

IOT BASED HEALTH MONITORING SYSTEM USING NODEMCU

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Electronics and Communication Engineering

By

JAI VASANTHAN (3613163)



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
SCHOOL OF ELECTRICAL AND ELECTRONICS

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

APRIL - 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade 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 JAI VASANTHAN (3613163), who carried out the project entitled "IOT BASED HEALTH MONITORING SYSTEM USING NODEMCU " under our supervision from November 2020 to April 2021.

Internal Guide

Mrs.K.SRILATHA.,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

I JAI VASANTHAN here by declare that the Project Report entitled “IOT BASED HEALTH MONITORING SYSTEM USING NODEMCU” done by me under the guidance of Mrs.K.SRILATHA,M.E.,(Ph.D)., Assistant Professor, Department of Electronics and Communication Engineering is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in ELECTRONICS AND COMMUNICATION ENGINEERING.

DATE:
PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

1.

.

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of SATHYABAMA for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

I convey my thanks to Dr.N.M.NANDHITHA.,M.E.,Ph.D., Dean, School of ELECTRICAL AND ELECTRONICS and Dr.T.RAVI.,M.E.,Ph.D., Head of the Department, Dept. of Electronics and Communication Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express our sincere and deep sense of gratitude to my Project Guide Mrs.K.SRILATHA.,M.E.,(Ph.D)., Assistant Professor, Dept of Electronics and Communication Engineering for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of us projectwork.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Electronics and Communication Engineering who were helpful in many ways for the completion of the project.

ABSTRACT

Health is the most important part of any human's life. Most people live a busy life in which going to a doctor for weekly or even monthly check-up is an impossible task. Without monitoring your health it is not possible to know whether you are a healthy or a sick person. So to resolve these issues, we design a project that can monitor your health every day without going to a doctor. Internet of Things (IoT) is a new technological paradigm that can connect things from various fields through the Internet. For the IoT connected healthcare applications, all the sensors are connected to Arduino UNO. Through this system, we can measure ECG, heartbeat, BP. Through sensors, it is possible to measure all these values. Here all the sensors are powered using a solar power system. All these analog sensors can be connected to Arduino through any of the six analog pins. These values are then used for detecting any critical situation. In the case of a critical situation, an alert can be given as a message. Also, it is possible to monitor the person's health from any location in the world through the Thingspeak cloud. Data from sensors is uploaded to the Thingspeak periodically without any interruption if the internet is available. Here ESP8266 Wi-Fi module is used for connecting Arduino to the internet. Healthcare monitoring system in hospitals and many other health centers has experienced significant growth, and portable healthcare monitoring systems with emerging technologies are becoming of great concern to many countries worldwide nowadays.

The advent of Internet of Things (IoT) technologies facilitates the progress of healthcare from face-to-face consulting to telemedicine. This paper proposes a smart healthcare system in IoT environment that can monitor a patient's basic health signs as well as the room condition where the patients are now in real-time. In this system, five sensors are used to capture the data from hospital environment named heart beat sensor, body temperature sensor through this project we can measure a person's different health parameters effortlessly. The condition of the patients is conveyed via a portal to medical staff, where they can process and analyze the current situation of the patients. The developed prototype is well suited for healthcare monitoring that is proved by the effectiveness of the system

CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	v
1	INTRODUCTION	1
2	LITERATURE SURVEY	3
3	EXISTING SYSTEM	6
	3.1 Patients health monitoring system	7
	3.2 drawbacks of existing system	8
4	PROPOSED SYSTEM	9
	4.1 block diagram	11
	Nodemcu	13
	How to use ESP8266 module	14
	Applications	28
	LM35-temperature sensor	29
	ECG	34
	BP Sensor	40
	Pulse sensor	44
5	RESULTS	53
6	CONCLUSION	55
	References	56

CHAPTER 1

INTRODUCTION

Most humans live a busy life in which going to a doctor for weekly or even monthly checkup is an impossible task. Without monitoring your health it is not possible to whether you are a healthy or sick person. This problem leads to the design of a product which monitors your health every day without going to a doctor. ECG is the most broadly used cardiovascular disease monitoring technique that measures the electrical activities of the heart. An ECG system is a non-invasive monitor for evaluating the heart electrical activity, for measuring the regularity/rate of heartbeats, and for identifying any damage to the heart. In this paper, a system is designed as a prototype for monitoring alerting based on the health of a person. This system is fully automated little or no human help is needed. Any doctor can monitor this person from anywhere through the internet. Health is always a major concern in every growth the human race is advancing in terms of technology.

Like the recent corona virus attack that has ruined the economy of China to an extent is an example how health care has become of major importance. In such areas where the epidemic is spread, it is always a better idea to monitor these patients using remote health monitoring technology. So Internet of Things (IoT) based health monitoring system is the current solution for it . Health is characterized as a full state of physical, mental, and social well-being and not merely a lack of illness. Health is a fundamental element of people's need for a better life. Unfortunately, the global health problem has created a dilemma because of certain factors, such as poor health services, the presence of large gaps between rural and urban areas, physicians, and nurses unavailability during the hardest time.

IoT is making any objects internally connected in the recent decade and it has been considered as the next technological revolution. Smart health monitoring mechanism , smart parking , smart home , smart city , smart climate , industrial sites , and agricultural fields are some of the applications of IoT. 1 Health is characterized as a full state of physical, mental, and social well-being and not merely a lack of illness. Health is a fundamental element of people's need for a better life. Unfortunately, the global health problem has created a dilemma because of certain factors, such as poor

health services, the presence of large gaps between rural and urban areas, physicians, and nurse's unavailability during the hardest time.

IoT is making any objects internally connected in the recent decade and it has been considered as the next technological revolution. Smart health monitoring mechanism , smart parking , smart home , smart city , smart climate , industrial sites , and agricultural fields are some of the applications of IoT. The most tremendous use of IoT is in healthcare management which provides health and environment condition tracking facilities. IoT is nothing but linking computers to the internet utilizing sensors and networks . These connected components can be used on devices for health monitoring. The used sensors then forward the information to distant locations like M2M, which are machinery for computers, machines for people, handheld devices, or smartphones . It is a simple, energy-efficient, much smarter, scalable, and interoperable way of tracking and optimizing care to any health problem. Nowadays, modern systems are providing a flexible interface , assistant devices , and mental health management to lead a smart life for the human being.

Heart rate and body temperature are the two most significant indicators for human health. Heart rate is the per-minute amount of heartbeats, commonly known as the pulse rate. To measure the pulse rate, an increase in the blood flow volume can be used by calculating the pulses. Normal heart rate ranges between 60 and 100 beats per minute for healthy people. The typical restful heart for adult males is roughly 70 bpm and for adult females 75 bpm . Female with 12 years of age and above, typically have higher rates of heart in contrast with males.

The temperature of human body is simply the heat of body and the sum of heat radiated by the body is scientifically determined. The average person's body temperature relies on different factors such as ambient temperature, the person's gender, and his eating habits. In healthy adults, it is likely to range between 97.8 °F (36.5 °C) and 99 °F (37.2 °C). Different factors such as flu, low-temperature hypothermia, or any other illness may lead to a change in body temperature. In almost all illnesses, fever is a typical indicator . Various methods exist to invasively and noninvasively assess the heart rate and body temperature. For the consumer, noninvasive approaches over a while have proven accurate and convenient . It is suggested that a healthcare should provide good room conditions to facilitate the patients . Some measures like room humidity, level of all gases like CO, and CO₂ can determine the quality of room environment.

CHAPTER 2

LITERATURE SURVEY

Internet of Things (IoT) and cloud computing plays a vital role in today's Tele-monitoring health system. This system keeps track of patient's physiological parameters through collection of body sensors' data using Raspberry Pi board. The patient's health card are developed by the doctors and displayed on a webpage where doctors and patients can access and communicate each other without physical presence[1]. Using cloud computing, the data can be stored, updated and accessed from anywhere in the world. It is very suitable for rural areas where medical facilities are not available. In Remote health monitoring system using IoT, Body wireless sensor Network (BWSN) is used to transmit the patients' health parameters collected through Raspberry Pi microcontroller to the physicians and caretaker wirelessly [2].

Being long range wireless technology, emergency situation of the patient's health is quickly detected and timely intervention leads to save the life of the patient. Owing to costlier healthcare and long waiting time in hospitals, the concept of in-home patient monitoring system have been emerging in the recent years. This system collects data of various body parameters through Biosensors, wearable devices and smart textiles and it transmits the data to central node server securely through Cipher text Policy Attribute Based Encryption (CP-ABE) method. In turn, the server shares the collected data to the hospitals for further treatment. The server rings alarm to the ambulance [3] during emergency situation.

It is very beneficial for elders and chronic patients who require continuous monitoring. The specialized healthcare monitoring system for elderly people is a growing need in the aging population world. This system performs basic health checkups by measuring the body parameters regularly and report the data to the doctors. The result data are then displayed as statements in a web application where doctors and patients can interact with each other[4].

Evaluation is of two parts: 1) Qualitative interviewing and 2) Quantitative Survey. The main challenge is to make elders equipped with for growing new technologies and to become familiarity towards Smartphone, computer, etc. IoT based Smart healthcare with the help of smart devices and objects improves the healthcare monitoring system effectively, thus by reducing the inefficiencies of existing healthcare system. Smart devices with new and upgraded technologies enhances the data accuracy to be

collected, real-time accessibility of patient's condition, intelligent integration of data collected, maintaining the integrated data smartly through cloud service, etc. [5].

IoT along with smart devices reduce complexity and complications in the healthcare system. The penetration of mobile technologies and smart devices over healthcare system cause huge impact on the world. The full-fledge utilization of M-health and E-health applications in today's world is made aware to the people for improving and maintaining the good quality of life. Apart from regular monitoring of patients condition through M-health system, the main objective is to educate them through recommendations of healthy eating habits and effective workout routines for improving their quality of healthy life [6].

In remote mobile health monitoring system, the patient health parameters are recorded by a smart phone by eliminating an additional hardware and transmit data through a web interface[7].

It facilitates end to end monitoring screen through three steps. Firstly, the real time health parameters are measured through wearable sensors and transmitted to a smart phone which shows the patient health status in graphical interface. Secondly, this system provides a data to family member and doctor through web interface for further monitoring. Thirdly, I provides real time alarm if the patient is at emergency situation such as heart attack, etc. Despite monitoring, there are quite few challenges in using the wearable tracking devices for a long time. Firstly, the daily use of wearable tracking devices is based mainly on small size, rough use and low energy consumption [8]. Secondly, the major challenge is of the accuracy, validity and integrity of measurement data with other devices. Thirdly, the usability and the experiences of the user with the device and its friendly supporting software play vital role in continuing regular and long period use of wearable tracking devices. The use of Internet of Things (IoT) and its e-Health applications in the Tele-medicine health system leads to seamless flow of information between doctors and patients, thus making healthcare cost effective and improving the quality of patients' treatment. This system uses the K53 Tower System platform for e-Health applications to expose the benefits of IoT in medical system. The two fundamental aspects in monitoring people at risk are: 1) Prevention 2) Effective and early intervention during medical emergency

References	Techniques/ Methodology	Advantages	Disadvantages
	Wellbeing monitoring through Wireless Sensor Network and cloud computing using IoT	Cost efficient technique and ubiquitous monitoring	Not easy to deploy WSN nodes compared to wired networks.
	Health monitoring through Wireless Body Area Sensor Network (WBASN)	Easy addition of new sensors to existing system.	Sensors should be low in complexity, small in size,
	Cipher text Policy Attribute Based Encryption (CP-ABE) for data security	Access based policy towards data protection.	Difficult to implement in non-interactive group of networks
	Technology Acceptance Model	Widely recognized technologies to be used for easy access	Adoption of new technologies is difficult to equip with for elders
	IOT with smart devices	Real-time data access and intelligent data integration	Constant updation and upgradation of devices is needed.
	Context Model through OWL and SWRL (Semantic Web Rule Language)	Contextual recommendations such as workout routine and healthy eating habits apart from regular monitoring	Recommendations are so general and not personalized for each treatment.
	Monitoring through Smartphone by Indoor Localization Algorithm	Wifi-fingerprints are used to find the location of patients in indoor environment during emergency situation.	Data may not be accurate always due to environmental interference by Wifi-signals.
	Wearable Monitoring devices	Custom monitoring through single platform of hardware and software	Multiple applications lead to complexity

CHAPTER 3

EXISTING SYSTEM

Real-time health monitoring devices provide real-time analysis of the patient's health parameters. Patients are sharing their healthcare information in real time with their caregivers through these devices for flexible health monitoring and management. Internet of medical things is vastly improving the healthcare condition of patients.

"Reacts' secure, versatile and interactive collaboration platform deployed on technology solutions like Philips Lumify can change education and patient care models and enable a positive disruptive change to healthcare," Dr. Yanick Beaulieu, CEO and Founder of IIT, said in a statement. "For years, clinicians in the field have been trying to piece together systems that offer real-time, interactive teleultrasound capabilities. Now, it has truly arrived."

Tracking patient monitoring and drug management has become possible because of the usage of the above Internet of Things (IoT) medical devices in the healthcare sector. Increasing aging population, surging healthcare expenses and an increase in the occurrence of chronic diseases is driving healthcare stakeholders to follow remote health management and patient monitoring seriously.

The system used for health monitoring is the fixed monitoring system, which can be detected only when the patient is in hospital or in bed. Recently accessible systems are huge in size and available only in the hospitals in Intensive Care Unit[5]. Nowadays, zig bee can be used to transmit the patient information to their loved ones or to their concerned doctors.

One of the popular existing health monitoring system would be RPM(Remote Patient Monitoring). The main benefit of remote patient monitoring or homecare telehealth, is that it allows a patient to use a mobile medical device to perform routine tests and send test results to a doctor in real-time without the necessity of visiting a medical institution. This advantage is of great importance for people with long-term illnesses, who have to communicate with healthcare professionals more often. Homecare telehealth enables easy and quick communication between physicians and patients suffering from sleep disorder, mental, cardiovascular, respiratory and other diseases. Doctors can counsel sick people online with no need to wait for the appointment date.

Incorporating RPM in chronic disease management can significantly improve a person's quality of life. Thanks to RPM, chronically ill people can spend more time at home with their families rather than at hospitals. A monitoring device is also able to inform patients about critical deterioration of health when they need to apply for emergency help as soon as possible. RPM allows doctors to reach out to potential patients. While many people can't stand going to clinics or, for certain reasons, can't afford visiting a doctor, RPM will bring free consultation with a physician right to their homes. This is how medical establishments can gain more visitors without overcrowding facilities.

Patients can save considerably thanks to RPM. With home telehealth, it's not necessary to go to a clinic for prescription or ask your physician about tests results. Besides, due to the decreasing number of unnecessary hospital visits, medical institutions become less crowded and can faster admit patients who need an emergency help. For instance, The specialists from the Oxford University Institute of Biomedical Engineering developed an app that allows patients to avoid unnecessary in-person visits. In the facility where the system was tested, the number of doctor appointments has dropped by 25 percent.

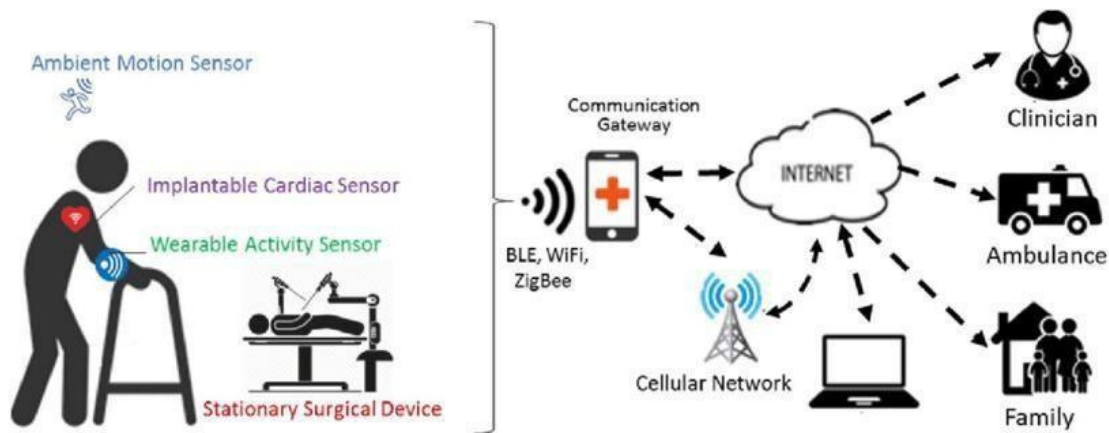


Fig: 3.1 patients health monitoring systems

There are common obstacles that health informatics technologies encounter that applies to Real-Time Health Monitoring Devices. Depending on the comorbidities monitored, Real-Time Health Monitoring Devices involves a diverse selection of devices in its implementation. Standardization is required for data exchange and interoperability among multiple components. Furthermore, Real-Time Health Monitoring Devices deployment is highly dependent on an extensive wireless telecommunications infrastructure, which may not be available or feasible in rural areas.

DRAWBACKS OF EXISTING SYSTEM

Is not accessible for everyone. RPM requires good broadband connectivity, which is hard to achieve for small healthcare institutions and rural hospitals. On the other hand, it's important to take into account that not everyone owns a smartphone, and elderly people often face difficulties in using modern gadgets, such as mobile phones.

Patients and doctors' skepticism. RPM appears to be the least effective patient engagement initiative according to NEJM Catalyst Insights Council survey. The researchers explain these statistics by the fact that the use of wearables is still not available for everybody. By the way, doctors that were taking part in the survey haven't noticed any improvements in chronic disease management. In addition, healthcare professionals based on their experience expressed doubt that the technology alone is likely to change behavior of higher-risk patients. Doctors are also concerned about the difficulties they can face handling received data. Some patients are afraid that their private health data can be obtained by third parties and used for dubious purposes.

The need of additional custom healthcare software. After the data have been collected, IT departments have to redirect it from RPM devices to electronic medical record systems (EMRs) by the means of multiple third-party applications.

Doubtful reliability. Popular fitness wearables tracking physical activity appeared to have large variations of accuracy with error margins up to 25 percent. The reliability of RPM data is called into question as well. For instance, a review in JAMA Dermatology showed that smartphone apps for melanoma detection have a 30 percent failure rate. The lack of reliability is the most serious problem that needs to be fixed before devices and applications could be used by healthcare providers.

In existing system, patient needs to get hospitalised for regular monitoring of the patient. It is not possible once he/she is discharged from the hospital. This system cannot be used at home. The existing systems are measuring the health parameters of the patient and send it through zig bee, Bluetooth protocol etc., These are used for only short range communication to transfer the data. Not all the time the doctor can fetch these details.

CHAPTER 4

PROPOSED SYSTEM

The core objective of this project is the design and implementation of a smart patient health tracking system. Fig.1 shows the overview of the proposed system. The sensors are embedded on the patient body to sense the temperature and heartbeat of the patient. Two more sensors are placed at home to sense the humidity and the temperature of the room where the patient is staying. These sensors are connected to a control unit, which calculates the values of all the four sensors. These calculated values are then transmitted through a IoT cloud to the base station. From the base station the values are then accessed by the doctor at any other location. Thus based on the temperature and heart beat values and the room sensor values, the doctor can decide the state of the patient and appropriate measures can be taken. The temperature sensor connected to the analog pin of the arduino controller is converted into digital value with the help of a DAC [10].

Using this digital data, the controller converts it into the actual temperature value in degree Celsius using the thermistor. 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) [7]. The digital pulses are given to a microcontroller for calculating the heart beat rate, given by the formula (Beats per minute) = $60 \times f$, where f is the pulse frequency. A humidity sensor (or hygrometer) senses, measures and reports both moisture and air temperature. Humidity sensors work by detecting changes that alter electrical currents or temperature in the air. IoT Server At whatever point the patient goes to the healing center premises, sensors sense the physiological signs and these signs are changed over to electrical signals [8]. Then simple electrical signal is changed over to digital signal (computerized information) which is put away in RFID. The digital computerized information is transmitted through Zigbee Protocol to the neighborhood server. Zigbee is appropriate convention for this framework. It comprises of greatest number of cell hubs. It is more favored for gadgets which are little in measure and expend less vitality. From nearby server the information is exchanged to the therapeutic server through WLAN. Medicinal server comprises of

substantial database as given in Table I. At the point when the information is exchanged to the therapeutic server, it checks whether the patient has any past medicinal record then the server adds the new information to that record and exchanges to the specialist. In the event that patient does not have any past therapeutic record then the server makes new ID and stores the information in its database [9]. This information is exchanged to the specialist for diagnosis. The complete Data transmission using IoT is given in Fig.

- In this system for 24x7 human health monitoring is designed and implemented
- In this system, the Arduino Uno board is used for collecting and processing all data
- Wireless devices have invaded the medical area with a wide range of capability.
- To monitor the patient details in periodic interval is on overhead using existing technologies.
- To overcome this we have changed recent wireless sensor technologies.
- Added advanced sensors like pulse oximeter for measuring blood pressure
- Different sensors are used for measuring different parameters
- All this data is uploaded to thingspeak for remote analysis

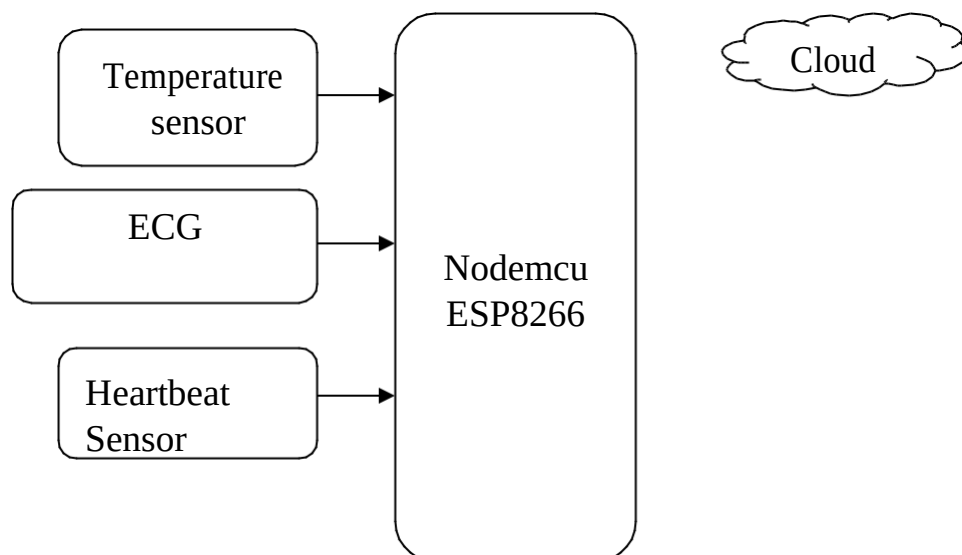
The system which we prefer to develop would not only help in monitoring the health of the patient when he is in bed but also when he is out of bed. The main idea of the system is to transmit the information through the webpage to continuous monitoring of the patient over internet. Such a system would continually detect the important body parameters like temperature, pulse rate and would compare it against predetermined range set and if these values cross the specific limit, it would immediately alert the doctor. In this system microcontroller is used to transmit the data.

With tons of new healthcare technology start-ups, IoT is rapidly revolutionizing the healthcare industry. In this project, we have designed the IoT Based Patient Health Monitoring System using ESP8266 & Arduino. The IoT platform used in this project is ThingSpeak. ThingSpeak is an open-source Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP protocol over the Internet or via a Local Area Network. This IoT device could read the pulse rate and measure the surrounding temperature. It continuously monitors the pulse rate and surrounding temperature and updates them to an IoT platform.

The Arduino Sketch running over the device implements the various functionalities of the project like reading sensor data, converting them into strings, passing them to the IoT platform, and displaying measured pulse rate and temperature on character LCD.

This is a simple block diagram that explains the IoT Based Patient Health Monitoring System using ESP8266 & Arduino. Pulse Sensor and LM35 Temperature Sensors measure BPM & Environmental Temperature respectively. The Arduino processes the code and displays it to 16*2 LCD Display. ESP8266 Wi-Fi module connects to Wi-Fi and sends the data to IoT device server. The IoT server used here is Thingspeak. Finally, the data can be monitored from any part of the world by logging into the Thingspeak channel.

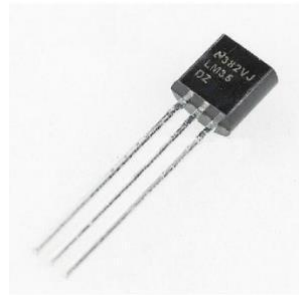
Block Diagram



MODULES:

Temperature sensor:

A temperature sensor is a device, usually an RTD (resistance temperature detector) or a thermocouple, that collects the data about temperature from a particular patient and converts the data into understandable form for a device or an observer. The collected data are uploaded to the cloud.



Heart beat sensor:

Heartbeat Sensor is an electronic device that is used to measure the heart rate i.e. speed of the heartbeat. Heart Rate can be monitored in two ways: one way is to manually check the pulse either at wrists or finger then the data are uploaded to the cloud.



Hardware Modules:

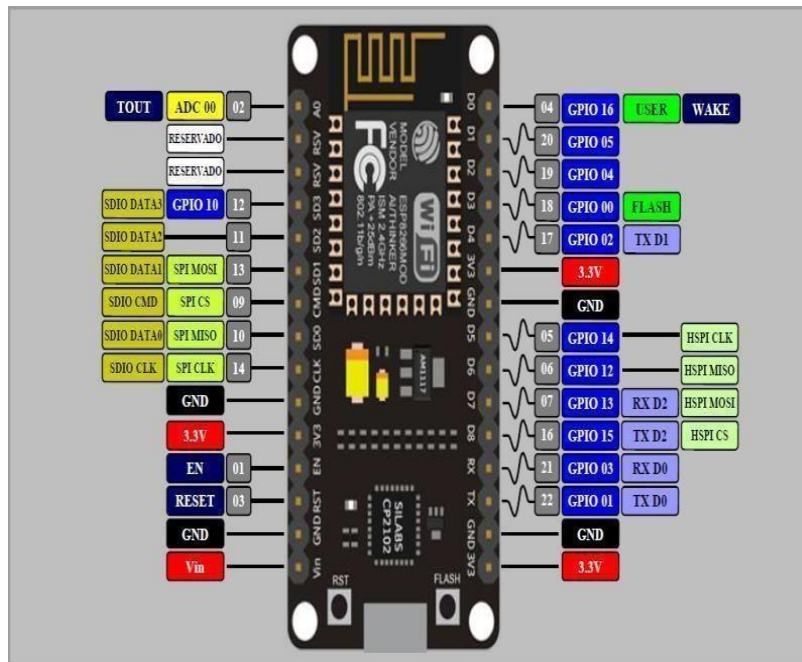
- ☐ NODE-MCU
- ☐ Temperature sensor
- ☐ Blood pressure sensor
- ☐ Heartbeat Sensor

Hardware explanation

Nodemcu

Introduction

General-purpose input/output (GPIO) is a pin on an IC (Integrated Circuit). It can be either input pin or output pin, whose behaviour can be controlled at the run time.



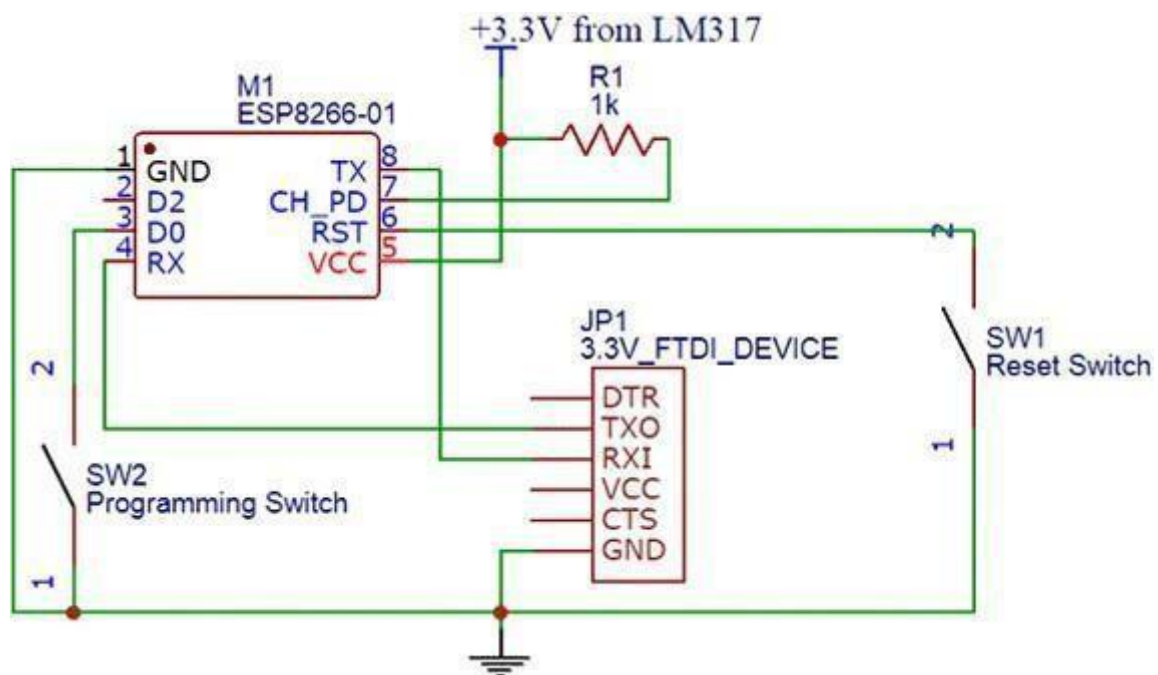
Where to use ESP8266-01

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. It can also fetch data from internet using API's hence your project could access any information that is available in the internet, thus making it smarter. Another exciting feature of this module is that it can be programmed using the Arduino IDE which makes it a lot more user friendly. However this version of the module has only 2 GPIO pins (you can hack it to use upto 4) so you have to use it along with another microcontroller like Arduino, else you can look onto the more standalone ESP-12 or ESP-32 versions. So if you are looking for a module to get started with IOT or to provide internet connectivity to your project then this module is the right choice for you.

How to use the ESP8266 Module

There are so many methods and IDEs available to with ESP modules, but the most commonly used on is the Arduino IDE. So let us discuss only about that further below.

The ESP8266 module works with 3.3V only, anything more than 3.7V would kill the module hence be cautions with your circuits. The best way to program an ESP-01 is by using the FTDI board that supports 3.3V programming. If you don't have one it is recommended to buy one or for time being you can also use an Arduino board. One commonly problem that every one faces with ESP-01 is the powering up problem. The module is a bit power hungry while programming and hence you can power it with a 3.3V pin on Arduino or just use a potential divider. So it is important to make a small voltage regulator for 3.31v that could supply a minimum of 500mA. One recommended regulator is the LM317 which could handle the job easily. A simplified circuit diagram for using the ESP8266-01 module is given below

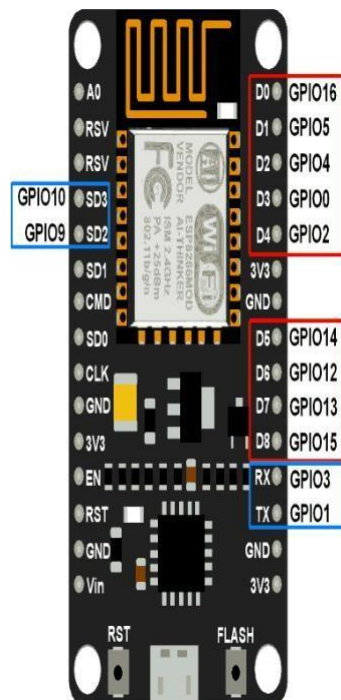


The switch SW2 (Programming Switch) should be held pressed to hold the GPIO-0 pin to ground. This way we can enter into the programming mode and upload the code. Once the code is released the switch can be released.

GPIO PINS:

NodeMCU Development kit provides access to these GPIOs of ESP8266. The only thing to take care is that NodeMCU Dev kit pins are numbered differently than internal GPIO notations of ESP8266 as shown in below figure and table.

The Internet of Things (IoT) has been a trending field in the world of technology. It has changed the way we work. Physical objects and the digital world are connected now more than ever. Keeping this in mind, Expressive Systems (A Shanghai-based Semiconductor Company) has released an adorable, bite-sized Wi-Fi enabled microcontroller – ESP8266, it can monitor and control things from anywhere in the world



NodeMCU GPIOs

Pin Names on NodeMCU Development Kit	ESP8266 Internal GPIO Pin number
D0	GPIO16
D1	GPIO5
D2	GPIO4
D3	GPIO0
D4	GPIO2
D5	GPIO14
D6	GPIO12
D7	GPIO13
D8	GPIO15
D9/RX	GPIO3
D10/TX	GPIO1
D11/SD2	GPIO9
D12/SD3	GPIO10

ESP8266 is a system on a chip (SoC) design with components like the processor chip. The processor has around 16 GPIO lines, some of which are used internally to interface with other components of the SoC, like flash memory.

Since several lines are used internally within the ESP8266 SoC, we have about 11 GPIO pins remaining for GPIO purpose.

Now again 2 pins out of 11 are generally reserved for RX and TX in order to communicate with a host PC from which compiled object code is downloaded.

Hence finally, this leaves just 9 general purpose I/O pins i.e. D0 to D8.

As shown in above figure of NodeMCU Dev Kit. We can see RX, TX, SD2, SD3 pins are not mostly used as GPIOs since they are used for other internal process. But we can try with SD3 (D12) pin which mostly like to respond for GPIO/PWM/interrupt like functions.

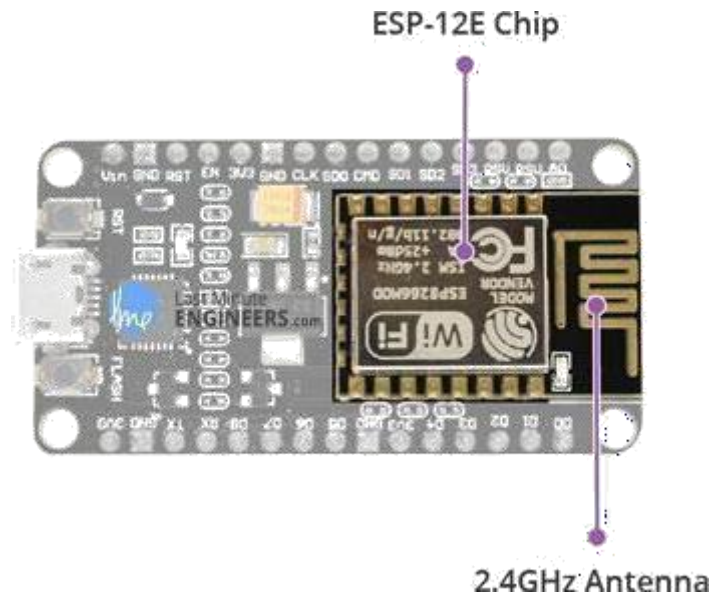
Note that D0/GPIO16 pin can be only used as GPIO read/write, no special functions are supported on it.

ESP-12E Module

The development board equips the ESP-12E module containing ESP8266 chip having Tensilica Xtensa® 32-bit LX106 RISC microprocessor which operates at 80 to 160 MHz adjustable clock frequency and supports RTOS.

ESP-12E Chip

- ❑ Tensilica Xtensa® 32-bit LX106
- ❑ 80 to 160 MHz Clock Freq.
- ❑ 128kB internal RAM
- ❑ 4MB external flash
- 802.11b/g/n Wi-Fi transceiver



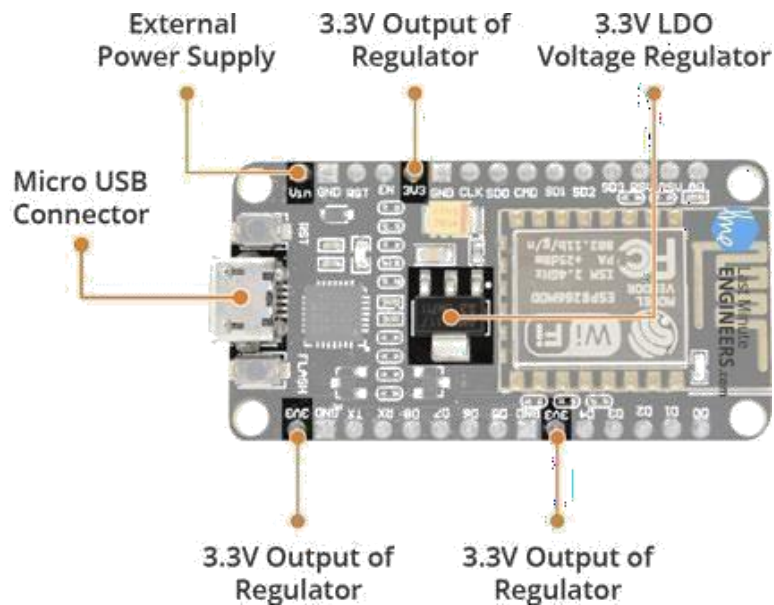
There's also 128 KB RAM and 4MB of Flash memory (for program and data storage) just enough to cope with the large strings that make up web pages, JSON/XML data, and everything we throw at IoT devices nowadays.

The ESP8266 Integrates 802.11b/g/n HT40 Wi-Fi transceiver, so it can not only connect to a WiFi network and interact with the Internet, but it can also set up a network of its own, allowing other devices to connect directly to it. This makes the ESP8266 NodeMCU even more versatile

Power Requirement

As the operating voltage range of ESP8266 is 3V to 3.6V, the board comes with a LDO voltage regulator to keep the voltage steady at 3.3V. It can reliably supply up to 600mA, which should be more than enough when ESP8266 pulls as much as 80mA during RF transmissions. The output of the regulator is also broken out to one of the sides of the board and labeled as 3V3. This pin can be used to supply power to external components.

- ☐ Operating Voltage: 2.5V to 3.6V
- ☐ On-board 3.3V 600mA regulator
- ☐ 80mA Operating Current
20 μ A during Sleep Mode



Power to the ESP8266 NodeMCU is supplied via the on-board MicroB USB connector. Alternatively, if you have a regulated 5V voltage source, the VIN pin can be used to directly supply the ESP8266 and its peripherals.

WARNING

The ESP8266 requires a 3.3V power supply and 3.3V logic levels for communication. The GPIO pins are not 5V-tolerant! If you want to interface the board with 5V (or higher) components, you'll need to do some level shifting.

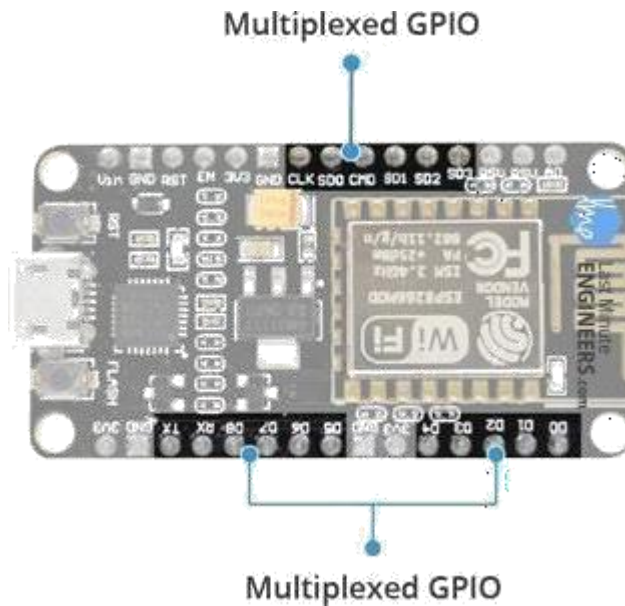
Peripherals and I/O

The ESP8266 NodeMCU has total 17 GPIO pins broken out to the pin headers on both sides of the development board. These pins can be assigned to all sorts of peripheral duties, including:

- ☐ ADC channel – A 10-bit ADC channel.
- ☐ UART interface – UART interface is used to load code serially.
- ☐ PWM outputs – PWM pins for dimming LEDs or controlling motors.
- ☐ SPI, I2C & I2S interface – SPI and I2C interface to hook up all sorts of sensors and peripherals.
- ☐ I2S interface – I2S interface if you want to add sound to your

project. **Multiplexed I/Os**

- ☐ 1 ADC channels
- ☐ 2 UART interfaces
- ☐ 4 PWM outputs
- ☐ SPI, I2C & I2S interface



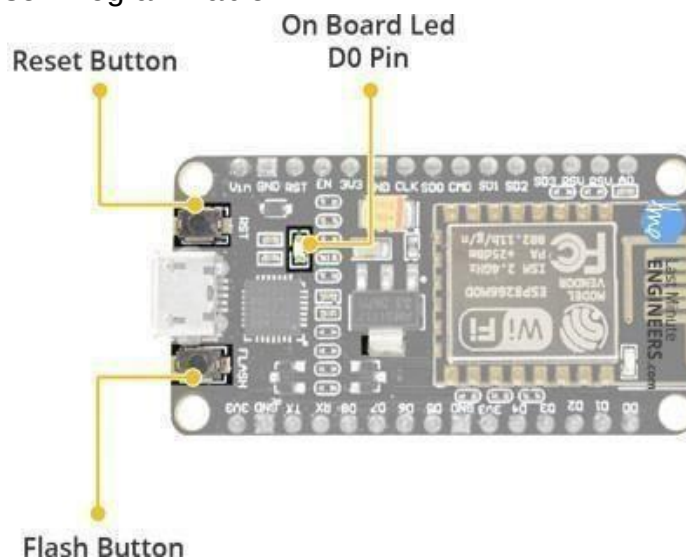
the ESP8266's pin multiplexing feature (Multiple peripherals multiplexed on a single GPIO pin). Meaning a single GPIO pin can act as PWM/UART/SPI.

On-board Switches & LED Indicator

The ESP8266 NodeMCU features two buttons. One marked as RST located on the top left corner is the Reset button, used of course to reset the ESP8266 chip. The other FLASH button on the bottom left corner is the download button used while upgrading firmware.

Switches & Indicators

- ☐ RST – Reset the ESP8266 chip
- ☐ FLASH – Download new programs
- ☐ Blue LED – User Programmable

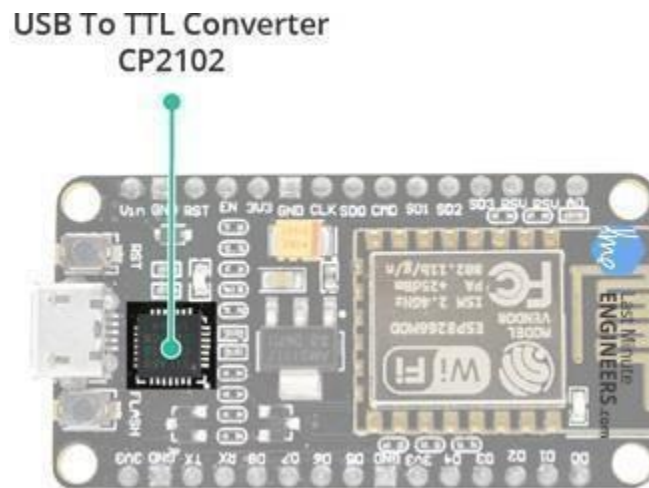


The board also has a LED indicator which is user programmable and is connected to the D0 pin of the board.

Serial Communication

The board includes CP2102 USB-to-UART Bridge Controller from Silicon Labs, which converts USB signal to serial and allows your computer to program and communicate with the ESP8266 chip.

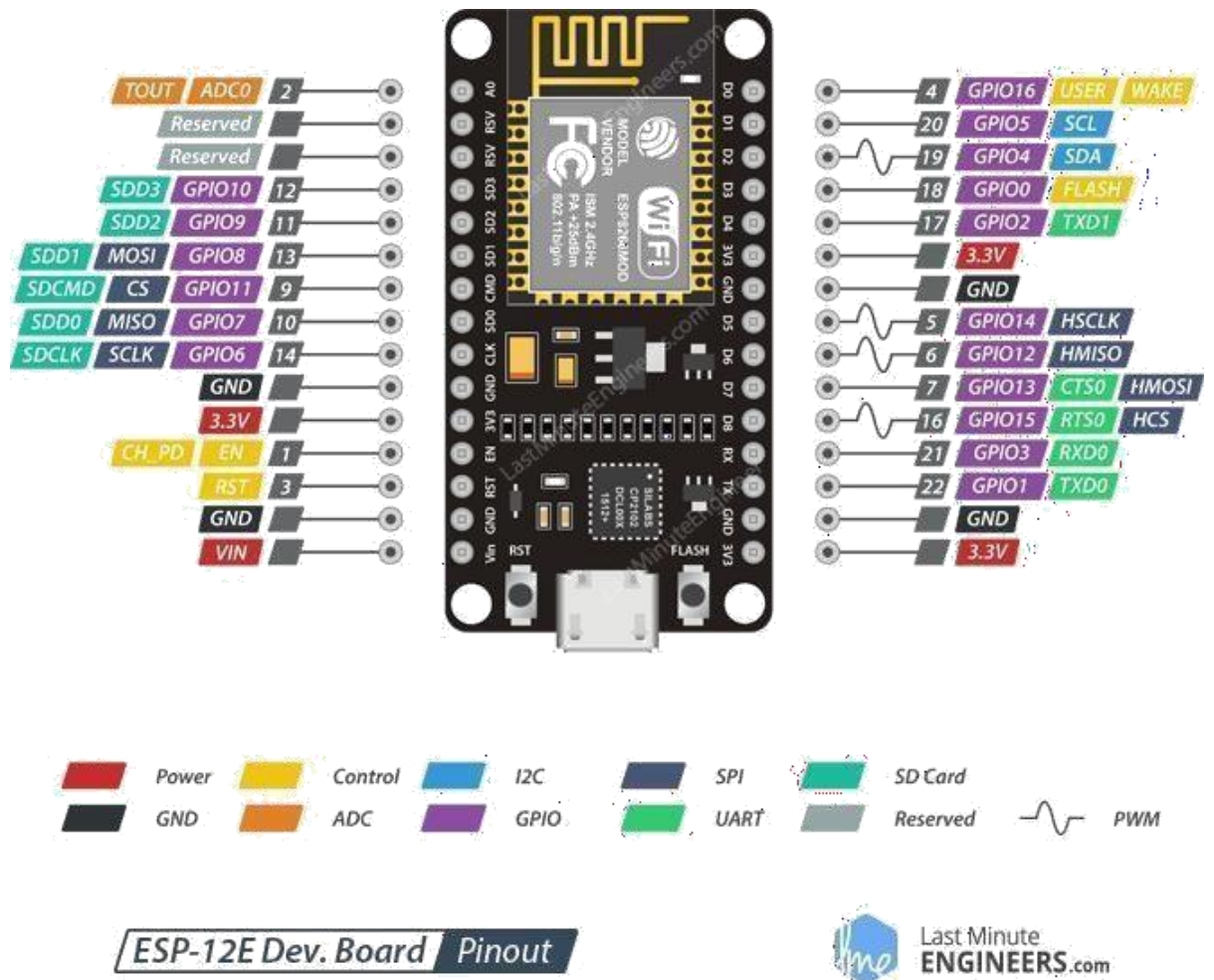
- ☐ CP2102 USB-to-UART converter
- ☐ 4.5 Mbps communication speed
- ☐ Flow Control support



If you have an older version of CP2102 driver installed on your PC, we recommend upgrading now.

ESP8266 NodeMCU Pinout

The ESP8266 NodeMCU has total 30 pins that interface it to the outside world. The connections are as follows:



For the sake of simplicity, we will make groups of pins with similar functionalities **Power Pins**

There are four power pins viz. one VIN pin & three 3.3V pins. The VIN pin can be used to directly supply the ESP8266 and its peripherals, if you have a regulated 5V voltage source. The 3.3V pins are the output of an on-board voltage regulator. These pins can be used to supply power to external components.

GND

GND is a ground pin of ESP8266 NodeMCU development board.

I2C Pins

I2C Pins are used to hook up all sorts of I2C sensors and peripherals in your project. Both I2C Master and I2C Slave are supported. I2C interface functionality can be realized programmatically, and the clock frequency is 100 kHz at a maximum. It should be noted that I2C clock frequency should be higher than the slowest clock frequency of the slave device.

digital enabled GPIO can be configured to internal pull-up or pull-down, or set to high impedance. When configured as an input, it can also be set to edge-trigger or level-trigger to generate CPU interrupts.

ADC Channel

The NodeMCU is embedded with a 10-bit precision SAR ADC. The two viz. functions can be implemented using ADC Testing power supply voltage of TOUT VDD3P3 pin and testing input voltage of pin. However, they cannot be implemented at the same time.

UART Pins

ESP8266 NodeMCU has 2 UART interfaces, i.e. UART0 and UART1, which provide asynchronous communication (RS232 and RS485), and can communicate at up to 4.5 Mbps. UART0 (TXD0, RXD0, RST0 & CTS0 pins) can be used for communication. It supports flow control. However, UART1 (TXD1 pin) features only data transmit signal so, it is usually used for printing log.

SPI Pins

ESP8266 features two SPIs (SPI and HSPI) in slave and master modes. These SPIs also support the following general-purpose SPI features:

- ☐ 4 timing modes of the SPI format transfer
- ☐ Up to 80 MHz and the divided clocks of 80 MHz
- ☐ Up to 64-Byte FIFO

SDIO Pins

ESP8266 features Secure Digital Input/Output Interface (SDIO) which is used to directly interface SD cards. 4-bit 25 MHz SDIO v1.1 and 4-bit 50 MHz SDIO v2.0 are supported.

PWM Pins

The board has 4 channels of Pulse Width Modulation (PWM). The PWM output can be implemented programmatically and used for driving digital motors and LEDs. PWM frequency range is adjustable from 1000 μ s to 10000 μ s, i.e., between 100 Hz and 1 kHz.

Control Pins

Control Pins are used to control ESP8266. These pins include Chip Enable pin (EN), Reset pin (RST) and WAKE pin.

- ☐ EN pin – The ESP8266 chip is enabled when EN pin is pulled HIGH. When pulled LOW the chip works at minimum power.
- ☐ RST pin – RST pin is used to reset the ESP8266 chip.
- ☐ WAKE pin – Wake pin is used to wake the chip from deep-sleep.

ESP8266 Development Platforms

Now, let's move on to the interesting stuff!

There are a variety of development platforms that can be equipped to program the ESP8266. You can go with Espruino – JavaScript SDK and firmware closely emulating Node.js, or use Mongoose OS – An operating system for IoT devices (recommended platform by Espressif Systems and Google Cloud IoT) or use a software development kit (SDK) provided by Espressif or one of the platforms listed on Wikipedia.

Fortunately, the amazing ESP8266 community took the IDE selection a step further by creating an Arduino add-on. If you're just getting started programming the ESP8266, this is the environment we recommend beginning with, and the one we'll document in this tutorial.

This ESP8266 add-on for Arduino is based on the amazing work by Ivan Grokhotkov and the rest of the ESP8266 community. Check out the ESP8266 Arduino GitHub repository for more information.

Installing the ESP8266 Core on Windows OS

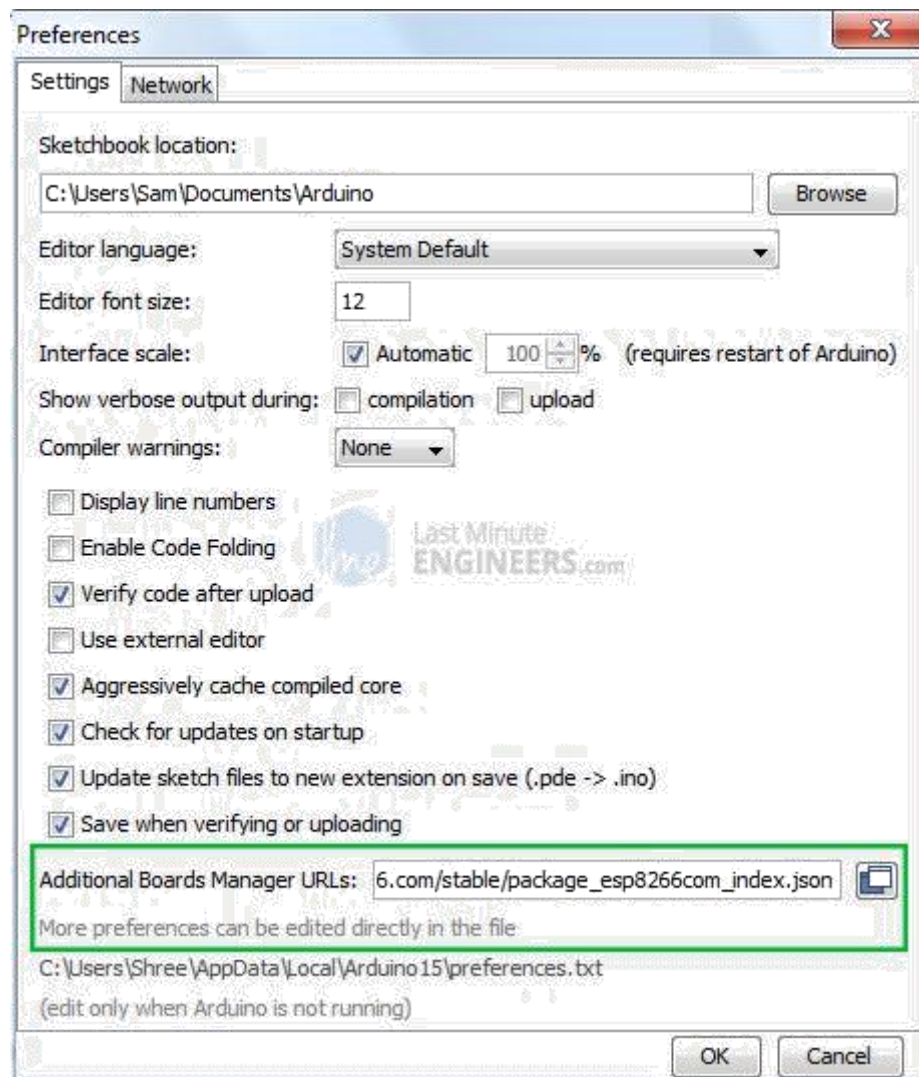
Let's proceed with installing ESP8266 Arduino core.

The first thing is having latest Arduino IDE (Arduino 1.6.4 or higher) installed on your PC. If don't have it, we recommend upgrading now.

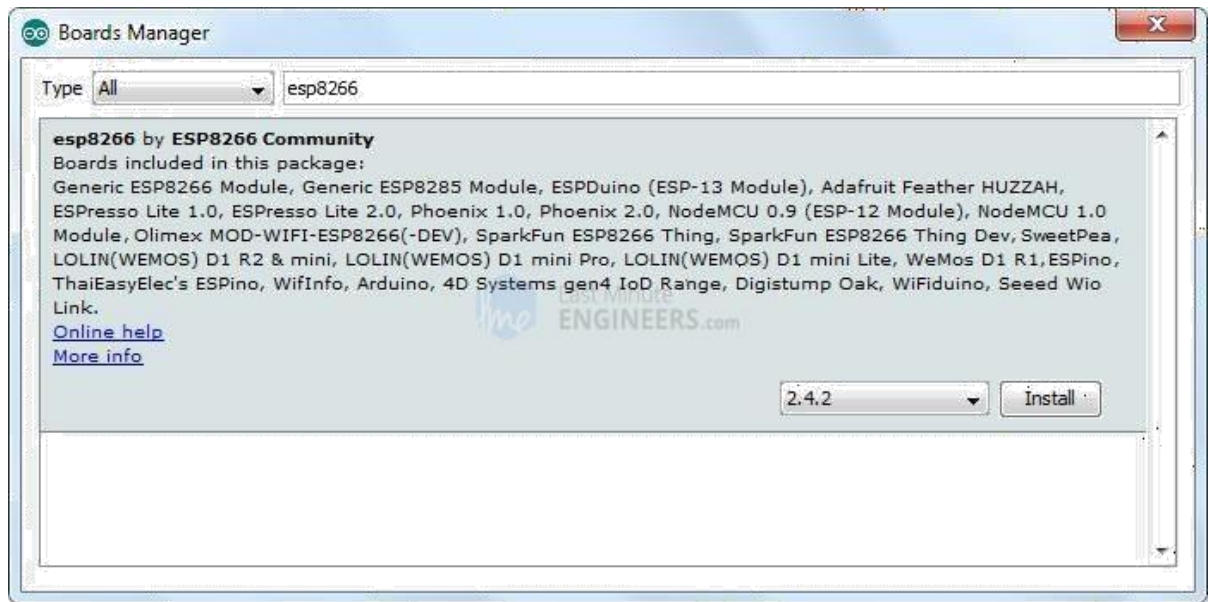
Latest Arduino IDE

To begin, we'll need to update the board manager with a custom URL. Open up Arduino IDE and go to File > Preferences. Then, copy below URL into the Additional Board Manager URLs text box situated on the bottom of the window:

http://arduino.esp8266.com/stable/package_esp8266com_index.json



Hit OK. Then navigate to the Board Manager by going to Tools > Boards > Boards Manager. There should be a couple new entries in addition to the standard Arduino boards. Filter your search by typing esp8266. Click on that entry and select Install.



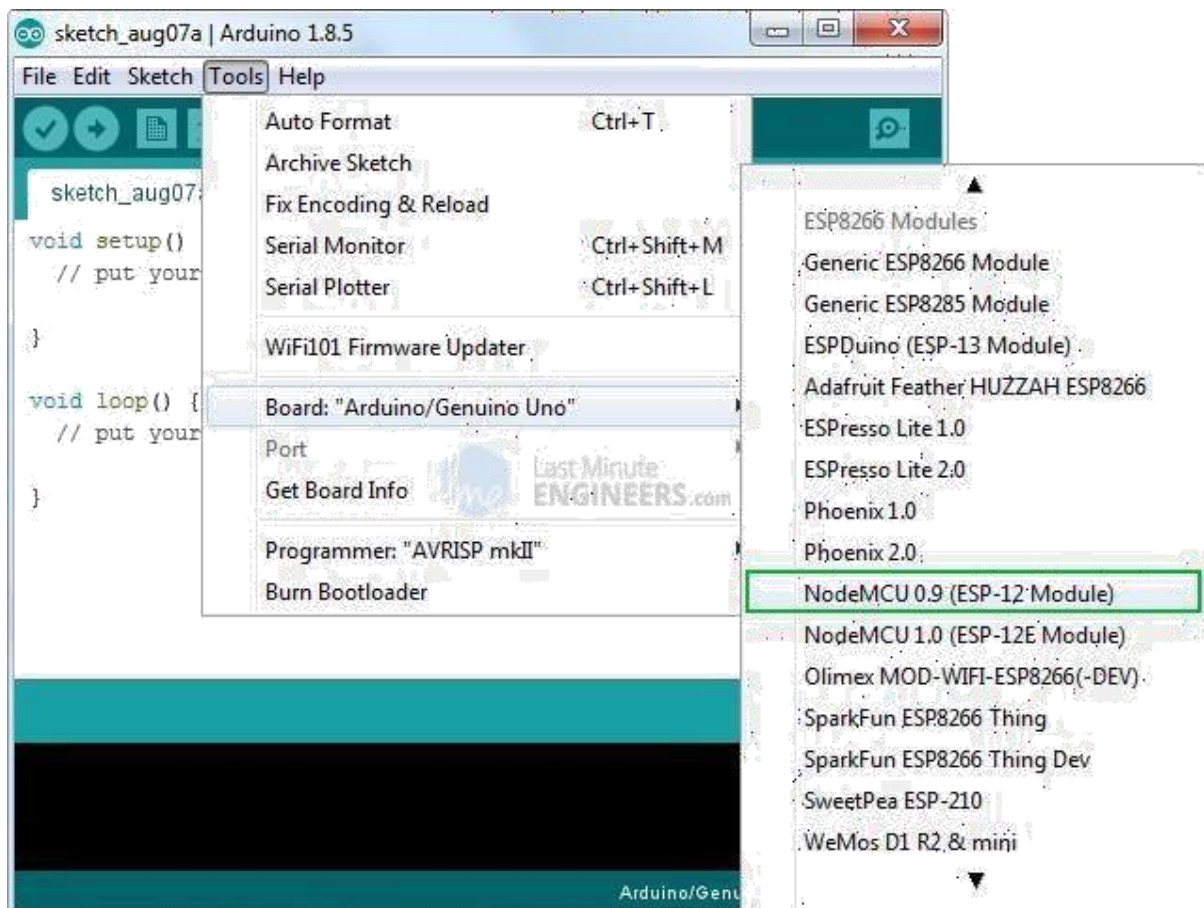
The board definitions and tools for the ESP8266 include a whole new set of gcc, g++, and other reasonably large, compiled binaries, so it may take a few minutes to download and install (the archived file is ~110MB). Once the installation has completed, a small INSTALLED text will appear next to the entry. You can now close the Board Manager.

Arduino Example: Blink

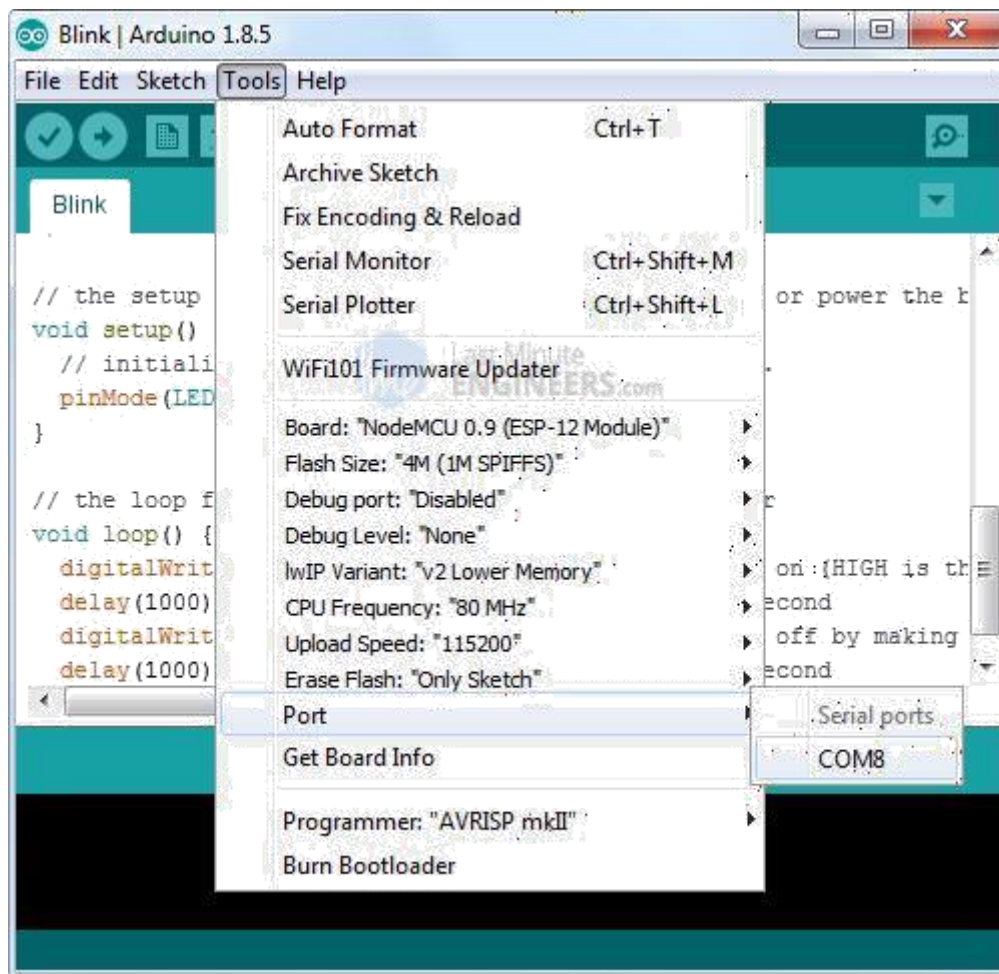
To make sure ESP8266 Arduino core and the NodeMCU are properly set up, we'll upload the simplest sketch of all – The Blink!

We will use the on-board LED for this test. As mentioned earlier in this tutorial, D0 pin of the board is connected to on-board Blue LED & is user programmable. Perfect!

Before we get to uploading sketch & playing with LED, we need to make sure that the board is selected properly in Arduino IDE. Open Arduino IDE and select NodeMCU 0.9 (ESP-12 Module) option under your Arduino IDE > Tools > Board menu.



Now, plug your ESP8266 NodeMCU into your computer via micro-B USB cable. Once the board is plugged in, it should be assigned a unique COM port. On Windows machines, this will be something like COM#, and on Mac/Linux computers it will come in the form of /dev/tty.usbserial-XXXXXX. Select this serial port under the Arduino IDE > Tools > Port menu. Also select the Upload Speed : 115200



WARNING

More attention needs to be given to selecting board, choosing COM port and selecting Upload speed. You may get `espcomm_upload_mem` error while uploading new sketches, if failed to do so.

Once you are done, try the example sketch

below. `void setup()`

```
{
    pinMode(D0, OUTPUT);
```

```
}
```

`void loop()`

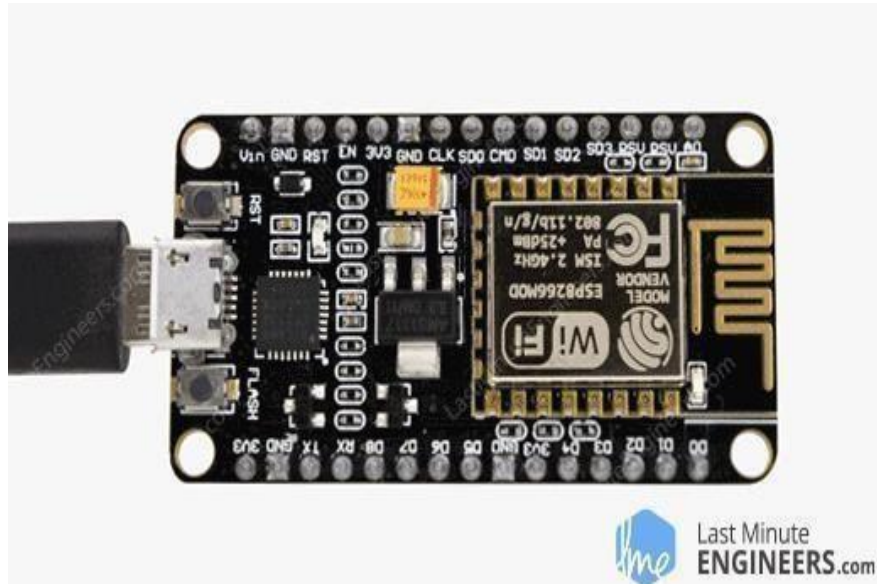
```
{
    digitalWrite(D0,
HIGH); delay(500);
    digitalWrite(D0, LOW);
```

```

    delay(500);
}

```

Once the code is uploaded, LED will start blinking. You may need to tap the RST button to get your ESP8266 to begin running the sketch.



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

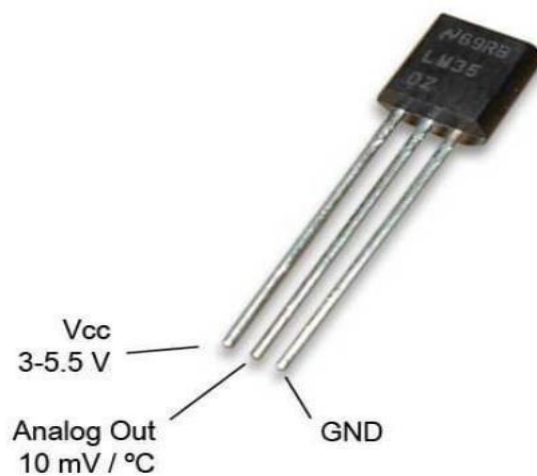
Applications

- ☐ IOT Projects
- ☐ Access Point Portals
- ☐ Wireless Data logging
- ☐ Smart Home Automation

- ❑ Learn basics of networking
- ❑ Portable Electronics
- ❑ Smart bulbs and Sockets

LM35 - Precision Centigrade Temperature Sensor

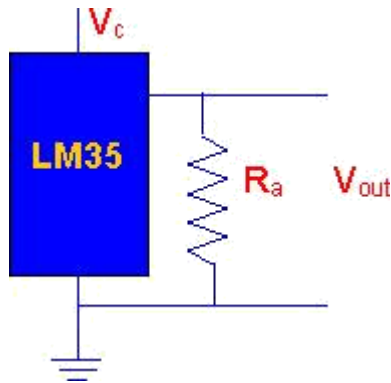
A temperature sensor is a device, usually an RTD (resistance temperature detector) or a thermocouple, that collects the data about temperature from a particular source and converts the data into understandable form for a device or an observer. Temperature sensors are used in many applications like HV and AC system environmental controls, food processing units, medical devices, chemical handling and automotive under the hood monitoring and controlling systems, etc.



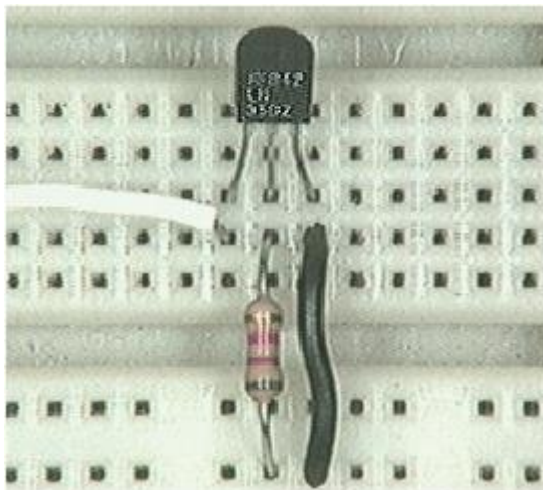
The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of $\pm\frac{1}{4}^{\circ}\text{C}$ at room temperature and $\pm\frac{3}{4}^{\circ}\text{C}$ over a full -55 to +150°C temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 μA from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35 is rated to operate over a -55° to +150°C temperature range, while the LM35C is rated for a -40° to +110°C range (-10° with improved accuracy). The LM35 series is available packaged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package.

Operation of LM35:

- The LM35 can be connected easily in the same way as other integrated circuit temperature sensors. It can be stuck or established to a surface and its temperature will be within around the range of 0.01°C of the surface temperature.
- This presumes that the ambient air temperature is just about the same as the surface temperature; if the air temperature were much higher or lower than the surface temperature, the actual temperature of the LM35 die would be at an intermediate temperature between the surface temperature and the air temperature.



- Here is a photo of the LM 35 wired on a circuit board.
 - The white wire in the photo goes to the power supply.
 - Both the resistor and the black wire go to ground.
 - The output voltage is measured from the middle pin to ground.



- What Can You Expect When You Use An LM35?
 - You will need to use a voltmeter to sense Vout.
 - The output voltage is converted to temperature by a simple conversion factor.
 - The sensor has a sensitivity of 10mV / °C.
 - Use a conversion factor that is the reciprocal, that is 100V/ °C.
 - The general equation used to convert output voltage to temperature is:
 - Temperature (°C) = Vout * (100 °C/V)
 - So if Vout is 1V , then, Temperature = 100 °C
 - The output voltage varies linearly with temperature.

Features

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy guaranteeable (at +25°C)
- Rated for full -55° to +150°C range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than 60 µA current drain
- Low self-heating, 0.08°C in still air
- Nonlinearity only $\pm 1/4^\circ\text{C}$ typical
- Low impedance output, 0.1 Ohm for 1 mA load

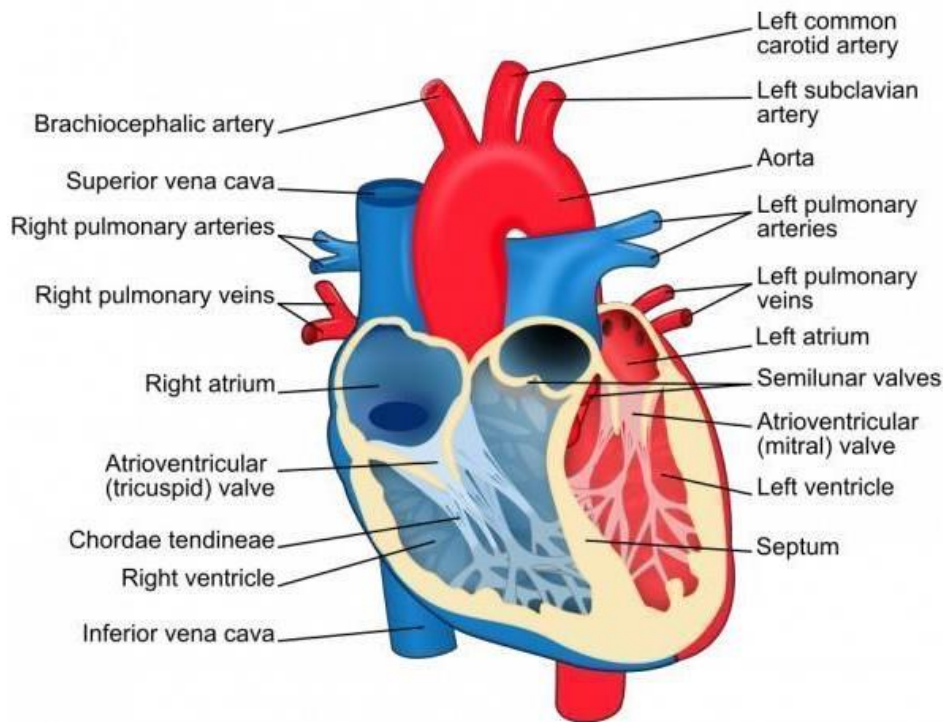
Types of Temperature Sensors

There are different types of temperature sensors that have sensing capacity depending upon their range of application. Different types of temperature sensors are as follows:

- Thermocouples
- Resistor temperature detectors
- Thermistors
- Infrared sensors
- Semiconductors
- Thermometers

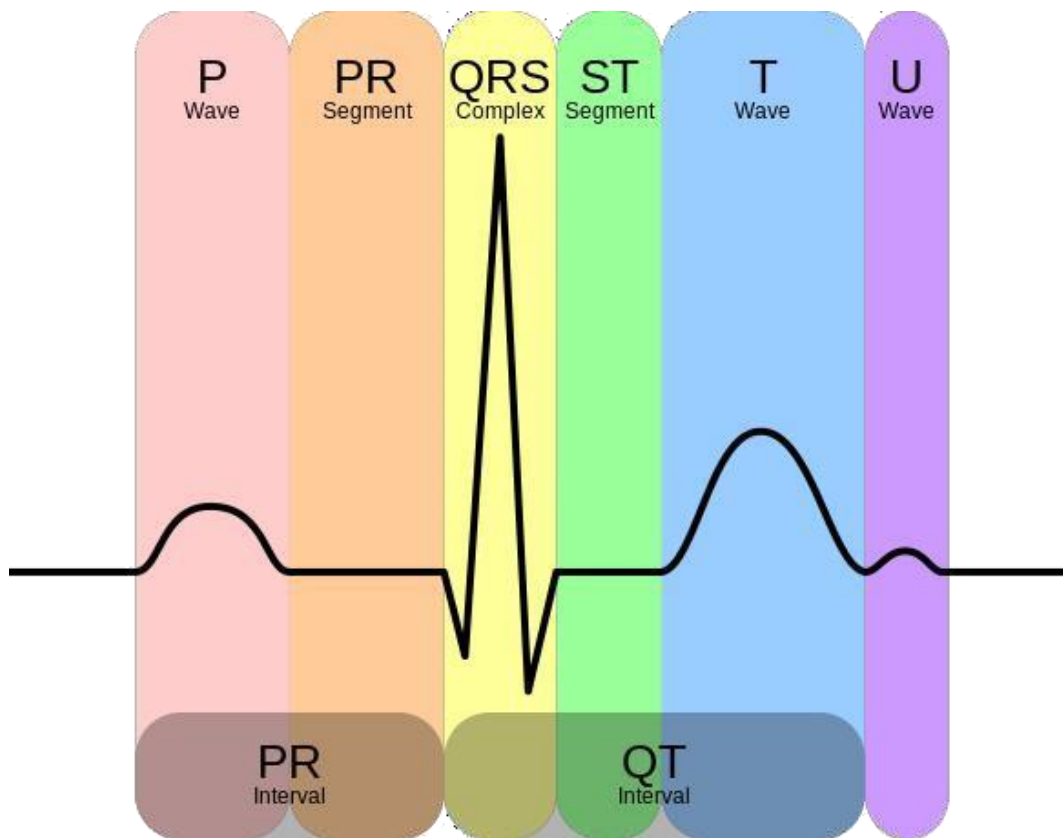
ECG

An ECG is a paper or digital recording of the electrical signals in the heart. It is also called an electrocardiogram or an EKG. The ECG is used to determine heart rate, heart rhythm and other information regarding the heart's condition. ECGs are used to help diagnose heart arrhythmias, heart attacks, pacemaker function and heart failure.



Understanding the ECG

In general terms, let's look at what an ECG is representing and how we're able to sense it. The ECG is separated into two basic Intervals, the PR Interval and the QT Interval, described below.



PR Interval

The PR interval is the initial wave generated by an electrical impulse traveling from the right atrium to the left. The right atrium is the first chamber to see an electrical impulse. This electrical impulse causes the chambers to "depolarize". This forces it to contract and drain deoxygenated blood from both the Superior and Inferior vena cava into the right ventricle. As the electrical impulse travels across the top of the heart it then triggers the left atrium to contract. The left atrium is responsible for receiving newly oxygenated blood from the lungs into the left ventricle via the left and right pulmonary veins. The pulmonary veins are red in the diagram because they are carrying oxygenated blood. They are still called veins because veins carry blood towards the heart. Science!

QT Interval

The QT Interval is where things get really interesting. The QRS is a complex process that generates the signature "beep" in cardiac monitors. During QRS both ventricles begin to pump. The right ventricle begins to pump deoxygenated blood into the lungs through the left and right pulmonary arteries. The pulmonary arteries are blue in the diagram because they are carrying deoxygenated blood. They are still called arteries because arteries carry blood away the heart. Science, Again! The left ventricle is also beginning to pump freshly oxygenated blood through the aorta and into the rest of the body. After the initial contraction comes the ST segment. The ST segment is fairly quiet electrically as it is the time where the ventricles waiting to be "re-polarized". Finally the T wave becomes present to actively "re-polarize", or relax the ventricles. This relaxation phase resets the ventricles to be filled again by the atriums.

Medical uses of ECG

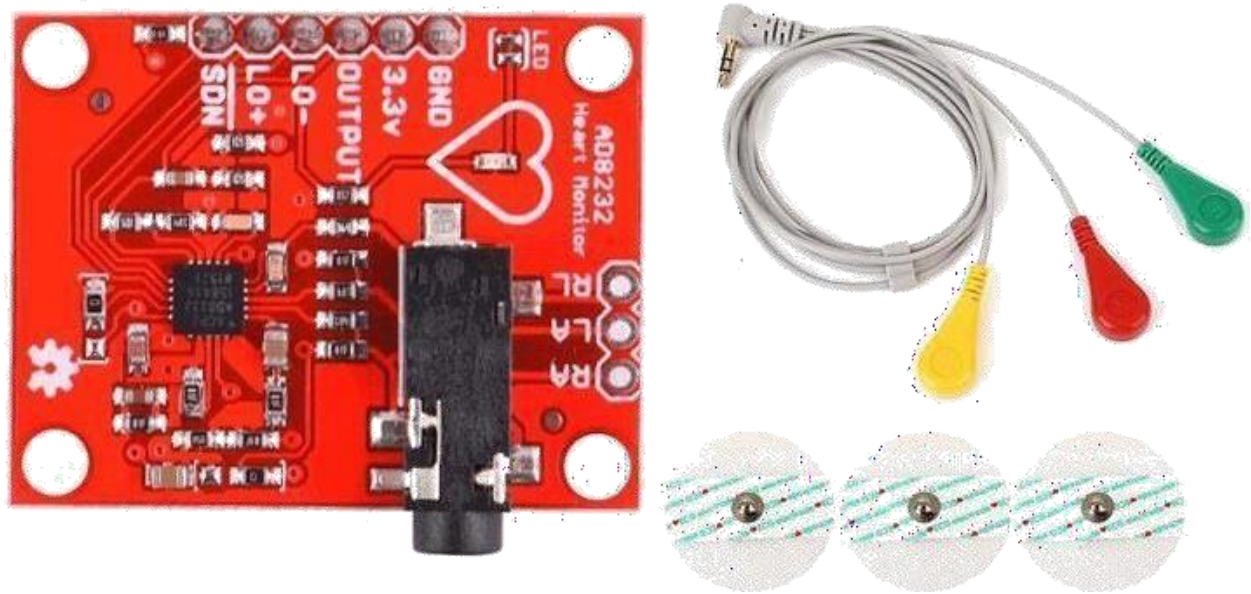
An electrocardiogram can be a useful way to find out whether your high blood pressure has caused any damage to your heart or blood vessels. Because of this, you may be asked to have an ECG when you are first diagnosed with high blood pressure.

Some of the things an ECG reading can detect are:

1. cholesterol clogging up your heart's blood supply
2. a heart attack in the past
3. enlargement of one side of the heart
4. abnormal heart rhythms

AD8232 ECG Sensor

This sensor is a cost-effective board used to measure the electrical activity of the heart. This electrical activity can be charted as an ECG or Electrocardiogram and output as an analog reading. ECGs can be extremely noisy, the AD8232 Single Lead Heart Rate Monitor acts as an op amp to help obtain a clear signal from the PR and QT Intervals easily.

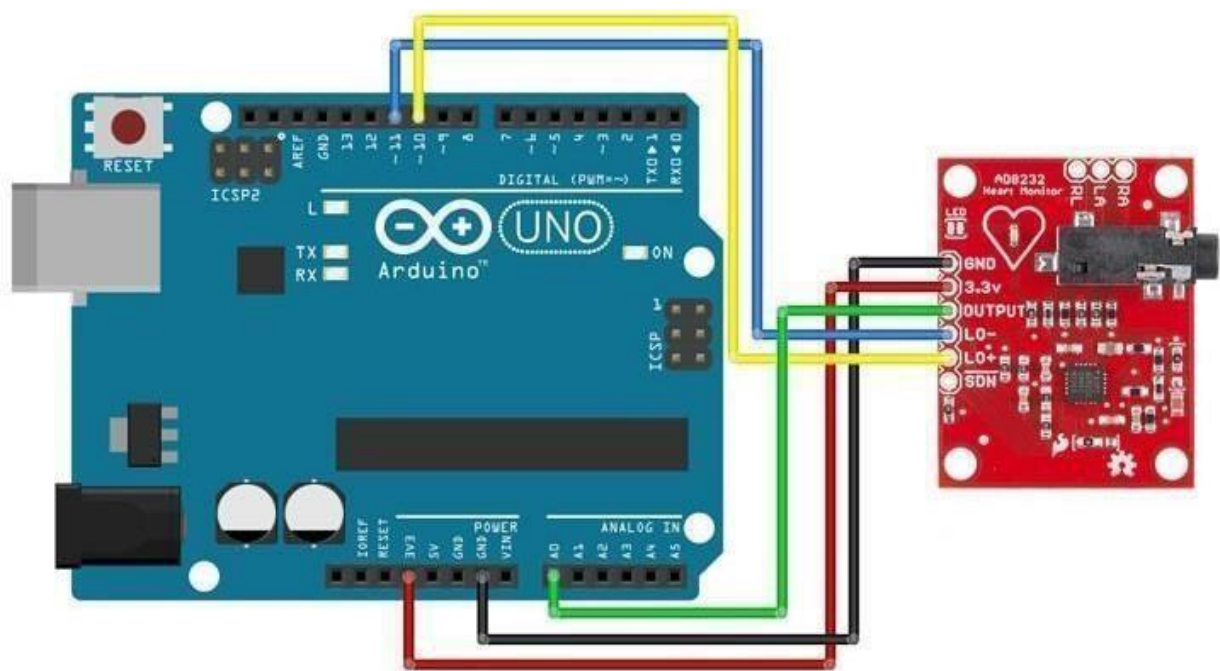


The AD8232 is an integrated signal conditioning block for ECG and other biopotential measurement applications. It is designed to extract, amplify, and filter small biopotential signals in the presence of noisy conditions, such as those created by motion or remote electrode placement.

The AD8232 module breaks out nine connections from the IC that you can solder pins, wires, or other connectors to. SDN, LO+, LO-, OUTPUT, 3.3V, GND provide essential pins for operating this monitor with an Arduino or other development board. Also provided on this board are RA (Right Arm), LA (Left Arm), and RL (Right Leg) pins to attach and use your own custom sensors. Additionally, there is an LED indicator light that will pulsate to the rhythm of a heart beat.

Circuit Diagram/Connection between Arduino and ECG Sensor AD8232

The AD8232 Heart Rate Monitor breaks out nine connections from the IC. We traditionally call these connections “pins” because they come from the pins on the IC, but they are actually holes that you can solder wires or header pins to.

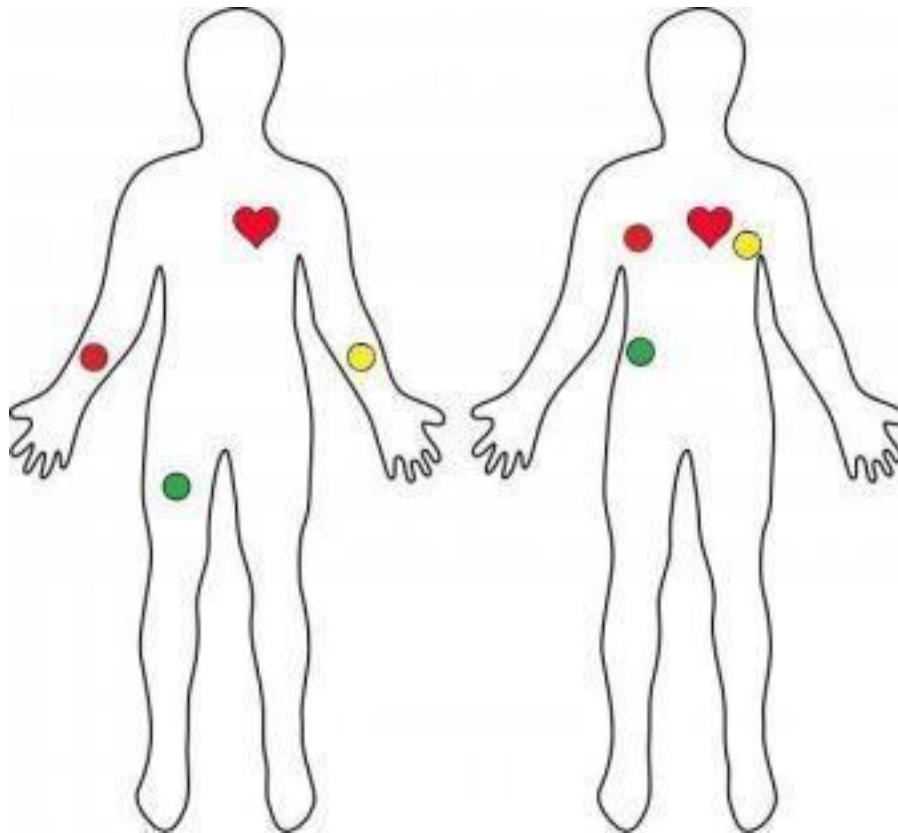


We'll connect five of the nine pins on the board to Arduino. The five pins you need are labeled GND, 3.3v, OUTPUT, LO-, and LO+.

Board Label	Pin Function	Arduino Connection
GND	Ground	GND
3.3v	3.3v Power Supply	3.3v
OUTPUT	Output Signal	A0
LO-	Leads-off Detect -	11
LO+	Leads-off Detect +	10
SDN	Shutdown	Not used

AD8232 ECG Sensor Placement on Body

It is recommended to snap the sensor pads on the leads before application to the body. The closer to the heart the pads are, the better the measurement. The cables are color coded to help identify proper placement.



Red: RA (Right Arm)

Yellow: LA (Left Arm)

Green: RL (Right

Leg) **Features:**

- Operating Voltage - 3.3V
- Analog Output
- Leads-Off Detection
- Shutdown Pin
- LED Indicator
- 3.5mm Jack for Biomedical Pad Connection or Use 3 pin header

BP sensor

Our final project is to design and build a portable blood pressure monitor device that can measure a user's blood pressures and heart rate through an inflatable hand cuff. The device is consisted of three main parts: external hardware's, analog circuit, and microcontroller. The analog circuit converts the pressure value inside the cuff into readable and usable analog waveforms. The MCU samples the waveforms and performs A/D conversion so that further calculations can be made.

It is undeniable that nowadays people are more aware of the health conditions. One of the most widely used methods to test the health conditions of an individual is to measure his/her blood pressures and heart rate. We, as ones of those who are concerned about their health, decided to work on this subject matter because we would like to build something that is useful and useable in real life.

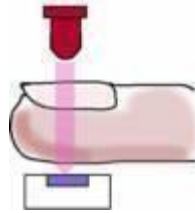
Design

1) How blood pressures are measured

Usually when the doctor measures the patient's blood pressure, he will pump the air into the cuff and use the stethoscope to listen to the sounds of the blood in the artery of the patient's arm. At the start, the air is pumped to be above the systolic value. At this point, the doctor will hear nothing through the stethoscope. After the pressure is released gradually, at some point, the doctor will begin to hear the sound of the heart beats. At this point, the pressure in the cuff corresponds to the systolic pressure. After the pressure decreases further, the doctor will continue hearing the sound. And at some point, the sounds will begin to disappear. At this point, the pressure in the cuff corresponds to the diastolic pressure.

The air will be pumped into the cuff to be around 20 mmHg above average systolic pressure (about 120 mmHg for an average). After that the air will be slowly released from the cuff causing the pressure in the cuff to decrease. As the cuff is slowly deflated, we will be measuring the tiny oscillation in the air pressure of the arm cuff. The systolic pressure will be the pressure at which the pulsation starts to occur. We will use the MCU to detect the point at which this oscillation happens and then record the pressure in the cuff. Then the pressure in the cuff will decrease further. The diastolic pressure will be taken at the point in which the oscillation starts to disappear.

2) How Heart beat are measured



The skin may be illuminated with visible (red) or infrared LEDs using transmitted or reflected light for detection. The very small changes in reflectivity or in transmittance caused by the varying blood content of human tissue are almost invisible. Various noise sources may produce disturbance signals with amplitudes equal or even higher than the amplitude of the pulse signal. Valid pulse measurement therefore requires extensive preprocessing of the raw signal.

The new signal processing approach presented here combines analog and digital signal processing in a way that both parts can be kept simple but in combination are very effective in suppressing disturbance signals.

The setup described here uses a red LED for transmitted light illumination and a pin Photodiode as detector. With only slight changes in the preamplifier circuit the same hard- and software could be used with other illumination and detection concepts. The detectors photo current (AC Part) is converted to voltage and amplified by an inexpensive operational amplifier (LM358). Amplification is combined with simple 2nd order analog filtering. A PIC16F877 microcontroller converts the analog signal with 10 bits resolution to a digital signal. An average is calculated from 250 readings taken over a 20 milliseconds period (This equals one period of the european power line frequency of 50 Hz).

When the heart beats, a pressure wave moves out along the arteries at a few metres per second (appreciably faster than the blood actually flows). This pressure wave can be felt at the wrist, but it also causes an increase in the blood volume in the tissues, which can be detected by a plethysmograph. This word comes from the Greek "plethysmos" for increase and is a term for a "fullness" (ie change in volume) measuring device. Over the years, all sorts of Heath- Robinson devices have been used but described here is a photoelectric pulse plethysmograph, which is robust and easy to make and which will allow the beating of the heart to be recorded without the need to make direct electrical connections to the body.

Sensor

The sensor consists of a light source and photodetector; light is shone through the tissues and variation in blood volume alters the amount of light falling on the detector. The source and detector can be mounted side by side to look at changes in reflected light or on either side of a finger or earlobe to detect changes in transmitted light. The particular arrangement here uses a wooden clothes peg to hold an infra red light emitting diode and a matched phototransistor (see figures 1 and 2). The infra red filter of the phototransistor reduces interference from fluorescent lights, which have a large AC component in their output.



Figure 1: Sensor

The peg is drilled with 3mm holes to take the led, the phototransistor, the pair of wires linking the two and the 2-core screened output cable. The holes for the led and phototransistor are drilled in one go so that they line up. The ends of each side of the peg are filed on the inside to enlarge the gap and pieces of black closed-cell foam (cannibalized from a mouse mat and punched with 3mm holes) are stuck in place (Super Glue / Crazy Glue) to improve grip and make a (more or less) light-tight seal against the skin. At this point, the spring should be adjusted so that the peg will grip an ear lobe while at the same time not being so tight that it excludes blood from a finger. Pieces of strip-board glued to the peg are used to make connections to the wires; two copper strips wide for the led (anode and cathode connections) and three for the phototransistor side (collector, emitter and led cathode, and led anode - led wires coming through the peg from the other side). The light-emitting diode and the phototransistor are wedged in their holes and soldered to their respective pieces of strip board. Neither component is critical and many other types will work. The wires are then soldered in place; the screen of the connecting lead is soldered to the emitter and cathode copper pad. Once everything has been checked and proved to work, the connections and the backs of the components should be covered with a bead of an opaque silicone rubber caulk which will insulate and keep out extraneous light. All bare wires should be covered.

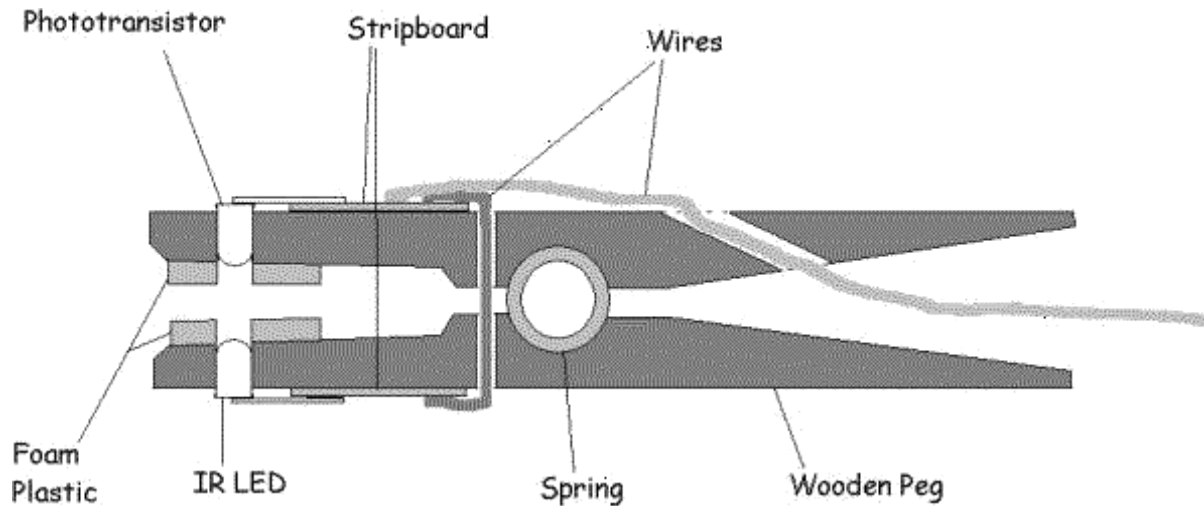
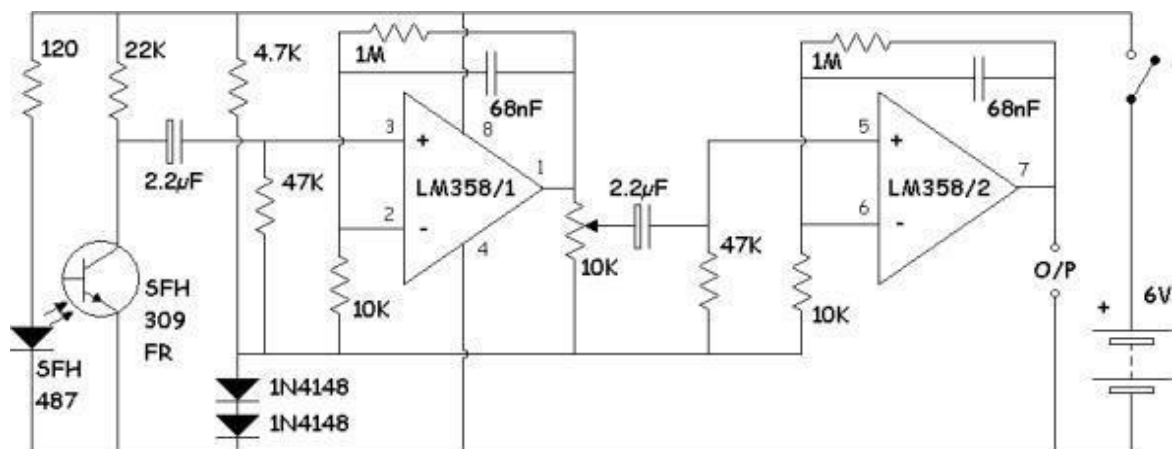


Figure 2: Constructing the sensor

Amplifier

The amplifier (see figure 3) uses an LM358 dual op amp to provide two identical broadly-tuned band pass stages with gains of 100. Again, the type of op amp is not particularly critical, as long as it will work at 6V and drive the output rail to rail. The signal frequencies are boxed in by movement artefacts at the low end (generated by the peg moving and distorting the underlying tissues; light pegs are better) and at the top end by mains-hum interference. The circuit runs from a single 6 Volt battery and the output zero is offset by about 1 Volt by referring everything to an internal common line at a voltage set by a pair of forward-biased silicon diodes. This is convenient for interfaces with a 0-5Volt input. The potentiometer allows the overall gain to be adjusted so as to prevent clipping on large signals. Components are not critical but the two 2.2 μ F capacitors must be able to stand some reverse bias so they should be non-polarized or tantalum. The circuit can easily be made up on a small piece of strip board.



Pulse sensor

The Heartbeat rate information knowing is very useful while doing exercise, studying, etc. But, the heartbeat rate can be complicated to calculate. To overcome this problem, the pulse sensor or [heartbeat sensor](#) is used. This is a plug & play sensor mainly designed for [Arduino board](#) which can be used by makers, students, developers, artists who can utilize the heartbeat information into their projects. This sensor uses an easy optical pulse sensor along with amplification & cancellation of noise to make a circuit. By using this circuit, we can get fast and reliable heartbeat readings.

What do you mean by Heartbeat?

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 photo phlethysmography. 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).

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.

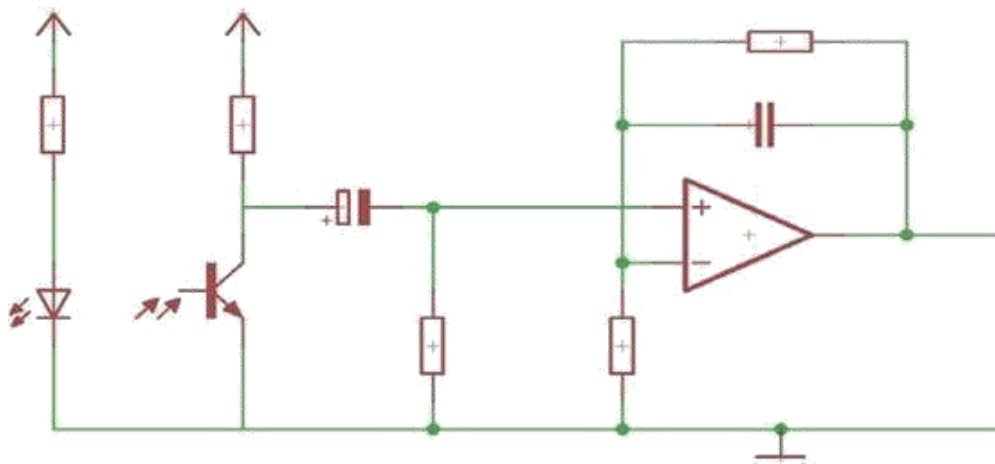
Reflection: Light emitted from the light emitting device is reflected by the regions.



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.

Features

- Biometric Pulse Rate or Heart Rate detecting sensor
- Plug and Play type sensor
- Operating Voltage: +5V or +3.3V
- Current Consumption: 4mA
- Inbuilt Amplification and Noise cancellation circuit.
- Diameter: 0.625"
- Thickness: 0.125" Thick
-

Pin Configuration

Pin Number	Pin Name	Wire Colour	Description
1	Ground	Black	Connected to the ground of the system

2	Vcc	Red	Connect to +5V or +3.3V supply voltage
3	Signal	Purple	Pulsating output signal.

Applications

- Sleep Tracking
- Anxiety monitoring
- Remote patient monitoring/alarm system
- Health bands
- Advanced gaming consoles

Software Modules:

- Arduino-IDE
- Embedded-‘C’

Software explanation

Arduino Software (IDE)

• Writing Sketches

- File
- Edit
- Sketch
- Tools
- Help

- Sketchbook
- Tabs, Multiple Files, and Compilation
- Uploading
- Libraries

Additional commands are found within the five menus: File, Edit, Sketch, Tools, Help. The menus are context sensitive, which means only those items relevant to the work currently being carried out are available.

File

- ☐ New
Creates a new instance of the editor, with the bare minimum structure of a sketch already in place.
- ☐ Open
Allows to load a sketch file browsing through the computer drives and folders.
- ☐ Open Recent Provides a short list of the most recent sketches, ready to be opened.
- ☐ Sketchbook
Shows the current sketches within the sketchbook folder structure; clicking on any name opens the corresponding sketch in a new editor instance. Examples
- ☐ Any example provided by the Arduino Software (IDE) or library shows up in this menu item. All the examples are structured in a tree that allows easy access by topic or library.
- ☐ Close
Closes the instance of the Arduino Software from which it is clicked.
- ☐ Save
Saves the sketch with the current name. If the file hasn't been named before, a name will be provided in a "Save as.." window.
- ☐ Save as...
Allows to save the current sketch with a different name.
- ☐ Page Setup It shows the Page Setup window for printing. Print
- ☐ Sends the current sketch to the printer according to the settings defined in Page Setup.
- ☐ Preferences
Opens the Preferences window where some settings of the IDE may be customized, as the language of the IDE interface.
- ☐ Quit
Closes all IDE windows. The same sketches open when Quit was chosen will be automatically reopened the next time you start the IDE.

Edit

- ☐ Undo/Redo
Goes back of one or more steps you did while editing; when you go back, you may go forward with Redo.
- ☐ Cut
Removes the selected text from the editor and places it into the clipboard.
- ☐ Copy
Duplicates the selected text in the editor and places it into the clipboard.
- ☐ Copy for
Forum Copies the code of your sketch to the clipboard in a form suitable for posting to the forum, complete with syntax coloring.
- ☐ Copy as HTML
Copies the code of your sketch to the clipboard as HTML, suitable for embedding in web pages.
- ☐ Paste
Puts the contents of the clipboard at the cursor position, in the editor.
- ☐ Select All
Selects and highlights the whole content of the editor.
- ☐ Comment/Uncomment
Puts or removes the // comment marker at the beginning of each selected line.
- ☐ Increase/Decrease Indent
Adds or subtracts a space at the beginning of each selected line, moving the text one space on the right or eliminating a space at the beginning. Find
- ☐
Opens the Find and Replace window where you can specify text to search inside the current sketch according to several options.
- ☐ Find Next Highlights the next occurrence - if any - of the string specified as the search item in the Find window, relative to the cursor position.
- ☐ Find Previous Highlights the previous occurrence - if any - of the string specified as the search item in the Find window relative to the cursor position.

Sketch

- ☐ Verify/Compile
Checks your sketch for errors compiling it; it will report memory usage for code and variables in the console area.
- ☐ Upload
Compiles and loads the binary file onto the configured board through the configured Port.
- ☐ Upload Using
Programmer This will overwrite the bootloader on the board; you will need to use Tools > Burn Bootloader to restore it and be able to Upload to USB serial port again. However, it allows you to use the full capacity of the Flash memory for your sketch.

- ☐ **Export Compiled Binary** Saves a .hex file that may be kept as archive or sent to the board using other tools.
- ☐ **Show Sketch Folder** Opens the current sketch folder.
- ☐ **Include Library**
Adds a library to your sketch by inserting #include statements at the start of your code. For more details, see libraries below. Additionally, from this menu item you can access the Library Manager and import new libraries from .zip files.
- ☐ **Add File...**
Adds a source file to the sketch (it will be copied from its current location). The new file appears in a new tab in the sketch window. Files can be removed from the sketch using the tab menu accessible clicking on the small triangle icon below the serial monitor one on the right side of the toolbar.

Tools

- ☐ **Auto Format**
This formats your code nicely: i.e. indents it so that opening and closing curly braces line up, and that the statements inside curly braces are indented more.
- ☐ **Archive Sketch** Archives a copy of the current sketch in .zip format. The archive is placed in the same directory as the sketch.
- ☐ **Fix Encoding &**
Reload Fixes possible discrepancies between the editor char map encoding and other operating systems char maps.
- ☐ **Serial Monitor**
Opens the serial monitor window and initiates the exchange of data with any connected board on the currently selected Port. This usually resets the board, if the board supports Reset over serial port opening.
- ☐ **Board**
Select the board that you're using. See below for descriptions of the various boards.
- ☐ **Port**
This menu contains all the serial devices (real or virtual) on your machine. It should automatically refresh every time you open the top-level tools menu.
- ☐ **Programmer**
For selecting a hardware programmer when programming a board or chip and not using the onboard USB-serial connection. Normally you won't need this, but if you're burning a bootloader to a new microcontroller, you will use this.
- ☐ **Burn Bootloader**
The items in this menu allow you to burn a bootloader onto the microcontroller on an Arduino board. This is not required for normal use of an Arduino or Genuino board but is useful if you purchase a new ATmega microcontroller (which normally come without a bootloader). Ensure that you've selected the correct board from the Boards menu before burning the bootloader on the target board. This command also sets the right fuses.

Help

Here you find easy access to a number of documents that come with the Arduino Software (IDE). You have access to Getting Started, Reference, this guide to the IDE and other documents locally, without an internet connection. The documents are a local copy of the online ones and may link back to our online website.

☐ Find in

Reference This is the only interactive function of the Help menu: it directly selects the relevant page in the local copy of the Reference for the function or command under the cursor.

Sketchbook

The Arduino Software (IDE) uses the concept of a sketchbook: a standard place to store your programs (or sketches). The sketches in your sketchbook can be opened from the File > Sketchbook menu or from the Open button on the toolbar. The first time you run the Arduino software, it will automatically create a directory for your sketchbook. You can view or change the location of the sketchbook location from with the Preferences dialog.

Beginning with version 1.0, files are saved with a .ino file extension. Previous versions use the .pde extension. You may still open .pde named files in version 1.0 and later, the software will automatically rename the extension to .ino.

Tabs, Multiple Files, and Compilation

Allows you to manage sketches with more than one file (each of which appears in its own tab). These can be normal Arduino code files (no visible extension), C files (.c extension), C++ files (.cpp), or header files (.h).

Uploading

Before uploading your sketch, you need to select the correct items from the Tools > Board and Tools > Port menus. The boards are described below. On the Mac, the serial port is probably something like /dev/tty.usbmodem241 (for an Uno or Mega2560 or Leonardo) or /dev/tty.usbserial-1B1 (for a Duemilanove or earlier USB board), or /dev/tty.USA19QW1b1P1.1 (for a serial board connected with a Keyspan USB-to-Serial adapter). On Windows, it's probably COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) - to find out, you look for USB serial device in the ports section of the Windows Device Manager. On Linux, it should be /dev/ttyACMx, /dev/ttyUSBx or similar. Once you've selected the correct serial port and board, press the upload button in the toolbar or select the Upload item from the Sketch menu. Current Arduino boards will reset automatically and begin the upload. With older boards (pre- Diecimila) that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is

uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error.

When you upload a sketch, you're using the Arduino bootloader, a small program that has been loaded on to the microcontroller on your board. It allows you to upload code without using any additional hardware. The bootloader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The bootloader will blink the on-board (pin 13) LED when it starts (i.e. when the board resets).

Libraries

Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, select it from the Sketch > Import Library menu. This will insert one or more `#include` statements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the board with your sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its `#include` statements from the top of your code.

There is a list of libraries in the reference. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources or through the Library Manager. Starting with version 1.0.5 of the IDE, you can import a library from a zip file and use it in an open sketch. See these instructions for installing a third-party library.

To write your own library, see this tutorial.

Third-Party Hardware

Support for third-party hardware can be added to the hardware directory of your sketchbook directory. Platforms installed there may include board definitions (which appear in the board menu), core libraries, bootloaders, and programmer definitions. To install, create the hardware directory, then unzip the third-party platform into its own sub-directory. (Don't use "arduino" as the sub-directory name or you'll override the built-in Arduino platform.) To uninstall, simply delete its directory.

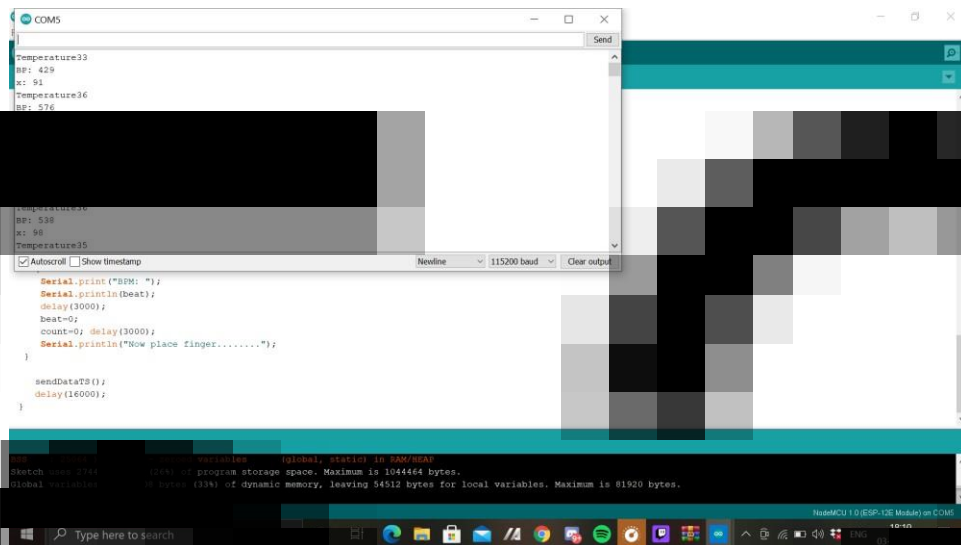
For details on creating packages for third-party hardware, see the Arduino IDE 1.5 3rd party Hardware specification.

Serial Monitor

Displays serial data being sent from the Arduino or Genuino board (USB or serial board). To send data to the board, enter text and click on the "send" button or press enter. Choose the baud rate from the drop-down that matches the rate passed to `Serial.begin` in your sketch. Note that on Windows, Mac or Linux, the Arduino or Genuino board will reset (rerun your sketch execution to the beginning) when you connect with the serial monitor.

CHAPTER 5

RESULTS



```
COM5
Temperature33
BP: 429
x: 91
Temperature36
BP: 576

Temperature35
BP: 538
x: 98

Autoscroll Show timestamp
Newline 115200 baud Clear output

Serial.print("BPM: ");
Serial.println(beat);
delay(3000);
beat=0;
count=0; delay(3000);
Serial.println("Now place finger.....");
}

sendDataTS();
delay(16000);
}
```

Sketch uses 2734 bytes (33% of program storage space. Maximum is 1044464 bytes.
Global variables: 0 bytes (0% of dynamic memory, leaving 54512 bytes for local variables. Maximum is 61920 bytes.

Arduino IDE interface showing the serial monitor and code editor. The serial monitor displays the output of the program, which includes temperature and heart rate data. The code editor shows the Arduino sketch, which includes the serial print statements and the ThingSpeak API calls.

Data from sensors is uploaded to ThingSpeak periodically without any interruption if the internet is available. Here ESP8266 Wi-Fi module is used for connecting Arduino to the internet. Healthcare monitoring system in hospitals and many other health centers has experienced significant growth, and portable healthcare monitoring systems with emerging technologies are becoming of great concern to many countries worldwide nowadays.

Using thing speak, we are able to get various health parameters in Realtime which is also saved in cloud for further reference. The above figure is the output obtained from thing speak which can be used for further analysis.

Thing Speak is an open-source Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP and MQTT protocol over the Internet or via a Local Area Network. ThingSpeak enables the creation of sensor logging applications, location tracking applications, and a social network of things with status updates

Boards

The board selection has two effects: it sets the parameters (e.g. CPU speed and baud rate) used when compiling and uploading sketches; and sets the file and fuse settings used by the burn bootloader command. Some of the board definitions differ only in the latter, so even if you've been uploading successfully with a particular selection you'll want to check it before burning the bootloader. You can find a comparison table between the various boards [here](#).

Arduino Software (IDE) includes the built in support for the boards in the following list, all based on the AVR Core. The Boards Manager included in the standard installation allows to add support for the growing number of new boards based on different cores like Arduino Due, Arduino Zero, Edison, Galileo and so on.

- ☐ Arduino Yùn
An ATmega32u4 running at 16 MHz with auto-reset, 12 Analog In, 20 Digital I/O and 7 PWM.
- ☐ Arduino/Genuino Uno
An ATmega328 running at 16 MHz with auto-reset, 6 Analog In, 14 Digital I/O and 6 PWM.
- ☐ Arduino Diecimila or Duemilanove w/
ATmega168 An ATmega168 running at 16
MHz with auto-reset.
- ☐ Arduino Nano w/ ATmega328
An ATmega328 running at 16 MHz with auto-reset. Has eight analog
- ☐ inputs. Arduino/Genuino Mega 2560
An ATmega2560 running at 16 MHz with auto-reset, 16 Analog In, 54 Digital I/O and 15 PWM.
- ☐ Arduino Mega
An ATmega1280 running at 16 MHz with auto-reset, 16 Analog In, 54 Digital I/O and 15 PWM.
- ☐ Arduino Mega ADK
An ATmega2560 running at 16 MHz with auto-reset, 16 Analog In, 54 Digital I/O and 15 PWM.
- ☐ Arduino Leonardo
An ATmega32u4 running at 16 MHz with auto-reset, 12 Analog In, 20 Digital I/O and 7 PWM.
- ☐ Arduino/Genuino Micro
An ATmega32u4 running at 16 MHz with auto-reset, 12 Analog In, 20 Digital I/O and 7 PWM.

- Arduino Ethernet
Equivalent to Arduino UNO with an Ethernet shield: An ATmega328 running at 16 MHz with auto-reset, 6 Analog In, 14 Digital I/O and 6 PWM.
- Arduino Fio
An ATmega328 running at 8 MHz with auto-reset. Equivalent to Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328, 6 Analog In, 14 Digital I/O and 6 PWM.
- Arduino BT w/ ATmega328
ATmega328 running at 16 MHz. The bootloader burned (4 KB) includes codes to initialize the on-board bluetooth module, 6 Analog In, 14 Digital I/O and 6 PWM..
- LilyPad Arduino USB
An ATmega32u4 running at 8 MHz with auto-reset, 4 Analog In, 9 Digital I/O and 4 PWM.
- LilyPad Arduino
An ATmega168 or ATmega132 running at 8 MHz with auto-reset, 6 Analog In, 14 Digital I/O and 6 PWM.
- Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328
- An ATmega328 running at 16 MHz with auto-reset. Equivalent to Arduino Duemilanove or Nano w/ ATmega328; 6 Analog In, 14 Digital I/O and 6 PWM. ArduinoNG or older w/ ATmega168

CHAPTER 6

CONCLUSION

The Internet of Things is considered now as one of the feasible solutions for any remote value tracking especially in the field of health monitoring. It facilitates that the individual prosperity parameter data is secured inside the cloud, stays in the hospital are reduced for conventional routine examinations and most important that the health can be monitored and disease diagnosed by any doctor at any distance. In this paper, an IoT based health monitoring system was developed. The system monitored body temperature, pulse rate and room humidity and temperature using sensors. These sensor values are then sent to a medical server using wireless communication. These data are then received in an authorized personals smart phone with IoT platform. With the values received the doctor then diagnose the disease and the state of health of the patient.

References:

- [1] Abasi Julius and Zhang Jian-Min, "IoT Based Patient Health Monitoring System Using LabVIEW and Wireless Sensor Network", published on the International Journal of Science and Research (IJSR), ISSN (Online): 2319-7064, Volume 6, Issue 3, March 2017
- [2] Shivam Gupta, Shivam Kashaudhan, Devesh Chandra Pandey, Prakhar Pratap Singh Gaur, "IoT based Patient Health Monitoring System", published on the International Research Journal of Engineering and Technology (IRJET), ISSN: 2395-0072, Volume 4, Issue 3, March 2017
- [3] Moeen Hassanaliereagh, Alex Page, Tolga Soyata, Gaurav Sharma, Mehmet Aktas, Gonzalo Mateos, Burak Kantarci, Silvana Andreescu, "Health Monitoring and Management Using Internet-of-Things (IoT) Sensing with Cloud-based Processing: Opportunities and Challenges", published on 2015 IEEE International Conference on Services Computing, 978-1-4673-7281-7/15 © 2015 IEEE, DOI 10.1109/SCC.2015.47
- [4] Ahmed Abdelgawad, Kumar Yelamarthi and Ahmed Khattab, "IoT Based Health Monitoring System for Active and Assisted Living", published by ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2017, O. Gaggi et al. (Eds.): GOODTECHS 2016, LNICST 195, pp. 11–20, 2017, DOI: 10.1007/978-3-319-61949-1_2
- [5] Hemalatha Rallapalli and Pavani Bethelli, "IoT Based Patient Monitoring System", published on Int'l Journal of Computing, Communications & Instrumentation Engg. (IJCCIE) Vol. 4, Issue 1 (2017) ISSN 2349-1469 EISSN 2349-1477
- [6] Jorge Gómez, Byron Oviedo and Emilio Zhumab, "Patient Monitoring System Based on Internet of Things", published on the 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016)
- [7] T.Saraswathi and S.Amutha, "IoT Based Wireless Healthcare Monitoring", published on International Journal of Emerging Trends in Science and Technology (IJETST) Vol. 03, Issue 05, Pages 784-792, May 2016, ISSN 2348-9480

- [8] Mobyen Uddin Ahmed, Mats Björkman, Aida Ćaušević, Hossein Fotouhi and Maria Lindén, "An Overview on the Internet of Things for Health Monitoring Systems". In: Mandler B. et al. (eds) Internet of Things. IoT Infrastructures. IoT360 2015. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 169. Springer,
- [9] Dalyah Y. Al-Jamal, Maryam H. Eshtaiwi and Liyakathunisa Syed, "IoT Based Process Model for Heart Monitoring Process", published on International Journal of Computer and Systems Engineering Vol. 11, No. 6, 2017
- [10] Mingzhe Jiang, Tuan Nguyen Gia, Arman Anzanpour, Amir-Mohammad Rahmani, Tomi Westerlund, Sanna Salanterä, Pasi Liljeberg and Hannu Tenhunen, "IoT-based Remote Facial Expression Monitoring System with sEMG Signal", published on 2016 IEEE Sensors Applications Symposium (SAS), 978-1-4799-7250-0/16 ©2016 IEEE
- [11] N. Ebrahim, M. J. Deen and T. Mondal, "A wireless wearable ECG sensor for long- term applications," *IEEE Commun. Mag.*, vol. 50, no. 1, pp. 36–43, Jan. 2012.
- [12] J. M. Cano-Garcia, E. Gonzalez-Parada, V. Alarcon-Collantes and E. Casilari-Perez, "A PDA-based portable wireless ECG monitor for medical personal area networks," in *Proc. MELECON*, 2006, pp. 713–716.
- [13] M. Hong, Z. Yajun and H. Xiaoping, "Portable ECG measurement device based on MSP430 MCU," in *Proc. BMEI*, 2008, pp. 667–671.
- [14] Lin X, Mahmud S, Jones E, Shaker A, Miskinis A, Kanan S, Kim J-H. Virtual reality-based musical therapy for mental health management. In: 2020 10th Annual computing and communication workshop and conference (CCWC). IEEE; 2020. p. 948–52.
- [15] Santoso D, Dalu Setiaji F. Non-contact portable infrared thermometer for rapid influenza screening. In: 2015 International conference on automation, cognitive science, optics, micro electro-mechanical system, and information technology (ICACOMIT). IEEE; 2015. p. 18–23.