PROJECT REPORT

Submitted for DATA STRUCTURES AND ALGORITHMS (CSE2003)

By

Rohit Subramanian 17BCE1291
B.Shruti 17BCE1299
Jayakrishnan.L 14BEC1194

Slot : E1

Name of faculty : Dr .SALEENA B

(SCSE)

VIT
UNIVERSITY
(Estd. u/s 3 of UGC Act 1956)
VELLORE ■ CHENNAI
www.vit.ac.in

MARCH, 2018

CERTIFICATE

This is to certify that the Project work entitled " **Elderly Health Monitor System**" that is being submitted by "**B Shruti**" for CAL in BTech Data Structures and Algorithms CSE2003 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 : Chennai

Date :

Signature of Student: (B Shruti)

Signature of Faculty: (Dr. Nirmal Thyagu)

ACKNOWLEDGEMENTS

It is my privilege to express my sincere regards to my dean for providing me this wonderful opportunity to learn.

First of all I would like to thank my friends and family for giving encouragement, enthusiasm and invaluable motivation to me. Without all this I might have not been able to complete this project.

I deeply express my sincere thanks to Dr. B.Saleena for her valuable inputs, able guidance, encouragement and valuable advice and criticism that helped me complete this project.

B Shruti
Rohit Subramanian
Jayakrishnan L

ABSTRACT

According to research done, approximately 2000 people died monthly due to carelessness of their health and most of these have been in the old age and working age sector. This is because they don't have time for themselves and forget about their health management due to a heavy workload or just negligence which has become pretty common in this generation. So, I have decided to make an internet of things based healthcare project for the healthcare units that have to keep a regular eye on the elderly and bed ridden patients. The project aims at measuring the pulse rate of the patient and temperature and humidity of the room the patient is admitted in. In case of any emergencies or critical situation, it alerts the nearest ambulance and hence might result in saving lives. The best part of the project is that it can be used by everyone and makes taking care of the patients easier and faster.

SOFTWARE/APPARATUS USED

1. Arduino Uno
2. DHT11 Sensor
3. Pulse Sensor
4. Breadboard
5. 16x2 LCD
6. LED lights
7. Connecting Wires
8. Potentiometer
9. Resistor
10. Battery
11. Fritzig
12. Dev C++

INTRODUCTION

The main goal of the project is to monitor the patient and the room he is admitted in. The project is mainly aimed at those people who are bed ridden and need constant monitoring. The project is being implemented by **Internet Of Things (IoT)**. The Internet of things (IoT) is the network of physical devices, vehicles, home appliances and other items embedded with electronics, software, sensors, actuators, and connectivity which enables these objects to connect and exchange data. Each thing is uniquely identifiable through its embedded computing system but is able to inter-operate within the existing Internet infrastructure. The IoT allows objects to be sensed or controlled remotely across existing network infrastructure, creating opportunities for more direct integration of the physical world into computer-based systems, and resulting in improved efficiency, accuracy and economic benefit in addition to reduced human intervention. When IoT is augmented with sensors and actuators, the technology becomes an instance of the more general class of cyber-physical systems, which also encompasses technologies such as smart grids, virtual power plants, smart homes, intelligent transportation and smart cities.

Arduino is an open source computer hardware and software company, project, and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical and digital world. The project's products are distributed as open-source hardware and software, which are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form, or as do-it-yourself (DIY) kits.
Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards or Breadboards (*shields*) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

The **great-circle distance** or orthodromic **distance** is the shortest **distance** between two points on the surface of a sphere, measured along the surface of the sphere (as opposed to a straight line through the sphere's interior).

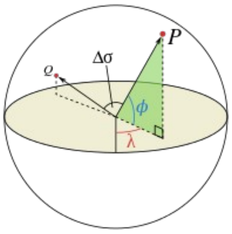
$$\begin{cases} \Delta\sigma = \arccos\left(\sin\phi_1\cdot\sin\phi_2+\cos\phi_1\cdot\cos\phi_2\cdot\cos(\Delta\lambda)\right)\\ d \;\; = r\,\Delta\sigma \end{cases}$$

Where (ϕ_1, λ_1) and (ϕ_2, λ_2) is the latitude and longitude of point A and point B

$\Delta\phi, \Delta\lambda$ are the absolute differences

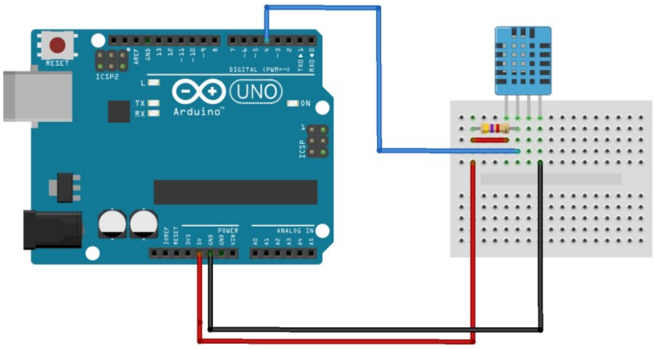
r is the mean radius of the Earth (= 6731 km , WGS84 ellipsoid definition)

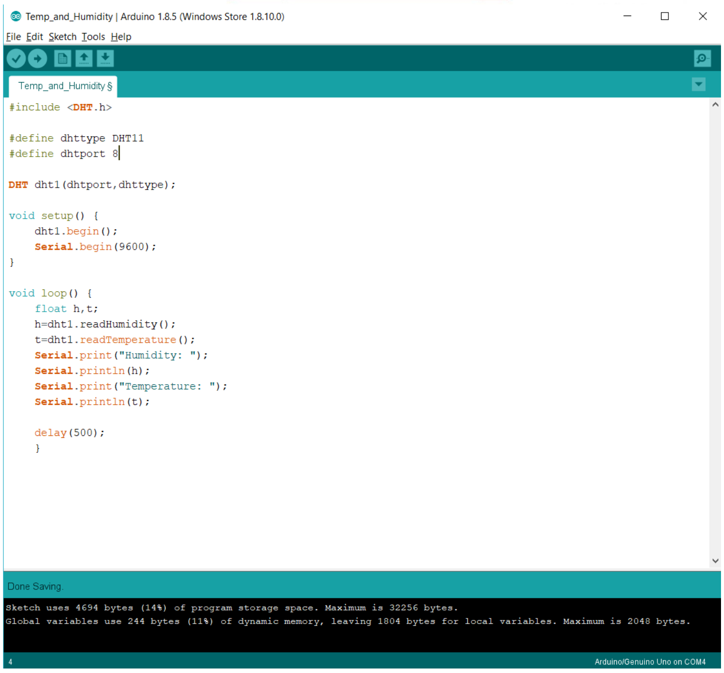
d is the surface distance between the two points



METHODOLOGY

1.TEMPERATURE AND HUMIDITY SENSOR (DHT 11)



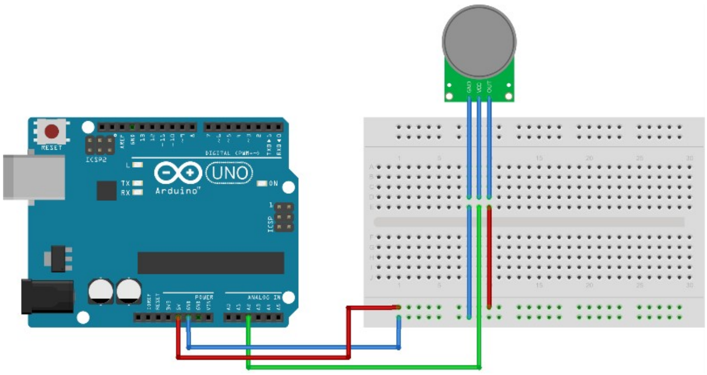

Temp_and_Humidity | Arduino 1.8.5 (Windows Store 1.8.10.0)
File Edit Sketch Tools Help
Temp_and_Humidity §

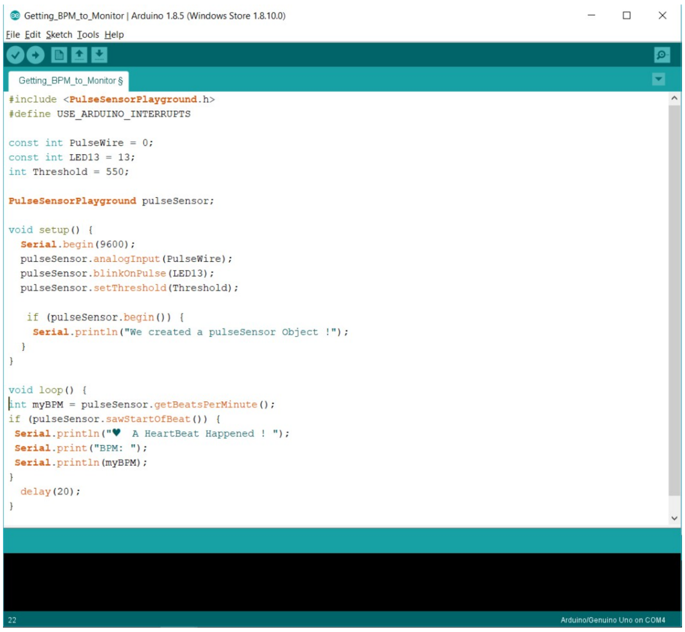
```
#include <DHT.h>

#define dhttype DHT11
#define dhtport 8

DHT dht1(dhtport,dhttype);

void setup() {
    dht1.begin();
    Serial.begin(9600);
}

void loop() {
    float h,t;
    h=dht1.readHumidity();
    t=dht1.readTemperature();
    Serial.print("Humidity: ");
    Serial.println(h);
    Serial.print("Temperature: ");
    Serial.println(t);

    delay(500);
    }
```

Done Saving.

```
Sketch uses 4694 bytes (14%) of program storage space. Maximum is 32256 bytes.
Global variables use 244 bytes (11%) of dynamic memory, leaving 1804 bytes for local variables. Maximum is 2048 bytes.
```

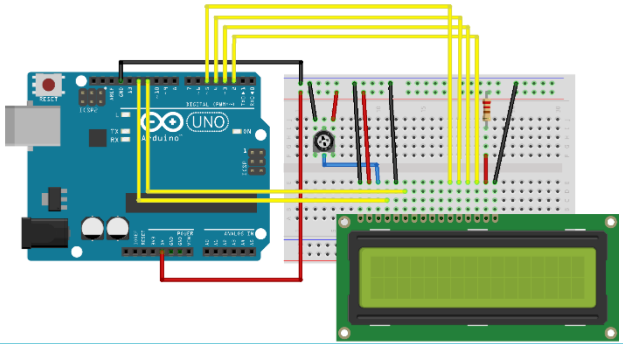
2.PULSE RATE SENSOR

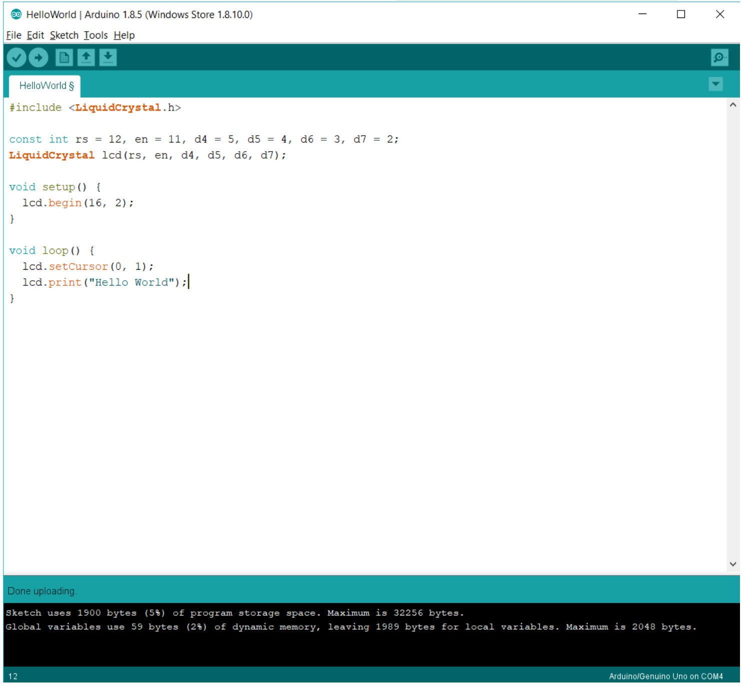



Getting_BPM_to_Monitor | Arduino 1.8.5 (Windows Store 1.8.10.0)
File Edit Sketch Tools Help
Getting_BPM_to_Monitor §
22
Arduino/Genuino Uno on COM4

```
#include <PulseSensorPlayground.h>
#define USE_ARDUINO_INTERRUPTS

const int PulseWire = 0;
const int LED13 = 13;
int Threshold = 550;

PulseSensorPlayground pulseSensor;

void setup() {
  Serial.begin(9600);
  pulseSensor.analogInput(PulseWire);
  pulseSensor.blinkOnPulse(LED13);
  pulseSensor.setThreshold(Threshold);

   if (pulseSensor.begin()) {
    Serial.println("We created a pulseSensor Object !");
  }
}

void loop() {
int myBPM = pulseSensor.getBeatsPerMinute();
if (pulseSensor.sawStartOfBeat()) {
 Serial.println("♥  A HeartBeat Happened ! ");
 Serial.print("BPM: ");
 Serial.println(myBPM);
}
  delay(20);
}
```

3.LCD DISPLAY




HelloWorld | Arduino 1.8.5 (Windows Store 1.8.10.0)
File Edit Sketch Tools Help
HelloWorld §
12
Arduino/Genuino Uno on COM4

```
#include <LiquidCrystal.h>

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

void setup() {
  lcd.begin(16, 2);
}

void loop() {
  lcd.setCursor(0, 1);
  lcd.print("Hello World");
}
```

```
Done uploading.
Sketch uses 1900 bytes (5%) of program storage space. Maximum is 32256 bytes.
Global variables use 59 bytes (2%) of dynamic memory, leaving 1989 bytes for local variables. Maximum is 2048 bytes.
```

<u>TIME COMPLEXITY(for all Arduino codes)-</u>

void setup() -O(1)

void loop() - O(n)

4.ALGORITHM FOR GREAT CIRCLE DISTANCE FORMULA

1. Obtain latitude and longitude of each ambulance in string format along with their user-id and name from the JSON encoded input file.
2. Convert latitude and longitude of the ambulance present in string format to double.
3. Convert latitude and longitude of both, the user and the ambulance present in degrees to radians.
4. Calculate distance between the user's location and the ambulance using Great Circle Distance formula.
5. If distance is found to be less than or equal to 50 kms then output the user-id and name of the ambulance driver to a new file else take no action.

```
//function to find dist
double distanceEarth(double lat2d, double lon2d)
{
   double lat1, lon1, lat2, lon2,
          delta_lon, central_ang;

   lat1 = degtorad(lat1d);
   lon1 = degtorad(lon1d);
   lat2 = degtorad(lat2d);
   lon2 = degtorad(lon2d);

   delta_lon = lon2 - lon1;

   // great circle distance formula.
   central_ang = acos ( sin(lat1) *
                 sin(lat2) + cos(lat1) *
                 cos(lat2) * cos(delta_lon) );

   return (earth_radius * central_ang);
}

// Function to convert degree to radian.
double degtorad(double deg)
{
   return ( deg * pi / 180);
}
```

TIME COMPLEXITY- O(n)

4.ALGORITHM FOR CRITICAL CONDITION

```
int critical=0;
if((t<20 || t>30) &&( h<50 || h>75))
{
        critical=1;
}
if(myBPM<40 || myBPM>120)
{
        critical=1;
}
if (critical==1)
{
        lcd.print(“CRITICAL”);
}
else
{
        lcd.print(“NORMAL”);
}
```

TIME COMPLEXITY-

```
if-then-else statements
if (condition) {
   sequence of statements 1
}
else {
   sequence of statements 2
}
```

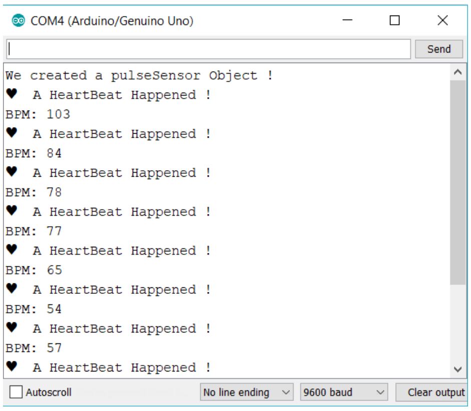
Here, either sequence 1 will execute, or sequence 2 will execute. Therefore, the worst-case time is the slowest of the two possibilities: max(time(sequence 1), time(sequence 2)). For example, if sequence 1 is O(N) and sequence 2 is O(1) the worst-case time for the whole if-then-else statement would be O(N).

SOURCE CODE

```
#include <DHT.h>

#define dhttype DHT11
#define dhtport 8
#define USE_ARDUINO_INTERRUPTS true
#include <PulseSensorPlayground.h>

const int PulseWire = 0;
int Threshold = 550;
int ledPin=5;

DHT dht1(dhtport,dhttype);
PulseSensorPlayground pulseSensor;
void setup() {
   dht1.begin();
   Serial.begin(9600);
   pinMode(ledPin,OUTPUT);
  pulseSensor.analogInput(PulseWire);
  pulseSensor.setThreshold(Threshold);
  pinMode(ledPin,OUTPUT);
   if (pulseSensor.begin()) {
    Serial.println("We created a pulseSensor Object !");
  }
}
```

```
void loop() {
   float h,t;
   h=dht1.readHumidity();
   t=dht1.readTemperature();
   Serial.print("Humidity: ");
   Serial.println(h);
   Serial.print("Temperature: ");
   Serial.println(t);
   int myBPM = pulseSensor.getBeatsPerMinute();
   int flag1=0,flag2=0;
 if (pulseSensor.sawStartOfBeat()) {
 Serial.println("♥  A HeartBeat Happened ! ");
 Serial.print("BPM: ");
 Serial.println(myBPM);
 if (myBPM<40 || myBPM>120)
 {
   Serial.println("Critical");
   flag1=1;
 }
 }
   if((t>25 || t<20)&& (h>75 || h<50))
   {flag2=1;}
   if(flag1==1 || flag2==1)
   digitalWrite(ledPin,HIGH);
   else
   digitalWrite(ledPin,LOW);

   delay(500);
   }
```

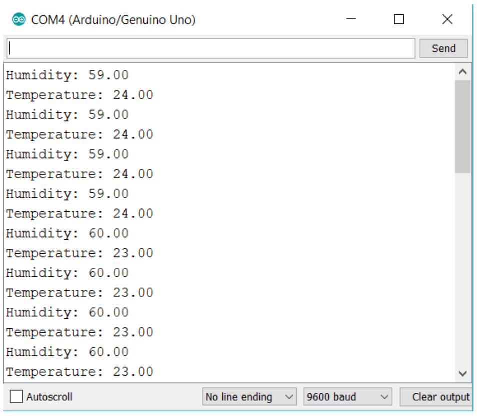
RESULTS AND DISCUSSION



No line ending

9600 baud

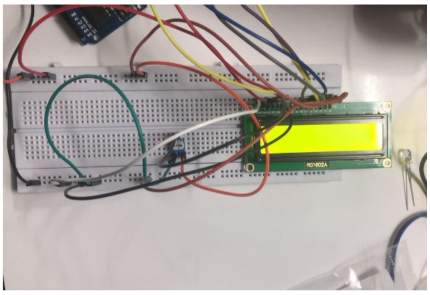
```
We created a pulseSensor Object !
♥  A HeartBeat Happened !
BPM: 103
♥  A HeartBeat Happened !
BPM: 84
♥  A HeartBeat Happened !
BPM: 78
♥  A HeartBeat Happened !
BPM: 77
♥  A HeartBeat Happened !
BPM: 65
♥  A HeartBeat Happened !
BPM: 54
♥  A HeartBeat Happened !
BPM: 57
♥  A HeartBeat Happened !
```

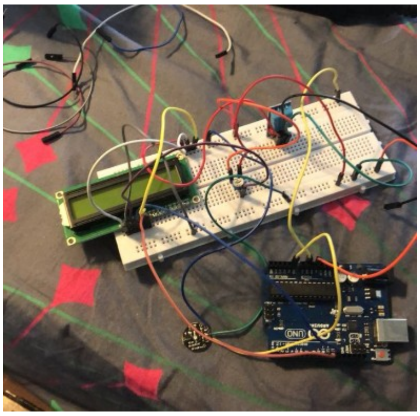


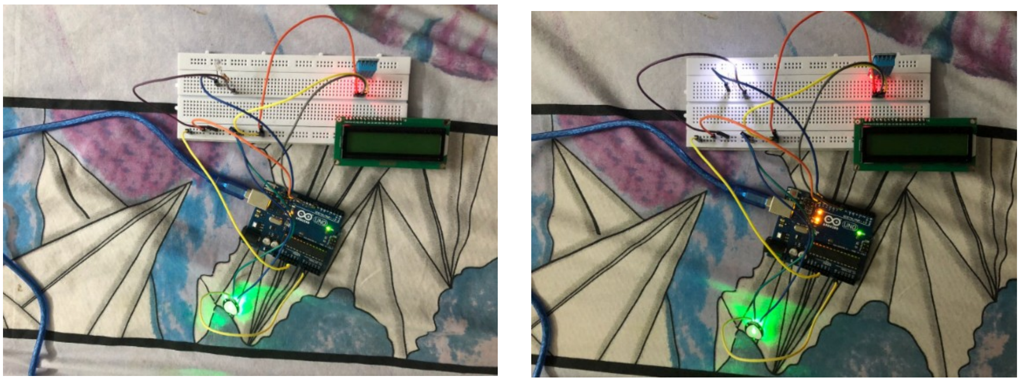
No line ending

9600 baud

```
Humidity: 59.00
Temperature: 24.00
Humidity: 59.00
Temperature: 24.00
Humidity: 59.00
Temperature: 24.00
Humidity: 59.00
Temperature: 24.00
Humidity: 60.00
Temperature: 23.00
Humidity: 60.00
Temperature: 23.00
Humidity: 60.00
Temperature: 23.00
Humidity: 60.00
Temperature: 23.00
```







LED to show Critical condition or not

CONCLUSION

The project “Health monitoring System” has been successfully designed and tested. It has been developed by integrating features of all the hardware components used. Presence of every module has been reasoned out and placed carefully thus contributing to the best working of the unit. Secondly, using Arduino and with the help of growing technology the project has been successfully implemented. The whole health monitoring system, which we have proposed can be integrated into a small compact unit as small as a cell phone or a wrist watch. This will help the patients to easily carry this device with them wherever they go.

The whole health monitor system can be made more efficient by using much sophisticated and state of the art sensors. Also it can be made into a compact wearable unit that can be easily used in everyday life by a person of any age.

REFERENCES

https://en.wikipedia.org/wiki/Great-circle_distance

https://www.geeksforgeeks.org/finding-cabs-nearby-using-great-circle-distance-formula/

https://www.arduino.cc/en/Tutorial/HomePage

https://github.com/adafruit/DHT-sensor-library/blob/master/DHT.h

https://pulsesensor.com/pages/installing-our-playground-for-pulsesensor-arduino

http://fritzing.org/projects/lcd-screen-setup-tutorial

*****