User Manual Instructions
This document is a template of a user manual report. The client may wish to make improvements on the prototype or need to fix it if something goes wrong or another group of students may work to make a more rugged prototype. The report needs to be clear for someone else to use, maintain or reproduce the project. Include as many images and diagrams for a better understanding.
In general, if you are not sure exactly what to include, imagine that this document was the only thing that you had. Imagine also that your job was to add a new feature to the project that is described in your document. What would you need to know?
In the case of very large design files, you may need to store them in MakerRepo. For such files, then you still need to provide a description of what is contained in those files in your document and enough information so that the files can be used effectively.

GNG1103/2101
Design Project User Manual

[TITLE]

Submitted by:
[Loh Yi Ting, STUDENT NUMBER]
[Wu Yi Jun, STUDENT NUMBER]
[Yang Zhao, STUDENT NUMBER]
[Isaac Seow Yew Keat, 300019029]

5th April 2020
University of Ottawa

Abstract
The opioid crisis continues to plague Canada. As of 2016, opioid overdose has claimed over 10,000 lives; Out of which 94% of the deaths were not incidental. Overdoses involve both prescribed, non-prescribed and even non-legal drugs. The opioid crisis continues to affect different parts of the Canadian population with Canadian youths being the fastest growing group to affected by the opioid crisis. As a result, Tali Cahil, a registered nurse who works in Sandy

Hill Community Centre, has approached the University of Ottawa's Engineering Design (GNG1103) class to seek out innovations that could help curb the growing overdose numbers. This paper derives from the research and findings done by Group 4 of the GNG1103 class. This paper aims to provide readers with a better insight on the opioid overdose crisis, a thorough description on the process that was used to create the group's opioid overdose prevention device and a summary on the lessons learnt.

Table of Contents

List of Figures

Insert your list of figures here.

List of Tables

Insert your tables here

List of Acronyms

Acronym	Definition

1 Introduction

At the beginning of the course Tali presented the Opioid Crisis to the course. She explained the need for a device that can detect an Opioid Overdose and stated her requirements:

- The device must accurately read the user’s SpO2 and BPM
- The device must be able to send out an alert during an overdose
- The device is preferably wearable
- The device must be discreet
- The device must be able to operate for a sustainable period of time without requiring charging

Each group was also given a 100CAD budget to adhere to and full access to UOttawa’s Makerspace facilities. Given the following requirements

1.1 The Opioid Overdose Crisis

The Opioid Overdose Crisis has been an issue in Canada’s community for a long time. Opioids are a form of painkillers; they are commonly taken to temporarily suppress the user’s pain. Opioids lower the breath rate of the user and overdoses lead to the user losing consciousness and, in many cases, causes the user to die. The most common way to save the user is to administer Naloxone, a medication that temporarily stops the Opioid’s breath loss effect. This give medical personnel enough time to act and to save the victim.

Due to the nature of the overdose, medication must be administered swiftly in order to save the victim and prevent any further potential brain damage done to the victim. Currently, there are 13,900 deaths that are linked to Opioid overdoses in Ontario (from January 2016 to June 2019). Out of these numbers:

- 75% were male.
- 47% were unemployed, 18% were employed and 33% had an unknown employment status.
- 50% of the victims were alone during the time of the overdose.
- 50% of the victims required a resuscitation.
- More than 50% of the victims were between the ages of 25 to 44.

As these numbers continue to grow and affect present day Ottawa, it is without a doubt that immediate action is required to slow or even, curb this crisis.

1.1.1 The Device

The device will be built centred around the Arduino nano, a node Micro-controller Unit that can be programmed to carry out tasks. The device consists of a oximeter subsystem and an alerting subsystem. The device will be house within a watch-like case design that the user will wear on their wrist.

The oximeter subsystem will be responsible for tracking the user's vitals. The subsystem will be tuned so that it is as accurate as possible. This allows the device to function as an oximeter but equipped with an alert subsystem then can alert important personnel. The oximeter subsystem will have a screen that allows the user to be wary of their vitals. This also could reduce the time for medical personnel to act as they are presenting with some degree of information upon arrival.

The alert subsystem will be set up to co-work with the oximeter subsystem. Its role is to alert a contact once the oximeter subsystem detects vital readings that are akin to an overdose. The alert subsystem has to act swiftly and with little to no time wasted as far as possible.

The casing will have a discrete design, to reduce attention drawn to the user due to potential social discrimination. The casing will be lightweight and easily wearable. It should not affect the user's mobility and day to day movements at all. The casing will have to be optimized so that the two subsystems can function as designed.

It is therefore essential that the three main components come together to function well as a unit. As this device will be the lifeline of the user, it is important that various tests are run to ensure that the device operates consistently and optimally.

1.1.1.1 Prototyping

Prototyping is essential part of the process in order to gain more information of our device's capabilities. During the prototyping phase it is important that extensive testing is conducted in order to make suitable adjustments and constant improvements on the device. The prototyping phase was split into three main sessions. Each session had a different objective and focused on the different subsystems.

The first session focused on the exterior outlook of the device. The design was planned to follow strictly to the requirements listed by our client, Tali. The design was planned to look like a bracelet, which modeled the idea of an everyday object so as to keep the anonymity of the device. Acrylic was chosen to be the main material of the casing due to its dexterity and waterproofing ability. Acrylic is also a lightweight material that would less likely obstruct the user's movements, this makes it extremely convenient to put on. As for the strap, we have decided to use a rubber material commonly used in watch straps. As aforementioned this models the device after a watch/bracelet, allowing it to avoid drawing attention from others. The strap design's focus is always to have the device stay on the user, this also ensures that the oximeter component is in contact with the user's skin as the oximeter subsystem requires skin contact to function. With these in mind, the design was carried out to be as in Figure 1.(Insert picture after this)

The second session revolve around the oximeter subsystem. This component is the most vital out of the three, as it serves the most basic function of tracking the user's health status. In order to keep a close eye on the user's health the oximeter subsystem must be fully functioning and optimal. The readings taken from the oximeter must be accurate to ensure that false readings are not taken. The oximeter subsystem uses the MAX30100 as the blood pulse oxygen sensor; it is used to sense the blood oxygen level and pulse rate of the user, the information take from the sensor will be fed to the Arduino Nano and relayed to an OLED screen which will then display the readings taken for the user to see.

The final session was used to set up the alerting subsystem and to debug the entire device as a whole working unit. The alert system consists of a GSM module, this module connects to a 2G SIM card and is able to send text messages to a pre-set contact. In order to determine the reliability of the module, we first sent a text message to a test phone while nearby the module, once we have determined that the module is able to consistently send messages, we increased the distance from the module to the test phone. This helps us to determine the range of functionality of our alert subsystem. An app was also created as a User-Interface, allowing our users to manually key in their location and to set an emergency contact. It is important to note that 911 does not take in calls from non-human parties.

Finally, after we determined the functionality of each subsystem, we soldered all the components onto the Arduino nano. We ran the electrical components as a whole unit multiple times to ensure that the device was consistent. Through this series of debugging we made sure that there are no potential critical failures in the device's operating system. During this same timeline, we had the case 3D-printed using the Makerspace facilities. Once the case was ready, we fitted the case in and tested the device a few times again for consistency's sake.

[prototype picture]

2 How the Prototype is Made

Explain in detail how the prototype was built including design considerations and calculations. Separate it into categories (mechanical, electrical, software, etc) and explain the importance of each. If there are options for material that you considered or analyzed, explain which ones might be feasible and which ones that your analysis shows are not feasible (with some results to back up your statements, as required).

For example, if stainless steel was an arbitrary choice for a particular part, you could indicate that other materials (e.g. plastic or wood) might also be an option, but were not tested. However, if metal that resists corrosion is the basic requirement and you tested several materials before choosing stainless steel (i.e. the choice is not arbitrary) then you can indicate this here, along with supporting data. Sometimes, material needs to be swapped, if no longer obtainable or if no longer cost-effective, present any work that you did that might help another designer make material substitutions or even note the basic requirements (e.g. must resist corrosion in a humid room environment for 30 years).

The same is true for time-critical portions of software or expensive/sensitive electronic functionality. Basically, if you were worried about a portion of the design and "settled" on a particular solution or method, then it needs to be documented. This includes the testing or analysis that you did to arrive at that specific solution.

2.1 Mechanical

2.1.1 BOM (Bill of Materials)

List all the parts and materials in this category (if available, include links to each item).

2.1.2 Equipment list

List all the equipment that was needed to build this part.

2.1.3 Instructions

2.1.3.1 Oximeter subsystem

For the Oximeter subsystem, the MAX30100 Blood Oxygen and Pulse oximeter sensor is used. In order to for the sensor to properly function, it is important to de-solder the three surface mounted resistors. Next, three 4.7 k Ohm resistors are required to be externally connected to the system. The reason is that the MAX30100 sensor chip is incompatible with the current output of the Arduino nano and therefore requires this adjustment for the sensor to properly function.

The following figure shows the wiring diagram for the oximeter subsystem.

(Image taken from https://www.hackster.io/umar-sear/arduino-heart-rate-monitor-a8e9e1)

Once the wiring is in place, plug the Arduino nano to the Arduino IDE console, and run the following code:

(Insert code)

The device should function as follows:

(Insert picture)

2.1.3.2 Alert subsystem

The Alert subsystem uses the GSM 800L module, it sends text messages to a contact that has been set in the system using a 2G MicroSIM card. The subsystem requires a battery to power the circuit as the Arduino nano's power output is not sufficient for the GSM 800L. The battery will also be required to power the final device as a whole so it is important to calculate battery requirements before purchasing.

The below figure shows the wiring for the subsystem:

Following the wiring, the code must be uploaded to run and test the subsystem. At this point, the GSM800L should light up.
(Insert code and image of working GSM)

3 How to Use the Prototype

Explain in detail the functions of the prototype and how it works.
Explain how the user safely operates the prototype.
Explain how the user installs the prototype to use (if applicable).

4 How to Maintain the Prototype

Explain the tests that were done on the prototype for validation of the final design. Present all of the applicable results that you obtained (i.e. data collected; performance graphs, etc.). List any issues or special requirements for sustained usage.
Describe regular maintenance that should be performed on the prototype to avoid failure.
Explain what parts may be prone to break and need to be replaced.

5 Conclusions et Recommendations for Future Work

Summarise your lessons learned and your work and suggest most productive avenues for future work.

6 Bibliography

APPENDICES

APPENDIX I: Design Files

Include all design files with explanations so that the clients or other students from following semesters can continue your project. Also provide the MakerRepo link to your project.

APPENDIX II: Other Appendices

You can include other critical and important work here. Maybe they are not important in the structure of this document but need to be included.

This document is a template of a user manual report. The client may wish to make improvements on the prototype or need to fix it if something goes wrong or another group of students may work to make a more rugged prototype. The report needs to be clear for someone else to use, maintain or reproduce the project. Include as many images and diagrams for a better understanding.

In general, if you are not sure exactly what to include, imagine that this document was the ***only*** thing that you had. Imagine also that your job was to add a new feature to the project that is described in your document. What would *you* need to know?

In the case of very large design files, you may need to store them in MakerRepo. For such files, then you still need to provide a description of what is contained in those files in your document and enough information so that the files can be used effectively.

GNG1103/2101

Design Project User Manual

[TITLE]

Submitted by:

[Loh Yi Ting, 300134587]

[Wu Yi Jun, STUDENT NUMBER]

[Yang Zhao, STUDENT NUMBER]

[Isaac Seow Yew Keat, 300019029]

5th April 2020

University of Ottawa

Abstract

The opioid crisis continues to plague Canada. As of 2016, opioid overdose has claimed over 10,000 lives; Out of which 94% of the deaths were not incidental. Overdoses involve both prescribed, non-prescribed and even non-legal drugs. The opioid crisis continues to affect different parts of the Canadian population with Canadian youths being the fastest growing group to affected by the opioid crisis. As a result, Tali Cahil, a registered nurse who works in Sandy Hill Community Centre, has approached the University of Ottawa's Engineering Design (GNG1103) class to seek out innovations that could help curb the growing overdose numbers. This paper derives from the research and findings done by Group 4 of the GNG1103 class. This paper aims to provide readers with a better insight on the opioid overdose crisis, a thorough description on the process that was used to create the group's opioid overdose prevention device and a summary on the lessons learnt.

Table of Contents

List of Figures

Insert your list of figures here.

List of Tables

Insert your tables here

List of Acronyms

Acronym	Definition

1 Introduction

At the beginning of the course Tali presented the Opioid Crisis to the course. She explained the need for a device that can detect an Opioid Overdose and stated her requirements:

- The device must accurately read the user's SpO2 and BPM
- The device must be able to send out an alert during an overdose
- The device is preferably wearable
- The device must be discreet
- The device must be able to operate for a sustainable period of time without requiring charging

Each group was also given a 100CAD budget to adhere to and full access to UOttawa's Makerspace facilities. Given the following requirements.

1.1 The Opioid Overdose Crisis

The Opioid Overdose Crisis has been an issue in Canada's community for a long time. Opioids are a form of painkillers; they are commonly taken to temporarily suppress the user's pain. Opioids lower the breath rate of the user and overdoses lead to the user losing consciousness and, in many cases, causes the user to die. The most common way to save the user is to administer Naloxone, a medication that temporarily stops the Opioid's breath loss effect. This gives medical personnel enough time to act and to save the victim.

Due to the nature of the overdose, medication must be administered swiftly in order to save the victim and prevent any further potential brain damage done to the victim. Currently, there are 13,900 deaths that are linked to Opioid overdoses in Ontario (from January 2016 to June 2019). Out of these numbers:

- 75% were male.
- 47% were unemployed, 18% were employed and 33% had an unknown employment status.
- 50% of the victims were alone during the time of the overdose.
- 50% of the victims required a resuscitation.
- More than 50% of the victims were between the ages of 25 to 44.

As these numbers continue to grow and affect present day Ottawa, it is without a doubt that immediate action is required to slow or even, curb this crisis.

1.1.1 The Device

The device will be built centred around the Arduino nano, a node Micro-controller Unit that can be programmed to carry out tasks. The device consists of an oximeter subsystem and an alerting subsystem. The device will be housed within a watch-like case design that the user will wear on their wrist.

The oximeter subsystem will be responsible for tracking the user's vitals. The subsystem will be tuned so that it is as accurate as possible. This allows the device to function as an oximeter but equipped with an alert subsystem then can alert important personnel. The oximeter subsystem will have a screen that allows the user to be wary of their vitals. This could also reduce the time for medical personnel to act as they are presenting with some degree of information upon arrival.

The alert subsystem will be set up to co-work with the oximeter subsystem. Its role is to alert a contact once the oximeter subsystem detects vital readings that are akin to an overdose. The alert subsystem has to act swiftly and with little to no time wasted as far as possible.

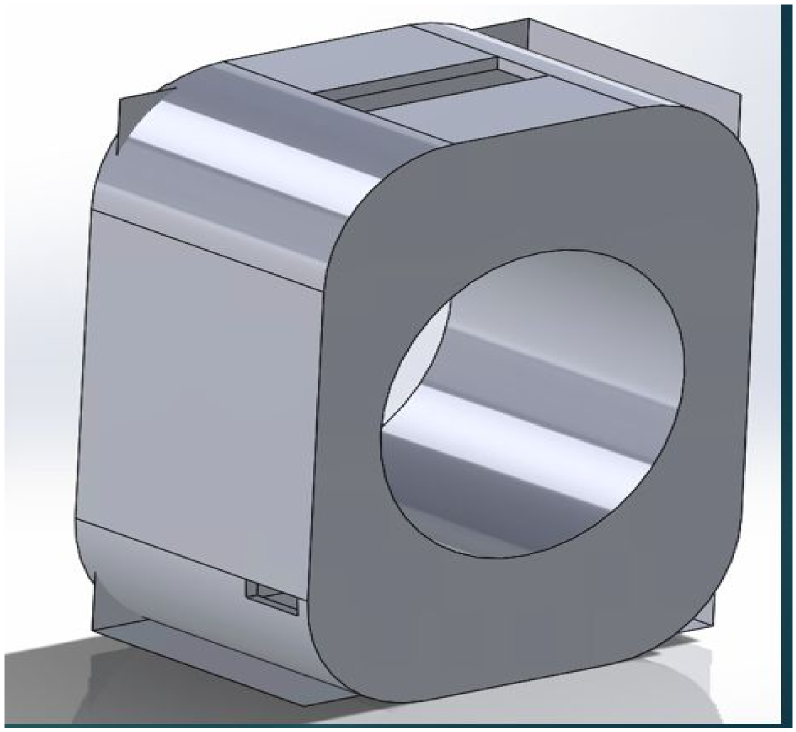
The casing will have a discrete design, to reduce attention drawn to the user due to potential social discrimination. The casing will be lightweight and easily wearable. It should not affect the user's mobility and day to day movements at all. The casing will have to be optimized so that the two subsystems can function as designed.

It is therefore essential that the three main components come together to function well as a unit. As this device will be the lifeline of the user, it is important that various tests are run to ensure that the device operates consistently and optimally.

1.1.1.1 Prototyping

Prototyping is an essential part of the process in order to gain more information of our device's capabilities. During the prototyping phase it is important that extensive testing is conducted in order to make suitable adjustments and constant improvements on the device. The prototyping phase was split into three main sessions. Each session had a different objective and focused on the different subsystems.

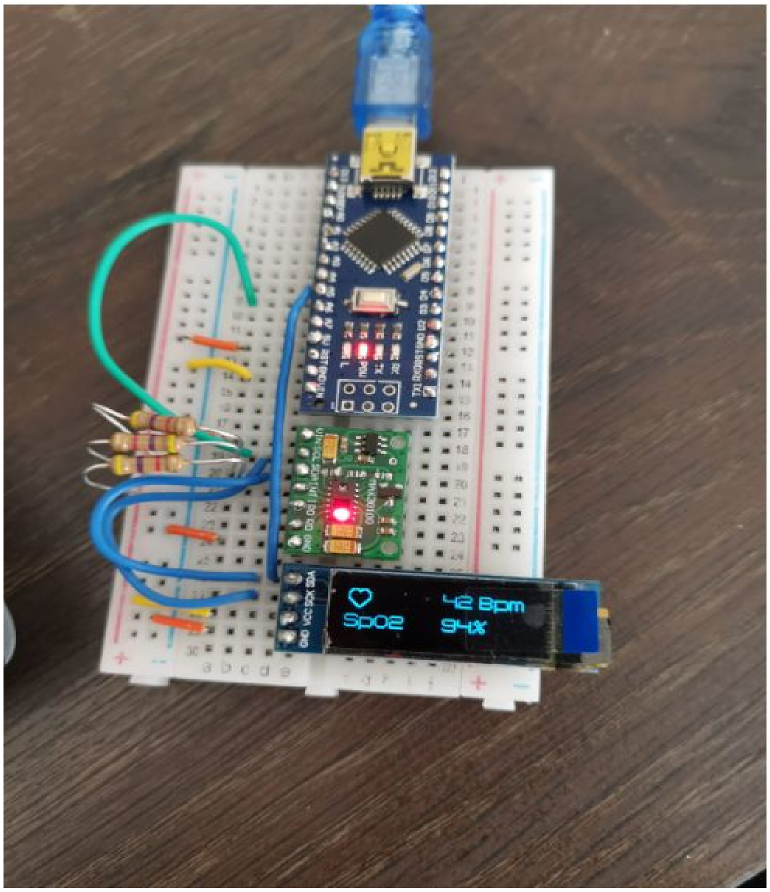
The first session focused on the exterior outlook of the device. The design was planned to follow strictly to the requirements listed by our client, Tali. The design was planned to look like a bracelet, which modeled the idea of an everyday object so as to keep the anonymity of the device. Acrylic was chosen to be the main material of the casing due to its dexterity and waterproofing ability. Acrylic is also a lightweight material that would less likely obstruct the user's movements, this makes it extremely convenient to put on. As for the strap, we have decided to use a rubber material commonly used in watch straps. As aforementioned this models the device after a watch/bracelet, allowing it to avoid drawing attention from others. The strap design's focus is always to have the device stay on the user, this also ensures that the oximeter component is in contact with the user's skin as the oximeter subsystem requires skin contact to function. With these in mind, the design was carried out to be as in Figure 1.(Insert picture after this)



The second session revolves around the oximeter subsystem. This component is the most vital out of the three, as it serves the most basic function of tracking the user's health status. In order to keep a close eye on the user's health the oximeter subsystem must be fully functioning and optimal. The readings taken from the oximeter must be accurate to ensure that false readings are not taken. The oximeter subsystem uses the MAX30100 as the blood pulse oxygen sensor; it is used to sense the blood oxygen level and pulse rate of the user, the information taken from the sensor will be fed to the Arduino Nano and relayed to an OLED screen which will then display the readings taken for the user to see.

The final session was used to set up the alerting subsystem and to debug the entire device as a whole working unit. The alert system consists of a GSM module, this module connects to a 2G SIM card and is able to send text messages to a pre-set contact. In order to determine the reliability of the module, we first sent a text message to a test phone while nearby the module, once we have determined that the module is able to consistently send messages, we increased the distance from the module to the test phone. This helps us to determine the range of functionality of our alert subsystem. An app was also created as a User-Interface, allowing our users to manually key in their location and to set an emergency contact. It is important to note that 911 does not take in calls from non-human parties.

Finally, after we determined the functionality of each subsystem, we soldered all the components onto the Arduino nano. We ran the electrical components as a whole unit multiple times to ensure that the device was consistent. Through this series of debugging we made sure that there are no potential critical failures in the device's operating system. During this same timeline, we had the case 3D-printed using the Makerspace facilities. Once the case was ready, we fitted the case in and tested the device a few times again for consistency's sake.



2 How the Prototype is Made

2.1 Mechanical

2.1.1 BOM (Bill of Materials)

Arduino Nano	
Component	Cost
Arduino Nano	$27.37
MAX30100 Pulse Oximeter Heart Rate Sensor	$11.50
SIM800L GPRS GSM Module MicroSIM card	$15.87
0.91 inch OLED Display Module (I2C &SSD1306;**1pc**)	$4.90
4.7k Ohm Resistor (**3pcs)**	$2.30
Total cost with 4.7k Ohm Resistor: **$61.94** Total cost without 4.7k Ohm Resistor: $59.64	

2.1.2 Equipment list

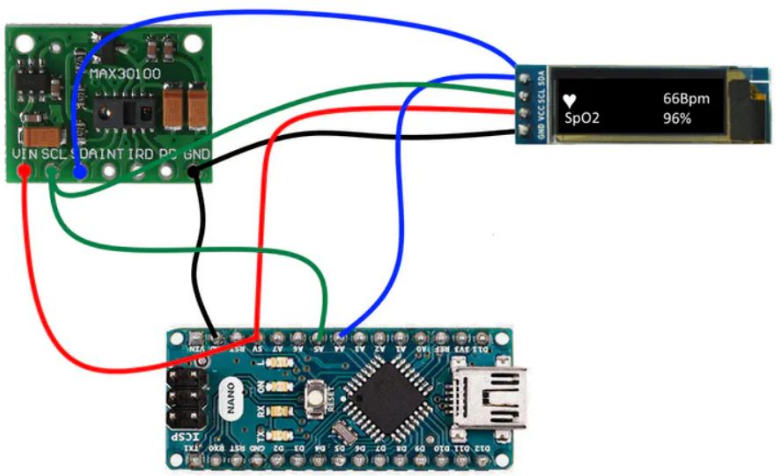
1. Arduino Nano Module
2. MAX 30100 Pulse Oximeter Heart Sensor
3. SIM800L GSM Module
4. 0.91 inch OLED DISPLAY

2.1.3 Instructions

2.1.3.1Oximeter subsystem

For the Oximeter subsystem, the MAX30100 Blood Oxygen and Pulse oximeter sensor is used. In order for the sensor to properly function, it is important to de-solder the three surface mounted resistors. Next, three 4.7 k Ohm resistors are required to be externally connected to the system. The reason is that the MAX30100 sensor chip is incompatible with the current output of the Arduino nano and therefore requires this adjustment for the sensor to properly function.

The following figure shows the wiring diagram for the oximeter subsystem.



(Image taken from https://www.hackster.io/umar-sear/arduino-heart-rate-monitor-a8e9e1)

Once the wiring is in place, plug the Arduino nano to the Arduino IDE console, and run the following code:

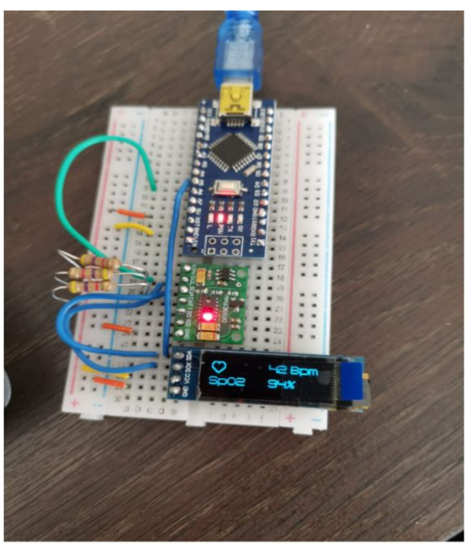
```
#include
"MAX30100_PulseOximeter.h"
#include <Wire.h>

#define REPORTING_PERIOD_MS     500

// PulseOximeter is the higher level interface to the sensor
// it offers:
//  * beat detection reporting
//  * heart rate calculation
//  * SpO2 (oxidation level) calculation
PulseOximeter pox;

const int numReadings=10;
float filterweight=0.5;
uint32_t tsLastReport = 0;
uint32_t last_beat=0;
int readIndex=0;
int average_beat=0;
int average_SpO2=0;
bool calculation_complete=false;
bool calculating=false;
bool initialized=false;
byte beat=0;

// Callback (registered below) fired when a pulse is detected
void onBeatDetected()
{
//  show_beat();
  last_beat=millis();
}

void display_calculating(int j)
{
  if (not calculating) {
      calculating=true;
    initialized=false;
  }
  Serial.print(". ");
}
void display_values()
```

```
{
  Serial.println("");
  Serial.print(average_beat);
  Serial.println(" Bpm");
  Serial.print("SpO2 ");
  Serial.print(average_SpO2);
  Serial.println("%");
}

void initial_display()
{
  if (not initialized)
  {
    Serial.print("Place finger on the sensor");
    initialized=true;
  }
}

void calculate_average(int beat, int SpO2)
{
  if (readIndex==numReadings) {
    calculation_complete=true;
    calculating=false;
    initialized=false;
    readIndex=0;
    display_values();
  }

  if (not calculation_complete and beat>30 and beat<220 and
SpO2>50) {
    average_beat = filterweight * (beat) + (1 - filterweight )
* average_beat;
    average_SpO2 = filterweight * (SpO2) + (1 - filterweight )
* average_SpO2;
    readIndex++;
    display_calculating(readIndex);
  }
}

void setup()
{
```

```
    Serial.begin(115200);
    pox.begin();
    pox.setOnBeatDetectedCallback(onBeatDetected);
    initial_display();
}

void loop()
{
    // Make sure to call update as fast as possible
    pox.update();
    if ((millis() - tsLastReport > REPORTING_PERIOD_MS) and
(not calculation_complete)) {
        calculate_average(pox.getHeartRate(),pox.getSpO2());
        tsLastReport = millis();
    }
    if ((millis()-last_beat>10000)) {
      calculation_complete=false;
      average_beat=0;
      average_SpO2=0;
      initial_display();
    }
}
```

(code taken from:https://github.com/umarsear/MAX30100-Heart-Beat-Monitor/blob/master/MAX30100_Heart_Beat_Serial.ino)

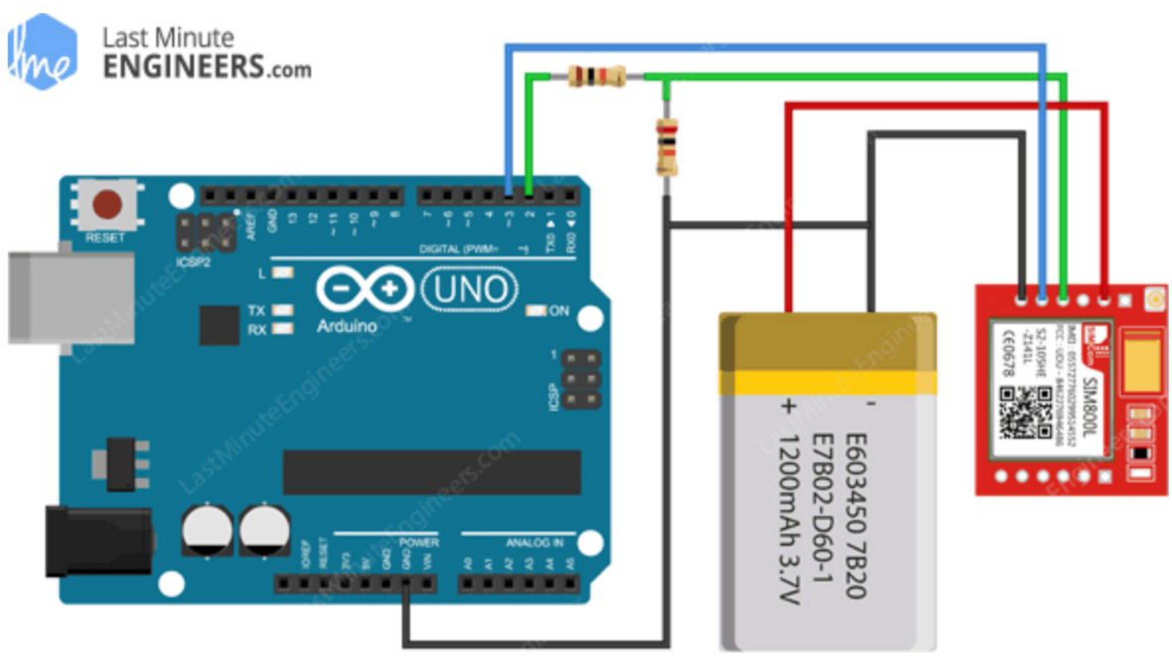
The device should function as follows:



2.1.3.2Alert subsystem

The Alert subsystem uses the GSM 800L module, it sends text messages to a contact that has been set in the system using a 2G MicroSIM card. The subsystem requires a battery to power the circuit as the Arduino nano's power output is not sufficient for the GSM 800L. The battery will also be required to power the final device as a whole so it is important to calculate battery requirements before purchasing.

The below figure shows the wiring for the subsystem:



Wiring SIM800L GSM GPRS Module with Arduino UNO

(Image taken from https://lastminuteengineers.com/sim800l-gsm-module-arduino-tutorial/)

Following the wiring, the code must be uploaded to run and test the subsystem. At this point, the GSM800L should light up. The code is as follows:

```
#include <SoftwareSerial.h>

//Create software serial object to communicate with SIM800L
SoftwareSerial mySerial(3, 2); //SIM800L Tx & Rx is connected to Arduino #3 & #2

void setup()
{
  //Begin serial communication with Arduino and Arduino IDE (Serial Monitor)
  Serial.begin(9600);

  //Begin serial communication with Arduino and SIM800L
  mySerial.begin(9600);

  Serial.println("Initializing...");
  delay(1000);

  mySerial.println("AT"); //Once the handshake test is successful, it will back to OK
  updateSerial();
  mySerial.println("AT+CSQ"); //Signal quality test, value range is 0-31 , 31 is the best
  updateSerial();
  mySerial.println("AT+CCID"); //Read SIM information to confirm whether the SIM is plugged
  updateSerial();
  mySerial.println("AT+CREG?"); //Check whether it has registered in the network
  updateSerial();
}

void loop()
{




  updateSerial();
}

void updateSerial()
{
  delay(500);
  while (Serial.available())
  {
    mySerial.write(Serial.read());//Forward what Serial received to Software Serial Port
  }
  while(mySerial.available())
  {
    Serial.write(mySerial.read());//Forward what Software Serial received to Serial Port
  }
}
```

(code taken from: https://lastminuteengineers.com/sim800l-gsm-module-arduino-tutorial/)

3 How to Use the Prototype

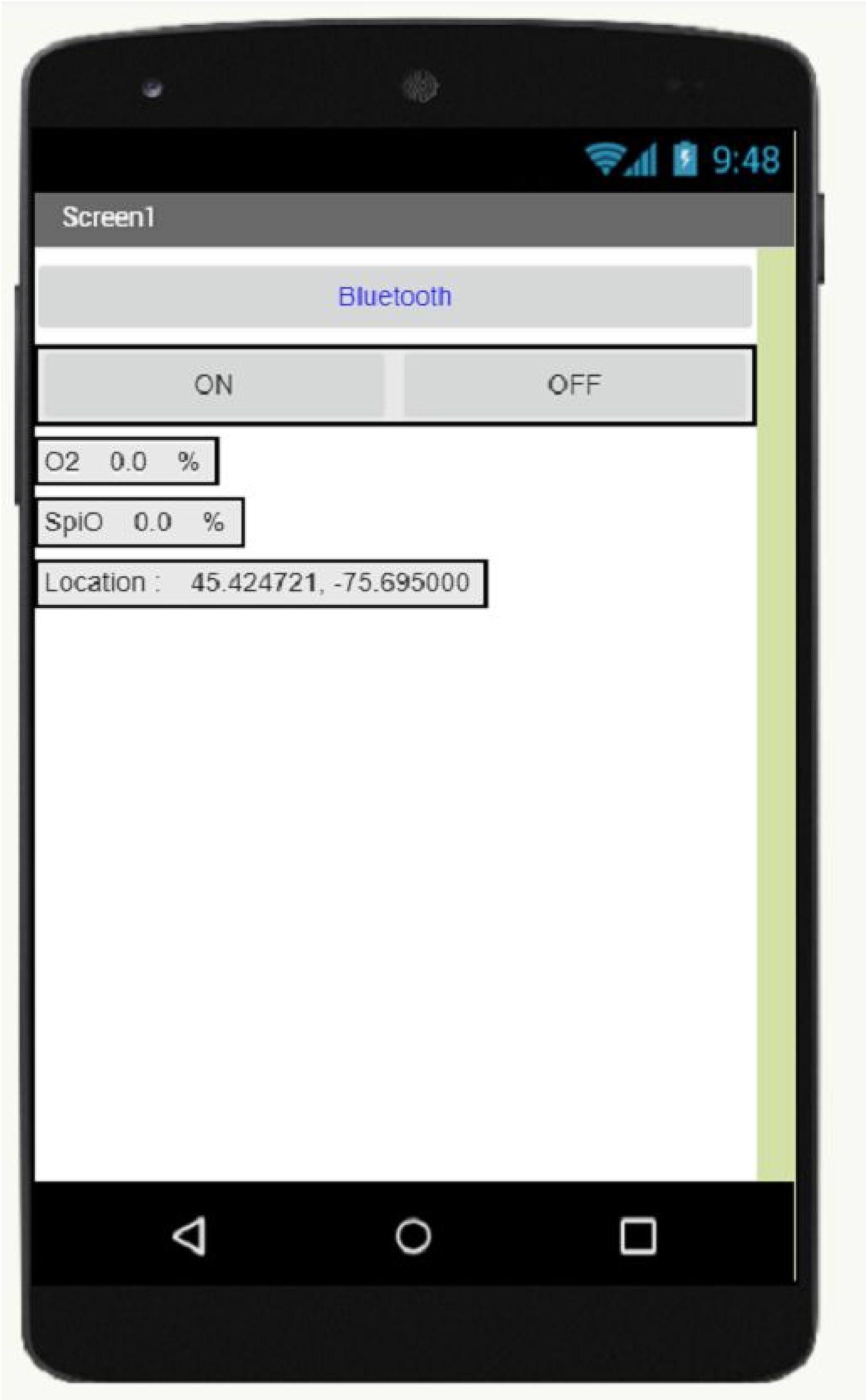
To use the product, the user puts their hand through the strap, much like wearing a watch or a bracelet. It is advisable to adjust the strap to ensure that the device fits snugly on the user's wrist. Turn the device on and wait for the OLED screen to light up, the screen should show that the device is reading the user's vitals.

The figure below shows the message showed on the OLED screen.

It is important to note that while the device is designed to stay on to the user, it is optimal for the user to minimize intense physical activity to avoid causing the device to slip off, which may cause the device to read nothing from the user.

To charge the device, any standard Android USB cable is sufficient. As there is no form of circuit breaking technology in the device, it is advisable that the user takes precaution when charging the device, this is to prevent the device from overheating.

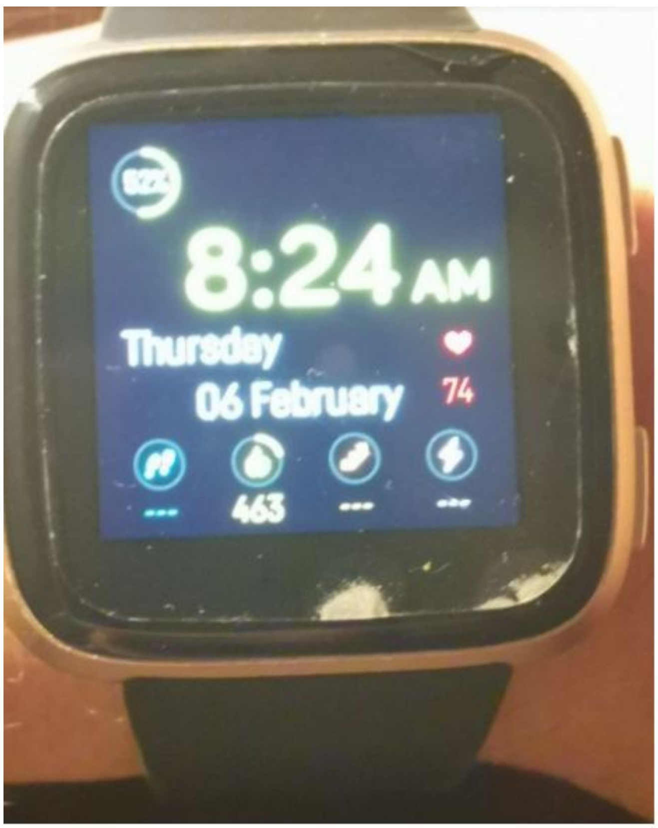
To enable the alert system of our device, the user would first need to connect their phone to our device with a phone application via Bluetooth. This can be done by clicking on the "ON" icon on the phone application. Once connected, the application will display the blood oxygen reading, respiratory rate, and the location of our user with coordinates. Once the vitals of our user reach a dangerously low level, it will trigger the phone application and an alert message would be sent to the emergency contact of our user. So that immediate medical help can be provided.

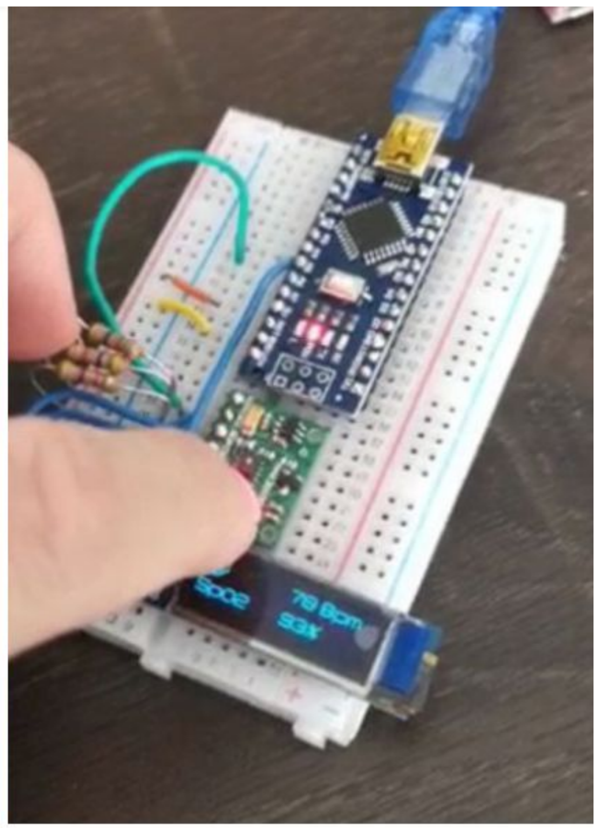

9:48
Screen1
Bluetooth
ON
OFF
O2 0.0 %
SpiO 0.0 %
Location : 45.424721, -75.695000

4 How to Maintain the Prototype

Maintenance is essential in order to keep the device in optimal condition. In order to ensure that the device is always functioning at its peak performance, the device has to be tested for consistency at least twice a month. As the device is a potential life saving device, it is important that we calibrate the device’s vital reading to an accurate degree.

We benchmarked our device’s reading to another commercial device, a Fitbit, as show below.





It is highly suggested that the MAX30100 module to be replaced with the GY MAX30100 module instead. This is because the MAX30100 module’s requirements to desolder the surface mounted resistors may very likely lead to damage to the module and this in turn could lead to a critical failure. The GY MAX30100 on the other hand is fully suited for the Arduino Nano, therefore uses the same wiring as the MAX30100 without the need to desolder anything, this would definitely make the device more accurate and reliable.

It is also recommended that the GSM 800L module is to be replace with a LTE Shield Module. As 2G SIM service is getting less common in Canada, the only known provider of the service is Rogers. Rogers stated that they will cease operations for their 2G services by the end of 2020, hence this would run the Alert system obsolete. A LTE Shield would be a fitting replacement as most people have smartphones and it is likely that LTE services are widely usable by most people. However, one thing to note is that the LTE Shield is much costlier than the GSM module, despite that the LTE Shield would be a long term investment as the LTE service is less likely to be taken down in the near future.

5 Conclusions et Recommendations for Future Work

In conclusion, we have had a fruitful time working on the device, we have managed to get a functioning final prototype within the time constraints. There are a few lessons that could be learnt to further improve on future work. Firstly, it is essential to do thorough research before executing any action. As there was a 100CAD budget, we had to be very careful of how often and how much money we were spending. Due to our poor pre-research, we had many instances where we were unable to get the parts we needed at the required time. Next is to plan a rough timeline, due to the nature of this device many parts are not easily obtainable in many stores. Hence, it is most probable that most of the shopping will be done on amazon. It is important to note that due to the nature of online shopping it is extremely hard to accurately gauge when a product will arrive, therefore giving an adequate amount of leeway time would be best. Lastly, it would be optimal to spend most of the time with hands-on work on the prototypes. Due to the time constraint of the project, it is not desirable to waste time off other tasks. In our case, we wasted too much time running debugs and trying to fix errors, the most common errors stem from poor soldering. All in all, it was a learning journey for all of us, starting from having minimal knowledge to acquiring a different skill sets in order to pull off this project. We are all very grateful to have been on such a learning curve.

6 Bibliography

Most information was taken from Tali's presentation, who took information from :

https://health-infobase.canada.ca/substance-related-harms

https://www.publichealthontario.ca/en/data-and-analysis/substanceuse/interactive-opioid-tool

https://www.ottawapublichealth.ca/en/public-health-topics/drugs.aspx

https://www.straight.com/news/1252756/without-harm-reduction-overdoseswould-have-killed-thousands-more-bc-analysis-finds

https://www.cihi.ca/en/opioids-in-canada/2018/opioid-related-harms-incanada/smaller-communities-feeling-impact-of-opioid-crisis-in-canada

https://www.theglobeandmail.com/canada/article-for-canadas-working-men-anopioid-crisis-and-a-blue-collar-code-of/

Information for the device making was taken from:

Heart Rate monitor:

https://www.hackster.io/umar-sear/arduino-heart-rate-monitor-a8e9e1

GSM Module:

https://lastminuteengineers.com/sim800l-gsm-module-arduino-tutorial/

APPENDICES

APPENDIX I: Design Files

Link to MakerRepo:

https://makerepo.com/iseow034/gng-1103-c4-opioid-overdose-device

APPENDIX II: Other Appendices