**SMART GLOVE FOR HEALTH MONITORING**

A Project report submitted

In partial fulfilment of the requirements

For the award of the Degree of

**BACHELOR OF TECHNOLOGY**

In

**ELECTRONICS & COMMUNICATIONS ENGINEERING**

by

**VISHAL VERMA (1603031196)**

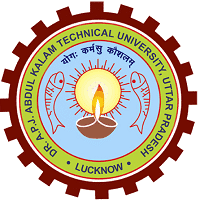
**KAJAL PATHAK (1603031077)**

**UJJAWAL CHANDELA (1603031184)**

**NISHANT KUMAR PATEL (1703031902)**

Under The Supervision Of

**Dr. SWATI VAID**



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING INDERPRASTHA ENGINEERING COLLEGE

Plot No 63, Site IV, Sahibabad, Ghaziabad, Uttar Pradesh 201010

Affiliated to Dr. A.P.J. Abdul Kalam Technical University, Lucknow

August 2018

**INDERPRASTHA ENGINEERING COLLEGE**

Plot No 63, Site IV, Sahibabad, Ghaziabad, Uttar Pradesh 201010

Affiliated to Dr. A.P.J. Abdul Kalam Technical University, Lucknow



**CERTIFICATE**

This is Certified that **Vishal Verma, Ujjawal Chandela, Nishant Kumar Patel and Kajal Pathak** have carried out the project work presented in this report entitled **“Smart Glove For Health Monitoring”** for the award of **Bachelor of Technology** from Inderprastha Engineering College, Ghaziabad, under my supervision. The report embodies result of original work and studies carried out by Student himself/herself and the contents of the report do not form the basis for the award of any other degree to the candidate or to anybody else.

**Dr. Swati Vaid**

(Asst Professor)

Date: 30/04/2020

**Dr. V.K. Gupta**

(Head of Department)

Date: 30/04/2020

***DECLARATION***

*I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text*.

*Signature: Signature:*

*Name: Vishal Verma Name: Ujjawal Chandela*

*Roll No.:1603031196 Roll No.:1603031184*

*Date:30/04/2020 Date:30/04/2020*

*Signature: Signature:*

*Name: Nishant Kumar Patel Name: Kajal Pathak*

*Roll No.:1703031902 Roll No.:1603031077*

*Date:30/04/2020 Date:30/04/2020*

***ACKNOWLEDGEMENT***

*We take this opportunity to thank our teachers and friends who helped us throughout the project.*

*We would like to thank my guide for the project (****Dr.Swati Vaid, Asst. Professor, Electronics & Communication Engineering****) for her/his valuable advice and time during development of project.*

*We would also like to thank all faculty and staff members of Electronics & Communication Engineering for their constant support during the development of the project.*

*Signature: Signature:*

*Name: Vishal Verma Name: Ujjawal Chandela*

*Roll No.:1603031196 Roll No.:1603031184*

*Date:30/04/2020 Date:30/04/2020*

*Signature: Signature:*

*Name: Nishant Kumar Patel Name: Kajal Pathak*

*Roll No.:1703031902 Roll No.:1603031077*

*Date:30/04/2020 Date:30/04/2020*

***ABSTRACT***

*This project provides a solution to their problem. In the project, we have designed a smart*

*glove for health monitoring. This glove is basically a simple glove which can be worn by anyone, out there looking for medical assistance at their home. The material of the glove is elastic and durable so that it can worn by people of different physique and it could be used again and again. When a person wears this glove and puts his fingers on the sensor, the sensors get soon activated to analyze his vitals. The sensors analyze the pulse rate and body temperature of the person within the standard range.*

*These results will be displayed on the module, attached on the thumb area. Then these results are further transmitted on the server for remote monitoring. These vitals will be monitored by the doctor to prescribe the medicine. There are a lot of people who are below the poverty line and cannot afford expensive treatments. So we also tried to make this glove available to everyone This smart glove is very cheap as compared to other medical facilities. It will act as a boon for the elderly patients and reduce their death rate.*

***TABLE OF CONTENTS***

**CERTIFICATE ii**

**DECLARATION iii**

**ACKNOWLEDGEMENT iv**

**ABSTRACT v**

**TABLE OF CONTENTS vi**

**LIST OF FIGURES vii**

**LIST OF TABLES ix**

1. **INTRODUCTION** ..…1

|  |  |
| --- | --- |
| * 1. Problem Definition   2. Introduction about your project.   3. Project Overview/Specifications | .....1  …..1  …..1 |
| **1.4** Problem Statement | .....3 |
| **1.5** Outcomes & Main Contribution of Thesis | ….3 |
| **1.6** Applications and Advantages | ….3 |
| **1.7** Organization of the thesis | ….4 |

# LITERATURE REVIEW .…. 5

* 1. [Previous work …..5](#_TOC_250001)
  2. [Present work …..6](#_TOC_250000)
  3. Feasibility Study …..6

1. HARDWARE & SOFTWARE REQUIREMENT ANALYSIS 8
   1. Hardware Requirements …..8
   2. Software Requirements ...17
2. **SYSTEM ANALYSIS & DESIGN …..**19
   1. Interfaced Temperature Sensor with LilyPad **…..**19

## Interfaced Pulse Sensor with LilyPad **…..**20

## Interfaced seven segment Display with LilyPad **…..**21

## Interfaced Arduino with NodeMCU **…..**22

|  |  |  |
| --- | --- | --- |
| 5. | **CODING /CORE MODULE** | …...24 |
|  | * 1. Programming Logic   2. Program | …...24  ……25 |

|  |  |  |
| --- | --- | --- |
| 6. | **RESULTS / OUTPUTS & TESTING** | ……31 |
|  | **6.2**Testing of Pulse Rate or Heart Rate | ……31 |
|  | **6.3**Testing of Body Temperature | ……33 |

1. **CONCLUSIONS** ……36
2. **REFERENCES** ……37

# *LIST OF FIGURES*

***Fig No. TITLE page no.***

*Fig. 3.1 LilyPad Arduino Main Board ....7*

*Fig. 3.2 Circuit Diagram Of Temperature Sensor ….8*

*Fig. 3.3 Pulse Sensor ….8*

*Fig. 3.4 Pin Diagram Of Pulse Sensor ….8*

*Fig. 3.5 Working Of Pulse Sensor ….9*

*Fig. 3.6 7 Segment Display ….11*

*Fig. 3.7 Structure Of 7 Segment Display ….12*

*Fig. 3.8 Pin Diagram Of Nodemcu ….13*

*Fig. 3.9 Screenshot Of Software Arduino Ide ….14*

*Fig. 3.10 Screenshot Of Channel Information On Things Speak ….15*

*Fig. 4.1 Lm35 Interfaced With Lilypad Arduino ….17*

*Fig: 4.2 Pulse Sensor Interfaced With Lilypad Arduino ….18*

*Fig: 4.3 TM1637 Module Interfaced With Lilypad Arduino ….19*

*Fig. 4.4 Screenshot Of API Key On Things Speak ….20*

*Fig. 5.1 Programming Logic …..21*

*Fig. 6.1 Picture Displaying Pulse Rate (Bpm) ….34*

*Fig. 6.2. Heartbeat Of The Person Is Getting Displayed On The Serial Monitor ….34*

*Fig. 6.3 Variation Of Pulse Rate As Observed From Serial Monitor ….35*

*Fig. 6.4 Picture Displaying Body Temperature (Deg. C) ….36*

*Fig. 6.5 Screenshot Of Channel Information On ThingsSpeak ….37*

*Fig. 6.6 Screenshot Of API Key On ThingsSpeak ….37*

*Fig. 6.7 Modified Block Diagram …..38*

*Fig. 6.8 Smart Health Monitoring Glove …..39*

*Fig. 6.9 Real Data Uploaded On Thingspeak …..39*

# *LIST OF TABLES*

***Table No. TITLE Page No.***

*Table 3.1 Specification of The System ….10*

*Table 4.1 Connection between Temperature Sensor and LilyPad ….16*

*Table 4.2 Connection between Pulse Sensor and LilyPad ….17*

*Table 4.3 Connection between Display Module and LilyPad ….18*

*Table 4.4 Connection between LilyPad and NodeMCU ….19*

# CHAPTER 1.

**INTRODUCTION**

## Problem Definition

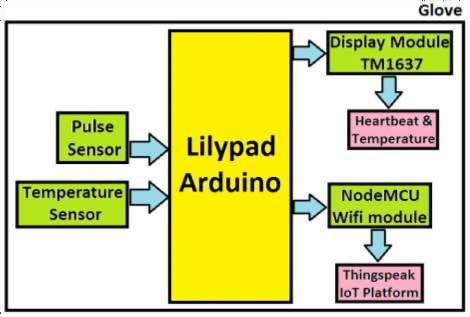
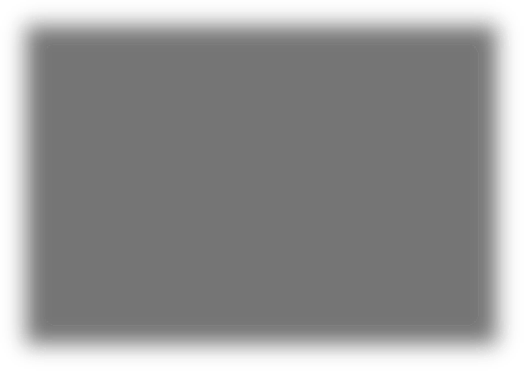
In extreme conditions like pandemics (covid-19) and areas affected with contagious diseases, visiting hospitals for regular health check-ups are generally avoided. In such scenarios, remote monitoring of health are highly advised. Also, if the affected areas are rural and they lack proper health management systems, inexpensive health monitoring equipments are preferred. To resolve all the above issues, a system to monitor the health remotely has been developed in this project.

## Introduction

Remote healthcare monitoring allows people to continue to stay at home rather than visit expensive healthcare facilities such as hospitals or nursing homes. It thus provides an efficient and cost-effective alternative to on-site clinical monitoring. Such systems equipped with non-invasive and unobtrusive wearable sensors can be viable diagnostic tools to the healthcare personnel for monitoring important physiological signs and activities of the patients in real-time, from a distant facility.[1]

## Project Overview

In this project, a health monitoring system is implemented on a glove. It has a microcontroller (LilyPad Arduino) which is interfaced with Temperature sensor(LM35), pulse sensor, LCD display and Wi-Fi Module for observing the state of patient`s health in terms of body temperature and pulse rate and uploading this data on IOT platform. The advantage of the proposed solution is its simplicity, cost-efficiency, and scalability with home based IOT systems. The sensors collect the data when patient applies his finger on to it.



*Fig 1: Block diagram of system*



***Table 1****: Specification of System*

|  |  |  |
| --- | --- | --- |
| **Module** | **Item** | **Specification** |
| LilyPad Arduino | Operating voltage, digital pins, Flash memory | 5V, 14,16KB |
| Temperature sensor | Temperature range, power, output impedance | -55C to  150C,430V,0.1W  for 1mA load |
| Pulse sensor | LED, gain, Power | Infrared LED,100,3.3V |
| NodeMCU Wi-Fi module | Power ,baud rate range | 3.3V,9600BPS,  Up to 10m |
| LCD | Power, display | 5V,plasma display |

## Objectives

The following are the main objectives of this work:

1. Development of glove attached with electronic equipments for measuring pulse rate and body temperature along with the display module to show the readings
2. Connection of the glove with IOT Platform for uploading the data on internet and real time monitoring of patient health.

## Outcomes of the project

The health monitoring system is successfully implemented on hardware. The display of the system shows the temperature reading between 36.5–37.5 deg C and pulse rate data varies for adults (18 years of age and older), a normal resting heart rate is between 60 and 100 beats per minute (bpm) , for a 30-year-old person the maximum heart rate must be within 95 to 162 bpm . where the pulse rate and temperature reading is null when the finger is not applied to the sensor

.

## Applications and Advantages

* Possible for a doctor to observe a patient’s heart rate per minute and body temperature all the time.
* Easy to assemble and really helpful in desperate situation to check the vitals of injured or sick person.
* It is cheaper and affordable than other medical instrument used for diagnosis.
* It is an easy-to-use medical device, which can be operated by anyone without any prior technical knowledge.
* They can be used for monitoring and real time health data acquisition of military professionals.
* They are battery operated devices so it could be used again and again. It provides better treatment facility to those areas where there is shortage of electricity.
* This device will provide the real time values of the patient’s vitals for the most efficient and effective treatment of the patient.
* If the temperature of body and the heart rate go above the specific range, the system can be programmed to automatically send an alert message to nearest doctor and family members.

## Organization of thesis

* Chapter I describe general background, problem statement, objective of the system, outcomes and contributions, advantage and applications.
* Chapter II deal with literature review, it explain previous works related to this project, present work and feasibility.
* Chapter III Hardware and software, describe all components and software and it’s working.
* Chapter IV system design, explain interfacing of the component
* Chapter V Programming logic, explanation of programming logic and coding for temperature sensor, pulse sensor and NodeMCU.
* Chapter VI Results and discussion, testing of temperature and testing of pulse sensor. it gives the detail about analysis and observation of the project,
* Chapter VII conclusion it develop the clear view about the project
* Chapter VIII Reference

# CHAPTER 2.

**LITERATURE SURVEY**



## Previous Work

Several wearable systems [3] have been reported in the literature to monitor the health and diagnosing the patients with arthritis. Heart diseases etc. Some of the designs with smart glove as wearable health monitoring system are as follows

Kumar K R et.al [4] proposed a sensor loaded glove with a complete unit to assist paralytic patients. The system converted the finger bending of the patient into auditory speech. If the same action was repeated thrice, indicating that the requirement is not yet fulfilled, the system sends a text message to the concerned care taker with the help of GSM module.

LuayFraiwanet.al [5] proposed a glove for monitoring mental health. The primary function of this glove was to continuously monitor changes in skin temperature, galvanic skin resistance, and heart rate variability (inter-beat-intervals and cardiac vagal tone index).The wearable glove was integrated with a skin temperature sensor, a galvanic skin resistance sensor, and a pulse sensor, and it recorded then transmit the raw signals from the sensors through Wi-Fi to the cloud database.

G. MahammadGouseet.al [6] proposed a wearable system for hand gesture recognition. Ribin Jones S.B et.al [7] designed a health Monitoring wearable glove which can be worn and used to display the heart rate of any individual on the display mounted on the glove.

Dhwani Patel et.al [8] designed a sensor that can analyze the movement while moving the hand and simultaneously detect the direction of hand movement. The main principle involved in the designing of the sensor was whenever the sensor undergoes a deflection while moving, electrical output was produced which was be converted to a certain voltage. An accelerometer was also employed with the sensor to measure the speed and direction of the hand movement while motion. The main aim of this project was to detect movement disorders of aged individuals at an early stage.

Ms. Ramya Devi et.al [9] proposed an IoT (Internet of Things) based healthcare system to integrate various technologies of wearable devices, sensors and wireless sensor networks to provide intensive service to improve the quality of services in the elderly healthcare system. The aim of the project was to continually monitor the heart beats and temperature of the

user. Priyanka Kakria et.al [10] proposed a real-time heart monitoring system. The system was conceived to provide a doctor-patient interface for two-way communication. The main goal of this study is to enable remote cardiac patients to access the newest health services, which is probably not possible due to the low doctor-to patient ratio.

In this project, the concepts discussed in the above mentioned research works are implemented and a smart glove is designed for measuring the vital signs like body temperature and heart rate. The design is discussed in detail in next section.

## Present Work

The present health monitoring system is implemented on a glove. It has a microcontroller (LilyPad Arduino) which is interfaced with Temperature sensor(LM35), pulse sensor, LCD display and Wi-Fi Module for observing the state of patient`s health in terms of body temperature and pulse rate and uploading this data on IOT platform. The advantage of the proposed solution is its simplicity, cost-efficiency, and scalability with home based IOT systems.

## Feasibility

Watches, bracelets, and necklaces are no longer just accessories, but rather gadgets that can serve as lifelines during emergencies.

As the market expands, the possibilities only grow for how technology can play a vital role in connecting users to public safety when assistance is needed most.

The growing elderly population, accompanied by the increasing prevalence of chronic diseases associated with ageing, will have profound implications for the health care system for decades to come. Therefore, we are proposing a system which enables continuous monitoring for elderly people health in real-time to prevent chronic diseases, thus preventing hospitalization that burden the healthcare systems and costs.

.

# CHAPTER 3

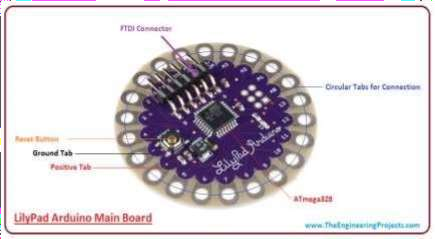
**HARDWARE AND SOFTWARE REQUIREMENTS**

Hardware

1. LilyPad Arduino**:**

The board depends on the ATmega168V (the low-power adaptation of the

ATmega168), which is the main processing unit of the systems. The LilyPad Arduino is stitched with the glove. It is a microcontroller board intended for wearables and e-materials. It will be sewed to material and similarly mounted power supplies, sensors and actuators with conducting thread.



*Fig.3.1 LilyPad Arduino main board*

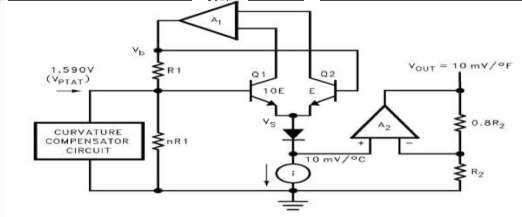
## Temperature Sensor(LM35**):**

LM35 temperature sensor is employed to get the temperature of body. it is calibrated linearly in Celsius. It doesn’t require external calibration. The LM35 IC has a 3 pin-2 for power supply and one for analog output. It is a low voltage IC, which uses + 5VDC power. The output pin provides an analog voltage output that is in direct proportion to the temperature of Celsius (centigrade). Pin 2 produces 1 mW at 0.1 C (10 mV per degree). To get the degree value in Celsius, the voltage output must be taken ,

Working Of Temperature Sensor

There are two transistors in the center of the drawing. One has ten times the emitter area of the other. This means it has one tenth of the current density, since the same current is going through both transistors. [10] This resistor causes a voltage across R1, which is proportional to the absolute temperature and is linear in range. The "almost" part is taken into account by a special circuit that straightens the slightly curved graph of voltage and temperature. The amplifier at the top ensures that the voltage across the left transistor (Q1) is proportional to the output of the two transistors at absolute temperature (PAT).

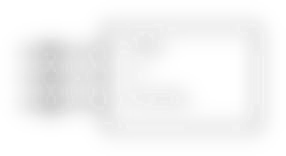
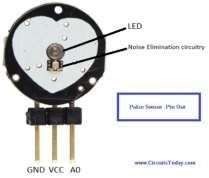
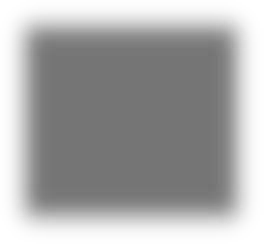
Amplifier on the right converts the absolute temperature (measured in Kelvin) in either Fahrenheit or Celsius depends on the component (LM34 or LM35). A small circle with an "i" is a constant current source circuit. Both resistors are calibrated to produce the most accurate temperature sensors in the factory. There are many transistors in integrated circuits - two in the middle, some in each amplifier, some in a constant current source and some in a curved compensation circuit. All of that is fit into the tiny package with three leads.



*Fig. 3.2 Circuit Diagram of Temperature sensor*

## Pulse Sensor(SEN-11574):

The primary heart rate sensor consists of a light-emitting diode and a detector or photodiode. Pulses of the heart cause a difference in blood flow to the tissues. When body tissue glows with a light source, LED, it either reflects (a finger tissue) or transmits the light(earlobe).



*Fig 3.3pulse Sensor*

*Fig.3.4pin diagram of pulse Sensor*

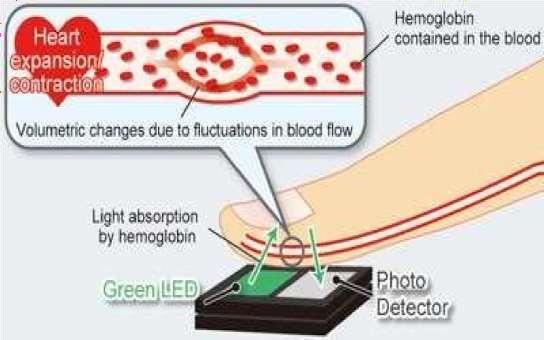
8

Some light is absorbed by the blood and the transmitted or reflected light is received by a light detector. The amount of light absorbed depends on the amount of blood in that tissue. The detector output is in the form of an electrical signal and is proportional to the heart rate. Here, the sensor reads the amount of amount of blood flowing through finger.

Working Of Pulse Sensor

The pulse / heartbeat sensor is very easy to operate. There are two elements of the sensor, one side LED is paired with an ambient light sensor and the other side we have some circuit. This Responsible for circuit expansion and noise cancellation work. The LED on the front of the sensor is placed on a vein in our human body. It may be your fingertip or you may have ear tips, but keep it directly above the vein. The LED now emits light falling directly on the vein. The blood flow in the veins only happens when the heart is pumping, so we can also monitor the heart rate if we monitor the blood flow.

If blood flow is detected, the ambient light sensor receives more light when reflected from the blood, and this slight change in the received light is periodically analyzed to determine our heart rate..



*Fig. 3.5 Working of pulse sensor*

*Table 3.1****: Specification of System***

|  |  |  |
| --- | --- | --- |
| **Module** | **Item** | **Specification** |
| LilyPad Arduino | Operating voltage, digital pins, Flash memory | 5V,  14,16KB |
| Temperature sensor | Temperature range, power r, output impedance | -55C to 150C,430V,  0.1W for 1mA load |
| Pulse sensor | LED, gain, Power | Infrared LED,100,3.3V |
| Wi-Fi module | Power, baud rate  ,range | 3.3V,9600BPS,  Up to 10m |
| LCD | Power, display | 5V,plasma display |

## TM1637 Display Module

It is used for displaying numbers. The module consists of four 7- segment displays working together. The module working is based on „TM1637‟ IC present internally and hence the name „TM1637 display‟.

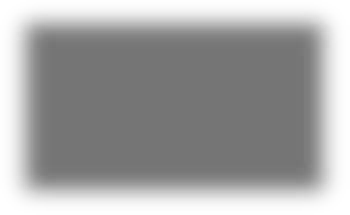
*TM1637 Display Features and Specifications*

* Two wire interface
* Eight adjustable luminance levels
* Grove compatible interface (3.3V/5V)
* Four alpha-numeric digits
* Operating voltage: 3.3V – 5.5V
* Operating current consumption: 80mA
* Operating temperature: -10ºC to +80ºC

The **TM1637 IC** on the module receives the serial data sent by the controller. The chip drives the 4 display segments according to the code.[11] The segments light up to display



the desired character.







***Fig. 3.6 7 Segment Display***



Types of 7 segment display

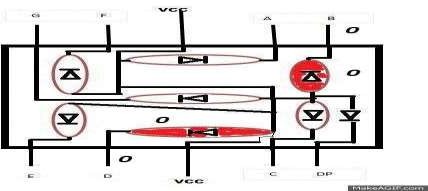
Seven segment display is composed of lad’s that are arranged in 2 configuration.

First configuration has all the lad’s anode’s connected together and this configuration 7 segment display is known as common anode 7 segment display .Other Configuration Unlike the previous one, all the cathodes of lad’s are connected to each other and this configuration is called the common cathode display. The seven segment at these configuration bases are divided into two types, the common anode (CA) and the common cathode (CC).Both configuration has some pros and cons.

Common Anode 7-Segment Display

For common anode, apply +5 V to Vcc pin in series to a 510 ohm-1k ohm resistor. This resistor is very important always use it ,otherwise your seven segment display will damaged by over current. Note both the Vcc pins are short so apply +5 volts on only one pin and leave other empty.

* + In common Anode the Cathode (-) side of lad’s are connected to a, b, c, d, e, f, g pins of seven segment display.
  + common anode seven segment display’s led becomes lit when we ground any a, b, c, d, e, f, g pin.
  + Common Anode seven segment display’s colour is usually grey.



*Fig. 3.7 structure of 7 segment display*

Common Cathode 7-Segment Display

* + For common cathode make GND pin ground. Only make one GND pin ground leave the other empty because both the GND pins are shorted. 1
  + Apply +5 volts to dp(decimal/display point) pin in series to a 510 ohm-1k ohm resistor to limit the current. If you want it to illuminate for ever apply +5v. If you want to control dp(decimal/display point) led than connect it to some control system, microcontroller etc.
  + Now if your small circle led let’s it means that your seven segment is properly working and now you can use it.
  + In Common cathode lad’s anode(+) sides are connected to a,b,c,d,e,f,g pins of seven segment display.
  + In common cathode seven segment display’s led becomes lit when we apply some

+positive voltage on any a,b,c,d,e,f,g pin.

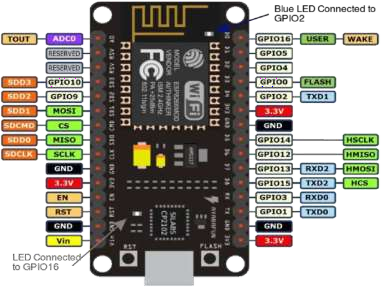
* + Common Cathode seven segment display’s colour is usually black.

Below you can see the 7 segment display pinout for both common anode and cathode 7 segment display. Note the only difference is in power pins. Each 7 segment display has two power pins. You need to power only one rail at a time leave the other vacant.

## NODE MCU

It is an open-source firmware and development kit that helps you to prototype or build IoT products. The ESP8266 is mounted on NODEMCU.[12]This module is used to achieve communication between Arduino and IOT Platform

The ESP8266 Wi-Fi module could be a self-contained SOC with incorporated TCP/IP protocol stack which will offer any controller access to Wi-Fi network. It uses 802.11 b/g/n protocols. Standby power consumption is a smaller amount than 0.1mW.



### *Fig. 3.8 pin diagram of NodeMCU*

**3.2 Software**

Arduino IDE

Arduino Integrated Development Environment (IDE) is a cross-platform application (for Windows, MacOS, Linux) written in functions from C and C ++ [2]. It is used to write and upload programs to Arduino compatible boards. [E]By default, avrdude is used as the uploading tool to flash the user code onto official Arduino boards [8]..



*Fig. 3.9 screenshot software Arduino ide*

User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable [c](https://en.wikipedia.org/wiki/Cyclic_executive)yclic executiv[e](https://en.wikipedia.org/wiki/Cyclic_executive) [p](https://en.wikipedia.org/wiki/Cyclic_executive)rogram with the [G](https://en.wikipedia.org/wiki/GNU_toolchain)NU tool chai[n,](https://en.wikipedia.org/wiki/GNU_toolchain) also included with the IDE distribution.[[6] T](https://en.wikipedia.org/wiki/Arduino_IDE#cite_note-6)he Arduino IDE employs the program *avrdude* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware[7[] B](https://en.wikipedia.org/wiki/Arduino_IDE#cite_note-7)y default, avrdude is used as the uploading tool to flash the user code onto official Arduino boards [[8].](https://en.wikipedia.org/wiki/Arduino_IDE#cite_note-8)

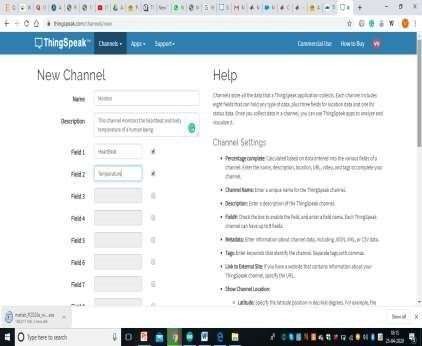
## Thing Speak

ThingSpeak™ is an IoT analytics platform service that allows you to aggregate, and an

ThingsSpeak provides with an instant visualization of the data your devices have posted. With the ability to execute matlab code on ThingSpeak, you can do online analysis and processing when data arrives. ThingsSpeak is often used for concept proof that prototype IoT systems and analytics require. ThingsSpeak allows you to consolidate, visualize, and analyze live data streams in the cloud..

Some of the key capabilities of ThingSpeak include the ability to:

* Easily configure devices to send data to ThingSpeak using popular IoT protocols.
* Visualize your sensor data in real-time.
* Aggregate data on-demand from third-party sources.
* Use the power of MATLAB to make sense of your IoT data.
* Run your IoT analytics automatically based on schedules or events.



*Fig. 3.10 Screenshot of Channel information on ThingSpeak*

# CHAPTER 4.

**SYSTEM DESIGN**

The entire glove is fabricated with the Arduino and sensors and the step for designing the glove are as follows:-

* Interfacing of Arduino with LM35
* Interfacing of Arduino with SN11527 Interfacing of Arduino with TM1637
* Interfacing of Arduino with NodeMCU
  1. Interfacing of Arduino with LM35

The following pin connections are made.

*Table 4.1 Connections between LM35 and LilyPad*

|  |  |
| --- | --- |
| **LM35** | **LilyPad Arduino** |
| Vcc | Vcc |
| Ground | Ground |
| Analog Out | A1(Analog Pin) |



*Fig: 4.1 Lm35 interfaced with LilyPad Arduino*

* 1. Interfacing of Arduino with SN11527

The following pin connections are made.

*Table 4.2 Connections between Pulse Sensor and LilyPad*

|  |  |
| --- | --- |
| Pulse sensor | LilyPad Arduino |
| 5V supply pin | 5V supply pin |
| Ground | Ground |
| Signal Pin | Analog Pin(A0) |



*Fig: 4.2 Pulse Sensor interfaced with LilyPad Arduino*

* 1. Interfacing of Arduino with TM1637

The following pin connections are made.

*Table 4.3 Connections between TM1637 and LilyPad*

|  |  |
| --- | --- |
| TM1637‟ module | LilyPad Arduino |
| Vcc | Vcc |
| Ground | Ground |
| CLK | Pin 10 |



*Fig: 4.3 TM1637 module interfaced with LilyPad Arduino*

* 1. Interfacing of Arduino with NodeMCU

The following pin connections are made.

*Table 4.4 Connections between LilyPad and NodeMCU*

|  |  |
| --- | --- |
| LilyPad Arduino | NodeMCU |
| Vcc | Vcc |
| Ground | Ground |
| D2 | Pin 9(digital pin) |
| D3 | Pin 10(digital Pin) |

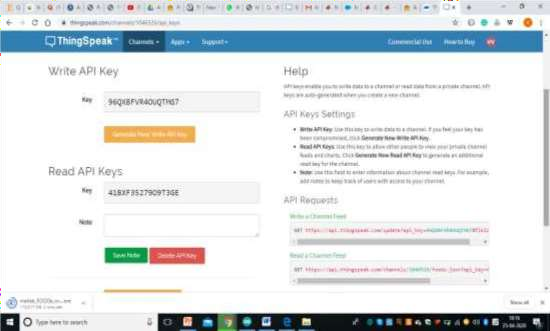
### ThingSpeak

Once the data was collected on the LilyPad, it is further transmitted to the ThingSpeak Server using NodeMCU for remote access.

To access the information on the server, we need to create a channel for transmitting and receiving the information as shown in figure: 4.4

**Write API Key**: Use this key to write data to a channel.

**Read API Keys**: Use this key to allow other people to view your private channel feeds and charts.



*Fig. 4.4 Screenshot of API Key on Things Speak*

And thus the entire data of our sensors can be monitored remotely by the doctor. He can prescribe the medicines to the patient accordingly

# CHAPTER 5.

**CODING**

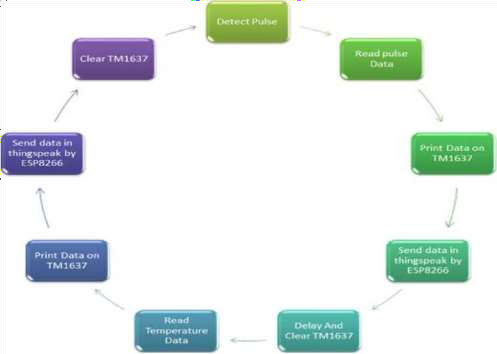
### Programming Logic

Pulse sensor, detects the pulse based on the threshold value set for it, using sawStartOfBeat() function.

Once the pulse is detected, data is read in beats/minute using getBeatsPerMinute() function. This data is printed on TM1637 display module using display.print() function and immediately sent on Thing Speak via ESP8266.Once the data is displayed, display is cleared soon after, This cycle is repeated for after each 2 seconds or whenever the pulse is detected for a set time. After set time Data from LM35 Temperature sensor is taken too for a set time

A stabilized Data Reading from temperature sensor is detected and fed to TM1637 display module using display.print() function and immediately sent on ThingSpeak via ESP8266

Display is cleared soon after and process repeated again for set time



*Fig.5.1 Programming Logic*

### 5.2 Program

#include <LiquidCrystal.h> LiquidCrystal lcd(12, 11, 5, 4, 3, 2); #include <SoftwareSerial.h>

float pulse = 0; float temp = 0;

SoftwareSerial ser(9,10);

String apiKey = "SNX3F6OPUW50DUL9"; String ssid = "sanjeev";

String pass = "sanjeev3512";

// Variables

int pulsePin = A0; // Pulse Sensor purple wire connected to analog pin 0 int blinkPin = 7 ; // pin to blink led at each beat

int fadePin = 13; // pin to do fancy classy fading blink at each beat int fadeRate = 0; // used to fade LED on with PWM on fadePin

// Volatile Variables, used in the interrupt service routine!

volatile int BPM; // int that holds raw Analog in 0. updated every 2mS volatile int Signal; // holds the incoming raw data

volatile int IBI = 600; // int that holds the time interval between beats! Must be seeded!

volatile boolean Pulse = false; // "True" when User's live heartbeat is detected. "False" when nota "live beat".

volatile boolean QS = false; // becomes true when Arduoino finds a beat.

// Regards Serial OutPut -- Set This Up to your needs

static boolean serialVisual = true; // Set to 'false' by Default. Re-set to 'true' to see Arduino Serial Monitor ASCII Visual Pulse

volatile int rate[10]; // array to hold last ten IBI values

volatile unsigned long sampleCounter = 0; // used to determine pulse timing volatile unsigned long lastBeatTime = 0; // used to find IBI

volatile int P = 512; // used to find peak in pulse wave, seeded volatile int T = 512; // used to find trough in pulse wave, seeded

volatile int thresh = 525; // used to find instant moment of heart beat, seeded volatile int amp = 100; // used to hold amplitude of pulse waveform, seeded

volatile boolean firstBeat = true; // used to seed rate array so we startup with reasonable BPM volatile boolean secondBeat = false; // used to seed rate array so we startup with reasonable BPM

void setup()

{

lcd.begin(16, 2);

pinMode(blinkPin,OUTPUT); // pin that will blink to your heartbeat! pinMode(fadePin,OUTPUT); // pin that will fade to your heartbeat! Serial.begin(115200); // we agree to talk fast!

interruptSetup(); // sets up to read Pulse Sensor signal every 2mS

// IF YOU ARE POWERING The Pulse Sensor AT VOLTAGE LESS THAN THE BOARD VOLTAGE,

// UN-COMMENT THE NEXT LINE AND APPLY THAT VOLTAGE TO THE A-REF PIN

// analogReference(EXTERNAL); lcd.clear();

lcd.setCursor(0,0); lcd.print(" Patient Health"); lcd.setCursor(0,1); lcd.print(" Monitoring "); delay(4000);

lcd.clear(); lcd.setCursor(0,0); lcd.print("Initializing ");

delay(5000);

lcd.clear(); lcd.setCursor(0,0); lcd.print("Getting Data. ");

ser.begin(9600); ser.println("AT"); delay(1000); ser.println("AT+GMR"); delay(1000);

ser.println("AT+CWMODE=3"); delay(1000); ser.println("AT+RST"); delay(5000); ser.println("AT+CIPMUX=1"); delay(1000);

String cmd="AT+CWJAP=\"Alexahome\",\"98765432\""; ser.println(cmd);

delay(1000); ser.println("AT+CIFSR"); delay(1000);

}

// Where the Magic Happens void loop()

{

serialOutput();

if (QS == true) // A Heartbeat Was Found

{

// BPM and IBI have been Determined

// Quantified Self "QS" true when Arduino finds a heartbeat

fadeRate = 255; // Makes the LED Fade Effect Happen, Set 'fadeRate' Variable to 255 to fade LED with pulse

serialOutputWhenBeatHappens(); // A Beat Happened, Output that to serial. QS = false; // reset the Quantified Self flag for next time

}

ledFadeToBeat(); // Makes the LED Fade Effect Happen delay(20); // take a break

read\_temp(); esp\_8266();

}

void ledFadeToBeat()

{

fadeRate -= 15; // set LED fade value

fadeRate = constrain(fadeRate,0,255); // keep LED fade value from going into negative numbers! analogWrite(fadePin,fadeRate); // fade LED

}

void interruptSetup()

{

// Initializes Timer2 to throw an interrupt every 2mS.

TCCR2A = 0x02; // DISABLE PWM ON DIGITAL PINS 3 AND 11, AND GO INTO CTC MODE TCCR2B = 0x06; // DON'T FORCE COMPARE, 256 PRESCALER

OCR2A = 0X7C; // SET THE TOP OF THE COUNT TO 124 FOR 500Hz SAMPLE RATE TIMSK2 = 0x02; // ENABLE INTERRUPT ON MATCH BETWEEN TIMER2 AND OCR2A sei(); // MAKE SURE GLOBAL INTERRUPTS ARE ENABLED

}

void serialOutput()

{ // Decide How To Output Serial. if (serialVisual == true)

{

arduinoSerialMonitorVisual('-', Signal); // goes to function that makes Serial Monitor Visualizer

}

else

{

sendDataToSerial('S', Signal); // goes to sendDataToSerial function

}

}

void serialOutputWhenBeatHappens()

{

if (serialVisual == true) // Code to Make the Serial Monitor Visualizer Work

{

Serial.print("\*\*\* Heart-Beat Happened \*\*\* "); //ASCII Art Madness Serial.print("BPM: ");

Serial.println(BPM);

}

else

{

sendDataToSerial('B',BPM); // send heart rate with a 'B' prefix sendDataToSerial('Q',IBI); // send time between beats with a 'Q' prefix

}

}

void arduinoSerialMonitorVisual(char symbol, int data )

{

const int sensorMin = 0; // sensor minimum, discovered through experiment const int sensorMax = 1024; // sensor maximum, discovered through experiment int sensorReading = data; // map the sensor range to a range of 12 options:

int range = map(sensorReading, sensorMin, sensorMax, 0, 11);

// do something different depending on the

// range value:

switch (range)

{

case 0:

Serial.println(""); /////ASCII Art Madness break;

case 1:

Serial.println("---"); break;

case 2:

Serial.println(" ");

break; case 3:

Serial.println(" ");

break; case 4:

Serial.println(" ");

break; case 5:

Serial.println(" |-");

break; case 6:

Serial.println("--------------| ");

break; case 7:

Serial.println("--------------| ");

break; case 8:

Serial.println("--------------| ");

break;

case 9:

Serial.println("--------------| ");

break; case 10:

Serial.println("--------------| ");

break; case 11:

Serial.println("--------------| ");

break;

}

}

void sendDataToSerial(char symbol, int data )

{

Serial.print(symbol); Serial.println(data);

}

ISR(TIMER2\_COMPA\_vect) //triggered when Timer2 counts to 124

{

cli(); // disable interrupts while we do this

Signal = analogRead(pulsePin); // read the Pulse Sensor sampleCounter += 2; // keep track of the time in mS with this variable

int N = sampleCounter - lastBeatTime; // monitor the time since the last beat to avoid noise

// find the peak and trough of the pulse wave

if(Signal < thresh && N > (IBI/5)\*3) // avoid dichrotic noise by waiting 3/5 of last IBI

{

if (Signal < T) // T is the trough

{

T = Signal; // keep track of lowest point in pulse wave

}

}

if(Signal > thresh && Signal > P)

{ // thresh condition helps avoid noise P = Signal; // P is the peak

} // keep track of highest point in pulse wave

// NOW IT'S TIME TO LOOK FOR THE HEART BEAT

// signal surges up in value every time there is a pulse if (N > 250)

{ // avoid high frequency noise

if ( (Signal > thresh) && (Pulse == false) && (N > (IBI/5)\*3) )

{

Pulse = true; // set the Pulse flag when we think there is a pulse digitalWrite(blinkPin,HIGH); // turn on pin 13 LED

IBI = sampleCounter - lastBeatTime; // measure time between beats in mS lastBeatTime = sampleCounter; // keep track of time for next pulse

if(secondBeat)

{ // if this is the second beat, if secondBeat == TRUE secondBeat = false; // clear secondBeat flag

for(int i=0; i<=9; i++) // seed the running total to get a realisitic BPM at startup

{

rate[i] = IBI;

}

}

if(firstBeat) // if it's the first time we found a beat, if firstBeat == TRUE

{

firstBeat = false; // clear firstBeat flag secondBeat = true; // set the second beat flag

sei(); // enable interrupts again

return; // IBI value is unreliable so discard it

}

// keep a running total of the last 10 IBI values

word runningTotal = 0; // clear the runningTotal variable for(int i=0; i<=8; i++)

{ // shift data in the rate array

rate[i] = rate[i+1]; // and drop the oldest IBI value runningTotal += rate[i]; // add up the 9 oldest IBI values

}

rate[9] = IBI; // add the latest IBI to the rate array runningTotal += rate[9]; // add the latest IBI to runningTotal runningTotal /= 10; // average the last 10 IBI values

BPM = 60000/runningTotal; // how many beats can fit into a minute? that's BPM! QS = true; // set Quantified Self flag

// QS FLAG IS NOT CLEARED INSIDE THIS ISR

pulse = BPM;

if (Signal < thresh && Pulse == true)

{ // when the values are going down, the beat is over digitalWrite(blinkPin,LOW); // turn off pin 13 LED

Pulse = false; // reset the Pulse flag so we can do it again amp = P - T; // get amplitude of the pulse wave

thresh = amp/2 + T; // set thresh at 50% of the amplitude P = thresh; // reset these for next time

T = thresh;

}

if (N > 2500)

{ // if 2.5 seconds go by without a beat thresh = 512; // set thresh default

P = 512; // set P default

T = 512; // set T default

lastBeatTime = sampleCounter; // bring the lastBeatTime up to date firstBeat = true; // set these to avoid noise

secondBeat = false; // when we get the heartbeat back

}

sei(); // enable interrupts when youre done!

}// end isr

void esp\_8266()

{

// TCP connection AT+CIPSTART=4,"TCP","184.106.153.149",80 String cmd = "AT+CIPSTART=4,\"TCP\",\"";

cmd += "184.106.153.149"; // api.thingspeak.com cmd += "\",80";

ser.println(cmd); Serial.println(cmd); if(ser.find("Error"))

{

Serial.println("AT+CIPSTART error"); return;

}

String getStr = "GET /update?api\_key="; getStr += apiKey;

getStr +="&field1="; getStr +=String(temp); getStr +="&field2="; getStr +=String(pulse); getStr += "\r\n\r\n";

// send data length

cmd = "AT+CIPSEND=4,";

cmd += String(getStr.length()); ser.println(cmd); Serial.println(cmd); delay(1000);

ser.print(getStr);

Serial.println(getStr); //thingspeak needs 15 sec delay between updates delay(3000);

}

void read\_temp()

{

int temp\_val = analogRead(A1); float mv = (temp\_val/1024.0)\*5000; float cel = mv/10;

temp = (cel\*9)/5 + 32; Serial.print("Temperature:"); Serial.println(temp); lcd.clear(); lcd.setCursor(0,0); lcd.print("BPM :"); lcd.setCursor(7,0); lcd.print(BPM); lcd.setCursor(0,1); lcd.print("Temp.:"); lcd.setCursor(7,1); lcd.print(temp); lcd.setCursor(13,1); lcd.print("F");

}

# CHAPTER. 6

**RESULTS & DISCUSSIONS**:

After designing the glove, the working of the glove was tested and the results were analysed.

### Testing of Pulse Rate or Heart Rate

For adults (18 years of age and older), a normal resting heart rate is between 60 and 100 beats per minute (bpm) but depending on the person?s physical condition and age this parameter changes.

For example for a 30-year-old person, the maximum heart rate must be within 95 to 162 bpm.

Calculations for heart rate for a 30 year old healthy personis as follows:[13] The range of heart rate is calculated by subtracting the age from 220.i.e.: 220 – 30 = 190.

* 50 percent: 190 x 0.50 = 95 bpm
* 85 percent: 190 x 0.85 = 162 bpm

For a 60-year-old person, the target zone would be between 80 and 136 bpm.

We have tested this glove with a 21 years old person, the calculated heartbeat range must be within 99.5 to 169.5 bpm, because =220-21=199.

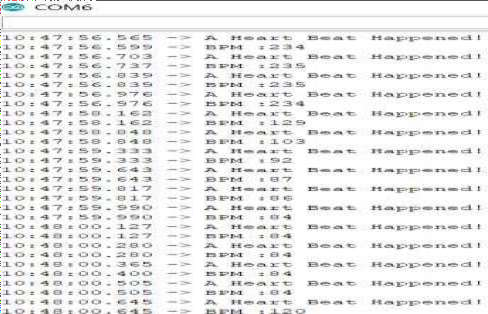
Minimum Heart rate=0.5\*199=99.5 Maximum Heart rate=0.85\*199=169.5

After testing our glove with a normal person of 21 years of age, the heartbeat per minute displayed on the display module was 88bpm as shown in the Fig.6.1



*Fig. 6.1 Picture Displaying Pulse rate (bpm) of a 21 years old healthy person on glove*

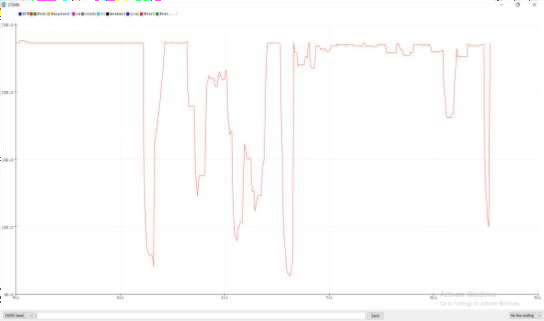
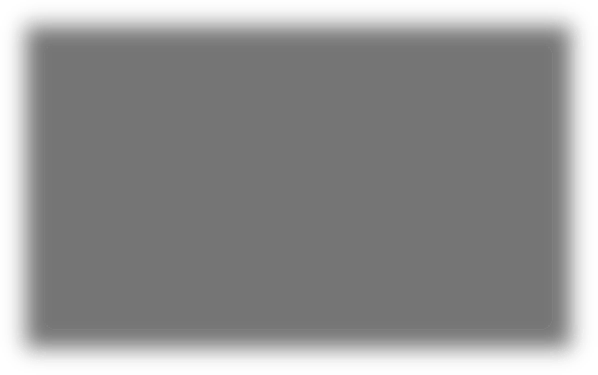
While the pulse sensor is measuring heartbeat, the readings can be displayed on the Arduino Serial Monitor as shown in Fig 2.



*Fig. 6.2. Heartbeat of the person is getting displayed on the Serial Monitor.*

However, the complete variation of pulse rate with time is observed using serial monitor and

it is shown in Fig.3



*Fig. 6.3 Variation of Pulse rate as observed from serial monitor*



### Testing of Body Temperature

To measure the temperature of the human body,LM35 sensor has been used. It senses the temperature of the body and converts it into the voltage in millivolts(Vout), which is further processed/evaluated to give the temperature in degree Celsius.

It can measure temperature from -55 degree Celsius to +150 degree Celsius. The voltage output of the LM35 increases 10mV per degree Celsius rise in temperature.

tempC= Vout/0.01; tempc=(tempc/10.0); Ttemp=tempc\*100; The final reading of temperature sensor displayed on the 7-segment display was 36.4ᶛC.



*Fig. 6.4 Picture Displaying body temperature (deg. C) of a 21 years old healthy person on glove.*

Once the data was collected on the Arduino Platform ,it is further transmitted to the ThingSpeak Server using NodeMCU for remote access.

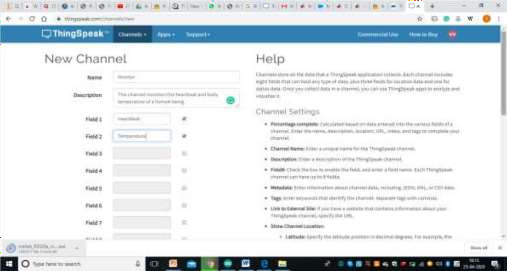
To access the information on the server, we need to create a channel for transmitting and receiving the information as shown in figure:

API keys enable you to write data to a channel or read data from a private channel. There are two types of API keys in the channel:

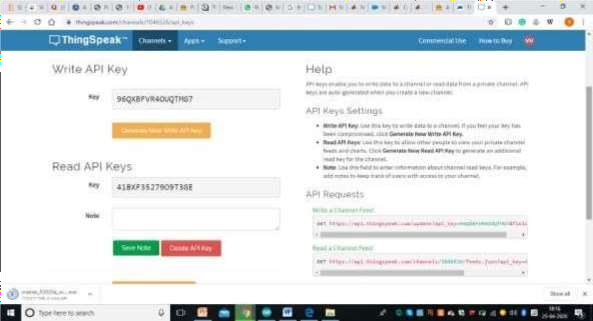
**Write API Key**: Use this key to write data to a channel.

**Read API Keys**: Use this key to allow other people to view your private channel feeds and charts.

And thus the entire data of our sensors can be monitored remotely by the doctor. He can prescribe the medicines to the patient accordingly



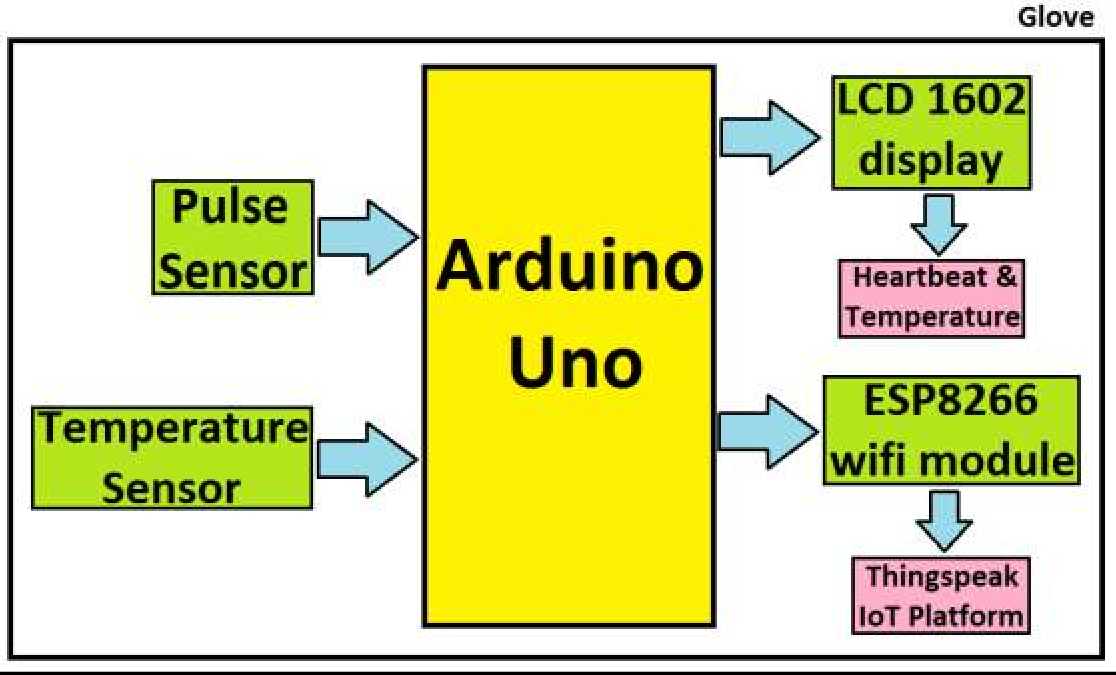
*Fig. 6.5Screenshot of Channel information on Things Speak*



*Fig. 6.6 Screenshot of API Key on Things Speak*

# PROGRESS DURING PANDEMIC

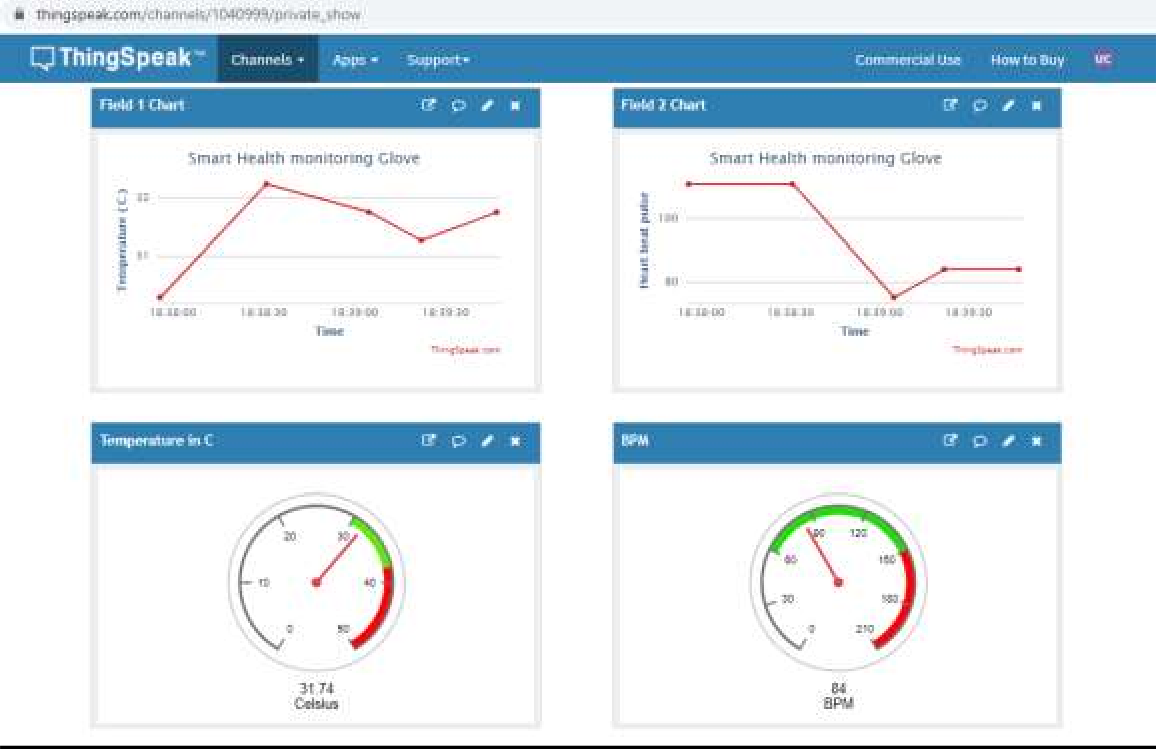
* LilyPad Arduino replaced with Arduino Uno to facilitate more pins for future enhancement in the current model for Smart health monitoring glove.
  + *Effortless functions*
  + *Examples of codes*
  + *Ready to Use*
* TM1637 display replaced with LCD 1602 display to distinctively show Heart rate and temperature simultaneously.
  + *16 Columns and 2 Rows for Display*
  + *Alphanumeric LCD display module, meaning can display alphabets and numbers*
  + *works on both 8-bit and 4-bit mode*
* NodeMCU replaced with ESP8266 wifi module for better interfacing and avoid unnecessary clumsiness
  + *Low cost, compact*
  + *Can be used as Station or Access Point or both combined*



*Fig. 6.7 Modified Block Diagram*



*Fig. 6.8 Smart Health Monitoring Glove*



*Fig. 6.9. Real data uploaded on ThingSpeak*

# CHAPTER 7

**CONCLUSION**

The implementation and working of a *Smart Glove for Health Monitoring* is presented in this thesis. The testing of this glove is done by measuring the vitals of a normal healthy person, pulse rate is found to be 88bpm and temperature measured is 36.4ᶛC. It has low cost, being easily accessible and used for personal health monitoring and is expected for early detection and better treatment of various medical conditions as well as disease prevention and better understanding and self management of diseases. This system is very helpful for the remote monitoring of patients staying at home. The system empowers specialists to analyze and distinguish the conditions of versatility and help heart related problem of the patients without expecting them to be present at the emergency clinic.

# CHAPTER 8.

**REFERENCES**

1. Ribin Jones S.B, Dr. N. Kumar, Swaraj S Paul, "Health Monitoring Wearable Glove" in September,2018
2. L. Gatzoulis; I. Iakovidis (Sep, Oct 2007): Wearable and portable ehealth systems. IEEE Eng. Med. Biol.
3. Selvaraj, Suresh kumar, and Suresh Sundara varadhan. "Challenges and opportunities in IoT healthcare systems: a systematic review." *SN Applied Sciences* 2, no. 1 (2020): 139.
4. Kumara K R, Ankita Kadam, Neha Rane, Shraddha Vernekar, Asma Gouda,"Sensor Based Wearable System to Assist Paralytic Patient with Continuous Health Monitoring" in May,2018.
5. Luay Fraiwan, Tasnim Basmaji, Omnia Hassanin,"A Mobile Mental Health Monitoring System: A Smart Glove" in 2018.
6. Dr. G. Mahammad Gouse, Mir Abdul Samim Ansari, Shubiya Sultana, SwathiJha, Sruthi Krishna.U, "Advanced Data Peripheral Glove with Real-time Vital Health Monitoring System" in 2017..
7. Dhwani patel, Sarthak nayak and Atluri Venkatkrishna, “Design of a wearable sensor glove to measure human hand movement” in 2018.
8. Ms. Ramya Devi, Sri Bargavi Yuvarani, “E Health Glove using IoT” in 2019.
9. Priyanka Kakria, N.K.Tripathi and Peerapong Kitipawang, “A Real-Time Health Monitoring System for Remote Cardiac Patients Using Smartphone and Wearable Sensors” in 12 November 2015.
10. Yogita Bobade; Prof. R. M. Walli (2015): A Review of Wearable Health Monitoring Systems. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering.
11. Hermawan Kemis; Ndibanje Bruce; Hoon Jae Lee(Nov. 2013): On Pulse Sensor based uHealthcare Monitoring Application with Arduino. Journal of Next Generation Information Technology(JNI

# Smart glove for Health Monitoring

*Nishant Kumar Patel Vishal Verma*

*Electronic and communication Engineering Electronic and Communication Engineering*

*Inderprastha Engineering college Inderprastha Engineering college*

*Ghaziabad, India Ghaziabad, India*

[*Patel.nishant47@gnail.com*](mailto:Patel.nishant47@gnail.com)[*vverma1a1@gmail.com*](mailto:vverma1a1@gmail.com)

*Ujjawal Chandela Kajal Pathak*

*Electronic and communication Engineering Electronic and Communication Engineering*

*Inderprastha Engineering College Inderprastha Engineering College*

*Ghaziabad, India Ghaziabad, India*

[*uujjawalchandela4@gmail.com*](mailto:uujjawalchandela4@gmail.com)[*kajalpathak391@gmail.com*](mailto:kajalpathak391@gmail.com)

**Abstract - This paper discusses the construction of IOT based health monitoring glove. Body temperature and heart rate can be measured using this glove. It has a functionality of uploading the data on IoT platform i.e. ThingSpeak, which can be further used to observe patient’s condition. It consists of LilyPad Arduino board interfaced with temperature sensor LM35 and reflective type pulse sensorSEN-11574. Wi-Fi module ESP8266 is employed for wireless data communication between LilyPad Arduino and IOT platform.**

***Keywords—Health monitoring system, pulse sensor, temperature sensor, IOT***

* 1. **INTRODUCTION**

In extreme conditions like pandemics (covid-19) and areas affected with contagious diseases, visiting hospitals for regular health check-ups are generally avoided. In such scenarios, remote tracking of health are highly advised. Also, if the affected areas are rural and they lack proper health management systems, inexpensive health monitoring equipments are preferred. For this reason different designs are reported in the literature which give simple and guaranteed caring unit.

Remotely tracking of health gives patients the ease of home and not in costly hospitals, clinics or nursing homes. Providing an capable and cost effective substitute to real- time health monitoring.[6] Framework structures set up with on request and submissive wearable sensors can help for analytic intends to the healthcare crew for tracking significant health signs and exercises of the patients progressively, from a remote facility.[6]

In this manner, it is consistent and evident that wearable sensors play out a fundamental job in such following game plan structures that gained up the thought of numerous researchers, entrepreneurs, and tech giants in recent history.

Several wearable systems [1] have been described in the literature to track the health and diagnosing the patients with arthritis, heart diseases etc. few of the proposed systems with smart glove as wearable health tracking system are as follows. Kumar K R et.al [2] proposed a sensor loaded glove with a complete unit to aid paralytic patients. The system converted the finger bending of the subject into auditory speech. A similar activity was rehashed threefold, showing that the necessity isn't yet satisfied, the system sends an instant message to the concerned guardian with the aid of GSM module. Luay Fraiwan et.al [3] proposed a glove for tracking mental health. The essential capacity of this glove was to persistently screen changes in skin temperature, galvanic skin opposition, and pulse fluctuation The wearable glove was integrated with a skin temperature sensor, a galvanic skin resistance sensor, and a pulse sensor, and it records the fresh raw signals from the sensors then processes it to transmit with help of Wi-Fi onto the cloud. G. Mahammad Gouse et.al [4] proposed a wearable system for hand gesture recognition.

Ribin Jones S.B et.al [5] constructed a health Monitoring wearable glove which can be used to

exhibit the pulse rate of the subject on the display screen mounted over the proposed glove.

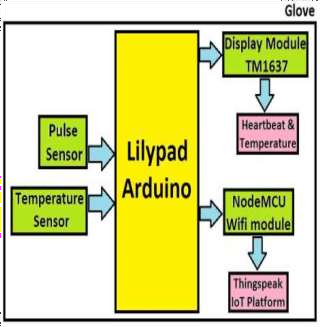
In this paper, the concepts discussed in the above mentioned research works are implemented and a smart glove is constructed for measuring the vital signs like body heat and pulse rate. This device collects the data in real time and upload it on IOT platform for further analysis and diagnosis of the disease of a patient. The system construct is explained in detail in next section.

* 1. **HEALTH MONITORING GLOVE**

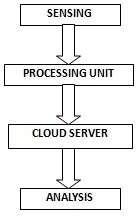
The present health tracking system is implemented on a glove; it has a microcontroller LilyPad Arduino which is interfaced with temperature sensor (lm35), pulse sensor, LCD display and Wi-Fi module for observing the present state of the patient’s health in term of heart rate and body warmth/temperature and uploading this processed data to the IOT/Cloud platform.

* 1. **SYSTEM ARCHITECTURE**

1. ***BLOCK DIAGRAM***



*Fig 1: Block diagram of system*



*Fig 2: Working of system*

The advantage of the expected system its simplicity, cost-effective and scalability with home based IOT system.

Fig.1 shows the expected system. The health monitoring sensors collect health related data for data acquisition. Communication is done by microcontroller for sending data on IOT by WIFI module. Processing is done at server. All data collected and aggregated at server point. If we wish to understand information of patients it is shown on web content i.e. ThingSpeak.

Patients will be wearing the health monitoring gloves, in this device we use two sensors, one temperature sensor and second is pulse sensor. Using these devices, we detect and monitor the patient's vital signs.

Fig 2 shows the work flow of system. The results collected from sensor are analysed i.e. if abnormal behavior has been detected , then emergency plan activated to tell the Doctor about patient’s health .So it reduces critical conditions in Hospital.

1. ***Components Used***
   1. ***LilyPad Arduino :***

The board works on the ATmega168V (the low- power adaptation of the ATmega168), which is the main processing unit of the systems. The LilyPad Arduino is stitched with the glove.

* 1. ***Temperature Sensor:***

LM35 temperature sensor is employed to get the temperature of body. It is calibrated linearly in Celsius. It doesn’t require external calibration. The LM35 IC has 3 pins - 2 for the powering up system and remaining for the simple yield analog signal output. It is a low voltage IC which utilizes around

+5VDC of power. Output pin gives a simple analog voltage that is sprightly relative to the Celsius (centigrade) temperature. Pin 2 gives a yield of 1 millivolts for every 0.1°C (10mV per degree).So to get the degree esteem in Celsius, all that must be done is to take the voltage output and division by 10-this give out the worth degrees in Celsius

* 1. ***Pulse Sensor:***

The main components pulse rate sensor comprises of are LED or light emitting diode and a photodiode. The heart beat beats makes a fluctuation in the progression of blood tissues. At the point when a tissue of body is flashed with the light source, it either reflects (a finger tissue) or let the light through (earlobe). A portion of the glow is devoured by the tissue blood and the reflected or the transmitted light is gathered by the photodiode. The measure of light devoured varies on the blood volume in that tissue. The photodiode yield is gotten in electrical sign structure and is corresponding to the beat rate. Here, the sensor reads the amount of blood volume coursing through fingers.

***Table 1****: Specification of System*

* 1. ***NodeMCU***

|  |  |  |
| --- | --- | --- |
| Module | Item | Specification |
| LilyPad Arduino | Operating voltage, digital pins, Flash  memory | 5V, 14,16KB |
| Temperature sensor | Temperature range, power, output impedance | 55Cto150C,43 0V,0 .1W for 1mA load |
| Pulse sensor | LED, gain, Power | InfraredLED,1 00,3.3V |
| NodeMCU Wi-Fi module | Power ,baud rate  ,range | 3.3V,9600BPS  , Up to 10m |
| Lcd | Power, display | 5V,plasma  display |

It is an open-source firmware and development kit that helps you to prototype or builds IoT products. The ESP8266 is mounted on NODEMCU. This module is used to establish connection between Arduino and IOT Platform

The ESP8266 Wi-Fi module could be a self- contained SOC with incorporated TCP/IP protocol stack which will offer any controller access to Wi- Fi network. It uses 802.11 b/g/n protocols. Standby power consumption is a smaller amount than 0.1mW.

* 1. ***IoT Platform:***

Internet of Things (IoT) characterizes a creating pattern where a broad number of devices (things) are linked to the web. These linked devices communicate with encompassing systems and individuals and often provide sensor data to cloud storage and cloud computing resources where the information is refined and evaluated to realize important insights. Modest shared computing capacity and expanded gadget availability is empowering this pattern.

It has three distinct components: storage, data analysis, and visualization. The system is intended for long haul stockpiling of patient's biomedical data also helping healthcare experts with diagnostic data. Cloud based clinical information stockpiling and the forthright difficulties have been broadly tended to in the writing

IoT solutions are made for several vertical applications like atmosphere and environmental factors checking and control, wellbeing observing, transport following, motorized plants observing and control, and personal home computerization.

Use the ThingSpeak platform to send data to the public cloud from any Internet-enabled device.

1. You would then be able to design action and alerts upheld your continuous information and enable the worth of your data through visual tools.
2. Use the ThingSpeak offers a platform for developers that enable them to simply capture sensors data and switch it into useful information.
   1. **RESULTS**

After designing the glove, the working of the glove was tested and the results were analysed.

Testing of Pulse Rate or Heart Rate

For adults (18 years of age and older), ordinary resting pulse is somewhere in the range of 60 and 100 beats for every moment (bpm) yet relying upon the individual's state of being and age this parameter changes.

For example for a 30-year-old person, The maximum heart rate must be within 95 to 162 bpm

Calculations for pulse rate for a 30 year old healthy person are as follows:

The range of heart rate is calculated by removing the age from 220.i.e: 220 – 30 = 190.

* 50 percent: 190 x 0.50 = 95 bpm
* 85 percent: 190 x 0.85 = 162 bpm

*Fig. 3 Picture Displaying Pulse rate (bpm) of a 21 years old healthy person on glove*



*Fig. 4 Variation of Pulse rate and temperature sensor in thingspeak*

For a 60-year-old individual, the objective zone would be somewhere in the range of 80 and 136 bpm

We have tested this glove with a 21 years old person; the calculated heartbeat range must be within 99.5 to 169.5 bpm, because =220-21=199

Minimum Heart rate=0.5\*199=99.5 Maximum Heart rate=0.85\*199=169.5

After testing our glove with a normal person of 21 years of age, the heartbeat per minute displayed on the displaying module was 88bpm as shown in the Fig.1. However, the complete variation of pulse rate with time is observed using serial monitor and it is shown in Fig.2.

Testing of Body Temperature

To measure the temperature of the human body, LM35 sensor has been used. It senses the heat from the body and converts it into the voltage output in millivolts (Vout), which is further processed/ evaluated to give the thermal reading in degree Celsius.

It can read temperature in range as low as -55 degree Celsius to as high as +150 degree Celsius. The voltage signal of the tempc=Vout/0.01;

tempc=(tempc/10.0);

Ttemp=tempc\*100;

The final reading of temperature sensor displayed on the 7- segment display was 36.4ᶛC.

Once the data was collected on the Arduino Platform ,it is further transmitted to the ThingSpeak Server using NodeMCU for remote access.

To access the information on the server, we need to create a channel for transmitting and receiving the information as shown in figure:

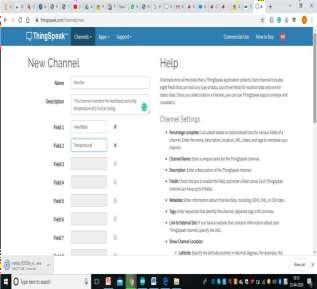
API keys enable you to write on or read data from a private created channel. There are two types of API keys in the channel:

**Write API Key**: Use this key to write data to a channel.

**Read API Keys**: Use this key to allow other people to view your private channel feeds and charts.

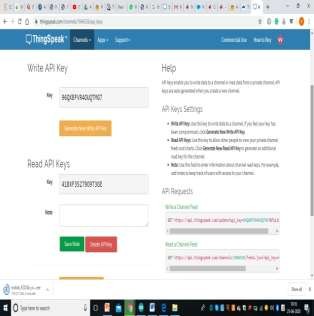


*Fig. 5 Picture Displaying body temperature (deg. C) of a 21 years old healthy person on glove*.



*Fig. 6 Screenshot of Channel information on Things Speak.*

And thus the entire data of our sensors can be monitored remotely by the doctor. He can prescribe the medicines to the patient accordingly.



*Fig. 7 Screenshot of API Key on Things Speak*

**LOCKDOWN WORK PROGRESS**

1. **ARDUINO UNO**

We are using latest Arduino Uno uses an atmega328 microcontroller which is manufactured by Atmel, giving you input voltage of around 7 volt or 12 volt to this port. The operating voltage of the microcontroller is 5 volt. It has a CPU speed of 16 megahertz there are analog input or output pins which is in 6 in number there are the 14 digital input/output pins out of which 6 can be used for pulse width modulation.

**ADVANTAGES**

1. ***Effortless functions:*** During coding of Arduino, you will notice some functions which make the life so easy. Another advantage of Arduino is its automatic unit conversion capability. You can say that during debugging you don’t have to worry about the units conversions.
2. ***Examples of codes:*** Another big advantage of Arduino is its library of examples present inside the software of Arduino.
3. ***Ready to Use:*** The biggest advantage of Arduino uno is its ready to use the structure. As Arduino uno comes into a complete package form which includes the [5V regulator,](http://engineerexperiences.com/5-volt-supply.html) a burner, an oscillator, a micro-controller, [serial communication](http://engineerexperiences.com/serial-com-with-pc.html) interface, an LED and headers for the connections.
4. **ESP8266 WIFI MODULE**

The **ESP8266** is a very user friendly and low cost device to provide internet connectivity to your projects. The module can work both as a Access point (can create hotspot) and as a station (can connect to Wi-Fi), hence it can easily fetch data and upload it to the internet making **Internet of Things** as easy as possible.

**ESP8266-01 Features**

* + Low cost, compact and powerful Wi-Fi Module
  + Power Supply: +3.3V only
  + Current Consumption: 100mA
  + I/O Voltage: 3.6V (max)
  + I/O source current: 12mA (max)
  + Built-in low power 32-bit MCU @ 80MHz
  + 512kB Flash Memory
  + Can be used as Station or Access Point or both combined
  + Supports Deep sleep (<10uA)
  + Supports serial communication hence compatible with many development platform like Arduino
  + Can be programmed using Arduino IDE or AT- commands or Lua Script

1. **1602 DISPLAY MODULE**

LCD modules are very commonly used in most embedded projects, the reason being its cheap price, availability and programmer friendly. Most of us would have come across these displays in our day to day life, either at PCO’s or calculators.

**16×2 LCD** is named so because; it has 16 Columns and 2 Rows. There are a lot of combinations available like, 8×1, 8×2, 10×2, 16×1, etc. but the most used one is the 16×2 LCD. So, it will have (16×2=32) 32 characters in total and each character will be made of 5×8 Pixel Dots.

**Features of 16×2 LCD module**

* + Operating Voltage is 4.7V to 5.3V
  + Current consumption is 1mA without backlight
  + Alphanumeric LCD display module, meaning can display alphabets and numbers
  + Consists of two rows and each row can print 16 characters.
  + Each character is build by a 5×8 pixel box
  + Can work on both 8-bit and 4-bit mode
  + It can also display any custom generated characters
  + Available in Green and Blue Backlight

1. **REFERENCES**
   1. Selvaraj, Sureshkumar, and Suresh Sundaravaradhan.

"Challenges and opportunities in IoT healthcare

systems: a systematic review." *SN Applied Sciences*

2, no. 1 (2020): 139.

* 1. Kumara K R, Ankita Kadam, Neha Rane, Shraddha Vernekar, Asma Gouda,"Sensor Based Wearable System to Assist Paralytic Patient with Continuous Health Monitoring" in May,2018
  2. Omnia Hassanin ,Luay Fraiwan, Tasnim Basmaji, "A Mobile Mental Health Monitoring System: A Smart Glove" in 2018
  3. Dr. G. Mahammad Gouse, Mir Abdul Samim Ansari, Shubiya Sultana, Swathi Jha, Sruthi Krishna.U, "Advanced Data Peripheral Glove with Real-time Vital Health Monitoring System" in 2017
  4. Ribin Jones S.B, Dr. N. Kumar, Swaraj S Paul, "Health Monitoring Wearable Glove" in September,2018
  5. [Sumit Majumder,](https://www.ncbi.nlm.nih.gov/pubmed/?term=Majumder%20S%5BAuthor%5D&amp;cauthor=true&amp;cauthor_uid=28085085)1 [Tapas Mondal,](https://www.ncbi.nlm.nih.gov/pubmed/?term=Mondal%20T%5BAuthor%5D&amp;cauthor=true&amp;cauthor_uid=28085085)2 and [M. Jamal Deen](https://www.ncbi.nlm.nih.gov/pubmed/?term=Deen%20MJ%5BAuthor%5D&amp;cauthor=true&amp;cauthor_uid=28085085) article on Wearable Sensors for Remote Health Monitoring.
  6. M. Sathya , S. Madha , K. Jayanthi, Mirza Mansoor Baig & Hamid Gholamhosseini Internet of things (IoT) based health monitoring system and challenges 2018 international journal of engineering and technology

**V. CONCLUSION**

This paper discusses the construction of a smart wearable glove, which is used to track the health of patient. This device is used to measure the pulse rate and temperature of the patient.

It has low cost, it is easily accessible and used for personal health monitoring and are expected for early revelation and better treatment of various medical conditions as well as disease prevention and better understanding and self-management of diseases.

This system is very helpful for the far-off monitoring of patients staying at home. The system empowers specialists to analyze and distinguish the conditions of versatility and help heart related problem of the subjects without needing them to be available at the emergency clinic.