Life-360 (Medical Device)

B.E.Sc Electrical Engineering

ECE 2242 Electrical Laboratory

Husein Mokbel and Basheer Ghebeh

Professor Marinov

2022 04 01

Declaration Statement

I, Husein Mokbel, claim that all the work present in this report is my own, except where specifically referenced.

I, Basheer Ghebeh, claim that all the work present in this report is my own, except where specifically referenced.

Abstract

The goal was to create a convenient way at home for people who are sick to evaluate their health on a general scale. The central component of this project, or rather device, is the MAX-30100 itself. This is the case because this device alone offers the main purpose of the project, which is to measure one's heart rate and internal blood oxygen levels. The measurement of one's internal body temperature is done by another sensor, the LM-35. The overall system eventually progressed to a more complex system. Such a system was established through the integration of a bluetooth into our project. The purpose of the project was no longer based upon the simple act of self-assessment, but rather the monitoring of one's health done by another individual. Such an advancement opens doors for the device to become more applicable. The outcome of the work was that it was partially functional. The first few initial stages of the project, upon multiple debugging activities, were successful. Such successful stages were the Tinker cad simulations, Breadboard builds, and PCB design. The final soldered build had the central components operating, however, there were a few dysfunctional parts. Overall, from a learning standpoint, the project was a success despite the few faults at the end.

Acknowledgement

I would like to thank my parents for motivating and supporting my decision from the start to pursue engineering. I would also like to thank the Professor for being incredibly patient and helpful throughout the entire process of our project. Additionally, I want to thank the Teaching Assistants for being very generous, helpful, and insightful at every step of this project.

Table of Contents

Introduction

The building and processes of a project are crucial for one to become very familiar within the domain of engineering. Thus, such familiarity with such concepts is highly important. The purpose of our project is to create an “at-home” medical device that would monitor one’s health. The goal here is to effectively remove the need of visiting a Doctor in person, as such a process is now unacceptable due to COVID-19 and other variants. The objectives of this project were established on a step-by-step basis. Initially, upon defining a problem and thus potential solutions, we narrowed down our options to the most viable and optimal solution. From thereon, we had to determine what actuators (Buzzer, LCD, Bluetooth), and sensors (LM-35, MAX-30100, and Button) to use. Secondly, such components were simulated on tinker-cad separately (for the sake of understanding code) and eventually the components were collectively simulated. Third, the components were built on the breadboard, and eventually the same build was replicated with the PCB.

Customer Needs Analysis And Business Opportunity

Alternative Ideas	Complexity	Cost	Functionality	Market Appeal	Engagment	Convenience	**Total**
Blood Pressure Sensor	3	2	1	2	3	2	13
Vitals Sensor w/ Personalized Report	2	1	1	3	1	3	11
On-the-Go Vitals Sensor	1	1	1	2	1	1	7

Table 1: Evaluating the various solutions on a point system

In table 1, the three solutions were evaluated and ranked based on their complexity, build, cost, functionality, market appeal, engagement, and convenience. Of course, some solutions were ranked higher depending on the category of focus; however, the goal here is to determine what solution is the most net effective. The business opportunity here that can be sought after by others who plan to continue this project is the pursuit of a watch device. In other words, the device will no longer be subject to one's room, and therefore portable. Moreover, the unique aspect of this device, from a market perspective, is that it is relatively cheap to construct.

Literature Review & Description

Our project assesses the problem because the assessment and monitoring of one's health could be accomplished at home. The goal is to not only generally assess one's health, but to also allow someone else to monitor the individual's health (Bluetooth). Our group is clearly aware that other similar designs with similar intentions exist [1]. For example, there are watches on the market that can essentially not only tell time, but also monitor and graph one's vital measurements such as heart-rate, blood pressure, blood oxygen levels and body temperature. Such evidence for other similar work was presented to us by our Professor, as he owned such a watch. A crucial component of this project is the MAX-30100. This device, which operates within 1.3 volts and 3.3 volts, measures one's blood oxygen levels and heart-rate . Already, one can see the similarities between this device and our Professor's watch. The way this operates is through the projection of red and infrared wavelengths (from a photodiode) in which the oxygenated and deoxygenated intercepts each respective wave. The rest of the wavelengths are reflected onto a receiver, in which it is then that the data is outputted [2]. The LM-35 is a simple temperature sensor in which it has a direct relationship with voltage (voltage of diode). In other words, as temperature increases, the voltage across the diode increases correspondingly with the appropriate ratio to output the temperature measurement [3]. The LCD, although originally operating with 16 pins, was simplified and soldered to operate using just 4 pins. Moreover, the buzzer was intended to be dependent on the button once pressed. Additionally, the function of the button is to reset the LCD screen to allow for new measurements to occur, and the buzzer sound was implemented to notify the user that the reset has occurred; thus the measuring of vitals can take place.

Problem Definition/Opportunity

The problem we have defined originates from a personal problem that one of our group members experienced, and from thereon we noticed that this is surely a problem that the general public faces as well. Due to uncertain times (COVID-19 and other variants), the ability to go get a general health assessment from a family doctor has been hindered. This is because family doctors, especially during peak virus outbreaks, do not allow for patients to come in for general assessments. Or, perhaps in other cases, in-person health assessments (in-person appointments) of individuals are prolonged. Thus, the solution to create an “at-home medical device” to generally assess one’s health avoids such a problem to some extent.

Design Constraints & Standards

Attributes & Constraint	Importance / Implications
Location of medical device	The goal was to create a device in which the user could use conveniently, not limited by a particular area (example: Bedroom)
Bluetooth (connection can reach 30 ft, 10m)	The monitoring of one's health by another individual (from a phone) can only be done if the person is within an appropriate range.
Size of the project (6.9cm x 5.3cm)	Relative to other devices, like watches that obtain the ability to measure your blood pressure, heart rate, blood oxygen levels, etc., the size of the Arduino makes our device difficult to operate in a convenient manner on someone's body.
Power limitation (5 volts required as input voltage)	The project itself is constrained by the need of a power supply. This limitation is clearly defined when considering other medical devices that are not constantly in need of a voltage external input (watch).
Time	The project was started in mid to late February and due at the end of March. More time would have allowed us to execute our project to its fullest potential and look for other reasonable additions to our solution.

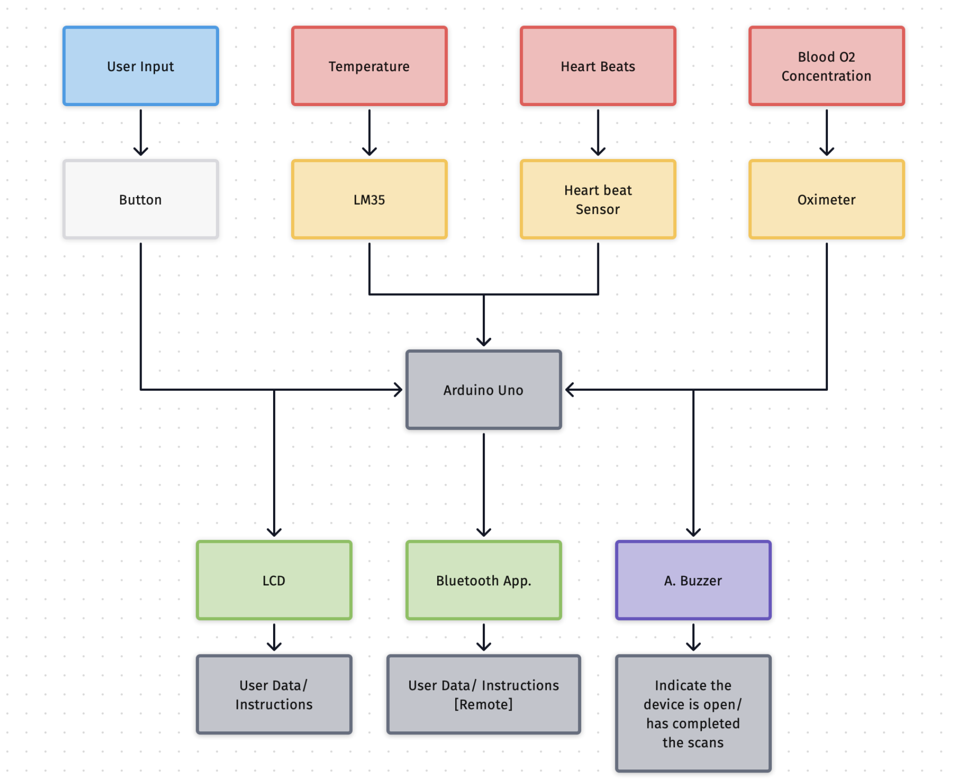
During the early stages of the project we needed to set clear, precise goals and constraints to ensure we knew our boundaries. This avoids setting unrealistic goals or having the project be too expensive: for us and the consumer. The time constraint was the most disciplining since it set us up for success. Allowing us to finish up all our projects before the deadline, otherwise we would have not been as productive.

Calculations

Component	Cost ($)
LM35	$2.12
MAX30100	$6.67
Button	$0.25
LCD	$9.93
Buzzer	$1.55
Bluetooth	$5.89
Arduino UNO R3	$21.44
Total	**$47.85**

Design

Block Diagram



After booting up the device, a loading screen will show up on the LCD, followed by the measurements of the LM35, heartbeat sensor and Oximeter (MAX30100).

Input would then be given by the user in the form of a button. The Arduino UNO would then inform the buzzer to make a high frequency sound: resetting the LCD and Bluetooth terminal.

TinkerCAD

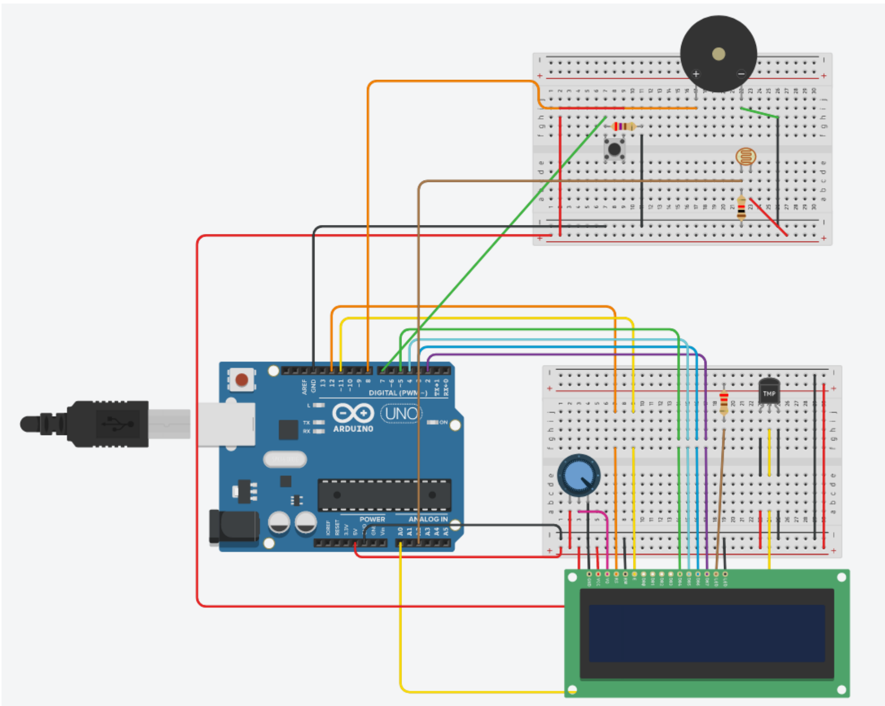


Figure 1: shows the tinkerCAD simulation with the 16-pin LCD, buzzer, button and temperature sensor.

The other components were not found and could not be emulated so we decided it would be best to simulate the rest of the blocks instead. TinkerCAD also does not have the I2C LCD so we had to resort to using the 16-pin LCD, which uses different code. This affected our working pace (Figure 1).

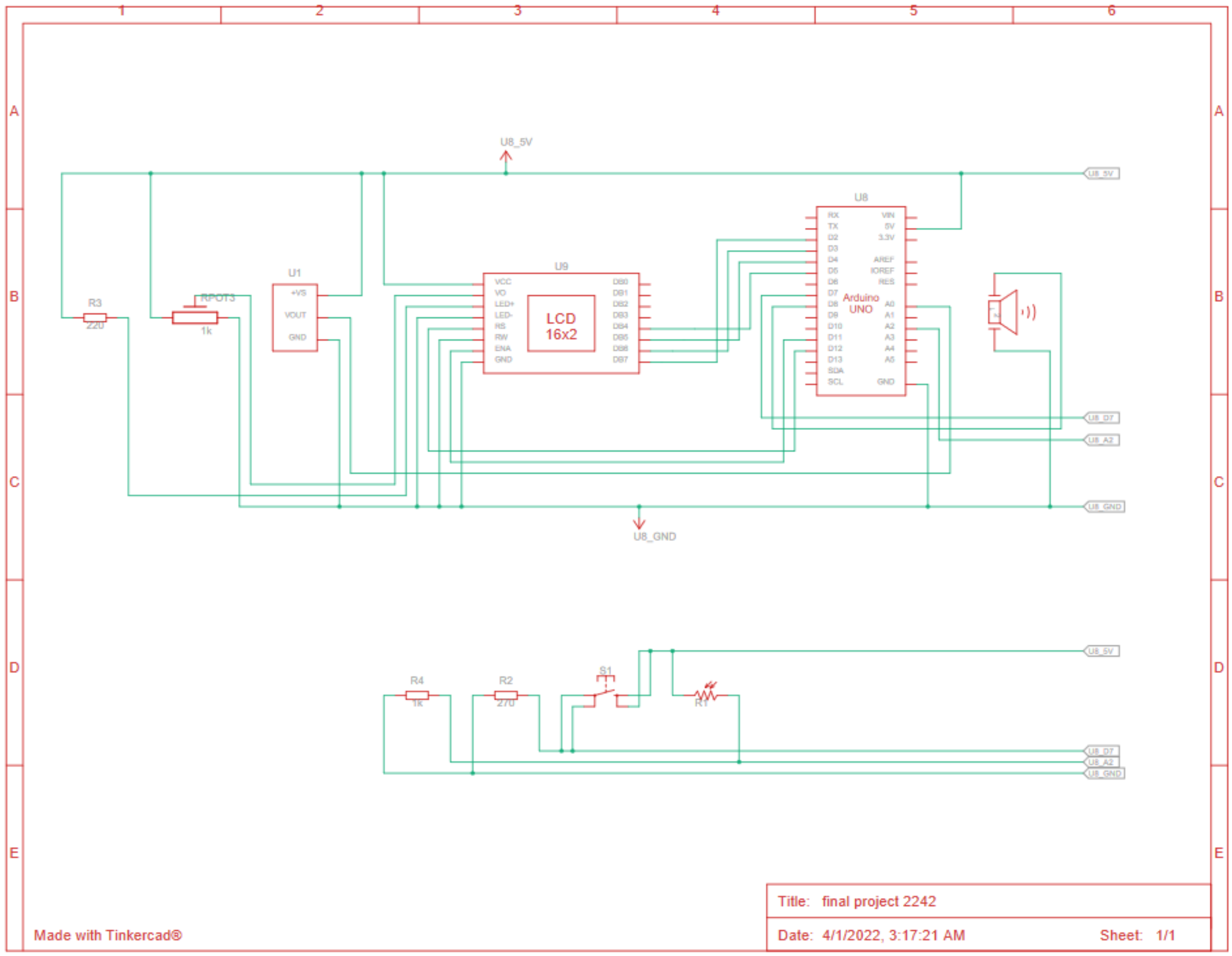


Figure 2: shows the schematic done by TinkerCAD.

The TinkerCAD schematic is quite different from what was done on EAGLE. It missed quite a few components and added a few more. The LCD for example took a large sum of the schematic when in reality, the LCD is only connected to 4 pins. The MAX30100 was not part of this schematic (Figure 2).

EAGLE

EAGLE Schematic

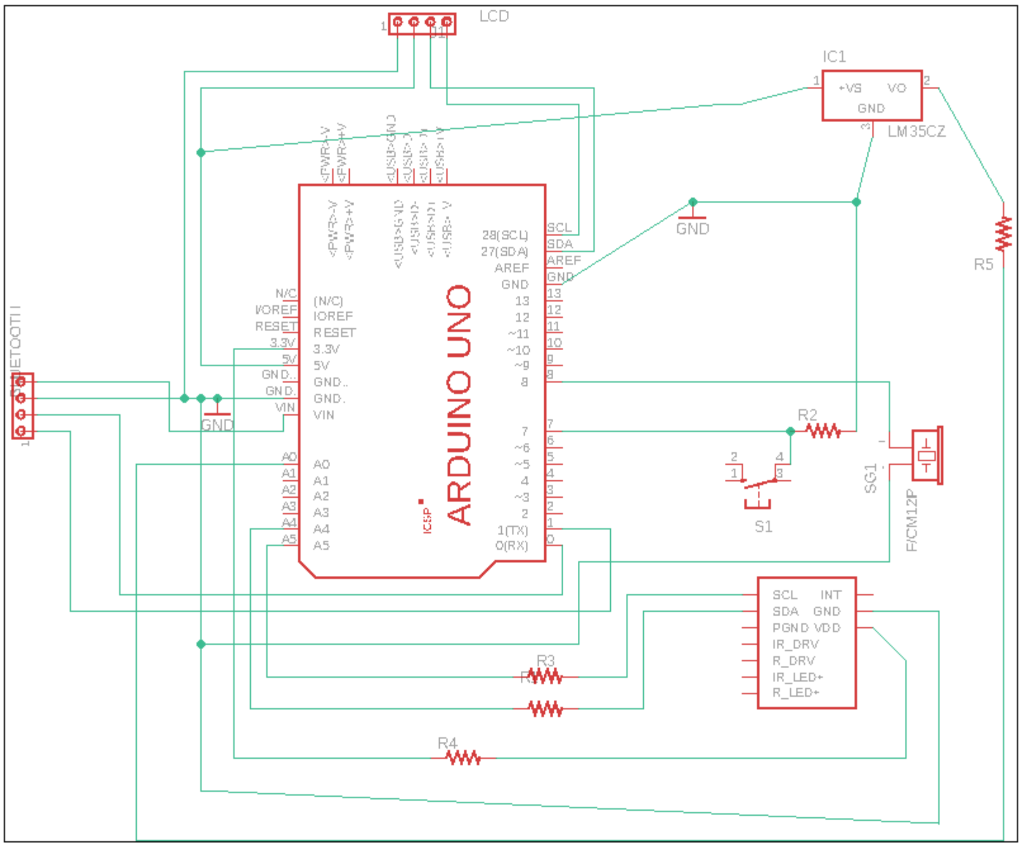


Figure 3: This is our first attempt at creating the Life360: the prototype.

The prototype schematic was inelegant and it did not translate well with the PCB. in addition, The air wires were virtually impossible to connect. This version also required more jumper resistors. It was inelegant and it did not translate well with the PCB (Figure 3).

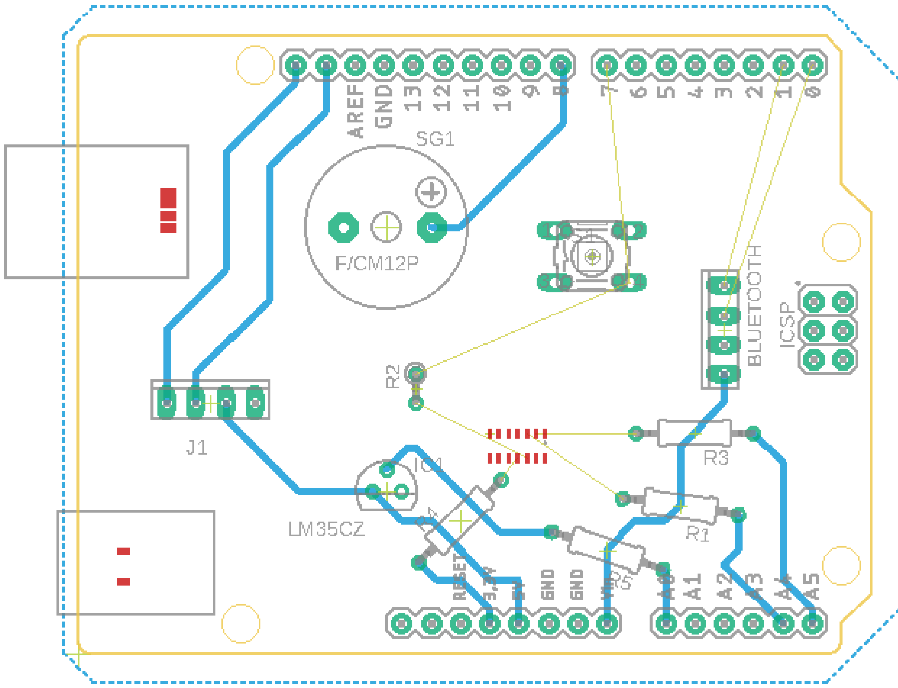


Figure 4: PCB of the prototype.

The airwires were not being connected. It was either an issue of EAGLE or some connection. The MAX30100 was also integrated from a library which did not work. We had to use pin connectors to allow the MAX30100 to properly fit on the PCB. This further ruined the airwire situation: requiring us to use more jumper resistors (Figure 4).

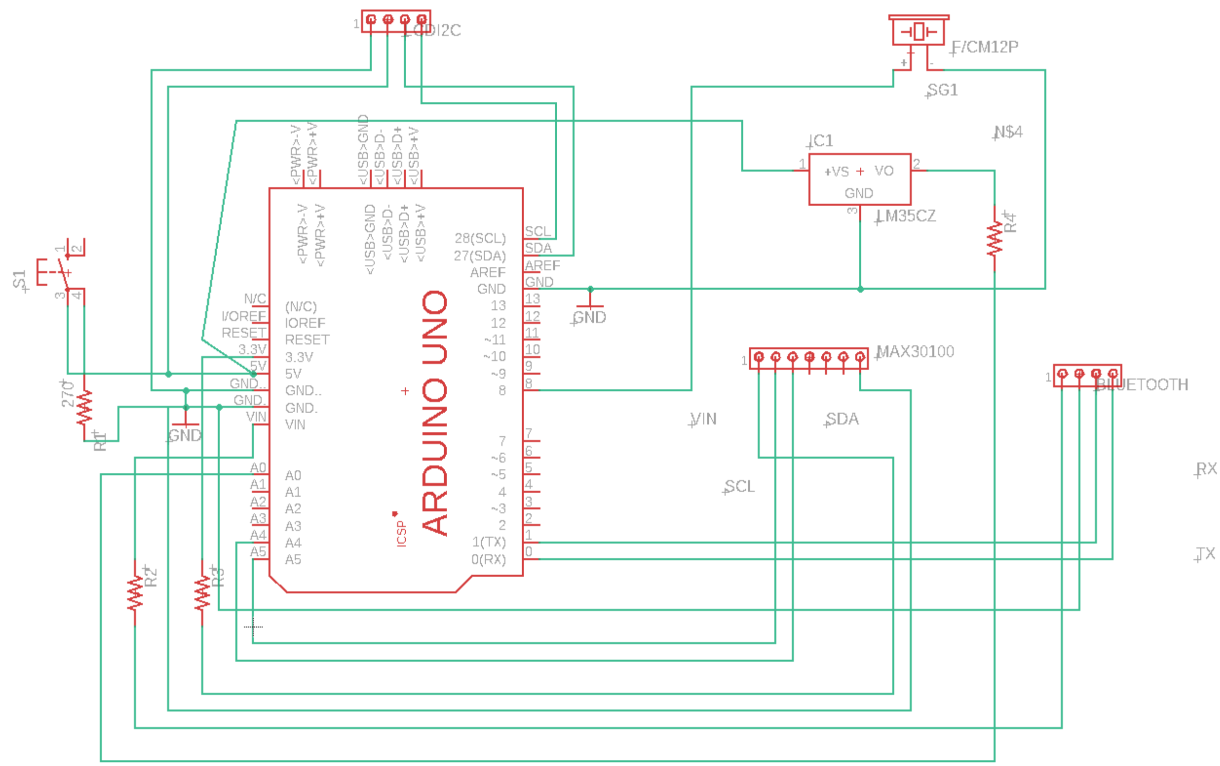


Figure 5: This is the final version of the EAGLE schematic.

It uses up the least amount of jumper resistors and translates well to a PCB. All of the components are labeled with the actual parts, rather than generic parts. The PCB was designed to be as small as possible, while still being easy to solder and attach to the Arduino. The PCB was designed to use through-hole components, although it could be modified to use surface-mount components. The PCB was designed to be powered by the desktop, although a 9V battery could be connected to the Arduino making it independent from the desktop (Figure 5).

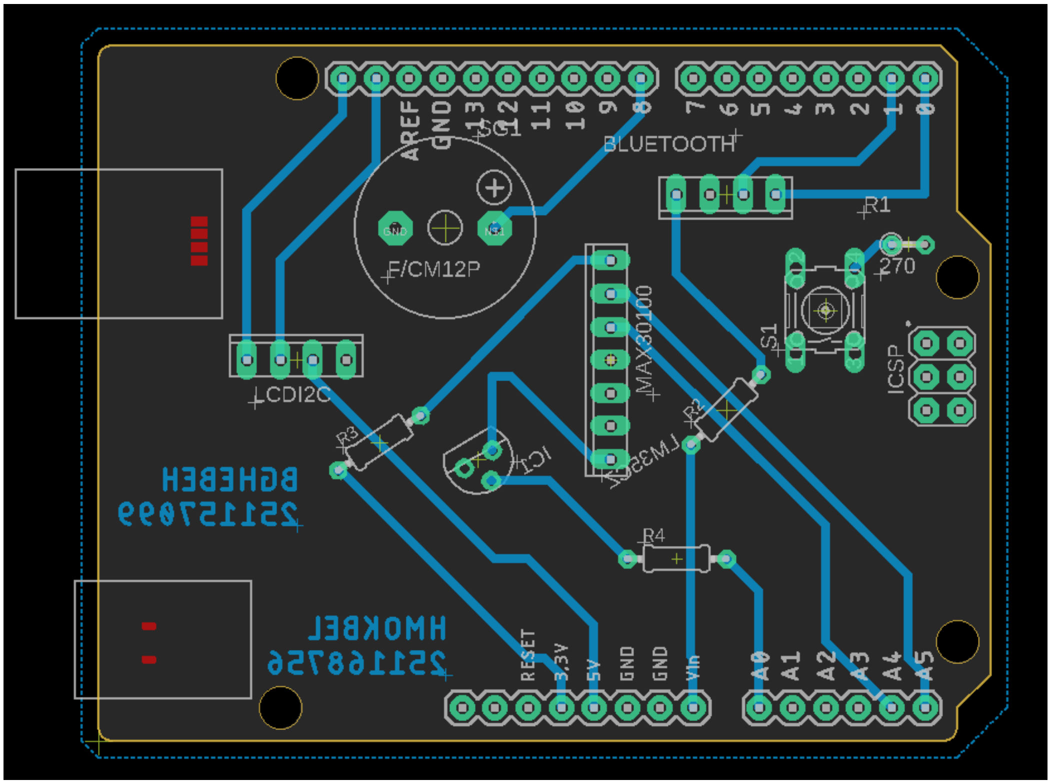


Figure 6: The final PCB with the pin connections for the MAX30100, LCD I2C and the Bluetooth.

The jumper wires help with the intersecting wires, especially around the analog pins. We used 0.05” spacing. The pins are connected to the PCB in the same order as the actual Arduino. I also included three jumpers for the analog pins. These are for the jumpers I used to connect the analog pins to the PCB. I also added a ground plane to help with electromagnetic interference [4]. We used an isolate of 0.04” for the polygon and connection metal (Figure 6).

Debugging Activities

Issue #1: Button was not working

Debugging Test: Switching the digital connection from one side of the button to another. The functionality of the button was evident in the serial monitor (1's when pressed and 0's when unpressed)

Result: The button became operational.

Issue #2: Serial monitor not working on bluetooth device

Debugging Test: Switching around the RX and TX pins on the bluetooth module

Result: The measurements from the MAX-30100 were now being transferred over onto the bluetooth app's serial monitor (thus allowing the monitoring of one's health by another person)

Issue #3: The buzzer produced little to no sound when the button was pressed was pressed

Debugging Test: We disconnected the buzzer from the entire circuit and isolated it alone on another breadboard to test if the issue is inherently from the buzzer alone. Another test we planned to do was to try out various other kinds of codes.

Result: The issue was indeed from the buzzer itself, and upon switching the buzzer with a new active piezo buzzer, the buzzer was finally operational

Issue #4: LM-35, switch button, and Buzzer were no longer operational upon soldering.

Debugging Test: We tried fixing our soldering mistakes by essentially removing the components and re-soldering.

Result: The dysfunctionality of the devices were still present. This is why we have defined our project's conclusion as partially functional, since the MAX-30100 (main module), bluetooth and LCD were still working after the soldering.

Issue #5: LCD was not functioning at all.

Debugging Test: The two debugging starting points we defined were potentially at the code or the device was simply no longer functional.

Result: Upon both of our debugging tests failing, we called upon a TA to help us. Turns out that the issue originated in the LCD potentiometer that is attached in the back.

Code: Explained (***See Appendix A***)

The code above is the heart of the device. It is basically what makes the device function the way it does. To make use of the code just copy the code script and paste on the Arduino IDE. Finally, run the code with the Arduino plugged in to run the software.

How the device works is very simple. The Arduino gets the input from the MAX30100, LM35 and prints it on the Serial Monitor. The LM35 reads analog input and calculates the temperature. The Serial Monitor and Bluetooth are interrelated: whenever the Serial Monitor gets updated, the Bluetooth terminal reads the Serial Monitor. The LCD also acts as a display system.

Whenever the measurements are updated, the LCD updates the values and prints them on the display using the I2C.

To reset the device, the button is connected to the digital pin where the Arduino reads the Digital Input. Depending on the value of the button (HIGH or LOW), the device will either reset activating the digital pin connected buzzer or continue measuring .

Precautions

- ❖ Ensure the LM-35 (temperature sensor) is attached and connected correctly, as evident in the schematics and diagrams in this report . Misconnections could lead to burning the LM-35 sensor module.
- ❖ Ensure that the MAX-30100 device is attached and connected correctly to the right voltage inputs, as evident in the schematics and diagrams in this report. Misconnections could lead to the burning of the MAX-30100 module.
- ❖ Ensure that the PCB design has zero errors before being printed. Potential short circuits in the PCB (i.e between 5 volts, 3.3 volts and ground) can fry the Arduino Uno Board.
- ❖ Ensure that you are working within a safe workspace
- ❖ Do not connect and disconnect components while the power is still on. Ensure that the power is off if changes need to be made.

User Manual

1) Install the Arduino Desktop IDE (Works for Windows, Mac OS, and Linux) and install the software Eagle. Install Bluetooth terminal app on an android device.

2) Download the following libraries: SoftwareSerial.h, Wire.h, LiquidCrystal_I2C.h, and MAX30100_PulseOximeter.h

3) Purchase the following: Arduino uno board, LM-35 sensor, Bluetooth Module HC-06, LCD 16 pin connection, LCD I2C attachment, Soldering kit, MAX-30100, breadboard, Button, Buzzer, jumper wires, and female-to-female connectors (10x). The price of all components is listed in the report.

4) It is advised to build the components individually first to test out the functionality of each component on the breadboard and check if your code is proper.

5) Integrate the entire build together as evident in this report and ensure overall functionality.

6) Design the PCB schematic and board on Eagle as done in this report. You can obtain the actual PCB from various online shops in Canada (Recommended shop and our sponsor: www.pcbway.com)

7) Build the entire project using the components ordered and PCB that you designed. Use the diagrams in this report for guidance.

8) Use the following code (see appendix) provided in this report and add it to Arduino's interface.

9) Run the project and happy tinkering!

Results & Evaluation

Goals	Test Preformed	Expectation/Results - 1 (not met) - 2 (relatively met) - 3 (exceeds expectation)	Explanation
Documentation of entire project	---	**2**	-The documentation process was completed in a step-by-step manner with appropriate diagrams, pictures, and figures.
Achieve Functionality of all sensors and actuators (individually) (March 11)	-Each sensor's functionality was tested through tinker cad and on the breadboard	**2**	-The full goal was not met. The button, buzzer, LM-35, and LCD were functioning as intended -For the MAX-30100 and Bluetooth, the code for both components was not right.
Breadboard (Collective functionality of all components (March 18)	-The entire project was built collectively as intended on the breadboard, excluding the Bluetooth module	**2**	-At this point of the project we had yet to figure out the code for the Bluetooth device
PCB (final build) 2 (March 25-29)	-The PCB design was first tested through eagle's software for errors. The overall functionality of the final project was tested (check for all sensors and actuators working)	**1**	-Upon soldering all the components onto the PCB, the LM-35, buzzer, and button were no longer working. The LCD, Bluetooth module, and MAX-30100 devices were functional.

Conclusion

The goal of the project was to create an ”at home” medical device to assess one’s general health. Our group’s objectives were to to create the right appropriate tinker-cad simulations, and PCB under the right number of sensors and actuators to accomplish our device’s overall functionality. The full functionality of the project was partially met; however, other aspects of the project were fully met (simulations, PCB design, functionality of individual components). From a retrospective standpoint, we should have depended on online resources far more heavily in the beginning of this project rather than venturing off with our own unique ideas.

Bibliography

[1] | B. S. Lim, Sujeong LimSujeong Lim is a Research Analyst at Counterpoint Research based out of Seoul Korea. In Counterpoint, and Sujeong Lim is a Research Analyst at Counterpoint Research based out of Seoul Korea. In Counterpoint, “Smartwatch market grows 24% yoy in 2021, records highest ever quarterly shipments in Q4,” *Counterpoint Research*, 14-Mar-2022. [Online]. Available: https://www.counterpointresearch.com/global-smartwatch-market-2021/. [Accessed: 01-Apr-2022].

[2] “MAX30100 Pulse Oximeter and Heart-Rate Sensor IC for Wearable Health.” Maxim Integrated Products, Inc, 2014.

[3] “LM35 Precision Centigrade Temperature Sensors.” Texas Instruments, 1999.

[4] W. D. Kimmel, “Using grounding to control EMI,” *mddionline.com*, 07-Aug-2017. [Online]. Available: https://www.mddionline.com/news/using-grounding-control-emi. [Accessed: 01-Apr-2022].

[5] “Iowa head and Neck Protocols,” *Pulse Oximetry Basic Principles and Interpretation | Iowa Head and Neck Protocols*. [Online]. Available: https://medicine.uiowa.edu/iowaprotocols/pulse-oximetry-basic-principles-and-interpretation. [Accessed: 01-Apr-2022].

[6] P. Barnhart, “Can you trust a wristband heart rate monitor's accuracy?,” *All The Stuff*, 21-Mar-2022. [Online]. Available: https://allthestuff.com/wrist-heart-rate-monitor-accuracy/. [Accessed: 01-Apr-2022].

[7] “8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash.” Atmel Corporation, San Jose, CA, 2015.

[8] Cypress Semiconductor Corporation. (2013). Character LCD with I2C Interface (I2C LCD). San Jose, CA.

Appendix A

Code

```
#include <SoftwareSerial.h>

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "MAX30100_PulseOximeter.h"

#define REPORTING_PERIOD_MS     1000


PulseOximeter pox;
uint32_t tsLastReport = 0;
LiquidCrystal_I2C lcd(0x27, 16, 2);

float sensorTemp = 0.0; // For storing raw data from LM-35
float Temperature = 0.0; // For storing the converted temperature
int analogInputPin = 0; // Data out of LM-35 is connected in Arduino's analog Pin A0
SoftwareSerial BTserial(0, 1); // RX | TX

void setup()
{
        Serial.begin(9600);
        Serial.print("Initializing pulse oximeter..");
        delay(1);
        pinMode(analogInputPin, INPUT);
        Serial.println("Enter AT commands:");
        delay(1);

        // HC-06 default serial speed is 9600
        BTserial.begin(9600);

        lcd.begin();
        lcd.backlight();

        // 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");
        for(;;);
        } else {
        Serial.println("SUCCESS");
        }
        pox.setIRLedCurrent(MAX30100_LED_CURR_7_6MA);
        Serial.begin(9600);

  lcd.begin();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Life360");
  lcd.setCursor(0, 1);
  lcd.print("Loading...");

  delay(2000);

  if (!pox.begin()) {
```

```
        Serial.println("FAILED");
        delay(1);
        for (;;);
  } else {
        Serial.println("SUCCESS");
        delay(1);
  }
  pox.setIRLedCurrent(MAX30100_LED_CURR_7_6MA);

}

void loop()
{


        // Make sure to call update as fast as possible
        pox.update();
        if (millis() - tsLastReport > REPORTING_PERIOD_MS) {

        Serial.print("Heart rate: ");
        delay(1);

        Serial.print(pox.getHeartRate());
        delay(1);

        Serial.print("bpm / SpO2: ");
        delay(1);

        Serial.print(pox.getSpO2());
        delay(1);

        Serial.println("%");
        delay(1);



        tsLastReport = millis();
        }
        sensorTemp = analogRead(analogInputPin); // Reads sensor data from pin A0
        Temperature = sensorTemp * 0.48875855; //(+Vcc * 1000 / 1023) / 10

        pox.update();
  if (millis() - tsLastReport > REPORTING_PERIOD_MS) {

        lcd.setCursor(0 , 0);
        lcd.print("BPM:");
        lcd.print(pox.getHeartRate());
        lcd.setCursor(8, 0);
        lcd.print("|");
        lcd.print("Temp: ");
        lcd.setCursor(0 , 1);
        lcd.print("Sp02:");
        lcd.print(pox.getSpO2());
        lcd.print(" ");
        lcd.setCursor(8 , 1);
        lcd.print("|");
        lcd.print(Temperature);
//      lcd.setCursor(14,0);
//      lcd.print("");

        if (pox.getSpO2() >= 96) {
```

```
        lcd.setCursor(15 , 1);
        lcd.write(1);
        }
        else if (pox.getSpO2() <= 95 && pox.getSpO2() >= 91) {
        lcd.setCursor(15 , 1);
        lcd.write(2);
        }
        else if (pox.getSpO2() <= 90) {
        lcd.setCursor(15 , 1);
        lcd.write(3);
        }
}
        if (BTserial.available())
        {
        Serial.write(BTserial.read());
        delay(1);

        }

        if (Serial.available())
        {
        BTserial.write(Serial.read());
        delay(1);
        }}
```