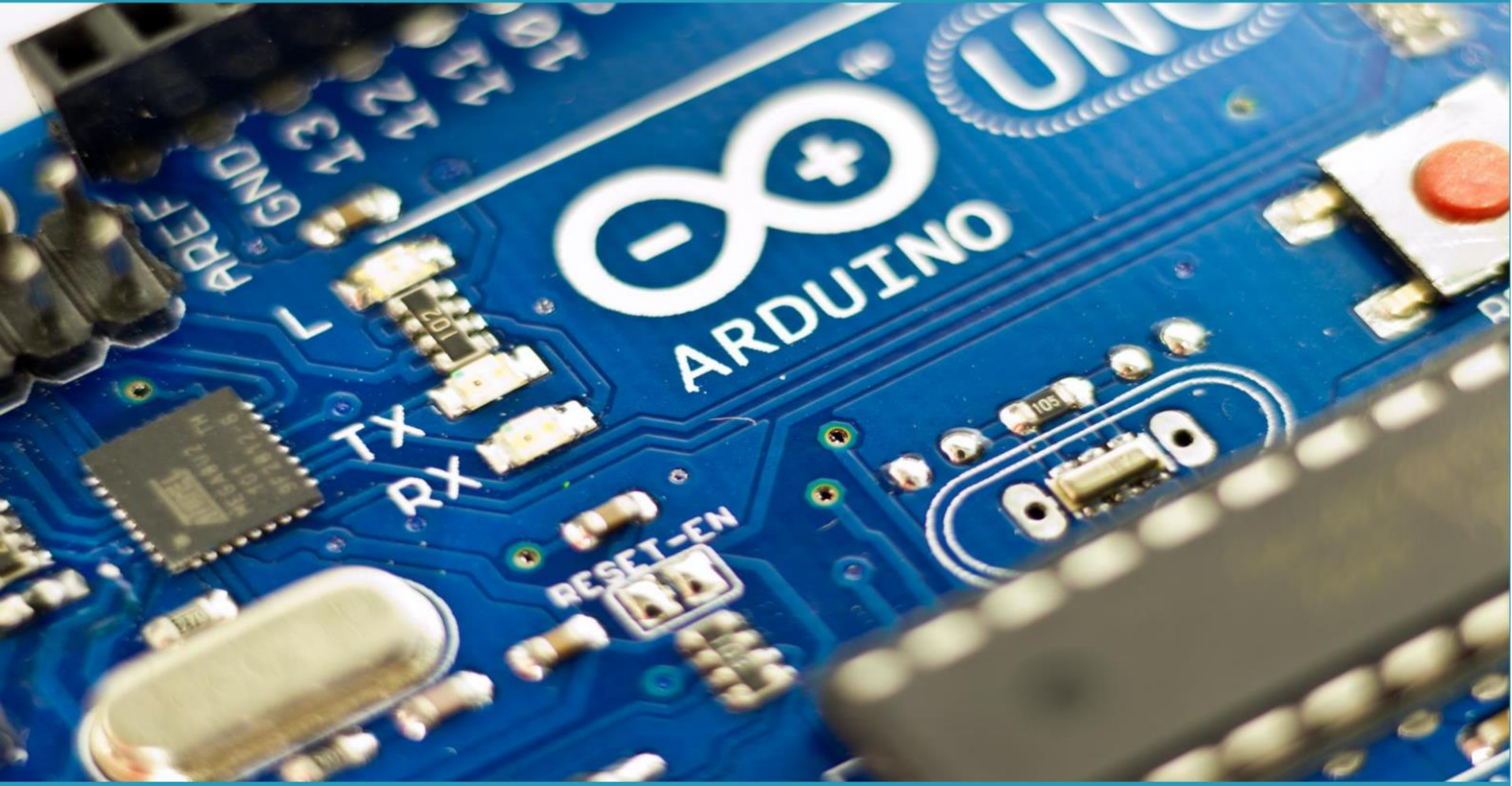


TEAM GO

Health & Care

Heartbeat and saturation monitor

Marie-Estelle GOUNALANE

Indrajit RAJASOORIAR

Mr. Aomar OSMANI

Mr Seyed MODARESI

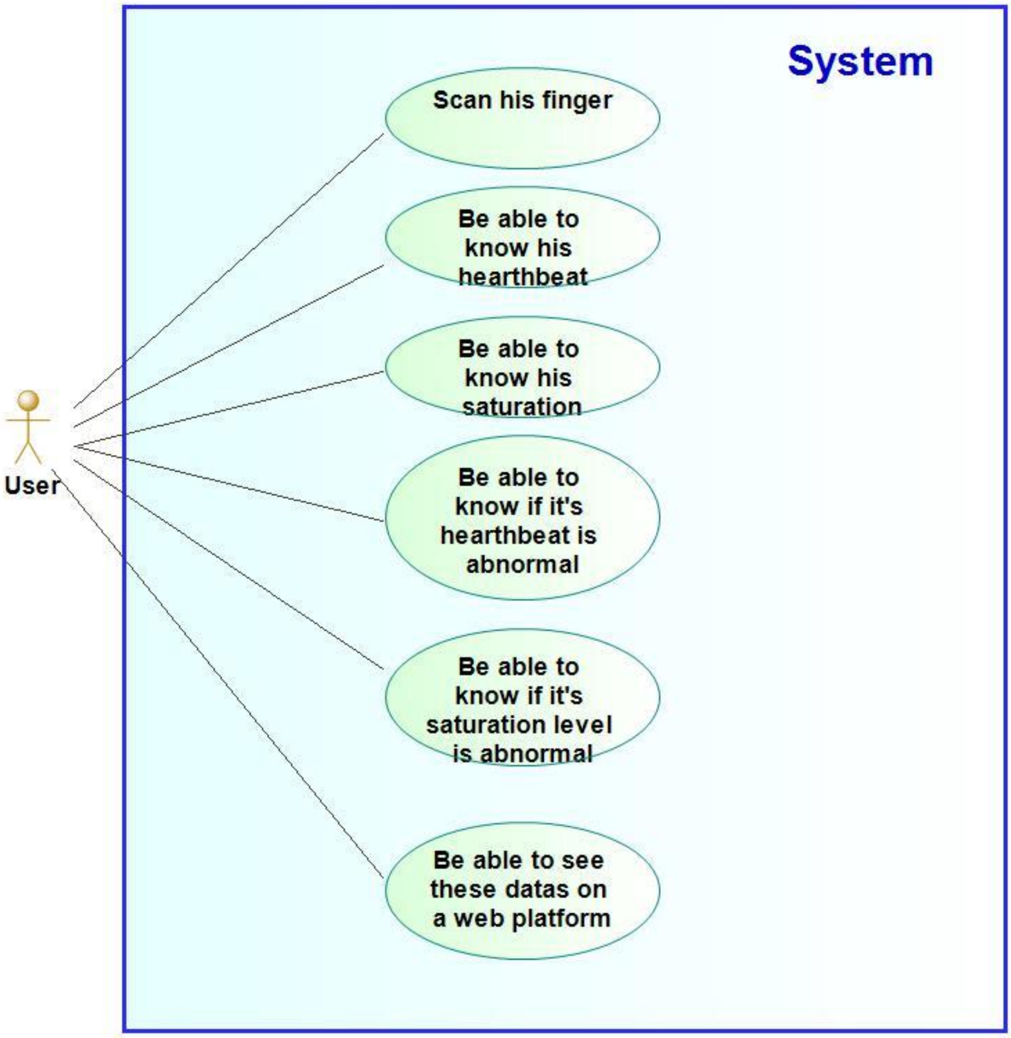
Year 2020/2021

Table des matières

I. Description of project

We wanted to create a real time heartbeat and saturation measurement device. It will show the user's heartbeat, the saturation rate furthermore, he will be notified if these measures are abnormal. To use this device, he has to put his finger on the sensor, it will scan his heartbeat and saturation level in the platform. The monitor has also 2 LED which will light if the level of the heart rate or the saturation level is abnormal.

Here is a use case diagram of the functionality of our project:


System
Scan his finger
Be able to know his hearthbeat
Be able to know his saturation
Be able to know if it's hearthbeat is abnormal
Be able to know if it's saturation level is abnormal
Be able to see these datas on a web platform
User

To use this prototype the user should first scan his finger so the device can get his rate heartrate and saturation level. Once the user scanned his finger, he should be able to know his heartbeat and his saturation level. Beside in the case when the heartbeat or saturation level are higher or lower than the normal recommended, the user will be notified. He should be able to see all these data in web platform.

II. Components list

To do this project we use the following materials:

- Arduino UNO
- Max 30100
- 1 Blue LED
- 1 Red LED
- 2 10 KOhm Resistance
- 10 Wires
- 1 Breadboard
- 1 Cable USB UNO R3

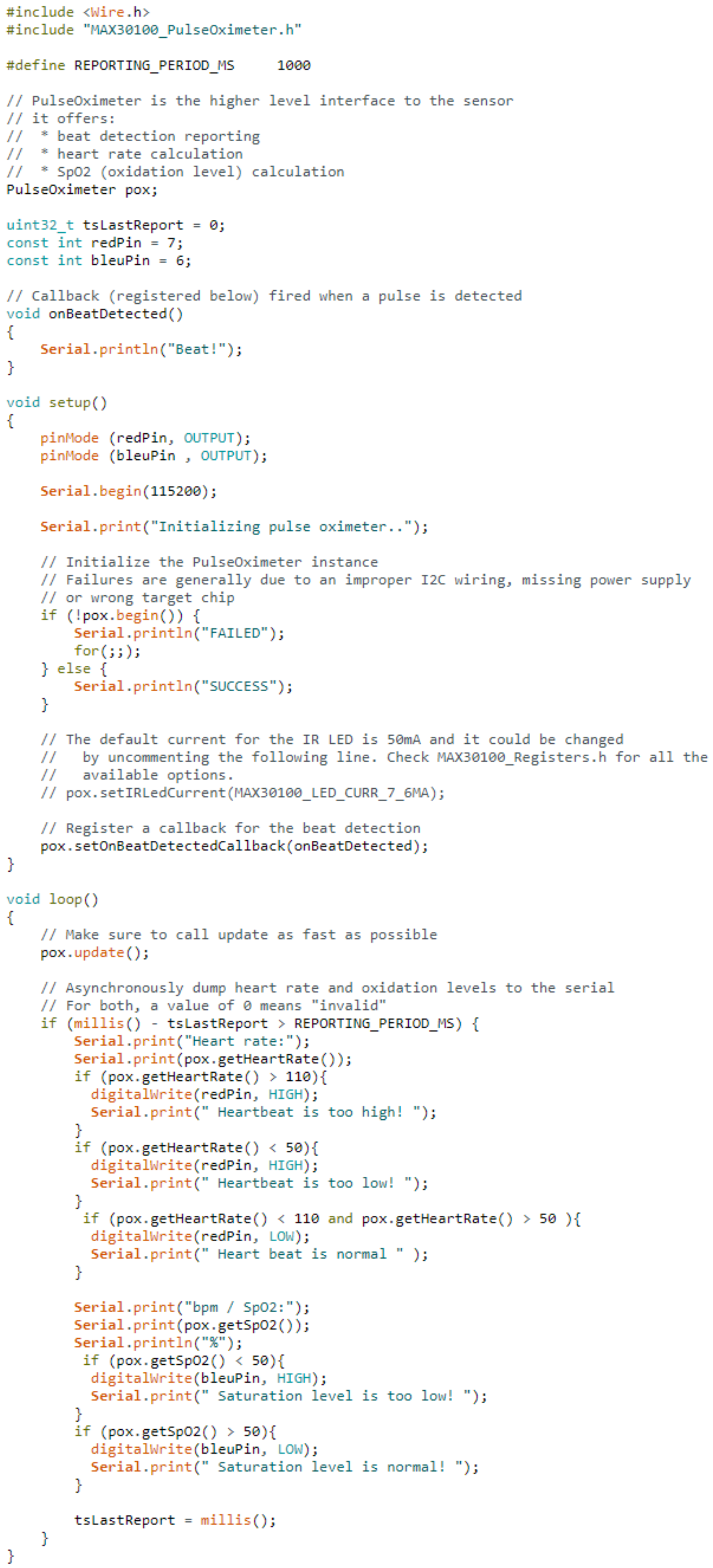
III. Software side

To build this project we used two software, which are:

- Arduino IDE : It is a platform to written and compile Arduino sketch that is similar to C and C++. It is used to upload code into a Arduino card.
- Node Red : It is a flow-based software used in IOT to connect hardware to an API.

With the help of Arduino IDE, we write a program to measure our heartbeat and saturation level and monitor it though a serial monitor. We then used Node Red to display the output in the web platform.

Node-RED is a flow-based development tool for visual programming developed by IBM. It's used in IoT to connect a hardware to an API. Hence, we produced a dashboard for our Heartbeat and Saturation level application



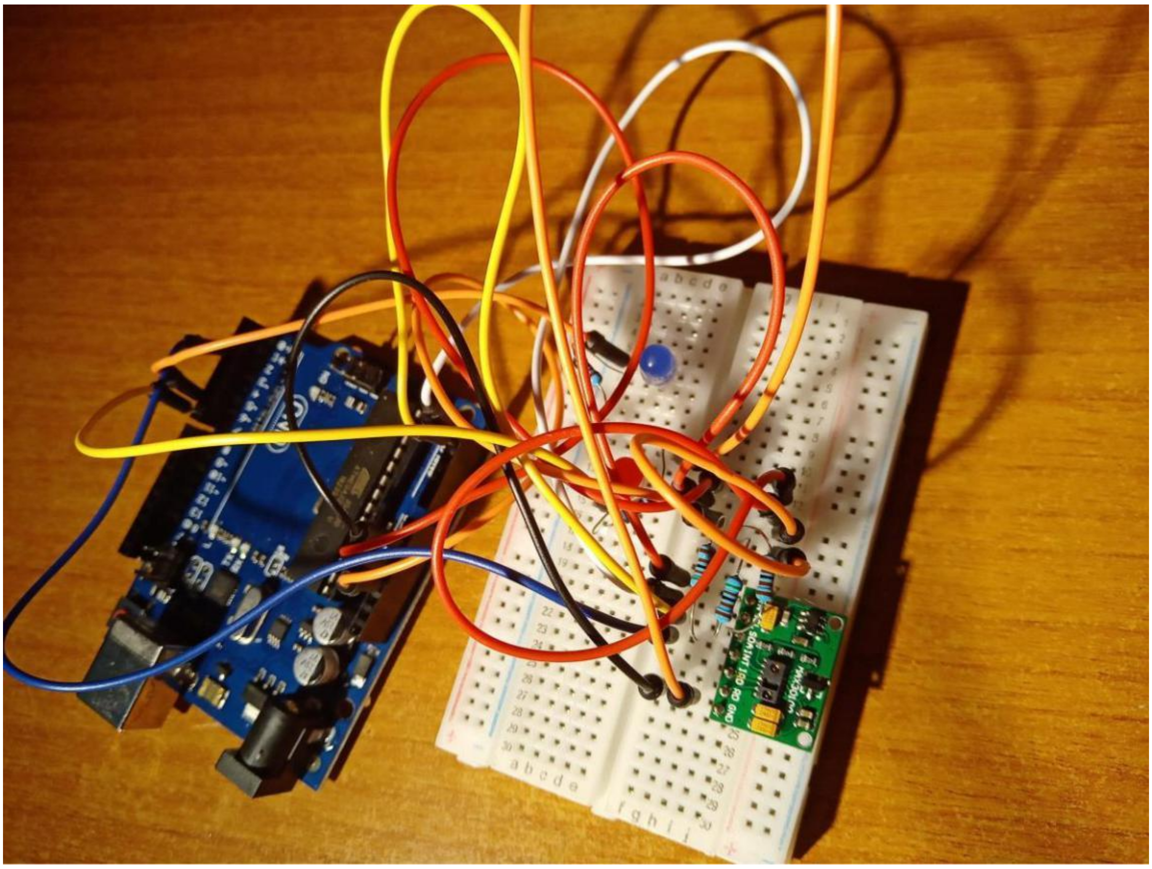
```
#include <Wire.h>
#include "MAX30100_PulseOximeter.h"

#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;
const int redPin = 7;
const int bleuPin = 6;

// Callback (registered below) fired when a pulse is detected
void onBeatDetected()
{
    Serial.println("Beat!");
}

void setup()
{
    pinMode (redPin, OUTPUT);
    pinMode (bleuPin , OUTPUT);

    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());
        if (pox.getHeartRate() > 110){
          digitalWrite(redPin, HIGH);
          Serial.print(" Heartbeat is too high! ");
        }
        if (pox.getHeartRate() < 50){
          digitalWrite(redPin, HIGH);
          Serial.print(" Heartbeat is too low! ");
        }
         if (pox.getHeartRate() < 110 and pox.getHeartRate() > 50 ){
          digitalWrite(redPin, LOW);
          Serial.print(" Heart beat is normal " );
        }

        Serial.print("bpm / SpO2:");
        Serial.print(pox.getSpO2());
        Serial.println("%");
         if (pox.getSpO2() < 50){
          digitalWrite(bleuPin, HIGH);
          Serial.print(" Saturation level is too low! ");
        }
        if (pox.getSpO2() > 50){
          digitalWrite(bleuPin, LOW);
          Serial.print(" Saturation level is normal! ");
        }

        tsLastReport = millis();
    }
}
```

IV. Problem we faced and alternative solution

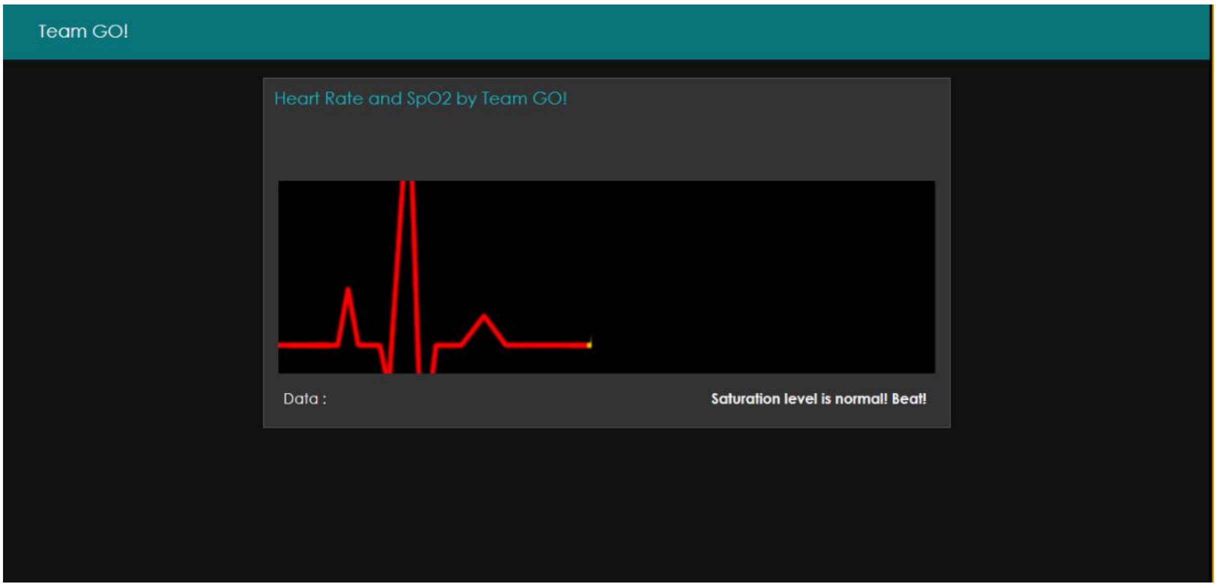
Initially we planned to use AWS EC2 to connect Arduino equipment to the web platform. However, we could not get a proper AWS account for this project, we did not have the credits for. Hence, we use the API Node Red instead.

V. Output of the application

This is the circuit:



This is the interface:



Team GO!

Heart Rate and SpO2 by Team GO!

Data :

Saturation level is normal! Beat!

VI. Conclusion

This is a basic program with an interface to show to the user his heartbeat and saturation level and if it's abnormal. However, we can make it more precise by adding some more information concerning the user's information like his age, his weight etc.. to increase the precision level of the results. Also, if it's possible by its budget we can use a cloud platform to run this application.