Cloud Based Secured Health Care Monitoring and Medicine Reminder System

J Component Project Report for

ECE4009 – Wireless and Mobile Communication
(G1+TG1)

Bachelor of Technology
In
Electronics and Communication Engineering
By

Pranoy Dev (16BEC0098)

Under the guidance of

Prof. Budhatiya Bhattacharrya

School of Electronics Engineering
Vellore Institute of Technology, Vellore –

FALL 2019

DECLARATION BY THE CANDIDATE

I hereby declare that the project report entitled **"MEDICINE REMINDER AND BODY MONITORING SYSTEM"** submitted by me to VIT University, Vellore in partial fulfilment of the requirement for the award of the degree of **B.Tech. (Electronics and Communication Engineering)** is a record of J component of project work carried out by me under the guidance **of Prof. Budhatiya Bhattacharrya.** I further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Vellore

CERTIFICATE

This is to certify that the project work titled "***System Cloud Based Secured Health Care Monitoring and Medicine Reminder System***" that is being submitted by

PRANOY DEV 16BEC0098

for Wireless and Mobile Communication (ECE4009) is a record of bonafide work done under my supervision. The contents of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted for any other CAL course.

Place: Vellore

INTRODUCTION

Motivation

This report describes the development of an automatic medicine reminder and a body monitoring system project at Vellore Institute of Technology. The primary aim of our project is to develop a system that can be used for people, mostly the aged, who can use this system to their high convenience. The project is focused on helping the needy to which, at times, becomes a problem in the fast and busy world today. We also send different human body parameters like body temperature, blood pressure, blood oxygen level and the humidity in the atmosphere around the patient to the respective doctor developing efficient perception, sensor fusion, heterogeneous swarm technologies, distributed command and control, and task decomposition.

The human body monitoring system is capable of monitoring a patient in areas where human cannot reach or are not permitted. It is reprogrammable and can be interchanged to provide multiple applications.

Objective

1. Automating Medicine intake process and health caution system
2. Reduce the time invested in taking external scans and checkups by implementing real-time sensor monitoring for important organs
3. Making the guardian know about the due medicines to be taken by the patient

Organization of the report

- Project description and Goals – Precise and brief description of the project and the goals achieved with this project.
- Technical specification – List of all the components used for this project and the pictures of the products used for a better reference.
- Design approach and details – Detailed block diagram of the project and the codes and standards followed in this project.
- Pictures of setup – Visual presentation of the project along with the components
- Reference – Inspiration and ideation web URL’s for this project

PROJECT DESCRIPTION

The Purpose of the medicine reminder and body monitoring system is to assist the elderly. As many of them live alone at home and is difficult for them to go and consult a doctor on a regular basis. With majority of the young generation moving to big cities and going far away from their parents this issue intensifies. The human body monitoring system is capable of monitoring a patient in areas where human cannot reach or are not permitted. It is reprogrammable and can be interchanged to provide multiple applications. Here the user is an old person. When the medical kit is received by the person he/she may enter the current time of the place. Now later as per the persons medical condition certain medicines would be given to him/her. The elder person can himself/herself enter the time of the medicine or the doctor can do it with the help of an app. The app also shows vital signs of the elderly person like heart rate, blood oxygen level, body temperature and humidity so as the doctor can keep on monitoring the patient's health improvement from a remote location. In situations if the elderly person lives in a town or city other than his family the access to his information can also be available to his/her family members.

The device is composed of two parts one is the medicine reminder part which is connected to the Arduino Uno. This part is responsible for collecting the data of the time to take the medicines. It compares the current time and the time when the medicine needs to be taken. When it is time to take the medicine it sends a signal to then node Mcu. This signal to the Node Mcu is the sent as a reminder in the app. The Arduino Uno is also connected to the LCD display and an hex key pad. The LCD display is used to show the current time and also to signal to a person when to take the medicine and the hex key pad is used as an input device to interact with the health kit.

The node Mcu the one responsible to pushing the sensor data into the internet over the WIFI. The node MCU is connected to various sensors like the pulse sensor, temperature sensor and the humidity. It also talks to the Arduino Uno. When using the app the date of the medicine is changed it sends this data to the Arduino Uno so as it updates the time.

GOALS

Our ultimate goal was to develop technologies that extend the sphere of awareness and mobility of an individual or group performing such operations, by increasing sensory "reach" and providing the ability to task multiple platforms as a single logical entity. The Human Body Monitoring System and Medicine Reminder seeks to augment human capabilities, reduce exposure to risk, and present timely, relevant information to the user.

PROPOSED TECHNICAL SOLUTION

- Health monitoring kit.
- Using app remote aces to ones vitals available.
- During any medical illness detection by the kit a warning message sent to doctor.
- Doctor can monitor if patient is taking medicine or not
- Reduction of medical cost to individuals.

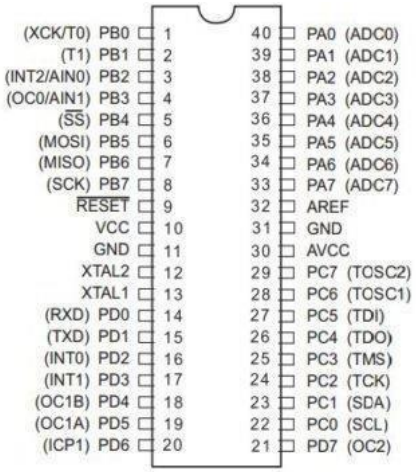
HARDWARE SPECIFICATIONS

1. NodeMCU -

NodeMCU 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 Espress if Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson, and spiffs.

2. Arduino UNO –

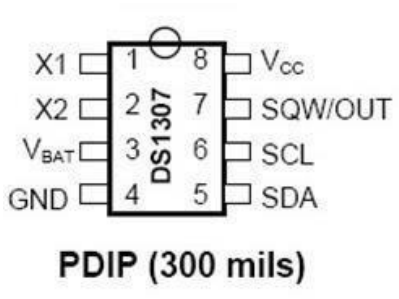
The Arduino UNO is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 Digital pins, 6 Analog pins, and programmable with the Arduino IDE (Integrated Development Environment) via a type B USB cable. It can be powered by a USB cable or by an external 9 volt



battery, though it accepts voltages between 7 and 20 volts. It is also similar to the Arduino Nano and Leonardo.

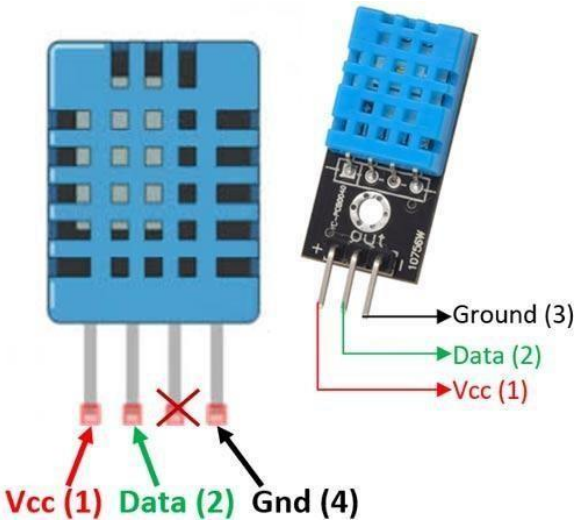
3. RTC DS1307 –

Real time clocks (RTC), as the name recommends are clock modules. The DS1307 real time clock (RTC) IC is an 8 pin device using an I2C interface. The DS1307 is a low-power clock/calendar with 56 bytes of battery backup SRAM. The clock/calendar provides seconds, minutes, hours, day, date, month and year qualified data. The end date of each month is automatically adjusted, especially for months with less than 31 days.



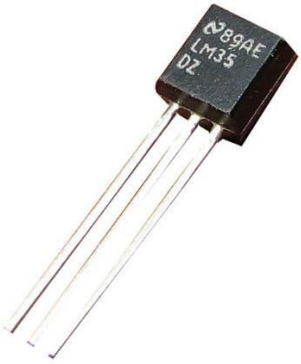
4. DHT11 –

The DHT11 is a basic, ultra low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin (no analog input pins needed). Its fairly simple to use, but requires careful timing to grab data. The only real downside of this sensor is you can only get new data from it once every 2 seconds, so when using our library, sensor readings can be up to 2 seconds old.



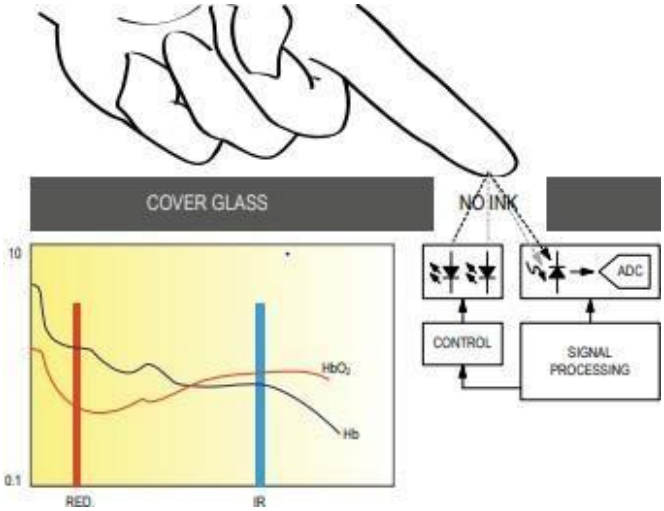
5. LM35 –

The LM35 series are precision integrated-circuit temperature devices with an output voltage linearly-proportional to the Centigrade temperature. The LM35 device has an advantage over linear temperature sensors calibrated in Kelvin, as the user is not required to subtract a large constant voltage from the output to obtain convenient Centigrade scaling. The LM35 device does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C over a full −55°C to 150°C temperature range. Lower cost is assured by trimming and calibration at the wafer level. The low-output impedance, linear output, and precise inherent calibration of the LM35 device makes interfacing to readout or control circuitry especially easy.



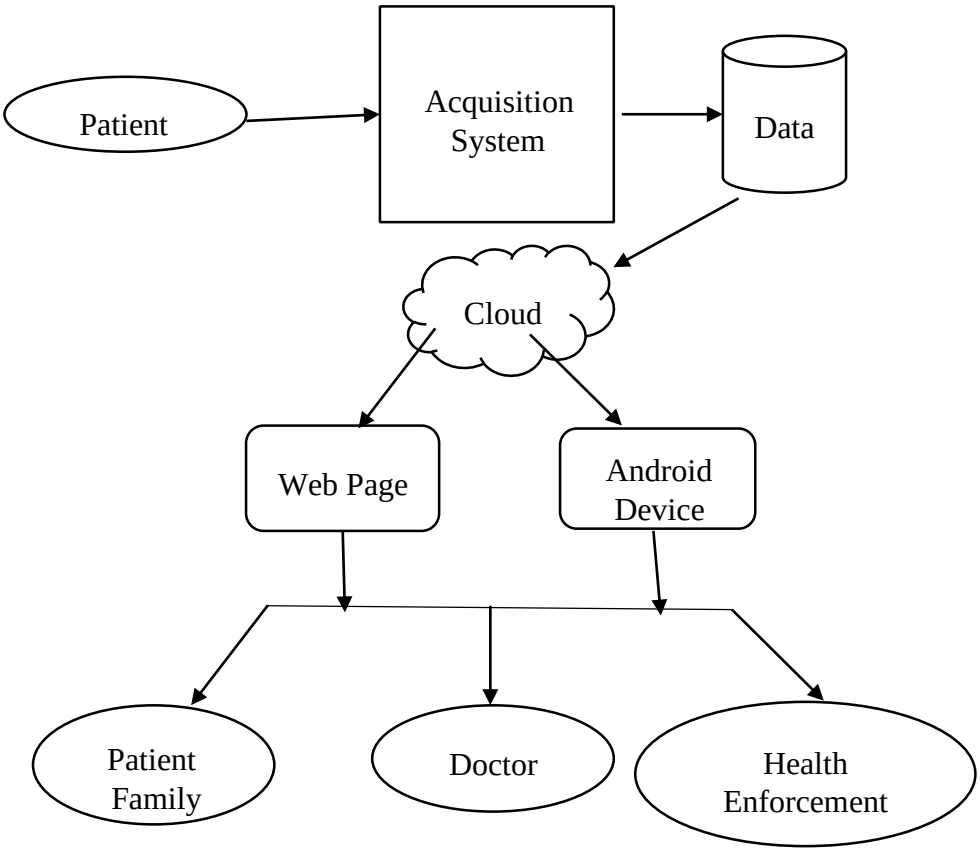
6. Pulse sensor (MAX30100)–

Pulse Sensor is a low cost, very small size a plug-and-play heart rate sensor for Arduino and Arduino compatible boards. It can be used by students, artists, athletes, makers, and game & mobile developers who want to easily incorporate live heart-rate data into their projects.

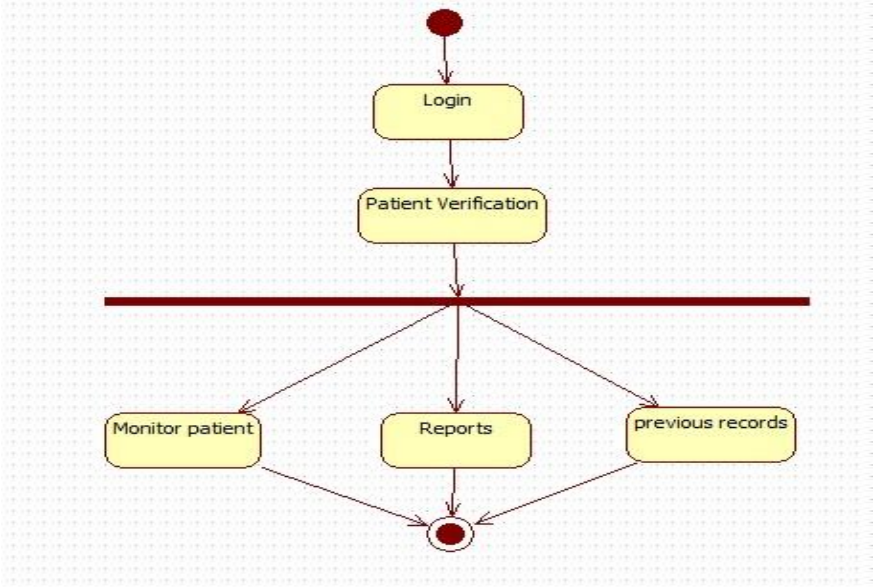


Pulse Sensor Amped adds amplification and noise cancellation circuitry to the hardware. It's noticeably faster and easier to get reliable pulse readings. Pulse Sensor works with either a 3V or 5V Arduino.

SYSTEM ARCHITECTURE

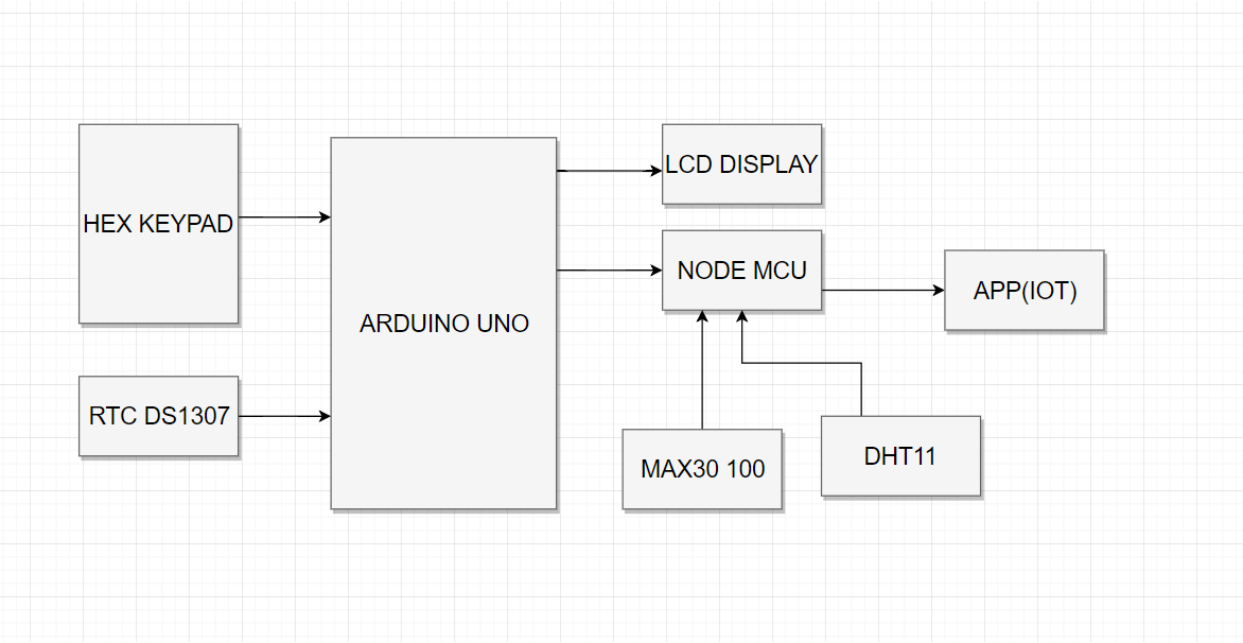


State chart Diagram



DESIGN APPROACH AND DETAILS

Block diagram:



The Arduino board is directly connected to the LCD and the clock. This module maintains the time and processes the sensor data sent from ESP8266. The Wifi module is directly connected to the humidity, temperature and pulse sensors. This module sends the data to the application in mobile. The Arduino board gives any notifications based on time and data sent to it from ESP8266.

CODES AND STANDARDS

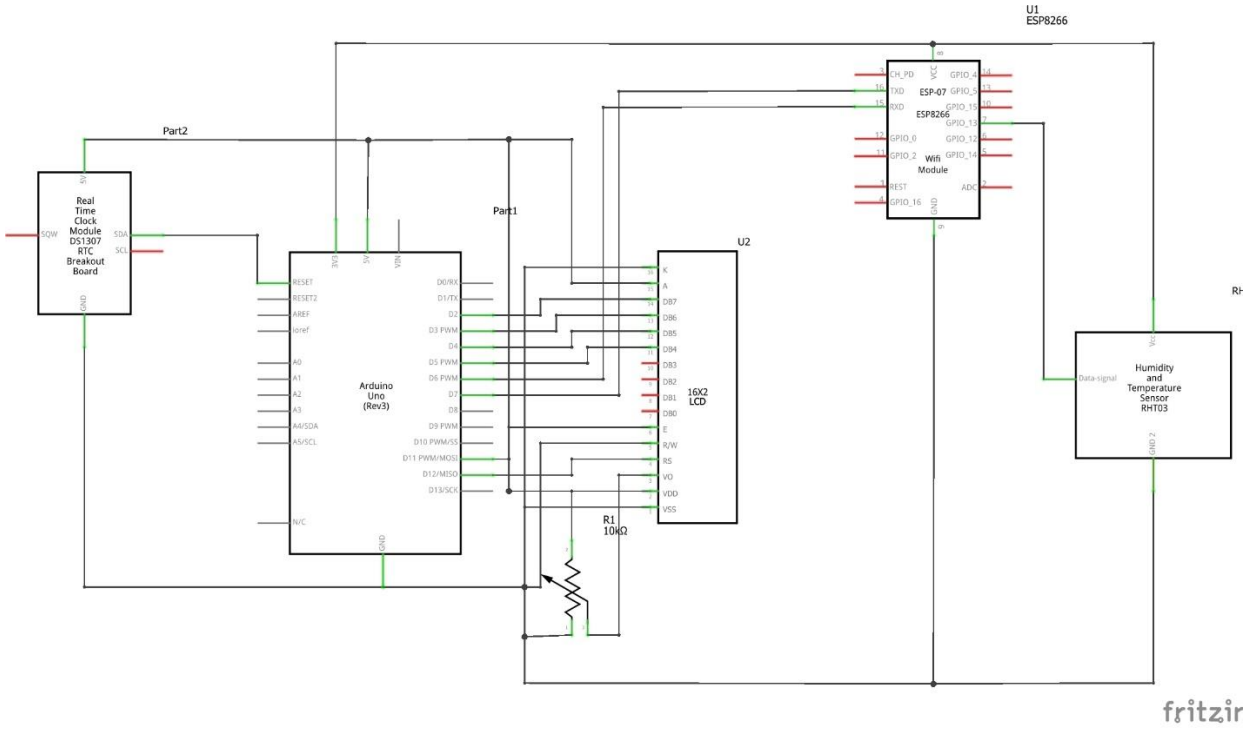
1. **I2C Protocol** is a serial bus protocol consisting of two signal lines such as SCL and SDL lines which are used to communicate with the devices. The SCL stands for a 'serial clock line' and this signal is always driven by the 'master device'. The SDL stands for the 'serial data line', and this signal is driven by either the master or the I2C peripherals. Both these SCL and SDL lines are in open-drain state when there is no transfer between I2C peripherals.

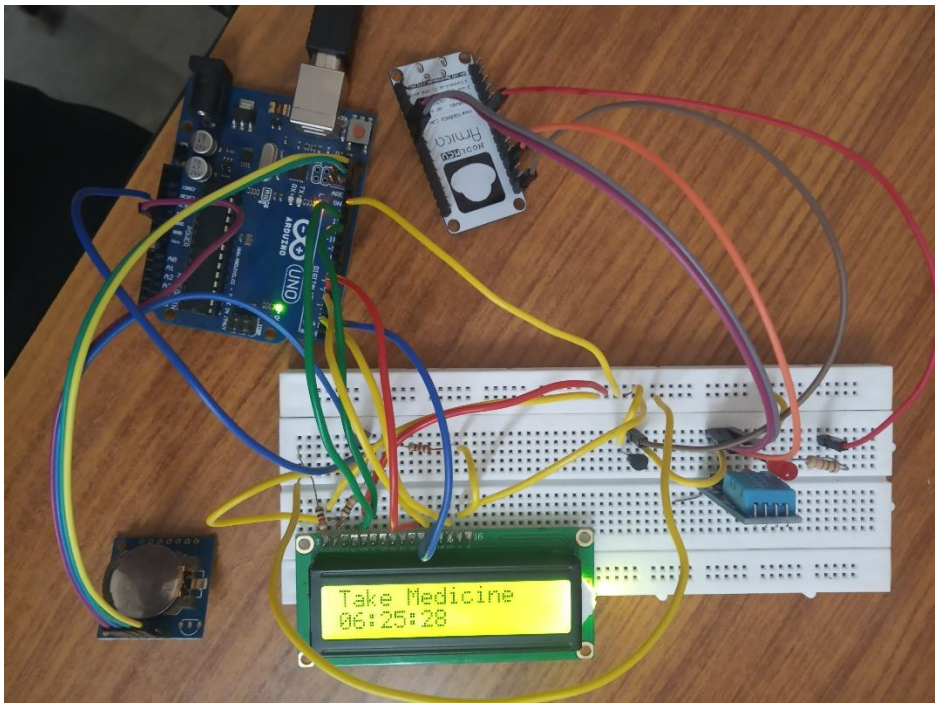
2. **SERIAL COMMUNCATION** In telecommunication and data transmission, serial communication is the process of sending data one bit at a time, sequentially, over a communication channel or computer bus. This is in contrast to parallel communication, where several bits are sent as a whole, on a link with several parallel channels.

3. **IPV4** Internet Protocol version 4 (IPv4) is the fourth version of the Internet Protocol (IP). It is one of the core protocols of standards-based internetworking methods in the Internet, and was the first version deployed for production in the ARPANET in 1983. It still routes most Internet traffic today,[1] despite the ongoing deployment of a successor protocol, IPv6. IPv4 is described in IETF publication RFC 791 (September 1981), replacing an earlier definition (RFC 760, January 1980).IPv4 is a connectionless protocol for use on packet-switched networks. It operates on a best effort delivery model, in that it does not guarantee delivery, nor does it assure proper sequencing or avoidance of duplicate delivery. These aspects, including data integrity, are addressed by an upper layer transport protocol, such as the Transmission Control Protocol (TCP).

4. **Blynk** it is a company that provides ipv4 server system and provides an API to enable to send and receive data over the internet.

PICTURES OF THE SETUP





PSUEDO CODE:

Code for NodeMCU ESP8266

```
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h>
#include <SoftwareSerial.h>
//SoftwareSerial s(D6,D5); // (Rx, Tx)
// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "7d41f82ba9724b4681d1a7b8d3b648e5";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "Redmi";
char pass[] = "Anubhav97";
int tempPin = A0;

#define DHTPIN 0          // D3
#define DHTTYPE DHT11     // DHT 11

DHT dht(DHTPIN, DHTTYPE);
BlynkTimer timer;

// This function sends Arduino's up time every second to Virtual Pin (5).
// In the app, Widget's reading frequency should be set to PUSH. This means
// that you define how often to send data to Blynk App.
void sendSensor()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature(); // or dht.readTemperature(true) for Fahrenheit

  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  // You can send any value at any time.
  // Please don't send more that 10 values per second.
  Blynk.virtualWrite(V5, t);
  Blynk.virtualWrite(V6, h);
}

void setup()
{
```

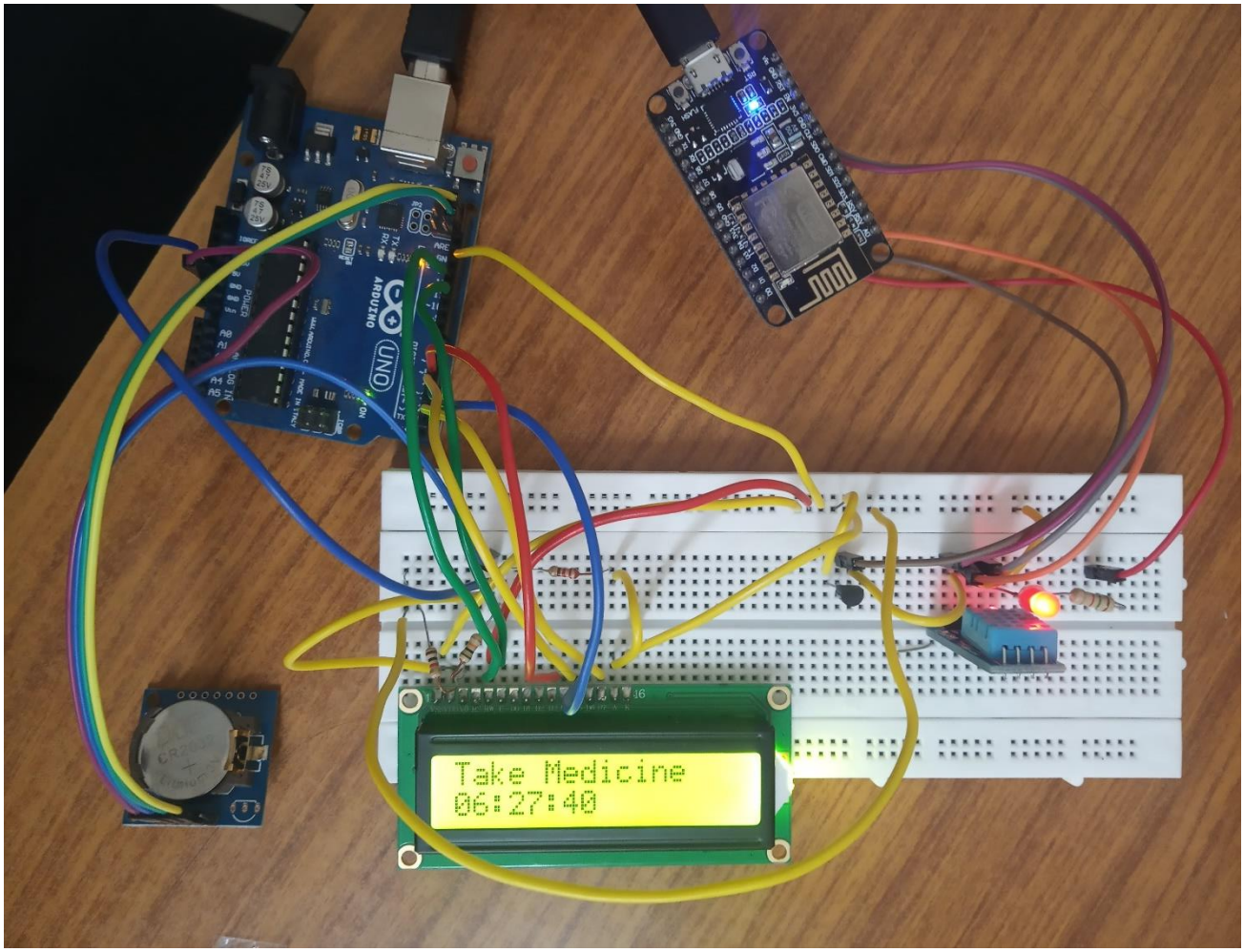
```
  // Debug console
  Serial.begin(9600);



pinMode(A0,INPUT);
  Blynk.begin(auth, ssid, pass);
  dht.begin();
  // Setup a function to be called every second
  timer.setInterval(1000L, sendSensor);
}

void loop()
{
  Blynk.run();
  timer.run();
  /*s.write("s");
  if (s.available()>0)
  {
    int data=s.read();
    Serial.println(data);
  }*/
}
```

Code for Arduino Uno

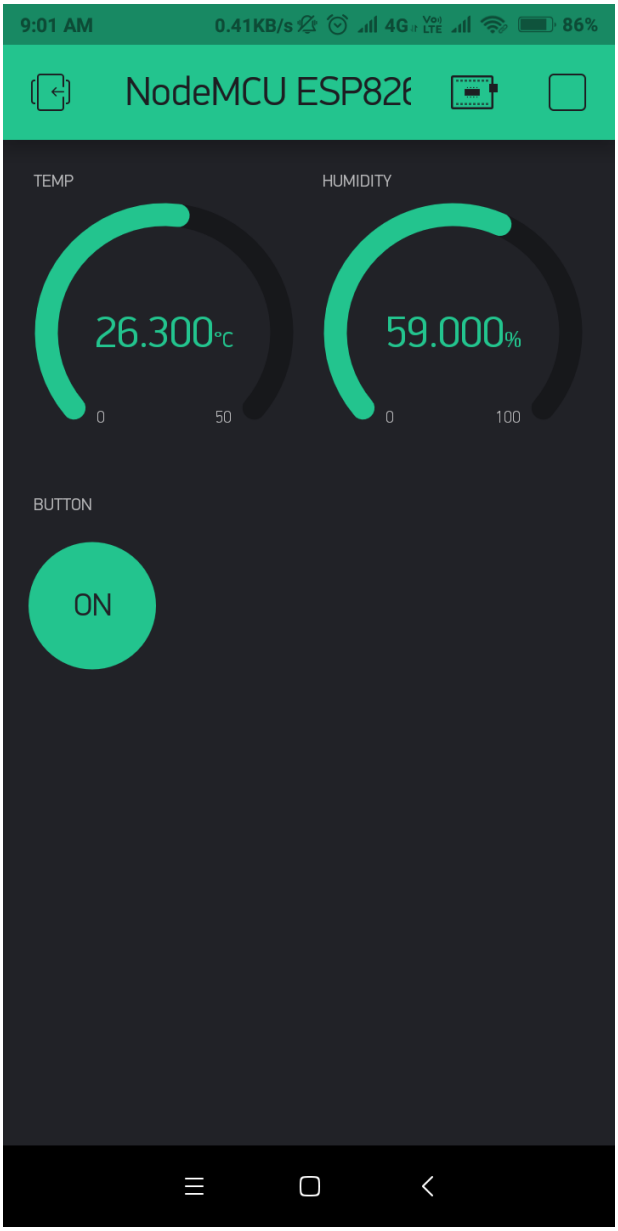
```
#include<Wire.h>
#include <SoftwareSerial.h>
#include "RTClib.h"
#include <LiquidCrystal.h>
RTC_DS1307 RTC;

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup ()
{
Serial.begin(9600);
Wire.begin();
RTC.begin(); // load the time from your computer.
if (! RTC.isrunning())
{
Serial.println("RTC is NOT running!");// This will reflect the time that your sketch was compiled
RTC.adjust(DateTime(__DATE__, __TIME__));
}
lcd.begin(16, 2);
```

```
lcd.print("Start");

void loop ()
{
DateTime now = RTC.now();
Serial.print(now.month(), DEC);
Serial.print('/');
Serial.print(now.day(), DEC);
Serial.print('/');
Serial.print(now.year(), DEC);
Serial.print(' ');
Serial.print(now.hour(), DEC);
Serial.print(':');
Serial.print(now.minute(), DEC);
Serial.print(':');
Serial.print(now.second(), DEC);
Serial.println();

   lcd.setCursor(0, 0);
   //lcd.print("Digital Clock");
   lcd.setCursor(0, 0);
   lcd.print(now.day(), DEC);
   lcd.print('/');
   lcd.print(now.month(), DEC);
   lcd.print('/');
   lcd.print(now.year(), DEC);
   lcd.print(' ');
   lcd.setCursor(0, 2);
    if (now.hour()<10)
   lcd.print('0');
   lcd.print(now.hour(), DEC);
   lcd.print(':');
    if (now.minute()<10)
   lcd.print('0');
   lcd.print(now.minute(), DEC);
   lcd.print(':');
   if (now.second()<10)
   lcd.print('0');
   lcd.print(now.second(), DEC);
   lcd.setCursor(12, 0);

if(now.hour()==6)
{
   lcd.setCursor(0,0);
  lcd.print("Take Medicine");
}
delay(1000);
}
```

Output:



Output and Circuit Diagram



Output in mobile

Conclusion

Health monitoring is the major problem in today's world. Due to lack of proper health monitoring, patient suffer from serious health issues. There are lots of IoT devices now days to monitor the health of patient over internet. Health experts are also taking advantage of these smart devices to keep an eye on their patients.

The project can be improved by using a greater number and variety of sensors. Along with that, platforms other than blynk can be used. Better algorithms which can analyze the patient based on the sensor data can be used to provide better diagnoses and improve the overall monitoring process

References:

- http://denethor.wlu.ca/common/keypad.shtml https://www.elprocus.com/rtc-ds1307/
- https://learn.adafruit.com/ds1307-real-time-clock-breakout-board-kit/what-is-an-rtc https://www.adafruit.com/product/386
- http://artofcircuits.com/product/dht11-digital-temperature-humidity-sensor https://en.wikipedia.org/wiki/NodeMCU https://en.wikipedia.org/wiki/Atmel_AVR
- https://www.engineersgarage.com/electronic-components/atmega16-microcontroller https://www.microchip.com/wwwproducts/en/ATmega16