

GNG1103
Design Project User Manual

**Opioid Overdose Monitoring Device User
Manual**

Submitted by

GNG 1103-C, Team 6

Abdelrahman Nasr,	0300132777
Daniel Trela,	0300129303
Parsa Sheykholeslami,	0300129835
Rikki Romana,	0300152011
Viktor Stojanovic,	0300137255

21 April, 2020

University of Ottawa

Abstract

This technical report is a user manual for Team C6's opioid overdose monitoring device, the O2-POD. In this report, it discusses the opioid problem in Canada, how Team C6 developed the O2-POD device to solve this problem, how the prototypes for the device are made, how to use the prototypes, how to maintain the prototypes, and the recommendations for future work for the device. The purpose of this technical report is to provide an outline on how to use the O2-POD, the completed work that Team C6 has done towards the product, and the steps to take to reproduce the product.

Table of Contents

Abstract	2
Table of Contents	3
List of Acronyms	6
1. Introduction	1
1.1 Context	1
1.2 Assumptions	1
1.3 Structure	1
1.4 Product	2
2. How the Prototype is Made	4
2.1 BOM (Bill of Materials)	4
2.2 Equipment list	4
2.3 Instructions	5
2.3.1 Box	5
2.3.2 Clamp	6
2.3.3 Electrical	8
2.3.4 Software	10
3. How to Use the Prototype	14
3.1 Functions	14
3.2 Operation	14
3.3 Setup	15
4. How to Maintain the Prototype	16
4.1 Tests	16
4.2 Maintenance	16
5. Conclusions and Recommendations for Future Work	17
5.1 Analysis	17
5.2 Latest Status and Recommendations	17
6. Bibliography	19
7. APPENDIX	20

List of Figures

Figure 1: Final prototype (Does not follow all specifications)	2
Figure 2: Wiring On Breadboard	7
Figure 3: Wiring and Pin Information for SD to ESP-32S	9
Figure 4: UI of the O2POD Application	12
Figure 5: MIT App Inventor code for Bluetooth connection with ESP32	12
Figure 6: MIT App Inventor code for receiving data from ESP32	13
Figure 7: MIT App Inventor code for sending a text with location and condition	13
Figure 8: MIT App Inventor code for getting the location of the android device	13

List of Tables

Table 1: Interpreted User Needs	2
Table 2: Design Criteria for the Device	3
Table 3: List of Materials and the Cost	4
Table 4: List of Equipments Used	4
Table 5: Wires to Connect ESP32 to MAX30100	8
Table 6: Summary of Figure 3	9

List of Acronyms

Acronym	Definition
IDE	Integrated Development Environment
BOM	Bill of Materials
MIT	Massachusetts Institute of Technology

1. Introduction

1.1 Context

The opioid crisis has caused over 10 000 overdose deaths in Canada since 2016. 94% of those are accidental and young Canadians aged 15-24 are the fastest growing overdose population. Overdoses are happening with both illegal and prescription drugs and touch every sector of the population. The main signs of an overdose is a low respiratory rate and low oxygen saturation levels in the blood. Our solution to help with this problem is to create a device for opioid users that can detect if they are overdosing and send a signal to loved ones/paramedics to alert them that something is wrong and indicate the location of the users [1].

1.2 Assumptions

Our group made a few assumptions when designing this product such as: an overdose being indicated by a decreased oxygen saturation level, oximeters having the ability to read accurate results on the ear, and user's being enticed to wear the device because of music playing abilities. Other important factors for this device are that it must be reliable and not be intrusive or hard to use for the users since they must want to use it consistently.

1.3 Structure

The structure of this document is the following. Firstly, we explain how to re-create the product by writing the materials needed and the different steps needed to create the product. Next, the document explains how to use the product effectively by guiding the user through the setup, operation and functions of the device. After that, the different tests done on the prototypes of the product and an explanation on maintaining the product are available. Finally, the document is finished with an analysis of the product which includes the lessons that we learned during this project and what we can take with us into future projects.

1.4 Product



Figure 1: Final prototype (Does not follow all specifications)

While we couldn't actually interview real opioid users, we gained our initial empathetic understanding of the problem through research online. Furthermore, our first client meeting really helped us to gain a more empathetic understanding of the problem. In the first client meeting, we learned that the opioid users in Canada are mainly single men in the trade. These people could be experiencing emotional damage. It was also crucial to keep in mind that the people that overdose on the drugs are not trying to commit suicide; they are people who use the drugs in hopes to function in their jobs and day to day life [1].

As seen in the tables below, we accumulated and prioritized the client's needs from our first client meeting.

Table 1: Interpreted User Needs [2]

Number	Interpreted Needs
1	Within the budget (\$100)
2	Able to use alone
3	Allows users to use both hands
4	Is discrete and non-obtrusive
5	Has a long battery life (at least a day) or can be recharged
6	Can detect an overdose and not cause deaths/injuries

The above table is the interpreted needs. With these user needs that were given to us, we created our main design criteria which is presented in the table below.

Table 2: Design Criteria for the Device [2]

Number	Design Criteria (Functional/Non-Functional)
1	Can read the blood oxygen saturation levels, locate user's location, and call/notify others for immediate help
2	Easy to use and understand for the user
3	Discrete/Small
4	Non-obtrusive
5	Wearable, comfortable, and secure (doesn't fall off the user)

Through analyzing the client's needs and our design criteria for the opioid users, our team developed this Problem Statement:

[2] A need exists for people who are taking opioids to monitor their health with a discreet, durable, and affordable device that can send their location to paramedics and call for immediate help when they are at risk of an overdose.

After we developed this problem statement, in Deliverable D [3], we brainstormed a lot of possible designs that our group can take on to make the actual prototypes for the opioid problem. This process can be seen in Deliverable D. Our group narrowed our design choices to three possible designs which we then presented in our second client meeting [4].

After the second client meeting, during our decision process and benchmarking we chose to go towards the Ear-clamp design as we liked the concept of the device looking like an accessory. We also liked the idea of having a main box/hub for the ear-clamp to be connected to. It was also suggested that the idea of an interchangeable model with the ear-clamp and glove would be good as some patients would prefer the glove model if they are not comfortable with the ear-clamp and vice versa. The Ear-clamp design is also very discrete as it will look like the user is wearing regular earbuds and is listening to music. Additionally, our design is different from other groups' design because the idea of having earbuds with the ear-clamp would be a great opportunity for the device to also provide an intervening sound for the patients if they are overdosing. Furthermore, this device would be linked to a phone application which will allow users to see their blood oxygen saturation levels and send their location upon overdose [4].

In summary, the main function of our product is that it can be securely clamped on to the user's ears, can read the blood oxygen saturation levels, locate user's location, and notify others for immediate help through the linked phone application when an overdose is detected.

2. How the Prototype is Made

2.1 BOM (Bill of Materials)

Below is the table showing the materials needed for the prototype and the cost for each.

Table 3: List of Materials and the Cost [5]

Material	Cost
Earbuds	\$13.88
Audio Jack	\$9.88
Storage Board	\$14.98
SD Card	\$5.94
Pulse Oximeter	\$11.50
ESP32S	\$13.99
Total	\$70.17

2.2 Equipment list

Below is the table showing the equipment used to create the product and where we got them.

Table 4: List of Equipments Used

Equipment	Source
3D printer	uOttawa MakerSpace
Solder	uOttawa MakerSpace
Solidworks	uOttawa Remote Apps
MIT app inventor	Internet
Arduino IDE	Internet

2.3 Instructions

We will explain all our major components. This includes the following: materials chosen, designs chosen, justification for choices, and a step by step procedure to recreate each component.

2.3.1 Box

Material:

The material which we chose to use was plastic since it was readily available and a cheap material to use. Although other materials could have been used, plastic was a good choice for our specific project since it was a non-expensive choice that was able to be purchased from the makerspace store. Since our project was in need of a box-like compartment that could store our electronics and could be held in one's pocket without being uncomfortable, we also knew plastic would be a good choice for this since it's semi resistant to water and small impacts. Although it is essentially an arbitrary choice on the material, plastic is an ideal choice since it's non-expensive and it is usually quick to print.

Design:

After seeing what would be needed for the earphone design, we found out that we needed a compartment to hold all of the components that would allow for the data from the sensor to be transmitted to the app that was created. When creating the box, we wanted to make something that people could carry comfortably in their pocket without causing discomfort and having it look sleek. It was decided as a group to model our design after modern smartphones. This included making the dimensions roughly the same size since we know that you can carry it in your pocket all day without causing discomfort and damage. This design was made quick after we decided to use the earphone design but one underlying problem was our dimensioning. We were consistently changing the components in order to get the prototype to run as smoothly as possible so it was difficult to get specific dimensions of the components [6].

Calculations:

From group discussion, the dimensions we agreed upon were:

Length: ~ 7.5 inches

Width: ~ 3.5 inches

Height: ~ 2 inches

These dimensions are slightly larger than an iPhone XR, but it was hypothesized that it would still be able to fit comfortably in the user's pocket [6,7].

Instructions:

The design of the box all the way through solidworks modelling of the box will be demonstrated using a numbered list.

1. Open the “remote apps” application from the University of Ottawa - Engineering Faculty website .
2. Open the Solidworks application and select a “Part Assembly”.
3. Choose the “Sketch” command at the top left of the command window
4. Choose “Front View” to sketch. This allows you to sketch something as if it were facing you directly eye to eye.
5. After this, select “Line” at the top of the screen. This allows you to draw straight lines from a start point to an end point.
6. Since the “Front view” has been chosen, use the “line” command to sketch the outline of a rectangle.
7. At the left of the screen when you have drawn a line, type in your desired length of the line in inches.
8. Then, click the checkmark to indicate the sketch being done.
9. Click “Extruded Boss/Base”, and type in the size of this dimension. This will be the length of the box. At this step you should have a 3D rectangular box.
10. At the top of the screen, click “Fillet” and choose the fillet size. The recommended fillet size should be under or around 1.0 inches (For a box, roughly the size of an iphone).
11. Select the line segment of the rectangle and fillet the edge. Repeat for all edges of the rectangle.
12. Once all the edges of the rectangle are filleted, select the checkmark to symbolize you are done with the fillet option.
13. The last step is to make the Hole for the wire to come out of the box. Select the “Hole Wizard” command.
14. Select the first out the eight types of holes and select “#2” as the size.
15. Then, select the face of where you would like the hole to be.
16. The box has been completed and is ready to be saved.

2.3.2 Clamp

Material:

For the clamp we went with plastic 3d printer filament. This material was easily accessible and inexpensive. This material was great for the clamp since it was very light meaning that it could fit on the ears of the users with more comfort.

Design:

When we decided on the earphone design worked best. We needed a way to connect the sensor to the ear and have the wires be hidden. We decided that a 3d printed clamp would be best. The clamp needed to be lightweight, not too visible and able to funnel the wires so that they are hidden. The clamp design that we came up with was as small as we could make it, it used light weight materials, and was able to funnel the wires so that they are hidden.

Calculations:

From group discussion, the dimensions we agreed on were:

Length: ~ 1.9 inches

Width: ~ 1.6 inches

Height: ~ 0.8 inches

Instructions

1. Open the “remote apps” application from the University of Ottawa - Engineering Faculty website.
2. Open the Solidworks application and select a “Part Assembly”.
3. Choose the “Sketch” command at the top left of the command window
4. Choose “Front View” to sketch. This allows you to sketch something as if it were facing you directly eye to eye.
5. After this, select “Line” at the top of the screen. This allows you to draw straight lines from a start point to an end point.
6. Since the “Front view” has been chosen, use the “line” command to sketch the outline of the side of the clamp.
7. At the left of the screen when you have drawn the outline, type in your desired length of the lines in inches.
8. Then, click the checkmark to indicate the sketch being done.
9. Click “Extruded Boss/Base”, and type in the size of this dimension. This will be the width of the clamp. At this step you should have a 3D general shape of the clamp.
10. Now it's time to make holes in to channel the earphone and sensor wires through.
11. This can vary depending on the sensor used and the size of the wires.
12. The way this works is a shape is drawn on the surface where the channel starts and then the extrude cut feature is used.

2.3.3 Electrical

ESP32/MAX30100:

The ESP32 was used as the microprocessor for the device because of its cheap price and small frame. The arduino nano was also considered as an option, but due to its significant price difference (9\$ extra) we chose the ESP32.

The MAX30100 was chosen as the oximeter for its cheap price and small frame. It was also crucial that it is compatible with the microprocessor, so the information can be obtained by the microprocessor. The MAX30100 has an official library for coding with the microprocessor.

Design:

The following figure below is the wiring of the ESP32S and MAX30100 on the breadboard.

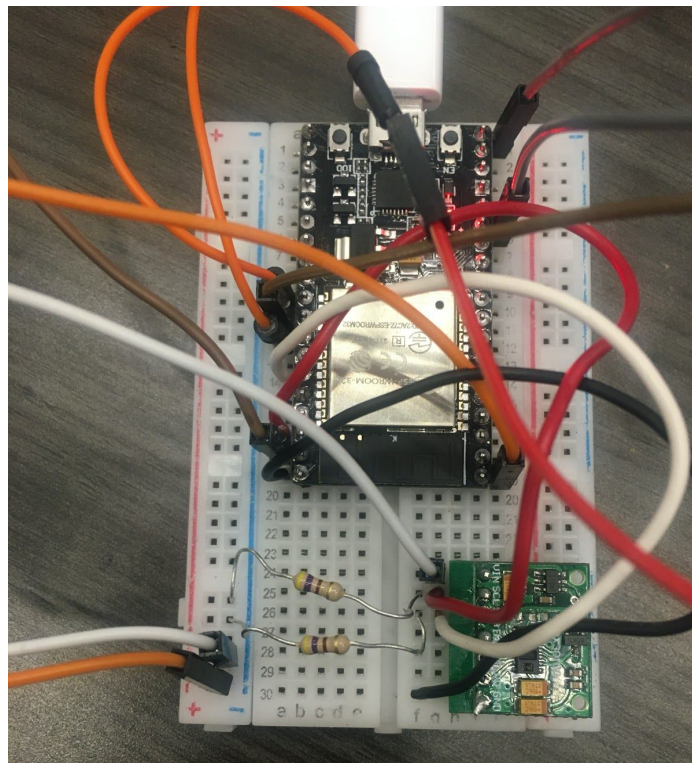


Figure 2: Wiring On Breadboard

Configuration:

The configuration of wires to connect the ESP32 to the MAX30100 is shown above as well as in the table below. **Please Note** that in the final design the MAX30100 sensor will not be located on the breadboard, but rather connected by wires next to the ear pods.

Table 5: Wires to Connect ESP32 to MAX30100 [6]

ESP32S	MAX30100	Used Resistor
3v3	Vin	N/A
Gnd	Gnd	N/A
SDA-21	SDA	4.7 K Ω
SCI-22	SCI	4.7 K Ω

ESP32/SD Card:

The SD card adapter was chosen purely for its cheap price. Since SD card adapters are small, practically all the candidates fit the device's standards. Additionally, the Audio Jack was chosen for its cheap price and ability to connect to a microprocessor.

The following figure below is the wiring for the SD card and the headphone jack which was not included in our latest prototype. But this is the wiring for the SD card that would have been done for the actual product.

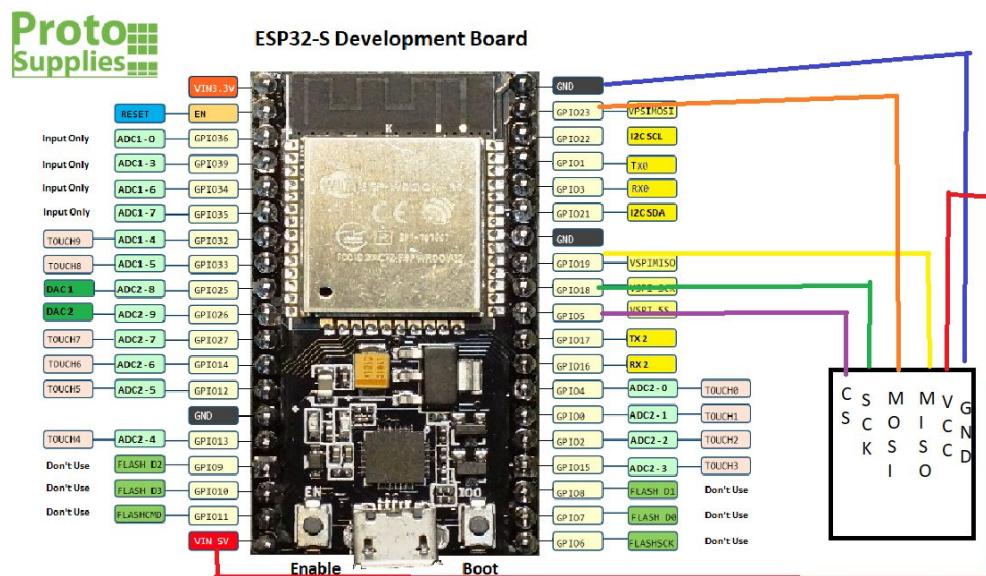


Figure 3: Wiring and Pin Information for SD to ESP-32S [8]

Table 6: Summary of Figure 3

ESP32S	SD Card Adapter
GPIO5	CS
GPIO18	SCK
GPIO23	MOSI
GPIO19	MISO
Vin 5V	VCC
GND	GND

2.3.4 Software

ESP32 Code:

IDE:

The Arduino IDE was used to upload code directly to the ESP32 because it was the software that we learned in one of the labs. This gave the team added expertise with this software. Additionally, this IDE is officially for uploading code to arduino-like devices and has libraries that relate to the MAX30100 sensor and ESP32.

Functions:

- Read data from MAX30100 oximeter and store into variables.
- Connect ESP32 to the specially made app via bluetooth.
- Send data stored in variables to the app.
- Receive data from the app to control music and change settings.

Instructions:

1. Acquire Arduino IDE from Arduino Software (IDE) article [9].
2. Acquire MAX31850_OneWire library from the Arduino MAX30100 GitHub [10] and include it into the IDE by going to Sketch->Include Library->Add ZIP. and insert the folder from the website.
3. Include ESP32 into the IDE by going to File->Preferences-> and input https://dl.espressif.com/dl/package_esp32_index.json into the Additional Boards Manager URLs.
4. Go to Tools->Board: and select the ESP32 Dev Module
5. Select the template by going to File->Example->MAX30100lib->MAX30100_Minimal
6. Select the template by going to File->Example->BluetoothSerial->SerialToSerialBT

7. Transfer all code from the BluetoothSerial example to the MAX30100 example by inputting the setup code to the other setup code and the loop code to the other loop code. Don't forget the #includes and BluetoothSerial SerialBT;
 8. In the loop, write "SerialBT.write(pox.getSpO2());". This will send the user's blood oxygen saturation to the app via bluetooth.
 9. Upload code to ESP32 using an android charging cable.
- The original code is located at O2Pod's MakerRepo.

App Code:

IDE:

MIT App Inventor was used to create the app for the android device. It was used because of its simplistic design and coding capabilities. They use block coding which has preset functions that can combine with others to complete tasks. They also provide essential tools like location sensors and texting features which is integral to the design of our product. Lastly, there are many resources for MIT App Inventor, so it is much easier to problem shoot and gather ideas to improve the application.

Functions:

- Be a User's interference. Change setting.
- Display information sent from the ESP32 derived from the MAX30100 oximeter.
- Detect an overdose using data from ESP32.
- Acquire location of user's phone using the phone's GPS.
- Sending text including user's condition and location to a desired recipient when an overdose is detected.

Instructions:

1. Visit the MIT App Inventor website [11].
2. Create a new project and recreate a similar UI to the following image. The requirements are the following: a listpicker for the bluetooth symbol, a bluetooth client, a clock, a location sensor, texting, and 2 text boxes.

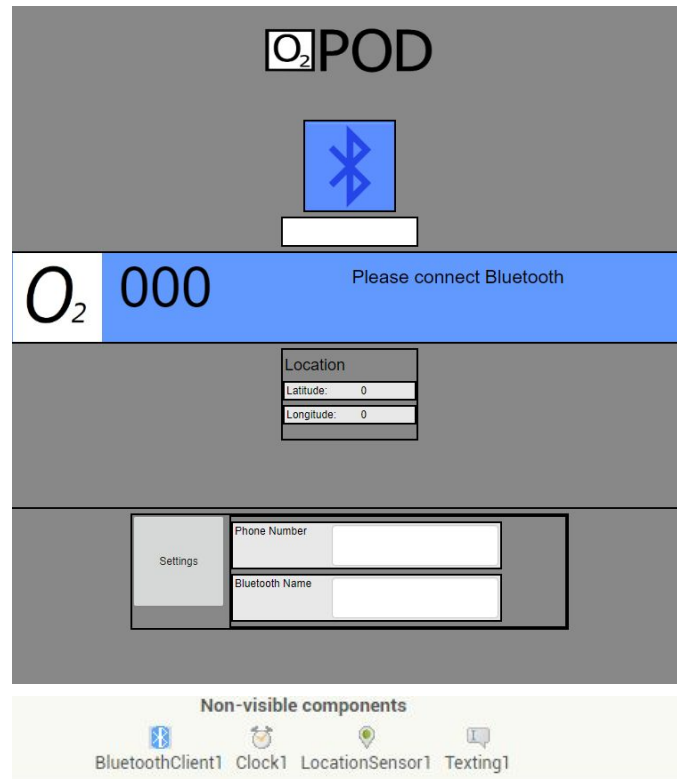


Figure 4: UI of the O2POD Application

3. Coding the bluetooth connection

Select functions from their respective entities and combine them as shown below.

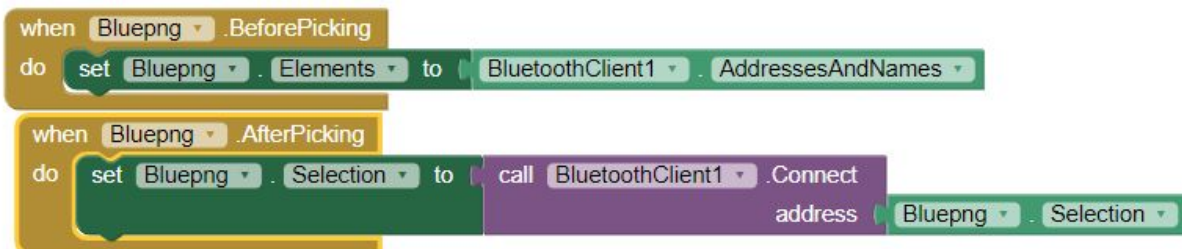


Figure 5: MIT App Inventor code for Bluetooth connection with ESP32

4. Coding to get data from ESP32 and display it

Select functions from their respective entities and combine them as shown below.

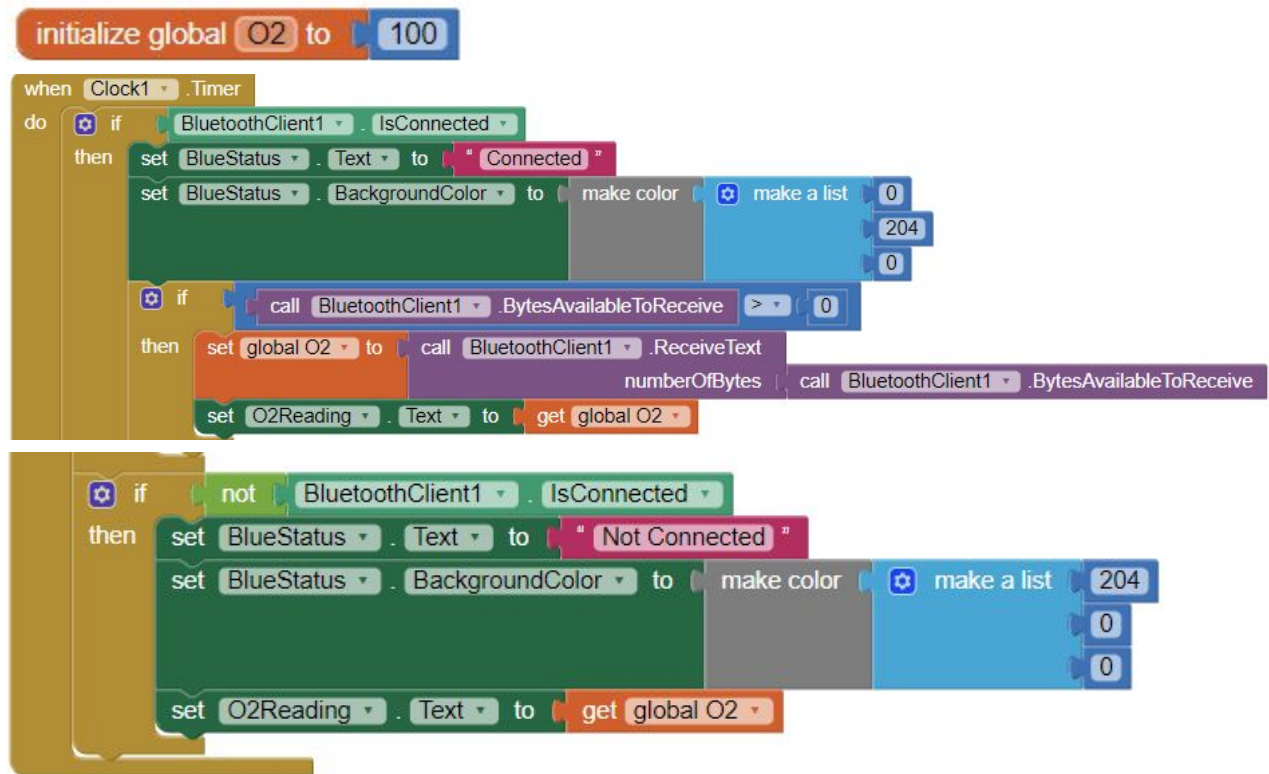


Figure 6: MIT App Inventor code for receiving data from ESP32

5. Texting code

Select functions from their respective entities and combine them as shown below.

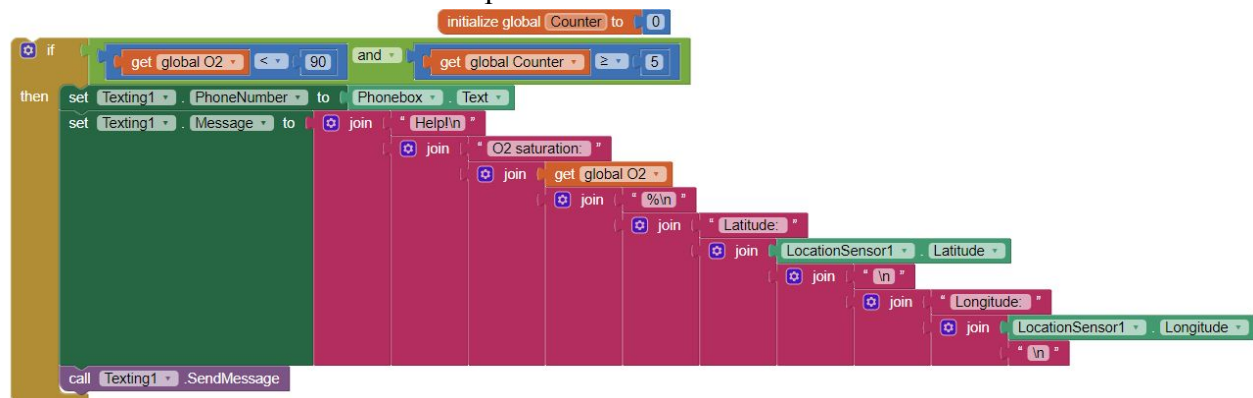


Figure 7: MIT App Inventor code for sending a text with location and condition

6. Location code

Select functions from their respective entities and combine them as shown below.



Figure 8: MIT App Inventor code for getting the location of the android device

3. How to Use the Prototype

3.1 Functions

Box: The box is to be treated as a device that is to be put in the users pocket. The box was made to have smooth edges and be dimensionally similar to a modern smartphone so it could cause no discomfort.

Clamp: The clamp is a device that is meant to house the pulse oximetry sensor and funnel the wires so that they are hidden. The clamp is designed to also protect the pulse oximetry sensor from getting damaged.

App: The application is used to interact with the device. It was made to be simplistic and easy to use, so the user can instinctively use it without problems. It uses bluetooth to connect to the O2Pod device and is used to detect if there is an overdose based on the data received from the device. It can text a chosen phone number in case of an overdose providing the location of the user and their current condition.

3.2 Operation

Explain how the user safely operates the prototype.

Step by step procedure

1. Before use, check the charge in the device by inspecting the power bank module. If there is only one light on, the device is at 25% batteries and requires charging.
2. Open the app and connect to the O2Pod by tapping on the bluetooth symbol and selecting the device.
3. Insert the phone number you would like to contact incase of an overdose in the settings menu.
4. Turn on GPS features and Cellular features.
5. Insert the earbuds and attach the clamp on your earlobe or any other flat surface. If your android device displays consistent results and the clamp is on securely, proceed.
6. Keep the box portion of the O2Pod securely on your person and with enough slack on your earbuds as to not pull on the clamp.
7. Store the device in a dry environment and do not let the inner components of the device have exposure to liquids.

3.3 Setup

Explain how the user installs the prototype to use (if applicable).

1. Download the .apk file from MakerRepo [See Appendix] for O2Pod and install it on your android device.
2. Charge and turn on the power bank by pressing the power button.
3. Go into your android device's settings and search with bluetooth to find a device called "ESP32test" and connect.
4. Turn on GPS features and Cellular features.
5. Provide phone number of desired recipient.

4. How to Maintain the Prototype

4.1 Tests

The following list below are the things we wanted to test.

- If the device fits comfortably in the pocket of the user.
- If the device can pick up accurate blood oxygen saturation levels and location (the longitude and latitude) of the user.
- If the device can send the data collected to the phone application, and the user can easily see his or her data on the application.
- If the phone application can successfully send a text to a third party upon overdose.
- If the device can be worn comfortably by a person, is secure on the ear, and can successfully do the following objectives above.

The following are the results for the tests in their respective order:

- This test showed that any user could comfortably fit the device in their pockets. The only users who could not do so were users that did not have clothes with pockets.
- The device could pick up accurate blood oxygen saturation levels and could indicate the location of the user when overdosing.
- The device could send data collected from the sensor to the phone application in a visible way for the user to see clearly.
- The phone application could successfully send a text to a third party when the sensor started sensing a blood oxygen level that would indicate an overdose.
- The device could be worn comfortably by a person and it remained secure on the ear.

4.2 Maintenance

Due to the demographic that we targeted for our product, the design was made to be as low maintenance as possible. The only things required to maintain in our product are the wires connecting the sensor to the box as they can be torn and tangled sometimes and it is important to keep them intact. In addition to this, it is important to make sure that the box does not get submerged in any liquid or receive any significant physical impacts so that the ESP32 and the wiring inside the box do not get affected. Also, it is important to keep the phones that the device is connected to charged so that they receive a text when the user is overdosing.

5. Conclusions and Recommendations for Future Work

5.1 Analysis

- One of the major obstacles was the COVID-19 virus which left the Makerspace facilities not to be used. As well, the printers were not able to be accessed.
- At first it was expected that the sensor would take some time to get it to work, but it took longer than expected since the wiring and placement of resistors and such had to be fixed a lot. As well, the actual 3-D production of the box took long since exact measurements of the components in the box could not be measured.
- During the testing phase, a problem which arose was the sensor. After continuing testing, it was unable to sense any readings and after checking everything, we came to the conclusion that our sensor broke. From this, it was suspected that the reason it had broken was because of our battery.
- After constantly ordering materials, It was seen that we could have saved money and bought something cheaper. For example, when the battery was bought, it was concluded that it was too heavy and strong for our device which was one of the reasons we believe the sensor stopped working at first.
- Lastly, during the testing phase, group members were starting to get flooded with midterm exams but it was agreed upon to meet each weekend early in the morning to work on the deliverables.

5.2 Latest Status and Recommendations

Throughout the 3 months that our group worked on this device we faced several problems that are mentioned above. Below is what we suggest to be the most productive avenues for future work to avoid the problem:

- Work on tasks as soon as possible as some tasks took longer to finish than what we expected
- For some tasks, it was better if one person only worked on it instead of the whole group as this gets specific tasks done quickly
- For some tasks it was necessary to ask for help from an actual person with the correct knowledge.
- Watching videos online can help especially with the wiring parts as it's easier to understand how to wire the components together by seeing people actually do it on a video.
- Help from other groups or other skilled students/people such as the CEED workers

In Deliverable H [8], the latest status of our device was discussed. This latest prototype can be seen in Figure 1 in this report. In summary, our latest prototype is a comprehensive prototype. Our comprehensive prototype is able to pick up the users' accurate blood oxygen saturation level data and location (their longitude and latitude), send the data to the Android

application, and can successfully send a text to a third party upon overdose. However, it is obviously not our ideal final product and is not aesthetically pleasing.

The following is the recommendations of what the further work we would have done if the pandemic did not occur:

- Solidworks & Printing of box and ear-clamp
- Work on the wiring for the audio jack and SD card for processor
- Fix the aesthetics/tube the wires together/put electronic components together
- Get final user feedback for our latest prototype and use that feedback to create the actual product

5.3 Ideal Product

In a Covid-19 free world, the final prototype was on track to be completely finished. All of the major components had been finished or were very close to being finished and the final product was on track to be finished by design day on March 26, 2020. The final product would have been an earphone design with the wires of the earbuds connected to a box compartment that would be held in the user's pocket. The sensor would have been connected to a 3D printed ear clamp which would help disguise the sensor and the earbud together which would have created a discrete design. The ideal product would have been able to accurately sense the user's blood oxygen saturation from the ear and deliver this data to the android application that was created.

In point form, the following is what our ideal final prototype should've looked and functioned like:

- All electronics are stored in a 3D printed box with filleted edges
- The sensor is stored in the 3D printed ear-clamp which is connected to the earbuds
- The SD card holder is connected to the ESP32S, but is stored in the box also along with the other electronic parts
- The earbuds can be connected to the audio jack that is connected to the ESP32S
- The wires are tubed together for easy handling and aesthetics
- When users wear the device, it can accurately pick up the users' SpO2 and location
- The device can send the information of the user to the application
- When the user is overdosing, the device will play intervening sounds to alert the user is overdosing
- The device will also send a text to a third party for help

6. Bibliography

- [1] Team 6. *Deliverable B: Needs identification and problem statement*, GNG 1103C, 2020.
- [2] Team 6. *Deliverable C: Design criteria*, GNG 1103C, 2020.
- [3] Team 6. *Deliverable D: Conceptual design*, GNG 1103C, 2020.
- [4] Team 6. *Deliverable E: Project plan and cost estimate*, GNG 1103C, 2020.
- [5] Team 6. *Deliverable J: Project presentations*, GNG 1103C, 2020.
- [6] Team 6. *Deliverable F: Prototype I and customer feedback*, GNG 1103C, 2020.
- [7] Team 6. *Deliverable G: Prototype II and customer feedback*, GNG 1103C, 2020.
- [8] Team 6. *Deliverable H: Prototype III and customer feedback*, GNG 1103C, 2020.
- [9] “Arduino Software,” *Arduino*. [Online]. Available:
<https://www.arduino.cc/en/main/software>. [Accessed: 20-Feb-2020]
- [10] Oxullo, “Arduino MAX30100,” *GitHub*, 10-Mar-2018. [Online]. Available:
<https://github.com/oxullo/Arduino-MAX30100>. [Accessed: 20-Feb-2020].
- [11] “Explore MIT App Inventor,” *MIT App Inventor*. [Online]. Available:
<https://appinventor.mit.edu/>. [Accessed: 20-Feb-2020].

7. APPENDIX

<https://makerepo.com/rikkiromana/o2pod-an-opioid-overdose-monitoring-device>