GNG5140: Engineering Design

Deliverable D: Initial Prototype Analysis & Test Results

Submitted by

Team Opioid Overdose

Ayham AlAkhras 300207406

Min Ju Kim 7296534

Varsha Srinivasan 300157999

March 7, 2021

University of Ottawa

#### 

Abstract

This report outlines the overall design concept of the opioid overdose monitor and the prototypes required to reach a fully functioning final comprehensive prototype. First, the device is broken down to four sub-systems, each encompassing an essential aspect of the opioid overdose monitor system. They are the measurement system, the signal processing algorithm, the ergonomic casing, and the user interface. These sub-systems are then evaluated, and big-picture design decisions are made. These decisions include the software and hardware used, the ergonomic casing for the device, and the intended design and prototyping strategy. The team decided to use Arduino components and software for the measurement system, a machine learning artificial intelligence for the signal processing, a facemask to house all components and measurement devices, and a Kotlin powered Android application for the user interface. Once the design is outlined, the key prototypes for this timeframe are listed: the circuit simulation prototype, the user interface prototype, and the ergonomic prototype. The three prototypes were developed by the team since the last deliverable and produced promising results. While there is still much to be done to complete these prototypes and integrate them into the final comprehensive prototype, the team has made excellent progress and is on track to complete this project on time.

Table of Contents

[Abstract i](#_Toc66049455)

[Table of Contents ii](#_Toc66049456)

[List of Figures iv](#_Toc66049457)

[List of Tables v](#_Toc66049458)

[List of Acronyms vi](#_Toc66049459)

[Introduction 7](#_Toc66049460)

[Progress Update 7](#_Toc66049461)

[1. Solution Concept 8](#_Toc66049462)

[1.1 Circuit 9](#_Toc66049463)

[1.2 Ergonomics 9](#_Toc66049464)

[1.3 Software 11](#_Toc66049465)

[1.3.1 User Interface 11](#_Toc66049466)

[1.3.2 Machine Learning 13](#_Toc66049467)

[2. Initial Prototype and Test Results 14](#_Toc66049468)

[2.1 Circuit Diagram and Simulation 14](#_Toc66049469)

[2.1.1 Circuit Diagram 15](#_Toc66049470)

[2.1.2 Components Used 16](#_Toc66049471)

[2.1.3 Circuit Explanation 17](#_Toc66049472)

[2.1.4 Working Model 18](#_Toc66049473)

[2.1.5 Prototype Analysis 18](#_Toc66049474)

[2.1.6 Component Substitutions for Prototype 22](#_Toc66049475)

[2.2 Ergonomic Prototype 23](#_Toc66049476)

[2.2.1 Ergonomic Test Results 24](#_Toc66049477)

[2.3 User Interface Drawings and Software Prototype 25](#_Toc66049478)

[2.3.1 UI Drawings 25](#_Toc66049479)

[2.3.2 UI Software Prototype 27](#_Toc66049480)

[2.3.3 Evaluation 30](#_Toc66049481)

[3. Updated Project Plan 31](#_Toc66049482)

[3.1 Plan for Deliverable E 31](#_Toc66049483)

[Conclusions and Recommendations for Future Work 33](#_Toc66049484)

[Bibliography 34](#_Toc66049485)

[APPENDICES 35](#_Toc66049486)

[APPENDIX I: Bill of Materials 35](#_Toc66049487)

[APPENDIX II: Arduino Code for Circuit Prototype 36](#_Toc66049488)

[APPENDIX III: UI 39](#_Toc66049489)

List of Figures

[Figure 1. Ergonomic Design Concept 11](#_Toc66049490)

[Figure 2. Circuit Diagram of Initial Prototype 15](#_Toc66049491)

[Figure 3. Normal Condition BPM and SpO2 19](#_Toc66049492)

[Figure 4. OD due to Marginally Low SpO2 19](#_Toc66049493)

[Figure 5. OD Alert when SpO2 Significantly Below 90% 20](#_Toc66049494)

[Figure 6. OD Alert due to Low BPM 21](#_Toc66049495)

[Figure 7. Servo Rotating due to Alert 21](#_Toc66049496)

[Figure 8. Error Border Values 22](#_Toc66049497)

[Figure 9. Initial Prototype with Cardboard showing a) the insides, b) a user wearing 24](#_Toc66049498)

[Figure 10: UI Drawings 26](#_Toc66049499)

[Figure 11: UI Software Prototype Screenshots 28](#_Toc66049500)

[Figure 12: Slide Button 29](#_Toc66049501)

[Figure 13. Updated Project Plan 32](#_Toc66049502)

List of Tables

[Table 1: Components Table 16](#_Toc66049503)

[Table 2: Component Substitution Table 22](#_Toc66049504)

[Table 3. Ergonomics Survey Results 24](#_Toc66049505)

[Table 4: UI Survey Results 30](#_Toc66049506)

List of Acronyms

|  |  |
| --- | --- |
| **Acronym** | **Definition** |
| OD | Opioid Overdose |
| UI | User Interface |
| ML | Machine Learning |
| BLE | Bluetooth Low Energy |
| IO | Input/Output |
| PWM | Pulse Width Modulation |
| USB | Universal Serial Bus |
| ICSP | In-Circuit Serial Programming |
| LCD | Liquid-crystal Display |
| VCC | Common Collector Voltage |
| GND | Ground |
| BPM | Breaths per Minute |
| SpO2 | Blood Oxygen Saturation |

# Introduction

There are numerous quotations from scholars across history emphasizing on the importance of good preparation and planning such as the quote *“Well begun is half done”* by the ancient Greek philosopher Aristotle. Even in prototyping, a good initial prototype will lay a foundation for a good final product.

## Progress Update

Back in Deliverable C, a list of project requirements was revised to give a clear description of how the prototype should perform [1]. Using this list, the initial design that will be introduced in this report was brainstormed and developed according to client needs.

This deliverable also reports the specifications and initial testing results for the initial prototypes. While results are promising, there are still many issues that need to be addressed before beginning integration and advanced testing.

Finally, the deliverable updates the project plan reported in Deliverable C to show a more detailed plan from now until the end of the design process.

# 1. Solution Concept

According to the client, the proposed design must satisfy the following criteria:

* + Cost effective design (100 CAD budget)
  + Monitors blood oxygen level and respiratory rate reliably
  + Lightweight and small to be wearable on the user
  + Discrete and compact package
  + Editable emergency contacts
  + Quick and accurate response to overdose indicators
  + Long battery life

To meet these requirements, the team proposed three basic subsystems for the device: an electrical measurement device, a user interface, the ergonomic unit, and a signal processing unit. Each subsystem plays a vital role in ensuring that the device operates effectively. The electrical measurement device was chosen as an Arduino processor with integrated sensors to cut down on costs and complexity. The Arduino device would then pair to a mobile application via Bluetooth, utilizing the built-in UI functionality and processing power of a smartphone to further cut down on costs. The mobile application developed for Android using Kotlin contains the UI and communicates with both the Arduino and AI algorithm. When enabled, the application detects an overdose then alerts the appropriate contacts according to the user’s choosing. The ergonomic unit was chosen to be a facemask due to its effectiveness and convenience as outlined in section 1.2. Though the facemask is slowly being phased out as vaccines for COVID 19 are being rolled out, we predict that they will still be worn often by some people in some situations, and that the users of our device care enough about their health to wear the mask. Lastly, the facemask offers an elegant method of breath rate measurement. Because respiratory rate is a much quicker indicator of an opioid overdose, effective measurement of it could alert responders minutes earlier than just a pulse oximeter; and these minutes can be crucial to saving the patient's life.

## 1.1 Circuit

The heart of any hardware module is its circuit. The efficiency of the circuit will determine the efficacy of the module and its parameters such as response time, data measurement, troubleshooting, etc. However, it must be connected to a mobile application for data processing and user interface. In this module, Tinkercad is utilized to generate the desired results with the basic components that the website can offer. The circuit is aimed to integrate a sensor to measure the Blood Oxygen Saturation and Breaths per minute with an Arduino nano 33 BLE sense board. For more details on the components and substitutions refer to section 2.1 circuit Prototype.

## 1.2 Ergonomics

The most important non-performance requirements listed for this project in Deliverable C is its discreetness, comfortability, and freedom of the user’s hands [1]. According to the client, there are stigma and discrimination around opioid users that prevent them from accessing services that reduce the risk of an OD such as a safe injection site [2]. Moreover, if the device were uncomfortable to wear for extended periods of time, it could sway the user from using the device when using.

To hold the blood oxygen and breath rate measurement sensors and associated circuitry, multiple discrete enclosures were brainstormed by the team. Devices were brainstormed according to the measurement requirements. The blood oximeter chosen must be placed on a thin piece of skin to effectively evaluate oxygen saturation in the blood; usually, blood oximeters are placed on a patient’s finger, toe, or earlobe. Breathing rate can be measured in one of three ways: via a sonar system, with a respiratory stretch sensor placed on the patient's chest, or using a humidity/pressure sensor near the patient's mouth/nose. The sonar system poses many challenges and accuracy flaws in a variety of test cases; thus, it was eliminated from consideration. Furthermore, the stretch sensor was deemed too complex, inconvenient, and expensive to include in the device and was thus also eliminated. All proposed devices could discreetly measure blood oxygen level but not respiratory rate without the use of a separate measurement device. The proposed designs included a pair of gloves, a shoe (seen in a previous project), a watch, and a pair of over-ear headphones. While these designs all offered discretion, none offered an effective respiratory rate measurement system. Thus, it was decided that this project will utilize a face mask to conceal all components and measurement systems. The face mask covers the microcontroller, a pulse oximeter is placed on the strap of the mask to be clipped on the user’s earlobe, and a humidity sensor in the mask is used to measure the user’s breathing pattern. Figure 1 shows the ergonomic concept of this prototype.

Diagram

Description automatically generated

Figure 1. Ergonomic Design Concept

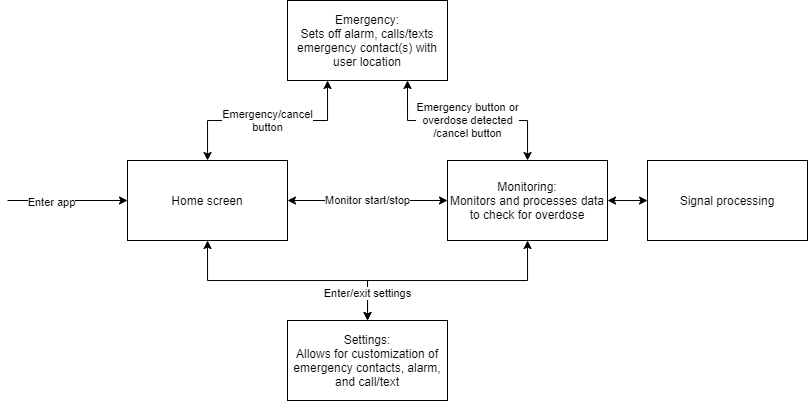
## 1.3 Software

### 1.3.1 User Interface

The user interface consists of four main screens that the user can interact with:

1. Home screen: the starting point for the user, navigates to other screens
2. Monitoring screen: displays an animation confirming that monitoring has begun, displays user’s vital signs, and monitors for overdose with the help of the signal processing unit
3. Emergency screen: triggered either manually by the user or automatically by the artificial intelligence onboard, sounds an alarm and calls/texts the user's emergency contacts
4. Settings: can be accessed through home or monitoring but not emergency screen, allows the user to customize their emergency contacts, select if the application calls or texts their emergency contact, and their alarm preferences such as sound and vibration

These screens are switchable according to the following flowchart:



The user interface is also planned to have a confirmation feature for emergency triggering and stopping to prevent accidental triggering or stopping of the system.

For all prototypes, the UI is developed in Android Studio using Kotlin for its functionality, effectiveness, and widespread adoption. The mobile application will run on any modern Android device allowing for good scalability later down the line. Android Studio also hosts a wide variety of features that can be utilized and implemented to increase the functionality of the device.

### 1.3.2 Machine Learning

One distinguishing feature of this prototype compared to previous iterations of an OD monitor is the implementation of machine learning (ML). The purpose of ML algorithm is to process the bio-signals coming from the sensors and to minimize biological variation in the data by setting up a baseline for each individual user.

# 2. Initial Prototype and Test Results

The following initial prototypes were deemed necessary to complete within the given timeframe of this deliverable:

* Circuit diagram and simulation
* Ergonomic prototype
* User interface drawings
* User interface software

Physical circuit, signal processing, and comprehensive prototypes will be developed later due to dependencies.

## 2.1 Circuit Diagram and Simulation

This circuit prototype [3] was developed on Tinkercad and it represents the design of a circuit integrated with a pulse oximeter and a respiratory rate measurement sensor which is achieved by using a humidity sensor from the Arduino Nano 33 BLE Sense.

### 2.1.1 Circuit Diagram

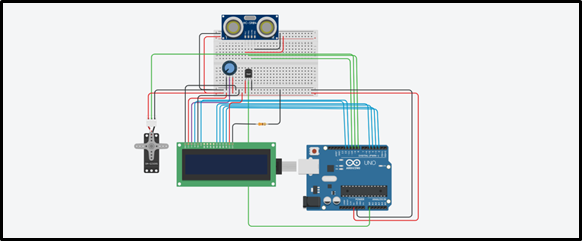


Figure 2. Circuit Diagram of Initial Prototype

### 2.1.2 Components Used

Components used in the prototype are as follows:

Table 1: Components Table

|  |  |
| --- | --- |
| **Arduino Uno**: | * microcontroller board based on the ATmega328P * 14 digital IO pins (of which 6 can be used as PWM outputs), 6 analog inputs, * a 16 MHz ceramic, a USB connection, a power jack, an ICSP header and a reset button. |
| **Ultrasonic Sensor HC-SR04:** | * sensor that can measure **distance.** It * emits an **ultrasound** at **40,000 Hz (40kHz)** which travels through the air and if there is an object or obstacle on its path It will bounce back to the module. Considering the travel time and the speed of the sound we can calculate the distance. |
| **Micro servo motor SM- S23095:** | Servo motors use feedback to determine the position of the shaft, one can control that position very precisely. As a result, servo motors are used to control the position of objects, rotate objects, move legs, arms, or hands of robots, move sensors etc. with high precision. Servo motors are small, and because they have built-in circuitry to control their movement, they can be connected directly to an Arduino. Most servo motors have the following three connections:   * Black/Brown ground wire. * Red power wire (around 5V). * Yellow or White PWM wire. |
| **Temperature sensor TMP36:** | The TMP36 is a low voltage, precision centigrade temperature sensor. It provides a voltage output that is linearly proportional to the Celsius temperature. It also does not require any external calibration to provide typical accuracies of ±1°C at +25°C and ±2°C over the −40°C to +125°C temperature range. |
| **LCD 16 x 2:** | The 16x2 LCD used in this experiment has a total of 16 pins. Eight of the pins are data lines (pins 7-14), two are for power and ground (pins 1 and 16), three are used to control the operation of LCD (pins 4-6), and one is used to adjust the LCD screen brightness (pin 3). The remaining two pins (15 and 16) power the backlight. |
| **Potentiometer:** | On an LCD the potentiometer is used to adjust the bias level of the LCD - that is the contrast. |
| **Tinkercad:** | *Tinkercad* is a free, easy-to-use simulator for 3D design, electronics, and coding. It is used to design and make new circuits and experiment with them. It is an online platform which can be used to simulate 2D or 3D models and circuits. |

### 2.1.3 Circuit Explanation

Micro servo motor has 3 pins: GND, VCC and Signal. The GND and VCC pins are used to supply power to the motor. The Signal pin controls the signal to drive the motor.

Ultrasonic sensor has 4 pins namely ground, VCC, echo, and trigger. The first two pins i.e., ground and VCC are used to supply power to the sensor. The trigger pin is used for input and it is responsible for sending ultrasonic waves for the purpose of measuring distance from object while echo pin is output pin which sets too high for the time taken by wave to return.

Temperature sensor has a VCC and a ground pin for power supply and the Vout pin is connecter to analog input A0 of the Arduino uno for the purpose of measuring the temperature of any individual.

In the circuit diagram, the ground and VCC pins of ultrasonic sensor, RC servo motor, LCD module, temperature sensor and potentiometer are connected to GND and 5V pins of the breadboard respectively which is connected to Arduino UNO to provide needed power to components. Next, trigger pin of ultrasonic sensor is connected to D8 (digital pin 8) and echo pin is connected to D10 on Arduino board. The control pin of servo motor is connected to D9 on Arduino board.

To wire LCD screen to our board, we connect the following pins: LCD RS pin to digital pin 12, LCD Enable pin to digital pin 11, LCD D4 pin to digital pin 5, LCD D5 pin to digital pin 4, LCD D6 pin to digital pin 3, LCD D7 pin to digital pin 2, LCD R/W pin to GND, LCD VSS pin to GND, LCD VCC pin to 5V, LCD LED+ to 5V through a 330-ohm resistor, LCD LED- to GND. Additionally, wire the 10k pot to +5V and GND, with its wiper (output) to LCD screens VO pin (pin3).

### 2.1.4 Working Model

In this prototype, the module is programmed to work in a way when there is an anomaly in the value of either the oxygen level or the breaths per minute, an alert is raised in the app which triggers the user interface via Bluetooth module present on the Arduino Nano 33 BLE Sense. Various images depicting the variation of output for various data combination is shown in the analysis section for better understanding of the test results. The code is given in the Appendix II.

### 2.1.5 Prototype Analysis

As it can be seen in Figure 3, constant monitoring of respiratory rate and oxygen level is being done.

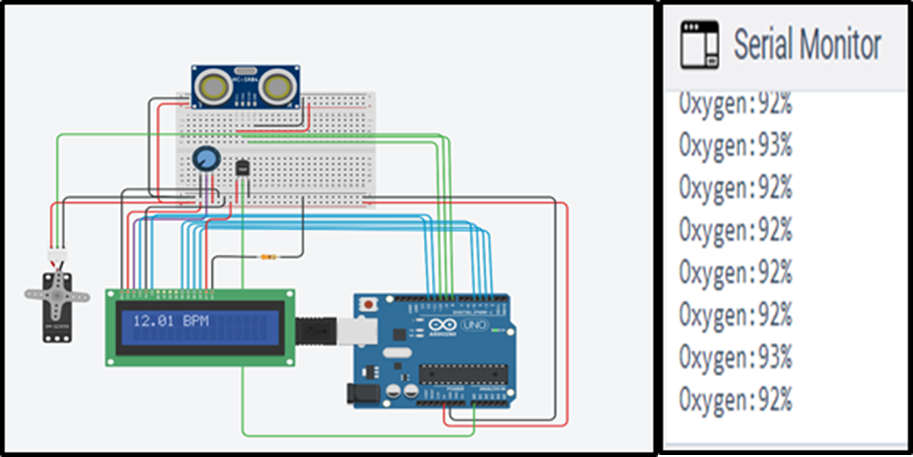


Figure 3. Normal Condition BPM and SpO2

The test case in Figure 4 is depicted for a marginal drop in the oxygen level which is stated as a danger signal as stated by the client. Here the BPM is kept at 14 but even if one criterion is fulfilled, the app is activated.

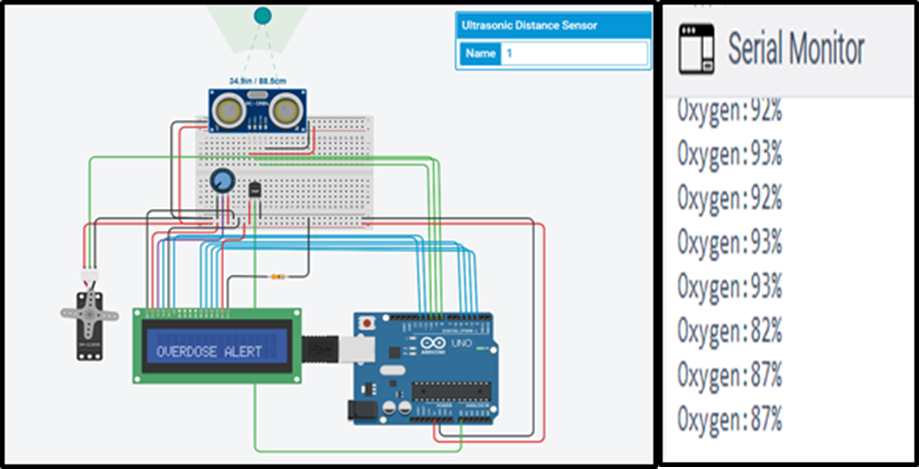


Figure 4. OD due to Marginally Low SpO2

The test case in Figure 5 is an extreme scenario, to check the performance of circuit in boundary conditions.

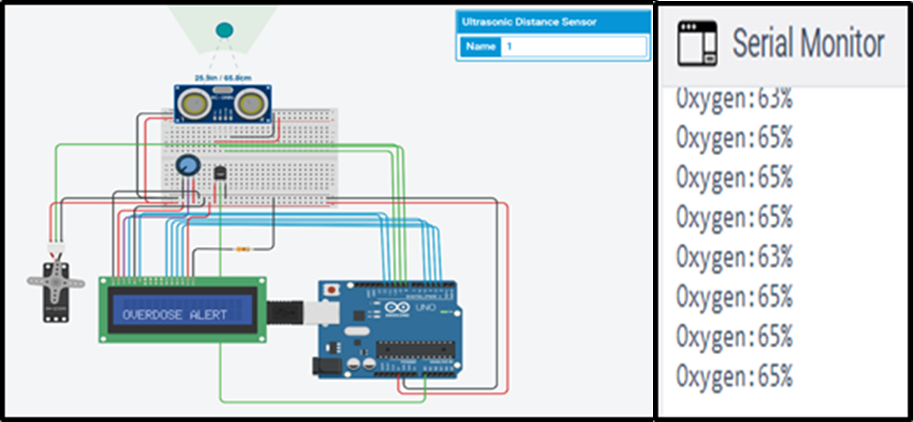


Figure 5. OD Alert when SpO2 Significantly Below 90%

The test case in Figure 6 is depicted for a drop in the respiratory rate which is stated as a danger signal as stated by the client. Here the oxygen is kept at 93 but even if one criterion is fulfilled, the app is activated.

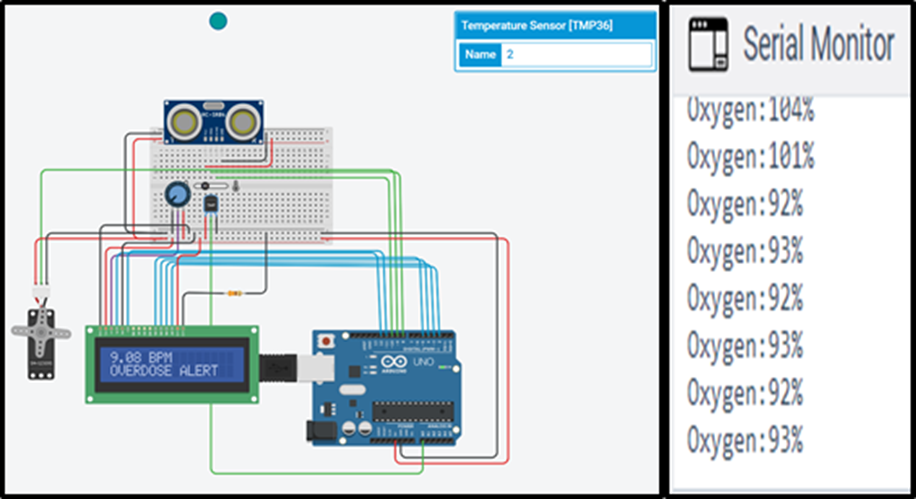


Figure 6. OD Alert due to Low BPM

It can be seen in Figure 7 below that the servo motor is rotating as an indication of the app being activated due to overdose alert.

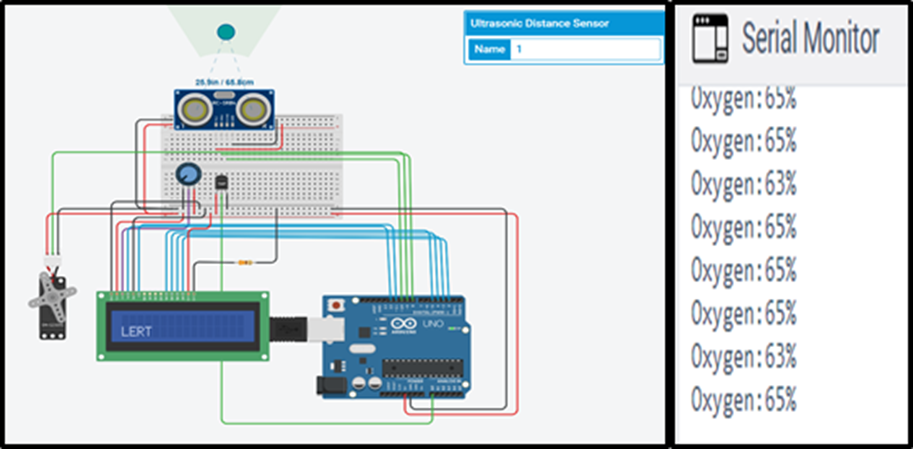


Figure 7. Servo Rotating due to Alert

If the sensor is not reading a predicted value, an error message is shown in the LCD as shown in Figure 8.

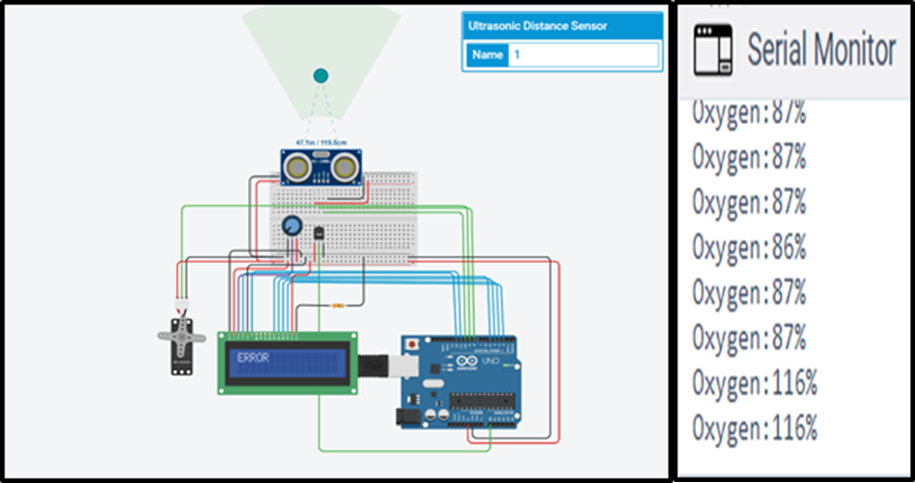


Figure 8. Error Border Values

### 2.1.6 Component Substitutions for Prototype

Table 2: Component Substitution Table

|  |  |  |
| --- | --- | --- |
| **Component Used in prototype** | **Substituted for** | **Expected performance from substitution** |
| Arduino Uno | **Arduino Nano 33 BLE Sense** | Perform as a microcontroller with inbuilt Bluetooth module to be able to communicate with the User Interface |
| Ultrasonic Sensor HC-SR04 | **Pulse oximeter** | Provide variable data to analyze the blood oxygen level and satisfy the monitoring condition |
| Micro servo motor SM- S23095 | **Circuit to app communication** | To mimic the action of circuit and app interaction so that it is visually monitored in the simulation |
| Temperature sensor TMP36 | **Humidity sensor** | Count the number of breaths per minute by sensing the variation in humidity in the mask due to respiration |
| LCD 16 x 2 | **User Interface** | Perform as a user interface and display the data collected for continuous monitoring |

## 2.2 Ergonomic Prototype

Unfortunately, none of the ordered hardware parts have arrived yet, thus. prototype testing using the actual parts could not be carried out. To test the prototype’s ergonomics without the hardware components, a model of them with the same dimensions were made in cardboard which was then glued to a placeholder mask using super glue. Figure x shows the interior of the face mask prototype and a user wearing the prototype. The blue and black pieces represent the Arduino Nano 33 BLE Sense and the MAX30102 pulse oximeter, respectively.

A person wearing a hat

Description automatically generated with medium confidence

Figure 9. Initial Prototype with Cardboard showing a) the insides, b) a user wearing

### 2.2.1 Ergonomic Test Results

Testing the ergonomics of the initial prototype consisted of a survey for three participants asking them about the ergonomic requirements of the project as mentioned in *1.2 Ergonomic Prototype*. The participants were asked four questions to rate their opinions of the prototype using a scale of 1 to 5, where 1 represents strong disagreement and 5 represent strong agreement. Table x shows the survey questions and answers.

Table 3. Ergonomics Survey Results

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Rated Answers (Scale of 1 to 5; 1 = Strongly disagree, 2 = Disagree, 3 = Neutral, 4 =Agree, 5 = Strongly agree)** | | |
| **Questions** | Participant 1 | Participant 2 | Participant 3 | |
| Is the device comfortable to wear for long period of time? | 4 | 3 | 3 | |
| Is the device small and discreet? | 4 | 4 | 4 | |
| Is the device hands-free? | 5 | 5 | 5 | |
| Does the pulse oximeter (black rectangle) provide good contact with the earlobe? | 4 | 1 | 4 | |
| Do you think you would use this device if you were a substance user? | 3 | 3 | 3 | |

Regardless of the survey results, all three participants complained of strong glue odour coming from the mask. Therefore, the microcontroller should be adhered to the mask using a different method than super glue. Another comment that they mentioned was the gluing the hardware parts hindered the device’s comfortability for some participants. Therefore, a method to adhere the hardware while being able to adjust their positions must be developed.

## 2.3 User Interface Drawings and Software Prototype

### 2.3.1 UI Drawings

The following rough drawings were developed to represent the UI of the system.

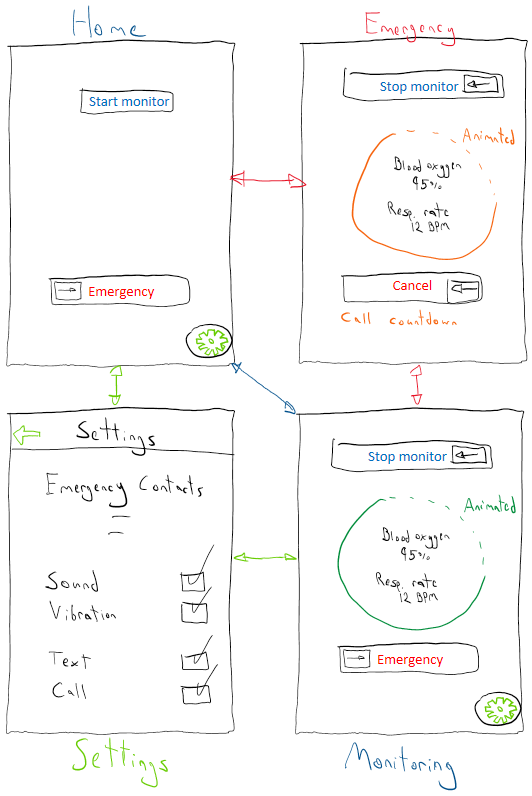
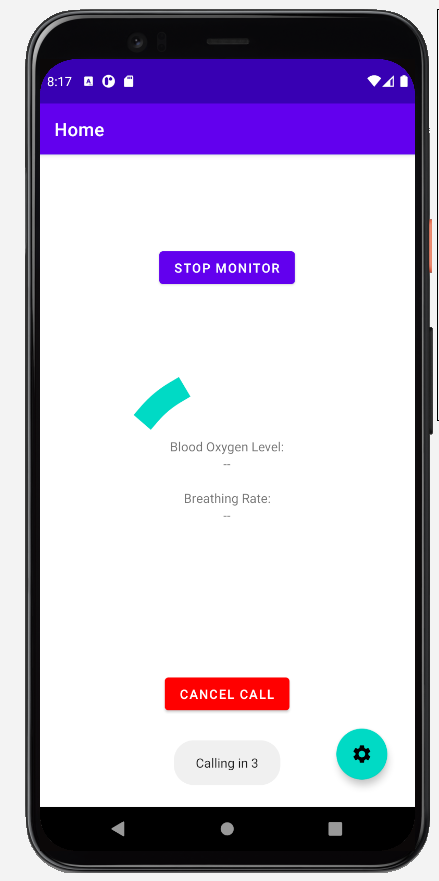
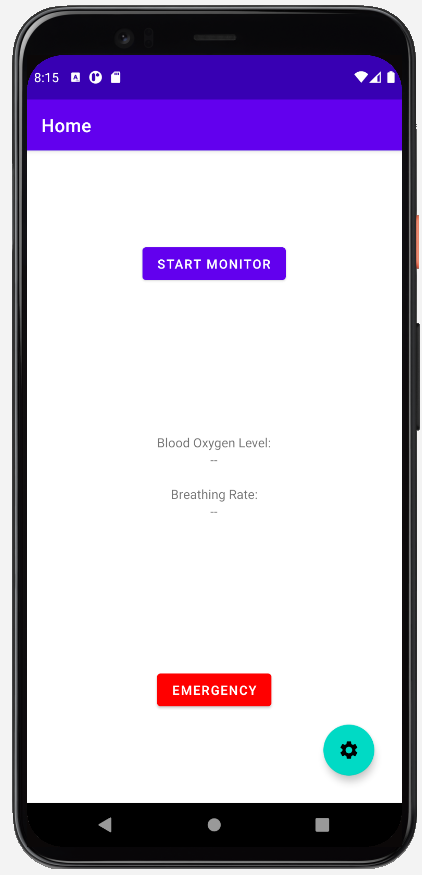


Figure 10: UI Drawings

In these drawings, the four interactable screens discussed earlier are shown and connected by arrows to indicate switching between screens. This switching is supported by the flowchart presented in section 1.3.1. Furthermore, the overall layout of the design is shown, utilizing a simplistic design that is both visually appealing and simple to use. This prototype also includes verification functionality, utilizing swipe buttons instead of simple buttons to ensure that when the user is stopping the monitor, requesting an emergency, or stopping an emergency.

### 2.3.2 UI Software Prototype

The drawings shown in section 2.3.1 were replicated as best as currently possible in an Android interface.



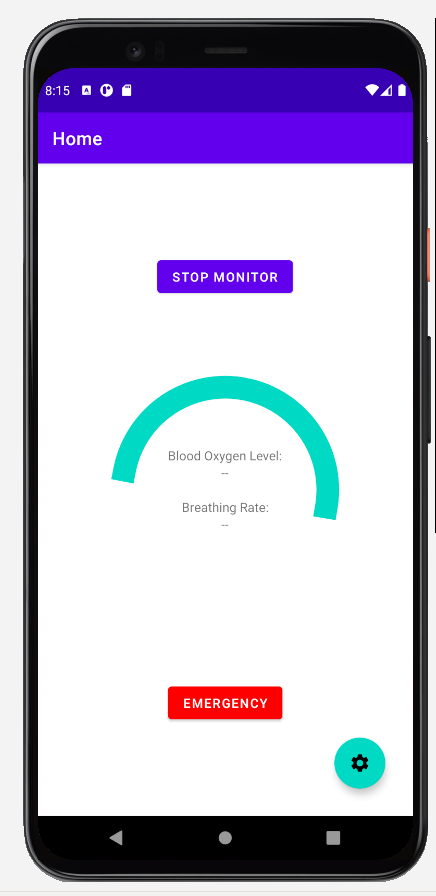
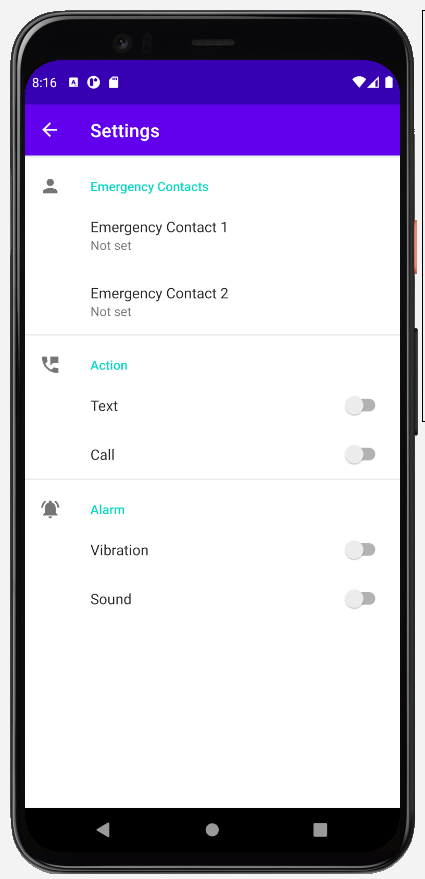


Figure 11: UI Software Prototype Screenshots

While this prototype does not contain all initially proposed functionality yet, it will be upgraded and adjusted in the coming weeks. Features such as the slide buttons, an alarm notification and pop-up, and additional emergency contact customization are all planned to be added before the comprehensive prototype. In fact, the slide button has already been found as an add-on and will be integrated into the design as soon as possible.



Figure 12: Slide Button

The application consists of a single activity and multiple fragments, this allows for smooth transitions between monitor stopping and starting. The settings screen utilizes the navigation feature in Android Studio and the default settings page provided. This screen was adjusted to fit the required settings. The main fragment then listens for clicks on the buttons displayed on screen and adjusts the animations and available buttons. While the UI has no back-end usability yet, this will be added once the physical prototype and AI prototype are past initial development. It is also worth noting that the application does not use the current industry standard of data binding or the older method of findViewById due to their complexity in fragments. Instead, Kotlin Android Extensions are used to simplify the data transfer process between the layout XML and the Kotlin code.

### 2.3.3 Evaluation

To assess the performance of the UI, three participants were asked to observe the UI and interact with it. The participants were then surveyed as in section 2.2.1, and a short focus group session was held to discuss issues with the design. The survey results were as follows:

Table 4: UI Survey Results

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Rated Answers (Scale of 1 to 5; 1 = Strongly disagree, 2 = Disagree, 3 = Neutral, 4 =Agree, 5 = Strongly agree)** | | |
| **Questions** | Participant 1 | Participant 2 | Participant 3 | |
| The UI is visually appealing | 5 | 4 | 5 | |
| The UI is easy to navigate | 5 | 5 | 5 | |
| The emergency alarm is effective in warning the user | 2 | 3 | 1 | |
| The settings offer all the customization I need | 4 | 5 | 5 | |
| I find it difficult to accidentally turn off the monitor or trigger an alarm | 1 | 2 | 1 | |

In the focus group session, the users indicated that the emergency alarm was not noticeable enough and should be a pop up rather than a small notification at the bottom. This was noted and additional visual cues will be added to the alarm. They also stated that it was too easy to trigger a false alarm or cancel a real one. This will be rectified with the addition of the aforementioned slider buttons.

# 3. Updated Project Plan

## 3.1 Plan for Deliverable E

For Deliverable E, the first step is to test the performance of the initial prototype. The initial performance testing was meant to be included in this deliverable but was omitted due to the hardware components not arriving in time. After testing the initial prototype, revisions can be made so that the prototype can be improved. The revision process will be divided into three parts: circuit, programming, and companion app.

After revising the design, the new prototype can be tested in its performance similar to the testing process of the initial prototype. Also, different test cases will be generated and simulated/tested in order to account for different situations that may arise while the device is in use. Figure 13 shows the detailed project plan from this deliverable, Deliverable D, to Deliverable H with milestones, dependencies, and task assignees. Tasks without assignees are regarded as team tasks, where all team members are expected to participate.

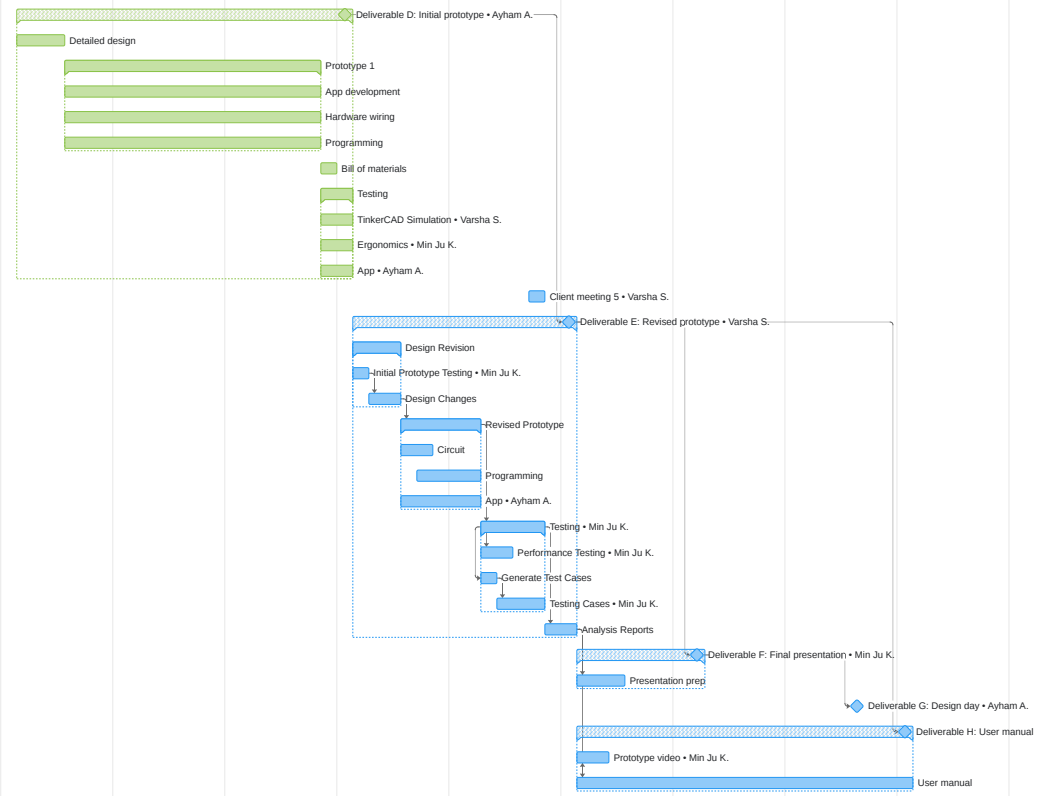


Figure 13. Updated Project Plan

# Conclusions and Recommendations for Future Work

Using the list of revised project requirements, the initial prototype was designed, built, and tested as shown in this deliverable. Even without the hardware components arriving on time, the initial testing provided valuable data of how well the initial prototype performed with respect to the project requirements.

An updated project plan was also generated which includes the initial prototype testing, task dependencies, and task assignees. This updated project plan provides a very clear roadmap of the step forward in this project and each team member’s responsibility.

Future works would involve testing the initial prototype to determine the areas of improvements, design, and implement those improvements. Also, various test cases will help improve the performance of the device. Finally, more user testing and feedback for the revised prototype will be received from the client after it is presented. The team has updated the client on the current state of the project and will perform user tests with the client once the prototypes are more refined.

# Bibliography

[1] AlAkhras, A., Kim, M. J., Srinivasan, V., (2021). *GNG5140: Engineering Design Deliverable C: Design Requirements and Project Plan*.

[2] Cahill, T. (2020). Canada’s Overdose Crisis [PowerPoint slides]. Ottawa, ON.

[3] AlAkhras, A., Kim, M. J., Srinivasan, V., (2021). *GNG5140: Engineering Design Circuit Prototype. https://www.tinkercad.com/things/iA4dJv22vJY*

APPENDICES

APPENDIX I: Bill of Materials

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| BOM Level | Part Name | Quantity | Unit of Measurement | Cost (CA$) | URL |
| 1 | Arduino Nano 33 BLE Sense | 1 | Each | 57.05 | <https://www.amazon.ca/Arduino-Nano-Sense-headers-Mounted/dp/B07WXKDVTL> |
| 2 | MAX30102 | 1 | Each | 17.42 | <https://www.amazon.ca/MakerFocus-Heart-Rate-MAX30102-Compatible-Arduino/dp/B07GGSM8SP> |
| 3 | Mini Breadboard | 1 | Each | 0 | Provided from Makerspace |
| 4 | Face Mask | 1 | Each | 0 | Already in Possession |
| 5 | Jumper Wire | 10 | Each | 0 | Provided from Makerspace |
| Total |  |  |  | 74.47 |  |

APPENDIX II: Arduino Code for Circuit Prototype

#include <LiquidCrystal.h>

#include <Servo.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2); //Digital pins to which you connect the LCD

const int inPin = 0;

// A0 is where you connect the sensor

int oxygen;

long duration;

float value;

int tmp = A1;

Servo myservo; // create servo object to mimic activating the app

int pos = 0; // variable to store the servo position

const int servo = 9; //define Servo Signal Pin

const int trigPin = 8; //define Trigger Pin

const int echoPin = 10; //define Echo Pin

void setup()

{

lcd.begin(16,2);

pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output

pinMode(echoPin, INPUT); // Sets the echoPin as an Input

pinMode(tmp,INPUT);

myservo.attach(servo); // attaches the servo on pin 9 to the servo object

myservo.write(0); // Sets Servo to initially 0 degrees

Serial.begin(9600); // Starts the serial communication

}

void loop()

{ value = analogRead(inPin)\*0.004882814;

value = (value - 0.5) \* 100.0;

lcd.setCursor(0,0);

lcd.clear();

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

// Sets the trigPin on HIGH state for 10 micro seconds

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

// Reads the echoPin, returns the sound wave travel time in microseconds

duration = pulseIn(echoPin, HIGH);

// Calculating the distance

oxygen= duration\*0.034/2;

// Prints the distance(oxygen) on the Serial Monitor

//Servo

if(oxygen>90 && oxygen <105)

{ //Check oxygen is less than 105% and greater than 90%

lcd.print(value);

lcd.print(" BPM");

delay(2000);

if(value <12)

{ lcd.setCursor(0,1);

lcd.print("OVERDOSE ALERT");

delay(1000);

for (int i = 0; i < 10; i++)

{

// scroll one position left:

lcd.scrollDisplayLeft();

// wait a bit:

delay(150);

}

myservo.write(180); //Ajust how far you want the servo to go.

delay(1000);

myservo.write(00); // Reset the servo to 0 Degrees

delay(3000); //Delay for the next reading

}

Serial.print("Oxygen:");

Serial.print(oxygen);

Serial.println("%");

}

else if (oxygen<90)

{

lcd.setCursor(0,1);

lcd.print("OVERDOSE ALERT");

delay(1000);

for (int i = 0; i < 10; i++)

{

// scroll one position left:

lcd.scrollDisplayLeft();

// wait a bit:

delay(150);

}

Serial.print("Oxygen:");

Serial.print(oxygen);

Serial.println("%");

myservo.write(180); //Ajust how far you want the servo to go.

delay(1000);

myservo.write(00); // Reset the servo to 0 Degrees

delay(3000); //Delay for the next reading

}

else if (oxygen>105)

{

lcd.print("ERROR");

delay(1000);

for (int i = 0; i < 10; i++)

{

// scroll one position left:

lcd.scrollDisplayLeft();

// wait a bit:

delay(150);

}

Serial.print("Oxygen:");

Serial.print(oxygen);

Serial.println("%");

delay(1000);

}

}

APPENDIX III: UI

Kotlin project file can be found on Microsoft Teams files:

<https://uottawa.sharepoint.com/sites/GNG5140W00-Opioid_Overdose1/Shared%20Documents/Opioid_Overdose%201/UI%20Files%20Deliverable%20D.rar>