Middlesex University London

CST3590 Individual Project

E-Health Monitoring System

Student Name: Hugo Miguel Pinto Martins
Student ID: M00647275
Supervisor: Dr Purav Shah
Module Leader: Dr Xiaochun Cheng
10th April 2020

A thesis submitted in partial fulfilment of the requirement
for the degree of Bachelor of Science in
Computer System Engineering

Abstract:

A lot of countries around the world lakes the capability to serve his city habitants for the simple day to day diseases or routine check-ups due to not having enough staff or infrastructures.

This report provides an analysis how this device would help communities around the world from the poor countries to the rich countries. The device will be placed in the patient or patients house, and from time to time will measure the Temperature, Heart Rate and the amount of oxygen in the blood. Sending the data to a website that can be monitored by several health professionals.

Table of Contents:

Chapter 1

Introduction

Now a days a considerable amount of hospitals and GP's struggle to assist all his users in a day to day base, for example in simple

Chapter 2

Literature Review

E-Health was born between 1999 and 2000 and is a combination of a several of electronics and a remote communication to a Doctor or Database. Is also a service that share or modify the information through technology, but E-health is more focus in sending the user health data [1]&[2]. The need to improve the health service and relieve the hospitals from people just going there for example to just book an appointment with a Doctor or Nurse, or even request medication by medical prescription. And now they can do it thought the phone or Website.

Since the conception of E-health in 1999 and 2000, there has been more and more usage of it and there is why the rise of data in the database. This program transformed the Health Systems of the countries that are using it. As for example, before all the Health data was recorded manually by health system staff, although now a days all the data goes to a server that is safely secure, and in case of any malfunction or destruction of that server that will be back up of all the health data of the patients.

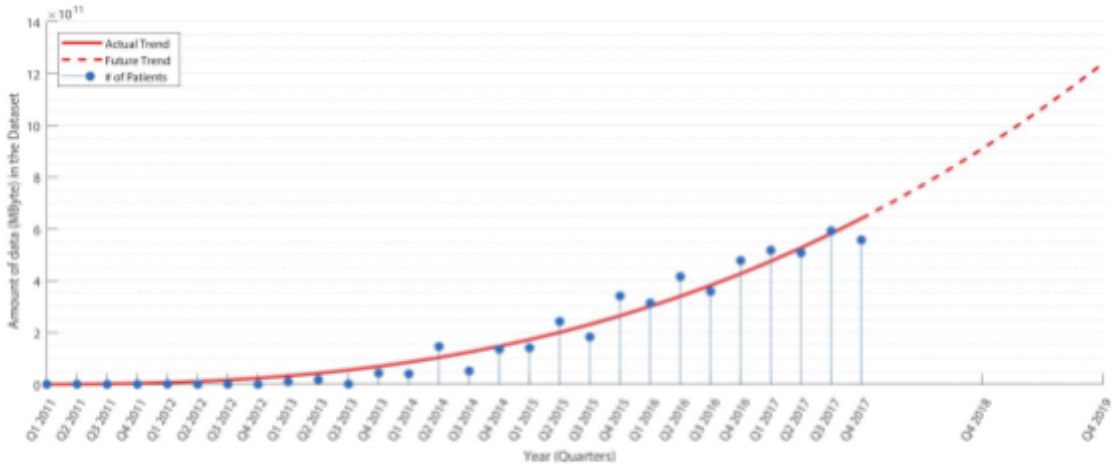


Figure - Amount of data(MB) in the Database

Every year that passes there are more and more countries, mainly in Europe, trying to implement this System in their countries spending millions to implement it. Mostly if not almost all countries with E-Health are in Europe this is because the European Union is creating laws that "demand" the countries to boost their Health System every year.

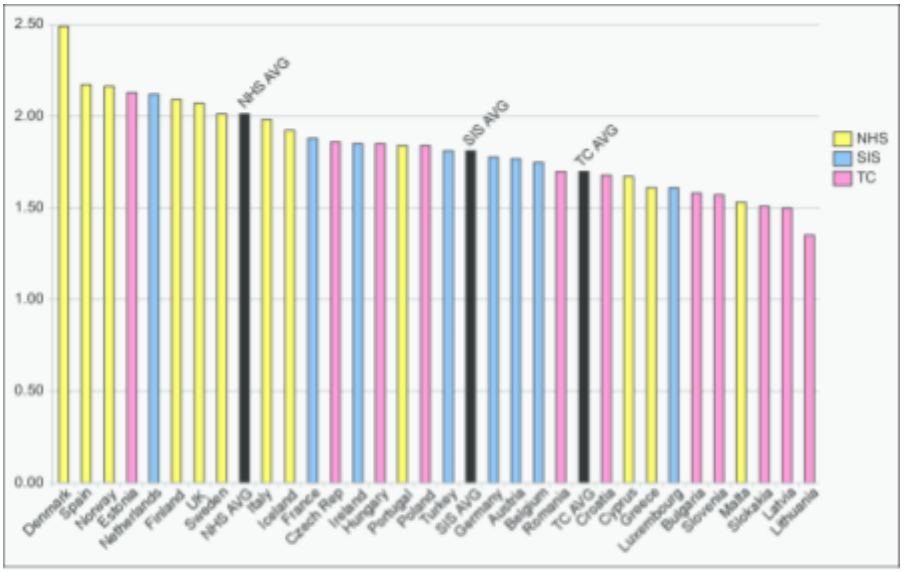


Figure - The adoption of E-Health by the national health model

How it can be seen on the image on the right side, the European Union has 3 types of countries. In red the countries that don't have at all E-Health Program and their health System is very poor like East countries, in the countries painted in yellow the program is being implemented like Italy, Spain and other ones. And at green the countries that already use the E-Health program just like almost all countries from Central Europe, North Countries and Portugal. Europe already has a lot of countries with the E-Health program (represented in green).

Figure - E-Health Europe Map

E-health can cover a lot of areas in the Health just Electronic Health record and stores all the patient data into a server. Telemedicine “exchange of information for the diagnosis and treatment of diseases and injuries, research and evaluation, and for the continuing education of health professionals” [3]. On ePrescribing the doctor can prescribe to the patient through a video or call. Big data can be used in research and development. Clinical decision support Access to decision support systems. Consumer health IT, now a days a smart watch, smartphones, Fitbit and many more can read almost all the vital signs. Knowledge Management System exhibits at the same time physical and mental characteristics, that all the organizations embraced [4].

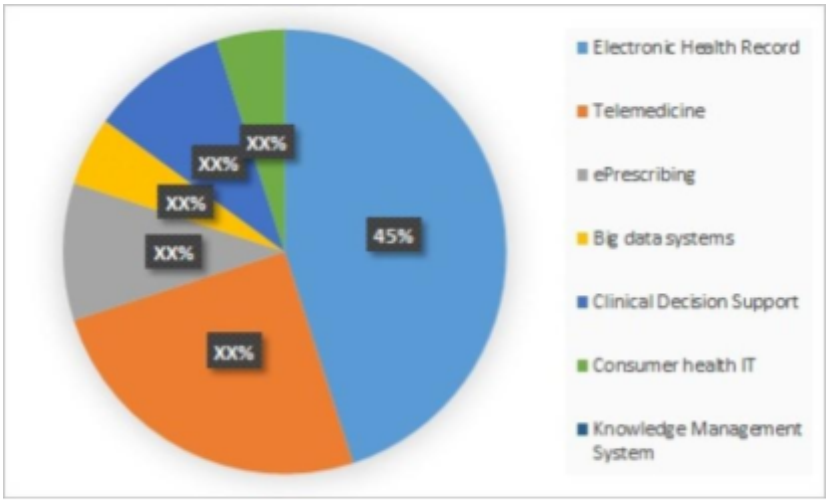


Figure - E-Health Services

Pros and Cons of E-Health

Pros of E-Health:

- All the data is online, no documents are stored on paper because an online version is “safer” due to the backups ;
- Reduce queues in the hospitals for medical prescriptions and to book appointments;
- Raise the quality of the health system;
- Reasonable cheaper;
- Patient health data available everywhere, the patient can go any hospital that has E-Health and the medical team can see all patient health history;
- Data backup(s), in the online version in mandatory to do a backup in case something happens to the server.

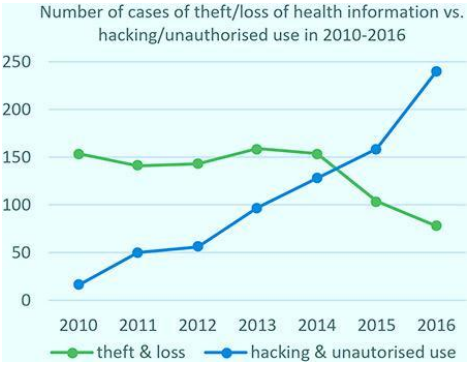
Cons of E-Health:

- Difficult for a country with a poor health system and economy to implement the system;
- All the patient personal data is online so any hacker can try to get into the servers and get all the patient personal data;

Threats

E-Health online threats are a big problem due to, in the database sometimes not only is has the patient health data with also sometimes contain financial information. Another big security threat are uneducated staff, they have big a role in the security of the system and if one is using a computer or laptop of the hospital and if installs some program or the website installs something into the computer that software or file just downloaded could a ransomware, if so that ransomware encrypt or lock essential services or even encrypt all the patient health data from the database [5].

The cyberattack on the 20th May 2017 that hit NHS, Hospitals and GP surgeries trough out England and Scotland is a good example of an cyberattack to the National Health System and it was called the biggest ransomware attack in history. The hackers send to the NHS staff that would seduce the reader to press the link in the email. Once the reader pressed the link, the ransomware block almost all desktop, laptop of hospital, all the email accounts were shootdown and encrypt some files on the NHS server. All the staff were affected so all of them were forced to go back to the old method, pen and paper . All that the ransomware was demanding was something between 300$ and 600$ worth of bitcoin to a specific bitcoin wallet address. A lot of doctors in England were forced to send their patient home and cancel the appointment [6].



And in this graph on the right side can confirm that the hacking & unauthorized used have increase regularly since 2010. But for the other end the cases that end in theft & loss of information has been decreasing.

Legal Issues

- *Licensing and reimbursement*, in connection to repayment of E-Health administrations, the EU Order 2011/24/EU gives the lawful system permitting patients to have their telemedicine administrations repaid indeed if given in a nation distinctive from their home country or residence country. Typically given that they drop inside the scope of repaid administrations in their nation of home and given that (in a few circumstances) they gotten an earlier authorization from their nation of home.

- *Data Protection*, in specific the up and coming EU Information Assurance Direction will give, among others, for a notice commitment in case of information breaches to the competent controller and the influenced individuals. An increment of fines up to 4% of the worldwide turnover of the past budgetary year. The commitment to actualize a security and security by plan approach.

- *Product Liability*, European controls on item obligation are exceptionally rigid and set out an administration of strict/ objective risk i.e. in case of failing patients should demonstrate as it were the imperfection to bolster their claim for harms and the burden of confirmation should be on makes to demonstrate that he harms were not caused by the charged deformity. The matter is indeed more complex within the case of telemedicine and E-Health administrations since a wrong treatment might for occasion be due to a delay within the communication of information by the telecom carrier and this information given to specialists might not be completely dependable. This would increment the hazard introduction for both specialists that shall depend on wrong information and producers of telemedicine administrations whose obligation may be connected to circumstances out of their control.

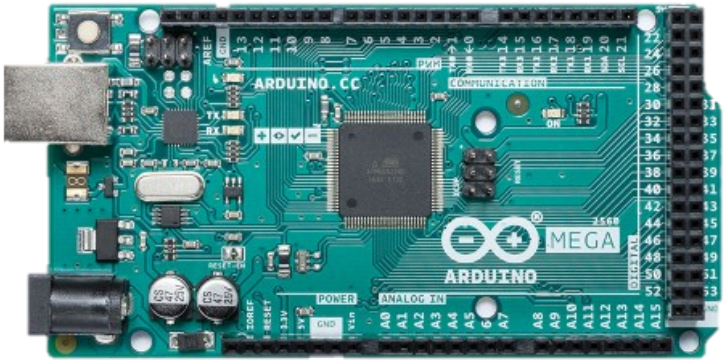
- *Information to patient*, since telemedicine and E-Health administrations frequently depend on information created by ga dgets utilized by patients themselves, in arrange to form such information solid and valuable for the arrangement of medicines, the data and enlightening given to patients are pivotal. And indeed, typically one of the most themes secured by the Italian rules on telemedicine. The Italian rules on telemedicine, moreover, endorse enlightening on the necessities to be met in terms of benefit levels between telemedicine suppliers and healing centre's as well as on ethical standards to be complied with. This can be subject that will gotten to be increasingly significant within the coming months/years [7].

There's no question that a key pre-condition for receiving E-Health is that both patients and experts must believe the innovation to be secure and successful. Building certainty in E-Health is, however, not close to having believe within the innovation itself. As well regularly we see at E-Health as a simple mechanical advancement in healthcare and have barely looked at the relationship between clients and innovation to recognize boundaries to acknowledgment. Components such as human interaction, changing relationship between patients and health experts have been generally side- lined , if not totally ignored , whereas there's adequate prove proposing that these components are vital to building certainty and trust. Therefore, we require subsequently to contribute assets in supporting all client bunches in obtaining the skillset required by modern interaction and ways of communication emerging from the utilize of eHealth, as well as modern parts and duties included.

Chapter 3
Approach and Methodology

- *Arduino Mega 2560*

The Arduino Mega 2560 is part of a big family of open source microcontrollers board but this board is based on ATmega single chip microcontroller. A microcontroller is a small IC (integrated circuit) design to controller a distinct operation in an ES (embedded system), and now a days can be found almost everywhere from vehicles, robots, home appliances among other devices. The microcontrollers are mostly a small incomplex PC (personal computer) that's designed to operate small features of larger components, without using a complex programming language and a complex OS (operating system).[1]



This microcontroller needs to be power via USB 2.0 Cable Type A/B connection or connected to an external power supply. The board operates with an Input voltage limit from external power source between 6 and 20 volts. But if the power source supplies less then 7 volts the board can become unstable and malfunction. And in case the boards is supplied with more than 12 volts the microcontroller may overheat and damage the board. The recommended input voltage ranges between 7 and 12 volts. [2]

The board has a total of 54 digital pins being from pin 0-53, a digital pin is a way of representing voltage in only 1 bit being 0 (around 0v) or 1 (above 2v).

These digital pins can be used has inputs or outputs and some pins have specialized function:

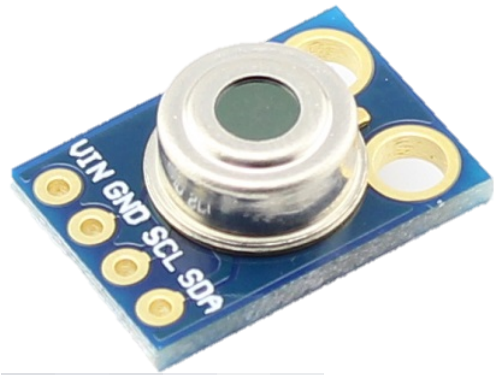
- Serial 0: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data o another microcontroller or others compatible boards/ sensors.
- PWM: 0 to 13 and 44 to 46. Provide 8-bit PWM (Pulse Width Modulation) output by outputting a constant frequency of ~500HZ, and the duty cycle can be changed by the ser with his own parameters. These pins are used mostly with LED's and DC motors.

- SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). MISO (Master In Slave Out) it's a pin to send data to the master device in case the Arduino is connected to another device and the Arduino being the slave. MOSI (Master Out Slave In) this pin the Arduino is defined has the master device and send data to his slaves. SCK (Serial Clock) in this pin a clock signal is generated by the master device to synchronize data transmission
- LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- I2C: 20 (SDA) and 21 (SCL). I2C is a communication protocol designed to enable communication between 2 boards using 2 wires referred to as SCL and SDA. The I2C works by giving a unique address, up to 255 devices. SCL is designed to generate a clock signal and synchronize the data transfers. SDA is used to transmit the data.[2]

This microcontroller also has 16 analog pins that can function as digital inputs or digital outputs. An analog signal is a continuous signal that can have any value between 0V and 5v sketching a signal identical to a wave.[2]

The Arduino Mega 2560 can be programmed with Arduino IDE. The programming language that the Arduino boards use is simply a set of C/C++ function that can be called from the code. [3]

- ## *MLX90614 Infrared Thermometer:*



MLX90614 is an infrared Thermometer specifically designed for non-contact temperature readings. The sensor has integrated on his board an IR (Infra-Red) sensitive detector chip, the sensor has a temperature readout resolution of 0.01°C giving this sensor a phenomenal accuracy. The board has a wide temperature range, -40°C to +125°C for ambient temperature and -70°C to +380°C for object temperature. The output of the sensor can be configurated in two ways, by digital output connecting to the PWM pin of a microcontroller, connecting the GND pin to ground, join SDA to a PWM pin of the microcontroller, attaching Vin to a 5V power source, join SCL and Vin and inserting a capacitor of 0.1µF between the junction of SCL with Vin, and ground. Or using I2C pins, by connecting the pins SDA and SCL to the SDA and SCL pins in the microcontroller, connect GND and Vin to ground and a 5v power source, and also inserting a capacitor of 0.1µF between the junction of SCL with Vin, and ground. By outputting using the I2C the user could add more different sensors or even more than one MLX90614 by just changing the address of the sensors. And the user would use only 2 ports leaving unoccupied rest of the other ports. This method helps if the microcontroller doesn't have a large amount of ports but will have a I2C port.

The operation of the MLX90614 is controlled by an indoor state machine, which controls the measurements and calculations of the object and ambient temperatures and does the post-processing of the temperatures to output them through the PWM output or the I2C compatible interface. The ASSP (Application Specific Standard Product) supports 2 IR sensors. The output of the IR sensors is amplified by a low noise low offset chopper amplifier with programmable gain, converted by a Sigma Delta modulator to one bit stream and fed to a strong DSP for further processing. The signal is treated by programmable (by means of EEPROM contend) FIR and IIR low pass filters for further reduction of the band width of the input to realize the specified noise performance and refresh rate. The output of the IIR filter is that the measurement result and is out there within the internal RAM. 3 different cells are available: One for the on-board temperature sensor and a couple of for the IR sensors. supported results of the above measurements, the corresponding ambient temperature Ta and object temperatures To are calculated. Both calculated temperatures have a resolution of 0.01˚C. the info for Ta and To are often read in two ways: Reading RAM cells dedicated for this purpose via the 2-wire interface (0.02°C resolution, fixed ranges), or through the PWM digital output (10 bit resolution, configurable range). within the last step of the measurement cycle, the measured Ta and To are rescaled to the specified output resolution of the PWM) and therefore the recalculated data is loaded within the registers of the PWM state machine, which creates a continuing frequency with a requirement cycle representing the measured data. [1]

- ## *MAX30102 Blood Oxygen/Heart-rate:*

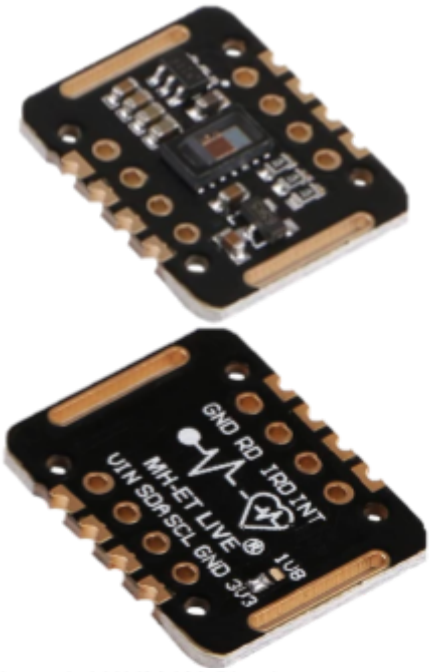
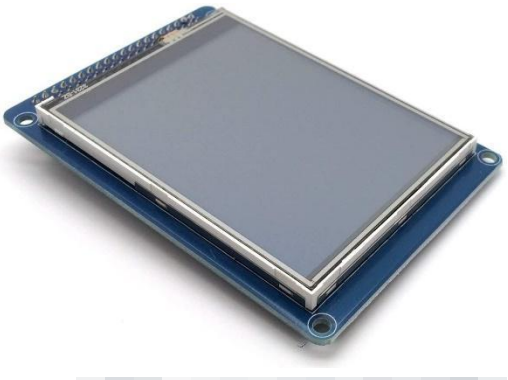


Figure 1- MAX30102 Blood Oxygen/Heart-rate

The MAX30102 is a biosensor and combination of 2 different sensors all in one sensor, has incorporated a heart rate monitor, blood oximeter sensor in a LED reflective solution, the board operates with 5v power source and only can communicate through the I2C interface.

This sensor can be found in a lot of smartphones, smartwatches, tablets and fitness devices due to his small form factor and ultra-low power consumption when used. The biosensor operates by shining a LED (Red, 660nm) and an IR (Infra-Red, 880nm) LED, when the user places a finger on top of the sensor the finger absorbs some of the light emitted by the LED'S. The sensor photodetector collects the light that was partially absorbed by the underlying tissue and peripheral blood and returns two values (Oxygen level and Heart Rate) using the I2C protocol and depending on the code that the user has on the microcontroller the sensor can send the heart-rate or oxygen level. [1]

- *AptoFun 3.2”TFT LCD Touch Display (TFT01_3.2):*



Is a 3.2 inches TFT LCD touch screen display with an RGB resolution of 240 pixels by 320pixels ideal for projects that involve Arduino uno, Mega R3, Mega2560 and Mega1280, that need some GUI (graphic user interface), digital photo frame, video terminals, instrumentation, GPS, game consoles, and many other. The LCD module uses his 40 pins to communicate to his adaptor shield (AptoFun TFT 3.2” Shield). It includes a controller, that is a support 16bit data interface. The modulo is design with touch screen controller that is included in the 40 IO pins. Another function that the TFT display has, is a SD Card socket, that uses the SPI mode to operate the SD card. [1]

- *AptoFun TFT 3.2” Shield:*

The TFT 3.2” shield is an adaptor for the AptoFun 3.2”TFT LCD Touch Display so that the LCD modulo connects properly to the Arduino Mega board IO ending in a stack of the 3 boards.

The main part of the project consistes in 6 different devices:

- Arduino Mega;
- Terminal Block Shield Board;
- AptoFun TFT 3.2” Shield;
- AptoFun 3.2”TFT LCD Touch Display;
- MLX90614 Infrared Thermometer;
- MAX30102 Blood Oxygen/Heart-rate.

To assemble the E-Health device starts by stacking the terminal block shield board on top of the Arduino mega aligning the bottom part of the block shield with the long IO pins with the headers of the Arduino mega giving to it more and better accessibility to the ports of the Arduino, comparable to the figure of 1st Step in the Attachments segment of the report. Right after step one is connect the 2 sensors essentials for the E-Health device. First will be the MLX90614 Infrared Thermometer that has 2 input and 2 output ports. With the Infrared sensor facing the up the ports will be located on the lower part of the sensor. From left to right, the pad VIN in the Thermometer sensor will be connected into the top row of the block shield (port number 27 on top row from left to right) with port +5V (between port Digital port 21 and Digital port 22), that will power the sensor with positive 5 volts, has it can be seen in the figure of the 2nd step in red. The 2nd pad from the left of the thermometer sensor called GND will be connected to the ground of the block shield on the top row, port GND (between port AREF and D13), in the figure of the 2nd step is the wired coloured black, establishing a ground for the sensor. Pad SCL its going to attach to SCL port in the block shield in the top row, 1st port from the left (before SDA port), coloured blue in the figure from 2nd step. And the last port in sensor to connect is SDA that connects to SDA port located on the top row of the block shield, 2nd port from the left (between SCL and AREF), the wired painted yellow in the 2nd step figure. And with port VIN and GND the sensor will be able to get power and ground, with port SCL and SDA the sensor capable to send and receive information from the Arduino.

Reference List

Approach and Methodology

Arduino Mega 2560

[1]- https://internetofthingsagenda.techtarget.com/definition/microcontroller

[2]- https://www.mouser.com/catalog/specsheets/ArduinoBoardMega2560.pdf

[3]- https://www.circuito.io/blog/arduino-uno-pinout/

Figure 1 - https://www.antratek.com/arduino-mega-2560

MLX90614 Infrared Thermometer

[1] - https://www.mouser.co.uk/pdfDocs/901090614-706680.pdf

Figure 1 - http://arduinolearning.com/code/mlx90614-infrared-thermometer.php

MAX30102 Blood Oxygen/Heart-rate

[1] - https://www.instructables.com/id/Pulse-Oximeter-With-Much-Improved-Precision/

Figure 1 - https://www.amazon.co.uk/gp/product/B07GKKN4W8/ref=ppx_yo_dt_b_asin_image_o03_s00?ie=UTF8&psc=1

AptoFun 3.2"TFT LCD Touch Display

[1] - https://aptofun.de/3.2_TFT_LCD_Screen_Module#Introduction

Figure 1 – https://aptofun.de/Datei:3.2_TFT_LCD_Screen_Module_1.jpg

AptoFun TFT 3.2" Shield

[1] –

Figure [1] -

Literature Review

[1]- https://www.mdpi.com/2078-2489/10/2/34/htm

[2]- https://www.slideshare.net/moniefeied/ehealth-5196362

[3]- https://apps.who.int/iris/bitstream/handle/10665/252529/9789241511780-eng.pdf?sequence=1

[4]-https://www.researchgate.net/publication/225977285_Knowledge_Management_and_E-Health

[5]- https://consoltech.com/blog/security-threats-healthcare-systems/

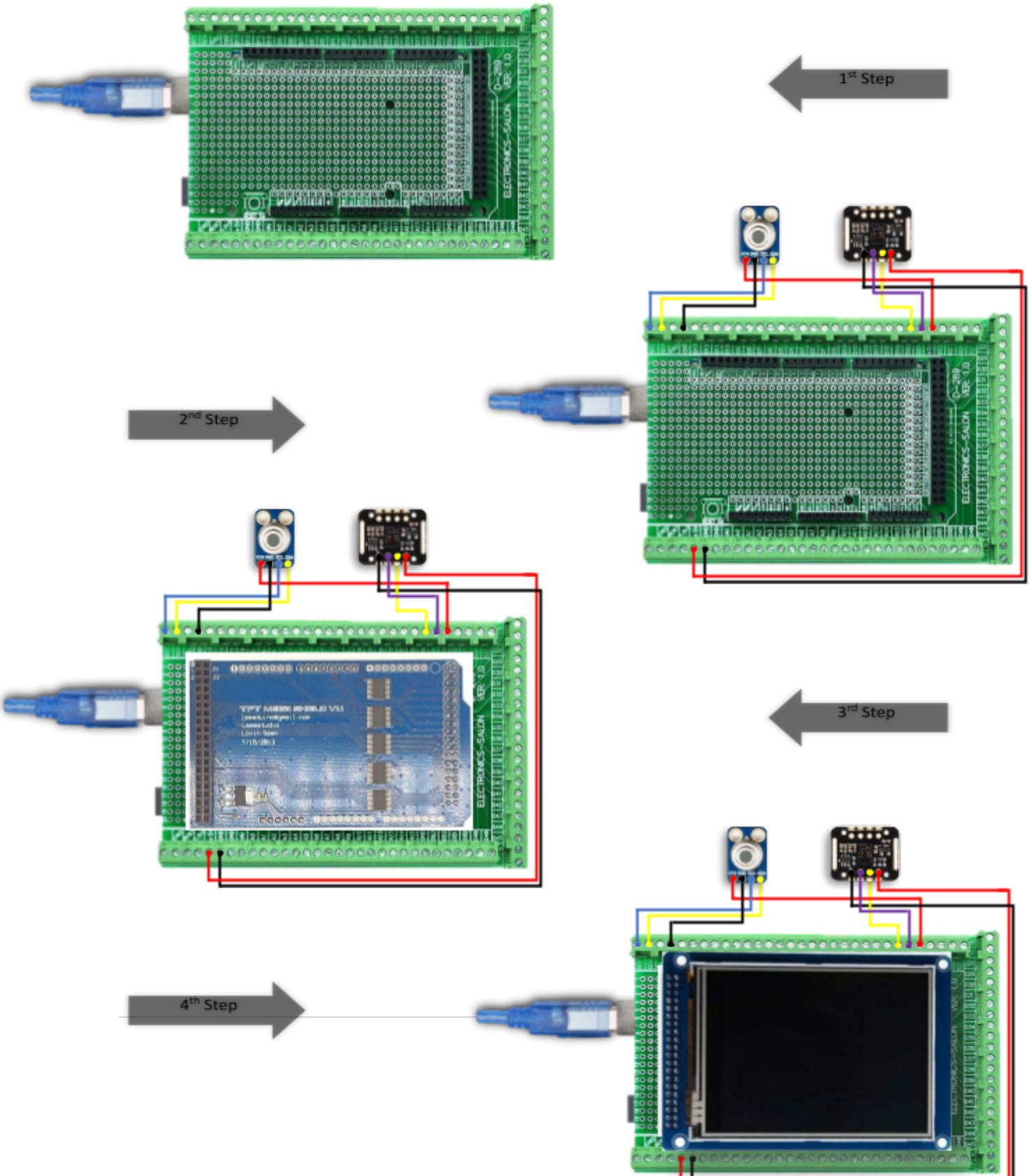
[6]- https://www.telegraph.co.uk/news/2017/05/13/nhs-cyber-attack-everything-need-know-biggest-ransomware-offensive/

[7]- https://www.gamingtechlaw.com/2014/05/top-5-legal-issues-ehealth.html

[8]- https://www.eu-patient.eu/News/News-Archive/The-future-of-eHealth-is-already-here/

Appendices:

E-Health device assembly



Arduino Mega Code (C / C++)

```
/*  Arduino TFT Tutorial
    Program made by Dejan Nedelkovski,
    www.HowToMechatronics.com
*/
/*  This program uses the UTFT and URTouch libraries
    made by Henning Karlsen.
    You can find and download them at:
    www.RinkyDinkElectronics.com
*/
String str;

#include <UTFT.h>
#include <URTouch.h>

//Temperature sensor Library, integers and objects
#include <Wire.h>
#include "Adafruit_MLX90614.h"
Adafruit_MLX90614 mlx = Adafruit_MLX90614();
int c, f, t = 0, g = 0;
int data;

//Sats Sensor Library, integers and Objects
#include "spo2_algorithm.h"
#define MAX_BRIGHTNESS 255
#if defined(__AVR_ATmega328P__) || defined(__AVR_ATmega168__)
//Arduino Uno doesn't have enough SRAM to store 100 samples of IR led
data and red led data in 32-bit format
//To solve this problem, 16-bit MSB of the sampled data will be
truncated. Samples become 16-bit data.
uint16_t irBuffer[100]; //infrared LED sensor data
uint16_t redBuffer[100];  //red LED sensor data
#else
uint32_t irBuffer[100]; //infrared LED sensor data
uint32_t redBuffer[100];  //red LED sensor data
#endif
int32_t bufferLength; //data length
int32_t spo2; //SPO2 value
int8_t validSPO2; //indicator to show if the SPO2 calculation is valid
int32_t heartRate; //heart rate value
int8_t validHeartRate; //indicator to show if the heart rate
calculation is valid
byte pulseLED = 11; //Must be on PWM pin
byte readLED = 13; //Blinks with each data read

//Heart rate Sensor Library, integers and objects
#include "MAX30105.h"
#include "heartRate.h"
MAX30105 particleSensor;
const byte RATE_SIZE = 4; //Increase this for more averaging. 4 is
good.
byte rates[RATE_SIZE]; //Array of heart rates
byte rateSpot = 0;
long lastBeat = 0; //Time at which the last beat occurred
```

```
float beatsPerMinute;
int beatAvg;



//==== Creating Objects
UTFT    myGLCD(ILI9341_16, 38, 39, 40, 41); //Parameters should be
adjusted to your Display/Schield model
URTouch  myTouch( 6, 5, 4, 3, 2);
//==== Defining Variables
extern uint8_t SmallFont[];
extern uint8_t BigFont[];
extern uint8_t SevenSegNumFont[];
extern uint8_t franklingothic_normal[];
int patienteiD = 1;

int x, y;
char currentPage, selectedUnit;
//Ultrasonic Sensor
const int VCC = 13;
const int trigPin = 11;
const int echoPin = 12;
long duration;

// RGB LEDs
const int redLed = 10;
const int greenLed = 9;
const int blueLed = 8;
int xR = 38;
int xG = 38;
int xB = 38;

void setup() {
  // Initial setup
  myGLCD.InitLCD();
  myGLCD.clrScr();
  myTouch.InitTouch();
  myTouch.setPrecision(PREC_HI);
  // Defining Pin Modes
  pinMode(VCC, OUTPUT); // VCC
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  digitalWrite(VCC, HIGH); // +5V - Pin 13 as VCC


  drawHomeScreen();  // Draws the Home Screen
  currentPage = '0'; // Indicates that we are at Home Screen
  selectedUnit = '0'; // Indicates the selected unit for the first
example, cms or inches

  //Setup for Temperature Sensor;
  mlx.begin();

  //Setup for HeartRate Sensor;
  Serial.begin(115200);
  particleSensor.begin();  // Initialize sensor
  particleSensor.setup(); //Configure sensor with default settings
```

```
  particleSensor.setPulseAmplitudeRed(0x0A); //Turn Red LED to low to
indicate sensor is running
  particleSensor.setPulseAmplitudeGreen(0); //Turn off Green LED
}
void loop() {
  // Home Screen
  if (currentPage == '0') {
    if (myTouch.dataAvailable()) {
      myTouch.read();
      x = myTouch.getX(); // X coordinate where the screen has been
pressed
      y = myTouch.getY(); // Y coordinates where the screen has been
pressed
      // If we press the Temperature Button
      if ((x >= 0) && (x <= 155) && (y >= 90) && (y <= 130)) {
        drawFrame(0, 90, 155, 130); // Custom Function -Highlighs the
buttons when it's pressed
        currentPage = '1'; // Indicates that we are the first example
        myGLCD.clrScr(); // Clears the screen
        drawTemperatureSensor(); // It is called only once, because in
the next iteration of the loop, this above if statement will be false
so this funtion won't be called. This function will draw the graphics
of the first example.
      }
      // If we press the Oxygen Saturation Button
      if ((x >= 165) && (x <= 319) && (y >= 90) && (y <= 130)) {
        drawFrame(165, 90, 319, 130);
        //Setup for Spo2
        pinMode(pulseLED, OUTPUT);
        pinMode(readLED, OUTPUT);
        byte ledBrightness = 60; //Options: 0=Off to 255=50mA
        byte sampleAverage = 4; //Options: 1, 2, 4, 8, 16, 32
        byte ledMode = 2; //Options: 1 = Red only, 2 = Red + IR, 3 =
Red + IR + Green
        byte sampleRate = 100; //Options: 50, 100, 200, 400, 800, 1000,
1600, 3200
        int pulseWidth = 411; //Options: 69, 118, 215, 411
        int adcRange = 4096; //Options: 2048, 4096, 8192, 16384
        particleSensor.setup(ledBrightness, sampleAverage, ledMode,
sampleRate, pulseWidth, adcRange); //Configure sensor with these
settings
        currentPage = '2';
        myGLCD.clrScr();
        drawSats();
      }
      // If we press the Heart Rate Button
      if ((x >= 35) && (x <= 285) && (y >= 140) && (y <= 180)) {
        drawFrame(35, 140, 285, 180);
        currentPage = '3';
        myGLCD.clrScr();
        myGLCD.setColor(114, 198, 206);
        myGLCD.fillRect(0, 0, 319, 239);
        drawHeartRate();
      }
      //If we press the Blood Pressure Button
      if ((x >= 35) && (x <= 285) && (y >= 190) && (y <= 230)) {
        drawFrame(35, 190, 285, 230);
```

```
        currentPage = '4';
        myGLCD.clrScr();
        myGLCD.setColor(114, 198, 206);
        myGLCD.fillRect(0, 0, 319, 239);
        drawBloodPressure();
      }
    }
  }
  // Temperature
  if (currentPage == '1') {
    getTemperature(); // Gets distance from the sensor and this
function is repeatedly called while we are at the first example in
order to print the lasest results from the distance sensor
    if (myTouch.dataAvailable()) {
      myTouch.read();
      x = myTouch.getX();
      y = myTouch.getY();

      // If we press the Centimeters Button
      if ((x >= 10) && (x <= 135) && (y >= 90) && (y <= 163)) {
        drawFrame(10, 135, 90, 163);
        selectedUnit = '0';
      }
      // If we press the Inches Button
      if ((x >= 10) && (x <= 90) && (y >= 173) && (y <= 201)) {
        drawFrame(10, 173, 90, 201);
        selectedUnit = '1';
      }
      // If we press the Back Button
      if ((x >= 10) && (x <= 60) && (y >= 10) && (y <= 36)) {
        drawFrame(10, 10, 60, 36);
        currentPage = '0'; // Indicates we are at home screen
        myGLCD.clrScr();
        drawHomeScreen(); // Draws the home screen
      }
    }
  }
  // Sats
  if (currentPage == '2') {
    setSats();
    if (myTouch.dataAvailable()) {
      myTouch.read();
      x = myTouch.getX();
      y = myTouch.getY();

      //Back button
      if ((x >= 10) && (x <= 60) && (y >= 10) && (y <= 36)) {
        drawFrame(10, 10, 60, 36);
        currentPage = '0';
        myGLCD.clrScr();
        drawHomeScreen();
      }
    }
  }
  // Heart Rate
  if (currentPage == '3') {
    setHeartRate();
```

```
    if (myTouch.dataAvailable()) {
      myTouch.read();
      x = myTouch.getX();
      y = myTouch.getY();

      //Back button
      if ((x >= 10) && (x <= 60) && (y >= 10) && (y <= 36)) {
        drawFrame(10, 10, 60, 36);
        currentPage = '0';
        myGLCD.clrScr();
        drawHomeScreen();
      }
    }
  }
  // Blood Pressure
  if (currentPage == '4') {
    if (myTouch.dataAvailable()) {
      myTouch.read();
      x = myTouch.getX();
      y = myTouch.getY();

      //Back button
      if ((x >= 10) && (x <= 60) && (y >= 10) && (y <= 36)) {
        drawFrame(10, 10, 60, 36);
        currentPage = '0';
        myGLCD.clrScr();
        drawHomeScreen();
      }
    }
  }
}
// ====== Custom Funtions ======
// drawHomeScreen - Custom Function
void drawHomeScreen() {
  // Title
  if (t >= 300 || g >= 300) {
    t = 0;
    g = 0;
  }
  myGLCD.fillScr(44, 47, 51);
  myGLCD.setBackColor(44, 47, 51); // Sets the background color of the
area where the text will be printed to black
  myGLCD.setColor(255, 255, 255); // Sets color to white
  myGLCD.setFont(franklingothic_normal); // Sets font to big
  myGLCD.print("E-Health Monitoring", CENTER, 10); // Prints the string
on the screen
  myGLCD.setColor(255, 0, 0); // Sets color to red
  myGLCD.drawLine(0, 32, 319, 32); // Draws the red line
  myGLCD.setColor(255, 255, 255); // Sets color to white
  myGLCD.setFont(SmallFont); // Sets the font to small
  myGLCD.print("by: Hugo Martins", CENTER, 55 ); // Prints the string


  // Button - Temperature
  myGLCD.setColor(16, 167, 103); // Sets green color
  myGLCD.fillRoundRect (0, 90, 155, 130); // Draws filled rounded
rectangle
```

```
  myGLCD.setColor(255, 255, 255); // Sets color to white
  myGLCD.drawRoundRect (0, 90, 155, 130); // Draws rounded rectangle
without a fill, so the overall appearance of the button looks like it
has a frame
  myGLCD.setFont(franklingothic_normal);// Sets the font to big
  myGLCD.setBackColor(16, 167, 103); // Sets the background color of
the area where the text will be printed to green, same as the button
  myGLCD.print("   Temp", LEFT + 1, 102); // Prints the string

  // Button - Blood Oxygen
  myGLCD.setColor(16, 167, 103); // Sets green color
  myGLCD.fillRoundRect (165, 90, 319, 130); // Draws filled rounded
rectangle
  myGLCD.setColor(255, 255, 255); // Sets color to white
  myGLCD.drawRoundRect (165, 90, 319, 130); // Draws rounded rectangle
without a fill, so the overall appearance of the button looks like it
has a frame
  myGLCD.setFont(franklingothic_normal); // Sets the font to big
  myGLCD.setBackColor(16, 167, 103); // Sets the background color of
the area where the text will be printed to green, same as the button
  myGLCD.print("  Sats", LEFT + 180, 102); // Prints the string

  // Button - Heart Rate
  myGLCD.setColor(16, 167, 103); // Sets green color
  myGLCD.fillRoundRect (35, 140, 285, 180); // Draws filled rounded
rectangle
  myGLCD.setColor(255, 255, 255); // Sets color to white
  myGLCD.drawRoundRect (35, 140, 285, 180); // Draws rounded rectangle
without a fill, so the overall appearance of the button looks like it
has a frame
  myGLCD.setFont(franklingothic_normal); // Sets the font to new font
  myGLCD.setBackColor(16, 167, 103); // Sets the background color of
the area where the text will be printed to green, same as the button
  myGLCD.print("Heart Rate", CENTER, 152); // Prints the string

  // Button - Blood Pressure
  myGLCD.setColor(16, 167, 103); // Sets green color
  myGLCD.fillRoundRect (35, 190, 285, 230); // Draws filled rounded
rectangle
  myGLCD.setColor(255, 255, 255);// Sets color to white
  myGLCD.drawRoundRect (35, 190, 285, 230);// Draws rounded rectangle
without a fill, so the overall appearance of the button looks like it
has a frame
  myGLCD.setFont(franklingothic_normal); // Sets the font to new font
  myGLCD.setBackColor(16, 167, 103); // Sets the background color of
the area where the text will be printed to green, same as the button
  myGLCD.print("Blood Pressure", CENTER, 202); // Prints the string

}
// Highlights the button when pressed
void drawFrame(int x1, int y1, int x2, int y2) {
  myGLCD.setColor(255, 0, 0);
  myGLCD.drawRoundRect (x1, y1, x2, y2);
  while (myTouch.dataAvailable())
    myTouch.read();
  myGLCD.setColor(255, 255, 255);
  myGLCD.drawRoundRect (x1, y1, x2, y2);
```

```
}
//================================================================================
==================================
void drawTemperatureSensor() {
  myGLCD.setColor(100, 155, 203);
  myGLCD.fillRoundRect (10, 10, 60, 36);
  myGLCD.setColor(255, 255, 255);
  myGLCD.drawRoundRect (10, 10, 60, 36);
  myGLCD.setFont(franklingothic_normal);
  myGLCD.setBackColor(100, 155, 203);
  myGLCD.print("<-", 18, 15);
  myGLCD.setBackColor(0, 0, 0);
  myGLCD.setFont(SmallFont);
  myGLCD.print("Back to Main Menu", 70, 18);
  myGLCD.setFont(franklingothic_normal);
  myGLCD.print("Temperature", CENTER, 50);
  myGLCD.print("", CENTER, 76); //!!!!!!!!!!!!!!!!!!!Change after for
sensor color!!!!!!!!!!!!!!!!!!
  myGLCD.setColor(255, 0, 0);
  myGLCD.drawLine(0, 100, 319, 100);
  myGLCD.setBackColor(0, 0, 0);
  myGLCD.setColor(255, 255, 255);
  myGLCD.setFont(SmallFont);
  myGLCD.print("Select Unit", 10, 114);
  myGLCD.setFont(franklingothic_normal);
  myGLCD.print("Temperature:", 120, 120);
  myGLCD.setColor(223, 77, 55);
  myGLCD.fillRoundRect (10, 135, 90, 163);
  myGLCD.setColor(225, 255, 255);
  myGLCD.drawRoundRect (10, 135, 90, 163);
  myGLCD.setBackColor(223, 77, 55);
  myGLCD.setColor(255, 255, 255);
  myGLCD.print("`C", 33, 140);
  myGLCD.setColor(223, 77, 55);
  myGLCD.fillRoundRect (10, 173, 90, 201);
  myGLCD.setColor(255, 255, 255);
  myGLCD.drawRoundRect (10, 173, 90, 201);
  myGLCD.setBackColor(223, 77, 55);
  myGLCD.setColor(255, 255, 255);
  myGLCD.print("`F", 33, 180);
  myGLCD.setBackColor(0, 0, 0);
  myGLCD.setFont(SmallFont);
  myGLCD.print("Source code at: HowToMechatronics.com", CENTER, 220);
}
//================================================
//===== getTemperature - Custom Function
void getTemperature() {
  c = mlx.readObjectTempC();
  f = mlx.readObjectTempF();
  // Prints the temperature in centigrade
  if (selectedUnit == '0' && c <= 400 && t < 300) {
    myGLCD.setFont(SevenSegNumFont);
    myGLCD.setColor(0, 255, 0);
    myGLCD.setBackColor(0, 0, 0);
    myGLCD.printNumI(c, 130, 145, 3, '0');
    myGLCD.setFont(franklingothic_normal);
    myGLCD.print("`C  ", 235, 178);
```

```
    t++;

    Serial.println(c);
  }
  // Prints the temperature in fahrenheit
  if (selectedUnit == '1' && f <= 160 && g < 300) {
    myGLCD.setFont(SevenSegNumFont);
    myGLCD.setColor(0, 255, 0);
    myGLCD.setBackColor(0, 0, 0);
    myGLCD.printNumI(f, 130, 145, 3, '0');
    myGLCD.setFont(franklingothic_normal);
    myGLCD.print("`F", 235, 178);
    g++;
    Serial.println(f);
  }
}
//====================================================================
==================================
void drawSats() {
  myGLCD.setColor(100, 155, 203);
  myGLCD.fillRoundRect (10, 10, 60, 36);
  myGLCD.setColor(255, 255, 255);
  myGLCD.drawRoundRect (10, 10, 60, 36);
  myGLCD.setFont(franklingothic_normal);
  myGLCD.setBackColor(100, 155, 203);
  myGLCD.print("<-", 18, 15);
  myGLCD.setBackColor(0, 0, 0);
  myGLCD.setFont(SmallFont);
  myGLCD.print("Back to Main Menu", 70, 18);
  myGLCD.setFont(franklingothic_normal);
  myGLCD.print("Blood Saturation", CENTER, 60);
  myGLCD.print("", CENTER, 76); //!!!!!!!!!!!!!!!!!!!Change after for
sensor color!!!!!!!!!!!!!!!!!!
  myGLCD.setColor(255, 0, 0);
  myGLCD.drawLine(0, 100, 319, 100);
  myGLCD.setBackColor(0, 0, 0);
  myGLCD.setColor(255, 255, 255);
  myGLCD.setFont(SmallFont);
  myGLCD.print("Source code at: HowToMechatronics.com", CENTER, 220);
}
//============= setLedColor()
void setSats() {
  bufferLength = 100; //buffer length of 100 stores 4 seconds of
samples running at 25sps

  //read the first 100 samples, and determine the signal range
  for (byte i = 0 ; i < bufferLength ; i++)
  {
    while (particleSensor.available() == false) //do we have new data?
      particleSensor.check(); //Check the sensor for new data

    redBuffer[i] = particleSensor.getRed();
    irBuffer[i] = particleSensor.getIR();
    particleSensor.nextSample(); //We're finished with this sample so
move to next sample
  }
```

```
  //calculate heart rate and SpO2 after first 100 samples (first 4
seconds of samples)
  maxim_heart_rate_and_oxygen_saturation(irBuffer, bufferLength,
redBuffer, &spo2, &validSPO2, &heartRate, &validHeartRate);

  //Continuously taking samples from MAX30102.  Heart rate and SpO2 are
calculated every 1 second
  while (1)
  {
    //dumping the first 25 sets of samples in the memory and shift the
last 75 sets of samples to the top
    for (byte i = 25; i < 100; i++)
    {
      redBuffer[i - 25] = redBuffer[i];
      irBuffer[i - 25] = irBuffer[i];
    }
    //take 25 sets of samples before calculating the heart rate.
    for (byte i = 75; i < 100; i++)
    {
      while (particleSensor.available() == false) //do we have new
data?
        particleSensor.check(); //Check the sensor for new data

      digitalWrite(readLED, !digitalRead(readLED)); //Blink onboard LED
with every data read

      redBuffer[i] = particleSensor.getRed();
      irBuffer[i] = particleSensor.getIR();
      particleSensor.nextSample(); //We're finished with this sample so
move to next sample

      //send samples and calculation result to terminal program through
UART
      myGLCD.setFont(SevenSegNumFont);
      myGLCD.setColor(0, 255, 0);
      myGLCD.setBackColor(0, 0, 0);
      myGLCD.printNumI(spo2, 130, 145, 3, '0');
      myGLCD.setFont(franklingothic_normal);
      myGLCD.print("%", 235, 178);
      Serial.println(spo2);
    }
    //After gathering 25 new samples recalculate HR and SP02
    maxim_heart_rate_and_oxygen_saturation(irBuffer, bufferLength,
redBuffer, &spo2, &validSPO2, &heartRate, &validHeartRate);
  }
}

//=====================================================================
==================================
void drawHeartRate() {
  myGLCD.setColor(100, 155, 203);
  myGLCD.fillRoundRect (10, 10, 60, 36);
  myGLCD.setColor(255, 255, 255);
  myGLCD.drawRoundRect (10, 10, 60, 36);
  myGLCD.setFont(franklingothic_normal);
  myGLCD.setBackColor(100, 155, 203);
  myGLCD.print("<-", 18, 15);
```

```
  myGLCD.setBackColor(0, 0, 0);
  myGLCD.setFont(SmallFont);
  myGLCD.print("Back to Main Menu", 70, 18);
  myGLCD.setFont(franklingothic_normal);
  myGLCD.print("Heart Rate", CENTER, 60);
  myGLCD.print("", CENTER, 76); //!!!!!!!!!!!!!!!!!!!Change after for
sensor color!!!!!!!!!!!!!!!!!!
  myGLCD.setColor(255, 0, 0);
  myGLCD.drawLine(0, 100, 319, 100);
  myGLCD.setBackColor(0, 0, 0);
  myGLCD.setColor(255, 255, 255);
  myGLCD.setFont(SmallFont);
  myGLCD.print("Source code at: HowToMechatronics.com", CENTER, 220);
}
//============= setHeartRate()
void setHeartRate() {
  long irValue = particleSensor.getIR();
  if (checkForBeat(irValue) == true) {
    //We sensed a beat!
    long delta = millis() - lastBeat;
    lastBeat = millis();
    beatsPerMinute = 60 / (delta / 1000.0);
    if (beatsPerMinute < 255 && beatsPerMinute > 20) {
      rates[rateSpot++] = (byte)beatsPerMinute; //Store this reading in
the array
      rateSpot %= RATE_SIZE; //Wrap variable
      //Take average of readings
      beatAvg = 0;
      for (byte x = 0 ; x < RATE_SIZE ; x++)
        beatAvg += rates[x];
      beatAvg /= RATE_SIZE;
      myGLCD.setFont(SevenSegNumFont);
      myGLCD.setColor(0, 255, 0);
      myGLCD.setBackColor(0, 0, 0);
      myGLCD.printNumI(beatAvg, 130, 145, 3, '0');
      myGLCD.setFont(franklingothic_normal);
      myGLCD.print("BPM  ", 235, 178);
      Serial.println(beatAvg);
    }
  }
}
//=====================================================
void drawBloodPressure() {
  myGLCD.setColor(100, 155, 203);
  myGLCD.fillRoundRect (10, 10, 60, 36);
  myGLCD.setColor(255, 255, 255);
  myGLCD.drawRoundRect (10, 10, 60, 36);
  myGLCD.setFont(franklingothic_normal);
  myGLCD.setBackColor(100, 155, 203);
  myGLCD.print("<-", 18, 15);
  myGLCD.setBackColor(0, 0, 0);
  myGLCD.setFont(SmallFont);
  myGLCD.print("Back to Main Menu", 70, 18);
  myGLCD.setFont(franklingothic_normal);
  myGLCD.print("Blood Pressure", CENTER, 50);
}
```

Website Index Page (HTML)

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>E-Health</title>
        <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
        <link rel="icon" href="Logo.png"/>
        <style>
            body{
                background-color: lightgray;
            }
            .login-page{
                position: center;
                padding-top: 50px;
            }
            .form{
                position: flex;
                z-index: 1;
                background: #ffffff;
                max-width: 360px;
                padding: 35px;
                margin: auto;
                text-align: center;
            }
            .form input{
```

```
    font-family: "Roboto", sans-serif;
    outline: 1;
    background: #f2f2f2;
    width: 100%;
    border: 0;
    margin: 0 0 15px;
    padding: 15px;
    box-sizing: border-box;
    font-size: 14px;
}
.form button{
    cursor: pointer;
    font-family: "Roboto", sans-serif;
    text-decoration: uppercase;
    outline: 0;
    background: #2C2F33;
    border: 0px;
    padding: 15px;
    color: #ffffff;
    font-size: 14px;
}
.form button:hover,.form button:active{
    background: grey;
}
.form .message{
    font-family: "Roboto", sans-serif;
    margin: 15px 0 0;
    color: lightblack;
    fonr-size: 12px;
}
.form .message a{
    font-family: "Roboto", sans-serif;
    color: lightblack;
```

```
                text-decoration: none;
            }
            .form .register-form{
                display: none;
            }
            .imglogo{
                display: block;
                margin-left: auto;
                margin-right: auto;
            }
            .footer{
                position: fixed;
                left: 0;
                bottom: 0;
                width: 100%;
                background-color: red;
                color: white;
                text-align: center;
                font-family: "Roboto", sans-serif;
            }
        </style>
    </head>
    <body>
        <img src="Logo.png" class='imglogo' alt="Logo"/>
        <div class="login-page">
            <div class="form">
                <form class="register-form" action="Register_Page"
method="Post">
                    <input type="text" id='name' name="name_register"
placeholder="Full Name" required='true'/>
                    <input type="text" id='email' name="email_register"
placeholder="email" required='true'/>
                    <input type="password" id='password'
name="password_register" placeholder="password" required ='true'/>
```

```
                    <input type="text" id='DocID' name="DocID_register"
placeholder="Doctor ID" required ='true'/>

                    <button>Create</button>

                    <p class="message">Already Registered? <a
href="#">Login</a>

                    </p>

                </form>

                <form class="login-form" action="Login_Page"
method="Post">

                    <input type="text" id='DocID' name="DocID_login"
placeholder="Doctor ID" required='true'/>

                    <input type="password" id='password'
name="password_login" placeholder="password" required='true'/>

                    <button>Login</button>

                    <p class="message">Not Registered? <a
href="#">Register</a>

                    </p>

                </form>

            </div>

        </div>

        <script
src='https://code.jquery.com/jquery-3.2.1.min.js'></script>


        <script>

            $('.message a').click(function () {

                $('form').animate({height: "toggle", opacity:
"toggle"}, "slow");

            });

        </script>


        <footer class="footer">

            <p>Done by: Hugo Martins</p>

            <p>All Rights Reserved &#169;</p>

        </footer>

    </body>

</html>
```

Login_Page (Java)

```
/*
 * To change this license header, choose License Headers in Project
Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.annotation.WebServlet;
/**
 *
 * @author hugom
 */
@WebServlet(urlPatterns = {"/Login_Page"})
public class Login_Page extends HttpServlet {
    /**
     * Processes requests for both HTTP <code>GET</code> and
<code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        String DocID = request.getParameter("DocID_login");
        String pass = request.getParameter("password_login");
```

```
            // Connect to mysql and verify username password


        if(Validate.checkUser(DocID, pass))
        {
            RequestDispatcher rs =
request.getRequestDispatcher("Dashboard.html");
            rs.forward(request, response);
        }
        else
        {
           RequestDispatcher rs =
request.getRequestDispatcher("index.xhtml");
           rs.include(request, response);
        }
    }
}
```

Reagister_Page (Java)

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
/**
 *
 * @author hugom
 */
public class Register_Page extends HttpServlet {
    private PreparedStatement ps;

    /**
     *
     * @throws ServletException
     */
    @Override
    public void init() throws ServletException {
        initJDBC();
    }
    @Override
    protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            String Name = request.getParameter("name_register");
```

```
            String Email = request.getParameter("email_register");
            String DocID = request.getParameter("DocID_register");
            String pass = request.getParameter("password_register");
            try {
                addUser(DocID, Name, Email, pass);

response.setStatus(HttpServletResponse.SC_MOVED_TEMPORARILY);
                response.setHeader("Location", "Dashboard.html");
            } catch (SQLException ex) {
                out.println("not sended");
            }
        }
    }
    public void initJDBC() {
        try {
            Class.forName("org.apache.derby.jdbc.ClientDriver");
            Connection conn =
DriverManager.getConnection("jdbc:derby://localhost:1527/E-Health",
"root", "1234");
            ps = conn.prepareStatement("insert into DOCS
(DOCID,DOCNAME,DOCEMAIL,DOCPASS) values(?,?,?,?)");
        } catch (ClassNotFoundException | SQLException ex) {
            System.out.println(ex.getMessage());
        }
    }
    public void addUser(String DOCID, String DOCNAME, String DOCEMAIL,
String DOCPASS) throws SQLException {
        ps.setString(1, DOCID);
        ps.setString(2, DOCNAME);
        ps.setString(3, DOCEMAIL);
        ps.setString(4, DOCPASS);
        ps.executeUpdate();
    }
}
```

Validation (Java)

```
import java.sql.*;


public class Validate {

    public static boolean checkUser(String DOCID,String DOCPASS)

    {

        boolean st =false;

        try {


            //loading drivers for mysql

            Class.forName("com.mysql.jdbc.Driver");


            //creating connection with the database

            Connection con =
DriverManager.getConnection("jdbc:derby://localhost:1527/E-Health",
"root", "1234");

            PreparedStatement ps = con.prepareStatement("select * from
DOCS where DOCID=? and DOCPASS=?");

            ps.setString(1, DOCID);

            ps.setString(2, DOCPASS);

            ResultSet rs =ps.executeQuery();

            st = rs.next();


        }

        catch(Exception e) {

            e.printStackTrace();

        }

        return st;

    }

}
```

Dashboard (HTML)

```
<!DOCTYPE html>
<html>
    <head>
        <title>User Vital Sign</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width,
initial-scale=1.0">
        <link rel="icon" href="Logo.png"/>
        <link rel="stylesheet" href="WEB-INF/CSS _Style_Sheet.css"
type="text/css">

        <style>
            body{
                background-color: gray;
            }
        </style>
        <script>
            function Serial_Read() {
                var xhttp = new XMLHttpRequest();
                xhttp.onreadystatechange = function () {
                    if (this.readyState === 4 && this.status === 200) {

                        document.getElementById("details").innerHTML =
this.responseText;
                    }
                };
                xhttp.open("GET", "Read_From_Serial_Read_Method",
true);
                xhttp.send();
            }
        </script>
        <script src="raphael.2.1.0.min.js"></script>
        <script src="justgage.1.0.1.min.js"></script>
```

```
    </head>
    <body>
        <div class="gauge">
            <div class="flexbox">
                <div class="box">
                    <div id="g1" class="gauge"></div>
                </div>
                <div class="box">
                    <div id="g2" class="gauge"></div>
                </div>
                <div class="box">
                    <div id="g3" class="gauge"></div>
                </div>
                <div class="box">
                    <div id="g4" class="gauge"></div>
                </div>
            </div>
        </div>
        <p id="details"></p>
        <script>
            //call function Serial_Read() after page loaded
            window.onload = Serial_Read;
        </script>
    </body>
</html>
```

Read_From_Serial_Read_Method(Java/HTML)

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
/**
 *
 * @author hugom
 */
@WebServlet(urlPatterns = {"/Read_From_Serial_Read_Method"})
public class Read_From_Serial_Read_Method extends HttpServlet {
    /**
     * Processes requests for both HTTP <code>GET</code> and
<code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample
code. */
            out.println("<!DOCTYPE html>");
            out.println("<html>");
```

```
            out.println("<head>");
            out.println("<title>Servlet
Read_From_Serial_Read_Method</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Servlet Read_From_Serial_Read_Method at "
+ request.getContextPath() + "</h1>");
            out.println("</body>");
            out.println("</html>");
        }
    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods.
Click on the + sign on the left to edit the code.">
    /**
     * Handles the HTTP <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Handles the HTTP <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
```

```
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }
    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>
}
```