia, or euglycemia (normal). With this innovative technology, diabetic patients can have a better quality of life and control over their blood glucose levels.

Identifying the Unmet Need to Features

Updated Problem:

Type 2 Diabetes Mellitus (T2DM) is a condition that affects 38.4 million working-age adults in the United States, with an alarming 8.7 million people undiagnosed. Managing this condition is critical, as poor blood glucose control can lead to life-threatening complications, including cardiovascular disease and kidney failure. Several glucose monitoring solutions are currently available, including Abbott's Freestyle Libre and Dexcom G6. However, these devices do not have a focus upon on-demand detection of hyperglycemia or hypoglycemia, nor do they include an alerting or signaling mechanism for urgent glycemic conditions. As well, traditional glucose monitoring methods are invasive, inconvenient, and time-consuming, making them inconvenient for individuals with demanding schedules.

Updated Need Statement:

To address this gap, our team is developing a non-invasive glucose monitoring system that allows users to detect whether they are hypoglycemic, euglycemic, or hyperglycemic based on their current heart rate. This device provides real-time feedback, empowering individuals to make timely health decisions without disrupting their busy schedules.

Stakeholders:

There are many stakeholders for our device. The first stakeholder is the most important, and their use of the product is provided in detail. The other stakeholders are provided in decreasing importance.

<u>T2DM Patients:</u> The primary users are T2DM working-age adults (25-54) who want to take ownership of their health and well-being. These users are who the device is mainly intended for and the ones directly impacted by the condition this device is helping with. The device's creation is to directly help these users to efficiently monitor their blood glucose levels and take decisive action to alleviate the symptoms. This group benefits most from the convenience of non-invasive

glucose monitoring, as it minimizes disruptions to their productivity while providing timely health insights.

Healthcare Providers: Endocrinologists, nutritionists, and other healthcare professionals benefit from the device's ability to track glycemic trends over time. This data enables them to develop personalized treatment plans and monitor patient blood glucose levels more effectively.

Caregivers: For individuals requiring assistance, caregivers can use the monitor's data to make informed decisions, such as adjusting meals or medications to maintain stable glucose levels.

Engineers: The engineers responsible for designing, testing, and manufacturing the device must ensure its reliability, accuracy, and usability. Their role is pivotal in creating a product that meets regulatory standards and user needs.

Market Research

The market for this product reinforces the main idea that Type 2 diabetics will benefit from this product. As mentioned earlier, T2DM is a condition that affects 38.4 million working-age adults in the United States, with an alarming 8.7 million people undiagnosed. This means that GlucoPress's market size is estimated to be 38.4 million people, corresponding to the number of U.S. Type 2 Diabetic patients. Not only is this a sizable population, but most importantly, our product will be beneficial to a large portion of people out of 38.4 million due to the lack of such non-invasive blood glucose monitoring systems for working-class adults.

Regulatory Pathway:

The regulatory pathway for our non-invasive glucose monitoring device would involve several key stages to ensure its safety, efficacy, and compliance with regulatory standards. Initially, the device would undergo extensive research and development to refine its prototype, focusing on optimizing the sensor accuracy and reliability through laboratory testing, as well as obtaining a high accuracy for our Machine Learning algorithm prediction. Afterwards, preclinical testing would be conducted to ensure its safety. Clinical trials would proceed in three phases, such as Phase I with a small group of healthy individuals to ensure safety, followed by Phase II with a larger cohort that involves those with diabetes, followed by Phase III with a diverse population to assess across different demographics, ages, lifestyles, etc. Throughout these trials, participant safety pertinent to regulatory standards would be at the forefront.

The trial data would be analyzed to ensure the device's effectiveness in diverse real-world scenarios. A regulatory submission would be prepared once there is a positive trial outcome. This would involve a 510(k) Premarket Notification to the U.S. Food and Drug Administration to demonstrate substantial equivalence to predicate devices. It would likely be classified as a Class II medical device. Post-market surveillance would monitor the device's performance in general use. These steps would establish Glucopress as a trusted solution for diabetes management in the workplace for T2DM.

Essential Features:

The essential features of the device include a PPG sensor integrated with a patient-calibrated Machine Learning model to accurately estimate blood glucose levels. It provides on-demand, real-time feedback through a simple and intuitive LED alert system, making it easy for users to monitor their glycemic status at a glance. Additionally, the design is ergonomic, being lightweight and user-friendly, allowing seamless use at a desk.

Patient-Driven Solution

This device was engineered with patient-centric intentions. All of those involved in GlucoPress research and development can attest to knowing a loved one who is managing diabetes due to the unfortunate prevalence in our population. Our inspiration for creating GlucoPress was based on the motivation to help one of our engineer's fathers better manage his blood glucose levels, especially during work. We made sure to integrate his advice and feedback when designing our device, highlighting our empathy for the patient in all aspects of the design process.

Use Cases to MVP Requirements

Patient safety is at the forefront of our product development. A logical use case based on our product's functionality is described below with a scenario, preconditions, postconditions, and triggers. The preconditions would be that the GlucoPress system is activated and functional, meaning it is hooked up to a computer with both the Arduino IDE running the two programs (PPG Sensor Data Acquisition and the LED Signalling System Program) and the ML model ready to run. The postcondition would be that the patient takes necessary steps after receiving an alert about their blood glucose levels in relation to hyperglycemia, hypoglycemia, or euglycemia. The triggers would be the symptoms of hyperglycemia, like increased thirst, frequent urination, extreme hunger, blurred vision, and fatigue, or hypoglycemia, like shaking, faster heart rate, sweating, confusion, dizziness, and extreme hunger, for it to be necessary that the user checks their blood glucose levels. The basic flow of this use case can be shown below, demonstrating the systematic purpose of GlucoPress.

Stakeholders

Please refer to the "Identifying the Unmet Need to Features" section for the stakeholders.

Basic Flow

1. The GlucoPress system acquires the user's heart rate and oxygen saturation through a red LED and infrared LED on the PPG sensor.
2. Utilizing the PPG signal and other individualized user parameters, the ML algorithm through linear regression predicts the blood glucose in milligrams per deciliter with an 86% accuracy.
3. The blood glucose prediction in milligrams per deciliter is then put through an Arduino IDE program to trigger a solid red LED light if the blood glucose levels are greater than 130 mg/dL, a flashing red LED light if the blood glucose levels are less than 80 mg/dL and solid green LED light if the blood glucose levels are between 80 mg/dL and 130 mg/dL.

Illustrative Use Case Storyboard

Figure 1. UPDATED Use-Case Scenario Storyboard

Consider John, a type 2 diabetic heavily engaged in his work. After eating lunch, John begins to feel uneasy but dismisses the symptoms due to his workload. His non-invasive glucose monitor alerts him to a blood glucose level of 180 mg/dL, triggering a solid red light to indicate hyperglycemia. Prompted by this alert, John takes immediate measures to lower his blood sugar. Conversely, had his blood glucose fallen below 80 mg/dL, the monitor would display a flashing red light, signaling a hypoglycemic emergency. Within the normal range of 80–130 mg/dL, the device displays a reassuring green light. This device bridges the gap between traditional glucose monitoring and the demands of a modern work environment, enabling users like John to take proactive steps in managing their health.

Functional Requirements

Table 1. Functional Requirements Table

Overall Category	Functional Requirements	Solutions for Subfunctions	Category
Real-Time Monitoring	Real-time monitoring through the press of a finger for the blood glucose value to identify abnormal blood	Continuously be activated or “on” for the user to press their finger upon the device and continuously have a	Data

	glucose levels and for the user to be able to take immediate action to rectify their abnormal blood glucose levels should they occur.	red sensor light, indicating that the PPG sensor is on and ready to take readings from the patient to ultimately alert the user through the LED system.	
Non-Invasive Usage	Photoplethysmography signals (PPG) signals and the related software component to deliver a HR value non-invasively through a finger press.	This device needs to be connected to a computer connected to the Arduino IDE and Machine Learning algorithm.	Data
Machine Learning Model	In order for the device to function as a blood glucose monitor, the key software functionality is the Machine Learning Algorithm. This will input the HR obtained from the PPG sensor and output a predicted blood glucose value in mg/dL.	The ML model will require several other parameters such as weight, age range, height, gender, blood pressure characteristics, etc, in order to predict an accurate blood glucose level. The ML Algorithm should be coded in Python.	Data
Personalized Device Calibration	The device should require the user to initially input their physiological parameters for the ML algorithm to be able to predict a blood glucose level—as it will not be accurate if these parameters are not already loaded into the algorithm.	Thus, while setting up the device, the user should provide the following: -Weight (kg) -Height (cm) -Blood Pressure Characteristics (Systolic, Diastolic, Pulse Area) -Gender (1=Male, 0 = Female) -Age Range (Working age individuals should be in 2-4 Age Range)	Data
Display of Glucose Status Notification System (LED	The user should be able to clearly tell their blood sugar category from the LED	The organization of the LED bulbs on the system should be placed in a clear	Processing

System)	system. The user should be able to easily tell the difference from a red LED flash, a red LED on, and a green LED on.	way so that users can tell the difference between the notifications. The user manual should indicate the following: -Red LED Flash = Hypoglycemia This will occur if the predicted blood glucose level is below 80 mg/dL -Red LED On = Hyperglycemia This will occur if the predicted blood glucose level is above 130 mg/dL -Green LED On = Normal This will occur if the predicted blood glucose level is in between 80-130 mg/dL	
Portability and Connectivity	This device should be able to connect using a cable that is compatible with multiple types of computers to address the different computers/monitors used by working age individuals at their workplace. The device should contain a port that allows for seamless connection between the device and a computer.	The device should be designed to be lightweight, easy to carry to-and-fro work, and easy to integrate into the computer at their workplace and stay connected.	Interface
PPG Energy Signals to Parameter Conversion Code	The device should initially retrieve PPG signals from the sensor.	Using the code found from the Analog Devices Incorporated developer, it should output HR, SpO_2 and other parameters on the Arduino IDE. Using this	Processing

		conversion ensures that the HR can be accurately captured from the user's fingertip via the PPG sensor and input this HR into the ML portion of the device.	

Non-Functional Requirements

There are several non-functional requirements, such as reliability, accuracy, durability, and functionality, needed for our device to operate properly and safely. The reliability of the device is crucial for users experiencing symptoms of hyperglycemia and hypoglycemia to know their blood glucose levels to ultimately alleviate their symptoms. We plan to test this by simulating diverse workplace settings to ensure the device operates seamlessly. Accuracy is another important requirement that has implications all the way from interpreting the correct heart rate value, to accurately predicting blood glucose values with the ML model, and finally accurately displaying the user's blood glucose status via the LED Signaling System. This reduces the likelihood of false alerts, ensuring users receive precise feedback about their glycemic status. Validation of the device's accuracy will involve controlled testing against reference blood glucose measurements to ensure its outputs align with medical standards. Durability is necessary to ensure the device withstands daily use, including minor impacts or drops that might occur during a busy workday. To address this, we will design and test protective casings, ensuring the device is durable under real-world conditions. Lastly, functionality is imperative due to a computer plug-in being necessary for users to power the device. Testing will involve ensuring the device communicates effectively with multiple platforms and remains easy to use for individuals with varying levels of technological proficiency.

Preliminary Design Review (PDR)

The link to the PDR is found here: Link

Engineering Design Report

Summarized Problem and Need Statement:

There is a lack of non-invasive blood glucose monitoring devices in the healthcare market tailored to the growing population of working-age individuals (25–54) with Type 2 Diabetes Mellitus (T2DM). These individuals need a seamless, convenient, and accurate way to monitor their blood glucose levels throughout the day without disrupting their professional and personal routines. Current solutions, such as the standard finger-pricking devices and continuous glucose monitors (CGMs) like the Dexcom and Freestyle Libre devices, are invasive, time-consuming, and require a multi-step process that can be distracting and inconvenient.

Thus, a non-invasive, quick, and accurate solution is essential to allow patients to make quick and informed health decisions. A device that provides real-time glucose readings and alerts patients to abnormal blood sugar levels could help them take appropriate actions—such as administering insulin during hyperglycemia or consuming sugar during hypoglycemia—thereby improving their quality of life and health management while minimizing interruptions to their day-to-day activities.

List of Functional Requirements

Device As It Exists Now:

Real-Time, User-Initiated Monitoring

Real-time monitoring of blood glucose values is essential to identify abnormal blood glucose levels and for the user to be able to take immediate action to rectify their abnormal blood glucose levels. Thus, the device should be able to provide an indication of the patient's blood glucose level *on demand* to inform them of their blood sugar status. When the user chooses to sample their blood glucose level, they should be able to press their finger on the sensor, and the result should indicate their blood glucose status. Real-time monitoring that is fast and convenient is therefore essential for our product. To implement this function, the device should:

1) Continuously be activated or “on” for the user to press their finger upon the device and the PPG sensor more specifically

   a) For the PPG sensor used, the device should continuously have a red sensor light, indicating that the PPG sensor is on and ready to take readings from the patient.
2) Consist of a working PPG sensor that can take HR values and deliver them to a ML model that will deliver its blood glucose level to a LED threshold system.
   a) The device's data sub-inputs should be housed in an integrated software system.
3) Take less than 1 minute to deliver a result on the LED system (flashing red = hypoglycemic, still red = hyperglycemic, still green = normal).
4) Automatically reset after the 10 seconds of the LED being on has elapsed.
5) As an additional parameter for the user's benefit, the device should display the user's SpO_2 level.

Non-Invasive Operation of Device (Use of PPG Sensor)

The user should be able to monitor their blood glucose level without using any needles or puncturing the skin. The device should not require blood samples or sensor insertions into the skin. Instead, it should rely on photoplethysmography signals (PPG) and the related software component to deliver a HR value, which will be used for the remaining components of the device.

Machine Learning Algorithm for Blood Glucose Prediction

In order for the device to function as a blood glucose monitor, the key software functionality is the Machine Learning algorithm. This will input the HR obtained from the PPG sensor and output a predicted blood glucose value in mg/dL. It also will require several other parameters, such as weight, age range, height, gender, blood pressure characteristics, etc., in order to predict an accurate blood glucose level. The device should use a Linear Regression with Polynomial Feature Extraction, as this gives a R^2 value of around 86%. The features extracted for the model are the parameters aforementioned. The ML algorithm should be coded in Python.

Personalized Device Calibration

The device should require the user to initially input their physiological parameters for the ML algorithm to be able to predict a blood glucose level, as it will not be accurate if these parameters are not already loaded into the algorithm. Thus, while setting up the device, the user should provide the following:

- Weight (kg)
- Height (cm)
- Blood Pressure Characteristics (Systolic, Diastolic, Pulse Area)

- Gender (1=Male, 0 = Female)
- Age Range (Working age individuals should be in 2-5 Age Range)
  - The age range corresponds to the ages of the users.
  - Age Range = 1 means 0-20, Age Range = 2 means 21-30, Age Range = 3 means 31-40, Age Range = 4 means 41-50, Age Range = 5 means 51-60.

Once the above parameters are loaded into the ML model, the device is now calibrated and ready for the patient to monitor their glucose level but will not work for another individual as they have differing physiological parameters.

Efficient Display of Glucose Status Notification System (LED System)

The user should be able to efficiently, unmistakably, and clearly tell their blood sugar category from the LED system. The user should be able to easily tell the difference between a red LED flash, a red LED on, and a green LED on. The organization of the LED bulbs on the system should be placed in a clear way so that users can tell the difference between the notifications. The user manual should indicate the following:

- Red LED Flash = Hypoglycemia
  - This will occur if the predicted blood glucose level is below 80 mg/dL
- Red LED On = Hyperglycemia
  - This will occur if the predicted blood glucose level is above 130 mg/dL
- Green LED On = Normal
  - This will occur if the predicted blood glucose level is in between 80-130 mg/dL

Connectability and Compatibility of Device to PC/Computer

This device should be able to connect using a cable that is compatible with multiple types of computers to address the different computers/monitors used by working age individuals at their workplace. The device should contain a port that allows for seamless connection between the device and a computer. For instance the device should have:

- USB-A to USB-A
- USB-A to USB-C
- USB-B to USB-A (the device currently operates on this cable)
- All types of USB connectors depending on the ports of the two systems

Usable Design

The device should be designed to be lightweight, easy to carry to and from work, and easy to integrate into the computer at their workplace and stay connected. The device should not be too large to avoid taking up too much space on the user's workstation, as it will become inconvenient to use otherwise. The user should be able to customize their device to their parameters for the software portion of the device and to their aesthetic needs. For instance, the user should be able to choose different colors for the hardware component of their device depending on the user's desires.

PPG Energy Signals to Parameter Conversion Code

The device should initially retrieve PPG signals from the sensor, and using the code found from the Analog Devices Incorporated developer (see References), it should output HR, SpO_2,and other parameters on the Arduino IDE. Using this conversion ensures that the HR can be accurately captured from the user's fingertip via the PPG sensor and input this HR into the ML portion of the device.

Device As It Exists In The Future:

Integrated Software System

In order to ensure that the device has a seamless data flow, due to the fact that it involves a sensor for heart rate values, a ML model to predict blood glucose values, and an LED threshold system, the data flow should be done automatically with minimal user input. It should thus be housed in an integrated software system that is coded to take in HR values, run the ML model to obtain a blood glucose result, and input that value into the LED system to shine the corresponding LED tailored to the blood glucose categorization. If the user has to input several values into several systems, it will take too long and not be a convenient solution. Thus, in order to minimize user input, the software system should:

1) Take a HR value from the user pressing their finger on the PPG sensor (INPUT)
2) Input this obtained HR value into the ML model along with the preloaded parameters and obtain a predicted blood glucose level.
3) Take this blood glucose level and input it into the LED system connected to the Arduino circuit, and result in a FINAL output of either a red LED (hypoglycemia/hyperglycemia) or green LED activation.

Safety Compliance

Relevant safety standards need to be incorporated into the device to ensure its compliance with medical regulations. If it is marketed as a medical device, it must be subject to FDA regulation depending on the assigned classification of the device. We should perform Risk Analysis procedures such as:

- Failure Mode and Effect Analysis (FMEA)
- Fishbone Cause-and-Effect Diagrams
- Hazards Analysis and Critical Control Points (HACCP)

Additionally, we should test the device for its efficacy and perform tests to ensure it functions at various temperature, moisture, and pressure conditions. We can perform safety tests on the following components of the device:

- The LED light bulb notification system
- The PPG Sensor
- The USB ports and connection component (to ensure connections are stable and safe to touch)

System Overview - Black Box Inputs and Outputs

Below is a simplified black box diagram of the GlucoPress device with 3 inputs and 3 outputs. Each input is classified as energy, information, or both. Because our input/outputs are largely electrical in nature, there is no matter involved in the system. The inputs are the computer/PC connection to the GlucoPress system and the patient's fingertip that they will place on the PPG sensor. The outputs of the system are from the LED signalling component of the device, which will either provide a red or green light indicating the three blood glucose level categories. In addition, there is a Reset Button input, which is an indirect input that the system will only consider as an input if the system needs to be rebooted due to malfunctioning.

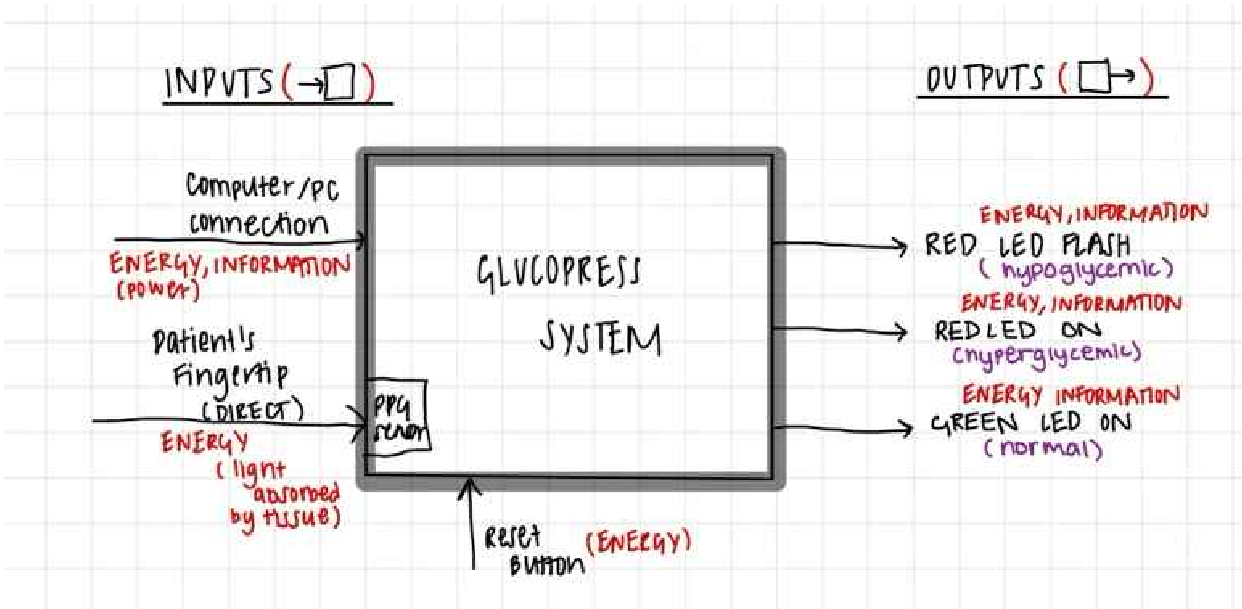


Figure 2. Black Box Diagram

Functional Decomposition

Table 2. Functional Decomposition Table with Function and the Corresponding Inputs and Outputs

Function	Inputs	Outputs
PPG sensor on the surface of the hardware component.	1. Fingerprint from user applying pressure onto the sensor	1. PPG Signals which are transmitted to the Arduino IDE for noise removal and signal processing 2. PPG Signals are represented through main outputs of HR, SpO_2, Temp, Clock, Ratio, and Correlation. Only HR and SpO_2 are relevant metrics for this system.
A computer/PC connection port in the	1. USB-B to USB-A connector, connecting the Arduino UNO	1. Cable connecting the Arduino UNO

bottom of the hardware device.	board to a computer/PC	R3 board to a computer/PC.
A reset button on the top of the hardware device box.	1. A 3D printer pillar that sits on top of the reset button on the Arduino UNO R3 board	1. When pressed, the Arduino UNO board will reset, and will show up as resetting/rebooting on the Arduino IDE.
A ML Algorithm written on Python using scikit-learn library.	1. Input parameters are: - Height (cm) - Weight (kg) - Gender - Index (from training dataset) - Patient ID (Patient # from training dataset) - PPG Signal - Age Range - Blood Pressure Characteristics (Systolic, Diastolic, Pulse-Area) - Heart Rate (bpm)	1. Predicted Blood Glucose Level (mg/dL)
An 2-color LED signaling system for glucose level indication to the user with set thresholds.	1. Blood Glucose Level Prediction (from the ML model)	1. Red Flash for 10 seconds, 1 flash/second = hypoglycemia (<80mg/dL) 2. Red LED ON for continuous 10 seconds = hyperglycemia (>130 mg/dL) 3. Green LED ON for continuous 10 seconds = normal (80-130 mg/dL)
The PPG sensor component should contain code that converts PPG signals	1. PPG Signals which are light transmission measures through tissue that gets affected by heartbeats.	1. Converts incoming PPG signals into HR, SpO_2, and other parameters.

to HR, SpO_2 and other vital parameters.		2. Sourced from the developer's GitHub, Analog Devices Inc for MAXREFDES117 # PPG Sensor.
The hardware should be housed in a design that is lightweight and around 8cm by 7cm by 7cm to ensure it does not take up a huge portion of the workstation.	1. Lightweight red PLA filaments used 2. Size currently: 7.8cm by 6.25cm by 6.5cm for MVP	1. A square box with openings for the LEDs (x2), the reset button pillar, the connection ports, and the PPG sensor

Concept Classification Tree:

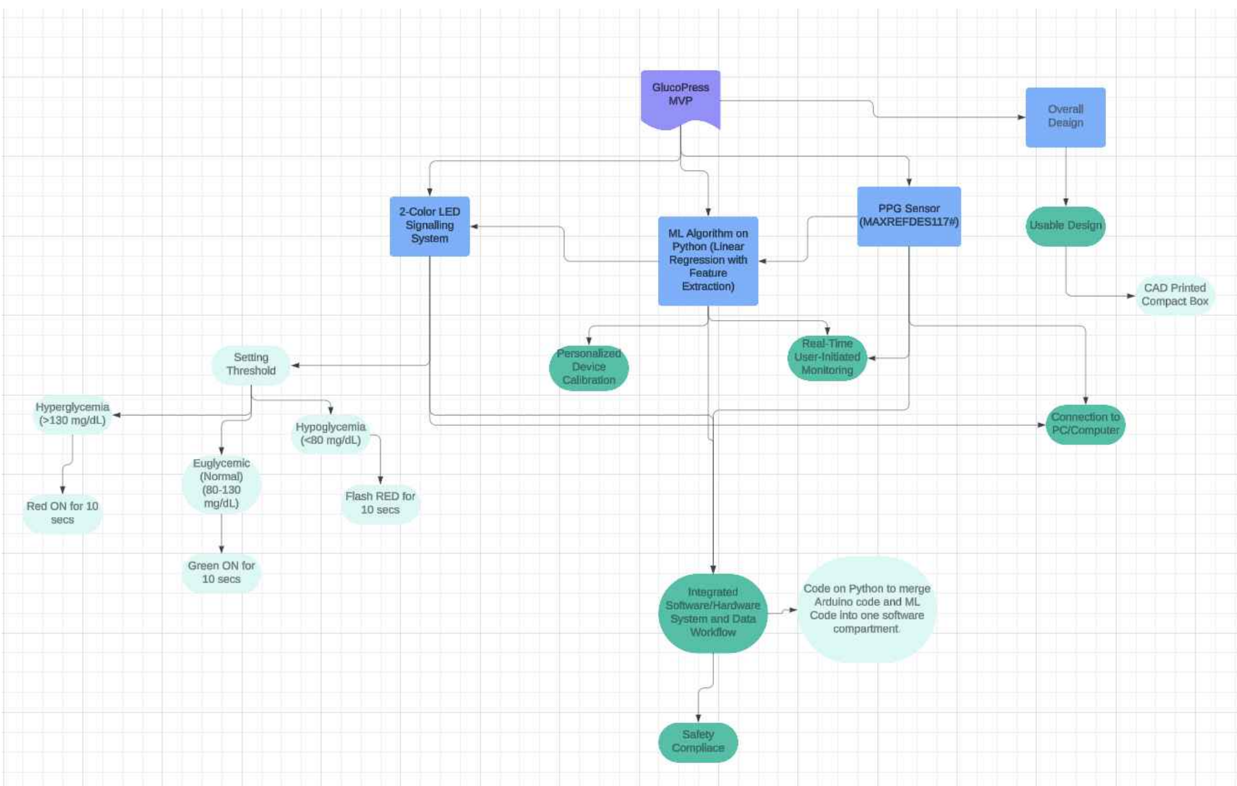


Figure 3: Concept Classification Tree.

Legend: Purple is the MVP which is called GlucoPress. Blue signifies the different components and parts of the MVP. Teal signifies the corresponding functional requirements associated with each component (only the most important ones have been listed on the tree). Note: there may be overlap of some requirements between the components. Light Blue signifies the outputs associated with the functional requirements. Note: the integrated software system box on the bottom of the tree is for the device as it exists in the future, as the current MVP does not include an integrated software system.

As you can see, there are several overlaps between the functional requirements. Real-time, user-initiated monitoring is associated with the ML algorithm and the PPG sensor, as both of them are required for the device to function and collect data in real time. Connection to PC/computer is also shared among all three major parts of the MVP (without the overall design component), as they all work together and require a computer to display the sub-inputs and sub-outputs. Safety compliance is shared among all three components, as the whole GlucoPress system will undergo safety testing to ensure it is compliant with medical regulations and up to standard.

Concept Combination Table:

Table 3. Concept Combination Table (Derived From Concept Classification Tree Above)

Functional Requirement	Real-Time User Initiated Monitoring	Personalized Device Calibration	Connection to PC/ Computer	Usable Design	Integrated Software System	Safety Compliance
2-Color LED Signalling System			In order for the LED signalling system to work on the Arduino UNO board, it needs to be connected to the Arduino IDE on the computer. When the user inputs the blood glucose prediction value from the ML model, it should		Future: The LED system should result in an LED on (either red or green) depending on the blood glucose value prediction from the ML model automatically.	Future: The LED system should be evaluated to ensure that no bodily harm is possible when the device is at high temperatures or extreme conditions. The LED should be evaluated for safety to ensure it does not

			correctly turn on the appropriate LED (red or green).			break at extreme conditions.
ML Algorithm	The ML algorithm will efficiently input the incoming HR data from the sensor and along with the preloaded physiological parameters will quickly (<30 seconds) output a blood glucose prediction	The physiological parameters associated with each patient will need to be inputted into the model prior to operating the device. This is to ensure the ML algorithm is predicting an accurate blood glucose result.	The ML algorithm is done on Python thus it will need to be connected to the computer in order to function		Future: This ML algorithm will be loaded onto the central software system of the device so the user does not have to input their HR by typing it in (the PPG sensor will send it to the algorithm itself).	Future: The ML model will need to be evaluated for patient privacy to ensure no data is being leaked. Thus, data should be encrypted.
PPG Sensor	The PPG Sensor should be continuously on (should show a red light from the sensor) so that the user can choose whenever they want to monitor their blood		The PPG sensor is connected to an Arduino UNO board and thus must be connected to the PC/Computer in order to collect data that will be		Future: The sensor will automatically send the PPG signals for data processing to the Arduino IDE and	Future: The PPG sensor should be evaluated for safety under extreme conditions so that the user does not incur any bodily

	glucose value in real time.		shown on the Arduino IDE.		the rest of the components, so the user does not have to click any key to start the data conversion from PPG signals to the HR and SpO_2 parameters.	harm from the device.
Overall Design				The design of the box should be small and compact enough that the user can place this device at their work station without it taking too much space. The house for the device can be CAD printed as PLA filament is quite sturdy.		

Overall Concept Screening and Discussion of Solution

By amalgamating the concepts presented in the concept combination table, we were able to pinpoint the primary solution we aimed to tackle with our project MVP. We wanted to create a safe way to non-invasively monitor an individual's blood glucose by utilizing their vital signs, such as their heart rate, in order to inform users of their current blood glucose status. We made the device's main output into a simple, easy-to-understand LED system that would shine green when it is normal and red when it is abnormal. These colors were chosen as they are quite intuitive for the user to understand (green = good, normal, and red = bad, abnormal). When working on the project in regards to the ML portion, we were finding it challenging to be able to predict a blood glucose level without the user-specific physiological parameters, and thus a key functional requirement needed to be included, which was the Personalized Device Calibration. This accurately reflects the main challenge engineers who are designing non-invasive blood glucose monitoring methods face, as it is difficult to output a blood glucose result that is accurate without taking into consideration the patient's weight, height, age, gender, and other characteristics that play a part in the user's blood glucose value.

In the beginning of the project, we initially wanted to use PPG signals without the conversion to HR in order to accurately measure the blood glucose, but we chose to instead convert the PPG signal to a HR, as not only was it also a parameter of the ML model, but it involved less data processing (such as sampling, filtering, and ensuring no data aliasing occurred), which would have been too cumbersome and time-consuming. We had to take into consideration the limited time we had to build this MVP, and we chose to instead utilize a resource from the developer to convert incoming PPG signals into HR, as this would match our group’s technical abilities. This is why one of the functional requirements is to take PPG signals and convert them into HR for the other systems (read the Integrated Software System column in the Concept Combination Table). This decision to convert incoming PPG signals into HR also impacted the rest of the device, such as the ML model (as HR is an input) and subsequently the LED signalling system based on the ML algorithm's output.

Concept Screening Matrix:

One concept that we can illustrate in a concept screening matrix to establish why we chose what we did is regarding the different ML models used to predict a patient’s blood glucose level. While exploring different ML models in Python, we came across several platforms that would be able to train a model (using automatic Machine Learning features), but we ended up going with the sci-kit learn library in Python and, more specifically, the linear regression with feature extraction. We explored solutions to train the dataset, such as Google Cloud's Vertex AI, which enables AutoML training, as well as other models in Python, such as Random Forest and Decision Tree, before ultimately settling down on Linear Regression with Polynomial Feature Extraction. The biggest reason for choosing the more simple regression technique was largely due to the simplicity of training and testing the dataset, as well as the highest R2 value that we found, which was 0.86 (see image from code below). Additionally, as the BME major does not include very intensive computer science courses, we were limited in the technical ability to grasp and understand more complex ML models such as RandomForest and DecisionTree, as well as be able to troubleshoot in the case that we created the model from AutoML on Google Cloud's Vertex AI.

```
r2_test = r2_score(y_test_scaled, y_test_predict)

#print('R2 score for Linear Regression Testing Data is: ', lr_r2_train)

print('R2 score for Poly Regression  Testing Data is: ', r2_test)
```

```
R2 score for Poly Regression  Testing Data is:  0.8643733604885401
```

Below is a Concept Screening Matrix for the different ML models explored.

Table 4. Concept Screening Matrix Table for using Linear Regression over other ML Models

Criteria	Linear (Poly) Regression with Feature Extraction	AutoML on Vertex AI (GoogleCloud)	RandomForest	DecisionTree
R^2 Value	0.86	–	0.54	0.33
Ease of Use	3/5	4/5 (if it worked)	2/5	2/5
Customizability	5/5	3/5	2/5	2/5
User Interaction	3/5	2/5	3/5	3/5
External Technical Support/Resources	5/5	3/5	3/5	3/5

As Linear Regression with Polynomial Feature Extraction was more accurate (based on the R^2 value) and easier to use and implement with a lot of external support on Python forums et cetera, we decided to move forward with Linear (Poly) Regression with Feature Extraction.

Overall Hardware Organization

Our overall solution organization is divided as follows:

- Mechanical Hardware:
  - Consists of a 3D printed box with dimensions of 7.8cm x 6.25cm x 6.5cm to house the electronic hardware components.
  - Refer to "Mechanical Drawings" Section in Technical Analysis and Code section below.
- Sensors:
  - This MVP solely utilizes the MAXREFDES117# PPG Sensor from Analog Devices Incorporated
  - Refer to the "Electrical Diagrams" section in Technical Analysis and Code section below.
- Electronic Hardware
  - Consists of a Arduino UNO R3 Board (I2C), a green LED, a red LED, and the sensor.
  - Refer to the "Electrical Diagrams" Section in Technical Analysis and Code section below.
- Data Acquisition Device
  - The data acquisition device is the MAX30102 PPG sensor, which is attached to MAXREFDES117#.
  - Refer to the "Digitized Signals" section to understand how the data is acquired.
- Processing Unit:
  - Arduino IDE houses the code that processes all of the signals.
  - Refer to the "Digitized Signals" section to understand how the data is cleaned and processed.

Wiring Diagram

For the wiring diagram housed in the CAD-printed box, refer to the "Electrical Diagrams" section in the Technical Analysis and Code section below.

Software Operation Flowchart

1. Functional Box Diagram of PPG Data Acquisition (inputs and outputs)

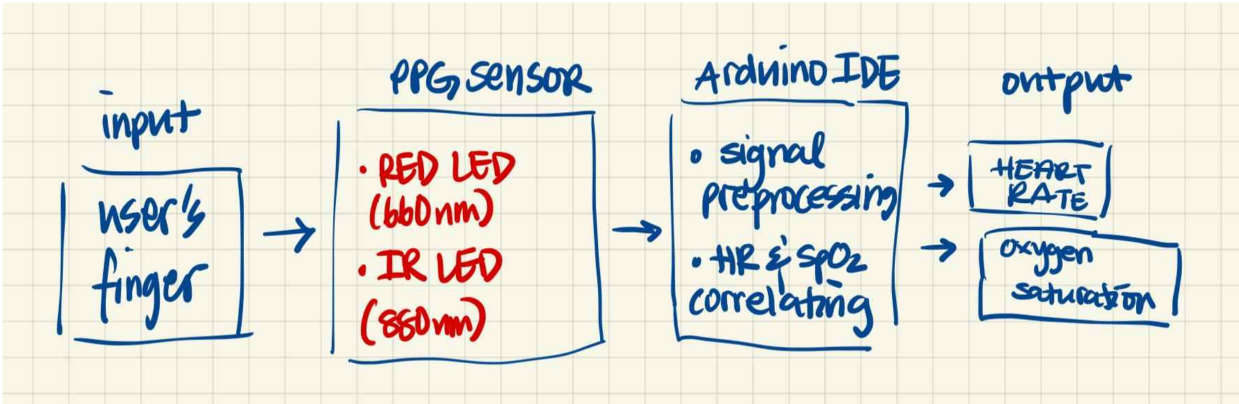


Figure 4. Functional Box Diagram of PPG Signal Acquisition on Arduino

For a more detailed explanation of the flow chart, please refer to the technical analysis portion.

2. Functional Box Diagram of ML Code (inputs and outputs)

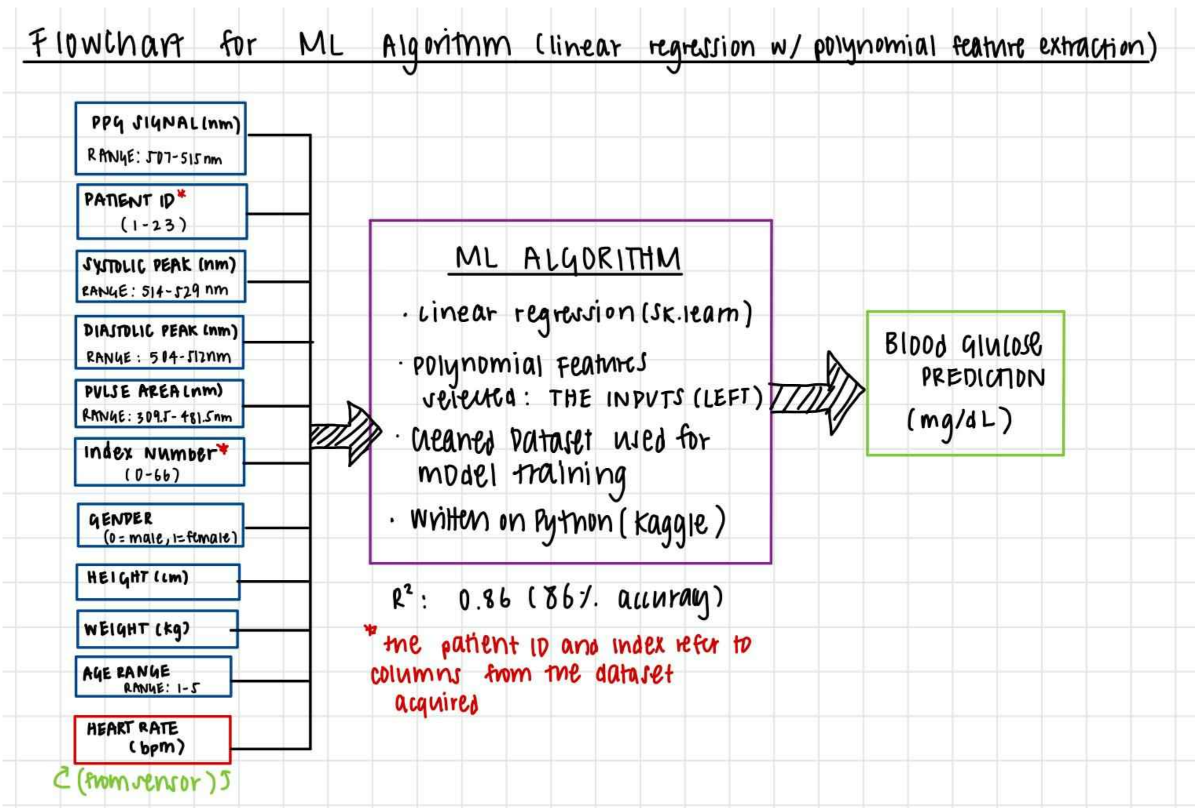


Figure 5. Functional Flowchart for ML Algorithm

Refer to Technical Analysis and Code Section for a clear description flowchart of the ML algorithm development process.

3. <u>Functional Box Diagram of LED Signalling System (inputs and outputs)</u>

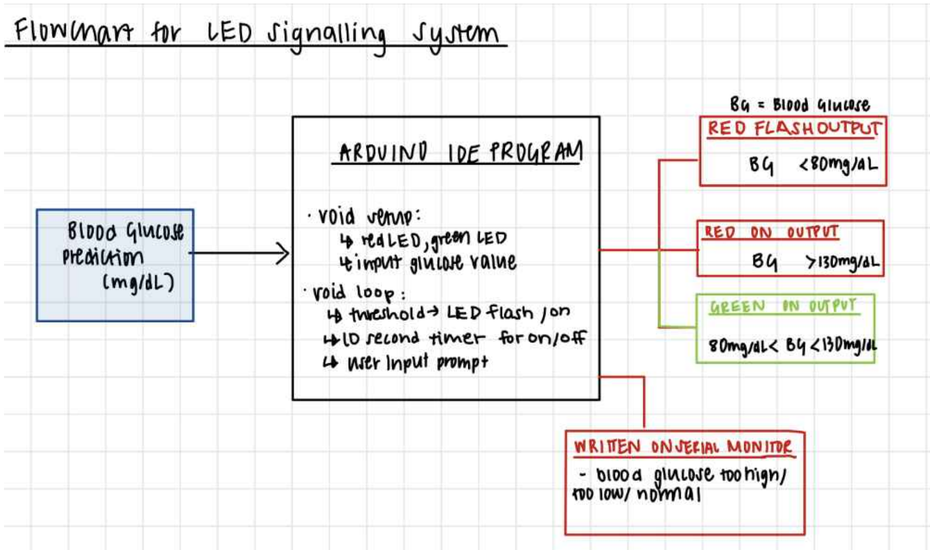


Figure 6. Functional Flowchart for LED Signalling System

Refer to technical analysis and code section for a more detailed flowchart of the development of the LED Signalling System on the Arduino IDE.

Digitized Signals

The MAXREFDES117# development board utilizes the MAX30102 PPG sensor to measure heart rate and oxygen saturation percentage. The sensor itself utilizes one red LED (660 nm) and one infrared (880 nm, IR). These lights would then bounce off the user's oxygenated or deoxygenated hemoglobin, which would come off as a different wavelength. This information will be captured by the photoreceptor on the MAX30102 before being processed in Arduino. The acquisition is set to be captured at 25 Hz in four seconds, which would result in one hundred data points per interval. Using the source code provided by user "MolecularD," the varying levels of wavelength can be processed to display the user's heart rate and oxygen saturation.

Specifically, the signals are preprocessed via mean-centering and baseline leveling. Mean-centering, or the removal of the DC component, is required as the raw data is received at the 10^5 range. The information from the arterial blood is specifically at the 10^2 range; thus, all 100 values must be adjusted as a result. Additionally, the data's index is adjusted from -49.5 to 49.5, from 0 to 99. Next, please refer to the figure below for the baseline leveling portion:

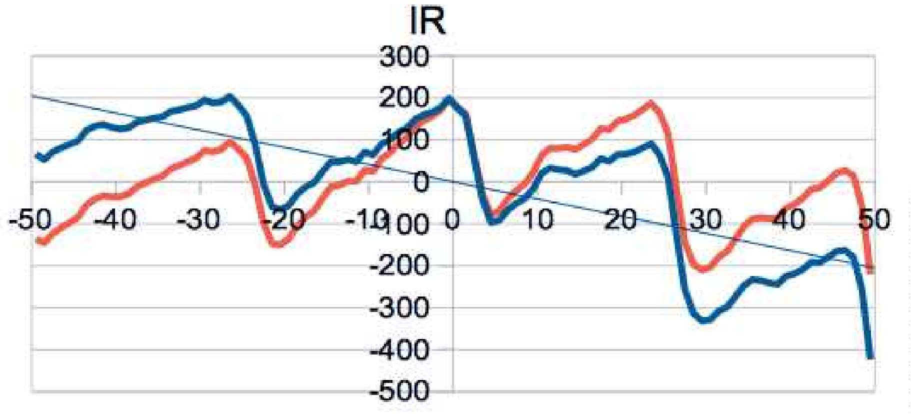


Figure 7. Measured IR values from index -49.5 to 49.5. The blue curves are the data points post mean-centering. The straight blue line is the baseline of the blue curve. The orange curve is the data point post baseline-leveling.

Baseline leveling is a required portion for the subsequent sections. This can be achieved by subtracting the blue curve from the blue line. As an aside, the mean-centering of the chart has left the equation of the graph to have an intercept of 0, which simplifies the math for the baseline leveling found in Appendix A. The resulting graph is an orange curve, which can now be used for the heart rate and oxygen saturation processing.

Using the relative autocorrelation formula found in Appendix B, the processed data can be interpolated to represent the user's heart rate. After calculating each autocorrelation value, the r_m values are rescaled by dividing with r_0 values. As a result, the final graphs point will yield values of r_m / r_0 , which yields the figure seen below.

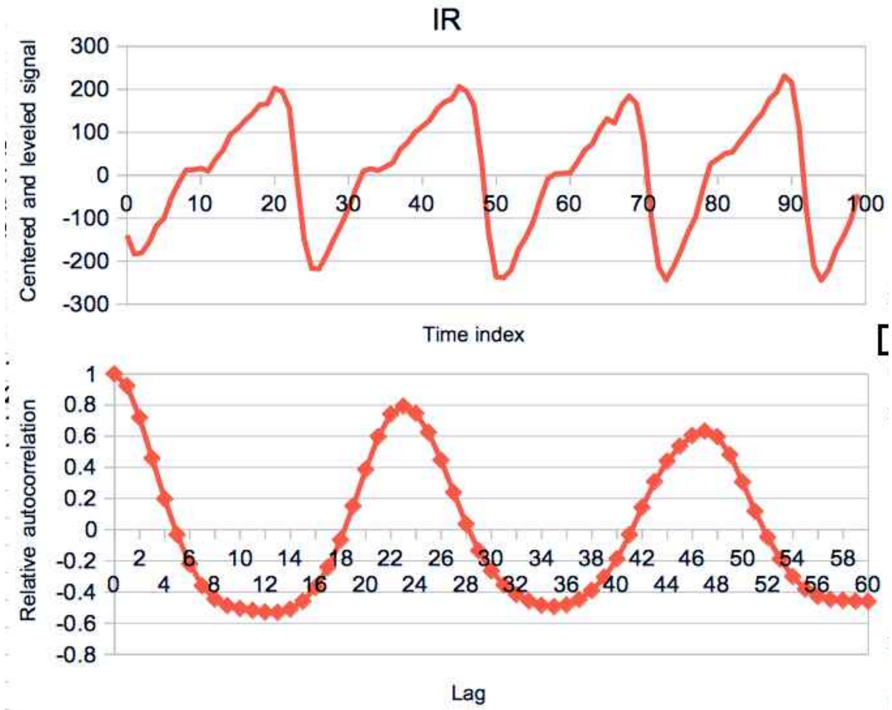


Figure 8. Raw preprocessed IR data (above) and relative autocorrelated IR data (below)

With all of this information, the heart rate is finally calculated by using the formula:

$$\text{HR} = \frac{(60)(25)}{m}$$

where m represents the lag of the signal at the first local maxima. So as an example, if the first maxima appears at m = 20, the heart rate would be:

$$\text{HR} = \frac{(60)(25)}{20} = 75 \text{ beats per minute (BPM)}$$

The heart rate formula assumes the initial heart rate value of 60 at m = 25 before beginning its autocorrelation search. In an effort to find the highest value, it essentially searches all the indexed lags until the local maxima is found.

Since the heart rate is the primary signal required for the Machine Learning model to operate, the digitization of the oxygen saturation can be found in the appendix.

Public Algorithms and Code

The code used to analyze the user's heart rate and oxygen saturation percentage was provided by Analog Devices Incorporated developers on GitHub here: Link. However, the "README" file references an additional article where user "MolecularD" created an improved algorithm. His code was used instead. Specifically, his code also uses existing Arduino libraries, which include "Arduino.h" and "math.h."

In reference to the ML algorithm, the sci-kit learn website was a great tool when developing the regression algorithm and training the dataset. The link to that website is here: Link. Additionally, several forums were helpful such as Stack Overflow while developing the model, especially while performing feature extraction.

ISO 13485 Standards

ISO 13485 processes were incorporated into the design and development of the GlucoPress MVP. For instance, design and development controls were documented, which include the functional requirements and features listed in the PDR, the functional requirements and design inputs/outputs, as well as risk analysis, which was a component of the functional requirements (under "Safety Compliance" in the "List of Functional Requirements" section). Furthermore, we incorporated documentation into our project, which included a lab notebook to list individual contributions each week as well as document challenges and pivots during the development process. This ensures that if troubleshooting is required, we are readily able to mitigate the issue and create a working and efficient device for diabetes patients. Additionally, after each hardware/software was incorporated, we would test the device to verify that each component connected to each other, that the design inputs and outputs connected to each other, and that it could be seamlessly integrated into one software in the future. Additionally, evaluation of the ML models (3 of them, to be exact, were done in order to determine the best-performing ML model). For instance, we tested the ML algorithm several times to make sure that it was outputting a blood glucose value that seemed sensible.

Other Engineering Standards

Other engineering standards were IEC 60601-1 (Medical Electrical Equipment—General Requirements for Safety), as once safety and risk tests occur, we ensure that the medical device operates safely without causing unnecessary harm to the patient. Additionally, we incorporated other engineering standards such as the IEC 62366 (Application of Usability Engineering to Medical Devices). We ensured that the design of the LED system was intuitive and did not require heavy explanation for the user to be able to understand the output of the MVP. Additionally, the ML code and Arduino code would use user prompts to ensure that when the user is inputting their parameters and their relevant information, they are able to follow the user prompts without much difficulty.

Accounting for Feedback from PDR:

Predicate "Device" (Proof-of-Concept Idea):
One of the main comments given in the PDR assignment was that there was no predicate device. While no exact device exists on the market that uses PPG sensors and ML algorithms to predict blood glucose levels, we found several articles that showed proof of concept using PPG sensors and ML algorithms.

In a journal article titled "Design of intelligent diabetes mellitus detection system using hybrid feature selection based XGBoost classifier," the authors mention designing a system using PPG signals and a XGBoost classification ML algorithm to predict blood glucose levels. In the article, they also detail that the dataset used to train their ML model had incorporated physiological parameters, thus we looked for a similar dataset with physiological parameters along with PPG signals and corresponding actual blood glucose levels to train the system.

Addressing Team Contribution
Additionally, another comment that was given was to detail each team member's contribution to each subsystem in the solution in this report. The contributions are detailed in the lab notebook below and at the end of this report. To summarize, Mya Cohen largely contributed to the LED subsystem, Boon Le was responsible for the hardware/electrical components and the PPG signal to HR output subsystem, and Karishma Rohatgi was responsible for the ML algorithm subsystem.

Functionality Demonstration/Demonstration Video

The link to the final presentation slides can be found here: Link
The link to the video can be found here: Link

Technical Analysis and Code

Electrical Diagrams

The diagram below was created using Fritzing and Microsoft PowerPoint. Microsoft PowerPoint was only used as an editing tool to insert the MAXREFDES117#. Fritzing was used to display the Arduino UNO R3 and all the electrical components, including the wires and LED. All wires have been color-coordinated to match the component that it is attached to. Blue wires are attached to the sensor, red wires are connected to the red LED, and green wires are connected to the green LED. Please note that the wires on the actual device are not the colors shown in figure 9. The board would be powered by a computer via USB-B, and the adapter is located next to the reset button in the form of a silver box.

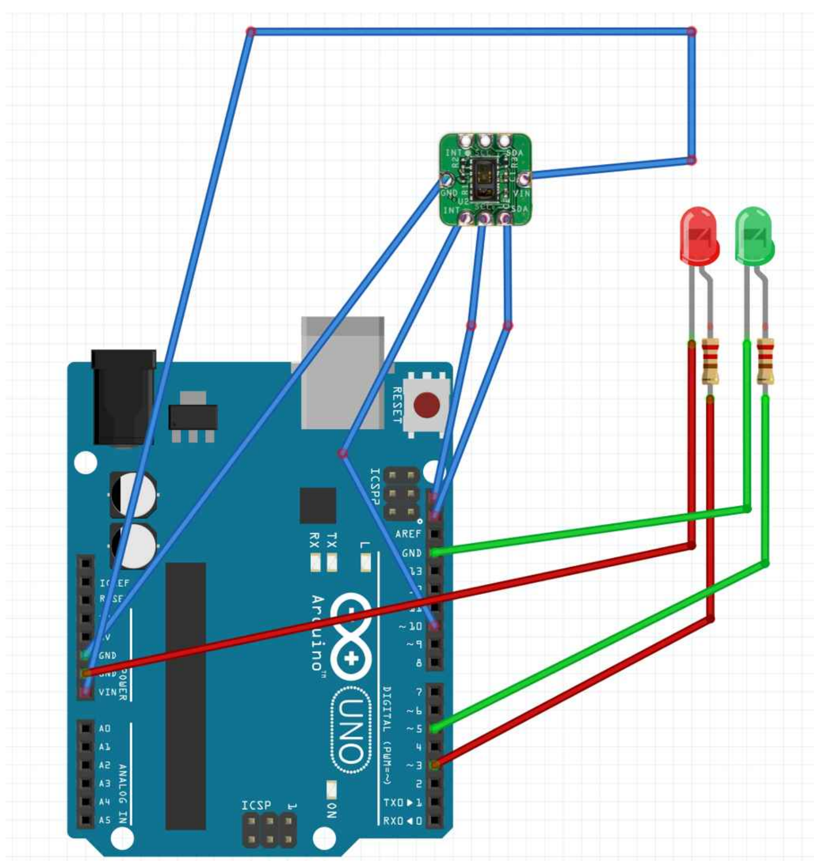


Figure 9. Electrical diagram for GlucoPress created using Fritzing and PowerPoint.

The I2C is split into three sections: power, analog in, and digital. The specific connections to the I2C are:

Red LED: The anode/long head is soldered to a 220 Ω resistor which is also soldered to a wire that is connected to pin 3 on the I2C. The cathode/short head is connected to the second GND (ground) pin on the power side of the board.

Green LED: The anode is soldered to a 220 Ω resistor which is also soldered to a wire that is connected to pin 5 on the I2C. The cathode is connected to the ground pin on the digital side of the board.

MAXREFDES117#: The sensor has five distinct connections: VIN, GND, SCL, SDA, and INT. Only one set of SCL, SDA, and INT connections are needed for this project. VIN is connected to VIN on the power side of the board. GND is connected to the first GND pin on the power side of the board. SCL and SDA connected to the SCL and SDA ports on the digital side of the board. INT is connected to port 10 of the digital side.

The final electrical diagram matched closely to the initial design of the circuit, which can be found in the Appendix C. Due to the fact that the board was delivered with female pin adapters pre-soldered, the end product does not use a protoboard to make connections. Although the electrical diagram does look similar to the one in figure 9, it does not account for the fragility of the soldered pieces. The connection between the resistor and wires were bound together by electrical tape to ensure that it would not break during assembly. This visual can be observed below in figure 10.

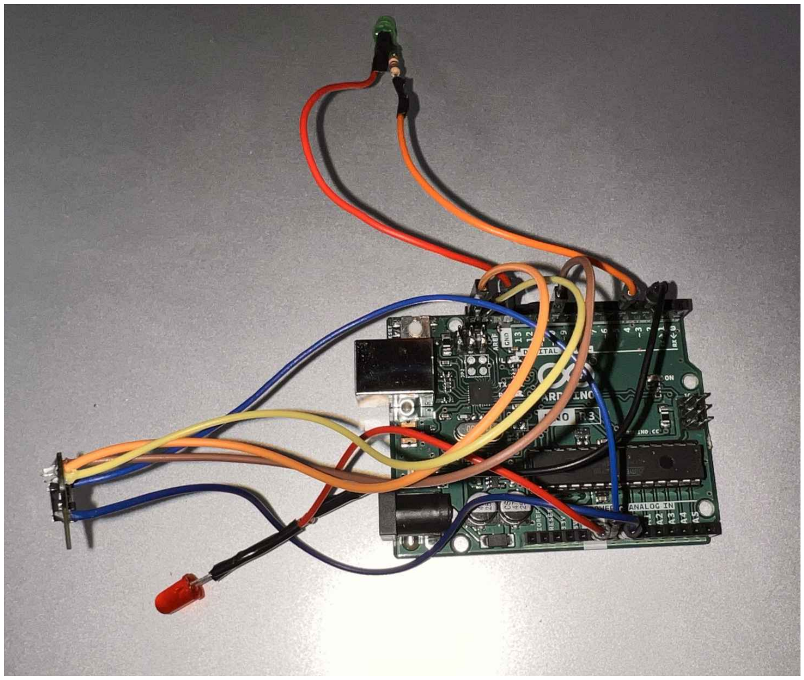


Figure 10. I2C with all components connected. The resistors specifically had to be bound by electrical tape to ensure that the wires didn't break during assembly.

Mechanical Drawings

The mechanical portion consists of the housing to store our I2C, sensors, and LEDs. The brainstorming process for the case can be found in Appendix D. But the most practical design ended up being a box that would contain the electronics in figure 9. The box can be separated into a top and bottom piece, where the bottom would secure the I2C while the top allows for the user's interactions. Since the I2C is the largest component of the device, the case needs to revolve around these dimensions. In figure 12, the measurements of the Arduino UNO were taken, and the bottom case's dimensions were decided to be 62.5 mm x 78 mm x 65 mm. The top of the case needed to be secured, which led to the making of clearance holes that aligned with the I2C's holes on the bottom case. The top case would then have three pilot holes for the 4x40 screw, allowing everything to be sealed together in a compact but durable design. One pilot hole had to be removed due to the abundance of components in a region of the board (where the reset button is). Thus, a pilot hole could fit, but longer screws would be needed. For the sake of consistency for the user, this pilot hole was just removed to ensure all the screws were the same. The top case needed to account for the adapters to the computer, sensors, LEDs, and access to the reset button. Holes were all measured and cut out to accommodate these items. Due to the design of the LEDs, they could rest at the top of the hole without being secured. But the reset button needed the addition of a pillar to allow the users to press it. The pillar's base matches the dimensions of the reset button's housing to give the base balance so that it would not fall over.

Lastly, the PPG sensor had to be secured to the top case, as movement could affect the device's effectiveness. Electrical tape served this purpose and ensured the user had accurate readings.

The fabrication process consisted of three distinct phases. First, multiple brainstorming sessions were held to come up with an initial design for the case. Dr. James Yoo of USC's Innovation Space worked closely with the case's progress to ensure that the design met the necessary criteria. Following these sessions came the use of SolidWorks to draw up all the CAD files. The second phase was the test prints. The material that was used was red PLA filament printed using Ultimaker's CURA application, which was present in the 3D printing lab of Denney Research Center's (DRB) basement.

The prints were conducted using premade settings to ensure speed and build quality with supports on. The PLA was tough enough that drops would not break the case and appeared to be sufficient for the purposes of the project. 4 x 40 crews were the fastening choice, as it would hold everything together, and its fit closely matched the Arduino UNO's holes. The initial print consisted of just the bottom case to determine the parameters of the I2C. It ended up being too small, and a 5mm revision was made on each border to fit the Arduino UNO. The second print yielded clearance holes too small and would not allow the screws to properly thread through. The holes for the LED had the same issue, and the wires could not fit through. The hole for the USB-B connection was also not tall enough. The final print had addressed all these issues, and everything was able to fit in the end.

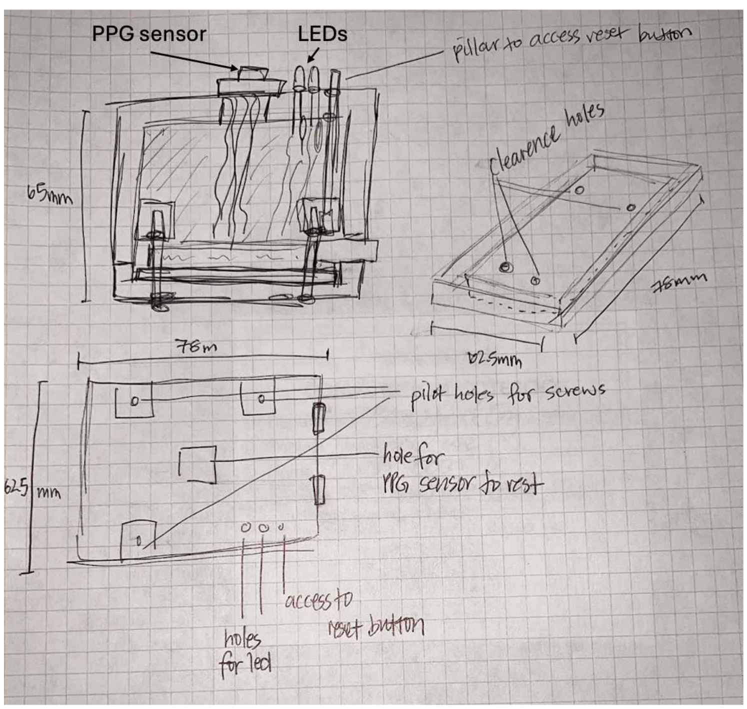


Figure 11. Initial Design of GlucoPress Case

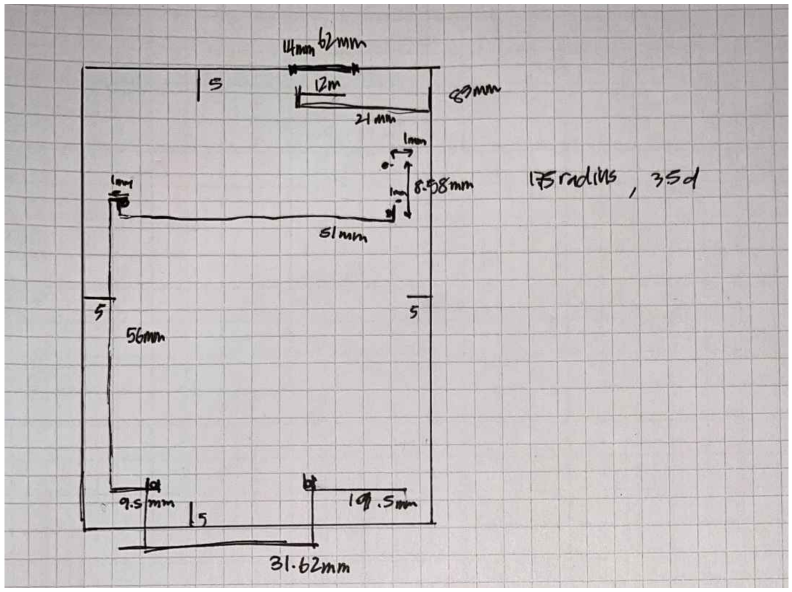


Figure 12. Measurements of GlucoPress's case and the proposed fastening system.

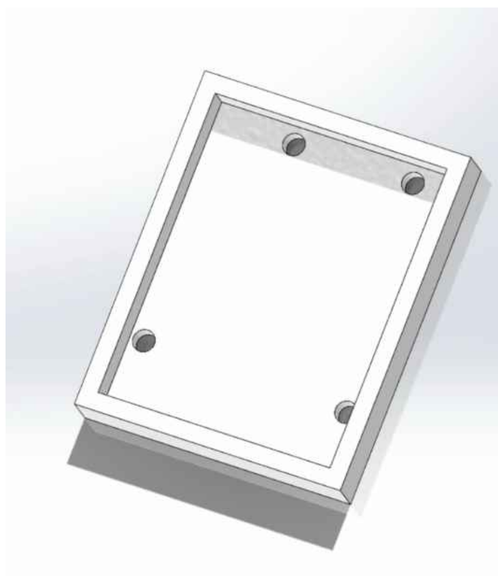


Figure 13. The bottom case. Features clearance holes for 4 x 40 screws that align with the holes on the Arduino UNO R3.

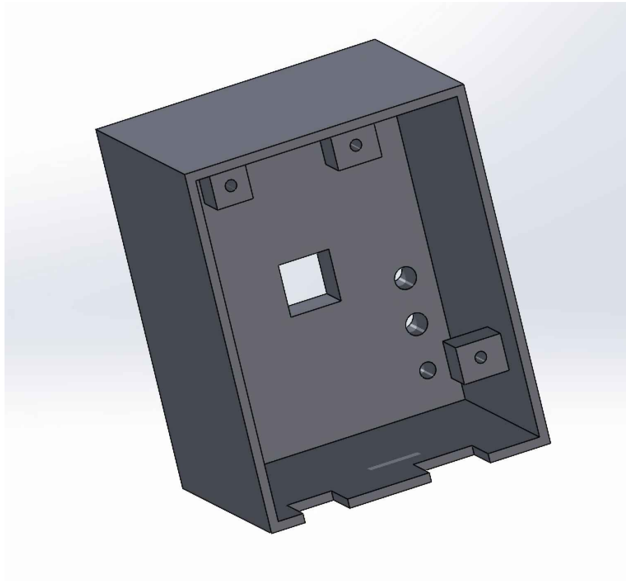


Figure 14. The top case. Features two holes cut out on the side of the case for the USB-B and power adapters, three pilot holes, two LED holes, one hole to access the reset button, and a square hole in the center for the PPG sensor to rest on.

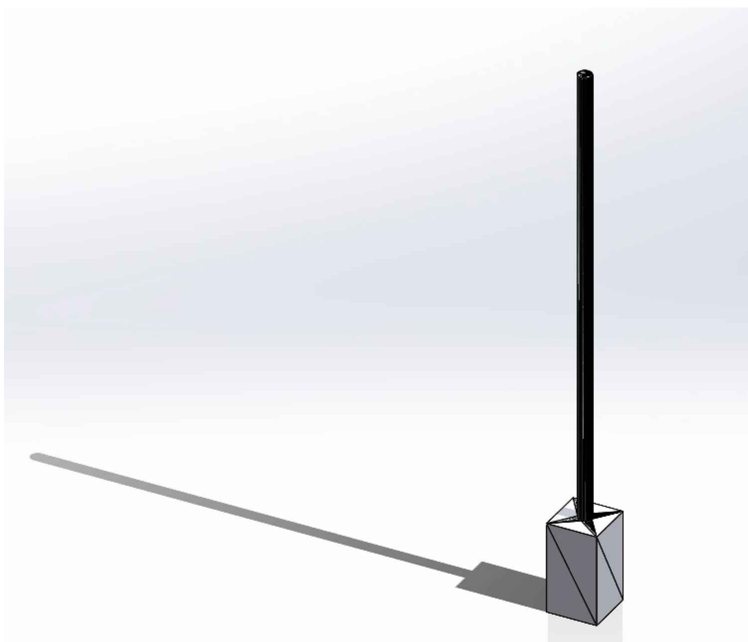


Figure 15. Pillar to access the reset button. The base is the same dimensions as the reset button's housing to ensure balance.

It should also be noted that an open source Arduino UNO R3 case was used to verify the measurements. The case can be found here: Link

Power Budget

To reiterate, GlucoPress utilizes the Arduino UNO R3 I2C, a red LED with a 220 Ω resistor, a green LED with a 220 Ω resistor, and the MAXREFDES117# PPG sensor. The I2C consists of the ATMEGA16U2-MU(R), the ATMEGA328P-PU, a KPT-2012SGC (green LED), and a 4xKPT-2012YC (yellow LED). The maximum current for each component is 261 mA, 410 mA, 5.6 mA, and 8.7 mA, respectively. This information is found using the data sheet in Appendix G, page 8. The power consumptions of the 5mm red and green LEDs draw about 20 mA of current according to Adafruit's information sheet (Appendix G) page 2. Although, the LEDs are not running the entire time, only when the user has inputted their blood glucose value in the ML model. Next, the PPG sensor utilizes 1.5 mA, found in their datasheet (Appendix G) page 3.

In all, the maximum total current consumption of the device is:

$$\text{Current (mA) consumption} = 261 + 410 + 5.6 + 8.7 + 20 + 20 + 1.5 = 726.8 \text{ mA}$$

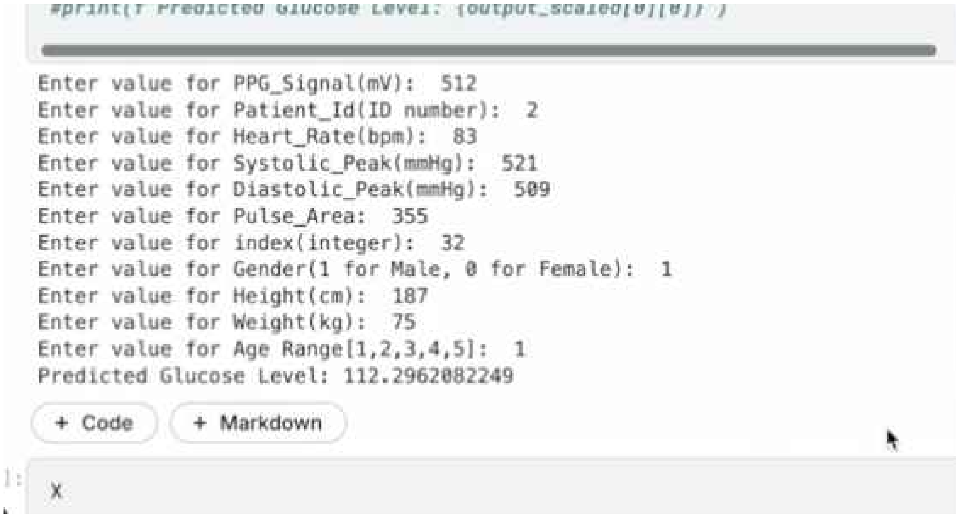
While the device does not utilize a battery, it does require a laptop to operate. During the MVP process, the primary laptop used was the Macbook Pro 14-inch with the Apple M1 Pro and 16 GB memory. This model comes with an estimated 6100 mAh rating. As a result, if the device was running constantly with the LED always on, it would take about eight hours to completely drain the laptop's battery–assuming the laptop had no other power consumption. Thus, eight hours is the absolute minimum amount of time that the device can run using this laptop. It can likely run for much longer since the LED won't always be on.

Software Code:

<u>PPG Sensor Code:</u>
The link to the folder containing the PPG signal processing is here: Link

<u>ML Model:</u>
The link to the PDF of the main script for the ML Model is here: Link
Furthermore, a link to the dataset used to train the ML model is linked here: Link

A screenshot of the system outputting a predicted blood glucose value once all the parameters are inputted is below:



```
#print(f'Predicted Glucose Level: {output_scaled[0][0]}')

Enter value for PPG_Signal(mV):  512
Enter value for Patient_Id(ID number):  2
Enter value for Heart_Rate(bpm):  83
Enter value for Systolic_Peak(mmHg):  521
Enter value for Diastolic_Peak(mmHg):  509
Enter value for Pulse_Area:  355
Enter value for index(integer):  32
Enter value for Gender(1 for Male, 0 for Female):  1
Enter value for Height(cm):  187
Enter value for Weight(kg):  75
Enter value for Age Range[1,2,3,4,5]:  1
Predicted Glucose Level: 112.2962082249
```


Figure 16. Screenshot of ML Code with Predicted Blood Glucose Level

<u>LED Signalling System Code:</u>
The link to the LED classification code is here: [Link]

<u>How-to-Run document</u>
How-to-Run document is provided here for *each* of the programs used in this MVP: [Link]

The YouTube video showcasing our demonstration of the MVP is linked here: [Link]

Software Architecture:

The block diagram below for the software architecture including each layer is shown. By following this diagram flow, our MVP software system can be better understood.

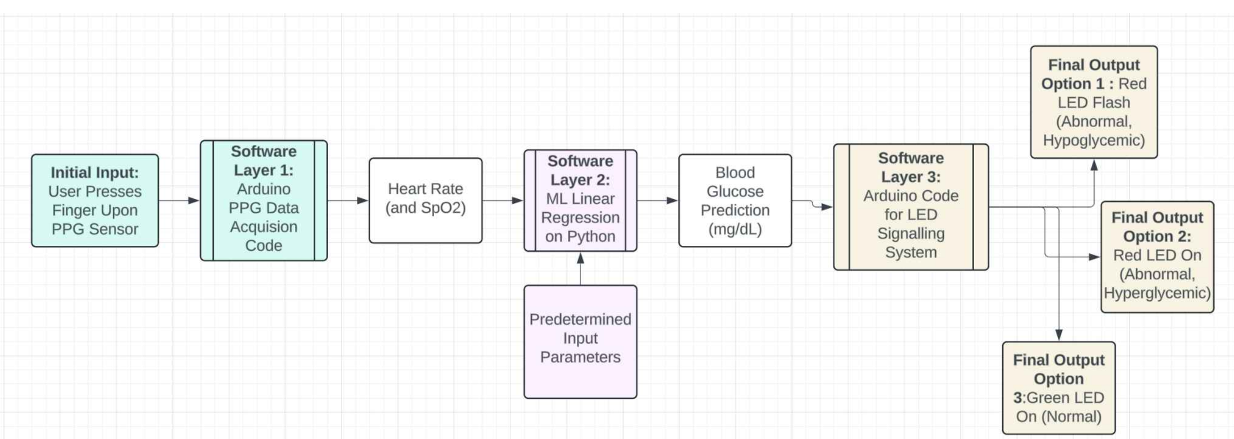


Figure 17. Software Layer Block Diagram across whole MVP (Design Pattern)

This system utilizes a three-layered architecture, with each subsequent layer dependent on the output of the layer before it. The initial layer is the first Arduino program, which runs a code that enables accurate PPG signal data acquisition from the user's fingertip and outputs a heart rate (hr) in bpm. Next, the HR that is recorded from the Arduino program will be inputted into the second software layer, which contains an ML algorithm that utilizes linear regression with polynomial feature extraction to take in the HR and other predetermined parameters and output a predicted blood glucose value (in mg/dL). The third and final layer is the Arduino program that takes the predicted blood glucose value and outputs one of three possible blood glucose scenarios determined by medically defined thresholds of abnormal and normal glucose levels via an LED signalling system. This program will notify users of what category (normal or abnormal (hypoglycemia or hyperglycemia)) their blood glucose value resides in via an LED signalling system using a red or green LED.

Use of the MAXREFDES117# with the Arduino UNO R3 microcontroller board
The PPG sensor (MAXREFDES117#) is connected to the Arduino microcontroller, and when the code provided by the developer (Analog Devices Inc.) is run, the code will convert the incoming PPG signals from the fingertip of the user to real-time vital parameters such as HR and SpO_2 among others. The PPG sensor is thus controlled by the Arduino UNO R3 microcontroller.

Use of the ML model on Python (run on a Kaggle notebook) for blood glucose prediction
The output of the previous software layer will now be one of the input parameters used to predict a blood glucose model using the ML model. The ML model was trained with a publicly available dataset and the main ML algorithm used is a linear regression from the sci-kit learn library, with added feature extraction to improve the accuracy of the blood glucose prediction. The ML model would not have been possible without scikit-learn modules and libraries, as well as the pandas and numpy libraries to compile the raw large dataset.

Use of the red and green LED bulbs with the Arduino UNO R3 microcontroller board
The final layer takes the output of the previous layer and uses it as the input for the LED signalling system. The red and green LED bulbs are controlled and connected to the microcontroller board and the code will ask the user to input the predicted blood glucose level. Once the user inputs their predicted blood glucose level in the serial monitor, the Arduino software code will set the value into a category based on the abnormal/normal blood glucose thresholds set, and a subsequent LED (either the red or green) will light up based on what category their blood glucose value resides in. This will be the final output shown to the user for the GlucoPress MVP.

Subroutines and External Routines:
Within the software architecture, there can be 3 layers with the first two layers being subroutines, and the third and final layer being the external routine, as the output of this software layer is the

final output delivered to the user which is the pertinent LED turning on depending on what category their blood glucose level resides in.

Table 5. Subroutines and their corresponding functionality, inputs, and outputs.

Subroutine	Functionality	Input	Output
Software Layer 1: Arduino Code for PPG Sensor Data Acquisition and Conversion to HR and SpO_2	This subroutine converts PPG signals from the patient to vital parameters.	The patient presses their finger upon the PPG sensor.	A heart rate value (in beats per minute) and SpO_2 value is displayed on the Serial Monitor.
Software Layer 2: Machine Learning Linear Regression with Polynomial Feature Extraction Algorithm For Blood Glucose Prediction	This subroutine converts a heart rate along with other pre-calibrated parameters to a blood glucose value.	Heart Rate from the previous layer along with other pre-calibrated parameters (refer to EDR section for more details on inputs).	A predicted blood glucose value.

Table 6. External Routines and their corresponding functionality, inputs, and outputs.

External Routine	Functionality	Input	Output
Software Layer 3: Arduino Code for LED Signalling System	Informs the user of their blood glucose status (hyperglycemia, hypoglycemia, normal)	Predicted blood glucose value from previous subroutine/layer.	LED On/Flashing: Red LED Flashing: Indicates hypoglycemia Red LED On: Indicates hyperglycemia Green LED On: Indicates normal level Note: LED's will turn off after 10 seconds have elapsed.

GUI Design:

Not applicable to this MVP.

Block Diagrams for Software Components:

ML Algorithm

To explain how the ML Model was trained and how it became a functioning component of our MVP please refer to this flowchart/block diagram for a clear overview.

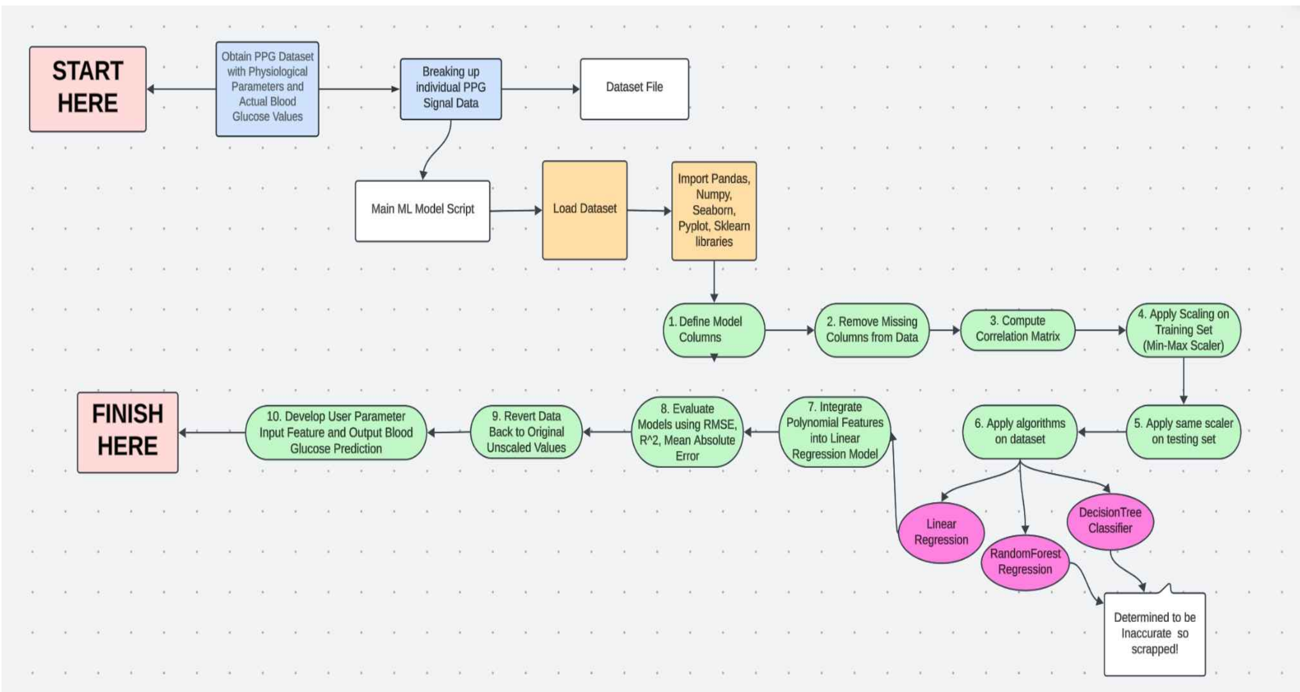


Figure 18. ML Algorithm Development Process

While this functional flowchart indicates the steps to reproduce the ML model given the PPG dataset, it should be noted that 3 main ML algorithms were evaluated using performance metrics. In the end, we chose to continue with Linear Regression with the selected polynomial features that are represented as the inputs. When evaluating the linear regression model (w/ features selected) compared to the other ML models via the R^2 performance metric, it was clear that the R^2 value of 0.86 (resulting from the linear regression model) was the baseline R^2 and no other ML models matched that R^2 value very closely. Thus, any developments made to the ML model and code in the future should take into account the R^2 baseline value of 0.86. The steps for training, testing, and evaluating the ML model are as follows:

1. Obtain and Process Dataset

2. Break up each PPG Signal Sample into Individual PPG Values Per Patient and Per Index, and associate these PPG values with the physiological parameters included in the dataset.
    a. This is necessary to increase the number of “data points” we have and associate the
3. Load Dataset into the Main Script
4. Define Model Columns (input parameters)
5. Remove Missing Columns from Data and Remove Outlier Data Points
6. Compute Correlation Matrix, and Drop Columns with Low Correlation
7. Apply Min-Max Scaler on Training Data
    a. This is necessary to scale each data point between 0 and 1 which makes the ML model easier to train
8. Apply Min-Max Scaler on Testing Data
9. Apply Algorithms on Dataset
    a. Linear Regression
        i. For the purposes of this training model, the train-test split was 80% data from the dataset used for training, and 20% of the data was split for testing.
    b. RandomForest
    c. DecisionTree
10. Apply Polynomial Feature Transformation on Linear Regression Model
11. Evaluate Models using Performance Metrics such as:
    a. Mean Squared Error (MSE)
    b. RMSE (Root Mean Squared Error)
    c. Mean Absolute Error
    d. R^2 Accuracy Value
12. Determine best R^2 Value and Scrap Low R^2 Value ML Models
    a. Note: refer to Appendix E for an explanation of the performance of the other ML models tested.
13. Write a Simple User Input Code to Allow Users to Input Their Physiological Parameters and any Relevant Inputs (Such as their Heart Rate (bpm) from the 1st Software Layer (PPG Data Acquisition and Conversion on Arduino IDE)

<u>LED Classification</u>

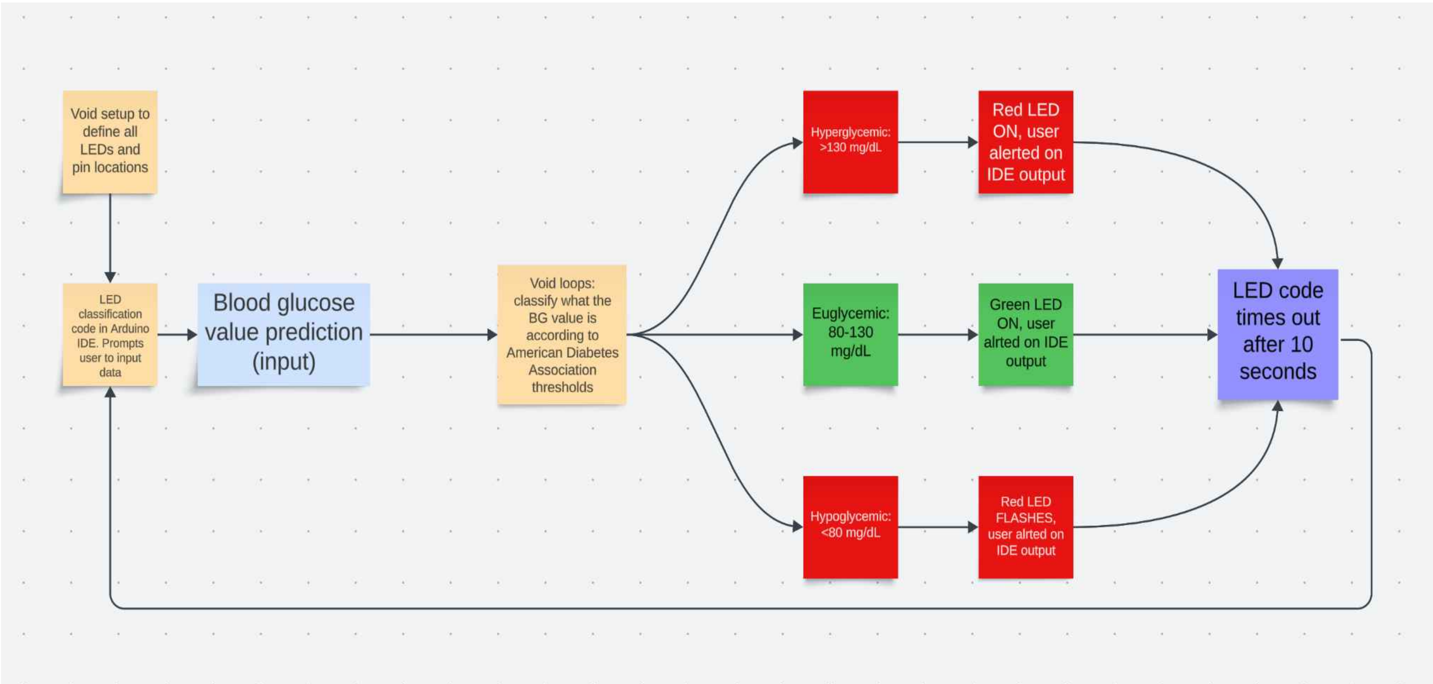


Figure 19. LED Signalling System Block Diagram

The LED classification code begins with setting up the location of the LEDs in the void setup. Then, the user will be prompted to input their blood glucose value from the ML algorithm. Once the data has been inputted, there are three possible scenarios that can occur:

1. The entered value is greater than 130 mg/dL, which means they are categorized in the hyperglycemic group. The red LED will turn and stay on while being accompanied by a text output stating that their blood glucose value is too high.
2. The entered value is between 80 and 130 mg/dL, which means they are euglycemic. The green LED will turn on and stay on while being accompanied by a text output stating that their blood glucose value is in the normal range.
3. The entered value is less than 80 mg/dL, which means they are hypoglycemic. The red LED will turn on and off (i.e., flash) while being accompanied by a text output stating that their blood glucose value is too low.

Each scenario will last ten seconds before timing out and returning back to prompting the user for their blood glucose value.

Microcontroller Code Functional Block Diagram:

The output of the MAXREFDES117# captures all wavelengths via its photodetector. But, the Arduino UNO R3 and developer code specifically sets a 25 Hz per four second sampling rate, which yields 100 data points per interval. As a result, the sensor initially captures data as an analog signal but converts it to digital via an analog-to-digital converter (ADC). Since the PPG sensor is required to be plugged into the user's computer, data exchange starts when the user answers the prompt to begin data acquisition. When this start has been detected, the I2C is immediately receiving digital signals from the sensor. Although, due to how the signals are received and processed (refer to "Digitized Signals"), there will not be a reading unless sufficient data points are acquired. Once sufficient and quality data is received, the I2C can spit out the outputs for the ML algorithm to consume.

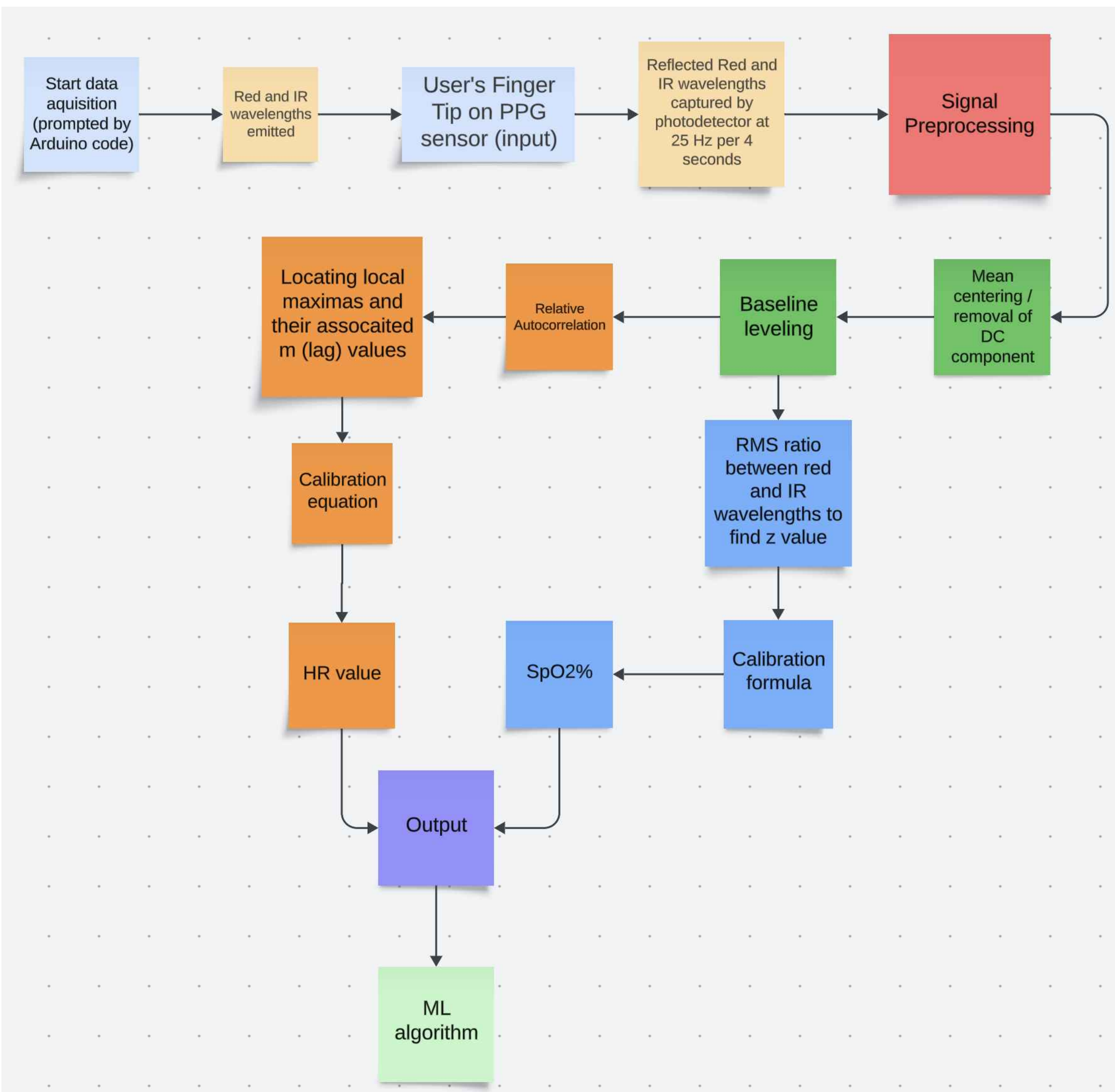


Figure 20. Microcontroller Code and Overall Functional Block Diagram

Strengths and Weaknesses

Overall, GlucoPress came together with working components. The hardware's assembly, sensor to I2C digitization, ML algorithm, and classification are all functional. Currently, there is a lack of non-invasive blood glucose monitoring systems currently on the market. There was only a proof-of-concept article by Prabha et al. to work off, so it was an incredible challenge to produce a device that could match the accuracy of a traditional method.

But not all battles come without shortcomings. One weakness of the ML algorithm is that while polynomial feature extraction was useful in boosting the accuracy of the ML algorithm, it requires a total of 11 input parameters in order to output a reasonable and sensible blood glucose value. The MVP as it exists currently requires the user to input the 11 parameter values each time they want to obtain a predicted blood glucose reading. The algorithm would not be able to generate a sensible blood glucose value without it, but it may be too cumbersome for the user to input their parameters in order to generate one blood glucose output. This effectively becomes an inconvenience for the user; thus, further iterations of this MVP must address and attempt to reduce the MVP parameters while improving or maintaining the R2 accuracy. But, as it stands, the device's 86% accuracy is still an incredible feat.

Another tough issue in this project was initially getting PPG signals from the MAXREFDES117# sensor. Due to the amount of noise being generated from the PPG signal, it was hard to obtain an accurate PPG signal value to display on the Serial Monitor. Thus, to overcome this issue, we chose to pivot the input of PPG Signals and change it to a HR. We did heavy research and found a code from the developer themselves that would be able to generate heart rate (HR) and several other parameters. Then, because HR was also one of the parameters present in the ML dataset, we decided to change the output of the PPG Data Acquisition component of our project from PPG Signals to HR, and the HR would essentially feed into the ML model.

Regarding the hardware, a lot of components were incredibly fragile despite a careful solder job. Specifically, the LED-to-resistor-to-wire connection was incredibly brittle. It had to be handled and maintained in an upright position. Minor bends would compound and eventually lead to the connection breaking. To prevent this from breaking, electrical tape was used to protect the signal. But electrical tape was a temporary fix, as it isn't a super rigid material. Heat shrink would have been a great but permanent alternative. At this point in the project, it was not an option as connections constantly had to be rechecked to ensure that they worked. Furthermore, the PPG sensor was secured solely by electrical tape. Assuming the user was incredibly rough in storing the device, the tape could eventually fall off, and the device would not be sealed anymore. Regardless, the portability is still a major strength of this design despite its fragility.

Future Developments

The following recommendations are listed in descending priority.

Since our current device is an MVP, there are plenty of improvements that could be made before the final version is iterated. Regarding the software, there needs to be a way to integrate all three programs into one display. Currently, the user has to shuffle between three different tabs for data acquisition/processing, ML prediction, and classification. Having one seamless window would make it infinitely easier for the user to find their glycemic status. Additionally, this new code should be accompanied by a GUI to increase this ease of use. Signal acquisition only reports heart rate values at specified time points with the developer's code. If it is possible, it would be ideal to take a user's heart rate for some time (e.g., one minute) and take the average of those values. That way, the heart rate may be more accurately captured. Right now, the Machine Learning algorithm requires 11 user inputs. There should be a way to decrease the amount of parameters while still maintaining or improving the R2 value of the prediction. Lastly, the device should only require the user to place their finger on the sensor, and the appropriate LED should light up. In other words, there should not be any other action required. The code should contain a loop that has the user's data pre-filled and immediately classifies their glycemic group with just the automated input of their heart rate.

Regarding the hardware, a mounting system to secure the PPG sensor would be ideal. As it stands, the MAXREFDES117# is held down on just electrical tape. The LEDs could utilize a "snap" fit to ensure that it is held in place with no movement. The reset button pillar could be girthier so that it would not wiggle around. The pilot holes inside the top case should be cleared from the Arduino UNO R3. There are some minor clearance issues that have caused the case to not incorporate all four screw holes. Lastly, it would be worth it to move to a battery-powered device to increase portability for the user. Since it would now separate the user from their computer, a cloud-based signaling system would be required to transmit data.

Environmental Impact

Since this project is done in the U.S., the environmental analysis will be done using U.S. guidelines for disposing of trash. Of the completed BOM in the section below, the device does contain hazardous waste that fit the criteria of the U.S. Environmental Protection Agency (waste code A1181). As a result, all electrical components (I2C, PPG sensor, LEDs, wires, and resistors) at the end of life should be disposed of with reference to the local ordinance's direction. In Los Angeles (L.A.) specifically, these items should be disposed of at dedicated e-waste stations. The other components (electrical tape, screws, and filament) are considered non-toxic but should be disposed of with reference to the local ordinance's direction. In L.A., electrical tape is considered general household waste and can be thrown in the trash. PLA filaments are plastic and should be taken to the proper graded recycling center. Screws should be taken to scrap metal centers for recycling as well.

Bill of Materials (BOM)

The complete BOM can be found in this Link, which provides a comprehensive view of what Group 8 ordered during the R&D phase. The final items and their cost required for the MVP are listed in the table below. Please note that this assumes that every part works perfectly and there is not a need for a backup. Additionally, some components, such as the LEDs, can only be bought in a large pack. The cost listed is that of the entire pack rather than just one item.

Table 7. BOM with final parts needed for the GlucoPress

Component/Part	Cost (USD)
Arduino UNO R3	$27.60
MAXREFDES117#	$23.82
Red LED	$4.00
Green LED	$4.00
220 Ω Resistors	$5.99
Electrical tape	$2.99
4 x 40 screws	Free (provided by USC Innovation Space)
4 x 40 drill bit	Free (provided by USC Innovation Space)
Wires	$15.95
Soldering equipment (iron and solder)	Free (provided by USC Innovation Space and DRB 351)
3D printer (Ultimaker 5)	Free (provided by USC Innovation Space)
PLA Filaments	Free (provided by USC Innovation Space)

In total, the finalized BOM was $84.35. This does not reflect the cost of taxes or shipping.

Verification

Overall, the MVP does meet the unmet needs for group 8. The project aimed to address the lack of accessible, easy, and convenient methods for working-age adults in the U.S. to monitor their diabetic health. To achieve this challenge, we were able to make a list of functional requirements addressed in the “Use Cases to MVP Requirements” section. In short, the use of PPG signals led to a noninvasive acquisition method to acquire a user’s heart rate data. Then, using a Machine Learning model, their blood glucose value could be predicted. Lastly, with an LED classification system, the user’s glycemic status could be properly categorized. This device is all held together in a convenient, lightweight, and portable design that is intended to be plugged into a user’s computer. With a working MVP, a patient willingly volunteered their blood glucose value to verify our model was functional. It cannot be stressed enough that this was *entirely* out of this person’s free will, and there was no conspiracy to coerce this user to provide their data.

While working on the project, our team experienced some boundaries that affected the MVP being realistic. Originally, we aimed to have the device be a portable and wearable device that constantly took data points from the user throughout the day. However, considering the size of our case and the lack of compatible hardware, this was just not realistic. As a result, there was a pivot to have a stationary device instead. Another constraint included the time allotted for this project. Given only a semester, it was extremely difficult to make everything work seamlessly.

A logical path to expand the MVP to include more functionality would be to acquire a different PPG sensor that could allow for cloud-based communications. This way, the user would actually be able to monitor their blood glucose levels throughout the entire day rather than when prompted. To support the device being wireless, there needs to be a battery connected as well so that the device could be powered throughout the day. But, given that this is a minimum viable product, group 8 has successfully achieved its goal to read and classify a user’s glycemic status. ISO 13485 was a forefront standard used during the future functionality brainstorm as it ensures that users have an easy and convenient way to utilize their product.

Technical Difficulty

The technical difficulty of this project is currently ranked at a 7 out of 10. The MVP used an Arduino UNO R3 microcontroller board, a PPG sensor, 2 LED bulbs, and a complex ML algorithm written in Python. The hardware was quite complex, having two prototype CAD prints and one final CAD print to house the microcontroller, the sensor, and the 2 LED bulbs, and designing a unique reset button pillar to allow the user to reboot the microcontroller when necessary. The PPG sensor was soldered with male-to-male end wires, and the LED bulb's anode was soldered to resistors. The latter soldering was quite difficult due to the material of the resistor wire, and thus several instances occurred where the soldering would not hold and the wires would fall apart. This would thus require an individual that has had plenty of soldering experience to get a reliable connection. The CAD case files were intricate as there needed to be perfect alignment for the screws to properly fit. Setting up the case should be straightforward following the proper manufacturing of the components. The PPG sensor measured blood volume changes in the skin by detecting changes in the light reflected from the skin. Like most sensors, the initial PPG data acquisition encountered a lot of noise, and it was difficult to obtain a working signal from the sensor on the Arduino IDE. Thus, we pivoted to recording HR using a source code from the sensor's developer, and we were able to determine a heart rate and SpO2 level that seemed sensible. The PPG sensor is considered one measurement and results in SpO2, HR, temperature, correlation, time, and ratio measurements. As we sourced this code from the developer, this portion of the project was not as difficult as the other components. We only used SpO2 and HR in the rest of the MVP. In order to use the HR to obtain a predicted blood glucose level, we inputted the HR along with several other parameters into a complex ML code. The ML code takes in 11 input parameters, such as PPG Signal (nm), Systolic Peak (nm), Diastolic Peak (nm), Pulse Area (nm), HR (bpm), Weight (kg), Height (cm), Age Range (1-5), Patient ID, Index, and Gender, and outputs one parameter, which is the predicted blood glucose level based on the dataset that the model was trained from. It uses a linear regression technique for the model training but includes feature selection and extraction based on a correlation matrix that was used in order to evaluate feature correlation with the dataset. This feature extraction portion of the ML algorithm was the most complex, as it involved several analyses of outliers using the Boxplot method and required the removal of data and outliers in order to make the model as accurate as possible. The linear regression used a 20% test and 80% training split in order to generate a high enough R2 value to be feasible for implementation in the rest of the MVP. Several tests were performed on the ML models, and evaluation of several models, such as the RandomForest Regressor and the DecisionTree Classifier, was done. For the final component of our MVP, we designed a LED system that would alert the user of what their blood glucose status was in accordance with the American Diabetes Association thresholds for hypoglycemia, hyperglycemia, and euglycemia (normal). This LED system would ask the user to input the blood glucose value, and the resulting output would be a lighting of the LED bulbs. This code was not that difficult to implement compared to the other two components of the MVP; however, we added an additional portion that would also alert the user on the Serial Monitor what their blood glucose status would be, and it would make sure the alert lasted for 10 seconds each time the user inputted their blood glucose level, which made the code a little bit more technically challenging. We rated the project quite technically difficult due to the fact that none of us were well versed in the Python language or Arduino/C++, so it was a learning curve for each contributing team member that had to take time away from the project to learn the basics of code. Once we learned the basics of the code, we were able to further challenge ourselves by implementing other functions and setups in our code. It should be noted that a lot of times when we encountered challenges, the large language AI model ChatGPT was a good resource to help us fill in our knowledge gaps. Thus, the overall rating for technical difficulty, given the large amount of time we had to spend learning the code and given the relative complexity of the hardware and software, was a 7/10.

Legacy and Lessons Learned

Risk Management

During the R&D phase, risk management drove how our group operated. ISO 13485 specifically outlines multiple risk-based clauses. The systematic methods to prevent dangers were important as we did not want members to be injured. First, all associated dangers were verbally outlined (soldering iron, 3D printing machine, tools, etc.), and each member was aware of them. These risks also included hypothetical scenarios to ensure that nothing came unplanned. Once these situations were documented, each case was analyzed fully. We looked at the probability of an event occurring, what to do in the case of injuries, and what future users would face as potential injuries. One example of a future problem could be that users may shock themselves. While the case does house everything safely, there is a possibility that the exposed connections from the PPG sensor could give a small shock to the user. This may elicit a minuscule injury, but it nevertheless should be averted. Electrical tape was put in to insulate and mitigate this possibility. Overall, a systematic approach to identifying risks and producing a safe device for users was an important part of the engineering process that future students should take into consideration.

Engineering Standards

As mentioned in the "Engineering Design Report" section, several engineering standards. To summarize, ISO 13485, IEC 60601-1, and IEC 62366 were used. ISO 13845 was primarily used as a design and development control. In particular, lots of documentation was conducted to ensure the traceability of the project's direction. Testing each component was also essential to ensure that the final product worked exactly as intended with no component failure. IEC 60601-1 was in place to ensure that team members were safe during the R&D process. The use of 3D printers, soldering irons, and live electrical components means that dangers are always present. Necessary personal protective equipment was always in use to ensure the safety of group 8. Lastly, IEC 62366 was in place, as accessibility and usability are two of the core principles of the MVP. The device had to be simple to use, as working-age adults already lack time as it is. Having simple software will allow these users to easily pick up the product and feel empowered to use it.

Standards, Medical Guidelines, and Institutional Knowledge

In our PDR, we outlined our initial testing and verification strategy to include volunteers from the BME 405 class to take their blood glucose measurement to prove our MVP was able to effectively predict blood glucose levels non-invasively. This blood glucose measurement would involve using a blood glucose finger-prick testing kit in order to verify the predicted blood glucose value from the MVP-matched actual experimental testing results. However, we learned that according to the IRB (Institutional Review Board), testing on individuals for verification purposes was not allowed. IRB approval is necessary to conduct studies involving human participants, such as having individuals volunteer to be finger-pricked using a blood glucose monitoring device for verification purposes while testing our MVP. However, due to the time and resource constraints of this class, obtaining IRB approval and recruiting volunteers for this purpose was not feasible. As a result, we refrained from using volunteers from the BME 405 class to verify our MVP was successful to comply with medical guidelines.

Constraints Operated Under

There were a multitude of constraints that posed limitations to our project. The time constraint of only 15 weeks to complete the project was a major limitation. The first few weeks were spent narrowing down our project idea between three orthogonal signals through a calcium level correlation, a PPG sensor, and a tremor sensor. After narrowing the decision down to only using the PPG sensor, we further refined our scope of our project to focus on a device incorporating Machine Learning for hyperglycemic and hypoglycemic notification. Effectively, we were able to build our device during the time period of 8-9 weeks, which was fortunately successful. We also had the budget constraint of $700, which prevented us from getting advanced sensors, but this was not as applicable as the time constraint on the project. A last limitation was our lack of extensive experience in developing our own machine learning algorithm and CAD files. Thus, learning how to troubleshoot and integrate the different aspects of the project together lent a new experience for us, taking more time and effort.

Details Regarding Future Clinical Trials Utilizing our MVP:

For validation purposes of our MVP, a clinical trial can be conducted using a finger-prick glucose monitoring device to prove our MVP provides accurate blood glucose readings. Before participating and beginning the clinical trial, the clinical trial will need to undergo IRB approval to ensure ethical compliance and safety given that this is a human clinical trial. Then, if the device is marketed as a Class II medical device, it will need to undergo FDA approval before going to market, if the MVP is proved to be successful. The study design can include at least 30 individuals, half having no diabetes and half having type 2 diabetes mellitus and are of the working age population (25-54 years old). The participants will first use the MVP to determine which category (hypoglycemia, hyperglycemia, and euglycemia) that they belong in clinically, and then use the invasive finger-pricking blood glucose monitor to validate that their blood glucose reading matches the category they were assigned to based on the MVP. In terms of volunteer recruitment, the volunteers will receive a document detailing the details of the trial and make sure that each volunteer provides an informed consent document prior to testing on those individuals. The key points to include in this document is the exact study design and method, the fact that there are minimal risks associated with this test due to the finger-pricking part of the study, and that they are able to withdraw from the study for any reason at any time among other details. The trial data would be stored in an encrypted, password-protected file on a cloud server only accessible by the study researchers and administrators. The data will also maintain patient privacy and thus not include incriminating and exposable details affecting the patient's security and safety. The study will also ensure the data complies with HIPAA standards of patient privacy. If participants have any adverse events associated with the clinical trial, they have the option to drop out of the study with no penalty. Adverse events should be reported to the IRB immediately and be included in the final design report. Additionally, all adverse events should be readily addressed and participants should receive medical care as necessary. If the prototype malfunctions, the data collection process of the clinical trial will stop and the issue will be documented.

Materials, Sensors, and Hardware Used in the Project

The main literature backing GlucoPress was from Prabha et al., where she used a particular PPG sensor that was listed at around $200. To follow the paper closely, our team initially decided to buy this sensor. But there was a conflict within our budget, as two of these (one for backup) would consist of over half of the available money. Therefore, with Dr. Mai's advice, we settled on the MAXREFDES117# sensor as it is the most common PPG signal with fast shipping. Due to the original vision of a portable design, we planned to use Photon 2 I2C with a cheap LED strip. Once all the parts arrived, there was a conflict within hardware. It turned out that the PPG sensor is only compatible with specific boards listed on their datasheet on Page 4, link found in Appendix G. Based on this newfound knowledge, the Arduino UNO R3 was selected, as there were many resources online that could aid in the design process. Once the PPG sensor was properly calibrated, the LED strip was another problem that came up, as there were no resources outlining what the LED could do. As a result, it was just not possible to initialize the LED in the LED classification code. Thus, we opted to use 5mm LED bulbs as they are also the most common with a lot of resources for help. These LEDs were also paired with resistors. Initially, a 1 kΩ resistor was used, but it did not illuminate within a 4-foot distance for the user to see. This prompted a switch to the 220 Ω resistor, which solved this problem. Overall, if future students should choose to replicate this device, an important step to keep in mind is to ensure that all the devices are compatible with each other before purchasing.

Knowledge of Contemporary Resources and Past Classes

Members of Group 8 found the following classes helpful:

- BME 202 - Control and Communication in the Nervous System
- EE 202L - Linear Circuits
- BME 413 - Bioengineering Signals and Systems
- QBIO 401 - Introduction to Computational Analysis of Biological Data
- BME 415 - Regulation of Medical Products
- WRIT 340 - Advanced Writing and Communication for Engineers

- BISC 220 - General Biology: Cell Biology and Physiology

BME 202 introduced basic topics of circuits and EE 202L expanded upon these concepts. Specifically, BME 202 started with basic RC circuits for students to have a general understanding of how electrical components work. EE 202L expanded upon all electrical components including op amps and transistors. Additionally, topics such as signal transformation into the s domain and laplace transforms were introduced. BME 413 then takes these topics and goes into plenty of depth. The entirety of 413's curriculum seemed incredibly relevant for BME 405. To name a few topics: fourier transforms (forward and inverse), correlation, autocorrelation, basic signal information (distinct and analog), convolutions, etc. could all be applicable to a project. QBIO 401 provided the power of Python to GlucoPress. Starting from basic scripts to then learning about advanced Machine Learning models really benefited the ML portion of the device. Specifically, the SciKit Learn libraries on Python played a pivotal role in the algorithm. BME 415 introduced plenty of engineering standards and the process of creating an MVP. In a way, BME 415 should be a prerequisite for BME 405 as it contains a lot of necessary background knowledge that allows students to understand the roadmap. WRIT 340's writing assignments and final project were reflective of the documentation process in ISO 13485. Finally, BISC 220 gave group 8 basic physiological knowledge of how the heart operates for HR values and how oxygenation occurs for SpO_2 % values.

Societal, Economic, Ethical, and Regulatory Considerations

This project incorporates societal, economic, ethical, and regulatory considerations to ensure it addresses the needs of users responsibly. Societally, the device attempts to improve the quality of life for working-age individuals with diabetes by offering an efficient, non-invasive glucose monitoring solution that can deliver results comparatively quickly. Economically, it has the potential to reduce healthcare costs by preventing complications, as the user can address their blood glucose level in real-time and take corrective actions. Additionally, since the device was reproduced at a comparatively low cost, it can compete with existing blood glucose monitors if manufactured on a wide scale—where parts become cheaper if a large amount is bought. Ethically, the project prioritizes data privacy, safety, and fairness by ensuring the ML algorithm is unbiased and inclusive across diverse populations. Additionally, if the MVP was to be tested using human clinical trials, IRB and FDA approval would be necessary. Regulatory compliance, including IRB approval for trials, adherence to ISO and FDA standards, and transparent labeling, ensures the device meets safety and performance requirements. These considerations collectively support the design and development of an impactful and useful medical device.

Laboratory Notebook

Linked here is the weekly-updated Lab Notebook: Link. It is also listed in the appendix as Appendix F.

We decided to keep a weekly update to ensure that individual team members' contributions are listed in explicit detail for organizational and troubleshooting purposes should any error occur in the MVP.

External Member Contributions

We would like to specially thank external members for their contributions: Dr. John Mai for advisement on project directions and team deadlines, Dr. James Yoo for his advisement on CAD case files and assistance with case design/assembly, Trent Benedick for his advisement on BOM and PPG code troubleshooting, Ray Peck for soldering and case assembly assistance, Alessandro Tasso for his crash course in Solidworks, and Myilan Muruganujan for his soldering assistance.

Team Contributions

Team Member	Contribution
Mya Cohen (Hardware/Software Assistance)	Initial LED Code, PPG Sensor Soldering Assistance, ML Initial Literature Review ● CDR Part 1: Demo-Video Slide Deck Assistance ● CDR Part 2: Presentation Script Assistance ● CDR Part 3: Section 1,2,10,11
Boon Le (Hardware)	Circuit Assembly, Circuit Design, Hardware Research/Prototyping/Testing, CAD Case Design/Printing, Sourcing/Optimizing PPG Sensor Code, Soldering ● CDR Part 1: Demo-Video Editing and Scripting ● CDR Part 2: Presentation Slide Deck Creation ● CDR Part 3: Section 4,7,8,9,10,11, overall final proof-reading and editing
Karishma Rohatgi (Software)	ML Model Training, ML Model Experimentation and Evaluation/Testing, Regression Feature Extraction and Selection, PPG Dataset Sourcing and Cleaning, Project

	Scope Literature Review, LED Code Troubleshooting ● CDR Part 1: Demo-Video Slide Deck Creation ● CDR Part 2: Presentation Slide Deck Creation ● CDR Part 3: Section 4,7,8,9,10,11, overall final proof-reading and editing

References

Aromring. "MAX30102_by_RF/README.Md at Master · Aromring/MAX30102_BY_RF." *GitHub*, github.com/aromring/MAX30102_by_RF/blob/master/README.md. Accessed 7 Dec. 2024.

Bahameish, Mariam, et al. "Strategies for Reliable Stress Recognition: A Machine Learning Approach Using Heart Rate Variability Features." *MDPI*, Multidisciplinary Digital Publishing Institute, 18 May 2024, www.mdpi.com/1424-8220/24/10/3210#:~:text=Stress%20recognition%2C%20particularly%20using%20machine,regarding%20the%20ML%20model%20performance.

"Check Your Blood Glucose: Diabetes Testing & Monitoring." *Diabetes Tests | ADA*, diabetes.org/living-with-diabetes/treatment-care/checking-your-blood-sugar. Accessed 2 Dec. 2024.

Deborah Leader, RN. "Oxygen Saturation and How It Is Tested." *Verywell Health*, Verywell Health, 3 June 2024, www.verywellhealth.com/oxygen-saturation-914796#:~:text=An%20oxygen%20saturation%20level%20below%2095%25%20is%20not%20normal.,under%2090%25%20requires%20emergency%20care.

Defining hazardous waste: listed, characteristic and mixed radiological wastes | US EPA. (2024, October 16). US EPA. https://www.epa.gov/hw/defining-hazardous-waste-listed-characteristic-and-mixed-radiological-wastes

Description of U.S. Adults Who Are Not Digitally Literate, nces.ed.gov/pubs2018/2018161.pdf. Accessed 3 Dec. 2024.

"Diabetes and Heart Health: Can Tracking Help?" *WebMD*, WebMD, www.webmd.com/diabetes/diabetes-heart-trackers. Accessed 2 Dec. 2024.

Free CAD designs, files & 3D models | The GrabCAD Community Library. (n.d.). https://grabcad.com/library/arduino-uno-case-3

Fu, Xiaomin, et al. "Implementation of Five Machine Learning Methods to Predict the 52-Week Blood Glucose Level in Patients with Type 2 Diabetes." *Frontiers in Endocrinology*, U.S. National Library of Medicine, 20 Jan. 2023, pmc.ncbi.nlm.nih.gov/articles/PMC9895792/.

"Getting Started." *Scikit*, scikit-learn.org/stable/getting_started.html. Accessed 7 Dec. 2024.

Instructables. (2019, June 9). *Pulse oximeter with much improved precision*. Instructables. https://www.instructables.com/Pulse-Oximeter-With-Much-Improved-Precision/

Kermani, Ali. "The Dataset of Photoplethysmography Signals Collected from a Pulse Sensor to Measure Blood Glucose Level." *Mendeley Data*, Mendeley Data, 26 June 2023, data.mendeley.com/datasets/37pm7jk7jn/1.

New international requirements for electrical and electronic waste | US EPA. (2024, November 1). US EPA. https://www.epa.gov/hwgenerators/new-international-requirements-electrical-and-electronic-waste#:~:text=*Waste%20electrical%20and%20electronic%20equipment,Cathode%20ray%20tubes.

OpenAI. (2024). *ChatGPT* (4.0) [Large Language Model]. https://chat.openai.com/chat

V;, Prabha A;Yadav J;Rani A;Singh. "Design of Intelligent Diabetes Mellitus Detection System Using Hybrid Feature Selection Based XGBoost Classifier." *Computers in Biology and Medicine*, U.S. National Library of Medicine, pubmed.ncbi.nlm.nih.gov/34329866/. Accessed 2 Dec. 2024.

Valensi, P., et al. "Influence of blood glucose on heart rate and cardiac autonomic function. the desir study." *Diabetic Medicine*, vol. 28, no. 4, 10 Mar. 2011, pp. 440–449, https://doi.org/10.1111/j.1464-5491.2010*A*.03222.x.

Appendices

Appendix A.

$$\beta = \frac{\sum_{t=-49.5}^{49.5} t \cdot y'_t}{\sum_{t=-49.5}^{49.5} t^2} = \frac{\sum_{t=-49.5}^{49.5} t \cdot y'_t}{83325}$$

$$y_t = y'_t - \beta \cdot t$$

Figure 21. Baseline leveling equation used to preprocess the data.

Appendix B.

$$r_m = \frac{1}{n-m} \sum_{t=1}^{n-m} y_t \cdot y_{t+m}$$

Figure 22. Autocorrelation Equation

The oxygen saturation percentage is found by the equation: (-45.06*Z + 30.354)*Z + 94.845. Z represents the ratio between the red and infrared wavelengths. Specifically, it is calculated using the below ratio:

$$Z = \frac{{AC_R}/{DC_R}}{{AC_{IR}}/{DC_{IR}}}$$

Figure 23. Z ratio between the red and IR wavelengths

Alternatively, the ratio can also be thought of as the root mean squared value (RMS) and is pictured below:

$$Z = \frac{{RMS(y)_R}/{\langle Y_R \rangle}}{{RMS(y)_{IR}}/{\langle Y_{IR} \rangle}}$$

Figure 24. Alternative equation and representation of fig. 23.

The equation utilizes the existing calibration from Analog Devices Incorporated.

Appendix C.

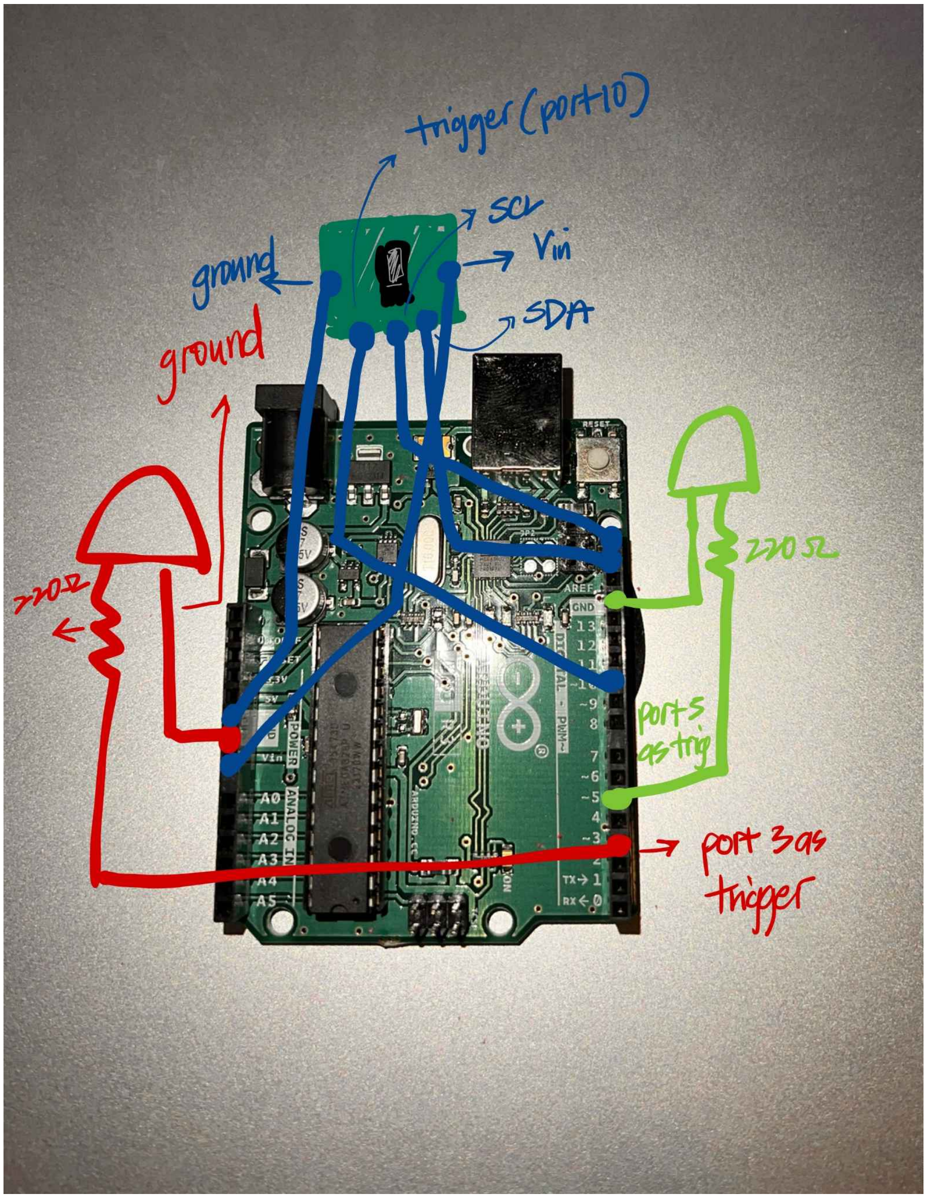


Figure 25. Initial design of the circuit with the components

Appendix D.

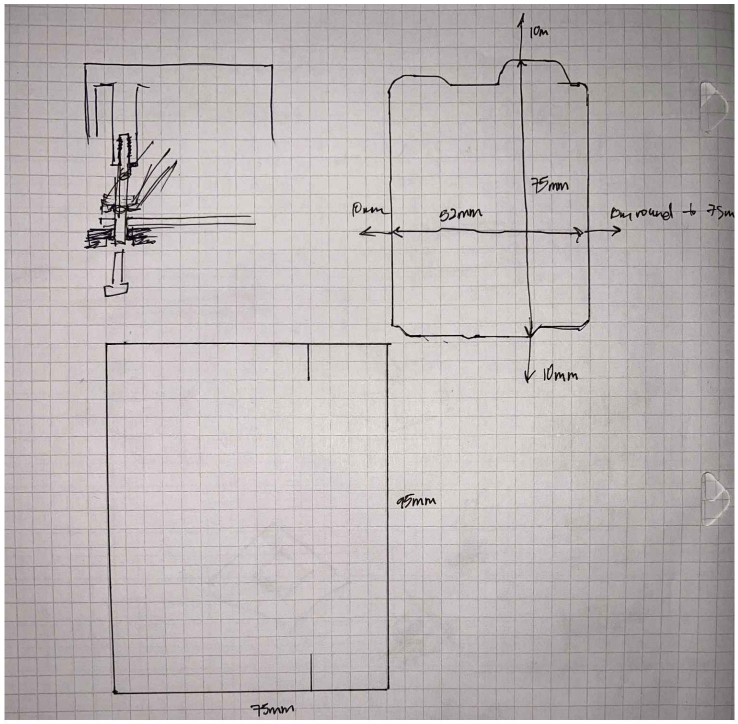


Figure 26. The initial outline of the Arduino UNO R3 board and the initial proposed dimensions. A first iteration of mounting can be observed on the top left.

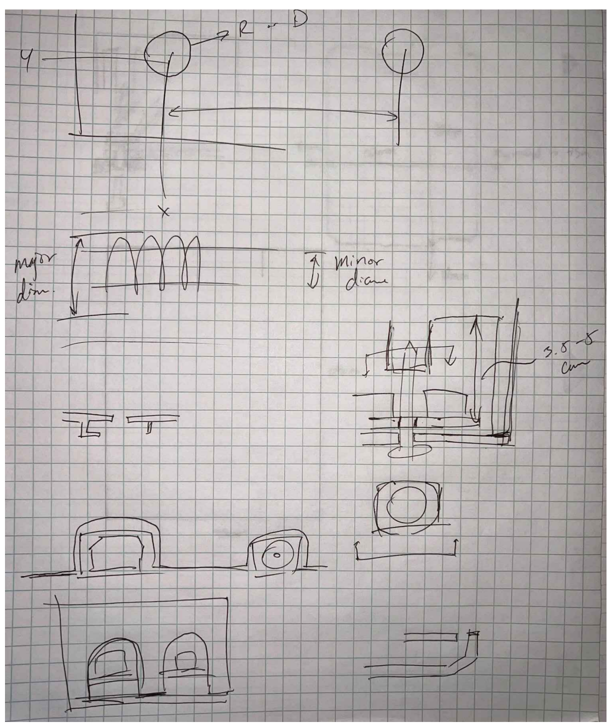


Figure 27. Further brainstorming on how to secure the top and bottom case together as well as basic CAD principles.

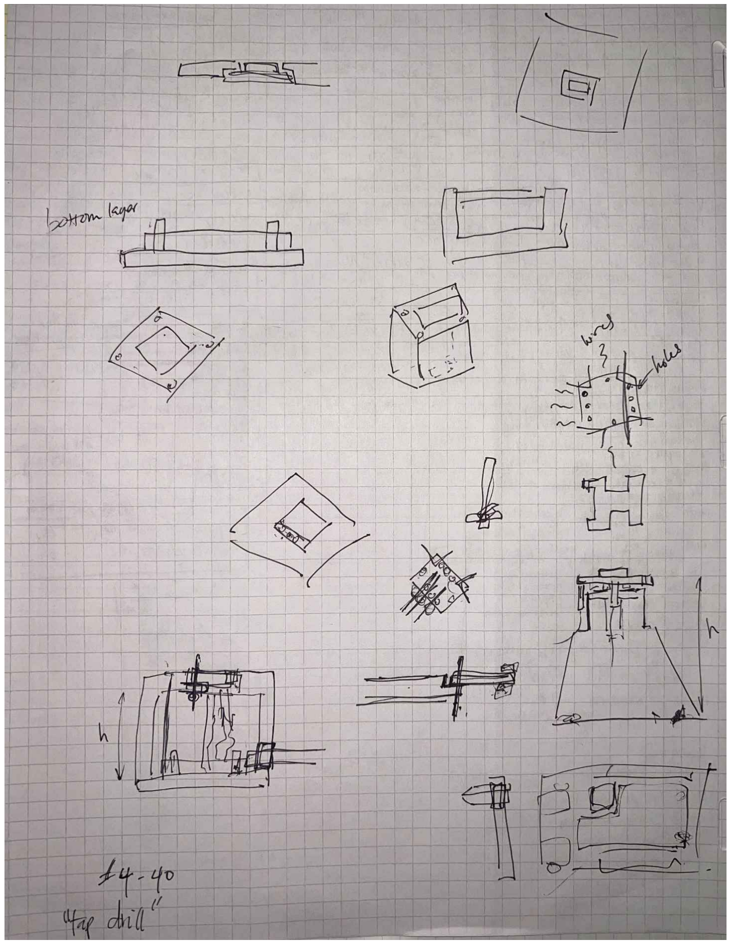


Figure 28. More proposed mounting systems. In particular, how the I2C would sit on the bottom case and how the PPG sensor would be secured.

Appendix E.

Evaluation of Other ML Model Performances and Comparison Between Methods:
We have already discussed that the best performing ML model was the linear regression with polynomial feature extraction, giving us an R^2 value of 0.86. In this appendix, we discuss the reason for scrapping the other ML Models. Our basis for excluding the other ML models such as DecisionTree and RandomForest were that the performance metrics were much lower than that of the linear regression with polynomial feature extraction model. The table below provides the performance metrics for each ML model tested with the obtained dataset.

<u>Table 8. Evaluation of ML Models in ML Algorithm Development Process</u>

ML Model →	Linear Regression (w/o Feature	DecisionTree Classifier	RandomForest Regression	Linear Regression (w/

	Extraction)			Feature Extraction)
R^2 Value	0.108	0.33	0.54	0.864
Mean Squared Error	0.057	0.04	0.03	0.069
Root Mean Squared Error	0.239	0.21	0.17	0.093
Mean Absolute Error	0.197	0.15	0.11	0.057

As you can see, the R^2 value of the Linear Regression with Polynomial Feature Extraction had a big jump compared to the other ML models evaluated. A key realization in this project was that sometimes the simpler ML models are actually the more accurate ones. The RandomForest and DecisionTree models are both quite complex in nature, and harder to fit the data to due to their high specificity which means more limitations and restrictions placed on the dataset. In the future, if this project was to be replicated, consider moving forward with the more simpler ML model which is linear regression. Moreover, attempt to improve the baseline R^2 value of 0.864.

Appendix F.

Laboratory Notebook: Link
Google Drive of All Associated Files: Link
Final Presentation: Link

Appendix G.

Arduino UNO R3 data sheet: Link
  Page number used in report: 8
MAXREFDES117# data sheet: Link
  Page number used in report: 3, 4
Green LED data sheet: Link
  Page number used in report: 2
Red LED data sheet: Link
  Page number used in report: 2