IoT Based Smart Healthcare

Md. Sharif Hasnat
011 143 003
Shamim Al Mamun
011 143 009
Fariha Hossain
011 143 017
Sumaiya Hossain
011 151 065

A project/thesis in the Department of Computer Science and Engineering presented in partial fulfillment of the requirements for the Degree of Bachelor/Master of Science in Computer Science and Engineering

United International University
Dhaka, Bangladesh

February, 2019

Declaration

We, Md. Sharif Hasnat, Shamim-Al-Mamun, Fariha Hossain, Sumaiya Hossain declare that this thesis titled, **“IoT Based Smart Healthcare”** and the work presented in it is our own. We confirm that:

- This work was done wholly or mainly while in candidature for a BSc degree at United International University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at United International University or any other institution, this has been clearly stated.
- Where we have consulted the published work of others, this is always clearly attributed.
- Where we have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- We have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, we have made clear exactly what was done by others and what we have contributed ourselves.

____

Md. Sharif Hasnat, 011 143 003 Department of CSE

____

Shamim-Al-Mamun, 011 143 009 Department of CSE

____

Fariha Hossain, 011 143 017 Department of CSE

____

Sumaiya Hossain, 011 151 065 Department of CSE

Certificate

I do hereby declare that the works embodied in this project entitled **“IoT Based Smart Healthcare”** is the outcome of an original work carried out by **Md. Sharif Hasnat, Shamim-Al-Mamun, Fariha Hossain, Sumaiya Hossain** under my supervision.

I further certify that the dissertation meets the requirements and the standard for the degree of BSc in Computer Science and Engineering.

Mohammad Mamun Elahi
Assistant Professor, Department of CSE
United International University

Abstract

In regular medical system in hospitals patients' medical records are written in papers and sometimes they forget to bring it next time or lose it between appointments. For this reason, doctors are mostly unaware about patients' previous condition. Our project is about a smart healthcare system that is based on IoT (Internet of things). Basically our focus on this project is to determine patient health condition and store the patients' medical records in a database. Smart healthcare is a system where patient health condition can be sensed by IOT based sensors, for example, heart rate sensor, pulse sensor, temperature sensor, blood pressure sensor. These sensors are connected with Arduino UNO by ESP8266 NodeMCU, which works as a gateway for connecting with the internet or cloud, where patients' data are stored in the real time database so that doctors can analyze patients' health condition and gives proper treatment.

Acknowledgement

First of all we would like to thank almighty Allah for giving us the strength and ability to successfully finish our project/thesis work. We like to thank to our mentor Mohammad Mamun Elahi, who accepted our project idea of "IoT based Smart Healthcare" and supported us throughout the journey. His advices, guidelines and patience has given us the opportunity to complete this project. Than we would like to thank our course teacher Dr. Salekul Islam, who supported us and encouraged us.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

We build a smart healthcare system using IoT devices includes some sensors which are connected to internet. The main motive of our project is to provide good services to our patient and make our hospitals smart.

1.1 Problem Statement

In Bangladesh there are 16 crores people and most of the people go to the govt. hospital for better treatment. The condition of our govt. hospitals is extremely bad. The equipment's the hospitals provide are old and not giving the exact measurement. So for this reason sometimes the patients get wrong treatment. Also, every time they go to the hospital they need to register their identity and tell about their previous treatment history. The world becomes smarter day by day. So, it is not smart that every time you need to carry your treatment history with you. How it would be if any such devices that could identify our diseases and also can record the history of our treatment. In another words, we always feel need of a smart healthcare system.

1.2 Motivation and Objective

The main motives of the project is to create a smart healthcare system where patient's medical records are stored in database and improve the quality of services in our hospitals. Use the IoT sensors in our medical system to give proper treatment to the patients. Doctor can maintain health parameter of its patient just be visiting website or URL.

1.3 Goals of the project

So our main goal in this project is to build a smart healthcare system where we use different types of sensors and collects the medical data accurately and send the data to database and provides a better treatment with data security.

1.4 Organization of the Report

The report is organized as follows. Chapter 2 discusses some theoretical background about IoT, sensors in healthcare system etc. Chapter 3 presents the proposed System architecture. Chapter 4 details the working procedure, results and experimental analysis. Finally, chapter 5 gives some concluding remarks and possible future works.

Chapter 2

Background and Literature Review

In this chapter we will discuss about IoT, sensors in healthcare system, present condition of IoT devices in healthcare system.

2.1 What is IoT?

The Internet of Things (IoT), refers to the billions of physical devices around the world that are now connected to the internet, collecting and sharing data.

Uses of IoT

- One of the most practical use of IoT in Smart Homes. Using different kind of sensor and actuators it's made our home smart.
- In Smart Cites like smart recycle, smart grid, smart communication system etc.
- In Farming like Smart Irrigation system.
- In Healthcare, it's made our healthcare smarter than before.
- In Retails Shops like Amazon Go.

2.2 IoT Sensors

There are different kind of sensors available in many IoT based sectors. More are discovering day by day. Let's discuss about some sensors created for healthcare system.

Motion Detection-Accelerometers: This sensor is used to detect patient's body position and using this position it allows other sensors to do their work.

Blood Pressure Sensor: It is used to measure blood pressure from human body.

Pulse Sensor: It is used to detect pulse from human body.

Glucometer: It is used to measure glucose level from human blood etc.

Oximeter Sensor: It is used for measuring the oxygen level in blood and it can measure pulse rate also.

2.3 Smart Healthcare in Developed Countries

In Developed Countries like USA, China, Russia, Singapore etc. the success of the use of IoT devices are increasing day by day. They are researching more about how to use IoT in their healthcare system. Now they are using mobile phones as a diagnostic tools for a wide variety of disease and conditions. In their countries now people don't go to hospital to test their daily health condition. All they do is they use different IoT based smart device to test their body condition and the result is saved to the cloud and sent to their personal doctors. Then doctor can track their health condition remotely. In case of sudden emergency situation, a notification system is sent to doctor to notify the current status of their health. So, many countries are already developed in this sector and they are getting smarter day by day discovering different IoT based technology.

2.4 Smart Healthcare in Developing Countries

Currently the healthcare systems are either based on hospital or household environ and its scope is confined to rural areas of the country. Also healthcare system in our country are facing challenges such as lack of ubiquity, good treatment, good service, use of outdated technique, lack of tracking health condition smartly and accurately and most importantly inaccurate storing of patients data. We all know that Bangladesh is developing country and its healthcare system is still old fashioned. Also in our country most of the attendance and nurse are not well trained and they also doesn't know how to do their job smartly. Even sometimes their fault causes patient death. But, as our country is developing day by day also the entire system is getting smarter. Our researcher and even students are giving different proposal to integrate IoT in our healthcare system. And this research are getting success day by day and we hope, in future our healthcare system will be smart and IoT based and remotely controlled [2].

2.5 IoT in Smart Healthcare

IoT (internet of things) is infrastructure of connected physical device. Smart healthcare is very useful application of IoT and we use it in low cheaper cost than any other system and its work well. A healthcare system where patient health condition can be identified by IoT based sensor (oximeter, temperature, pulse sensor) and it's stored in real-time database so that doctor can analysis patient health condition and get proper treatment present and future [1].

Chapter 3

Proposed System

In this chapter we discussed about the project model, instruments need for this project and the final model of our project.

3.1 Proposed Model

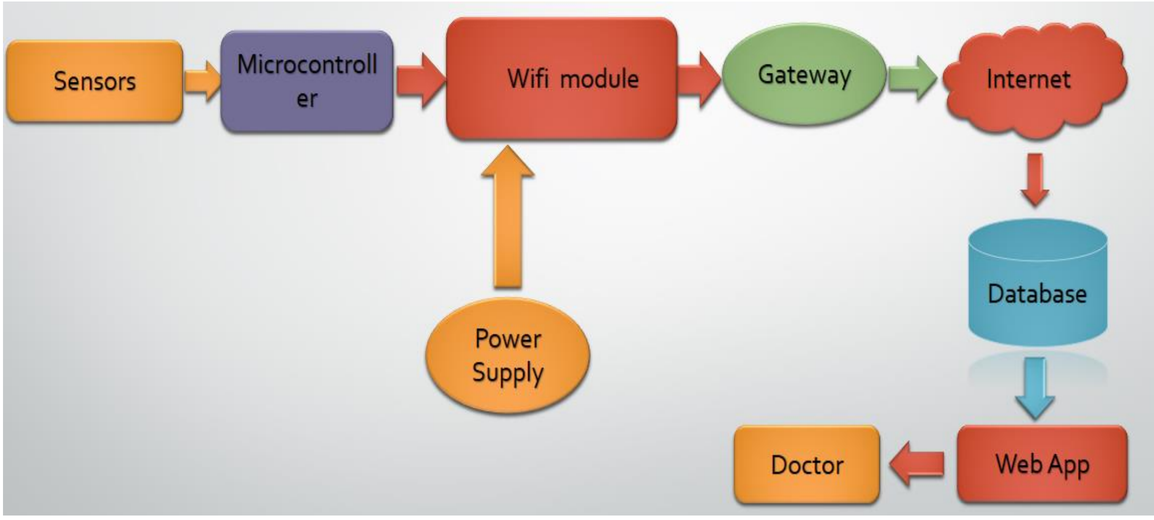


Figure 1: Proposed Model

In (Figure: 1) for sensors we use temperature sensor (DS18B20),oximeter sensor(max30100),for micro controller we use Arduino UNO and nodeMCU. NodeMCU also used for wifi module. For gateway we used a router with internet connection. We used Firebase for data storage. It is the initial model of our project.

3.1.1 NodeMCU

NodeMCU(Figure: 2) is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the development kits. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266 [8].It uses many open source projects.

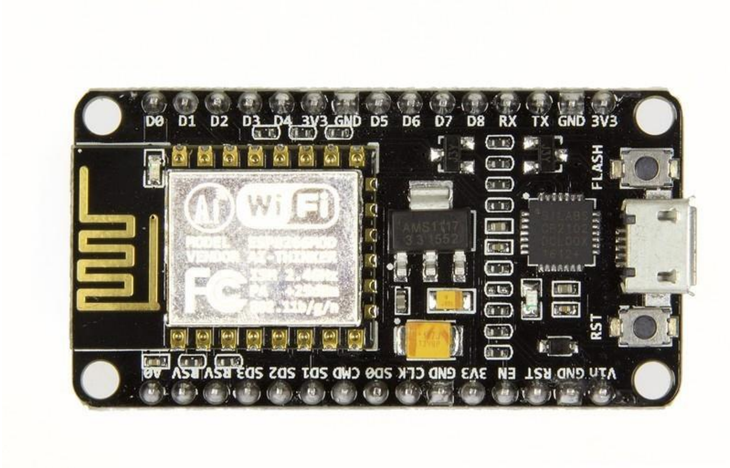


Figure 2: NodeMCU

3.1.2 Temperature Sensor

This is a pre-wired and waterproofed version of the Temperature Sensor DS18B20 (Figure: 3) sensor. Handy for when you need to measure something far away, or in wet condition. While the sensor is good up to 125°C the cable is jacketed in PVC so we suggest keeping it under 100°C. Because they are digital, you don't get any signal degradation even over long distances! These 1-wire digital temperature sensors are fairly precise (±0.5°C over much of the range) and can give up to 12 bits of precision from the onboard digital-to-analog converter. They work great with any micro controller using a single digital pin. It can measure the body temperature. It can measure the room temperature [7].

Figure 3: Temperature Sensor

3.1.3 Oximeter Sensor(MAX30100)

Figure 4: Oximeter Sensor

The MAX30100 (Figure: .4) is an integrated pulse oximetry and heart-rate monitor sensor solution. It combines two LEDs, a photo detector, optimized optics, and low-noise analog signal processing to detect pulse oximetry and heart-rate signals.

Uses of Oximeter Sensor

- Can measure the heartbeat or pulse rate.
- Can measure the oxygen level in blood.

3.1.4 RFID Sensor and Tags

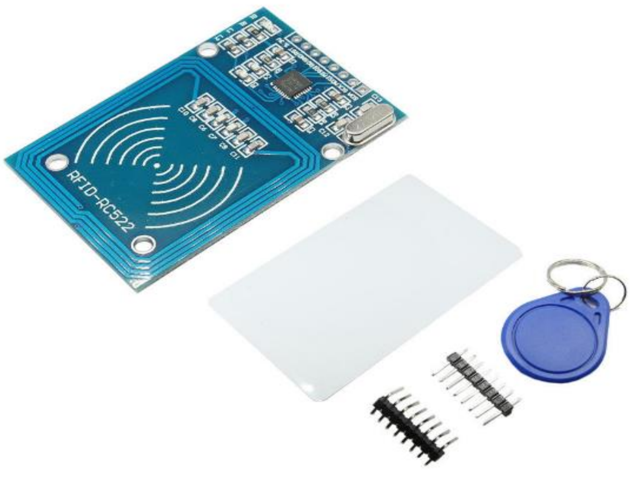


Figure 5: RFID Sensor and Tags

Radio-frequency identification (RFID) uses electromagnetic fields to automatically identify and track tags attached to objects. The tags contain electronically-stored information [9].

Uses Of RFID Sensor and Tags

- Commerce.
- Retail.
- Access Control.
- Track & Trace test vehicles and prototype parts.
- Passports.
- Animal Identification.
- Hospitals and healthcare.
- Libraries.
- Museums.
- Schools.
- Complement Barcode.

3.1.5 RTC Module

Figure 6: RTC Module

This handy device (Figure: 6) provides accurate time for years using a tiny coin-cell and it is very simple to connect to an Arduino project.

3.1.6 Other Components

- Internet connection
- Resistances
- Power supply

- Breadboards
- High quality USB cable
- Jumper wires
- LED’s
- A computer with Arduino IDE
- A wireless Router

3.2 System Block Diagram

So, the final block diagram of the project in Figure:7 as bellow.

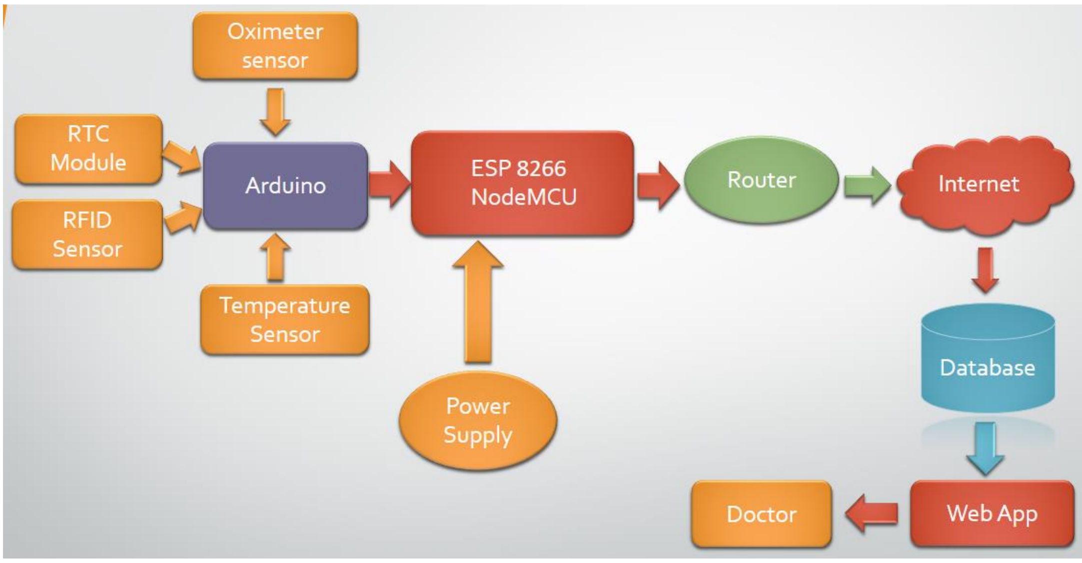


Figure 7: System Block Diagram

Chapter 4

Procedures and Results

In this section we discussed about the connection between sensor and Arduino UNO and also discussed about how they work.

4.1 Connection between sensors and Arduino UNO

In this section we discussed about the connection between sensor and Arduino UNO and also discussed about how they work.

4.1.1 Connection with Temperature Sensor

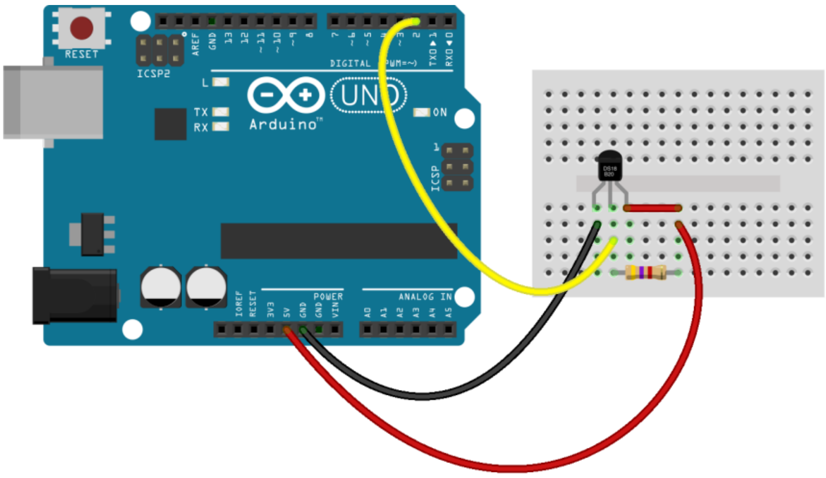


Figure 8: Connection with Temperature Sensor

The Figure: 8 belongs to how temperature sensor works with Arduino. At first we need to add connection between Temperature sensor's positive pin with the vcc pin of Arduino by a jumper wire. The data part of the temperature sensor should be connected with the any digital pin of Arduino. After that we need to compile the temperature sensor code in Arduino IDE for getting the value of temperature. Before that the silver part of temperature sensor should be attached with armpit. For better result we need to take it in our mouth. After compilation of code we need to open our serial monitor to see the output.

4.1.2 Connection with Oximeter Sensor

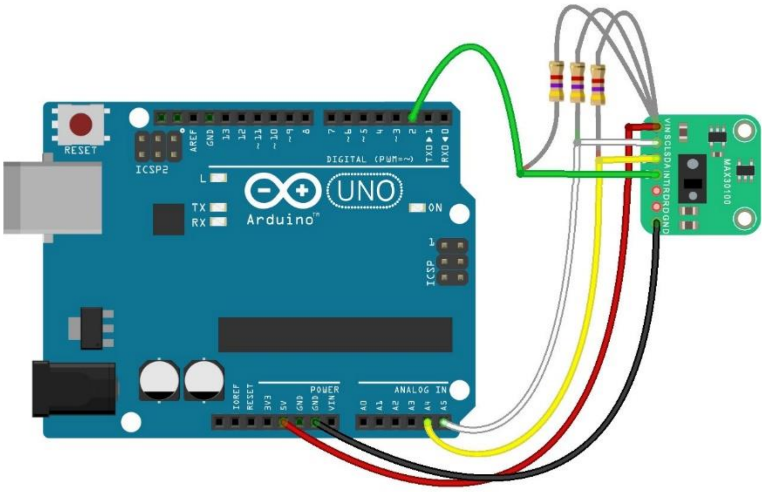


Figure 9: Connection with oximeter sensor

To establish connection between Arduino UNO and oximeter sensor we need to connect oximeter sensor's vin pin to Arduino's 5V pin, ground pin should connect with Arduino's ground pin. SCL and SDA part must connect serially with A4 and A5 pin of Arduino UNO. And atlast INT pin connects with D2 pin of Arduino UNO.

4.1.3 Connection with RFID Sensor

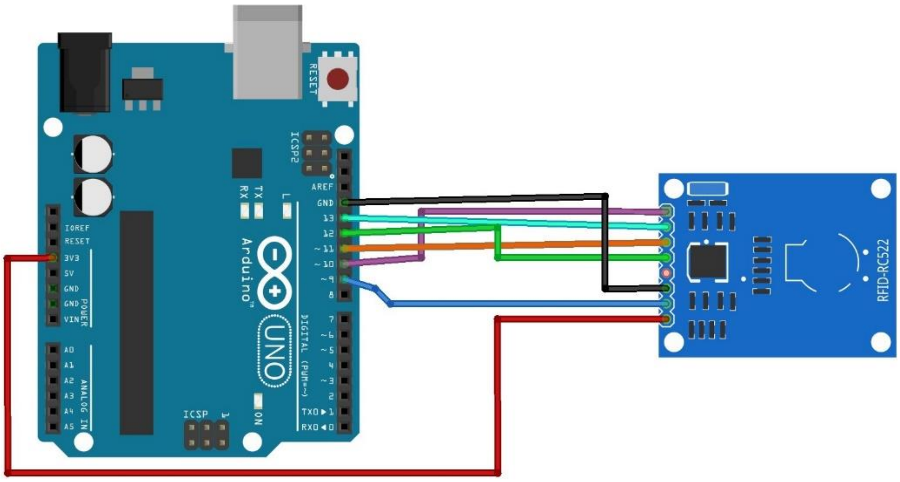


Figure 10: Connection with RFID sensor

To establish connection between RFID sensor and Arduino UNO connect the pins as like as mentioned in Table:1 .

RFID Sensor	Arduino UNO
VCC	3.3V
GND	GND
GND RST 3.3V	Digital Pin 9
MISO	Digital Pin 12
MOSI	Digital Pin 11
SCK	Digital Pin 13
SDA	Digital Pin 10

Table 1: Connection between RFID sensor and Arduino UNO

4.1.4 Connection with RTC Module

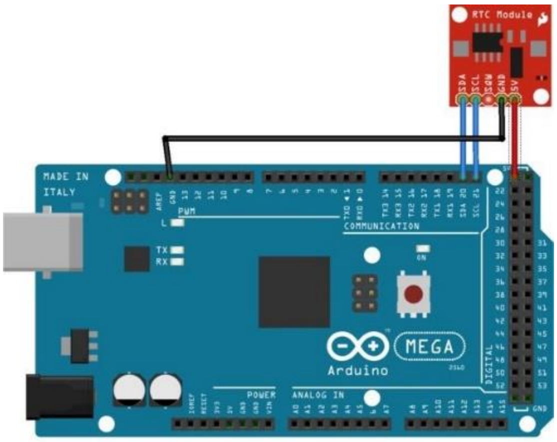


Figure 11: Connection with RTC Module

To create connection between RTC and Arduino UNO connect the pins as like as mentioned in Table: 2.

RTC	Arduino Uno
SDA	Analog Pin 4
SCL	Analog Pin 5
GND	GND
5V	5V

Table 2: Connection with RTC Module

4.2 Arduino IDE to NodeMCU connection

These steps mentioned in Figure:12 used to establish connection between Arduino IDE and nodeMCU.

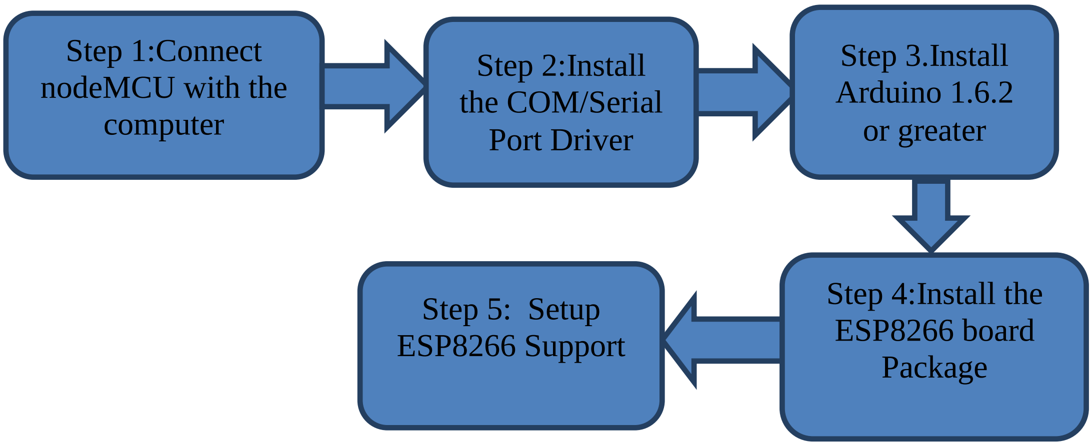


Figure 12: Steps of NodeMCU on Arduino IDE

4.3 Firbase Account Opening

At first we need to install "Firebase Arduino" library in Arduino IDE. In this library we must write router SSID & password and also firebase database link and secret key. To get firebase database link and secret key we have to follow the bellow steps.
Step 1: We need to create a Gmail account
Step 2: Browse http://www.firebase.google.com and sign in to firebase account
Step 3: Go to console and add a project with project name, project id & location
Step 4: Go into database option and collect the link provided by firebase
Step 5: Go to project setting->service account->database secrets->show secret key

After compilation the library code, nodeMCU will be connected with firebase database and now any sensor or device connects with nodeMCU can send data to firebase database.

4.4 Working Procedure

There are two users in the system, doctors and attendant. Doctors can search patient's information, prescribe the patients. Attendant can operate the hardware and registration process of the patients. When a patient first time come in to hospital the attendant will create an account for the patient (Figure: 13) with his full details (like name, address, phone number etc.) and also give him a RFID tag card with an unique Id . The tags contain electronically-stored information. Then the attendant will help patient for getting the values from the sensor and the sensors data will automatically store into firebase database (Figure: 14). If the patient already registered he does not need to register again for 2nd time when he comes. He just need to pass with the tests. After the tests when the patient will go to the doctor, the doctor will get the patient the data which is stored in database. He can also search patients all medical details, can prescribe patients. Doctor can also print the pdf version of prescription.

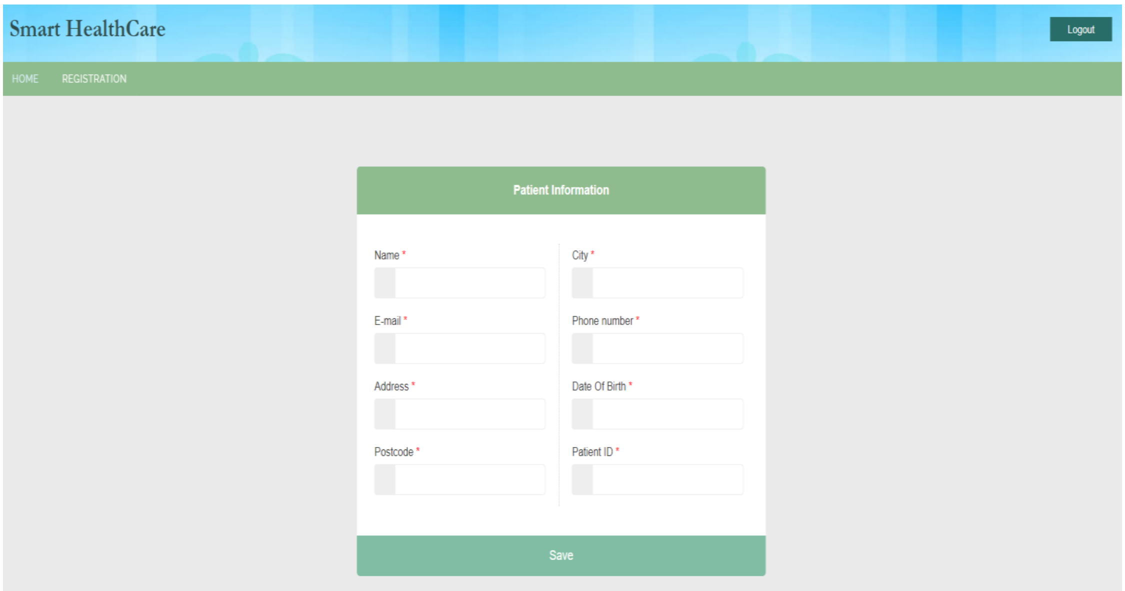


Figure 13: Patient Registration Page

Figure 14: Patient Data’s Store in Firebase Database

After the tests when the patient will go to the doctor, the doctor will get the patient's the data which is stored in database. He can also search patients all medical details (Figure: 15), can prescribe patients. Doctor can create the prescription (Figure: 16). All prescription's information will save into firebase database. Doctor can also print the pdf version of prescription.

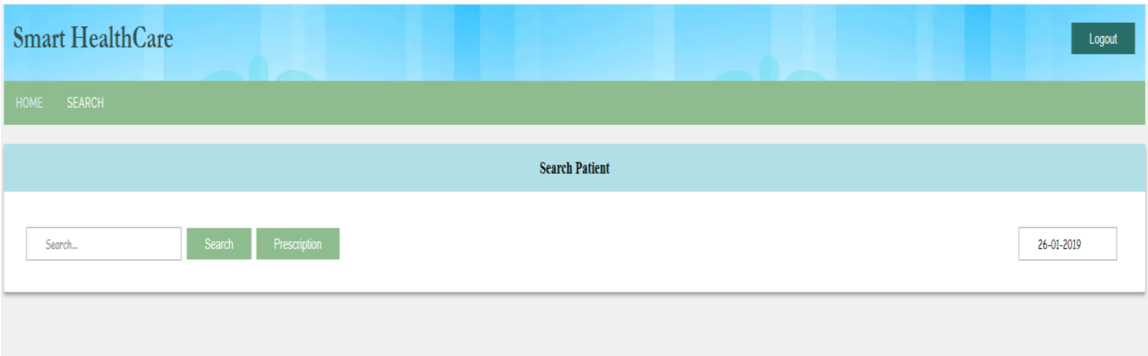
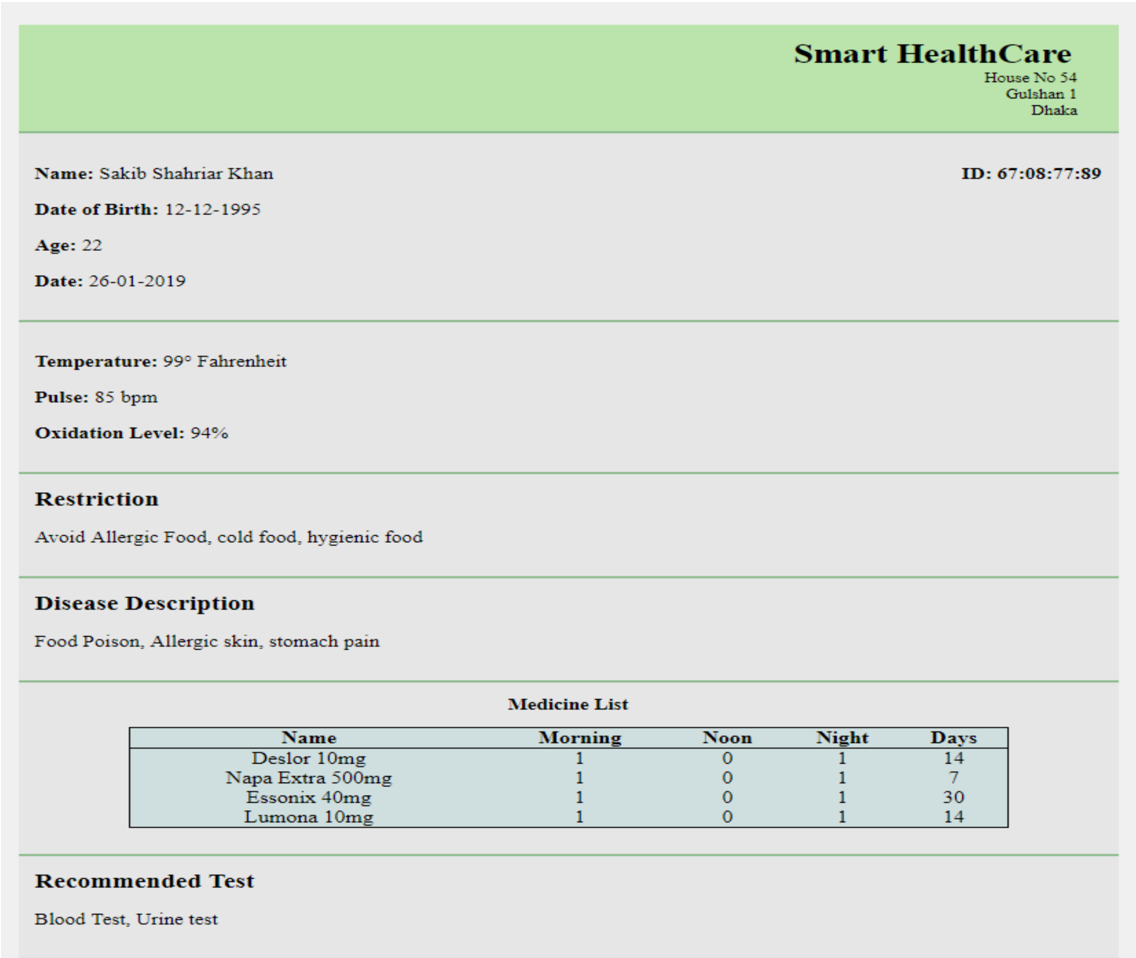


Figure 15: Doctor's Search Page



Smart HealthCare
House No 54
Gulshan 1
Dhaka

Name: Sakib Shahriar Khan

ID: 67:08:77:89

Date of Birth: 12-12-1995

Age: 22

Date: 26-01-2019

Temperature: 99° Fahrenheit

Pulse: 85 bpm

Oxidation Level: 94%

Restriction

Avoid Allergic Food, cold food, hygienic food

Disease Description

Food Poison, Allergic skin, stomach pain

Medicine List

Name	Morning	Noon	Night	Days
Deslor 10mg	1	0	1	14
Napa Extra 500mg	1	0	1	7
Essonix 40mg	1	0	1	30
Lumona 10mg	1	0	1	14

Recommended Test

Blood Test, Urine test

Figure 16: Prescription of a Patient

4.5 Variation of readings between sensors and real life devices

We made an experiment with 10 people using our temperature sensor and oximeter sensor and also done this experiment with the real life devices thermometer, Omron bp checker monitor. We know the normal temperature of human body is 97.7–99.5 °F. We get the average differences of 0.44 °F (Figure: 17) from the comparison between temperature sensor and thermometer value. We think we can reduce the differences using high quality temperature sensor.

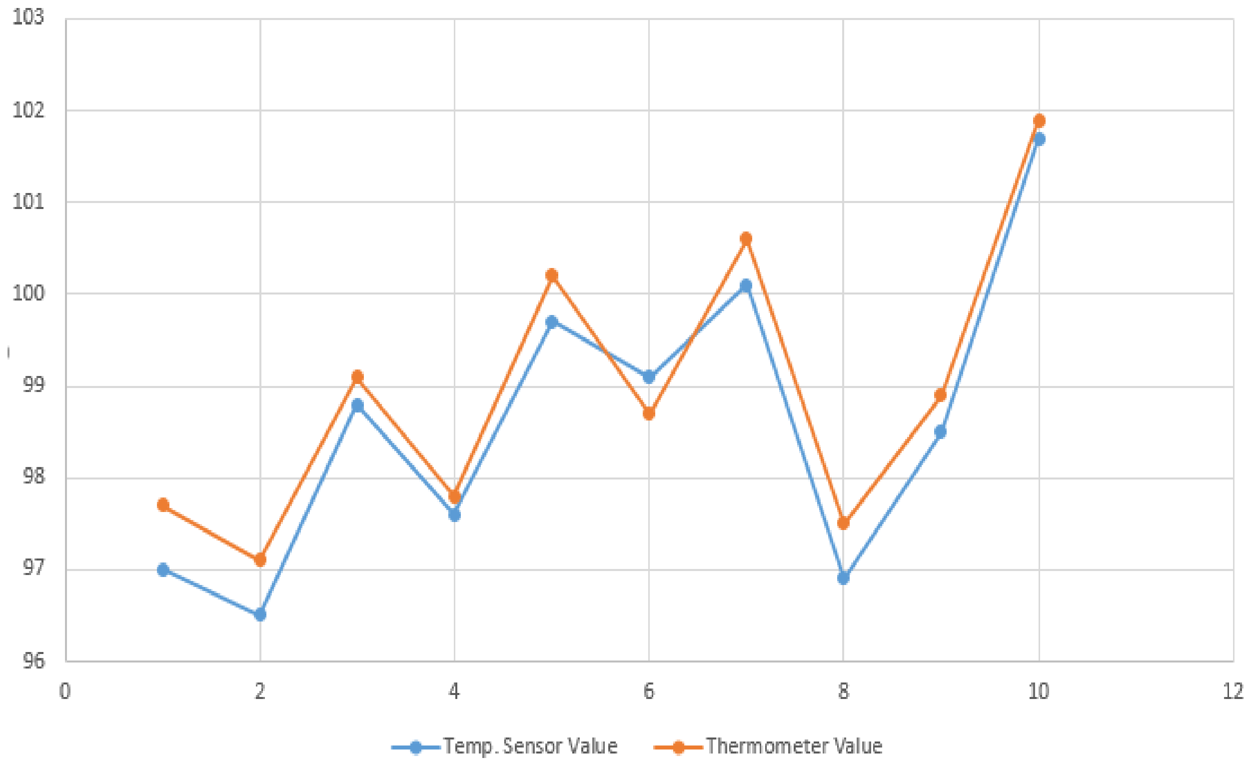


Figure 17: Differences between temperature sensor & thermometer value

The normal pulse rate of a human is 60-100 BPM. Most of the values we found near 60-100 (BPM). So we think our sensor works well. We get the average differences of 4.6 (BPM) (Figure: 18) from the comparison between oximeter sensor's heartrate and omron bp checker heart rate value. The reading will be more accurate if we reduce the light near the sensor and the sensor should be more.

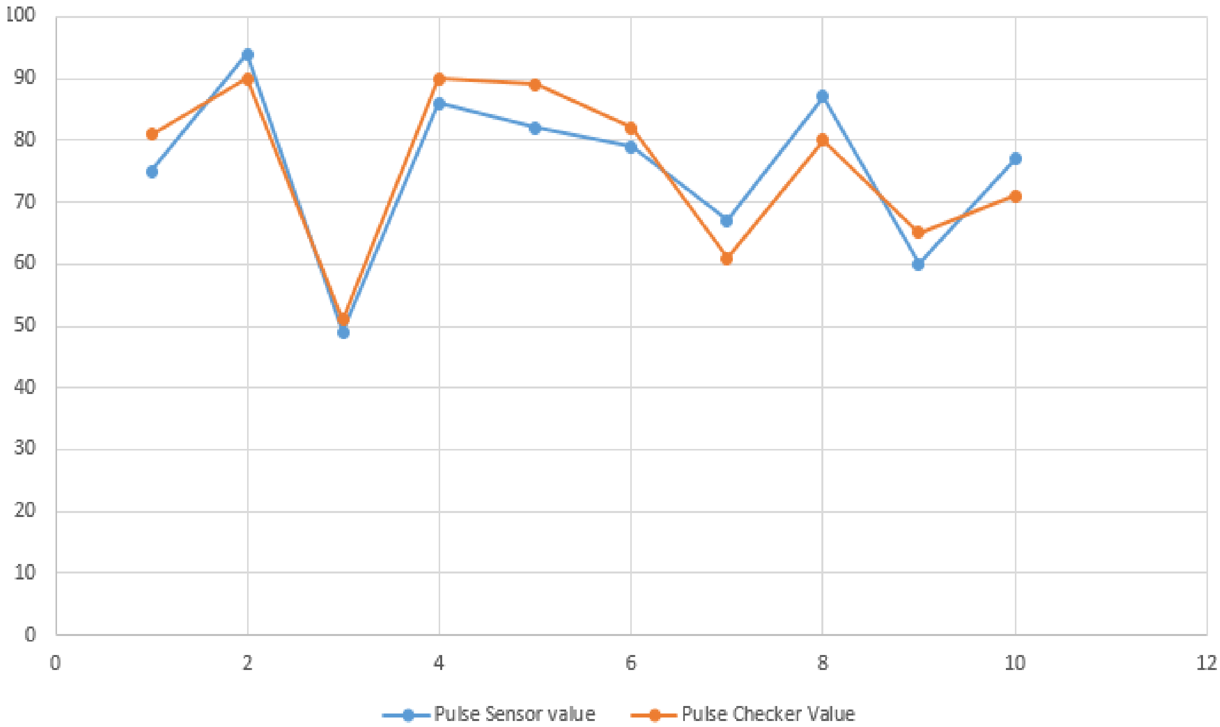


Figure 18: Differences between oximeter sensor heartrate & omron bp checker monitor's heart rate value

In this chapter, we have discussed how we developed the project, how the sensors worked and how we did the experiment with the sensors. The results are elaborately shown through tables, figures and graphs.

Chapter 5

Conclusion

We learn many things by completing this project .We learn about how IOT based sensor works in real life .This project is helpful for both Doctors and patients .Doctors can know the patient health condition by IoT based sensor and also analyzes health condition data stored in real time database. It saves time and patients get proper treatment instance.

5.1 Where this project can be used

This project can be used in hospitals and diagnostic centers for patient management system. And also it can be used in home for remote monitoring of health.

5.2 Limitations

The sensor we use their quality is not so good, so we did not get the efficient value as real life devices. Also the project board is little bit big in size so it can't be used as a portable device. Also cost is a factor. If we use high quality sensors then the project costs a little bit high.

5.3 Future Plan

We want to make the project portable, so we can carry it anywhere. We want to develop an android application for doctors and users also beautify our present website. We will try to add machine learning in patient's medical records for disease selection.

References

[1] Gómeza, J., Oviedob, B., & Zhumab, E. (2016). Patient Monitoring System Based on Internet of Things. *The 7th International Conference on Ambient Systems,Networks and Technologies* (pp. 90-97). Porto: Elsevier B.V.

[2] P. Wang, "The Real-Time Monitoring System for In-Patient Based on Zigbee," *2008 Second International Symposium on Intelligent Information Technology Application*, Shanghai, 2008, pp. 587-590.doi: 10.1109/IITA.2008.110.

[3] Abdelgawad, A., Yelamarthi, K., & Khattab, A. (2016). IoT-Based Health Monitoring System for Active and Assisted Living. *Smart Objects and Technologies for Social Good: Second International Conference*, (pp. 11-20). Venice.

[4] Tao Hu, Hao Zhang, Xinyan Zhu, Julaine Clunis, Gelan Yang, Depth sensor based human detection for indoor surveillance,Future Generation Computer Systems, Volume 88, 2018, Pages 540-551, ISSN 0167-739X, https://doi.org/10.1016/j.future.2018.05.083.

[5] UIU Digital Instituitional Repository System. (n.d.). [online] Available at: http://dspace.uiu.ac.bd/ [Accessed 1 Feb. 2019].

[6] University of Southampton Institutional Repository. (n.d.). [online] Available at: https://eprints.soton.ac.uk/ [Accessed 1 Feb. 2019].

[7] Robotdyn. (n.d.). *Digital Temperature Sensor - DS18B20, one wire*. [online] Available at: https://robotdyn.com/digital-temperature-sensor-ds18b20-one-wire.html [Accessed 1 Feb. 2019].

[8] WIKIPEDIA. (n.d.). *NodeMCU*. [online] Available at: https://en.wikipedia.org/wiki/NodeMCU [Accessed 4 Feb. 2019].

[9] WIKIPEDIA. (n.d.). *Radio-frequency identification*. [online] Available at: https://en.wikipedia.org/wiki/Radio-frequency_identification [Accessed 4 Feb. 2019].

Appendix A

Code for connection among sensors and Arduino Uno

```
#include "SPI.h"
#include "MFRC522.h"
#include <SoftwareSerial.h>
#include <DS3231.h>
#include <Wire.h>
#include "MAX30100_PulseOximeter.h"
#include <OneWire.h>
#include <DallasTemperature.h>

#define REPORTING_PERIOD_MS     1000
#define SS_PIN 10
#define RST_PIN 9

int pulse = 0;
int oxygen;
int arr[30];
int average;
int sum = 0;
int i = 0;
String date = "";
String puls = "";
String oxi = "";
int temp_sensor = 5;       // Pin DS18B20 Sensor is connected to
float temperature = 0;      //Variable to store the temperature in
float maxTemp = 0;

PulseOximeter pox;

uint32_t tsLastReport = 0;

DS3231  rtc(SDA, SCL);

OneWire oneWirePin(temp_sensor);

DallasTemperature sensors(&oneWirePin);

MFRC522 rfid(SS_PIN, RST_PIN);

MFRC522::MIFARE_Key key;

SoftwareSerial ArduinoUno(3,4);

// Callback (registered below) fired when a pulse is detected
void onBeatDetected()
{
```

```
    Serial.println("Beat!");
    digitalWrite(7,HIGH);
    delay(10);
    digitalWrite(7,LOW);
    delay(10);
}


void setup()
{
    Serial.begin(115200);
    sensors.begin();
    ArduinoUno.begin(4800);
    SPI.begin();
    rfid.PCD_Init();
    rtc.begin();
    pinMode(6, OUTPUT);
    pinMode(7, OUTPUT);

    // The following lines can be uncommented to set the date and time
    //rtc.setDOW(SATURDAY);     // Set Day-of-Week to SUNDAY
    //rtc.setTime(04, 52, 00);     // Set the time to 12:00:00 (24hr format)
    //rtc.setDate(12, 01, 2019);   // Set the date to January 1st, 2014
    Serial.print("Initializing pulse oximeter..");
    if (!pox.begin()) {
        Serial.println("FAILED");
        for(;;);
    } else {
        Serial.println("SUCCESS");
    }

    // 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 > REPORTING_PERIOD_MS) {
        pulse = pox.getHeartRate();
        oxygen = pox.getSpO2();

        if(pulse>30 && pulse<100){
          //Serial.println(pulse);
          if(i<20){
            arr[i] = pulse;
            i++;
```

```
      }
      //Serial.println(i);
    }

    if( i==20){
      for(int j=0;j<20;j++){
        sum = sum + arr[j];
      }
      average = sum/20;
      Serial.print("Average Pulse: ");
      Serial.println(average);
      puls = String(average);
      oxi = String(oxygen);
      //pulse = 0;
      //average = 0;
      //sum = 0;
      i++;
      digitalWrite(6,HIGH);
    }

    tsLastReport = millis();
}

if(i >= 20){
  sensors.requestTemperatures();

  temperature = sensors.getTempCByIndex(0);

  if(maxTemp < temperature){
    maxTemp = temperature;
    //Serial.print("Temperature is ");
    //Serial.println(maxTemp);
    delay(50);
    pulse = 0;

  }
  //------------------For RFID Sensor----------------------//
  if (!rfid.PICC_IsNewCardPresent() || !rfid.PICC_ReadCardSerial())
    return;
  // Serial.print(F("PICC type: "));
  MFRC522::PICC_Type piccType = rfid.PICC_GetType(rfid.uid.sak);
  // Serial.println(rfid.PICC_GetTypeName(piccType));
  // Check is the PICC of Classic MIFARE type
  if (piccType != MFRC522::PICC_TYPE_MIFARE_MINI &&
    piccType != MFRC522::PICC_TYPE_MIFARE_1K &&
    piccType != MFRC522::PICC_TYPE_MIFARE_4K) {
    Serial.println(F("Your tag is not of type MIFARE Classic."));
    return;
  }
```

```
    String strID = "";
    for (byte i = 0; i < 4; i++) {
      strID +=
      (rfid.uid.uidByte[i] < 0x10 ? "0" : "") +
      String(rfid.uid.uidByte[i], HEX) +
      (i!=3 ? ":" : "");
    }
    strID.toUpperCase();
  //-------------Date------------//

    maxTemp = (maxTemp*1.8)+32;

    date = rtc.getDateStr();
    //Time = rtc.getTimeStr();
    int len = date.length();
    char arr[len];
    String nDate = "";
    date.toCharArray(arr,len+1);
    for(int j=0;j<len;j++){
      if(j==2 || j==5){
        arr[j] = '-';
        nDate = nDate + arr[j];
      }
      else{
        nDate = nDate + arr[j];
      }
    }
    //Printing individual result
    Serial.println();
    Serial.print("Tag ID: ");
    Serial.println(strID);
    Serial.print("Temperature: ");
    Serial.println(maxTemp);
    Serial.print("Pulse: ");
    Serial.println(puls);
    Serial.print("Oxygen Level: ");
    Serial.println(oxi);
    Serial.print("Date: ");
    Serial.println(nDate);
    Serial.println();
    strID += String(maxTemp) + nDate + puls + oxi;
    maxTemp = 0.0;
    Serial.print("Tap card key: ");
    Serial.println(strID);
    ArduinoUno.print(strID);
    rfid.PICC_HaltA();
    rfid.PCD_StopCrypto1();
  }
}
```

Code for connection between Arduino UNO and NodeMCU

```
#include "SPI.h"
#include "MFRC522.h"
#include <SoftwareSerial.h>
#include <DS3231.h>
#include <Wire.h>
#include "MAX30100_PulseOximeter.h"
#include <OneWire.h>
#include <DallasTemperature.h>

#define REPORTING_PERIOD_MS     1000
#define SS_PIN 10
#define RST_PIN 9

int pulse = 0;
int oxygen;
int arr[30];
int average;
int sum = 0;
int i = 0;
String date = "";
String puls = "";
String oxi = "";
int temp_sensor = 5;       // Pin DS18B20 Sensor is connected to
float temperature = 0;      //Variable to store the temperature in
float maxTemp = 0;

PulseOximeter pox;

uint32_t tsLastReport = 0;

DS3231  rtc(SDA, SCL);

OneWire oneWirePin(temp_sensor);

DallasTemperature sensors(&oneWirePin);

MFRC522 rfid(SS_PIN, RST_PIN);

MFRC522::MIFARE_Key key;

SoftwareSerial ArduinoUno(3,4);

// Callback (registered below) fired when a pulse is detected
void onBeatDetected()
{
   Serial.println("Beat!");
   digitalWrite(7,HIGH);
   delay(10);
```

```
    digitalWrite(7,LOW);
    delay(10);
}


void setup()
{
    Serial.begin(115200);
    sensors.begin();
    ArduinoUno.begin(4800);
    SPI.begin();
    rfid.PCD_Init();
    rtc.begin();
    pinMode(6, OUTPUT);
    pinMode(7, OUTPUT);

    // The following lines can be uncommented to set the date and time
    //rtc.setDOW(SATURDAY);     // Set Day-of-Week to SUNDAY
    //rtc.setTime(04, 52, 00);     // Set the time to 12:00:00 (24hr format)
    //rtc.setDate(12, 01, 2019);   // Set the date to January 1st, 2014
    Serial.print("Initializing pulse oximeter..");
    if (!pox.begin()) {
        Serial.println("FAILED");
        for(;;);
    } else {
        Serial.println("SUCCESS");
    }

    // 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 > REPORTING_PERIOD_MS) {
        pulse = pox.getHeartRate();
        oxygen = pox.getSpO2();

        if(pulse>30 && pulse<100){
          //Serial.println(pulse);
          if(i<20){
            arr[i] = pulse;
            i++;
          }
          //Serial.println(i);
        }
```

```
    if( i==20){
      for(int j=0;j<20;j++){
        sum = sum + arr[j];
      }
      average = sum/20;
      Serial.print("Average Pulse: ");
      Serial.println(average);
      puls = String(average);
      oxi = String(oxygen);
      //pulse = 0;
      //average = 0;
      //sum = 0;
      i++;
      digitalWrite(6,HIGH);
    }

    tsLastReport = millis();
}

if(i >= 20){
  sensors.requestTemperatures();

  temperature = sensors.getTempCByIndex(0);

  if(maxTemp < temperature){
    maxTemp = temperature;
    //Serial.print("Temperature is ");
    //Serial.println(maxTemp);
    delay(50);
    pulse = 0;

  }

//------------------For RFID Sensor----------------------//
  if (!rfid.PICC_IsNewCardPresent() || !rfid.PICC_ReadCardSerial())
    return;

  // Serial.print(F("PICC type: "));
  MFRC522::PICC_Type piccType = rfid.PICC_GetType(rfid.uid.sak);
  // Serial.println(rfid.PICC_GetTypeName(piccType));

  // Check is the PICC of Classic MIFARE type
  if (piccType != MFRC522::PICC_TYPE_MIFARE_MINI &&
    piccType != MFRC522::PICC_TYPE_MIFARE_1K &&
    piccType != MFRC522::PICC_TYPE_MIFARE_4K) {
    Serial.println(F("Your tag is not of type MIFARE Classic."));
    return;
  }
```

```
  String strID = "";
  for (byte i = 0; i < 4; i++) {
    strID +=
    (rfid.uid.uidByte[i] < 0x10 ? "0" : "") +
    String(rfid.uid.uidByte[i], HEX) +
    (i!=3 ? ":" : "");
  }
  strID.toUpperCase();
//-------------Date------------//

  maxTemp = (maxTemp*1.8)+32;

  date = rtc.getDateStr();
  //Time = rtc.getTimeStr();
  int len = date.length();
  char arr[len];
  String nDate = "";
  date.toCharArray(arr,len+1);
  for(int j=0;j<len;j++){
    if(j==2 || j==5){
      arr[j] = '-';
      nDate = nDate + arr[j];
    }
    else{
      nDate = nDate + arr[j];
    }
  }

  //Printing individual result
  Serial.println();
  Serial.print("Tag ID: ");
  Serial.println(strID);
  Serial.print("Temperature: ");
  Serial.println(maxTemp);
  Serial.print("Pulse: ");
  Serial.println(puls);
  Serial.print("Oxygen Level: ");
  Serial.println(oxi);
  Serial.print("Date: ");
  Serial.println(nDate);
  Serial.println();

  strID += String(maxTemp) + nDate + puls + oxi;
  maxTemp = 0.0;
  Serial.print("Tap card key: ");
  Serial.println(strID);
  ArduinoUno.print(strID);
  rfid.PICC_HaltA();
  rfid.PCD_StopCrypto1();
}
```