Introduction

The aim of this deliverable is to develop the second prototype of the project.

The second prototype is a follow up on the first, based on the client feedback and the tests that have been carried out during the first prototyping session and several short meet-ups after that. Prototype 2 will mainly involve the incorporation of our client's feedback into a new design for Prototype 2, the assembling of the electrical subsystem and running tests on the oximeter's functionality. The second prototyping sessions will also include the introduction of the GSM module which is the planned alert system for the final product.

Prototype 2 is one of the last few prototypes before the final product, therefore, Prototype 2 will be a stepping stone for the team to further progress and improve for Prototype 3.

Client Feedback

Recently, we had the opportunity to pitch our Prototype 1 to our client,Tali Cahil. We received feedback on our first prototype, which mainly focused on the exterior design and concept of the device. After hearing our pitch,Tali mentioned that many developers have been working on watch concepts that are similar to ours and expressed her concern over the prototype's ability to stay fastened to the user's in order for the oximeter to maintain constant contact. It is a pressing concern as it ensures that the device is constantly tracking the user's blood oxygen level, making sure that the device is able to serve its intended purpose. Having taken her feedback into account, the team had to re-develop a new design and ensured that this criterion was made the top priority in the design aspect of the product. Tali also mentioned that the battery life of the device had to be as long as possible, this also includes making sure that charging the device/ changing its batteries does not inconvenience the users too much. Both aspects will be further explored in Prototype 2.

Discussion

Follow-up of Prototype 1

Following our first prototyping session, the team discovered certain areas of improvement:

1. The cut-outs for the display and oximeter were too big for the actual display and oximeter's size.

2. There was an error with the positioning of the cut-outs

3. The casing overall was far larger than the size of the components.

4. As per Tali's feedback, the device has to be well-fastened to the user, yet has to be comfortable, non-intrusive and discreet.

The team also noted that purchasing of the components need to be well-scheduled as the amount of work that was done for Prototype 1 was heavily capped due to the untimely arrival of the required electrical components. The team followed up on the first prototype's results and feedback and made drastic changes to the casing's design.

Exterior design

Combining the results from Prototype 1 as well as Tali's feedback, the team has decided to shift the previous cut out for the oximeter to the base of the casing. This would ensure that the oximeter is always in full contact with the user's wrist.

In order to securely keep the device on the user's wrist, the team has decided to experiment with elastic bands or wristbands. The reason is that elastic bands could keep the device securely fastened to the user's wrists. This ensures that the oximeter sensor is always in contact with the wrist, it is also more comfortable than tightening something like a watch strap, this would help to improve the comfort factor for users when they use it. It is also much likely to address the mass public as elastic bands stretch to fit whatever it is being wrapped around.

Electrical subsystem

The electrical subsystem will consist of the following components:

1. MAX 30100 Pulse Oximeter
2. SIM 800L GSM Module
3. IIC OLED SSD 1306
4. Arduino Nano

The entire project will run on the Arduino Nano, with the IIC OLED being the display, the GSM Module as the alert message system and the Pulse Oximeter as the blood oxygen sensor.

Pulse Oximeter

The team is using the MAX 30100 blood pulse oxygen sensor. The sensor is about 0.6 x 0.4 inch, the size of a person's fingertip. It uses an LED to emit red light which will then be used to measure the user's SpO2 (blood oxygen level). The main functionality of the project revolves around this component.

To start off, the team ran a few tests to explore the responsiveness of the MAX30100. The team tested the blood oxygen readings for 2 main body parts: the wrists and the fingers. The aim of the tests were to find out how responsive the pulse oximeter was. The team wanted to investigate if the sensor was able to pick up slight changes in oxygen level and that the component was able to run consistently without failure.

In order for the pulse oximeter to run on the Arduino nano, the team had to solder off the oximeter's surface mounted resistors and replace them with external resistors. The reason being that the MAX30100's voltage rating does not match the Arduino nano, hence the resistors will help to adjust the voltage. In order to use the MAX30100, the team used the following code:

MAX30100_Minimal

```
You should have received a copy of the GNU General Public License
along with this program.  If not, see <http://www.gnu.org/licenses/>.
*/

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

#define REPORTING_PERIOD_MS     1000

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

uint32_t tsLastReport = 0;

// Callback (registered below) fired when a pulse is detected
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");
        for(;;);
    } else {
        Serial.println("SUCCESS");
    }

    // The default current for the IR LED is 50mA and it could be changed
    //   by uncommenting the following line. Check MAX30100_Registers.h for all the
    //   available options.
    // 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();

    // Asynchronously dump heart rate and oxidation levels to the serial
    // For both, a value of 0 means "invalid"
    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();
    }
}
```

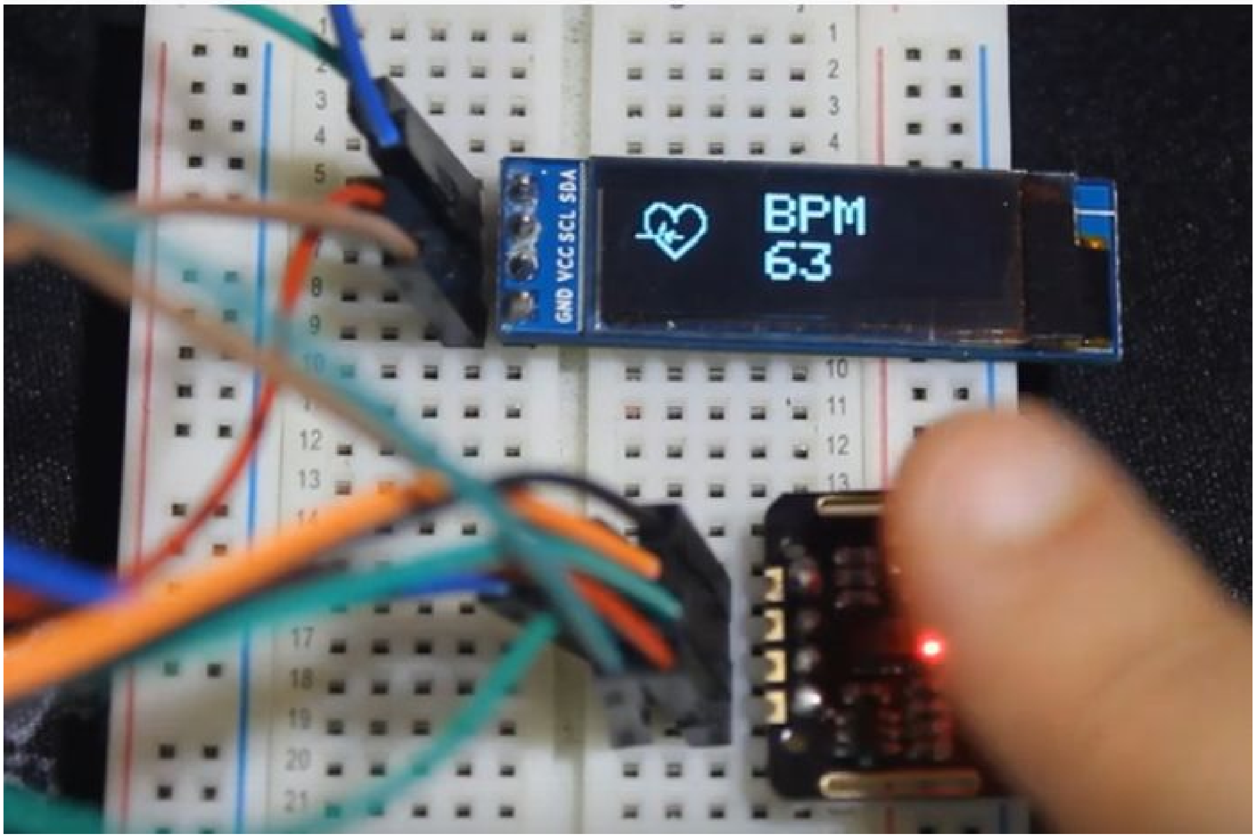
Error while printing.

Copy error messages

```
Sketch uses 9056 bytes (29%) of program storage space. Maximum is 30720 bytes.
Global variables use 729 bytes (35%) of dynamic memory, leaving 1319 bytes for local variables. Maximum is 2048 bytes.
```

61

Arduino Nano on COM9



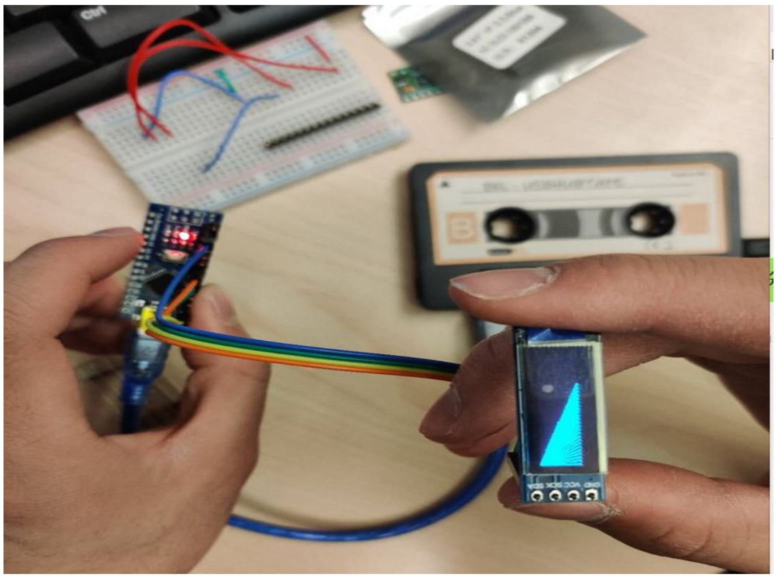
GSM Module

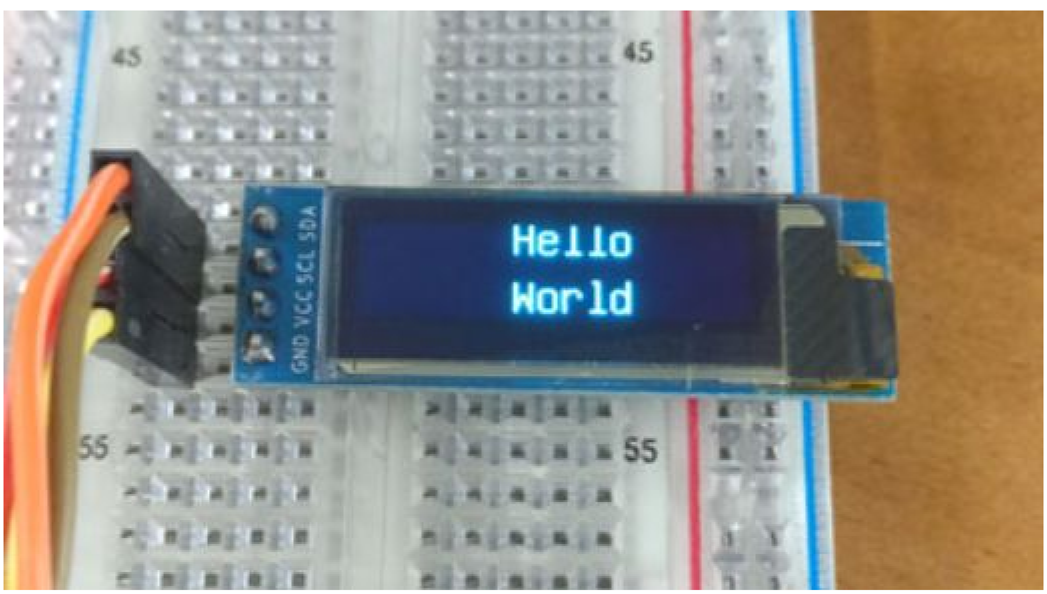
The team decided to use a GSM module as an alert system for the device. The GSM Module will be responsible for sending short messages (SMS) or calls to a pre-set emergency contact. Despite the ever-growing use of the internet and Wi-Fi, the team chose to use SMS / calls because they are much more reliable and easier to access than internet connectivity.

The aim of the tests for the GSM Module is to test the component's functionality. The team needs to ensure that the Module is able to consistently send out messages without fail. The team also ran tests to explore the minimum and maximum range the GSM module could send messages to.

Display

The team chose to use an OLED display for the project. OLED displays are not backlit and hence they have low energy consumption yet bright screens. However one thing to note is that as OLED displays can have each individual pixel controlled, it is very possible and easy for the display to completely blackout. The aim of the team's tests were to monitor the brightness and visibility of the LED, its energy consumption and if it is sustainable for the device.





After testing the OLED, the team found that the screen provided quite an adequate amount of brightness, it was not too dim and was very clear. An issue faced was that the OLED would disconnect from the wiring due to the lack of soldering, it would be good to note that a certain amount of soldering work will have to be put in in order for the OLED screen to stay connected and functioning.

Battery life

After analysis of the various components used, the team calculated the required battery to power the device. In order to both accommodate the needs of the device as well as to satisfy the client's requirements, we had to ensure that the battery we used was able to power the device for a sustainable period of time. We also had to make sure that the battery does not add too much weight or increase the size of the device, as it will make the users feel uncomfortable and less compelled to use them.

Below are the calculations:

LED- 20 mA
Arduino Nano - 15.5 mA
Pulse Oximeter - 280 mA
GSM - 300 mA

Total- 615.5mA
Battery that we are using 5000/615.5 = 8.12 hours
= 8 hours and 7 minutes of battery life

The team concluded that the Olight 217C50 5000mAH Customized Rechargeable Battery is most suited for the device. It will power the device for a good amount of time. However, the size of the battery may be slightly larger than what the team would desire it to be, hence the team will continue to put research into this aspect.

Conclusion

We have had a successful second round of prototyping, even though our prototype did not perform as planned. Most of our proof-of-concepts are valid as of our expectations and we are on schedule for our third round of prototyping. Hopefully, if everything goes as expected, our team would be prepared for Design Day after the completion of prototype three.

Our team would be working on combining the electrical components and physical casing of our device in the following weeks. We would also screen through our codes and make necessary changes to accommodate any changes. This will probably increase the duration of our weekly team meeting in order to complete the task mentioned above. Finally, our team would be doing a final round testing to ensure that all components of our device are functioning and ready for Design Day.