Health Care System Using Arduino

By

Dipasha Chokeda 2040375

Hitesh Salimath 2040365

Thulasi V 2040354

Under the guidance of

Prof. Johnson O V

Department of Physics and Electronics

A project report submitted in partial fulfillment for the award of degree of B.Sc (Electronics) of Christ (Deemed to be University), Bengaluru.

www.christuniversity.in

April 2023

Acknowledgment

We thank the god almighty for enabling us to complete our project by blessing us with his wisdom without which we would not have been able to complete it.

We express our deep sense of gratitude to Vice Chancellor Rev Fr. Jose CC, Christ (Deemed to be University) for providing us with the necessary facilities that helped us in accomplishing our project.

We deeply thank Pro-Vice Chancellor, Christ(Deemed to be University) for his moral support. We would like to thank the management and the Head of the Department of Physics and Electronics,Dr.Manoj B, for giving us an opportunity to make up this project.

On the successful completion of this project, we take this opportunity to express our gratitude and regards to our guides Prof.Johnson O V and Ms Raksha.S for their guidance, monitoring and constant encouragement throughout the project.

We want to thank our friends who were really helpful and critical in evaluating and giving constructive feedback which helped us improve our project. We would also like to appreciate the guidance given by the teachers during our project presentation has improved our presentation skills, thank you to their comments and advise. Finally we thank all our classmates, supporting faculty members and families for extending their support for our development of the project.

Abstract

we propose a smart health care system using Arduino, an open-source hardware and software platform, to monitor and manage health conditions in real-time. The system integrates various sensors and actuators, such as heart rate sensor, temperature sensor, and SpO_2(Oxygen Saturation) level to collect health data and provide timely interventions. which will be displayed on the phone screen on the app which is also made by us.This project hence will help them to get the basics checked in their house without much effort.This will save time as well as money. This project is designed to solve this problem.The user should just connect his phone , open our app and start monitoring his readings. If something seems unusual then the person can go to the hospital to seek professional help.This project is not 100 percent but 80 percent accurate. The smart health care system using Arduino has the potential to revolutionize the way health care is delivered, providing real-time monitoring and management of health conditions, reducing hospitalization rates, and improving patient outcomes.

Contents

Chapter 1

Introduction

Health care systems around the world are facing significant challenges, including rising costs, increasing prevalence of chronic diseases, and an aging population. With the advent of Internet of Things (IoT) and wearable devices, there is a growing interest in leveraging these technologies to develop smart health care systems that can provide personalized, remote, and continuous monitoring and management of health conditions. Arduino, an open-source hardware and software platform, offers a flexible and cost-effective solution for building such smart health care systems.

The Arduino-based smart health care system integrates various sensors, such as heart rate sensors, temperature sensors,and SpO_2 to collect real-time health data from patients. Patients can also access their health data and receive personalized recommendations for lifestyle changes through a mobile application.

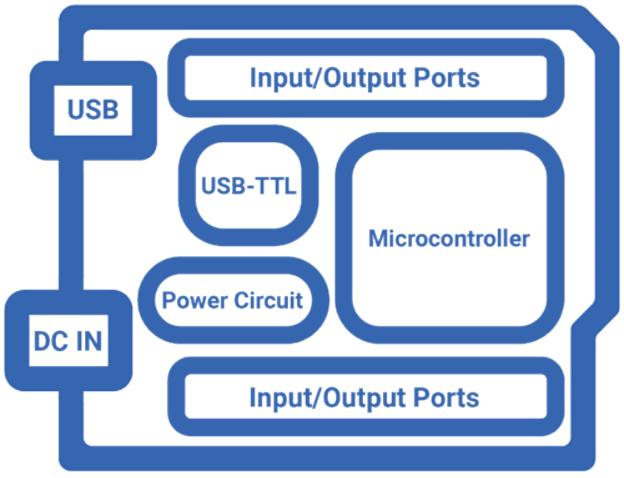
The smart health care system using Arduino has the potential to revolutionize the traditional health care model by enabling remote monitoring, proactive health management, and improved patient outcomes. This paper presents an overview of the smart health care system using Arduino, highlighting its features, potential benefits, and future directions for research and development.

Chapter 2

Details of the Circuit

2.1 Arduino UNO

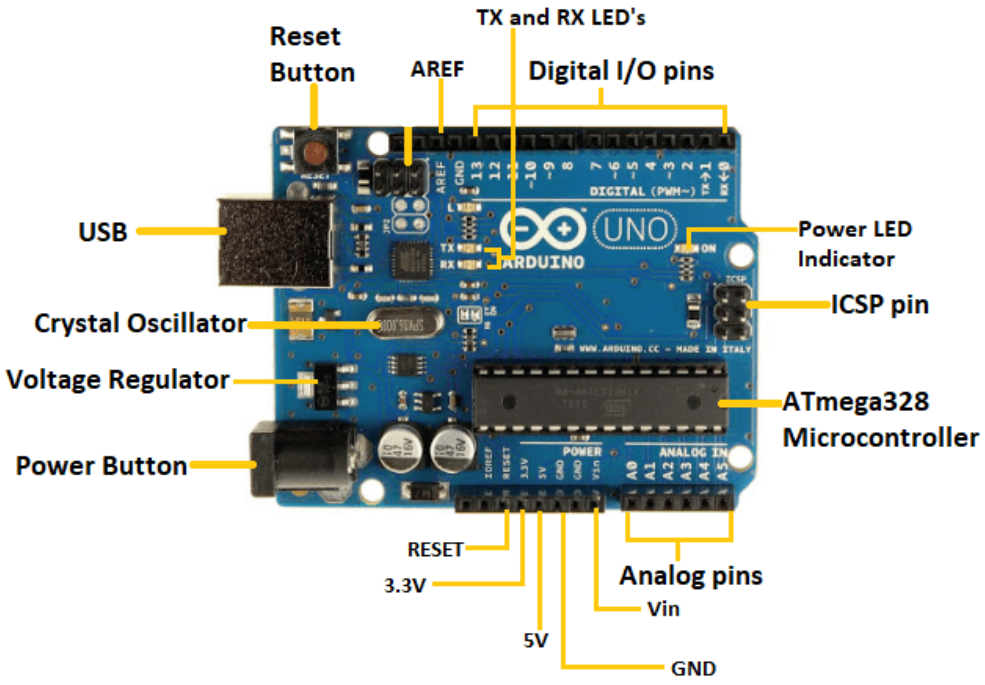
Arduino UNO is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.
The block diagram is as follows:



Arduino UNO is a low-cost, flexible, and easy-to-use programmable open-source microcontroller

board that can be integrated into a variety of electronic projects. This board can be interfaced with other Arduino boards, Arduino shields, Raspberry Pi boards and can control relays, LEDs, servos, and motors as an output.

The Arduino UNO includes 6 analog pin inputs, 14 digital pins, a USB connector, a power jack, and an ICSP (In-Circuit Serial Programming) header. It is programmed based on IDE, which stands for Integrated Development Environment. It can run on both online and offline platforms.



- ATmega328 Microcontroller- It is a single chip Microcontroller of the ATmel family. The processor code inside it is of 8-bit. It combines Memory (SRAM, EEPROM, and Flash), Analog to Digital Converter, SPI serial ports, I/O lines, registers, timer, external and internal interrupts, and oscillator.

- ICSP pin - The In-Circuit Serial Programming pin allows the user to program using the firmware of the Arduino board.

- Power LED Indicator- The ON status of LED shows the power is activated. When the

power is OFF, the LED will not light up.

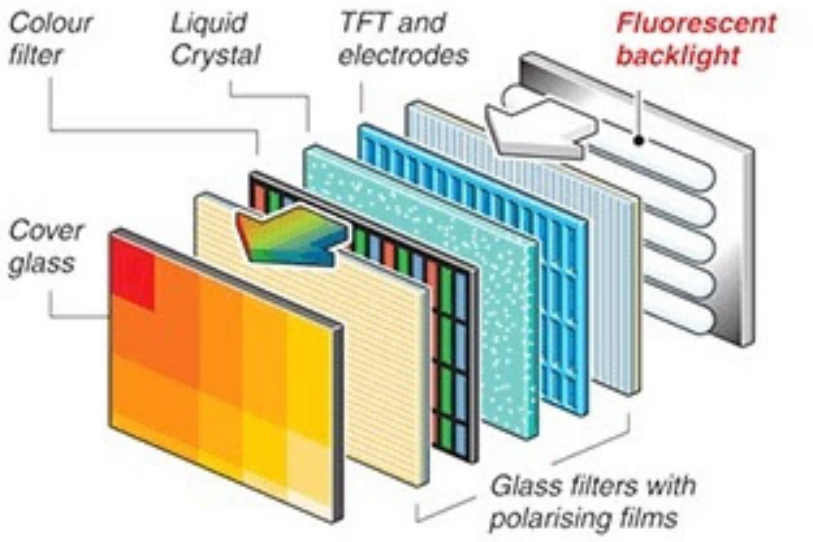
- Digital I/O pins- The digital pins have the value HIGH or LOW. The pins numbered from D0 to D13 are digital pins.
- TX and RX LED's- The successful flow of data is represented by the lighting of these LED's.
- AREF- The Analog Reference (AREF) pin is used to feed a reference voltage to the Arduino UNO board from the external power supply.
- Reset button- It is used to add a Reset button to the connection.
- USB- It allows the board to connect to the computer. It is essential for the programming of the Arduino UNO board.
- Crystal Oscillator- The Crystal oscillator has a frequency of 16MHz, which makes the Arduino UNO a powerful board.
- Voltage Regulator- The voltage regulator converts the input voltage to 5V.
- GND- Ground pins. The ground pin acts as a pin with zero voltage.
- Vin- It is the input voltage.
- Analog Pins- The pins numbered from A0 to A5 are analog pins. The function of Analog pins is to read the analog sensor used in the connection. It can also act as GPIO (General Purpose Input Output) pins.

The USB port in the Arduino board is used to connect the board to the computer using the USB cable. The cable acts as a serial port and as the power supply to interface the board. Such dual functioning makes it unique to recommend and easy to use for beginners.

2.2 Liquid Crystal Display

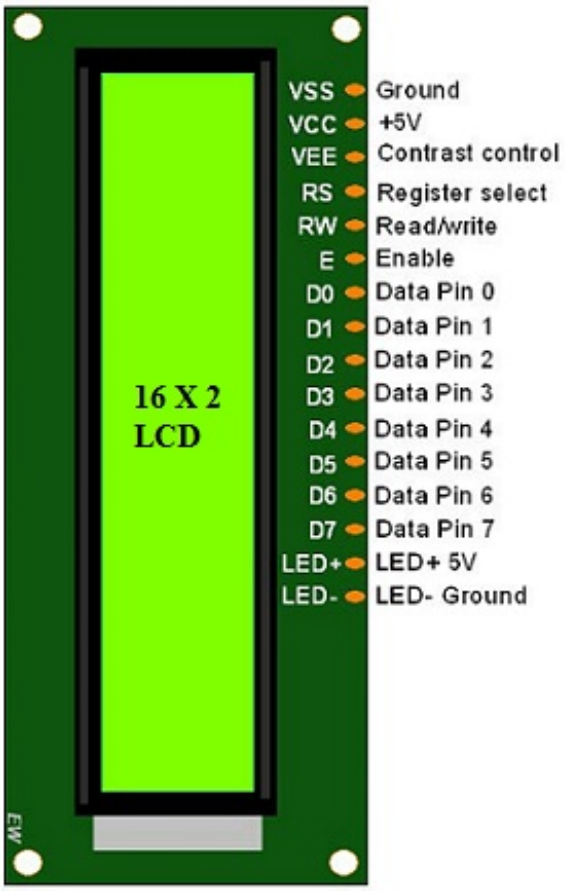
A liquid-crystal display (LCD) is a flat-panel display or other electronically modulated optical device that uses the light-modulating properties of liquid crystals combined with polarizers. Liquid crystals do not emit light directly but instead use a backlight or reflector to produce images in color or monochrome.

The block diagram is as follows:



These displays are mainly preferred for multi-segment light-emitting diodes and seven segments. The main benefits of using this module are inexpensive; simply programmable, animations, and there are no limitations for displaying custom characters, special and even animations, etc.

The pin diagram and description is shown below:



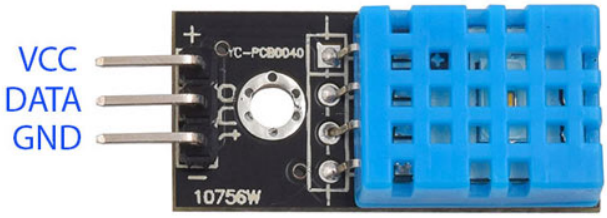
- Pin1 (Ground/Source Pin): This is a GND pin of display, used to connect the GND terminal of the microcontroller unit or power source.
- Pin2 (VCC/Source Pin): This is the voltage supply pin of the display, used to connect the supply pin of the power source.
- Pin3 (V0/VEE/Control Pin): This pin regulates the difference of the display, used to connect a changeable POT that can supply 0 to 5V.
- Pin4 (Register Select/Control Pin): This pin toggles among command or data register, used to connect a microcontroller unit pin and obtains either 0 or 1(0 = data mode, and 1 = command mode).

- Pin5 (Read/Write/Control Pin): This pin toggles the display among the read or writes operation, and it is connected to a microcontroller unit pin to get either 0 or 1 (0 = Write Operation, and 1 = Read Operation).
- Pin 6 (Enable/Control Pin): This pin should be held high to execute Read/Write process, and it is connected to the microcontroller unit and constantly held high.
- Pins 7-14 (Data Pins): These pins are used to send data to the display. These pins are connected in two-wire modes like 4-wire mode and 8-wire mode. In 4-wire mode, only four pins are connected to the microcontroller unit like 0 to 3, whereas in 8-wire mode, 8-pins are connected to microcontroller unit like 0 to 7.
- Pin15 (+ve pin of the LED): This pin is connected to +5V
- Pin 16 (-ve pin of the LED): This pin is connected to GND.

2.3 Temperature Sensor

The sensor that we have used for checking the temperature is DHT11.

The pin diagram of the sensor is as follows:

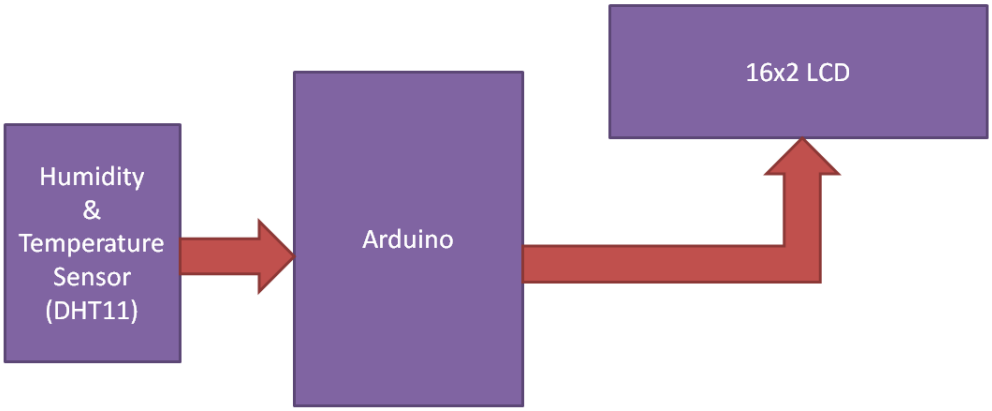


Pin 1 is the VCC pin and is connected to positive power supply.

Pin 2 is the data pin that takes the readings and sends it to the arduino.

Pin 3 is the GND pin which is connected to the ground.

The block diagram is as follows:



The temperature is taken by the sensor and is sent to the arduino which in turn send the data to LCD module and the results are displayed. Specifications of DTH11 are as follows:

- Operating Voltage: 3.5V to 5.5V
- Operating current: 0.3mA (measuring) 60uA (standby)
- Output: Serial data
- Temperature Range: 0°C to 50°C
- Resolution: Temperature and Humidity both are 16-bit
- Accuracy: ±1°C and ±1

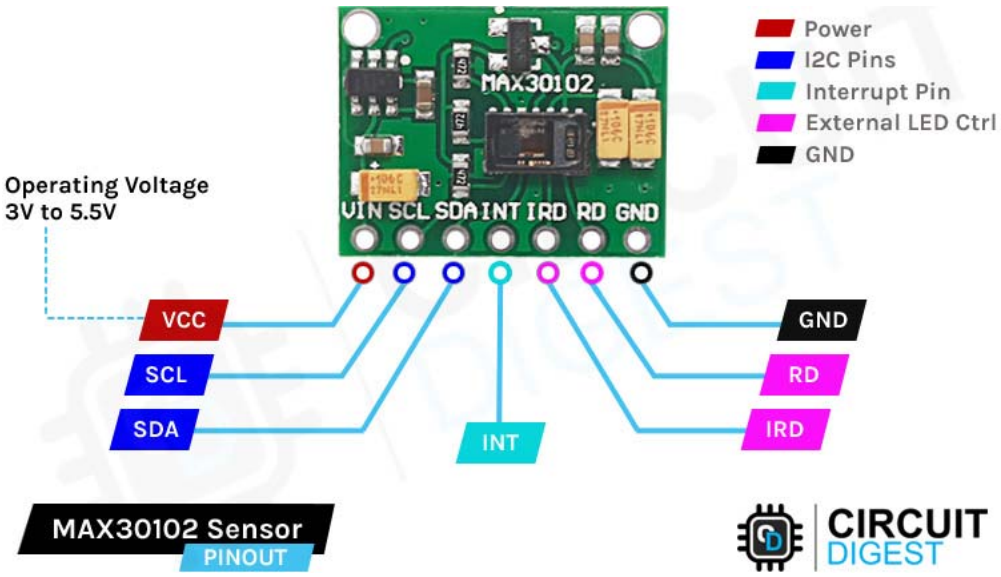
The DHT11 sensor can either be purchased as a sensor or as a module. Either way, the performance of the sensor is same. The sensor will come as a 4-pin package out of which only three pins will be used whereas the module will come with three pins as shown above.

The only difference between the sensor and module is that the module will have a filtering capacitor and pull-up resistor inbuilt, and for the sensor, you have to use them externally if required.

2.4 Heart Beat Sensor

The module features the MAX30102 – a modern (the successor to the MAX30100), integrated pulse oximeter and heart rate sensor IC, from Analog Devices. It combines two LEDs, a photodetector, optimized optics, and low-noise analog signal processing to detect pulse oximetry (SpO2) and heart rate (HR) signals.

The block diagram is as follows:



The advantage of the MAX30102 is its ability to add on to more powerful MCUs or MPUs from another manufacturer.

The MAX30102 has an on-chip temperature sensor that can be used to compensate for the changes in the environment and to calibrate the measurements. This is a reasonably precise temperature sensor that measures the 'die temperature' in the range of -40C to +85C with an accuracy of ±1C.

- Power supply: 3.3V to 5.5V
- Current draw: 600A (during measurements)
- 0.7A:(during standby mode)

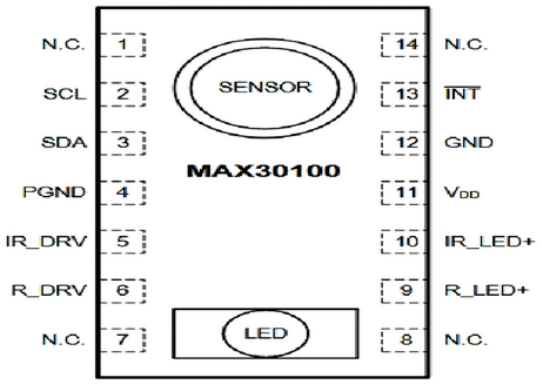
- Red LED Wavelength: 660nm
- IR LED Wavelength :880nm
- Temperature Range: -40C to +85C
- Temperature Accuracy: ±1C

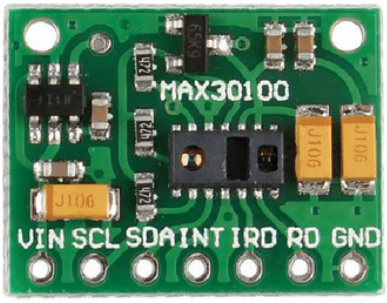
The MAX30102 chip requires two different supply voltages: 1.8V for the IC and 3.3V for the RED and IR LEDs. So the module comes with 3.3V and 1.8V regulators.

2.5 Oxygen level Sensor

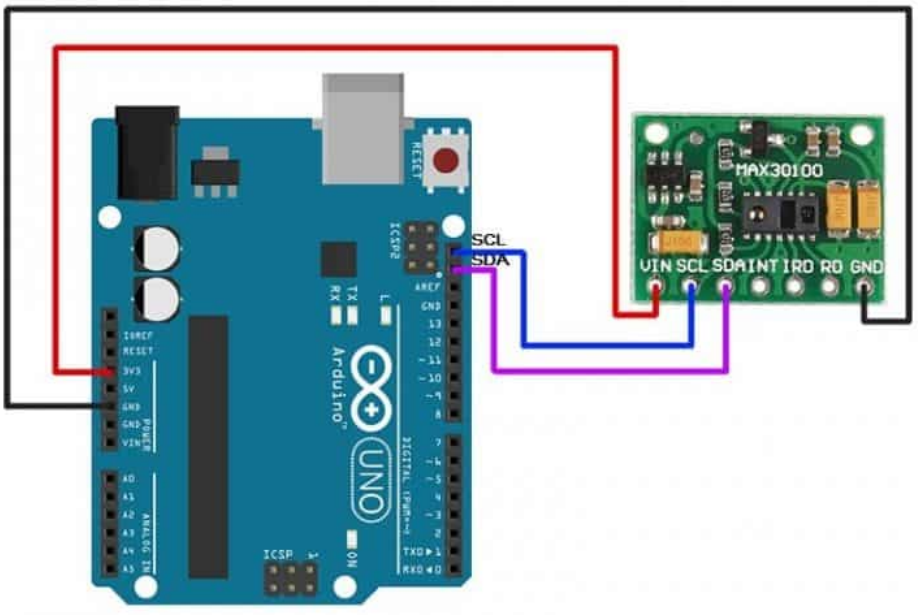
A pulse oximeter is a medical device that indirectly monitors the oxygen saturation of a patient's blood (as opposed to measuring oxygen saturation directly through a blood sample) and changes in blood volume in the skin, producing a photoplethysmogram that may be further processed into other measurements.

Pulse oximetry may be used to see if there is enough oxygen in the blood. This information is needed in many kinds of situations.





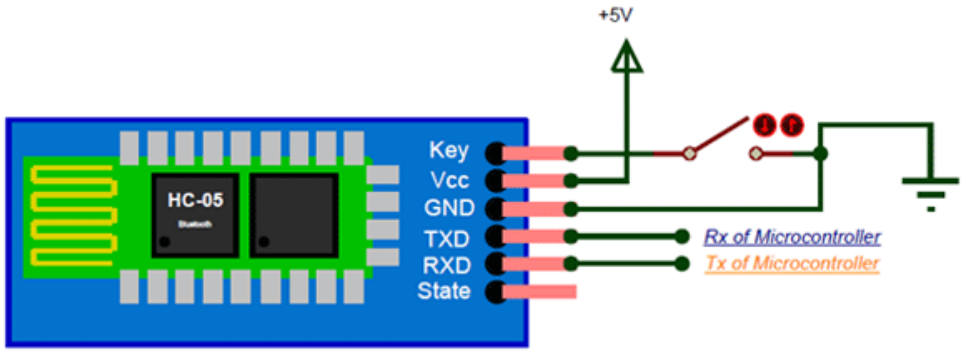
Connect the Vin pin of MAX30100 to Arduino 5V or 3.3V pin, GND to GND. Connect the I2C Pin of MAX30100, i.e SCL SDA to A5 A4 of Arduino.



Connect the Vin pin of MAX30100 to Arduino 5V or 3.3V pin, GND to GND. Connect the I2C Pin, SCL SDA of MAX30100 to A5 A4 of Arduino. Similarly connect the LCD pin 1, 5, 16 to GND of Arduino and 2, 15 to 5V VCC. Similarly connect LCD pin 4, 6, 11, 12, 13, 14 to Arduino pin 13, 12, 11, 10, 9, 8. Use 10K Potentiometer at pin 3 of LCD to adjust the contrast of LCD.

2.6 HC-05 Module

HC-05 has red LED which indicates connection status, whether the Bluetooth is connected or not. Before connecting to HC-05 module this red LED blinks continuously in a periodic manner. When it gets connected to any other Bluetooth device, its blinking slows down to two seconds. This module works on 3.3V.



When two devices are being paired, they randomly pick up one of the available 79 frequencies to make a connection, and, once that connection is established, they keep hopping across these frequencies many times a second.

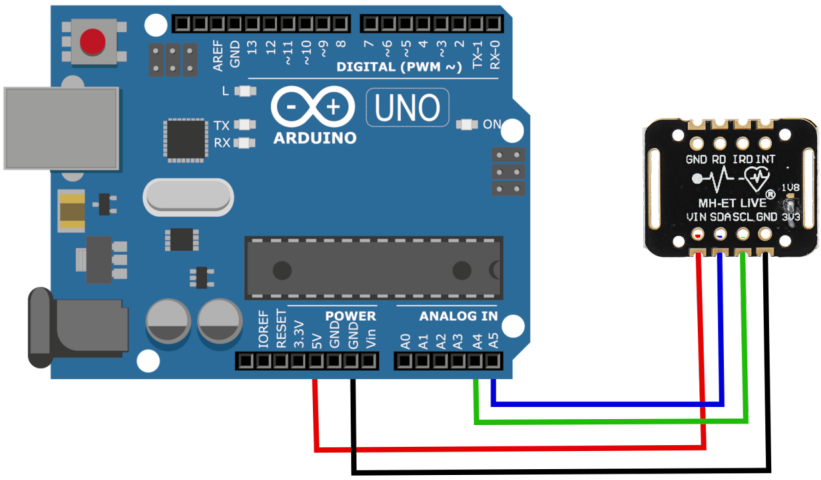
Chapter 3

Project Description

This project consists of three different sensors and using these sensors we can check the temperature,heart beat rate and the oxygen levels of blood of the patient.All these sensors are easily available and pocket friendly.

3.1 Heart Rate Monitoring

The MAX30102 is a very versatile sensor and it can also measure body temperature other than heart rate and blood oxygen level. This is a sensor designed by Analog Devices and features two LEDs (one Infrared and one Red), a photodetector, optics, and low-noise signal processing unit to detect pulse oximetry (SpO2) and heart rate (HR) signals.



MAX30102 sensor is used to detect blood oxygen and heart rate. First, infrared radiation is sent and reflected by hitting the finger, and then the amount of oxygen in the blood is determined by measuring the wave amplitude. Heart rate is also obtained by analyzing the time series response of this radiation. The MAX30102 is an integrated module compatible with the Arduino and STM32. It integrates a red LED with an infrared LED, a photoelectric detector, an optical device, and a low noise electronic circuit for ambient light suppression. Heart rate and blood oxygen data are also transmitted to the Arduino or other microcontrollers via I2C communication.

3.2 Working of MAX30102

The MAX30102 is a very versatile sensor and it can also measure body temperature other than heart rate and blood oxygen level. This is a sensor designed by Analog Devices and features two LEDs (one Infrared and one Red), a photodetector, optics, and low-noise signal processing unit to detect pulse oximetry (SpO2) and heart rate (HR) signals.

The main idea is that you shine a single LED at a time and check the amount of light that

is getting reflected back to the sensor. Based on the reflection you can determine the blood oxygen level and heart rate.

The operating voltage of the module is 3.3V to 5V with 600uA of Maximum current draw. But If we are talking about the MAX30102 IC on the module board the IC requires two different supply voltages for stable operation, which is why in the module board you can find two linear low dropout regulators. One is for 3.3V and the other one is for 1.8V. You can check out the parts marking section for better understanding. One of the most interesting features of this module is its power consumption, MAX30102 consumes less than 600A during measurement. And it consumes only 0.7A when it's in standby mode.

Because of this low power consumption, this device can be used in battery powered applications. Other than that this IC offers a I2C Interface for communication, a FIFO buffer for holding and sampling 16 different readings, and an ON-Chip Temperature Sensor to measure the body temperature.

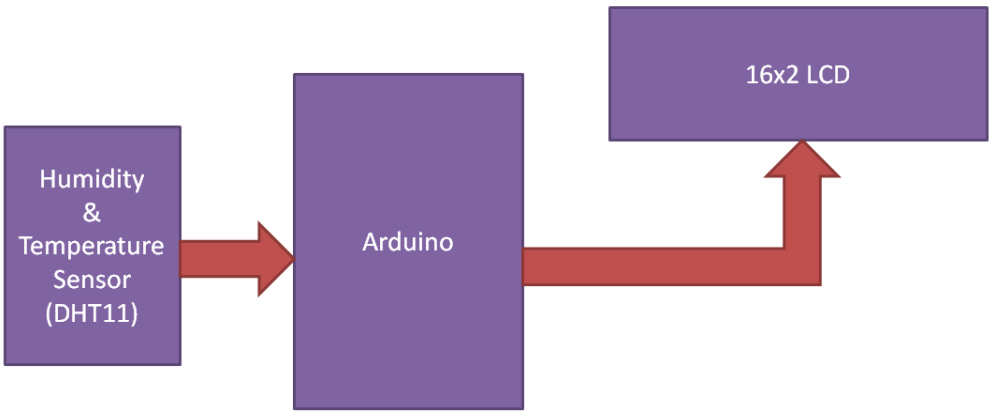
3.3 Temperature Monitoring

DHT11 Module features a temperature sensor complex with a calibrated digital signal output. The exclusive digital-signal-acquisition technique and temperature sensing technology ensure high reliability and excellent long-term stability. This sensor includes an NTC for temperature measurement and a resistive-type humidity measurement component for humidity measurement. These are connected to a high-performance 8-bit microcontroller, offering excellent quality, fast response, anti-interference ability, and cost-effectiveness.

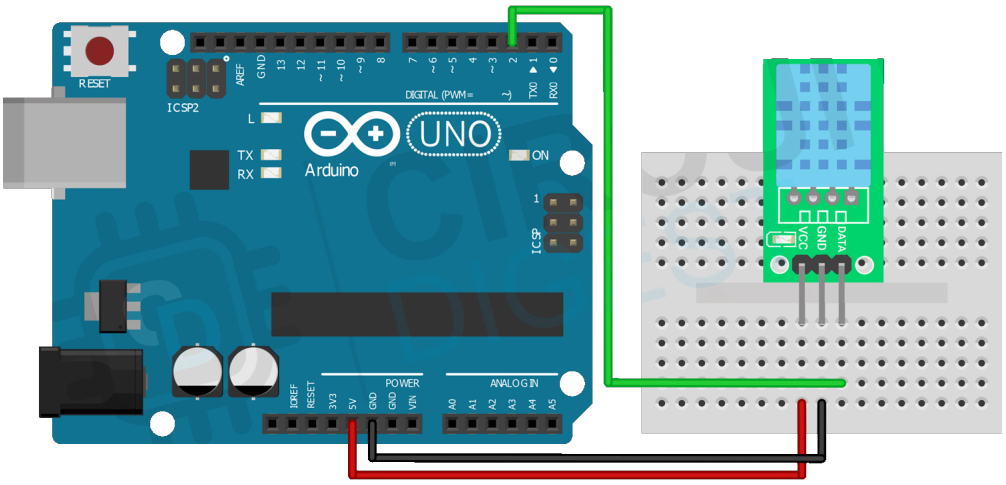
3.4 Working of DHT11 Sensor

This module sends data in form of pulse train of specific time period. Before sending data to Arduino it needs some initialize command with a time delay. And the whole process time is about 4ms. First of all Arduino sends a high to low start signal to DHT11 with 18µs delay

to ensure DHT's detection. And then arduino pull-up the data line and wait for 20-40µs for DHT's response. Once DHT detects starts signal, it will send a low voltage level response signal to arduino of time delay about 80µs. And then DHT controller pull up the data line and keeps it for 80µs for DHT's arranging of sending data. When data bus is at low voltage level it means that DHT11 is sending response signal. Once it is done, DHT again makes data line pull-up for 80µs for preparing data transmission. Data format that is sending by DHT to arduino for every bit begins with 50µs low voltage level and length of high voltage level signal determines whether data bit is "0" or "1".



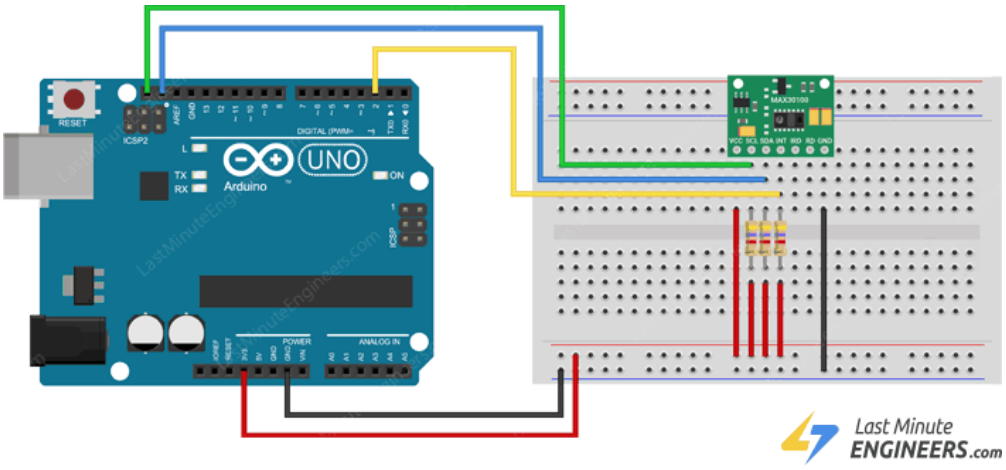
One important thing is to make sure pull up resistor value because if we are placing DHT sensor at ¡20 meter distance,5k pullup resistor is recommended. If placing DHT at longer the 20 meter then use appropriate value pull up resistor. It will send a low voltage level response signal to Arduino of time delay about 80µs. And then DHT controller pull up the data line and keeps it for 80µs for DHT's arranging of sending data. When data bus is at low voltage level it means that DHT11 is sending response signal. Once it is done, DHT again makes data line pull-up for 80µs for preparing data transmission. Data format that is sending by DHT to Arduino for every bit begins with 50µs low voltage level and length of high voltage level signal determines whether data bit is "0" or "1". One important thing is to make sure pull up resistor value because if we are placing DHT sensor at ¡20 meter distance,5k pullup resistor is recommended. If placing DHT at longer the 20 meter then use appropriate value pull up resistor.



Connections are pretty simple and only require three wires. Connect the VCC and GND of the module to the 5V and GND pins of the Arduino. Then connect the DATA pin to the Arduino's digital pin 2. We communicate with DHT11 through this pin.

3.5 Sp02 (Oxygen saturation) Monitoring

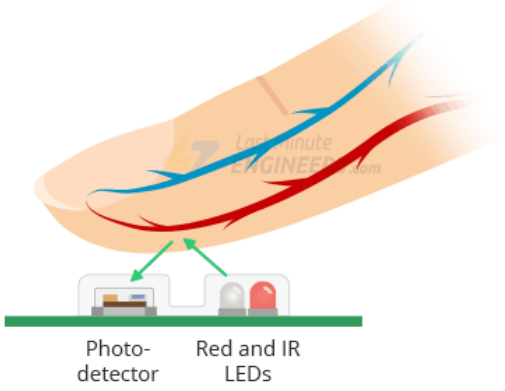
The MAX30100, or any optical pulse oximeter and heart-rate sensor for that matter, consists of a pair of high-intensity LEDs (RED and IR, both of different wavelengths) and a photodetector. The wavelengths of these LEDs are 660nm and 880nm, respectively.

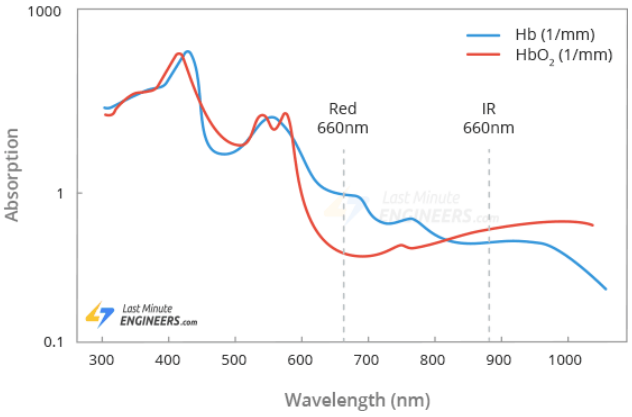


The MAX30100 works by shining both lights onto the finger or earlobe (or essentially anywhere where the skin isn't too thick, so both lights can easily penetrate the tissue) and measuring the amount of reflected light using a photodetector. This method of pulse detection through light is called Photoplethysmogram.

3.6 Working of MAX30100

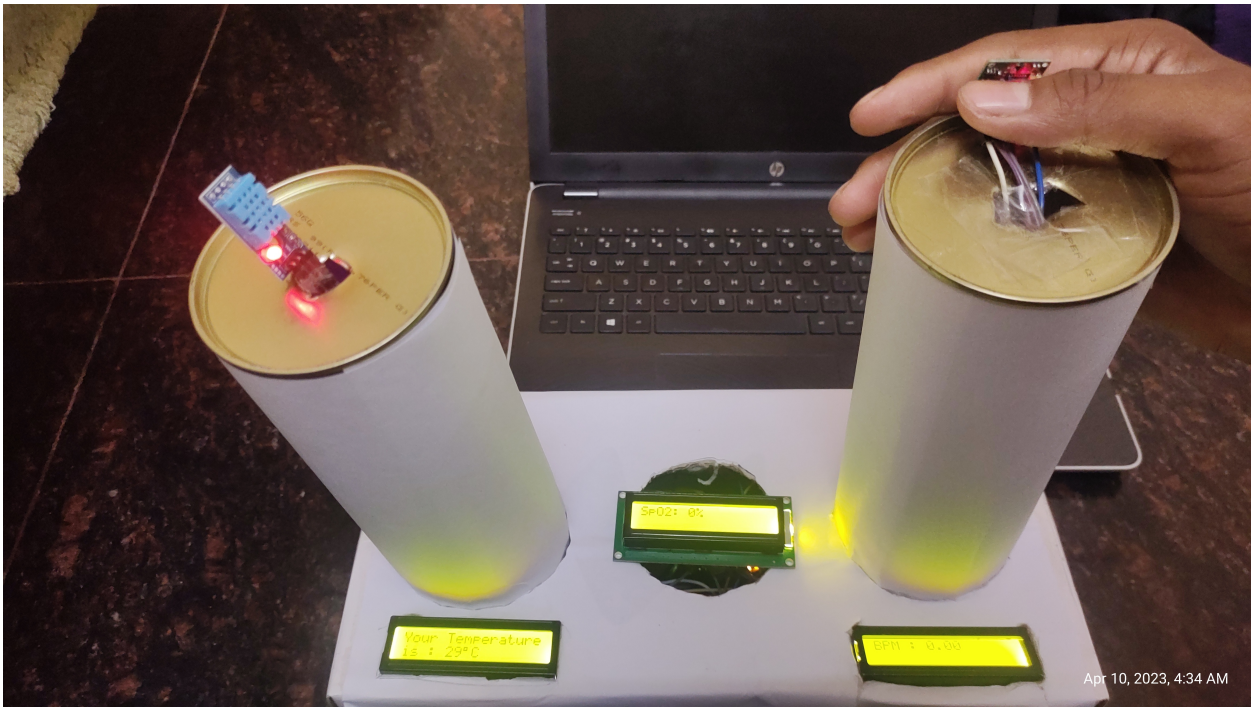
The pulse oximeter uses a cold light source that shines a light through the fingertip, making the tip appear to be red. By analyzing the light from the light source that passes through the finger, the device is able to determine the percentage of oxygen in the red blood cell.





The pulse oximeter observes a rapid measurement of oxygen saturation level in your body without using needles or taking a blood sample. The measured amount shown on the screen

reflects the saturation of your red blood cells with oxygen. This number gives your doctors and nurses an idea of what your treatment will be. The oxygen level may also help to determine if you need to receive supplemental oxygen. This saturation number (a good number would be over 90-92) differs from a value called the pO2 (a good number would be over 60-65) which is measured by obtaining blood from an artery. Your doctor can clarify the significance of your value related to your particular situation.



Project Model

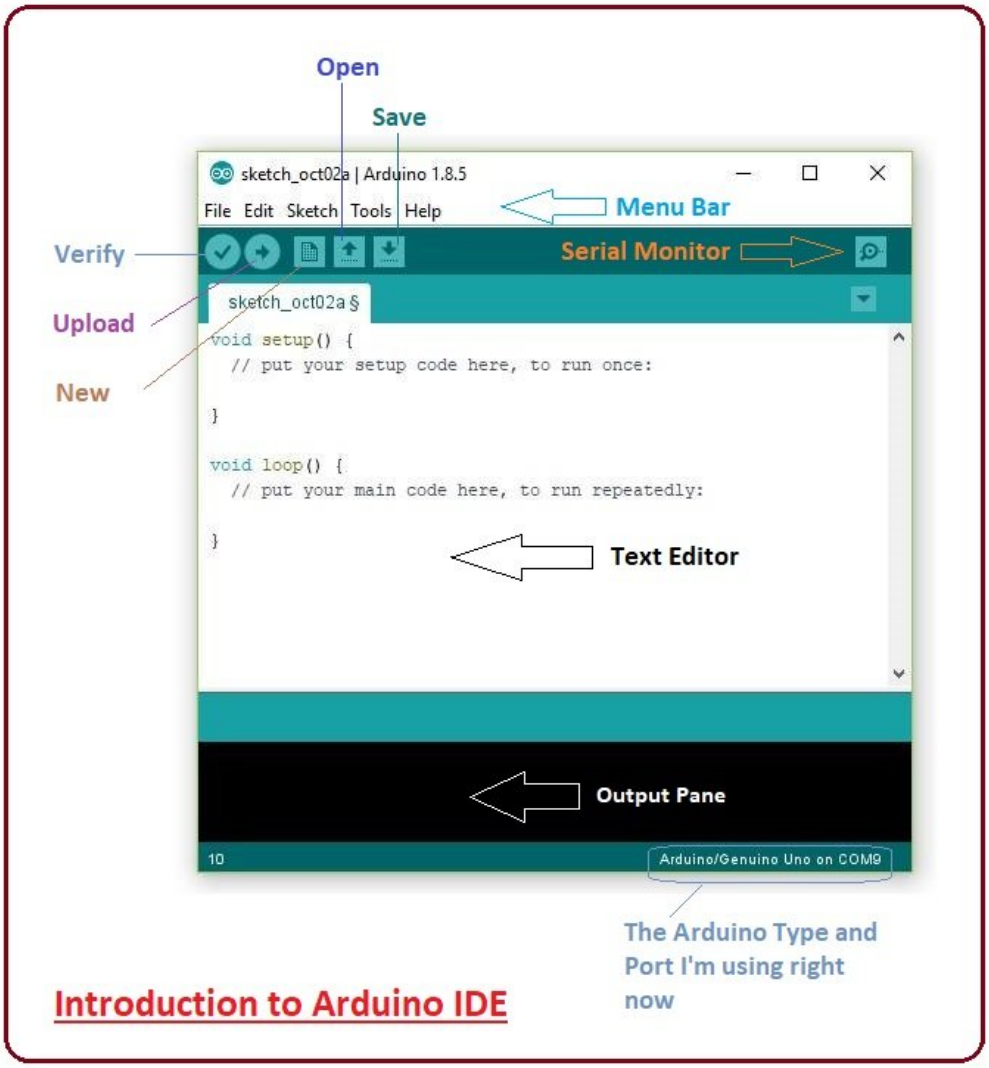
Chapter 4

Software tools and programming used

4.1 Arduino 1.8.19

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.



- Uploading: Before uploading your sketch, you need to select the correct items from the Tools > Board and Tools > Port menus. The boards are described below. On the Mac, the serial port is probably something like (for an UNO or Mega2560 or Leonardo) . On Windows, it's probably COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) - to find out, you look for USB serial device in the ports section of the Windows Device Manager. On Linux, it should be /dev/ttyACMx , /dev/ttyUSBx or similar. Once you've selected the correct serial port and board, press the upload button in the toolbar or select the Upload item from the Sketch menu. Current Arduino boards will reset automatically and begin the upload. With

older boards (pre-Diecimila) that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error.

When you upload a sketch, you're using the Arduino bootloader, a small program that has been loaded on to the microcontroller on your board. It allows you to upload code without using any additional hardware. The bootloader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The bootloader will blink the on-board (pin 13) LED when it starts (i.e. when the board resets).

- Libraries: Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, select it from the Sketch - Import Library menu. This will insert one or more include statements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the board with your sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its include statements from the top of your code.

  There is a list of libraries in the reference. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources or through the Library Manager. Starting with version 1.0.5 of the IDE, you do can import a library from a zip file and use it in an open sketch. See these instructions for installing a third-party library.

- Third-Party Hardware: Support for third-party hardware can be added to the hardware directory of your sketchbook directory. Platforms installed there may include board definitions (which appear in the board menu), core libraries, bootloaders, and programmer definitions. To install, create the hardware directory, then unzip the third-party platform into its own sub-directory. (Don't use "arduino" as the sub-directory name or you'll override the built-in Arduino platform.) To uninstall, simply delete its directory.

- Serial Monitor: This displays serial sent from the Arduino board over USB or serial connector. To send data to the board, enter text and click on the "send" button or press enter. Choose the baud rate from the drop-down menu that matches the rate passed to Serial.begin in your sketch. Note that on Windows, Mac or Linux the board will reset (it will rerun your sketch) when you connect with the serial monitor. Please note that the Serial Monitor does not process control characters; if your sketch needs a complete management of the serial communication with control characters, you can use an external terminal program and connect it to the COM port assigned to your Arduino board.

- Preferences: Some preferences can be set in the preferences dialog (found under the Arduino menu on the Mac, or File on Windows and Linux). The rest can be found in the preferences file, whose location is shown in the preference dialog.

Since version 1.0.1 , the Arduino Software (IDE) has been translated into 30+ different languages. By default, the IDE loads in the language selected by your operating system. (Note: on Windows and possibly Linux, this is determined by the locale setting which controls currency and date formats, not by the language the operating system is displayed in.)

If you would like to change the language manually, start the Arduino Software (IDE) and open the Preferences window. Next to the Editor Language there is a dropdown menu of currently supported languages. Select your preferred language from the menu, and restart the software to use the selected language. If your operating system language is not supported, the Arduino Software (IDE) will default to English.

You can return the software to its default setting of selecting its language based on your operating system by selecting System Default from the Editor Language drop-down. This setting will take effect when you restart the Arduino Software (IDE). Similarly, after changing your operating system's settings, you must restart the Arduino Software (IDE) to update it to the new default language.

4.2 CODES

4.2.1 Heart and SpO2 Monitoring

include <LiquidCrystal.h>

include <Wire.h>

include "MAX30100PulseOximeter.h"

LiquidCrystal lcd1(12, 11, 5, 4, 3, 2);

LiquidCrystal lcd2(12, 10, 5, 4, 3, 2);

define REPORTINGPERIODMS 1000

PulseOximeter pox;

uint32t tsLastReport = 0;

void onBeatDetected()

Serial.println("Beat!");

void setup()

Serial.begin(115200);

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

lcd1.begin(16,2);

lcd2.begin(16,2);

lcd1.print("Initializing...");

lcd2.print("Initializing...");

delay(3000);

lcd1.clear();

lcd2.clear();

// 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(MAX30100LEDCURR$_{76}MA$);

```
// Register a callback for the beat detection
pox.setOnBeatDetectedCallback(onBeatDetected);
void loop()
// Make sure to call update as fast as possible pox.update();
if (millis() - tsLastReport > REPORTINGPERIODMS)
Serial.print("Heart rate:");
Serial.print(pox.getHeartRate());
Serial.print("bpm / SpO2:");
Serial.print(pox.getSpO2());
Serial.println("
lcd1.clear();
lcd1.setCursor(0,0);
lcd1.print("BPM : ");
lcd1.print(pox.getHeartRate());
lcd2.clear();
lcd2.setCursor(0,0);
lcd2.print("SpO2: ");
lcd2.print(pox.getSpO2());
tsLastReport = millis();
```

4.2.2 Tempearture Monitoring

```
include <SoftwareSerial.h>
SoftwareSerial bt(8, 9); // RX, TX
include <LiquidCrystal.h>
include "dht.h"
define dataPin A0
LiquidCrystal lcd(2, 3, 4, 5, 6, 7);
dht DHT;
int temp;
void setup()
```

```
Serial.begin(9600);
bt.begin(9600);
Serial.println("Ready");
lcd.begin(16,2);
lcd.setCursor(0,0);
lcd.print(" WELCOME");
delay(2000);
lcd.clear();
void loop()
int readData = DHT.read11(dataPin);
temp = DHT.temperature;
lcd.setCursor(0,0);
lcd.print("Temp: ");
lcd.print(temp);
lcd.print((char)223); //degree symbol
lcd.print("C ");
bt.print(temp); //send distance to MIT App
bt.print(";");
delay(1000);
```

Chapter 5

Conclusion and future Scope

In this project we made a health care system using Arduino and different sensors namely DHT11,MAX30100 and MAX30102.Using this model people will be able to find out their temperature,SpO2 level as well as their heart rate level very easily,accurately and at a very low cost.
This project has the following applications:

- This DHT11 Temperature and Humidity Sensor features a calibrated digital output temperature and humidity sensor module. Its technology ensures high reliability and excellent long-term stability. A high-performance 8-bit microcontroller is connected.

- Temperature sensors are devices used to measure temperature in solids, liquids or gases. They are used within industrial applications and have many more commercial uses. Most of the temperature sensors we supply monitor temperature by measuring the change in resistance of an electrical current.

- It will detect the temperature at the greenhouse in the LCD display. From this, we can monitor the temperature by using the DTH11 sensor in the greenhouse, if the temperature and humidity are not in good condition, it will display in the LCD display. Ultra low power operation increases battery life for wearable devices. Advanced functionality

improves measurement performance, high SNR provides robust motion artifact resilience integrated. ambient, light cancellation high sample rate capability fast data output capability.

- In today's date, the electronic devices for measuring various body needs are already available in the market. The only problem with them is that they are very costly and that is why many people cannot afford it.Using this project we can build a full healthcare system which is pocket friendly and it will also save a lot of time.

References

1. L-H Wang, Y-M Hsiao, X-Q Xie, and S-Y Lee, "An Outdoor Intelligent Healthcare Monitoring Device for the Elderly," IEEE Trans. on Consumer Electronics, vol. 62, no. 2, pp. 128-135, May 2016.

2. Goutam Motika, Abinash Prusty," Wireless FetalHeartbeat Monitoring System Using ZigBee IEEE 802.15.4 Standard",2011 Second.International Conference on Emerging Applications of Information Technology, 978-0-7695-4329-1/11,2011 IEEE DOI 10.1109/EAIT.2011.

3. "Healthcare Monitoring System Using Wireless Sensor Network", D. Mahesh Kumar, Department of Electronics, PSG College of Arts and Science, Coimbatore - 641 014.Volume 04, Issue 01 Pages:1497-1500 (2012),ISSN:0975-0290.

4. Temperature Sensor datasheet, http://www.ti.com/lit/ds/symlink/lm35.pdf.

5. Heart Beat Sensor datasheet, http://www.sunrom.com/p/heart-beat-sensor-digital-pulse-out.

6. https://www.teachmemicro.com/max30100-arduino-heart-rate-sensor/

7. https://www.analog.com/en/products/max30102.htmlproduct-evaluationkit

8. https://circuitdigest.com/microcontroller-projects /how-max30102-pulse-oximeter-and-heart-rate -sensor-works-and-how-to-interface-with-arduino

9. https://www.arduino.cc/en/software