uOttawa

Faculté de génie
Faculty of Engineering

Project Deliverable F: **Prototype I and Customer Feedback**
GNG 1103 - Engineering Design

Overdose Detection Device

Submitted by

GNG1103, Section C01, Group 3
Brendan Pennington (300129445)
Tony Piro (300108404)
Tara Karner (300132379)
Heidi Camillia Zahiri (300135117)
Joe Wu (300152421)

March 8, 2020
University of Ottawa

Table of Contents

List of Tables

List of Figures

Introduction

In this deliverable, we will be going through the first prototype phase and feedback that was given during the recent client meeting. Along with this, we will be discussing our future plans for prototyping, and conduction tests to ensure that we are achieving our goals.

Firstly, we will discuss the first prototype to analyze the positive and negative aspects. This along with client feedback will allow us to develop the first prototype, and make possible improvements for the next prototype.

Following this, we will be discussing the changes that are going to be made to the sole in order for it to function optimally. It will be tested in many different ways with set objectives that are to be met during them. Another part of the prototype that will be worked on is the smartphone application, which will interact with the sole and be programmed using online software. We will discuss the details of our second prototype and the conclusions made from said prototype.

Finally, we will show some other university students from a variety of disciplines to get their opinion and get their suggestions for improvement. Using all of the feedback gathered as well as the information obtained from testing, we will create a more efficient prototype that will lead us to our final product.

Prototype I Results

Positive Outcomes:

- ➔ Able to get a tangible model from 3D printing in makerspace
- ➔ Enable us to plan out the locations of the components
- ➔ Allowed us to see if dimensions had to be reduced or expanded
- ➔ Research gave us a deeper understanding of how to begin prototype two

Negative Outcomes:

- ➔ The sole was very thick and may not be able to fit in a shoe while still being comfortable
- ➔ The sole was not flexible and had a very large curvature to it, which is unrealistic in comparison to many shoes
- ➔ Might not be able to take blood-oxygen levels through a thicker sock
- ➔ Flexibility is very low which can lead to discomfort or pain for the user

Improvements to be made:

- ➔ Smartphone App
  - ◆ We are continuing to try make the smartphone application that will allow the device to alert loved ones that an overdose is occuring
  - ◆ The bluetooth module inside of the sole will send a message to the app, which will then send a message to a smartphone of a chosen contact

Feedback from our customer (Tali Cahill):

- Has to measure blood oxygen levels even when walking/moving around
- Device has to be effective even when user has some socks on
- Device has to be well integrated into the user's daily life activities (ie. be comfortable, etc.)

Prototype II

The goal of our second prototype was to test out the functional aspect of our device. In prototype I, as mentioned before, the goal of our first prototype was to observe and draw conclusions from the physical and non functional aspect of our shoe insole. Now that we've obtained satisfactory feedback from our client, Tali Cahill, we have confirmation that our model is feasible and we now have to assess our microcontroller bluetooth system and our application that will allow us to monitor the blood oxygen levels through our phones.
Here is the list of tasks we followed for prototype II:

Table 1 - Assigned Tasks

Task description	**Person(s) responsible**	**Deadline**
Assemble Arduino components	Tara and Heidi	March 8th
Code Arduino IDE	Tara and Heidi	March 13th
Troubleshoot components (if needed)	Brendan	March 8th
Program Android App	Brendan	March 20th
Program Apple IOS App	Joe	March 20th
Create new improved 3D insole model	Tony	March 15th

We first had to figure out a way to connect our three main components (HC-05 Bluetooth module, MAX 30100 oximeter and Arduino Nano) with the help of a breadboard. To do so, we made the following connections:

- RXDm HC-05 to TX1 pin on Arduino
- TXD pin from HC-05 to RX0 pin on Arduino
- GND from Bluetooth to the GND (pin besides the RX0 pin) in the Arduino
- Vcc from Bluetooth to the 5V pin in the Arduino
- VIN from MAX30100 to the 5V pin in the Arduino
- SCL pin from MAX30100 to the A5 pin in the Arduino
- SDA pin from MAX30100 to the A4 pin in the Arduino
- INT pin from MAX30100 to the A2 pin in the Arduino
- GND pin from MAX30100 to the GND pin in the Arduino (pin between VIN and RST)

Figure 1- Arduino to pulse oximeter connections

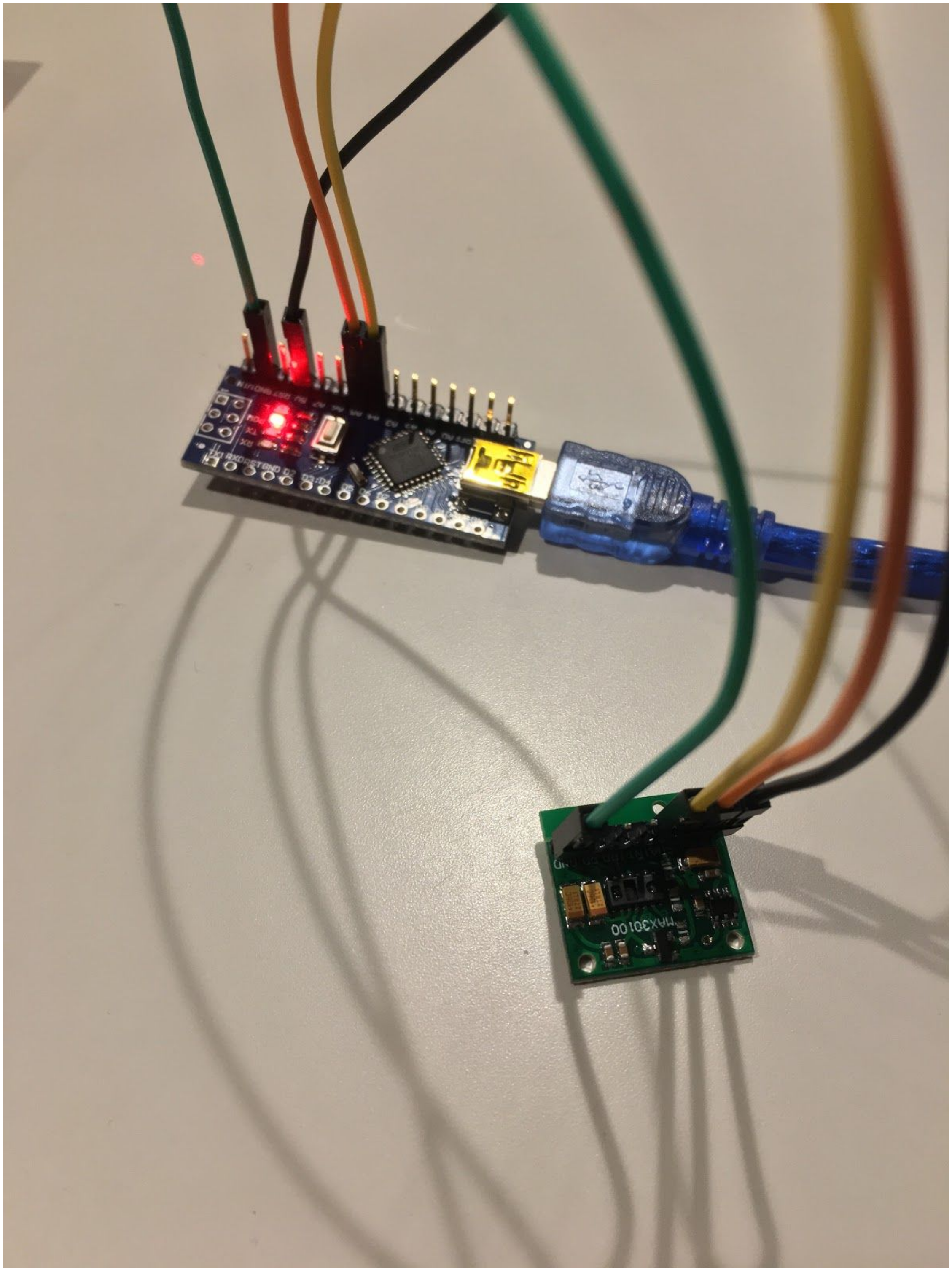


Figure 2- Arduino to bluetooth module connections

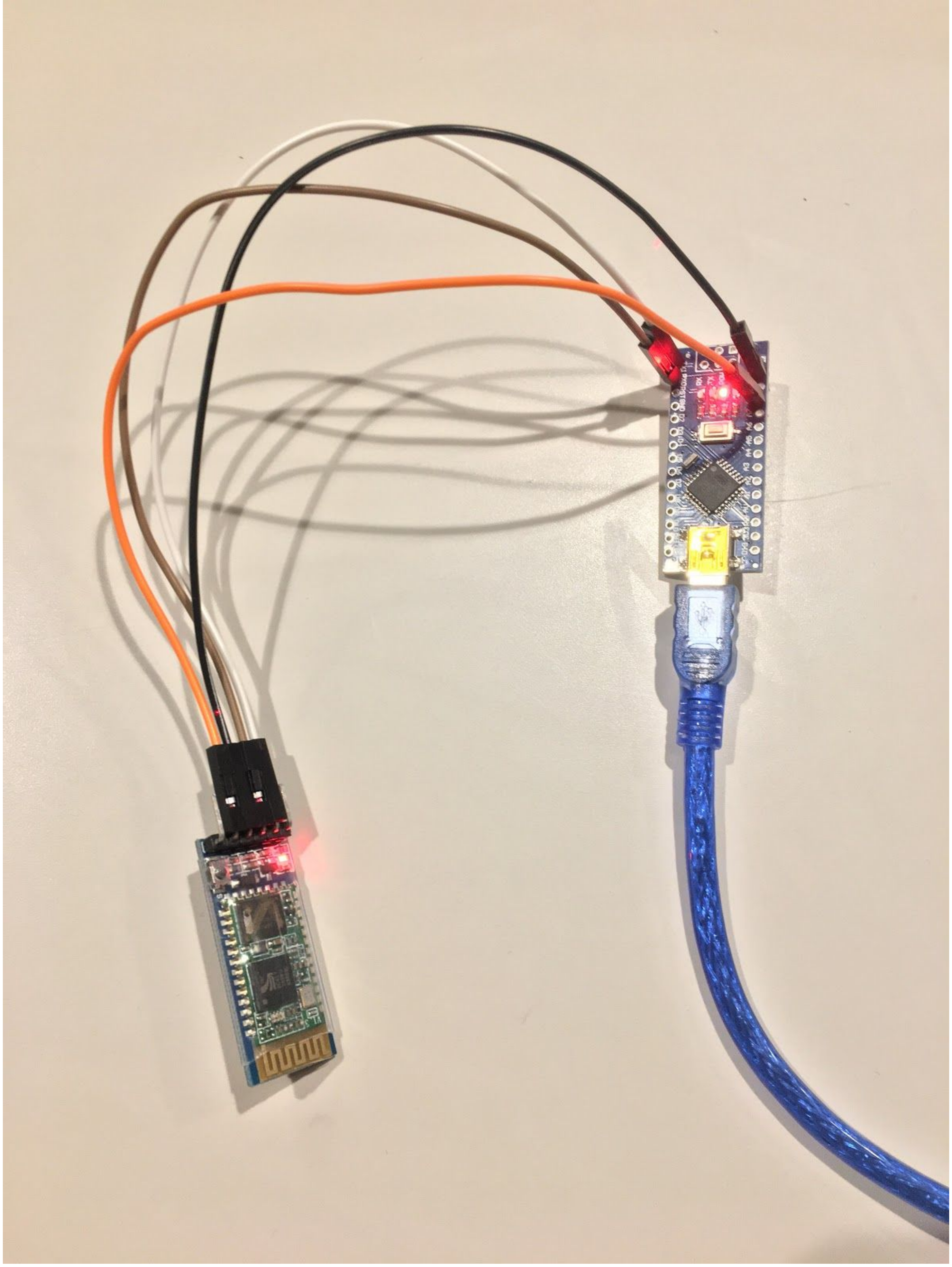
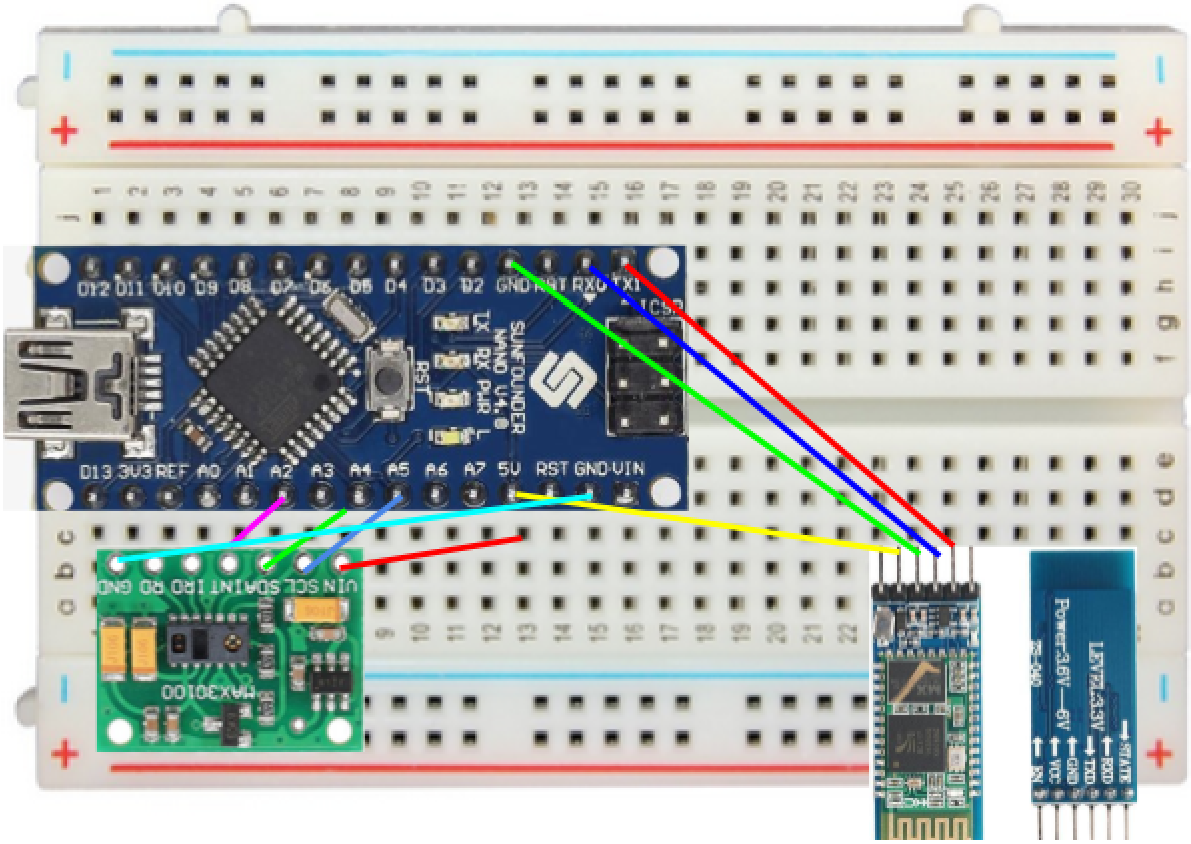
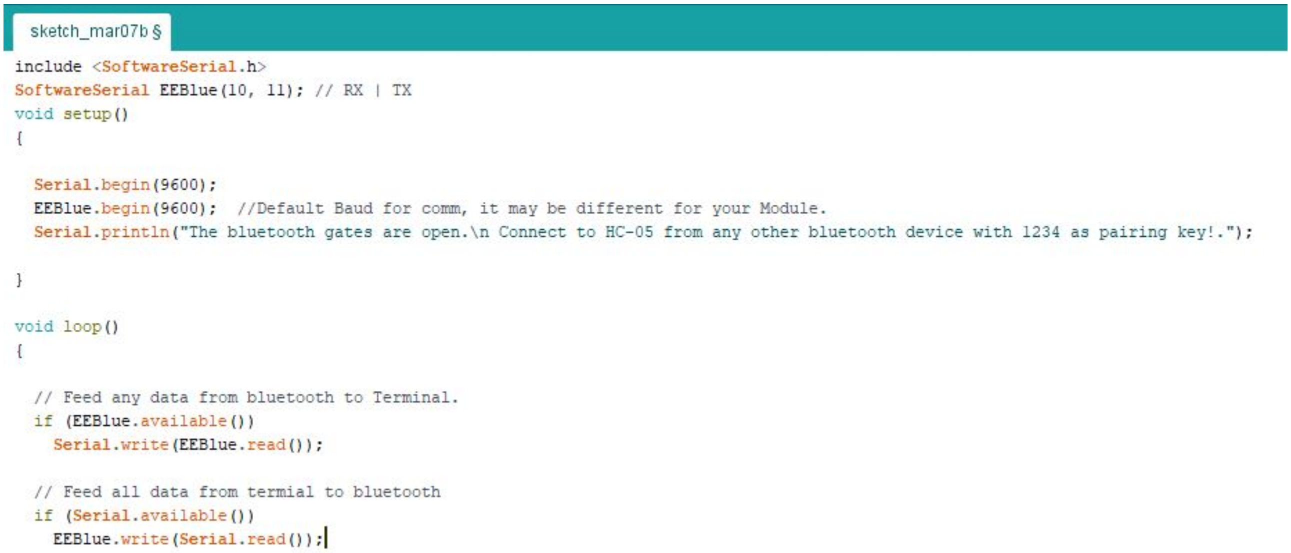


Figure 3- Circuit diagram of non-soldered components


D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 GND RST RX0 TX1
ICSP
SUNFOUNDER NANO V4.0
TX RX PWR L
RST
D13 3V3 REF A0 A1 A2 A3 A4 A5 A6 A7 5V RST GND VIN
GND RD IRD INT SDA SCL VIN
MAX30100
Power:3.6V—6V
LEVEL:3.3V
STATE
RXD
TXD
GND
VCC
EN

When assembling the parts together, we got the Bluetooth Module to successfully establish a connection with the Arduino. While sources from the Internet claimed that the HC-05 module was supposed to be blinking for two seconds at a time and then stay two seconds off, ours was blinking approximately at a 3 blink/second rate, which probably indicated a problem because it was impossible to upload the code that would allow the module to connect to a phone. We later figured out that we had to put the module into AT mode by clicking the button on the module, disconnect the 5V connection with the Arduino, all while keeping the HC-05 button pushed down, and then proceed to reconnect the 5V connection. After doing some testing, the Bluetooth module did in fact start blinking for two seconds at a time, significating that it was in AT mode and ready to be receptive to the code. We still have to test if the code will actually upload, since we are still early in our testing stages. The following code is what we're going to use to see if the Bluetooth module will connect to our phones.

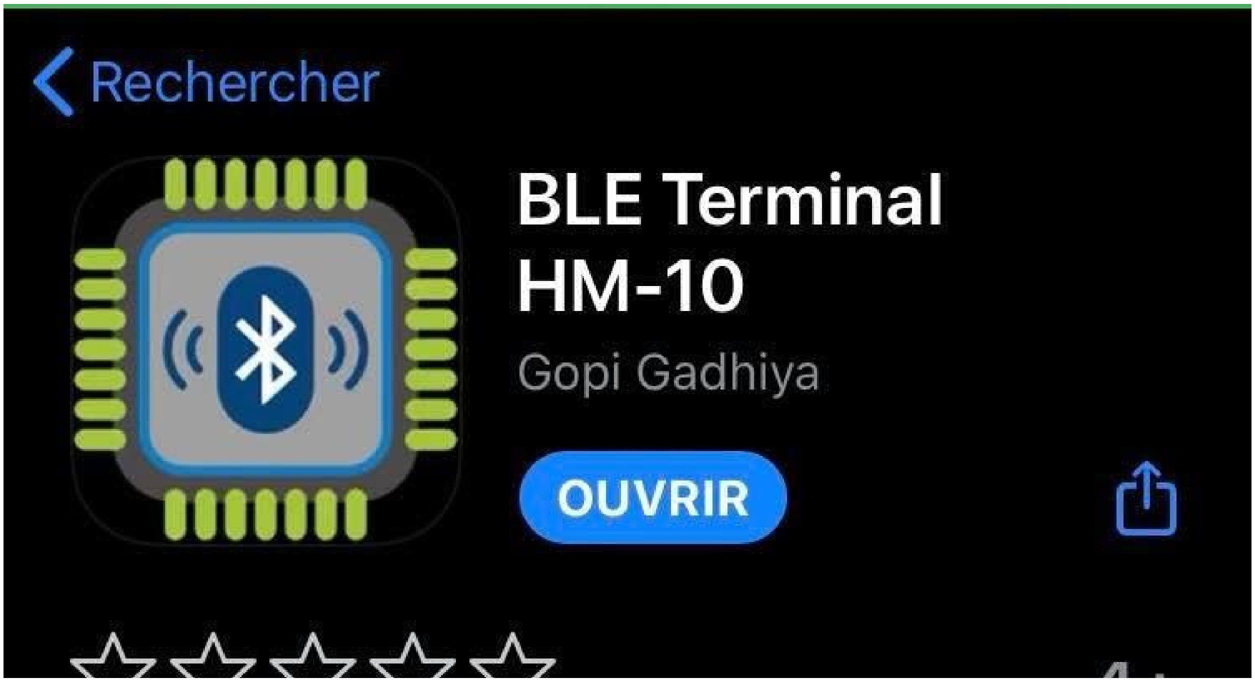
Figure 4- Arduino Code



```
include <SoftwareSerial.h>
SoftwareSerial EEBlue(10, 11); // RX | TX
void setup()
{

  Serial.begin(9600);
  EEBlue.begin(9600);  //Default Baud for comm, it may be different for your Module.
  Serial.println("The bluetooth gates are open.\n Connect to HC-05 from any other bluetooth device with 1234 as pairing key!.");

}

void loop()
{

  // Feed any data from bluetooth to Terminal.
  if (EEBlue.available())
    Serial.write(EEBlue.read());

  // Feed all data from termial to bluetooth
  if (Serial.available())
    EEBlue.write(Serial.read());
```

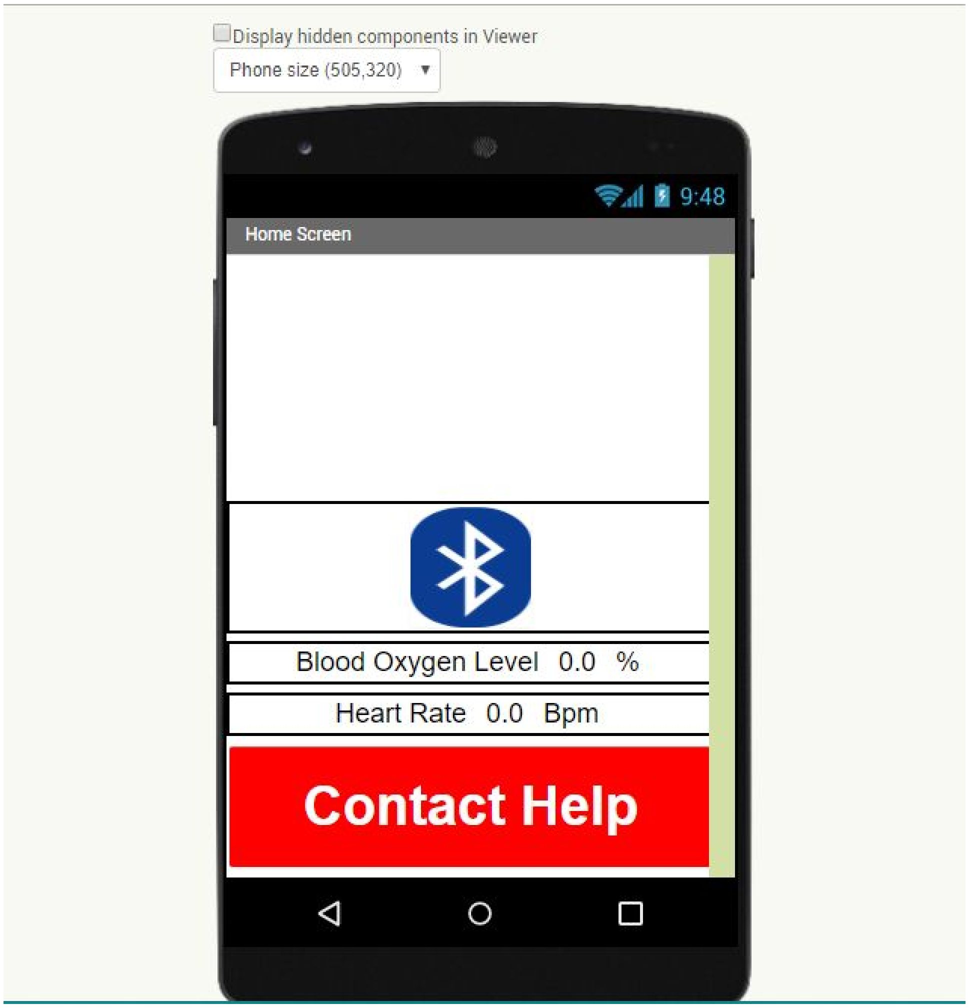
To connect the HC-05 bluetooth module to our phones, we will use a Bluetooth LE app, until the app we’re creating is ready. Any LE app would normally work but for testing we will use the following:

Figure 5- Bluetooth LE app



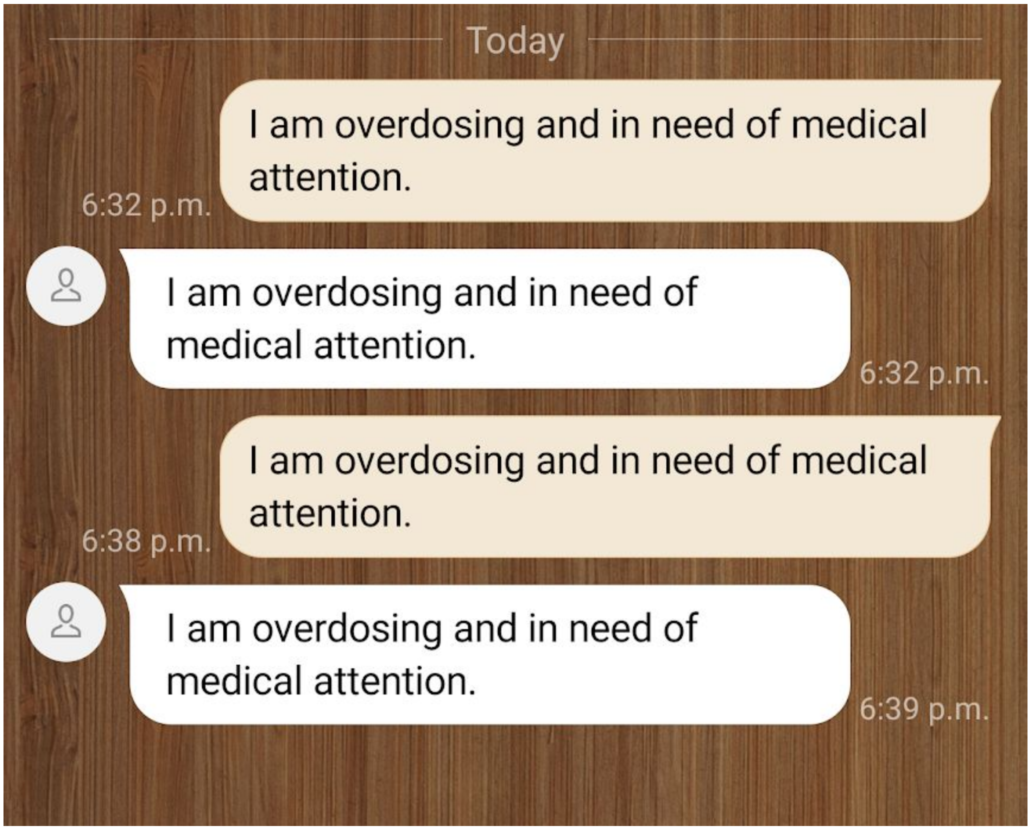
During this prototyping phase, we began working on the android app. We used MIT app inventor to make the app. For this phase our main priorities were: learning how the software operates, making a connection to the bluetooth module, the ability of contacting someone for help, and the creating a well organized user interface. We were able to have all four priorities completed during this testing phase.

Figure 6 - Prototype of the Android App



As of now, the app is able to communicate with the bluetooth module, and has the ability of sending premade texts to the user’s emergency contacts. We also have the feature of texting as many numbers at once as the user pleases. Once they input the phone numbers, all of them will be contacted when help is requested.

Figure 7 - Display of the message sent to the emergency contacts



Today

I am overdosing and in need of medical attention.
6:32 p.m.

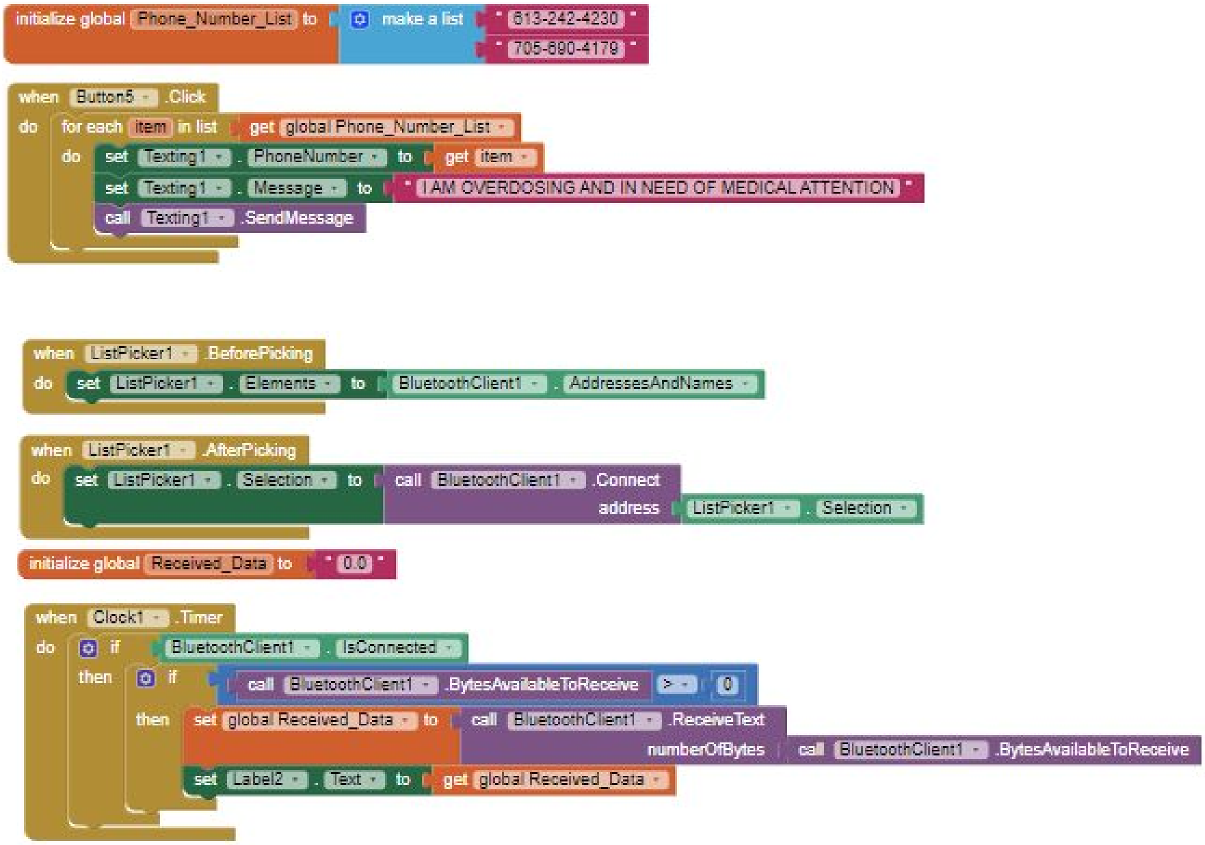
I am overdosing and in need of medical attention.
6:32 p.m.

I am overdosing and in need of medical attention.
6:38 p.m.

I am overdosing and in need of medical attention.
6:39 p.m.

For the future versions of the app, we are looking to upload live data from the device and display it on the user's phone. That way, even if the user is not currently overdosing, they can still know what state of health they are in. We are also working on incorporating the user's location into the app. This way the user can share their location with their emergency contacts, making it easier to get the user help as fast as possible.

Figure 8 - Block Coding for the Android App



```
initialize global Phone_Number_List to make a list " 613-242-4230 "
                                                   " 705-690-4179 "

when Button5 .Click
do  for each item in list get global Phone_Number_List
    do  set Texting1 . PhoneNumber to get item
        set Texting1 . Message to " I AM OVERDOSING AND IN NEED OF MEDICAL ATTENTION "
        call Texting1 .SendMessage

when ListPicker1 .BeforePicking
do  set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames

when ListPicker1 .AfterPicking
do  set ListPicker1 . Selection to call BluetoothClient1 .Connect
                                                     address ListPicker1 . Selection

initialize global Received_Data to " 0.0 "

when Clock1 .Timer
do  if BluetoothClient1 . IsConnected
    then  if call BluetoothClient1 .BytesAvailableToReceive > 0
          then  set global Received_Data to call BluetoothClient1 .ReceiveText
                                                  numberOfBytes call BluetoothClient1 .BytesAvailableToReceive
                set Label2 . Text to get global Received_Data
```

During this prototyping phase we created another 3D model of the sole in solidworks. Changes to the design are continuing to be made to ensure that the design specifications are met. The dimensions of the components are currently being measured and incorporated into the solidworks design, along with access room for wiring. This new model will have a bottom layer consisting of all of the hole and spacing for the components, as well as a top layer that will be much thinner and will sit on top of the main frame. This top layer will be able to be removed at any time as it will be attached through a push tab, which will be secured in place during use.

Testing of Prototype

In order to test our product, we will be putting the new model inside of a shoe and allowing one of the group members to wear it for a day. After this day of use, we will get their feedback on how comfortable the product was, and the improvements that need to be made. We will then get another group member to wear the insole for a day and again, get their feedback on the comfort of the shoe. After two tests by group members we will be finished with the physical testing.

The next portion of testing will be with all of the electronic components, in that we will be testing them to ensure they work, and have fast response times with accurate readings. To do this we will test the components in a variety of different temperature settings to ensure that it is able to function in all conditions. Once we have tested the components in these conditions our tests will be complete.

Upon completion of testing, we will be using the result to change the design in order to make an ideal final product. Of the previously set goals for testing, we will see if the goals are met and the ones that weren't in order to stay on track in the design process.

Figure 9- Top View of SolidWorks designed sole

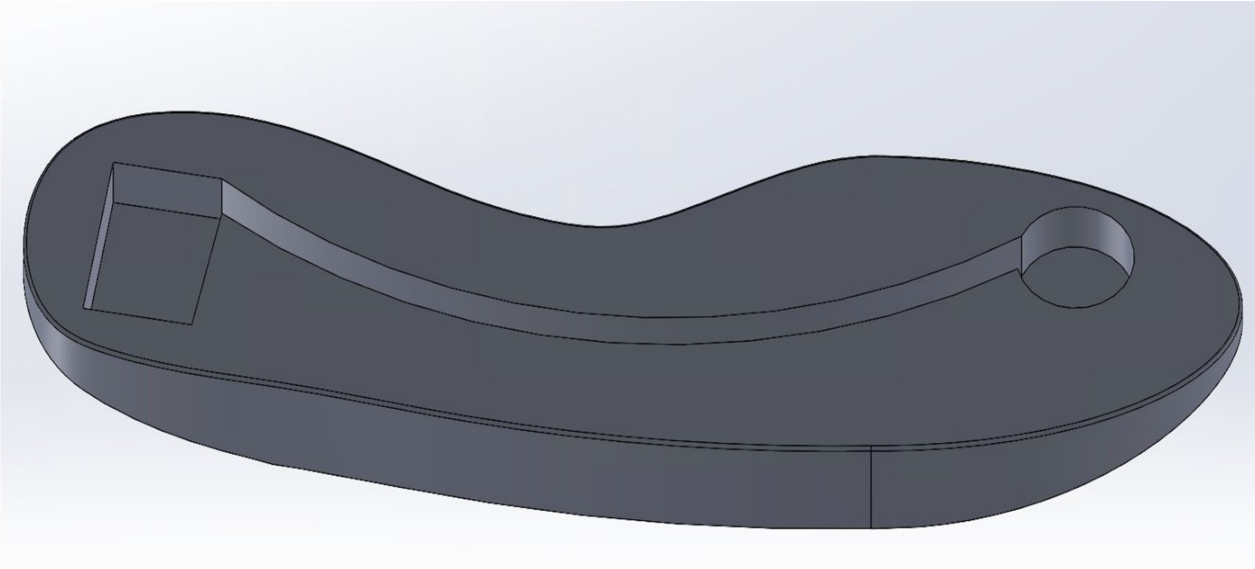
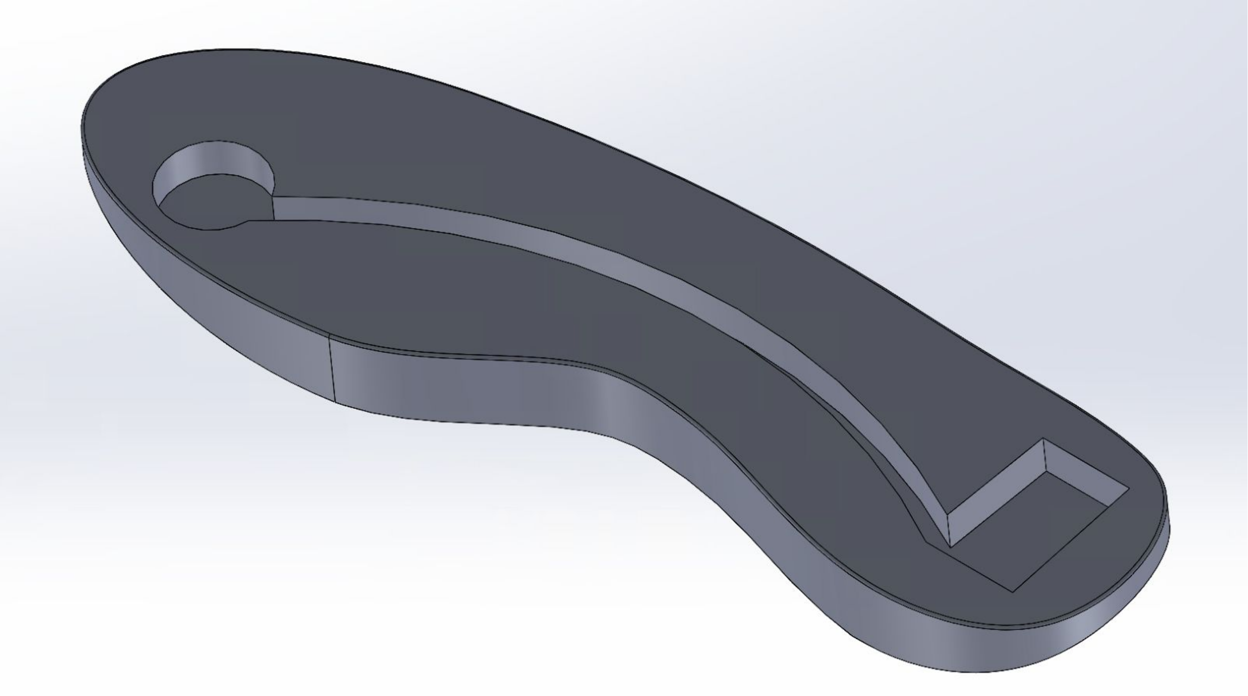


Figure 10- Isometric View of SolidWorks designed sole

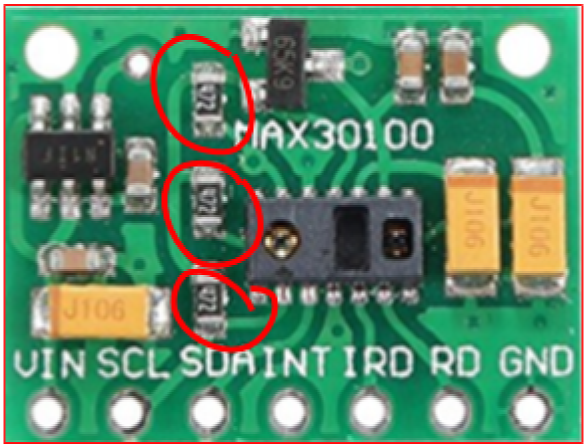


Risk and contingency update

Unfortunately, after testing out our pulse oximeter device and not being able to get it to work (ie. LED significating that the oximeter is on does not turn on), we found out that the MAX 30100 module has a design problem that leads it to not work with higher logic level microcontrollers like the Arduino Nano we are using for this project.

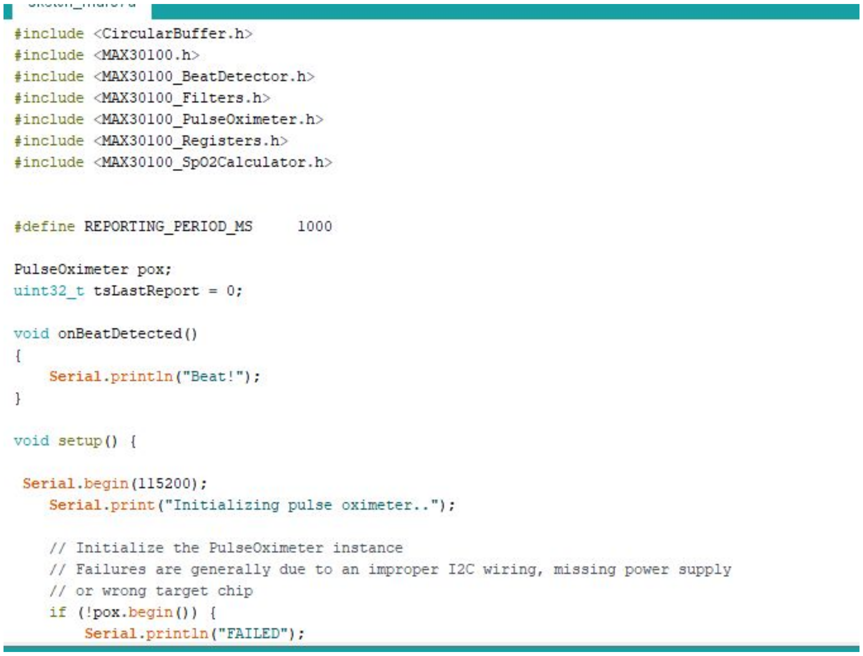
After doing some research on what can be done to fix this issue, we found out that maybe the oximeter would work if we removed the integrated resistors on the module. Unfortunately, we tried that and it was not successful. We have another copy of the MAX 30100 coming in the mail (should be here monday March 9th). Hopefully, this new oximeter will work. If it doesn't, we will have to do some more research into possible solutions that are feasible in the context of this project. We want to be the more proactive possible here, so we will actively look into ways to successfully integrate the MAX 30100 module into our circuit.

Figure 11 - Removed Resistors on MAX 30100



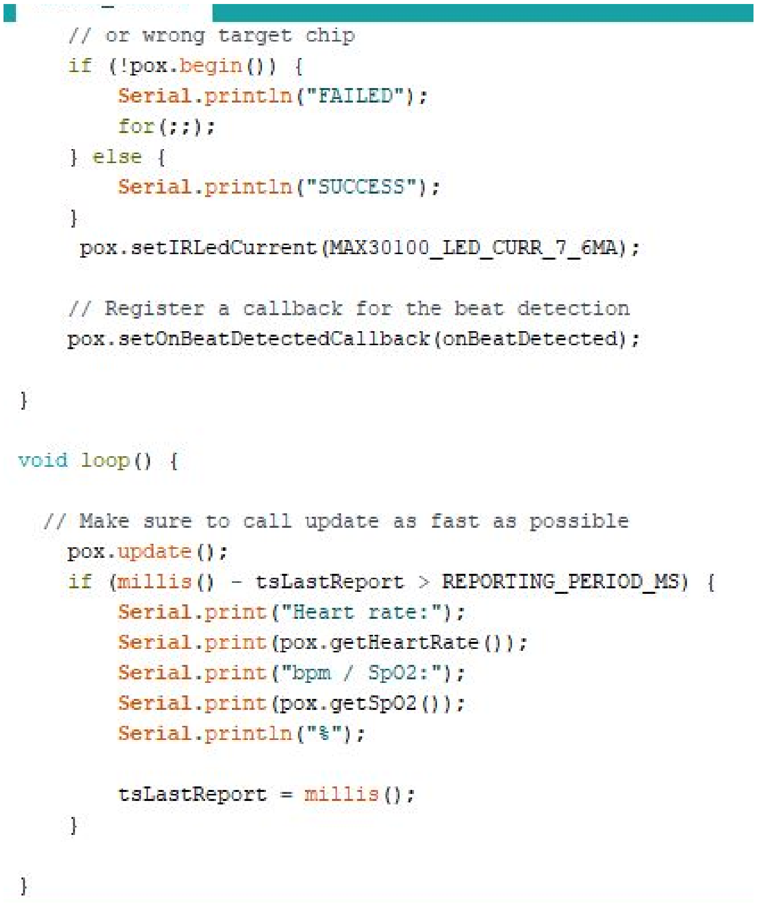
In the meantime, we have the code for the oximeter ready:

Figure 12 - Arduino Code for MAX 30100



```
#include <CircularBuffer.h>
#include <MAX30100.h>
#include <MAX30100_BeatDetector.h>
#include <MAX30100_Filters.h>
#include <MAX30100_PulseOximeter.h>
#include <MAX30100_Registers.h>
#include <MAX30100_SpO2Calculator.h>


#define REPORTING_PERIOD_MS     1000

PulseOximeter pox;
uint32_t tsLastReport = 0;

void onBeatDetected()
{
    Serial.println("Beat!");
}

void setup() {

 Serial.begin(115200);
    Serial.print("Initializing pulse oximeter..");

    // Initialize the PulseOximeter instance
    // Failures are generally due to an improper I2C wiring, missing power supply
    // or wrong target chip
    if (!pox.begin()) {
        Serial.println("FAILED");
```

Figure 13 - Arduino Code for MAX 30100



```
    // or wrong target chip
    if (!pox.begin()) {
        Serial.println("FAILED");
        for(;;);
    } else {
        Serial.println("SUCCESS");
    }
     pox.setIRLedCurrent(MAX30100_LED_CURR_7_6MA);

    // Register a callback for the beat detection
    pox.setOnBeatDetectedCallback(onBeatDetected);

}

void loop() {

  // Make sure to call update as fast as possible
    pox.update();
    if (millis() - tsLastReport > REPORTING_PERIOD_MS) {
        Serial.print("Heart rate:");
        Serial.print(pox.getHeartRate());
        Serial.print("bpm / SpO2:");
        Serial.print(pox.getSpO2());
        Serial.println("%");

        tsLastReport = millis();
    }

}
```

Conclusion

To conclude, we have made significant progress since our first prototype. The most significant aspect of this device is the software/micro controlling part of it, and after a week of experimenting with our electronic parts, we feel like we've made a lot of progress, even though our testing was not 100% successful.

We do not have much time left before Design Day and the official presentation of our third and final prototype. Our goals from here on until then is to be proactive and effectively apply our contingency plan if needed. We will work on bringing all of our individual/semi individual work together to merge the physical and non functional aspect of our device as well as the functional but non physical aspect of it.

Joe Wu did not contribute to this deliverable and hasn't on the past two.