

SECURED HEALTHCARE SYSTEM USING IoT

Submitted in partial fulfilment of the requirements for the award of

Bachelor of Engineering Degree

in

Electronics and Communication Engineering

BY

J. THANVITH (38130092)

K. BHARATH (38130105)



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION
ENGINEERING**

SCHOOL OF ELECTRICAL AND ELECTRONICS

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO UNIVERSITY)**

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600119.

MAY – 2022



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade “A” by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

www.sathyabama.ac.in



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **J. THANVITH (38130092)** and **K. BHARATH (37130360)**, who carried out the project entitled “**SECURED HEALTHCARE SYSTEM USING IoT**” under my supervision from OCTOBER 2021 to MAY 2022

Internal Guide

Dr. S. EMALDA ROSLIN, M.E., Ph.D.,

Head of the Department

Dr. T. RAVI, M.E., Ph.D.,

Submitted for Viva voce Examination held on_____

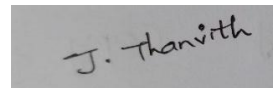
Internal Examiner

External Examiner

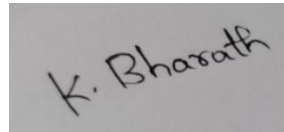
DECLARATION

We, **J.THANVITH (38130092)** and **K.BHARATH (38130105)**, hereby declare that the Project Report entitled **SECURED HEALTHCARE SYSTEM USING IoT** is done under the guidance of **Dr. S. EMALDA ROSLIN, M.E., Ph.D.**, is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in **Electronics and Communication Engineering**.

1.



2.



DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my profound gratitude to **Dr. N.M. NANDHITHA, M.E., Ph.D.**, Prof. & Dean, School of Electrical and Electronics and **Dr. T. RAVI, M.E., Ph.D.**, Head of the Department of Electronics and Communication Engineering, Sathyabama University for having been a constant source of support and encouragement for the completion of the project.

I consider my self-most fortunate and immensely privileged to have worked under the guidance of **Dr. S. EMALDA ROSLIN, M.E., Ph.D.**, Department of Electronics and Communication Engineering to complete this project successfully.

I express our heartfelt thanks to our family members and all our friends who helped me to complete our project successfully & finally, I thank the exciting Super Natural Power, the Almighty GOD.

ABSTRACT

Technology plays the most important role in healthcare not only for sensory devices but besides in communication, recording and display device. It is very important to monitor various medical parameters and post operational days. Therefore, the latest development in healthcare communication method using IoT is adapted. Internet of things serves as a catalyst for the healthcare and plays prominent role in wide range of healthcare applications. In this project the Arduino microcontroller is used as a gateway to communicate to the various sensors such as Temperature sensor, Respiration sensor, Pressure sensor, Humidity sensor and Pulse sensor.

The microcontroller picks up the sensor data and sends it to the network through IoT and hence provides real time monitoring of the health care parameters for doctors. The data can be access anytime by the doctor. Other than the main issue in remote patient monitoring system is that the data as to be securely transmitted to the destination end and provision is made to allow only authorized user to access the data. The security problem is be addressed by transmitting the data through the password protected internet which will be encrypted by standard AES128 and the Doctors can access the records by logging to the html webpage. For this reason, quick provisional medication can be easily done by this system. This system is capable with low power consumption capability, easy setup, high performance and time to time response.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	v
	LIST OF FIGURES	viii
	LIST OF ABBREVIATIONS	ix
1	INTRODUCTION	1
	1.1 PROTOTYPE OF IoT BASED HEALTHCARE	1
	1.2 MOTIVATION FROM REAL SCENARIOS	4
	1.3 OBJECTIVES	5
2	LITERATURE REVIEW	6
	2.1 LITERATURE SURVEY	6
	2.2 SUMMARY OF EXISTING SYSTEM	7
3	PROPOSED METHDOLOGY	9
	3.1 BLOCK DIAGRAM	9
	3.2 SYSTEM DESIGN	9
	3.3 HARDWARE REQUIREMENT AND CONFIGURATION	19
	3.4 ARDUINO and NodeMCU DESCRIPTION	19
	3.5 TECHNICAL SPECIFICATIONS	29
	3.6 COMMUNICATION AND TESTING	42
4	NEURAL NETWORKS FOR SECURED HEALTHCARE SYSTEM	45
	4.1 BPN PARAMETERS	45
	4.2 NEURAL NETWORK TRAINING (nntraintool)	47

CHAPTER NO	TITLE	PAGE NO
	4.3 COMPARISON OF ACTUAL AND DESIRED OUTPUT	47
	4.4 CODE	48
5	CONCLUSION	51
	5.1 SUMMARY	51
	5.2 OUTPUT	52
	REFERENCES	56

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
3.1	Block diagram	9
3.2	Structure of embedded systems	12
3.3	Embedded Systems having Heterogeneous Architectures	14
3.4	General Block of Power Supply Unit	16
3.5	Pin specifications	24
3.6	Structure of blink test	28
3.7	Connection diagram for the blinking test	28
3.8	Firmware of NodeMCU	29
3.9	DHT11 sensor	30
3.10	Start" and "Response" signals	30
3.11	Start, Response and Data signals in sequence	31
3.12	Heartbeat Sensor methodology	35
3.13	Pressure sensor	37
3.14	UNO board	38
4.1	BPN architecture	46
4.2	Neural Network Training (nntraintool)	47
4.3	Comparison of Actual and Design Output	47

LIST OF ABBREVIATIONS

AJDSP	- Android Java-DSP
BPN	- Back Propagation Network
CG RAM	- Character Generator RAM
DD RAM	- Display Data RAM
DR	- Data Register
DSP	- Digital Signal Processing
IDE	- Integrated development environment
IR	- Instruction Register
M2M	- Machine-To-Machine

CHAPTER 1

INTRODUCTION

1.1 PROTOTYPE OF IoT BASED HEALTHCARE

This paper presents a prototype machine-to-machine (M2M) healthcare solution that combines mobile and IPv6 techniques in a wireless sensor network to monitor the health condition of patients and provide a wide range of effective, comprehensive, and convenient healthcare services. A low-power embedded wearable sensor measures the health parameters dynamically, and is connected, according to the concept of IPv6 over low-power wireless personal area network, to the M2M node for wireless transmission through the internet or external IP-enabled networks via the M2M gateway. A visualization module of the server program graphically displays the recorded biomedical signals on Android mobile devices used by patients and doctors at the end of the networks in real-time. Our approach for a global M2M healthcare solution is managed to process the large amount of biomedical signals through the extended network combining IPv6 technique and mobile technology for daily lifestyle to users appropriately. This paper presents a working model which incorporates sensors to measure all these parameters like body temperature and Heart Beat rate and transfer it to the ARM 11 SMARTPHONE, so that the patient condition can be analyzed to by doctors in any part of the hospital wherever they are. Thus it reduces doctors work load and also gives more accurate results, wherever there is an abnormality felt by the patient, we have also incorporated saline monitoring system which gives an alarm when the saline bottle about to empty. The experimental setup can be operated for monitoring from anywhere covered by the Cellular (GSM) service by exchanging SMS messages with the remote mobile device. At the consultation unit, a dedicated application software is required to manage the follow of SMS messages from the mobile and display the temperature and heart beat of the patient. ISSN: 2278 – 7798

International Journal of Science, Engineering It is the responsibility of a nation to provide food, education and medical care to its citizens. Fulfilling this can promise effective efforts by human resource for the nation's development. Efforts should start right from providing timely and quality health assistance to pregnant ladies which will lead to the birth of healthy children. Here comes the need of technology which helps to take care of them. People in rural areas are not aware of proper medications and technological advancements to alleviate complications during pregnancy period. For instance, pregnant women should perform Ultra sound scans at least 2 times during pregnancy period to know more about fetal growth. Moreover, proper and timely checkups can ensure safe delivery. Women in rural areas lack knowledge about importance of proper medication. Medical Expenses are also unaffordable to them. Therefore in this system, some vital parameters like heartbeat, temperature and Kicking is measured. Sensors are attached in this system thus it helps to take reading and display on your mobile.

II. LITERATURE SURVEY

II.1 Health Monitoring Laboratories by Interfacing Physiological Sensors to Mobile Android Devices: Suhas Ranganath - Mahesh.K,

This paper describes, Android Java-DSP (AJDSP) as a mobile application that interfaces with sensors and enables simulation. This also helped in visualization of signal processing. In this system firstly there is creation of interface between both external sensors and on-board device sensors for monitoring the physiological parameter of human being. This paper also explored the trend of mobile sensing and adapted it towards improving digital signal processing (DSP), by building interfaces to medical sensor and external sensors. In this paper there is use of SHIMMER. It is a small wireless low power sensor International Journal of Advance Foundation and Research in Computer platform that can record and transmit physiological (Health related like ECG) and kinematic data in real-time. The drawback of this system is that it only monitors the patient which is admitted in the hospital. In this low power sensor are used.

According to present scenario, people prefer to consume junk foods due to work pressure which is unhealthy and it leads to obesity. Due to technological development and environmental factors people suffer because of stress which will mainly affect the pregnant ladies. Hence it is our duty to resolve the health issue of pregnant women. This paper presents a healthcare solution that combines web app and CC3200 techniques in a wireless sensor network to monitor the health condition

of patient and provide a wide range of effective, comprehensive, and convenient healthcare services. The specialist staying at a distance can monitor the pregnant ladies health condition so that he can save the life of the patient and also the infant. IOT technology is used so that we can monitor the patient condition easily from anywhere. Measures the health parameters dynamically and is connected to cloud using IOT, according to the concept of CC3200. This paper presents a working model which incorporates sensors to measure all these parameters like body temperature, pressure and pulse rate which is transferred to the microcontroller CC3200, so that the patient condition can be analyzed by doctors in any part of the hospital wherever they live. When patient reaches abnormal condition, an alarm will be passed to both the doctor and to the relation of the patient.

This paper considers or takes into account certain facts, which are heart attack and stroke as they are the major cause of hospitalization of the elder people. There is more chances of survival if the older people gets the treatment within an hour. it had also been developed. An android smart phone with accelerometer is used to detect a fall of the carrier, and this android device is known as healthcare device. The android phone is then connected to the monitoring system by using the TCP/IP protocol through Wi-Fi. Because of this system, elderly and chronically ill patients can stay independently in their own home and secure in the knowledge that they are being monitored. The drawback of this system is that it only considers elder people as there is more chances of sudden (emergency situation) outbreak in them like heart attack and stroke.

This system describes the design and implementation part of wearable ECG with the smart phone for the real time monitoring of health. In this system smart shirt are developed with ECG sensors and can be worn by any type of patient for monitoring his or her health in real time and get required treatment or prescription. These systems are mainly developed considering elder people in mind as they live alone in their homes. Therefore this system basically monitors the elderly people for self diagnosis purpose. The result of this system was the system could monitor and diagnose patients' heart conditions in real time, when they wear a sports-shirt with a ECG sensor in it. In addition to this, the system also provides graphical information with history management tools and an automatic emergency call system to the

patient to get the required treatment in time. The drawback of the system is that it only concentrates on elder people and it includes shirt (ECG sensor) for wearing which cost a lot.

This paper provides the image based system which acquires the ECG signal via digital camera; this information is performed on the tool like MATLAB and data sending through the internet network and stored in database. Then the original image is then availed to the doctor via Android mobiles. The purpose of this system is the vital signs and parameters from the ICU monitoring system and makes this data to be available to the doctor who may not be in the hospital and in the country. In case of any abnormality, the doctor is alerted by sending a notification from C2DM server to his mobile. The drawback of this paper is that, due to the slow internet connection the data will not be send to the doctor which is located remotely. The image is captured through the camera, which must be HD which cost a lot. The current high cost of healthcare and the trends of increasing costs due to the ageing society are important issues in the world. Thus the implementation of measures for cost reduction is one of the most significant current trends in healthcare. One way by means of which the cost reduction can be achieved by increasing the efficiency within healthcare organizations, for e.g., by reducing of the number of “days of stay in a hospital”, reduction of costly treatment programmes, and the number of face-to-face consultations. Another way is prevention by detecting people at risk in an early stage and providing guidance to prevent costly hospitalization. It is also important to motivate people to maintain a healthy lifestyle.

1.2 MOTIVATION FROM REAL SCENARIOS

We have been inspired by different scenarios that take place in our respective villages and cities. Many people living in rural regions face many problems regarding the health system due to a lack of proper machinery in their respective hospitals. Normal health patients the sense faced with minor problems which include BP, Sugar, and different problems for their regular checkups they need to reach the major hospitals. This affects badly on their financial structure and this is the main reason behind the clinics will loot their money. So, we have come up with an innovative low-budget smart health care system that helps the rural

people to get out these problems and clear all their regular checkups without any irritative problems.

1.3 OBJECTIVES

The detailed objectives are:

- To calculate temperature, humidity and pressure additionally.
- To store the data in digital mode so that anywhere can access.
- To secure the data with password.
- To produce health report of a patient frequently without any disturbance
- Arrangement of buzzer so that abnormal values can be notified easy.

CHAPTER 2

LITERATURE REVIEW

2.1 LITERATURE SURVEY

Literature Survey-1

TITLE - Secure Healthcare Data Aggregation and Transmission in IoT—A Survey

AUTHORS - ATA ULLAH , MUHAMMAD AZEEM, HUMAIRA ASHRAF

PUBLISHED - Received December 26, 2020, accepted January 12, 2021, date of publication January 19, 2021, date of current version January 29, 2021.

DESCRIPTION - This system is mainly designed for real time monitoring of the vital parameters of the patients. The patient's vitals parameters are transmitted to smart phones and laptops of the authorized person using a cloud server. These information can be stored and analyzed for further analysis and decision making. By determining the pattern of the parameters which is observed, the nature of the disease can be predicted.

Literature Survey-2

TITLE - Integrated Secure Health Domain System using IoT

AUTHORS - Nagarjuna Valeti, V.Ceronmani Sharmila

PUBLISHED - ISSN: 2277-3878, Volume-8 Issue-4, November 2019

DESCRIPTION-It is also important at rural areas or villages where nearby clinics can be in touch with city hospitals about their patient's health condition. This work

presents a smart health monitoring system that uses biomedical sensors to check patient's condition and uses internet to inform the concerned.

Literature Survey-3

TITLE - IoT based Smart Healthcare Monitoring Systems: A Literature Review

AUTHORS - R. Alekya, Neelima Devi Boddeti, K. Salomi Monica, Dr.R. Prabha

PUBLISHED - European Journal of Molecular & Clinical Medicine ISSN 2515-8260
Volume 7, Issue 11, 2021

DESCRIPTION- This paper helps to imagine how IoT can be incorporated into complex healthcare procedures. The "Mobile Healthcare Management System (HMS)" is one of the main IoT apps that link the Internet to mobile sensors, people, clinicians, networks and other connected devices.

Literature Survey-4

TITLE - A Review on Internet of Things (IoT) in Healthcare

AUTHORS - Mustafa Abdullah Azzawi, Rosilah Hassan and Khairul Azmi Abu Bakar

PUBLISHED - November 20-2016

DESCRIPTION-In this paper a review for IoT healthcare usage and a novel authentication mechanism for IoT networks have been proposed. The architecture of reliable and secure healthcare architecture has been proposed. ECC algorithm over the CoAP protocol has been used. The proposed authentication approach provides an efficient authentication mechanism with high security.

2.2 SUMMARY OF EXISTING SYSTEM

- The health care sensors are playing an essential role in hospitals. The patient monitoring system is one of the major developments because of its innovative technology.

- An automatic wireless health monitoring system is used to measure patient's body temperature and heartbeat by using embedded technology.
- The proposed system uses both the sensors like heartbeat sensor and temperature sensor. These sensors mainly involves in monitoring the condition of the patient.
- But the sensor data cannot be upload to cloud
- So each and every time doctor will go there and check the parameter
- In some conditions doctor may busy means they never monitoring the person.

CHAPTER 3

PROPOSED METHDOLOGY

3.1 BLOCK DIAGRAM

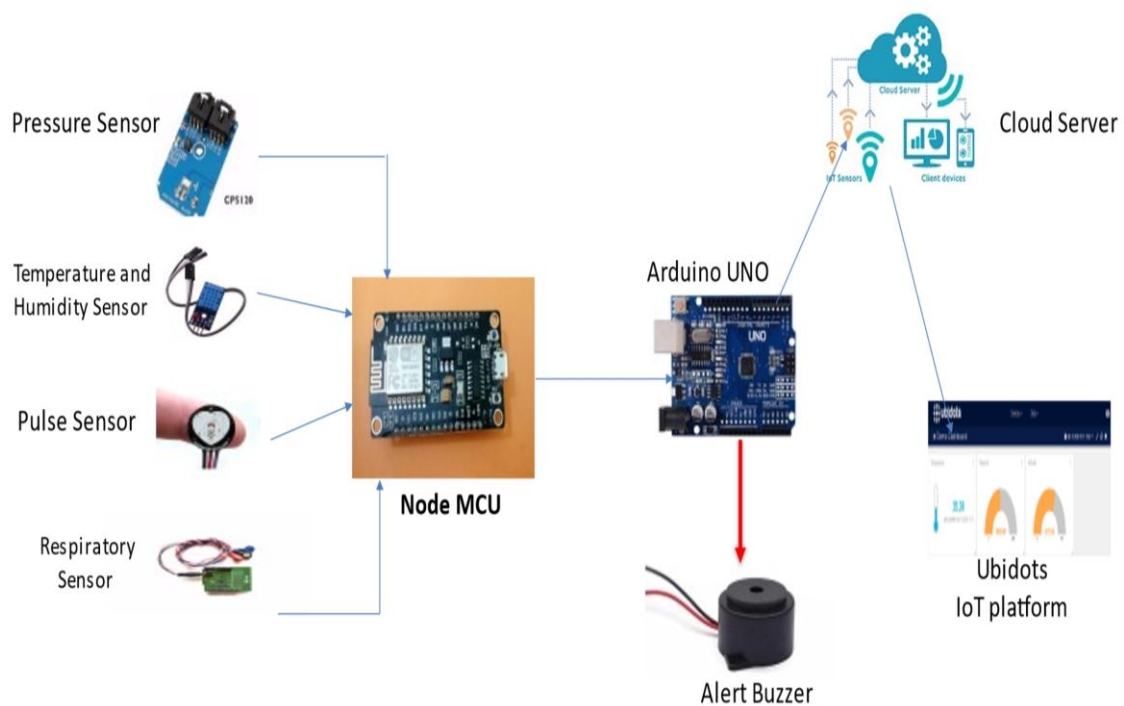


Fig. 3.1: Block diagram

3.2 SYSTEM DESIGN

Embedded Systems

Overview of embedded systems

An embedded system is a special-purpose computer system designed to perform one or a few dedicated functions, often with real-time computing constraints. It is usually embedded as part of a complete device including hardware and

mechanical parts. In contrast, a general-purpose computer, such as a personal computer, can do many different tasks depending on programming. Embedded systems have become very important today as they control many of the common devices we use.

Since the embedded system is dedicated to specific tasks, design engineers can optimize it, reducing the size and cost of the product, or increasing the reliability and performance. Some embedded systems are mass-produced, benefiting from economies of scale.

Physically, embedded systems range from portable devices such as digital watches and MP3 players, to large stationary installations like traffic lights, factory controllers, or the systems controlling nuclear power plants. Complexity varies from low, with a single microcontroller chip, to very high with multiple units, peripherals and networks mounted inside a large chassis or enclosure.

In general, "embedded system" is not an exactly defined term, as many systems have some element of programmability. For example, Handheld computers share some elements with embedded systems — such as the operating systems and microprocessors which power them — but are not truly embedded systems, because they allow different applications to be loaded and peripherals to be connected.

Embedded systems provide several functions

- Monitor the environment; embedded systems read data from input sensors. This data is then processed and the results displayed in some format to a user or users
- Control the environment; embedded systems generate and transmit commands for actuators.
- Transform the information; embedded systems transform the data collected in some meaningful way, such as data compression / decompression.

Although interaction with the external world via sensors and actuators is an important aspect of embedded systems, these systems also provide functionality specific to their applications. Embedded systems typically execute applications such as control laws, finite state machines, and signal processing algorithms. These systems must also detect and react to faults in both the internal computing environment as well as the surrounding electromechanical systems.

There are many categories of embedded systems, from communication devices to home appliances to control systems. Examples include;

- Communication devices
e.g.: modems, cellular phones
- Home Appliances
e.g.: CD player, VCR, microwave oven
- Control Systems
e.g.: Automobile anti-lock braking systems, robotics, satellite control

Block diagram of an embedded system

An embedded system usually contains an embedded processor. Many appliances that have a digital interface -- microwaves, VCRs, cars -- utilize embedded systems. Some embedded systems include an operating system. Others are very specialized resulting in the entire logic being implemented as a single program. These systems are embedded into some device for some specific purpose other than to provide general purpose computing

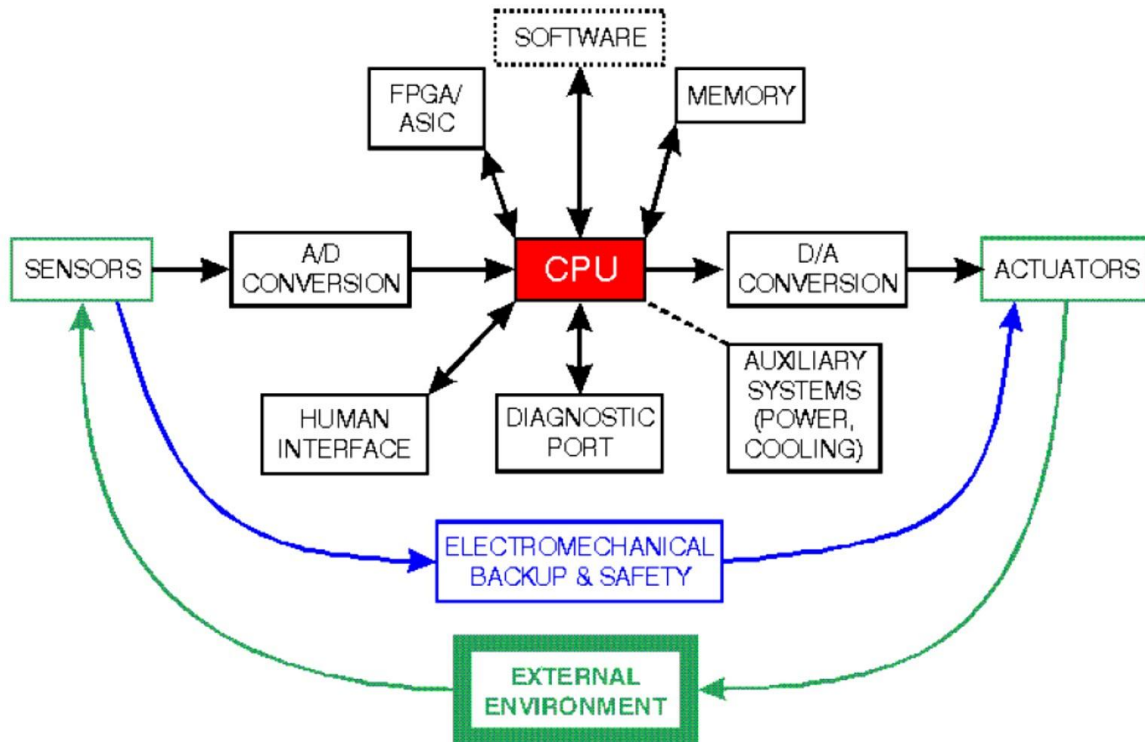


Fig. 3.2: Structure of embedded systems

Characteristics of embedded systems

Embedded systems are characterized by a unique set of characteristics. Each of these characteristics imposed a specific set of design constraints on embedded systems designers. The challenge to designing embedded systems is to conform to the specific set of constraints for the application.

Application Specific Systems

Embedded systems are not general-purpose computers. Embedded system designs are optimized for a specific application. Many of the job characteristics are known before the hardware is designed. This allows the designer to focus on the specific design constraints of a well-defined application. As such, there is limited user reprogram ability. Some embedded systems, however, require the flexibility of reprogram ability. Programmable DSPs are common for such applications.

Reactive Systems

As mentioned earlier, a typical embedded systems model responds to the environment via sensors and control the environment using actuators. This requires embedded systems to run at the speed of the environment. This characteristic of embedded system is called “reactive”. Reactive computation means that the system (primarily the software component) executes in response to external events. External events can be either periodic or aperiodic. Periodic events make it easier to schedule processing to guarantee performance. Aperiodic events are harder to schedule. The maximum event arrival rate must be estimated in order to accommodate worst case situations. Most embedded systems have a significant reactive component. One of the biggest challenges for embedded system designers is performing an accurate worst case design analysis on systems with statistical performance characteristics (e.g., cache memory on a DSP or other embedded processor). Real time system operation means that the correctness of a computation depends, in part, on the time at which it is delivered. Systems with this requirement must often design to worst case performance. But accurately predicting the worst case may be difficult on complicated architectures. This often leads to overly pessimistic estimates erring on the side of caution. Many embedded systems have a significant requirement for real time operation in order to meet external I/O and control stability requirements. Many real-time systems are also reactive systems.

Distributed Systems

A common characteristic of an embedded system is one that consists of communicating processes executing on several CPUs or ASICs which are connected by communication links. The reason for this is economy. Economical 4 8-bit microcontrollers may be cheaper than a 32-bit processors. Even after adding the cost of the communication links, this approach may be preferable. In this approach, multiple processors are usually required to handle multiple time-critical tasks. Devices under control of embedded systems may also be physically distributed.

Heterogeneous Architectures

Embedded systems often are composed of heterogeneous architectures (Fig 3.3). They may contain different processors in the same system solution. They may also be mixed signal systems. The combination of I/O interfaces, local and remote memories, and sensors and actuators makes embedded system design truly unique. Embedded systems also have tight design constraints, and heterogeneity provides better design flexibility.



Fig. 3.3: Embedded Systems having Heterogeneous Architectures

Harsh environment

Many embedded systems do not operate in a controlled environment. Excessive heat is often a problem, especially in applications involving combustion (e.g., many transportation applications). Additional problems can be caused for embedded computing by a need for protection from vibration, shock, lightning, power supply fluctuations, water, corrosion, fire, and general physical abuse.

System safety and reliability

As embedded system complexity and computing power continue to grow, they are starting to control more and more of the safety aspects of the overall system. These safety measures may be in the form of software as well as hardware control. Mechanical safety backups are normally activated when the computer system loses control in order to safely shut down system operation. Software safety and reliability is a bigger issue. Software doesn't normally "break" in the sense of

hardware. However software may be so complex that a set of unexpected circumstances can cause software failures leading to unsafe situations. Discussion of this topic is outside the scope of this book, but the challenges for embedded designers include designing reliable software and building cheap, available systems using unreliable components. The main challenge for embedded system designers is to obtain low-cost reliability with minimal redundancy.

Control of physical systems

One of the main reasons for embedding a computer is to interact with the environment. This is often done by monitoring and controlling external machinery. Embedded computers transform the analog signals from sensors into digital form for processing. Outputs must be transformed back to analog signal levels. When controlling physical equipment, large current loads may need to be switched in order to operate motors and other actuators. To meet these needs, embedded systems may need large computer circuit boards with many non-digital components. Embedded system designers must carefully balance system tradeoffs among analog components, power, mechanical, network, and digital hardware with corresponding software.

Small and low weight

Many embedded computers are physically located within some larger system. The form factor for the embedded system may be dictated by aesthetics. For example, the form factor for a missile may have to fit inside the nose of the missile. One of the challenges for embedded systems designers is to develop non-rectangular geometries for certain solutions. Weight can also be a critical constraint. Embedded automobile control systems, for example, must be light weight for fuel economy. Portable CD players must be light weight for portability purposes.

Cost sensitivity

Cost is an issue in most systems, but the sensitivity to cost changes can vary dramatically in embedded systems. This is mainly due to the effect of computer

costs have on profitability and is more a function of the proportion of cost changes compared to the total system cost.

Power management

Embedded systems have strict constraints on power. Given the portability requirements of many embedded systems, the need to conserve power is important to maintain battery life as long as possible. Minimization of heat production is another obvious concern for embedded systems.

Power Supply

All electronic circuits work only in low DC voltage, so we need a power supply unit to provide the appropriate voltage supply for their proper functioning. This unit consists of transformer, rectifier, filter & regulator. AC voltage of typically 230volts rms is connected to a transformer voltage down to the level to the desired ac voltage. A diode rectifier that provides the full wave rectified voltage that is initially filtered by a simple capacitor filter to produce a dc voltage. This resulting dc voltage usually has some ripple or ac voltage variation. A regulator circuit can use this dc input to provide dc voltage that not only has much less ripple voltage but also remains the same dc value even the dc voltage varies some what, or the load connected to the output dc voltages changes.

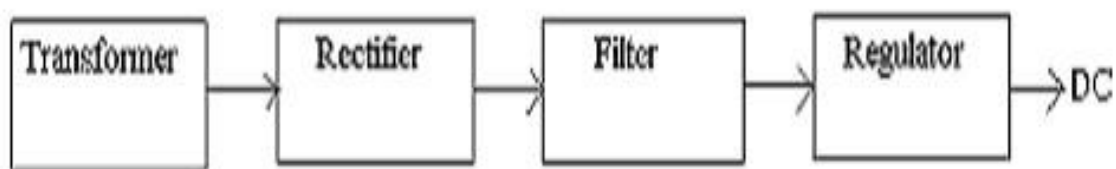


Fig. 3.4: General Block of Power Supply Unit

Transformer

A transformer is a static piece of which electric power in one circuit is transformed into electric power of same frequency in another circuit. It can raise or lower the voltage in the circuit, but with a corresponding decrease or increase in

current. It works with the principle of mutual induction. In our project we are using a step down transformer to providing a necessary supply for the electronic circuits. Here we step down a 230volts ac into 12volts ac.

Rectifier

A dc level obtained from a sinusoidal input can be improved 100% using a process called full wave rectification. Here in our project for full wave rectification we use bridge rectifier. From the basic bridge configuration we see that two diodes(say D2 & D3) are conducting while the other two diodes (D1 & D4) are in off state during the period $t = 0$ to $T/2$.Accordingly for the negative cycle of the input the conducting diodes are D1 & D4 .Thus the polarity across the load is the same.

In the bridge rectifier the diodes may be of variable types like 1N4001, 1N4003, 1N4004, 1N4005, 1N4007 etc.... can be used. But here we use 1N4007, because it can withstand up to 1000v.

Filters

In order to obtain a dc voltage of 0 Hz, we have to use a low pass filter. So that a capacitive filter circuit is used where a capacitor is connected at the rectifier output& a dc is obtained across it. The filtered waveform is essentially a dc voltage with negligible ripples & it is ultimately fed to the load.

Registers

The controller IC has two 8 bit registers, an instruction register (IR) and a data register (DR). The IR stores the instruction codes and address information for display data RAM (DD RAM) and character generator RAM (CG RAM). The IR can be written, but not read by the MPU.

The DR temporally stores data to be written to /read from the DD RAM or CG RAM. The data written to DR by the MPU, is automatically written to the DD RAM or CG RAM as an internal operation.

When an address code is written to IR, the data is automatically transferred from the DD RAM or CG RAM to the DR. data transfer between the MPU is then completed when the MPU reads the DR. likewise, for the next MPU read of the DR, data in DD RAM or CG RAM at the address is sent to the DR automatically. Similarly, for the MPU write of the DR, the next DD RAM or CG RAM address is selected for the write operation.

The dot-matrix liquid crystal display controller and driver LSI displays alphanumeric, Japanese kana characters, and symbols. It can be configured to drive a dot-matrix liquid crystal display under the control of a 4- or 8-bit microprocessor. Since all the functions such as display RAM, character generator, and liquid crystal driver, required for driving a dot-matrix liquid crystal display are internally provided on one chip, a minimal system can be interfaced with this controller/driver.

Regulators

The output voltage from the capacitor is more filtered & finally regulated. The voltage regulator is a device, which maintains the output voltage constant irrespective of the change in supply variations, load variations & temperature changes. Here we use fixed voltage regulator namely LM7805. The IC LM7805 is a +5v regulator which is used for microcontroller.

Features and Description Of Regulators

- Output Current up to 1A
- Output Voltages of 5, 6, 8, 9, 10, 12, 15, 18, 24V
- Thermal Overload Protection
- Short Circuit Protection
- Output Transistor Safe Operating Area Protection

3.3 HARDWARE REQUIREMENT AND CONFIGURATION

- Nodemcu
- Temperature sensor and humidity sensor
- Respiratory sensor
- Heartbeat sensor
- Pulse sensor

Arduino UNO board

Jumper cables

Hand Glove for easy representation

Software Requirement

- Arduino ide
- Arduino c

3.4 ARDUINO and NodeMCU DESCRIPTION

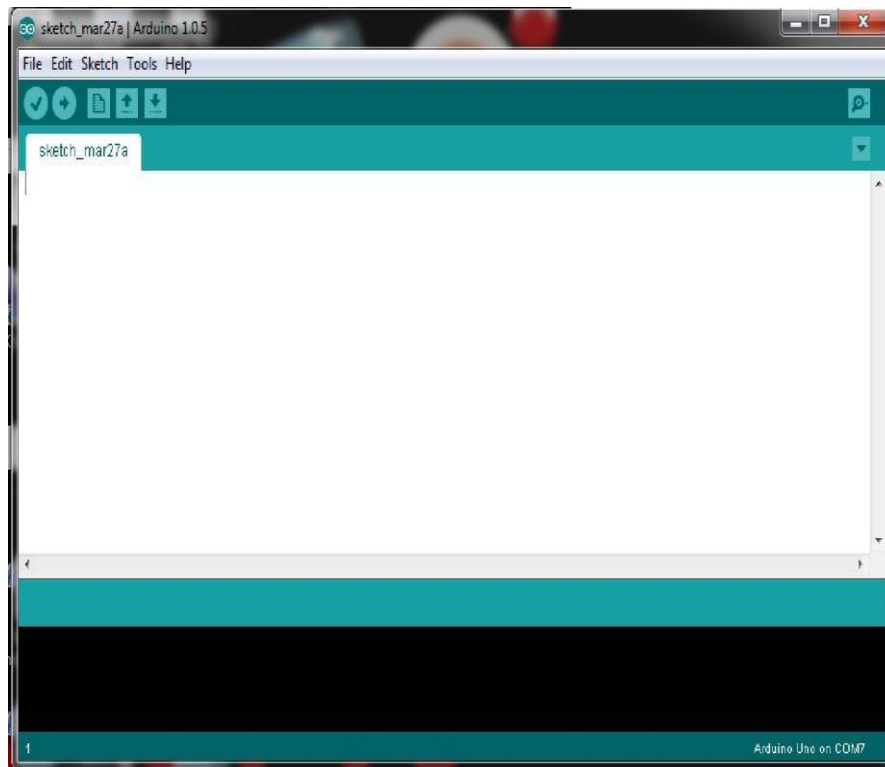
Arduino is a tool for making computers that can sense and control more of the physical world than your desktop computer. It's an open-source physical computing platform based on a simple microcontroller board, and a development environment for writing software for the board.

Arduino can be used to develop interactive objects, taking inputs from a variety of switches or sensors, and controlling a variety of lights, motors, and other physical outputs. Arduino projects can be stand-alone, or they can be communicate with software running on your computer. The boards can be assembled by hand or purchased preassembled; the open-source IDE can be downloaded for free.

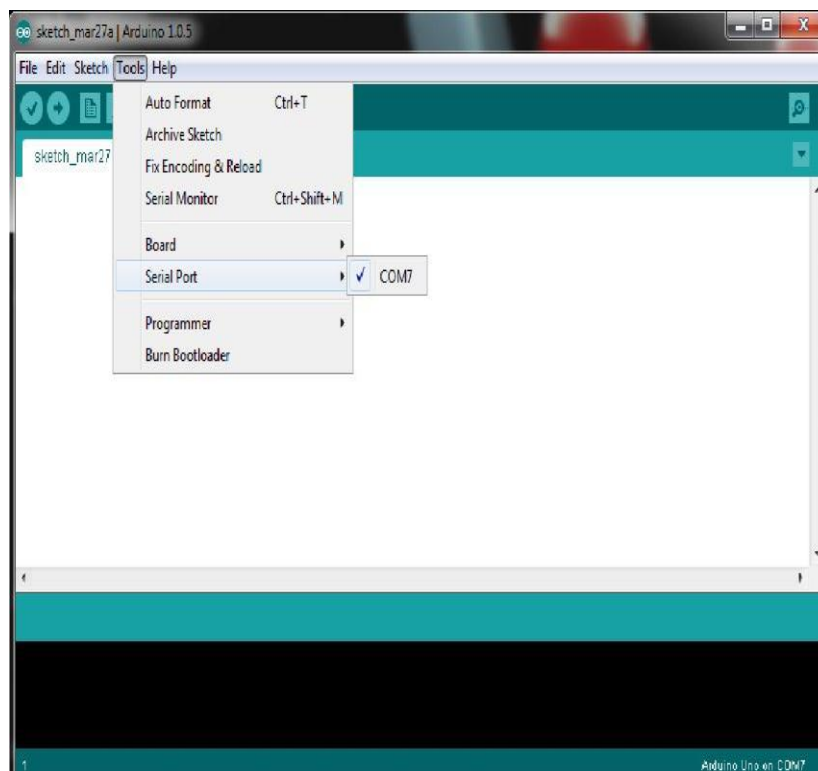
The Arduino programming language is an implementation of Wiring, a similar physical computing platform, which is based on the Processing multimedia programming environment.

1. *Open Arduino IDE as shown below*

Open Arduino IDE

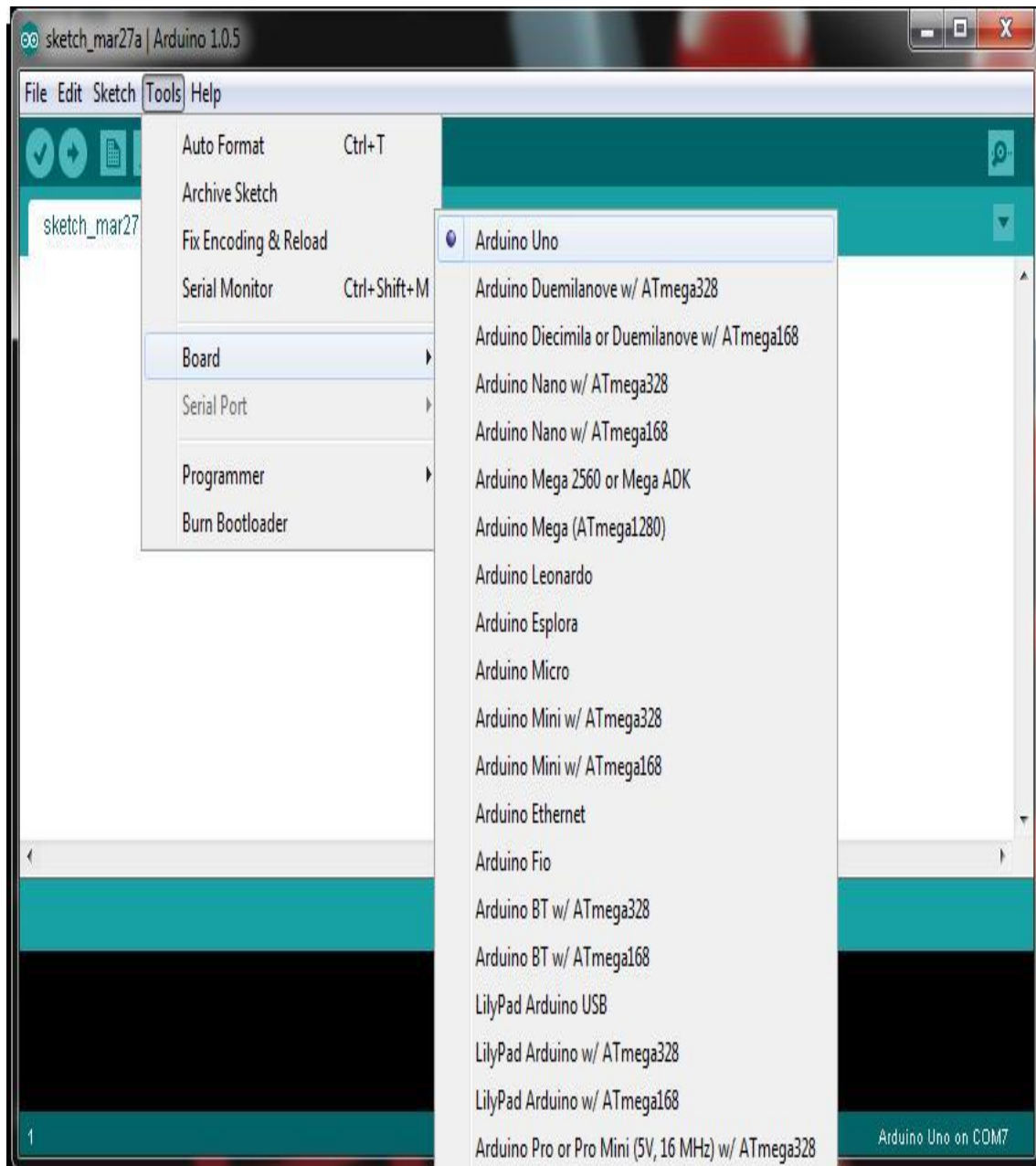


2. *Select the COM Port from tool*



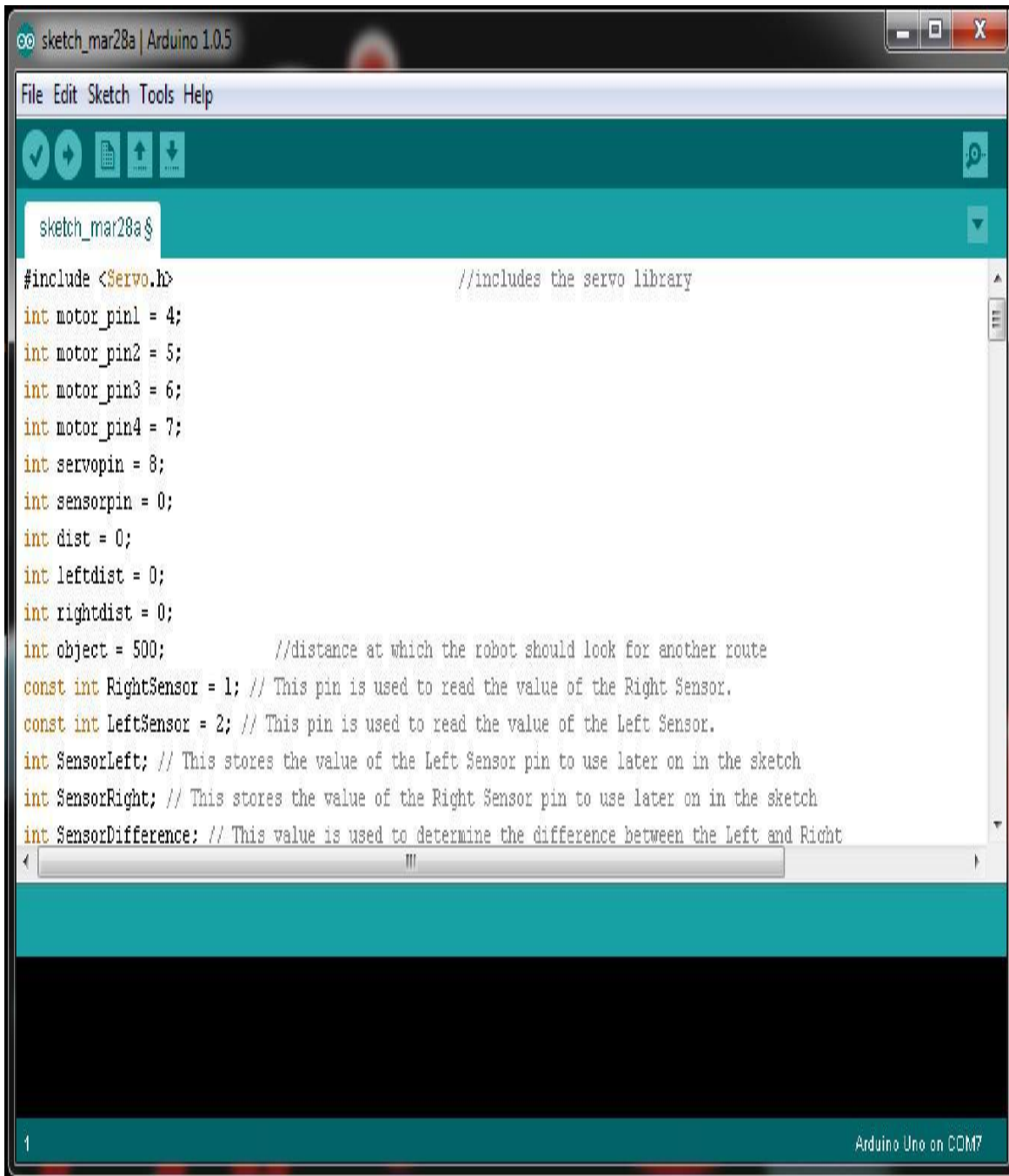
3. *Select the required Arduino board from Tools as shown below*

Select the required Arduino board



4. *Write the sketch in Arduino IDE*

Sketch in Arduino IDE



```
sketch_mar28a$
#include <Servo.h>                                //includes the servo library
int motor_pin1 = 4;
int motor_pin2 = 5;
int motor_pin3 = 6;
int motor_pin4 = 7;
int servopin = 8;
int sensorpin = 0;
int dist = 0;
int leftdist = 0;
int rightdist = 0;
int object = 500;                                //distance at which the robot should look for another route
const int RightSensor = 1; // This pin is used to read the value of the Right Sensor.
const int LeftSensor = 2; // This pin is used to read the value of the Left Sensor.
int SensorLeft; // This stores the value of the Left Sensor pin to use later on in the sketch
int SensorRight; // This stores the value of the Right Sensor pin to use later on in the sketch
int SensorDifference; // This value is used to determine the difference between the Left and Right
```

1 Arduino Uno on COM7

5. *Compile and upload the Sketch to Arduino board upload the Sketch to Arduino board*



NodeMCU Configuration

Specification

- Voltage: 3.3V.
- Wi-Fi Direct (P2P), soft-AP.

- Current consumption: 10uA~170mA.
- Flash memory attachable: 16MB max (512K normal).
- Integrated TCP/IP protocol stack.
- Processor: Tensilica L106 32-bit.
- Processor speed: 80~160MHz.
- RAM: 32K + 80K.
- GPIOs: 17 (multiplexed with other functions).
- Analog to Digital: 1 input with 1024 step resolution. • +19.5dBm output power in 802.11b mode
- 802.11 support: b/g/n.

Maximum concurrent TCP connections: 5.

Pin Definition

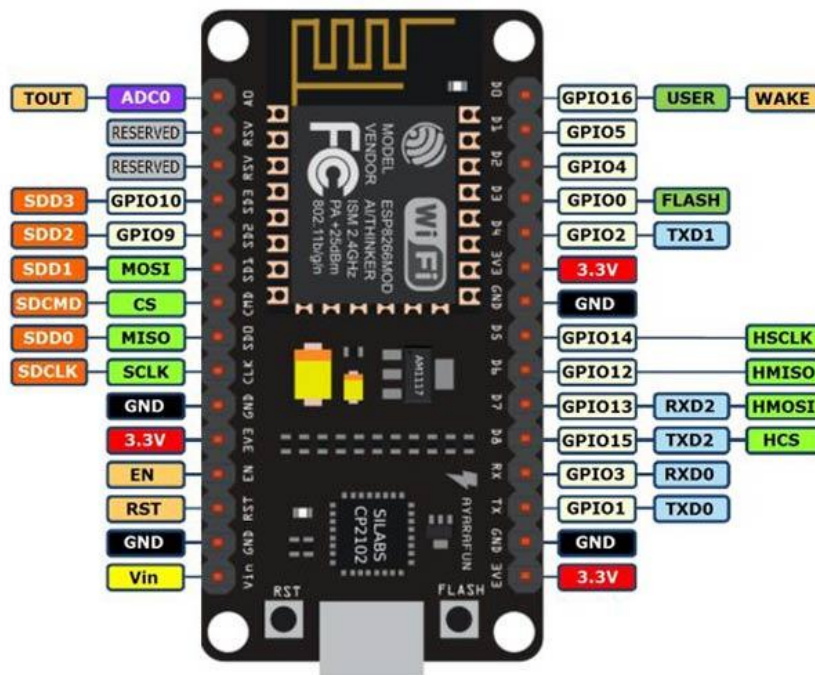


Fig. 3.5: Pin specifications

Using Arduino IDE

The most basic way to use the ESP8266 module is to use serial commands, as the chip is basically a WiFi/Serial transceiver. However, this is not convenient. What we recommend is using the very cool Arduino ESP8266 project, which is a modified version of the Arduino IDE that you need to install on your computer. This makes it very convenient to use the ESP8266 chip as we will be using the well-known Arduino IDE. Following the below step to install ESP8266 library to work in Arduino IDE environment.

Install the ESP8266 Board Package

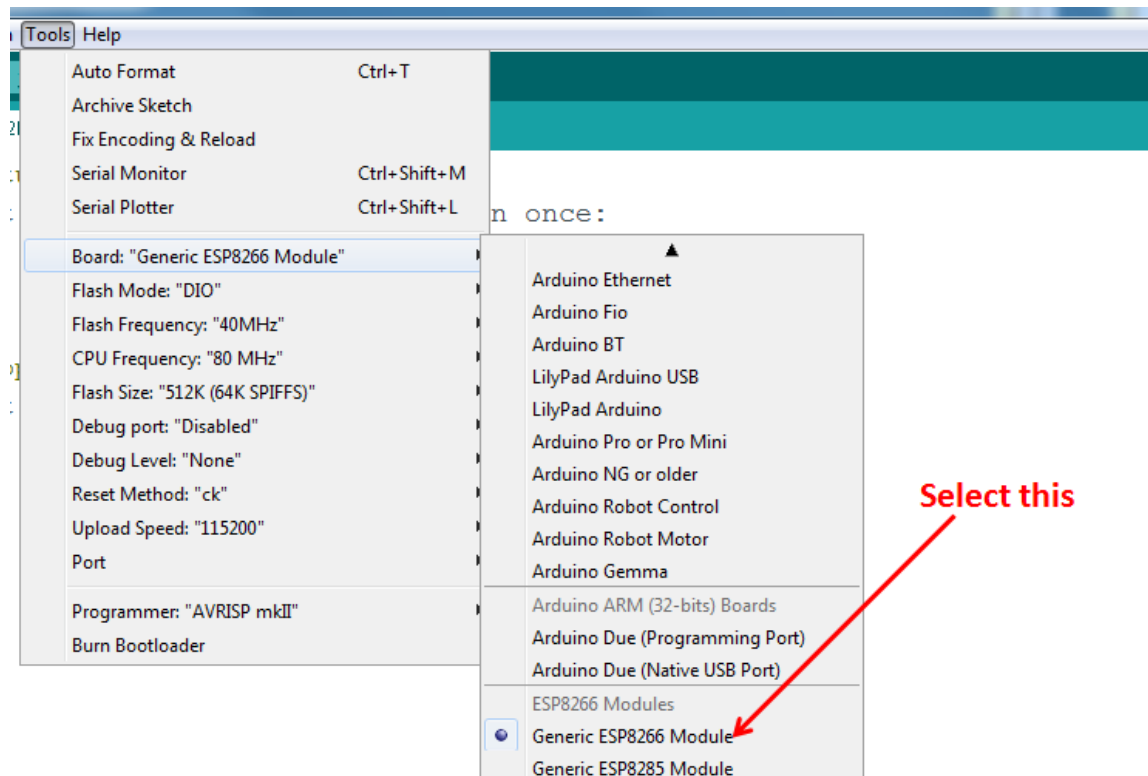
Enter http://arduino.esp8266.com/stable/package_esp8266com_index.json into Additional Board Manager URLs field in the Arduino v1.6.4+ preferences

Click 'File' -> 'Preferences' to access this panel. Next, use the Board manager to install the ESP8266 package.

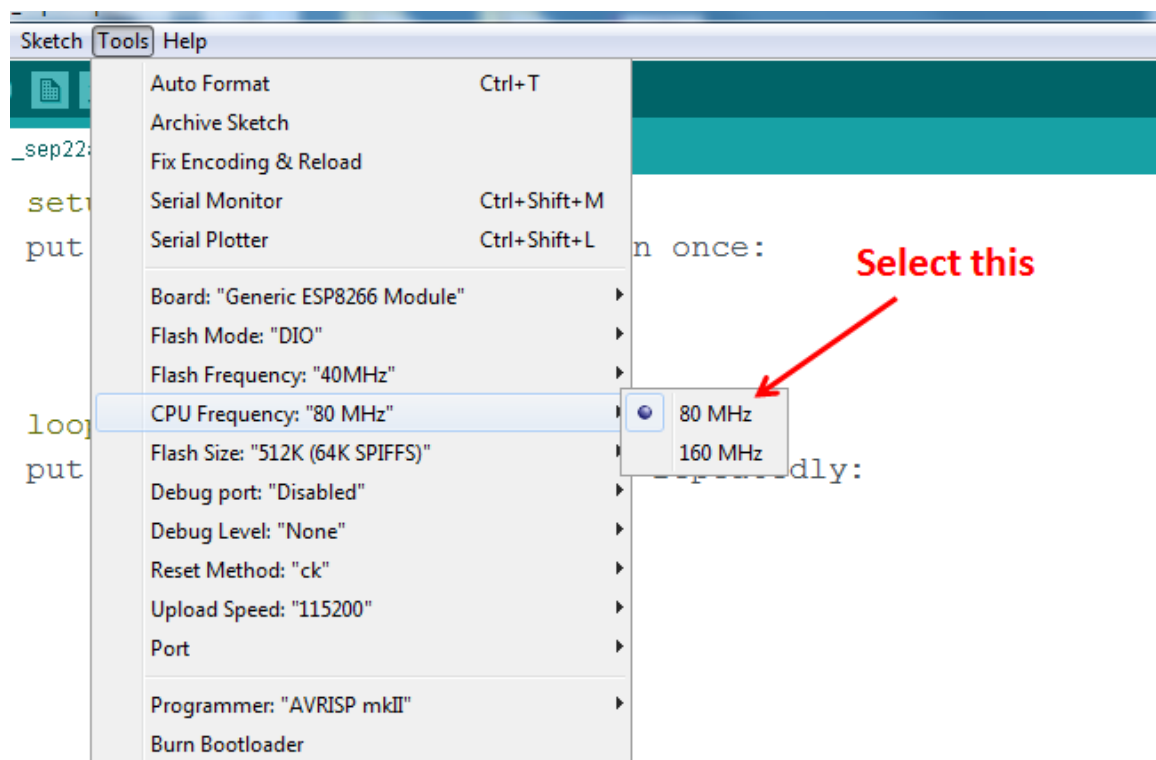
Click 'Tools' -> 'Board:' -> 'Board Manager...' to access this panel. Scroll down to ' esp8266 by ESP8266 Community ' and click "Install" button to install the ESP8266 library package. Once installation completed, close and re-open Arduino IDE for ESP8266 library to take effect.

Setup ESP8266 Support

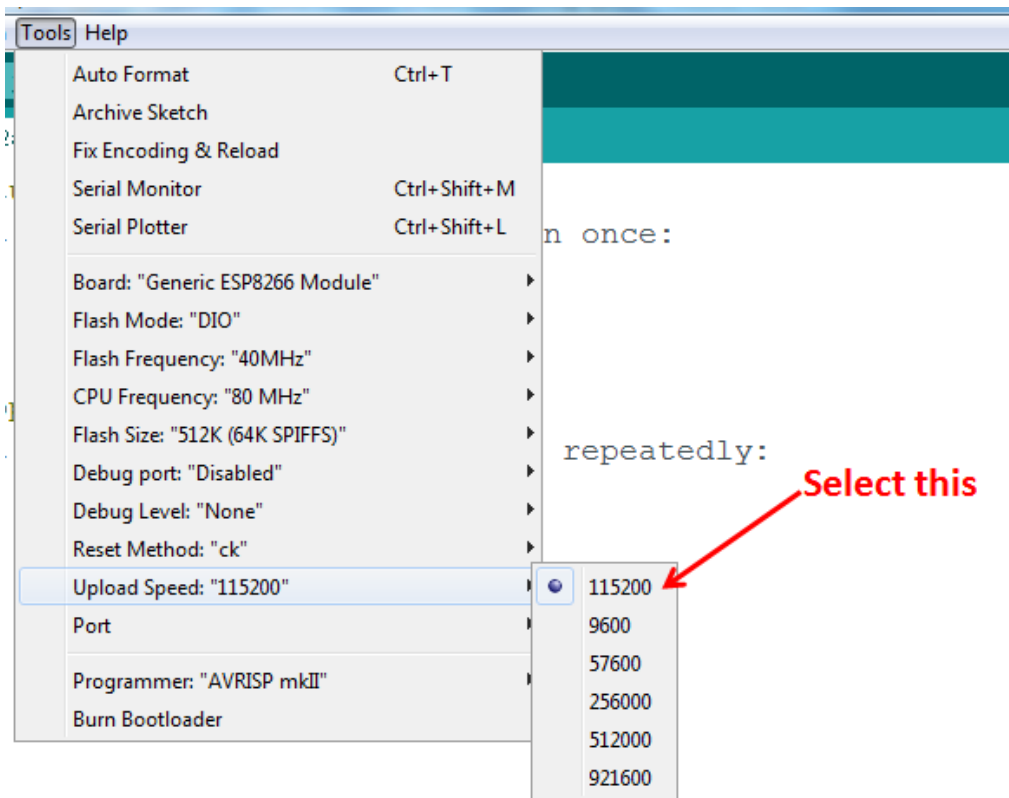
When you've restarted Arduino IDE, select 'Generic ESP8266 Module' from the 'Tools' -> 'Board:' dropdown menu.



Select 80 MHz as the CPU frequency (you can try 160 MHz overclock later)



Select '115200' baud upload speed is a good place to start - later on you can try higher speeds but 115200 is a good safe place to start.



Go to your Windows 'Device Manager' to find out which Com Port 'USB-Serial CH340' is assigned to. Select the matching COM/serial port for your CH340 USB-Serial interface.

Blink Test

We'll begin with the simple blink test. Enter this into the sketch window (and save since you'll have to). Connect a LED as shown in Figure 3.6.

Now you'll need to put the board into bootloader mode. You'll have to do this before each upload. There is no timeout for bootloader mode, so you don't have to rush!

- Hold down the 'Flash' button.
- While holding down 'Flash', press the 'RST' button.
- Release 'RST', then release 'Flash'

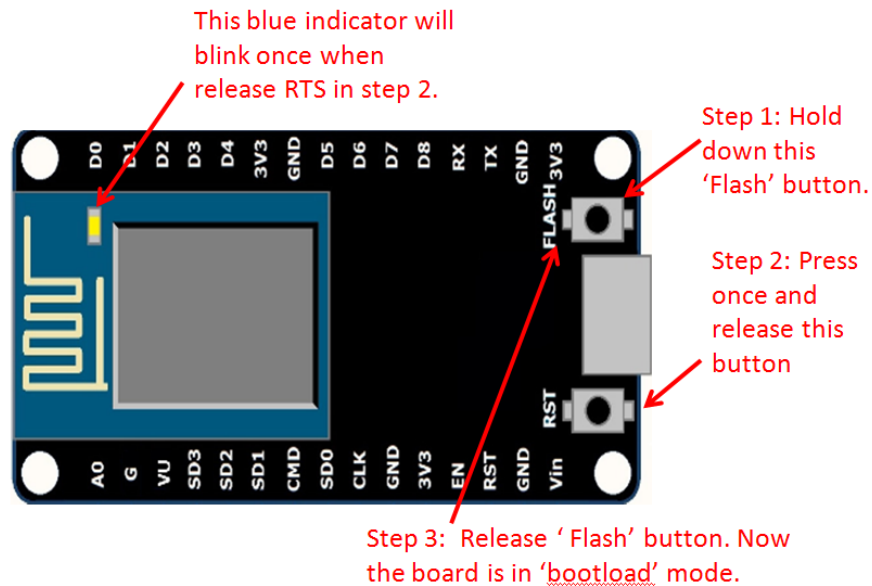


Fig. 3.6: Structure of blink test

Once the ESP board is in bootload mode, upload the sketch via the IDE

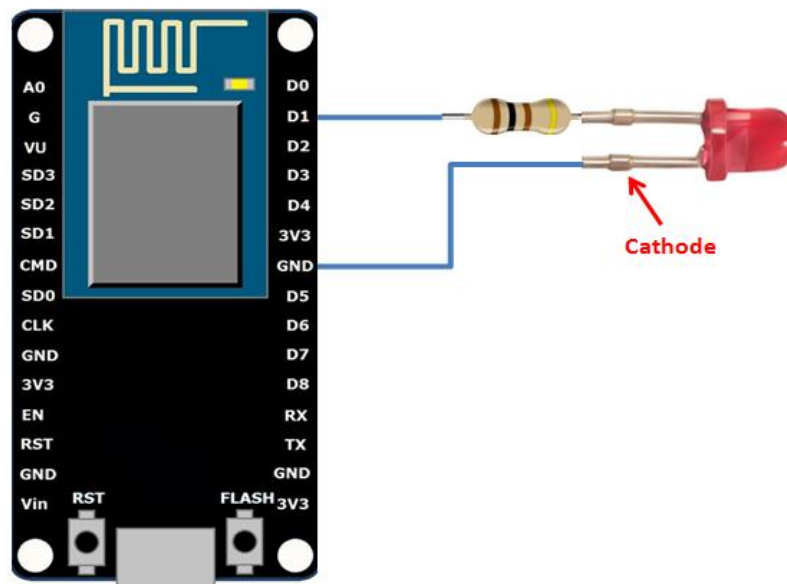


Fig. 3.7: Connection diagram for the blinking test

Flashing NodeMCU Firmware on the ESP8266 using Windows

NodeMCU is a firmware that allows you to program the ESP8266 modules with LUA script. And you'll find it very similar to the way you program your

Arduino. With just a few lines of code you can establish a WiFi connection, control the ESP8266 GPIOs, turning your ESP8266 into a web server and a lot more. In this tutorial we are going to use another ESP8266 module with pin header adapter board which is breadboard friendly.

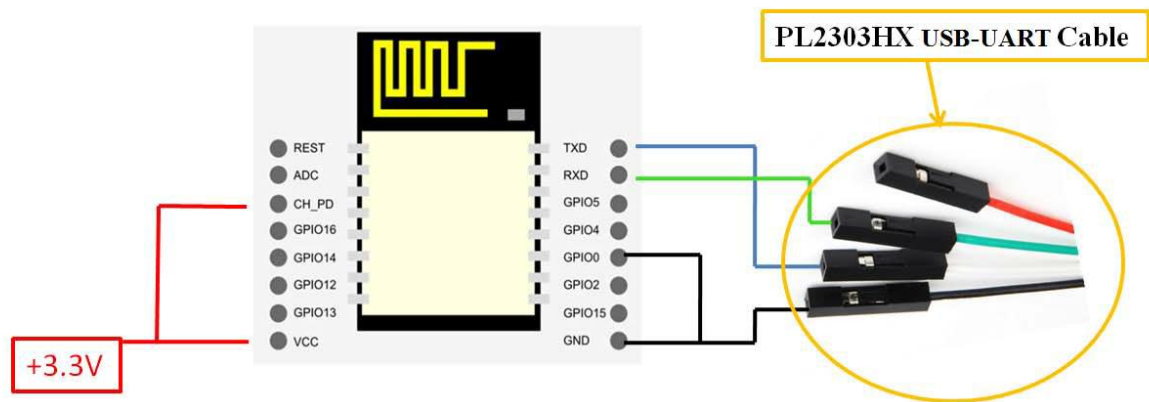


Fig. 3.8 Firmware of NodeMCU

3.5 TECHNICAL SPECIFICATIONS

About DHT11 sensor

The DHT11 sensor comes in a single row 4-pin package and operates from 3.5 to 5.5V power supply. It can measure temperature from 0-50 °C with an accuracy of $\pm 2^{\circ}\text{C}$ and relative humidity ranging from 20-95% with an accuracy of $\pm 5\%$. The sensor provides fully calibrated digital outputs for the two measurements. It has got its own proprietary 1-wire protocol, and therefore, the communication between the sensor and a microcontroller is not possible through a direct interface with any of its peripherals. The protocol must be implemented in the firmware of the MCU with precise timing required by the sensor. The following timing diagrams describe the data transfer protocol between a MCU and the DHT11 sensor. The MCU initiates data transmission by issuing a “Start” signal. The MCU pin must be configured as output for this purpose. The MCU first pulls the data line low for at least 18 ms and then pulls it high for next 20-40 μs before it releases it.

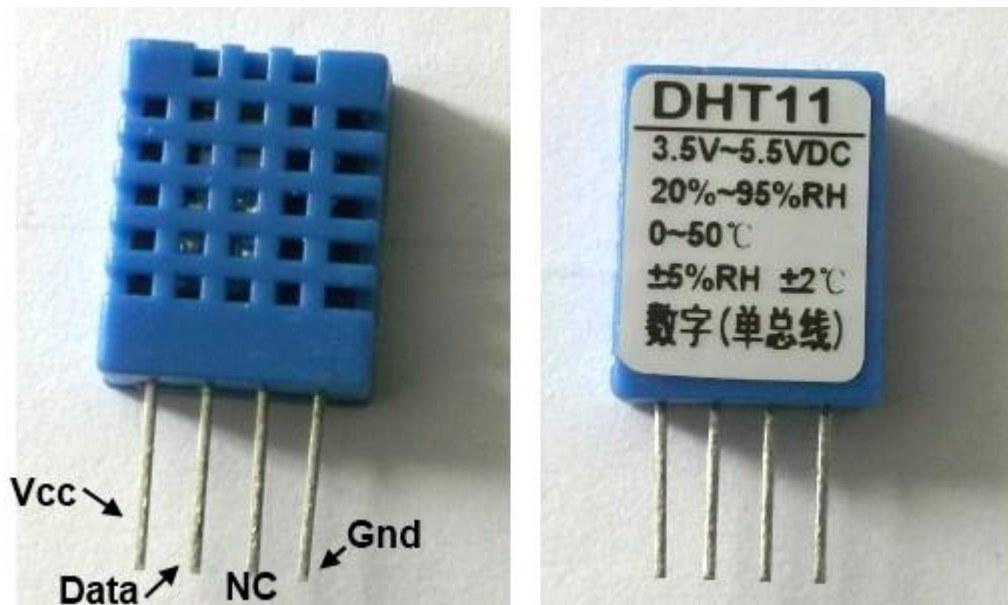


Fig. 3.9: DHT11 sensor

Next, the sensor responds to the MCU “Start” signal by pulling the line low for 80 μ s followed by a logic high signal that also lasts for 80 μ s. Remember that the MCU pin must be configured to input after finishing the “Start” signal. Once detecting the response signal from the sensor, the MCU should be ready to receive data from the sensor. The sensor then sends 40 bits (5 bytes) of data continuously in the data line. Note that while transmitting bytes, the sensor sends the most significant bit first.

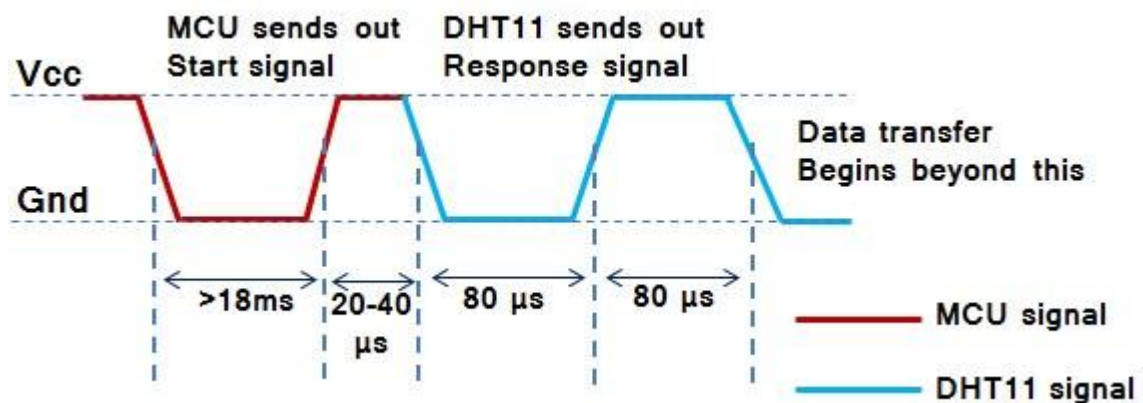


Fig. 3.10: Start" and "Response" signals

The 40-bit data from the sensor has the following structure.

Data (40-bit) = Integer Byte of RH + Decimal Byte of RH + Integer Byte of Temp. + Decimal Byte of Temp. + Checksum Byte

For DHT11 sensor, the decimal bytes of temperature and humidity measurements are always zero. Therefore, the first and third bytes of received data actually give the numeric values of the measured relative humidity (%) and temperature (°C). The last byte is the checksum byte which is used to make sure that the data transfer has happened without any error. If all the five bytes are transferred successfully then the checksum byte must be equal to the last 8 bits of the sum of the first four bytes, i.e.,

Checksum = Last 8 bits of (Integer Byte of RH + Decimal Byte of RH + Integer Byte of Temp. + Decimal Byte of Temp.)

Now lets talk about the most important thing, which is signalling for transmitting “0” and “1”. In order to send a bit of data, the sensor first pulls the line low for 50 ?s. Then it raises the line to high for 26-28 ?s if it has to send “0”, or for 70 ?s if the bit to be transmitted is “1”. So it is the width of the positive pulse that carries information about 1 and 0.

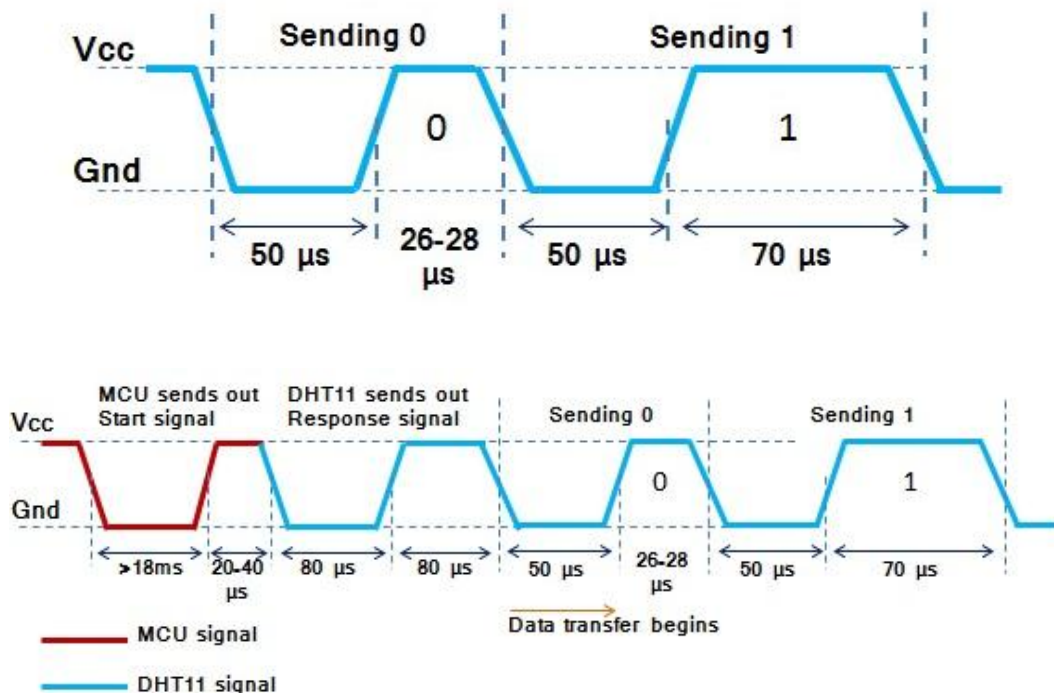


Fig. 3.11: Start, Response and Data signals in sequence

At the end of the last transmitted bit, the sensor pulls the data line low for 50 μ s and then releases it. The DHT11 sensor requires an external pull-up resistor to be connected between its Vcc and the data line so that under idle condition, the data line is always pulled high. After finishing the data transmission and releasing the data line, the DHT11 sensor goes to the low-power consumption mode until a new “Start” signal arrives from the MCU.

Respiratory sensor

Air and gas flows as well as line and therapy pressures have to be monitored and controlled in numerous medical instruments such as respiratory devices, aesthetic devices, sleep diagnosis devices, sleep apnea therapy devices (CPAP), spirometers and oxygen concentrators. High-quality sensors in respiratory devices measure minute flow rates around the zero point of the respiratory flow and also detect flow rates of several hundred l/min. Applications in aesthetic devices call for sensors with high resistance to anaesthetics. In controlled CPAP devices, pressure sensors continuously monitor the therapy pressure, thereby improving the comfort and quality of the treatment. Spirometers utilize special pneumotachographs that measure the respiratory flow using efficient differential pressure sensors.

First Sensor develops and manufactures highly reliable sensors and customized sensor systems as a strategic partner to medical product manufacturers in the area of breathing and respiration. Our complete advisory, development, production and service processes comply with the high requirements for medical products according to EN ISO 13485

The respiration sensor is fixed to a long hook and loop strap that is placed around the chest or abdomen. For most applications, placing one sensor around the abdomen is necessary. Optionally, you can place a second respiration sensor around the chest. Using two sensors is helpful for abdominal breathing exercises.

Unravel the strap and attach it around the abdomen (or torso) so that the sensor is in the front. Ask the client to breathe out as fully as possible and attach

the sensor so there is minimal tension. The fit should be snug enough that the strap stays fixed when the subject is relaxed.

There should also be enough slack in the rubber strap of the sensor so that expansion of the abdomen causes this rubber strap to expand without being overextended.

The respiration signal is a relative measure of abdomen expansion.

The Respiration Sensor is a sensitive girth sensor using an easy fitting high durability latex rubber band fixed with self-adhering belt. It detects chest or abdominal expansion/contraction and shows the respiration waveform and amplitude. It can be worn over clothing. Respiration can be measured by means of an inductive or resistive respiration belt, but also by means of measuring the EMG of the diaphragm.

For several medical purposes, the respiration of a patient needs to be measured. Sometimes we want to have an impression about the pulmonary function, sometimes we just want to detect and calculate the amount of apneas.

We have both inductive and resistive respiration belts available for the measurement of respiration. Next to that it is also possible to measure the nasal airflow by means of our differential pressure sensors.

The Future Internet goal is to provide an infrastructure to have an immediate access to information about the physical world and its objects. Physical objects can be applicable to different application domains, such as e-health, warehouse management, etc. Each application domain may have different types of physical devices. Each physical device can have its own specifications, which is required to use in order to interact with it. To achieve the future Internet goal, a layered vision is required that can facilitate data access. Internet of Things (IoT) is a vision that aims to integrate the virtual world of information to the real world of devices through a layered architecture.

Heartbeat Sensor

A person's heartbeat is the sound of the valves in his/her's heart contracting or expanding as they force blood from one region to another. The number of times the heart beats per minute (BPM), is the heart beat rate and the beat of the heart that can be felt in any artery that lies close to the skin is the pulse.

Two Ways to Measure a Heartbeat

Manual Way: Heart beat can be checked manually by checking one's pulses at two locations- wrist (the radial pulse) and the neck (carotid pulse). The procedure is to place the two fingers (index and middle finger) on the wrist (or neck below the windpipe) and count the number of pulses for 30 seconds and then multiplying that number by 2 to get the heart beat rate. However pressure should be applied minimum and also fingers should be moved up and down till the pulse is felt.

Using a sensor: Heart Beat can be measured based on optical power variation as light is scattered or absorbed during its path through the blood as the heart beat changes.

Principle of Heartbeat Sensor

The heartbeat sensor is based on the principle of photoplethysmography. It measures the change in volume of blood through any organ of the body which causes a change in the light intensity through that organ (a vascular region). In case of applications where heart pulse rate is to be monitored, the timing of the pulses is more important. The flow of blood volume is decided by the rate of heart pulses and since light is absorbed by blood, the signal pulses are equivalent to the heart beat pulses.

There are two types of photoplethysmography:

Transmission: Light emitted from the light emitting device is transmitted through any vascular region of the body like earlobe and received by the detector.

Reflection: Light emitted from the light emitting device is reflected by the regions.

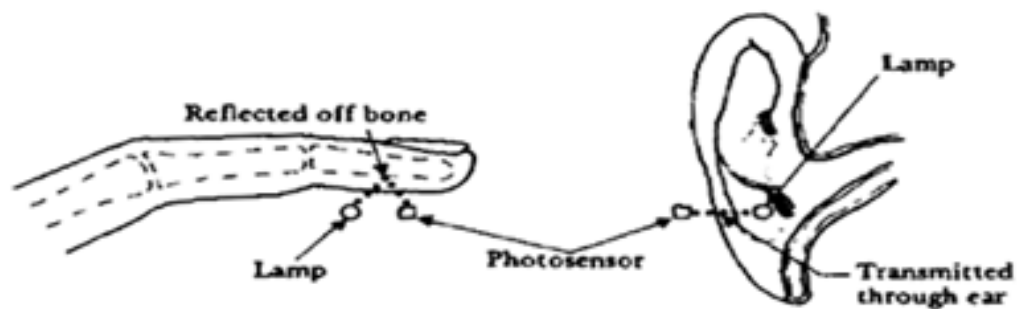
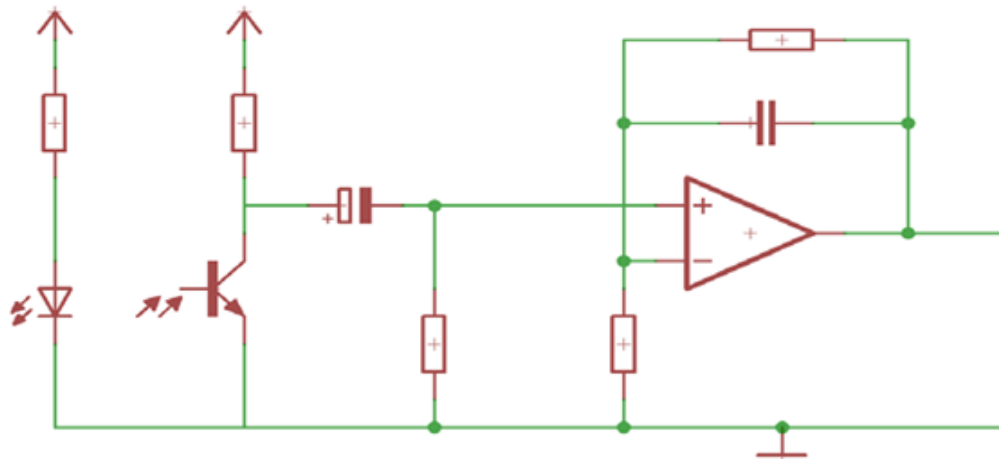


Fig. 3.12: Heartbeat Sensor methodology

Working of a Heartbeat Sensor

The basic heartbeat sensor consists of a light emitting diode and a detector like a light detecting resistor or a photodiode. The heart beat pulses causes a variation in the flow of blood to different regions of the body. When a tissue is illuminated with the light source, i.e. light emitted by the led, it either reflects (a finger tissue) or transmits the light (earlobe). Some of the light is absorbed by the blood and the transmitted or the reflected light is received by the light detector. The amount of light absorbed depends on the blood volume in that tissue. The detector output is in form of electrical signal and is proportional to the heart beat rate.

This signal is actually a DC signal relating to the tissues and the blood volume and the AC component synchronous with the heart beat and caused by pulsatile changes in arterial blood volume is superimposed on the DC signal. Thus the major requirement is to isolate that AC component as it is of prime importance.



To achieve the task of getting the AC signal, the output from the detector is first filtered using a 2 stage HP-LP circuit and is then converted to digital pulses using a comparator circuit or using simple ADC. The digital pulses are given to a microcontroller for calculating the heart beat rate,

$$\text{BPM(Beats per minute)} = 60 \cdot f$$

Where f is the pulse frequency

Practical Heartbeat Sensor

Practical heartbeat Sensor examples are Heart Rate Sensor (Product No PC-3147). It consists of an infrared led and an ldr embedded onto a clip like structure. The clip is attached to the organ (earlobe or the finger) with the detector part on the flesh.

Pressure Sensor

Pressure sensor is a device or instrument which is able to measure the pressure in gases or liquids. A pressure sensor consists of a pressure-sensitive element which can determine the pressure being applied and components to convert the information into an output signal. A pressure sensor will generate a signal based on the measurement of the pressure applied.

The demand for pressure measuring instruments increased during the steam age. When pressure sensing technologies were first manufactured they were

mechanical and used Bourdon tube gauges to move a needle and give a visual indication of pressure. Nowadays we measure pressure electronically using pressure transducers and pressure switches.

Static Pressure

Pressure can be defined as force per unit area that a fluid exerts on its surroundings. The basic physics of static Pressure (P), is calculated as force (F) divided by area (A).

$$P=F/A$$

The Force can be generated by liquids, gases, vapours or solid bodies.

The most commonly used pressure units are;

$$\text{Pa} - [\text{Pascal}] \text{ in } 1 \text{ Pa} = 1(\text{N/m}^2)$$



Fig. 3.13: Pressure sensor

Input and Output

The Arduino microcontroller is an easy to use yet powerful single board computer that has gained considerable traction in the hobby and professional market. The Arduino is open-source, which means hardware is reasonably priced and development software is free. This guide is for students in ME 2011, or students anywhere who are confronting the Arduino for the first time. For advanced Arduino users, prowl the web; there are lots of resources.

This guide covers the Arduino Uno board (Spark fun DEV-09950, \$29.95), a good choice for students and educators. With the Arduino board, you can write programs and create interface circuits to read switches and other sensors, and to control motors and lights with very little effort. Many of the pictures and drawings in this guide were taken from the documentation on the this is what the Arduino board looks like.



Fig. 3.14: UNO board

The Duemilanove board features an Atmel ATmega328 microcontroller operating at 5 V with 2 Kb of RAM, 32 Kb of flash memory for storing programs and 1 Kb of EEPROM for storing parameters. The clock speed is 16 MHz, which translates to about executing about 300,000 lines of C source code per second. The board has 14 digital I/O pins and 6 analog input pins. There is a USB connector for talking to the host computer and a DC power jack for connecting an external 6-20 V power source, for example a 9 V battery, when running a program while not connected to the host computer. Headers are provided for interfacing to the I/O pins using 22 g solid wire or header connectors.

The Arduino programming language is a simplified version of C/C++. If you know C, programming the Arduino will be familiar. If you do not know C, no need to worry as only a few commands are needed to perform useful functions.

An important feature of the Arduino is that you can create a control program on the host PC, download it to the Arduino and it will run automatically. Remove the USB cable connection to the PC, and the program will still run from the top each time you push the reset button. Remove the battery and put the Arduino

board in a closet for six months. When you reconnect the battery, the last program you stored will run. This means that you connect the board to the host PC to develop and debug your program, but once that is done, you no longer need the PC to run the program.

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

Summary

Microcontroller ATmega328

Operating Voltage 5V

Input Voltage (recommended) 7-12V

Input Voltage (limits) 6-20V

Digital I/O Pins 14 (of which 6 provide PWM output)

Analog Input Pins 6

DC Current per I/O Pin 40 mA

DC Current for 3.3V Pin 50 mA

Flash Memory 32 KB (ATmega328) of which 0.5 KB used by bootloader

SRAM 2 KB (ATmega328)

EEPROM 1 KB (ATmega328)

Clock Speed 16 MHz

Pin Configuration

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall- wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** this pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.
- **IOREF.** This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield

can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM.

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using `pinMode()`, `digital Write( )`, and `digital Read( )` functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kohms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and Transmit (TX) TTL serial data

These pins are connected to the corresponding pins of the ATmega8U2 USB-to- TTL Serial chip.

- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the `attach Interrupt ( )` function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the `analog Write ( )` function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication using the SPI library.
- **LED: 13.** There is a built- in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the analog Reference () function. Additionally, some pins have specialized functionality:

- **TWI: A4 or SDA pin and A5 or SCL pin.** Support TWI communication using the Wire library.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analog Reference ().
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

3.6 COMMUNICATION AND TESTING

Microcontrollers depend on a host computer for developing and compiling programs. The software used on the host computer is known as an integrated development environment, or IDE. For the Arduino, the development environment is based on the open source Processing platform (www.processing.org) which is described by its creators as a “programming language and environment for people who want to program images, animation, and interactions.” The Arduino programming language leverages an open source project known as Wiring (wiring.org.co). The Arduino language is based on good old- fashioned C. If you are unfamiliar with this language, don’t worry; it’s not hard to learn, and the Arduino IDE provides some feedback when you make mistakes in your programs.

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware

uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a `inf` file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A Software Serial library allows for serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

As you go through the list of programming statements available in the Arduino IDE (choose Help->Reference), you might think there isn't much power for doing things like running servos, operating stepper motors, reading potentiometers, or displaying text on an LCD. Like most any language based on C, the Arduino supports the notion of "libraries" code.

Repositories that extend core programming functionality. Libraries let you re-use code without having to physically copy and paste it into all your programs. The standard Arduino software installation comes with several libraries you may use, and you can download others from the Arduino support pages and from third-party websites that publish Arduino library code. A good example of a library you'll use with the Robot and likely many other robot projects is Servo. This library allows you to connect one or more hobby R/C servos to the Arduino's digital I/O pins. The Servo library comes with the standard Arduino installation package Library->Servo. This adds the line

```
#include <Servo.h>
```

Which tells the Arduino IDE that you wish to include the Servo library in your sketch. With the functionality of the library now available to you, you can use its various functions to control one or more servos. For example, you can use the

write function to rotate a servo to a specific position, from 0 to 180 degrees. The following code

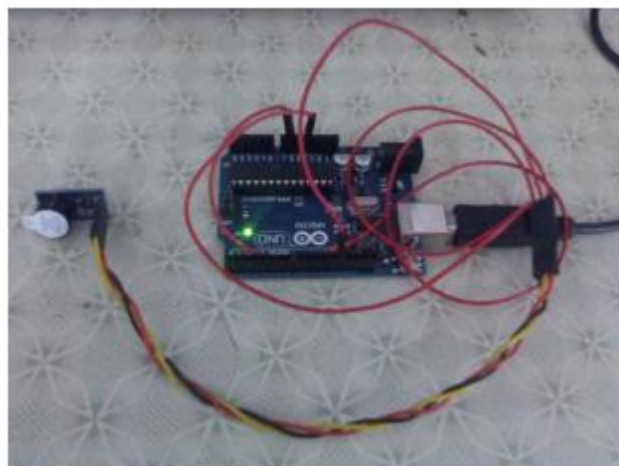
```
myServo.write(90);
```

Moves a servo to its midpoint, or 90 degree position. Structurally, Arduino sketches are very straightforward and are pretty easy to read and understand. The Arduino program contains two main parts: `setup ()` and `loop ()`. These are programming functions that do what their names suggest: `setup ()` sets up the Arduino hardware, such as specifying which I/O lines you plan to use, and whether

They are inputs or outputs. The `loop ()` function is repeated endlessly when the Arduino is operating.

Arduino IDE (Integrated development environment) is used to write the program and dump into the Arduino board

The sample sketch above is a blink which is also applicable for LEDs. The output is the turning on and off of the buzzer every other second. The picture below shows the setup of your module and Arduino:



CHAPTER 4

NEURAL NETWORKS FOR SECURED HEALTHCARE SYSTEM

4.1 BPN PARAMETERS

A Back Propagation Network (BPN) is multilayered neural network architecture that learns from a predefined set of input output example pairs called exemplars. The inputs are propagated from the first layer to the upper layers through the hidden layer until an output is generated. The obtained output is then compared with the desired target. The error is calculated and propagated backwards. For this reason the method is often called the back-propagation learning rule. Based on their contribution towards the original output, each unit in the intermediate layer receives a portion of the total error signal. This process repeats for all layers till each unit receives an error signal relative to the total error. In accordance to the error signal received, connection weights are then updated by each unit. The significance of this process is that as the network trains, the nodes in the intermediate layers organizes themselves such that all the units learn to recognize different features of the total input space. After training, when presented with an arbitrary input pattern that is noisy or incomplete, the units in the hidden layers of the network will respond with an active output if the new input contains a pattern that resembles the feature of the individual units learned to recognize during training. The size of the input layer and the output layer is usually determined by the application. Determining the number of units in the hidden layer is not clearly defined. The size of the hidden layer has to be a compromise between accuracy of the result and the required storage space. Weights should be assigned to small random values and the bias values must also be small (Haykin 1998, Yegnanarayana 2006, Hosseingholizadeh and Abhari 2009). A simple BPN architecture is shown in Fig 4.1.

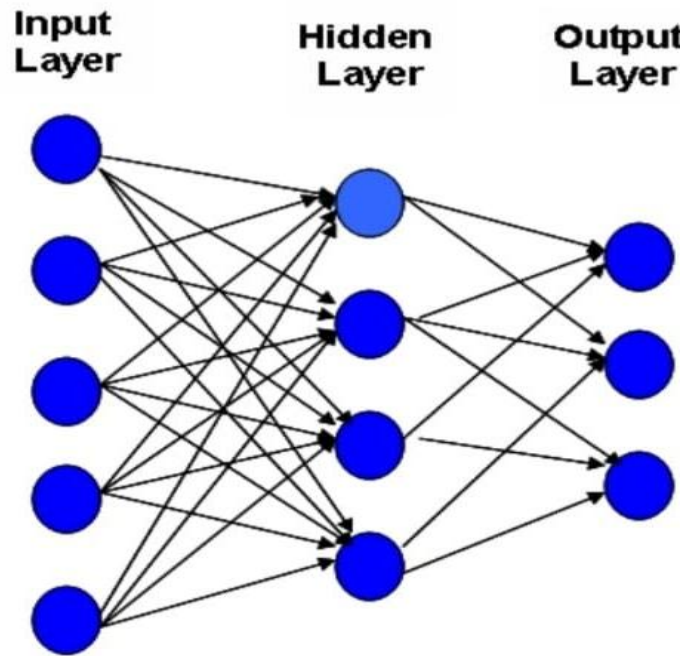


Fig. 4.1: BPN architecture

Parameters	Description
Number of layers	4 (10, 10, 5,1)
Activation function (hidden layer)	tansigmoidal
Activation function (output layer)	purelinear
Learning parameter	0.6
Momentum parameter	0.9

4.2 NEURAL NETWORK TRAINING (nntraintool)

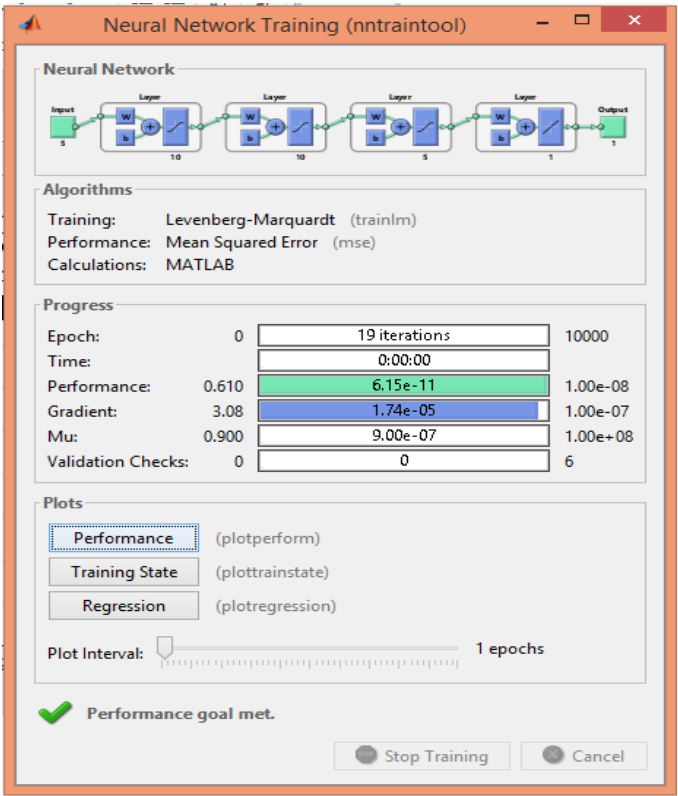


Fig. 4.2: Neural Network Training (nntraintool)

Mean square error = 0.1600

4.3 COMPARISON OF ACTUAL AND DESIRED OUTPUT

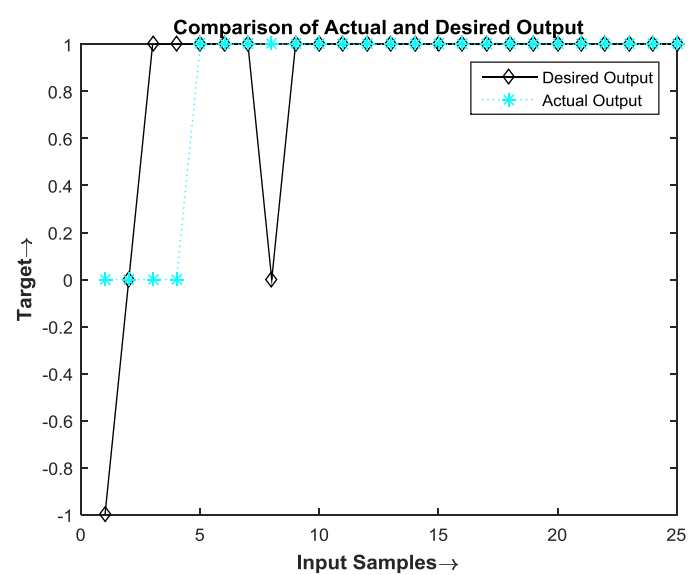


Fig. 4.3: Comparison of Actual and Design Output

4.4 CODE

```
close all;

clear all;

clc;

pressure=[101325 101561 101088 101099 101088 101156 101123 101256
101325 101060 101061 101045 101089 101100 101265 101000 101023
101028 101066 101189 101147 101159 101145 101023 101178 101796
101456 101111 101222 101333 101136 101169 101258 101147 101369
101258 101789 101236 101023 101036 101025 101078 101068 101006
101055 101069 101044 101006 101023 101089];

Hum=[49 52 47 66 47 56 89 67 30 33 33 56 45 56 47 32 30 50 55 90
91 21 22 25 23 56 54 55 45 47 48 41 42 43 46 38 39 37 34 36 45 56
78 34 45 30 29 31 50 49];

temp=[37 45 44 38 39 31 33 33 35 30 31 32 34 35 36 56 34 35 37 34
35 36 44 35 41 40 38 39 37 36 35 36 37 38 39 33 41 37 38 39 33 34
34 35 36 37 34 35 36 35];

pulse=[101 100 109 100 110 109 100 56 72 77 67 60 56 77 89 91 99 68 99
106 108 103 78 89 88 86 56 58 59 60 100 67 69 70 71 72 75 77 88 99
88 78 89 67 87 89 78 99 100 77];

resp=[1 1 1 1 0 1 1 1 0 1 1 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 1 1 0 1 0 1 1 0 0];

temp=temp./100;

Hum=Hum./100;

pressure=pressure./1000000;

pulse=pulse./1000;

resp=resp./1;
```

```

p=[temp(1:2:end);Hum(1:2:end);pressure(1:2:end);pulse(1:2:end);resp(1:2:end)];
x=[zeros(1,8) ones(1,42)];
t=x(1:2:end);
% t=t./10;
% t=t';
% net=newff(minmax(p),[10,10,5,1],{'tansig','tansig','tansig','purelin'});
net=newff(minmax(p),[10,10,5,1],{'tansig','tansig','tansig','purelin'});
net.trainparam.show=5;
net.trainparam.epochs=10000;
net.trainparam.lr=0.6;
net.trainparam.mu=0.9;
net.trainparam.goal=.00000001;
net.trainParam.mu_max = 1e+8;
net=train(net,p,t);
a=sim(net,p);
p1=[temp(2:2:end);Hum(2:2:end);pressure(2:2:end);pulse(2:2:end);resp(2:2:end)];
% p1=[Temp(2:2:end);Hum(2:2:end);press(2:2:end);rain(2:2:end);light(2:2:end)];
t1=x(2:2:end);
a1=round(sim(net,p1));
% a1=a1*10;
% t1=t1*10;
% figure,plot(1:length(a),a,'k+:',1:length(a),t,'c*');
figure,plot(1:length(a1),a1,'kd-',1:length(a1),t1,'c*');
% grid on
title('\bfComparison of Actual and Desired Output')
xlabel('\bfInput Samples\rightarrow')
ylabel('\bfTarget\rightarrow')

```

```
h = legend('Desired Output','Actual Output',2);  
sum=0;  
for i=1:length(a1)  
    sum=sum+(a1(i)-t1(i))^2;  
end  
mss=sum/length(a1);
```

CHAPTER 5

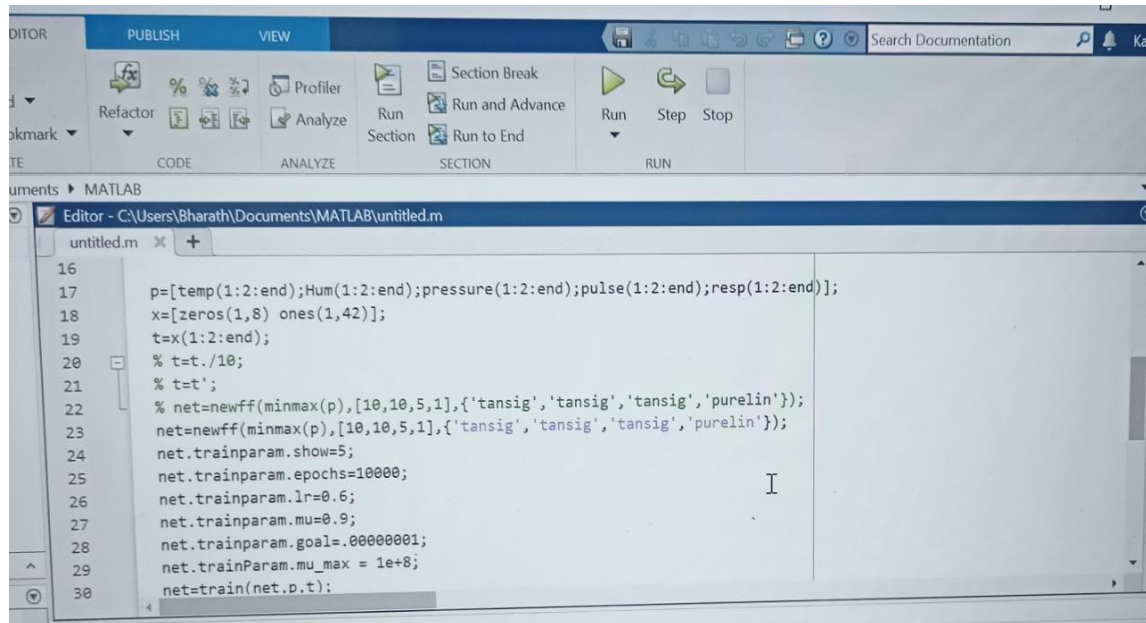
CONCLUSION

5.1 SUMMARY

We had successfully completed our project with a good result. As we expected we could predict the basic health condition of a person and could recognize the nature conditions. Any abnormal normal values in our environment or body conditions we could predict it in future and analyze a quick solution for it.

We have presented and proved the prototype for an automatic system that guarantees a constant monitoring of various health parameters and prediction of any kind of disease or disorder that prevents the patient from the pain of paying frequent visits to the hospitals. The proposed system can be set-up in the hospitals and massive amount of data can be obtained and stored in the online database. Even the results can be made to be accessed from mobile through an application. The system can be further improved further by adding artificial intelligence system components to facilitate the doctors and the patients. The data, consisting medical history of many patients' parameters and corresponding results, can be explored using data mining, in search of consistent patterns and systematic relationships in the disease. For instance, if a patient's health parameters are changing in the same pattern as those of a previous patient in the database, the consequences can also be estimated. If the similar patterns are found repeatedly, it would be easier for the doctors and medical researchers to find a remedy for the problem.

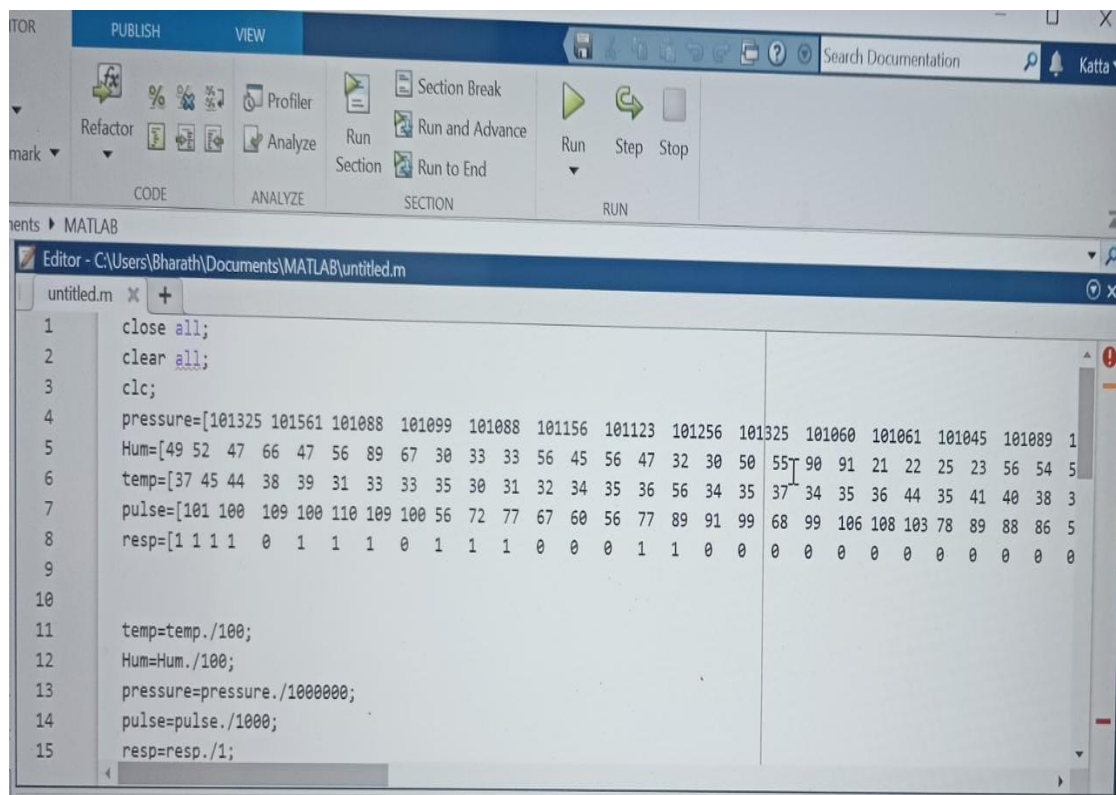
5.2 OUTPUT



```

16 p=[temp(1:2:end);Hum(1:2:end);pressure(1:2:end);pulse(1:2:end);resp(1:2:end)];
17 x=[zeros(1,8) ones(1,42)];
18 t=x(1:2:end);
19 % t=t./10;
20 % t=t';
21 % net=newff(minmax(p),[10,10,5,1],{'tansig','tansig','tansig','purelin'});
22 net=newff(minmax(p),[10,10,5,1],{'tansig','tansig','tansig','purelin'});
23 net.trainparam.show=5;
24 net.trainparam.epochs=10000;
25 net.trainparam.lr=0.6;
26 net.trainparam.mu=0.9;
27 net.trainparam.goal=.00000001;
28 net.trainParam.mu_max = 1e+8;
29 net=train(net.p,t);
30

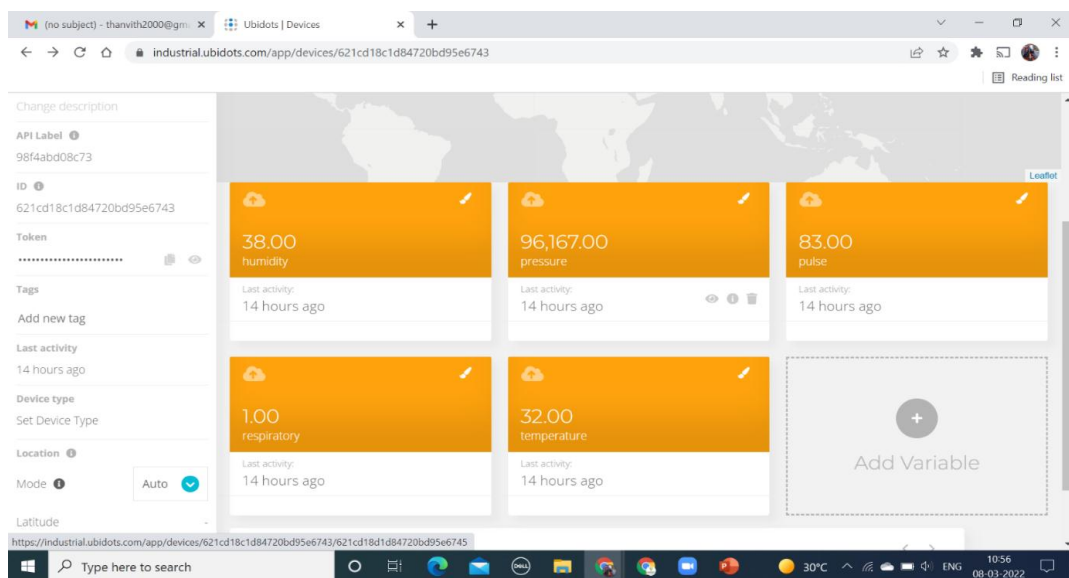
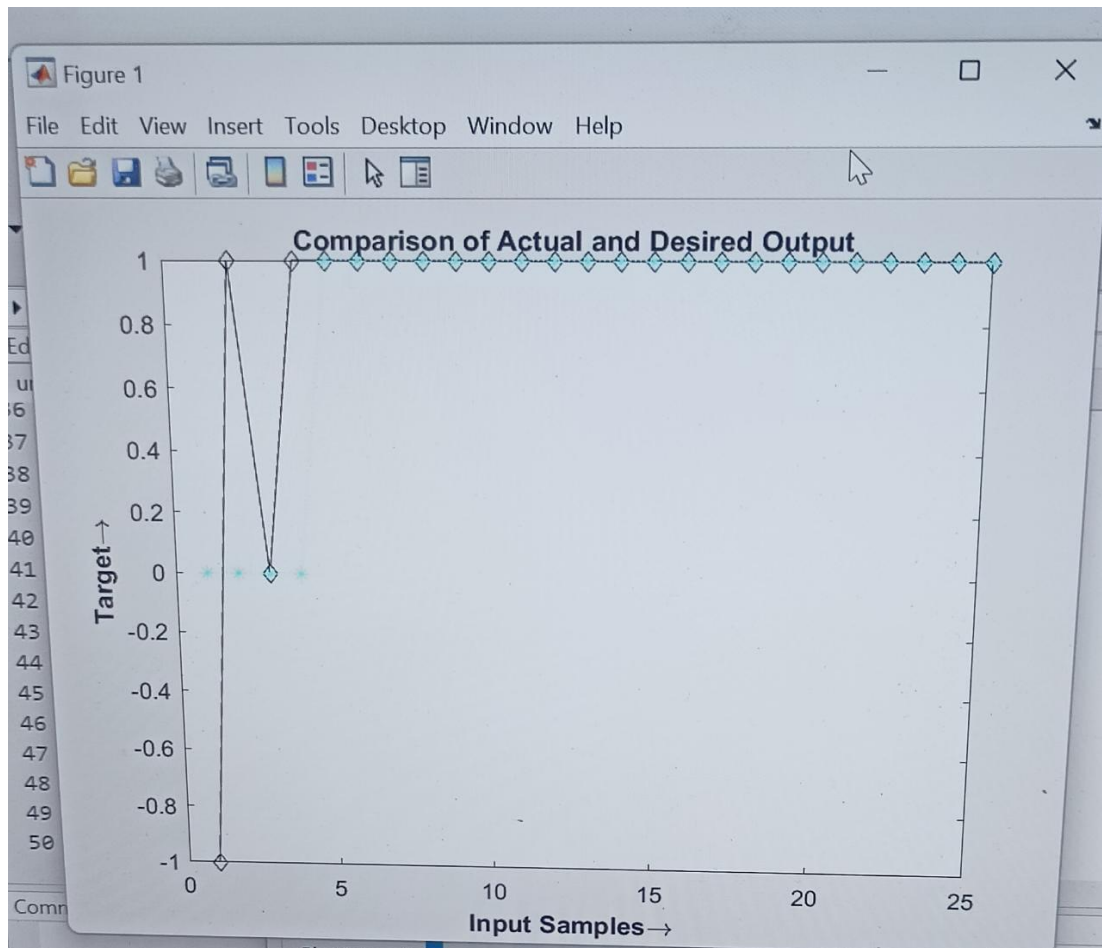
```

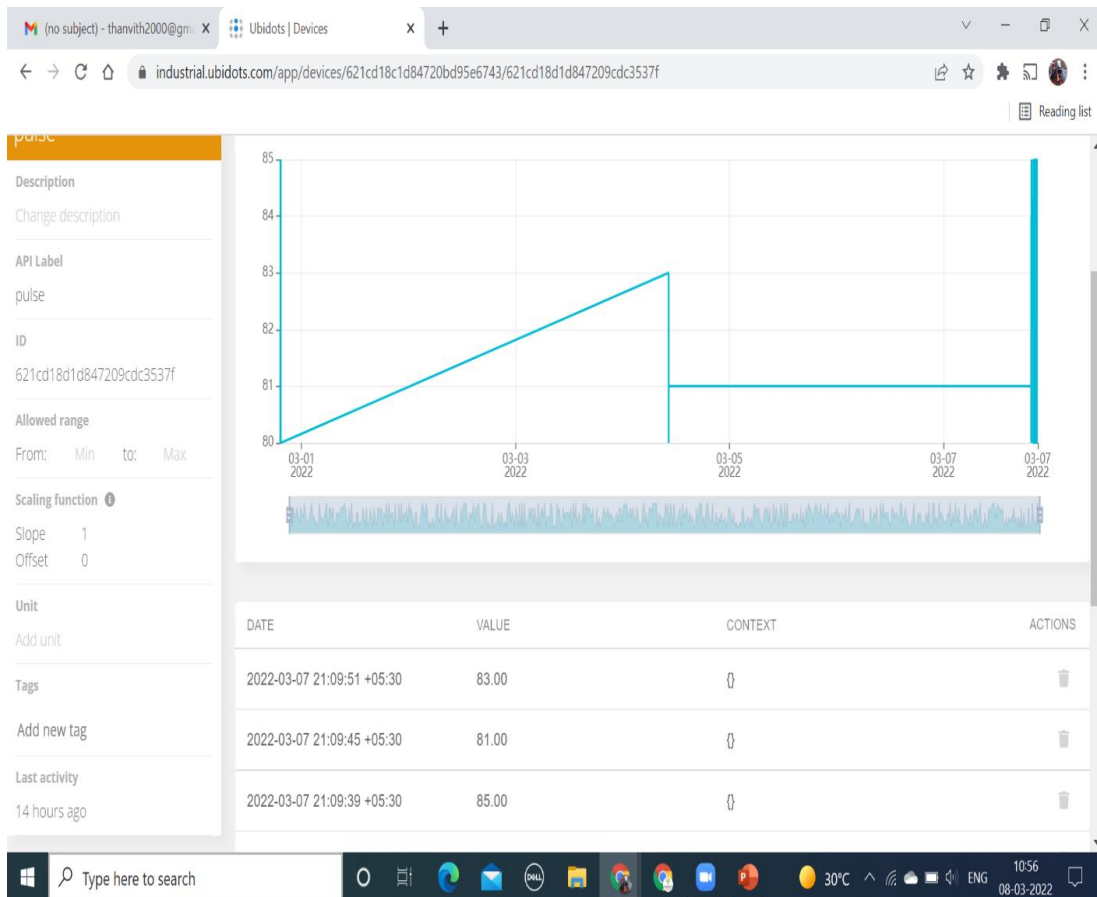


```

1 close all;
2 clear all;
3 clc;
4 pressure=[101325 101561 101088 101099 101088 101156 101123 101256 101325 101060 101061 101045 101089 1
5 Hum=[49 52 47 66 47 56 89 67 30 33 33 56 45 56 47 32 30 50 55 90 91 21 22 25 23 56 54 5
6 temp=[37 45 44 38 39 31 33 33 35 30 31 32 34 35 36 56 34 35 37 34 35 36 44 35 41 40 38 3
7 pulse=[101 100 109 100 110 109 100 56 72 77 67 60 56 77 89 91 99 68 99 106 108 103 78 89 88 86 5
8 resp=[1 1 1 1 0 1 1 1 0 1 1 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0
9
10
11 temp=temp./100;
12 Hum=Hum./100;
13 pressure=pressure./1000000;
14 pulse=pulse./1000;
15 resp=resp./1;

```





Ubidots | Devices

industrial.ubidots.com/app/devices/621cd18c1d84720bd95e6743/621cd18d1d847209cdc3537f

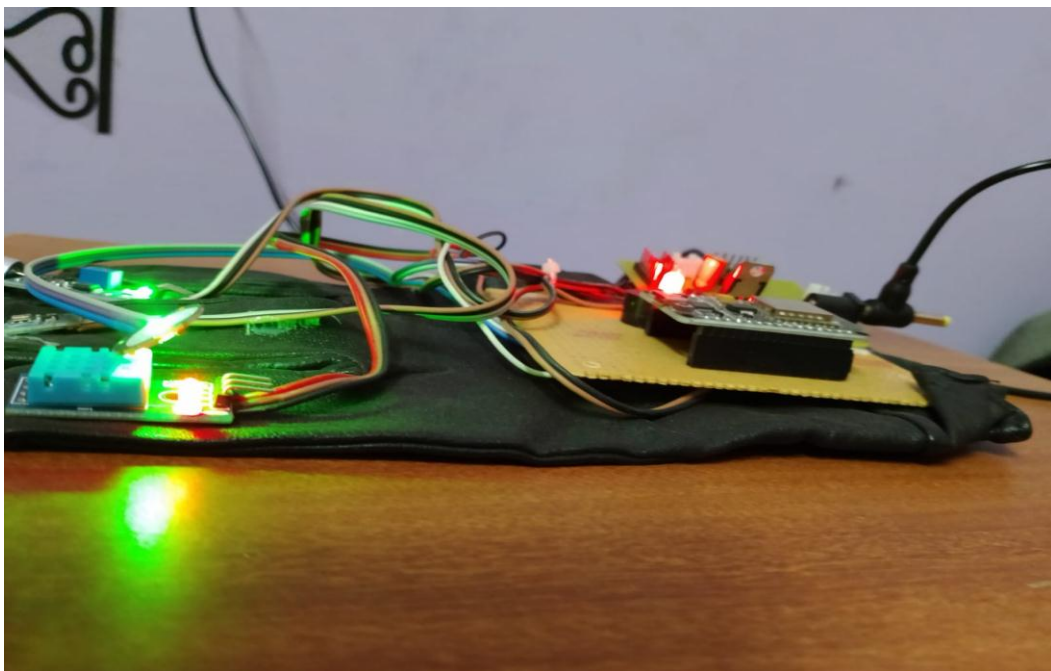
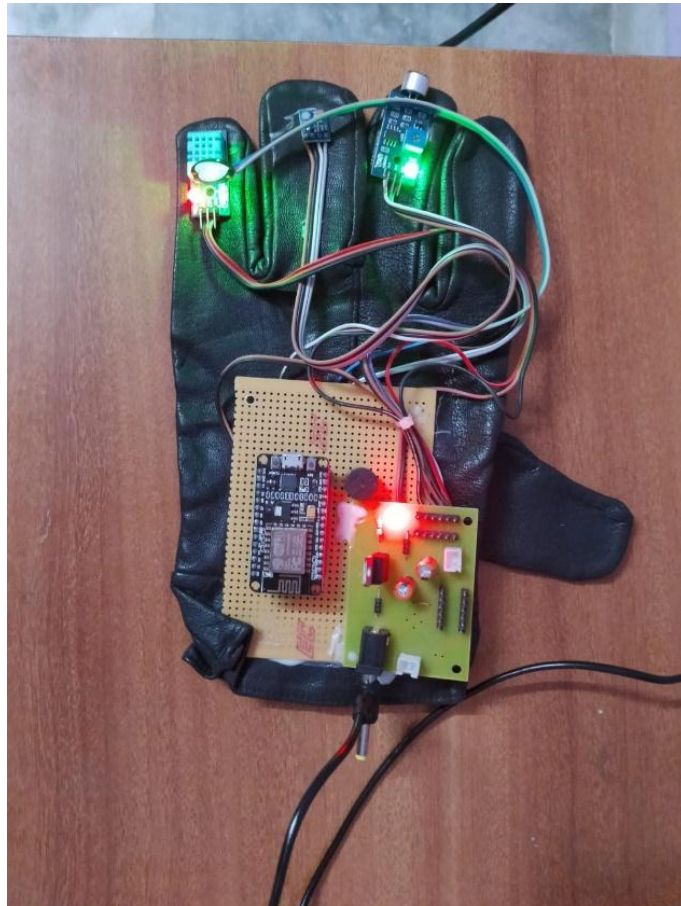
Reading list

DATE	VALUE	CONTEXT	ACTIONS
2022-03-07 21:09:51 +05:30	95166.00	{}	
2022-03-07 21:09:45 +05:30	95170.00	{}	
2022-03-07 21:09:39 +05:30	95165.00	{}	
2022-03-07 21:09:33 +05:30	95166.00	{}	
2022-03-07 21:09:28 +05:30	95165.00	{}	
2022-03-07 21:09:21 +05:30	95167.00	{}	
2022-03-07 21:09:15 +05:30	95168.00	{}	
2022-03-07 21:09:09 +05:30	95167.00	{}	
2022-03-07 21:09:02 +05:30	95170.00	{}	
2022-03-07 21:08:57 +05:30	95167.00	{}	

ROWS PER PAGE 10

Type here to search

10:57 08-03-2022



REFERENCES

- [1] A Mobility-aware Human-centric Cyber-physical System for Efficient and Secure Smart Healthcare, December 2021.
- [2] Boyle, T., and Aygun, R., Kennesaw state university hpc facilities and resources, 2021.
- [3] Emeka Chukwu, Lalit Garg, "A Systematic Review of Blockchain in Healthcare: Frameworks Prototypes and Implementations", Access IEEE, vol. 8, pp. 21196-21214, 2020.
- [4] Jabeen, F., Hamid, Z., Akhunzada, A., Abdul, W., and Ghouzali, S., "Trust and Reputation Management in Healthcare Systems: Taxonomy Requirements and Open Issues", IEEE Access, Vol. 6, pp. 17246-17263, 2018.
- [5] Luis Chuquimarca, Dahyana Roca, Washington Torres, Luis Amaya, Jaime Orozco, David Sánchez, "Mobile IoT device for BPM monitoring people with heart problems", Electrical Communication and Computer Engineering (ICECCE) 2020 International Conference on, pp. 1-5, 2020.
- [6] Mishra, K. N., and Chakraborty, C., "A novel approach towards using big data and iot for improving the efficiency of m-health systems" in Advanced computational intelligence techniques for virtual reality in healthcare, Springer, pp. 123-139, 2020.
- [7] Niharika Kumar, "IoT architecture and system design for healthcare systems" in , IEEE, 2019.
- [8] Tang, W., Ren, J., Deng, K. and Zhang, Y., "Secure data aggregation of lightweight e-Healthcare IoT devices with fair incentives", IEEE Internet Things J., Vol. 6, No. 5, pp. 8714-8726, Oct. 2019.
- [9] Wu, Christoph Rüdiger, Jean-Michel Redouté and Mehmet Rasit Yuce, "WE-Safe: A wearable IoT sensor node for safety applications via LoRa", IEEE, 2018.
- [10] Yulei Wu, "Robust Learning-Enabled Intelligence for the Internet of Things: A Survey From the Perspectives of Noisy Data and Adversarial Examples", Internet of Things Journal IEEE, Vol. 8, No. 12, pp. 9568-9579, 2021.