A project report on

IoT Based Health Monitoring System

Submitted in complete fulfillment of the requirements of

Bachelor of Engineering

in

Electronics and Telecommunication Engineering

by

Diksha Bhosale	**(3021159)**
Pratik Belamkar	**(3021107)**
Varun Somwanshi	**(3021151)**
Mudabbir Ahmad	**(3021133)**

Under the guidance of

Ms. Akshata Raut

Department of Electronics and Telecommunication Engineering
Fr. C. Rodrigues Institute of Technology, Vashi

University Of Mumbai

2024-2025

BONAFIDE CERTIFICATE

This is to certify that the project titled **IoT based Health Monitoring System** is a bonafide record of the work done by

Diksha Bhosale

Pratik Belamkar

Varun Somwanshi

Mudabbir Ahmad

submitted to the University of Mumbai in partial fulfillment of the requirement for the award confirming completion of **Major Project (REV- 2019 'C' Scheme) of Final Year, (BE Sem-VIII)** in **Electronics and Telecommunication Engineering** as laid down by **University of Mumbai** , during the **academic year of 2024-2025**

INTERNAL EXAMINER **EXTERNAL EXAMINER**

PROJECT GUIDE **HEAD OF THE DEPARTMENT**

(Akshta Raut) (Dr.Megha Kolhekar)

DECLARATION

We thus certify that the ideas included in this written contribution are entirely original with proper citations and references to any original sources used. Additionally, we affirm that we have followed all guidelines for academic honesty and integrity and that none of the ideas, data, facts, or sources in my work have been misrepresented, created, or falsified. We are aware that any infraction of the aforementioned will result in disciplinary action from the institute, which may even take legal action against sources that have not been properly referenced or from whom appropriate authorization has not been obtained where necessary.

Diksha Bhosale (3021159)	Signature:_______________
Pratik Belamkar (3021107)	Signature:_______________
Varun Somwanshi (3021151)	Signature:_______________
Mudabbir Ahmad (3021133)	Signature:_______________

Date:

ABSTRACT

The work that has been proposed and completed on the ”IoT-Based Patient Health Monitoring System” project is described in this paper. The primary objective of this project is to create a dependable, real-time health monitoring system that measures and analyzes critical health indicators, such as heart rate and blood oxygen saturation (SpO_2), using Internet of Things (IoT) technologies. The device detects changes in blood volume using visible and infrared light LEDs in combination with non-invasive photoplethysmography (PPG) techniques. The collected data is processed by a microcontroller before being wirelessly transmitted to a cloud-based platform for remote monitoring and analysis.

The implemented system consists of Internet of Things networking modules, sensors, and microcontroller-based processing. The hardware architecture includes a photodiode sensor, ordinary and infrared LEDs, and a signal processing unit to gather and clean the data. The software design ensures that health metrics are securely uploaded to the cloud, allowing healthcare professionals to access them from any place. The technology can generate warnings in the event of abnormal readings, ensuring that medical emergencies are handled quickly. This innovation would be especially helpful to patients who are elderly, have chronic conditions, or reside in remote areas with little access to hospitals.

The results of the system’s simulation demonstrate the effectiveness of PPG signals for real-time surveillance. SpO_2 and heart rate estimates were transmitted over the network with little latency. The proposed system showed reliable accuracy, cost, and portability when compared to traditional monitoring equipment. The cloud-based dashboard provided an intuitive user interface for real-time patient data display, and data logs were stored for future analysis and review.

When everything is taken into account, the ”IoT-Based Patient Health Monitoring System” offers a practical and expandable method of ongoing patient monitoring. The technology helps bridge the gap between patients and healthcare providers by enabling proactive treatment and decision-making based on real-time health data. It lays the foundation for future advancements such as the incorporation of electrocardiogram (ECG) or body temperature sensors and the use of machine learning to predictive healthcare analytics. The proposed paradigm is a significant step toward developing intelligent healthcare solutions in the age of digital transformation.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

ADC Analog To Digital Converter

DDR Double Data Rate

DSP Digital Signal Processor

FPGA Field Programming Gate Array

HDL Hardware Description Language

JTAG Joint Test Action Group

LED Light Emitting Diode

PCLK Pixel Clock

PLL Phase Lock Loop

PWM Pulse Width Modulation

SCCB Serial Camera Control Bus

SFM Structure From Motion

UART Universal Asynchronous Receiver Transmitter

VGA Video Graphics Array

XCLK External Clock

XDC Xilinx Design Constraint file

WNS Worst Negative Slack

CHAPTER 1

Introduction

In this study, the ”IoT-Based Patient Health Monitoring System” project is thoroughly and methodically reviewed. This includes the justification of the study, a review of prior similar work, the suggested technique, system implementation, simulation results, comparative analysis, and all other important components of the research. For scholars and students researching related subjects in the fields of healthcare technology and the Internet of Things (IoT), the study is a useful resource because it also describes the system’s possible uses and looks into future advancements.

1.1 Problem overview

Remote patient monitoring is being used more and more by the healthcare sector to enhance medical services, especially in cases where 24-hour care is necessary. Traditional health monitoring methods sometimes need large, cumbersome equipment and in-person evaluations, which are inappropriate for long-term or home-based monitoring. This causes a significant delay in the prompt delivery of healthcare, particularly for older patients, patients with chronic conditions, and people residing in remote or impoverished locations. Intelligent health monitoring systems that can continuously and non-invasively track vital parameters like blood oxygen saturation (SpO_2), body temperature, sodium, potassium, kidney creatine, blood pressure, sugar, heart rate, and blood pressure have been made possible by recent developments in the Internet of Things (IoT). These technologies give caretakers real-time notifications and facilitate cloud-based data sharing with medical specialists, allowing for remote diagnosis and prompt intervention.

Studies like [1] have demonstrated that when ESP-series microcontrollers (such the ESP8266) are interfaced with sensors like the MAX30100 and DHT11, real-time Wi-Fi transmission of health data is made possible. This makes healthcare more responsive and accessible by guaranteeing usability and accessibility across several sites.

There are still problems despite these technological developments, especially with data security, sensor calibration, power efficiency, and the use of reliable communication protocols.

With their portable and affordable designs, the two technologies in question aim to bridge the gap between patients and caregivers. Through system enhancement, real-time monitoring, and the addition of features like LabVIEW-based analysis and validation, this project seeks to answer the urgent demand for an IoT-based health monitoring platform that is fully integrated, secure, and clinically reliable..

1.2 Project Objectives

The main objectives of this project are:

1. Create a portable, affordable health monitoring gadget that tracks heart rate and SpO_2 using sensors like the MAX30100.
2. Connect the ESP8266 microcontroller for effective wireless communication, data processing, and collection. Send health information safely to a cloud-based platform using Internet of Things protocols such as Wi-Fi, MQTT, or HTTP.
3. Provide real-time patient data access to medical professionals and caregivers via an easy-to-use user interface (web or mobile application).
4. Install alarm systems to inform physicians or patients of any unusual health data.
5. Encourage remote health monitoring to help older or chronically ill people in rural areas become less dependent on hospitals.
6. Include LabVIEW as an extra platform for waveform visualization, offline signal processing, and real-time system verification.

1.3 Report outline

From conception to implementation and validation, each of the six comprehensive chapters that comprise the framework of this report covers a critical phase of the project.

Chapter 2: Literature Review Current studies on Internet of Things-based patient health monitoring systems are included in this chapter. With an emphasis on sensor integration, cloud connectivity, and data visualization, it condenses the design concepts and implementations of earlier models [1], [2]. Notable drawbacks include insufficient alert systems, restricted movement, and restricted parameter monitoring. The current project has been improved as a result of these deficiencies.

Chapter 3: Sensor Integration and Microcontroller System Design Key hardware elements including the ESP8266 microcontroller, the MAX30100 sensor, and more analog circuits are integrated in this chapter. The architecture facilitates PPG signal recording, amplification, digital processing, and wireless data transmission. There are discussions on design improvements over earlier efforts [3] along with schematics and diagrams.

Chapter 4: Interface for Cloud Monitoring and Communication The communication model utilized for safe MQTT or HTTP transmission of health data is covered in this chapter. The development of an online and mobile interface allows for the display of patient parameters in real time. In order to ensure system accuracy throughout the testing process, LabVIEW is also used to simulate various health conditions and verify signal integrity to guarantee system accuracy during testing. The system is compared with other platforms, such as the one proposed in [4].

Chapter 5: Analysis and Findings This chapter presents the system's accuracy, signal fidelity, latency, important parameters, and performance metrics. The readings are compared using standard clinical tools. LabVIEW is used for parameter display and offline waveform analysis to facilitate the evaluation. Comparative performance metrics from research [5] and [6] are used to benchmark the system.

Chapter 6: Overview and Future Directions This last chapter compiles the project's contributions, system effectiveness, and challenges encountered. Suggestions for future improvements include adding more sensors, improving energy efficiency, and using AI to diagnose patients more accurately. The potential contribution of LabVIEW to future research is also highlighted, especially in the domains of clinical-grade system development and testing [8].

CHAPTER 2

Literature Review

2.1 Introduction to Literature and Its Role in Research

The literature evaluation has a significant impact on the path of any research endeavor. It helps academics understand the past, look into technical advancements, and identify any gaps in the body of previous work. A comprehensive analysis of prior research offers valuable insights into the integration of sensors, communication technologies, real-time monitoring techniques, and cloud platforms in patient health monitoring for this IoT-based health monitoring system project. The evaluation also lays the groundwork for enhancing and creating a system that is more dependable, economical, and successful.

2.2 The Role of IoT in Environmental and Health Monitoring

By allowing objects to connect and operate independently, the Internet of Things (IoT) has greatly enhanced environmental and medical monitoring. In the context of health monitoring, IoT enables remote surveillance of vital indicators such as heart rate, oxygen saturation, temperature, and movement.Studies have demonstrated that wearable and embedded sensor systems, in conjunction with microcontrollers that enable Bluetooth or Wi-Fi, can help doctors and caregivers monitor patients' health in real time [1][2]. This approach is especially beneficial for older patients and those who live in remote areas with little access to conventional care.

2.3 Sensor Integration and Challenges

Sensor integration is a crucial part of IoT-based health systems. Sensors like the DHT11 (for temperature) and MAX30100 (for SpO_2 and heart rate) are often used in research prototypes [3][4]. Despite providing real-time data, these sensors often have problems, including inaccurate readings caused by ambient light or motion artifacts (especially in optical sensors), power consumption problems during continuous monitoring, and a lack of calibration against medical-grade equipment. Even though research has proposed methods such signal filtering and amplifier circuits [5], it is still challenging to integrate several sensors without increasing delay or signal loss.

2.4 Cloud Computing for IoT Data Management

Cloud computing enables the safe storage, remote access, and visualization of health data. Several studies [6], [7] use Firebase, ThingSpeak, and AWS IoT to send patient vitals from microcontrollers to the cloud. There have been applications for protocols such as HTTP and MQTT; MQTT has the benefit of low-latency and low-power communication [8]. However, problems with data security and privacy still exist. Many of the systems under review have insufficient authentication or encryption, putting sensitive health information at risk. Reliable internet connectivity is another crucial factor affecting system performance, especially in developing nations.

2.5 Real-Time Alert Mechanisms and Their Importance

One essential element of modern health monitoring systems is the inclusion of real-time alert mechanisms.These systems provide alerts via email, SMS, or mobile applications when a patient's vital signs meet certain standards. Studies have demonstrated that these alerts enhance patient outcomes by ensuring timely medical reactions [9]. However, certain systems may generate false alarms due to inaccurate sensor data or communication issues. Very few of the systems in use today employ complex filtering or decision-making algorithms before issuing alerts. Consequently, for more precise and dependable operation, smarter alert Systems that take a variety of health factors into account are necessary.

2.6 Identified Gaps and Future Directions

There are some gaps in the literature review. Many systems only monitor one or two health indicators, and few offer clinical-level accuracy or validation. Often, energy economy is overlooked while building portable gadgets. Most prototypes lack advanced signal processing, and alert systems are often very basic and prone to false positives.

2.6.1 Role of LabVIEW in Future Developments

LabVIEW is a tool for offline signal visualization and validation that has been used into some current research [10] to help close the gap between signal analysis and debugging. LabVIEW can: Display temperature readings and waveforms in real time; acquire real-time data from sensors via serial connectivity. Apply logic analysis and signal filtering techniques to improve the quality of your data.

CHAPTER 3

System Design

The Internet of Things-based patient health monitoring system's objective is to routinely assess vital signs including heart rate and blood oxygen saturation (SpO_2). from Figure 3.1 By combining sensors, microcontrollers, and cloud platforms, the system provides real-time monitoring and early identification of possible health problems, reducing delays in medical intervention[1],[4]. Data collected by the system is transmitted to cloud platforms such as AWS via the ESP8266 microcontroller, where it is accessible through web dashboards or mobile devices [3],[6]. Patients with chronic illnesses and the elderly benefit most from this technique. This arrangement facilitates better patient outcomes, ensures timely response, and reduces the burden on the healthcare system [8].

3.1 Wifi Connection

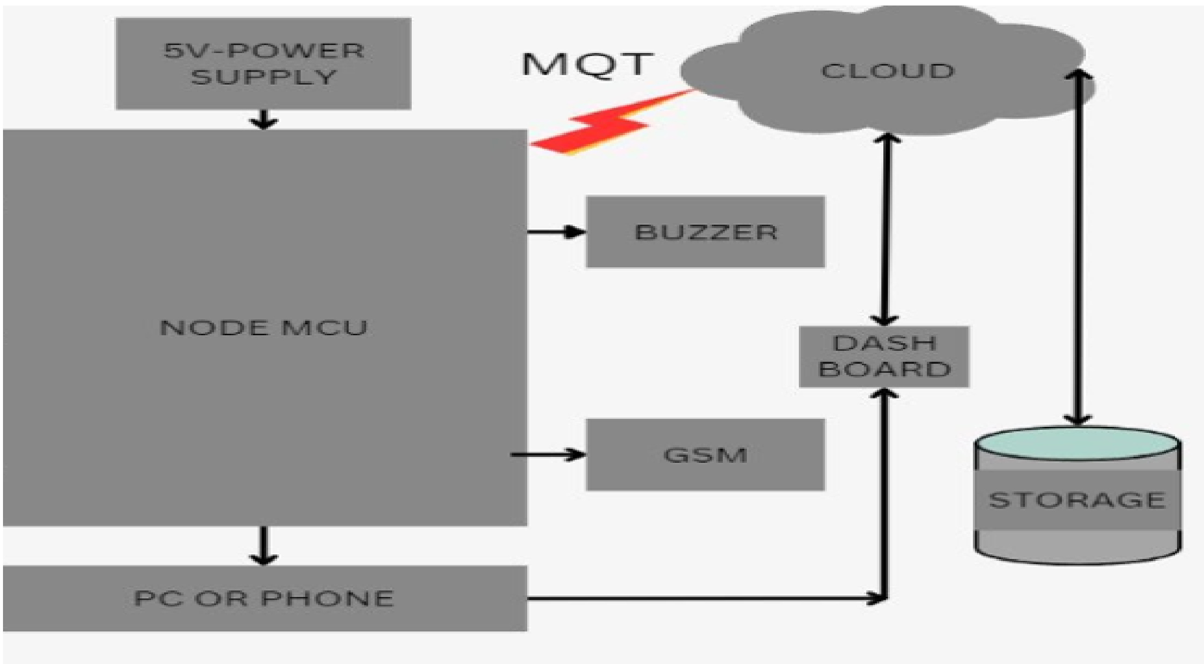


Figure 3.1: WiFi Connection

- **5V-POWER SUPPLY**: This block indicates the power source for the system, providing a stable 5V DC to the NodeMCU.

- **ESP8266**: The ESP8266 is a Wi-Fi-enabled microcontroller used to connect all sensors to the internet. It collects data from the sensors and sends it securely to AWS IoT Core for processing and visualization,

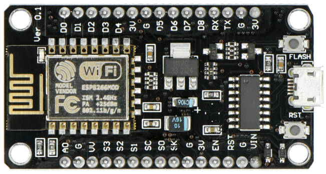


Figure 3.2: ESP8266

making it the central hub of communication and control in the system.

- **DHT11**: The DHT11 is a digital temperature and humidity sensor. It continuously monitors environmental conditions and provides real-time data to the ESP8266.

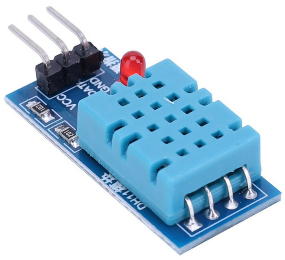


Figure 3.3: DHT11

Its role is crucial in health and environment-related monitoring applications due to its reliability and simplicity in integration.

- **MAX30100**: The MAX30100 is a pulse oximeter and heart-rate sensor. It detects heartbeats and measures blood oxygen saturation (SpO) levels. Information is transmitted to the

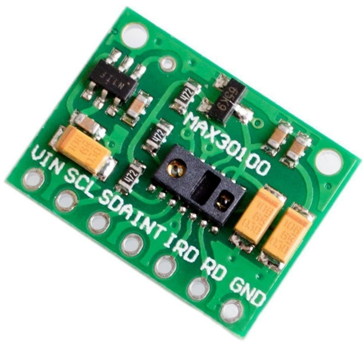


Figure 3.4: MAX30100

microcontroller, helping track vital health signs in real-time, making it ideal for patient or wellness monitoring.

- **Ultrasonic Sensor**: This sensor measures distance using ultrasonic waves. In this project, it checks the level of

Figure 3.5: Ultrasonic Sensor

saline or liquid, helping determine whether the container is full, normal, or near empty, which then triggers alerts as needed."

- **AWS IoT Core**: AWS IoT Core allows the ESP8266 to securely connect to cloud services. It facilitates the real-time transfer of data from the device to AWS, where it may be stored, shown, or analyzed further using other AWS services.

- **S**NS is used to send email alert alerts using the Simple Notification Service (Amazon SNS). SNS automatically alerts users or caregivers when sensor thresholds are surpassed (e.g., saline empty or aberrant vitals) to ensure timely response and action.

- **T**he Arduino IDE is the development environment used to create, compile, and upload code to the ESP8266. It supports all required libraries and provides serial monitoring for debugging and confirming sensor data and connection.

- **Amazon DynamoDB** Amazon DynamoDB is a fast and scalable NoSQL database solution from AWS. In this project, real-time health and sensor data, including temperature and humidity, heart rate, SpO_2, and liquid levels, are obtained from the ESP8266 via AWS IoT Core and saved in DynamoDB. It ensures efficient data logging, easy retrieval for analysis, and a seamless integration with other AWS services. Its serverless architecture and low latency make it ideal for handling frequent updates from IoT devices.

3.2 Architecture of Project

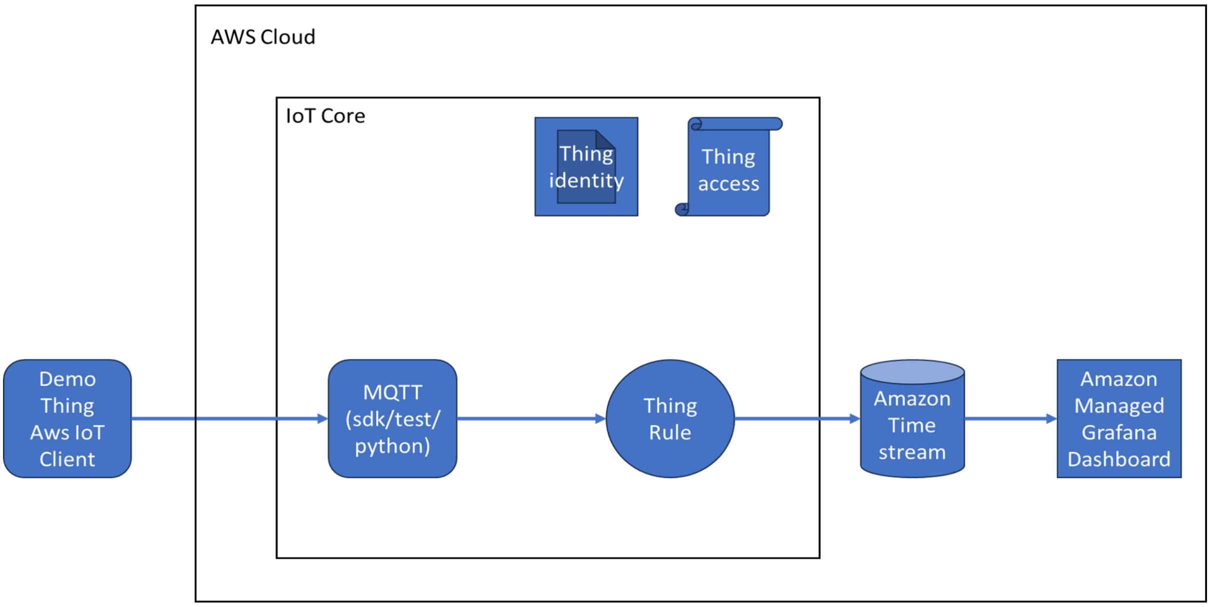


Figure 3.6: Architecture

Custom PCB The custom PCB used in this IoT-based healthcare monitoring system was designed using EasyEDA, a powerful online PCB design tool. The board integrates all the core components of the project, including the NodeMCU (ESP8266), which acts as the main controller for sensor interfacing and cloud communication. The HC-SR04 ultrasonic sensor,

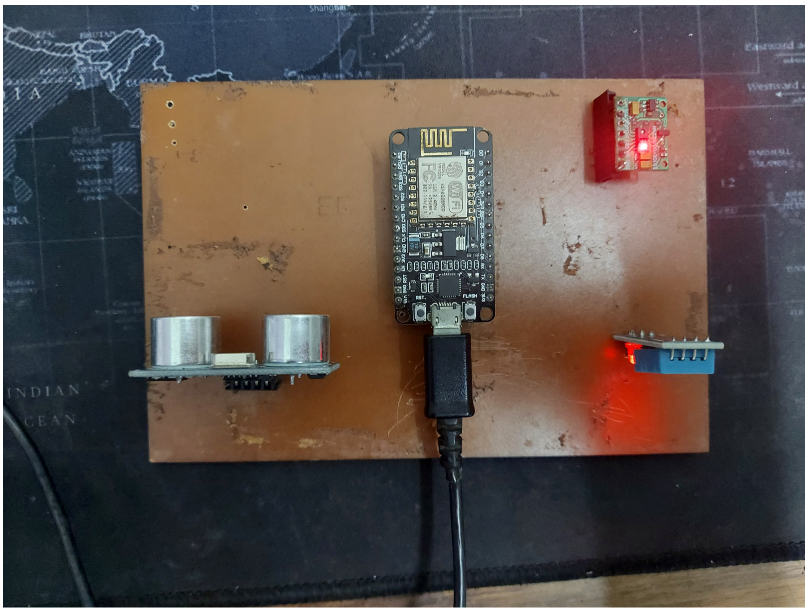


Figure 3.7: Custom PCB

positioned at the top, is used for fluid level detection. The DHT11 sensor, mounted on the lower left, measures ambient temperature and humidity. On the lower right, the MAX30100 pulse oximeter monitors heart rate and SpO_2 levels.

The PCB is powered via USB, supplying 5V to the NodeMCU, which then distributes power to the sensors. The compact and organized layout ensures efficient data collection, minimal wiring complexity, and stable connections. EasyEDA's design environment allowed for precise component placement and routing, making the board suitable for real-time applications. This PCB provides a reliable hardware foundation for cloud-connected health monitoring with AWS IoT Core and SNS integration.

,**Sample AWS IoT Client** This might be any Internet of Things device or emulator, such as a Raspberry Pi, virtual device, or Python script. It sends messages or sensor data by connecting to AWS IoT Core.

,**MQTT (SDK)** Beginning with MQTT is a lightweight communications protocol used by IoT devices. The client (your device) communicates with AWS IoT Core via an SDK (like the AWS IoT Device SDK), a test console, or a Python MQTT module.The IoT Rules Engine processes incoming MQTT messages. It filters, modifies, and routes data according to rules you provide. In this case, the rule transmits data to Amazon Timestream.

, **Amazon Timestream** A time series database allows for the long-term storage of temperature, humidity, and other data. created with consideration for time-stamped data. retains the sensor data obtained via IoT Rules.

,**Grafana Dashboard for Amazon Managed** Grafana is used to view the data. Amazon Managed Grafana, which displays dashboards in real time, is linked to Timestream. Top Section: Identity Access Identity of Thing: X.509 certificates or keys offer a unique identify for the device. Thing Access: IAM or IoT policies that outline what the Thing can do (such publishing to a topic).

3.3 AWS DataBase

Step 1: Circuit Configuration and Hardware Selection The ESP8266 NodeMCU microcontroller was selected due of its low power consumption and Wi-Fi functionality. Sensors in the system measure important variables, such as: Digital temperature and humidity sensor DHT11; SpO_2 and heart rate sensor MAX30100
Step 2: Calibration and Testing of Sensors The Arduino IDE was used to test the sensors. Water-based simulation was used to evaluate sensors based on temperature and fluid level. Appropriate calibration and noise filtering were employed to provide reliable results.
Step 3: Firmware Development Code was written using the Arduino IDE to collect sensor

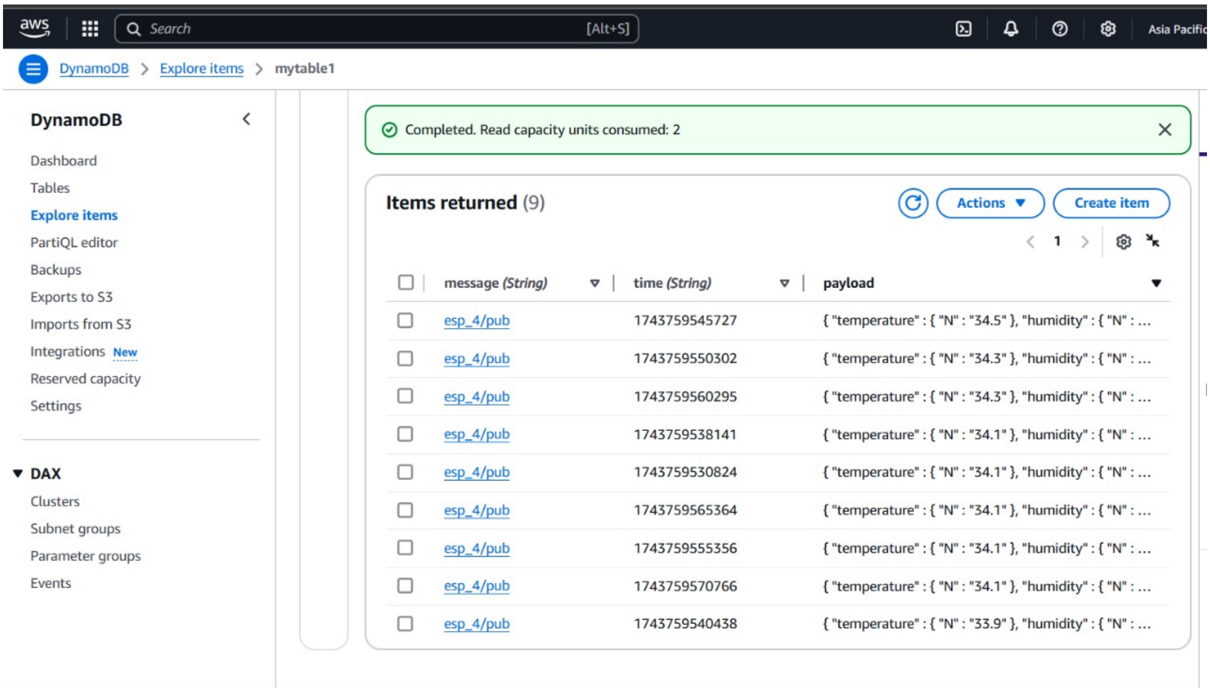


Figure 3.8: Dynamo DB

data, process it, and normalize the results. Connect to Wi-Fi and transfer data to the cloud. Step 4: Integration with Clouds in Real Time Data was sent to the ThingSpeak cloud platform using the HTTP protocol. Visual dashboards were used to set up data recording and alerts.

Step 5: Monitoring and Testing Several factors were used to test the entire system. Vital sign thresholds were set up to sound an alarm. Known values were used to validate test findings.

3.4 Labview

Step 1: Sensor Integration and Component Selection Important components include oxygen saturation, or SpO_2, a blood test indication. Temperature, blood pressure (BP), potassium, and salt levels The quantity of sugar The objective was to measure kidney function (creatinine). The proper sensors for each parameter were connected to a micro-controller (ESP32-CAM sans camera) on a PCB.

Step 2: Configuring Hardware Numerous analog and digital sensors are linked to the ESP32 microcontroller via jumper wires on a breadboard. Prior to being transmitted to a local interface or the cloud, real-time sensor data is gathered and processed.

Step 3: Calibration and Testing of Sensors Each sensor was inspected to guarantee accuracy and functionality. under carefully monitored conditions (for example, a temperature sensor immersed in water, as seen in Figure 3.3).

Step 4: LabVIEW Data Visualization LabVIEW was used to create a visual dashboard for monitoring sensor data in real time. The serial data from the ESP8266 was fed into LabVIEW using either the VISA protocol or serial communication blocks. Graphical indicators including plots, dials, and warnings were used to provide visual feedback.

Step 5: Alerts and Real-Time Monitoring The system continually tracks the patient's health indicators. If there are abnormal readings, real-time alerts are produced. This functionality may be expanded to include cloud-based logging or mobile alerts.

CHAPTER 4

Cloud Communication and Monitoring Interface

4.1 AWS IoT Core Device Messaging and Connectivity

AWS IoT Core provides the communication interface between your IoT sensors and the cloud. Devices like the ESP8266 broadcast sensor data (temperature, humidity, SpO, etc.) to AWS IoT Core via MQTT. Key attributes used: Device shadows for the virtual depiction of linked sensors.

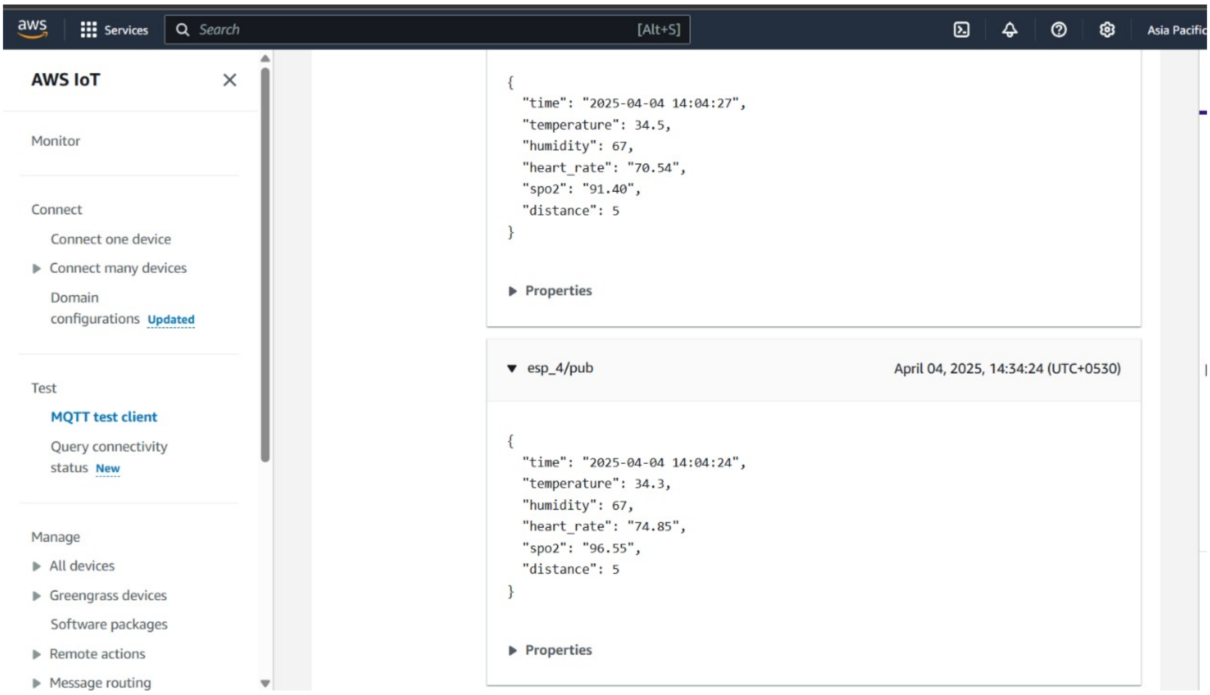


Figure 4.1: AWS IoT Core

MQTT topics for real-time publishing and subscription to sensor values.Secure communication authentication uses X.509 certificates. The ESP8266 sends temperature data (like 33.9°C) to AWS IoT via MQTT every few seconds.

4.2 AWS DynamoDB

Sensor Data Storage DynamoDB, a NoSQL database, stores incoming sensor data in JSON format with timestamps. High-speed, serverless storage options for structured data include: Dynamic database tables provide for quick searches and retrievals for analytics or notifications. Use case: Every field reading is captured for further examination and presentation.

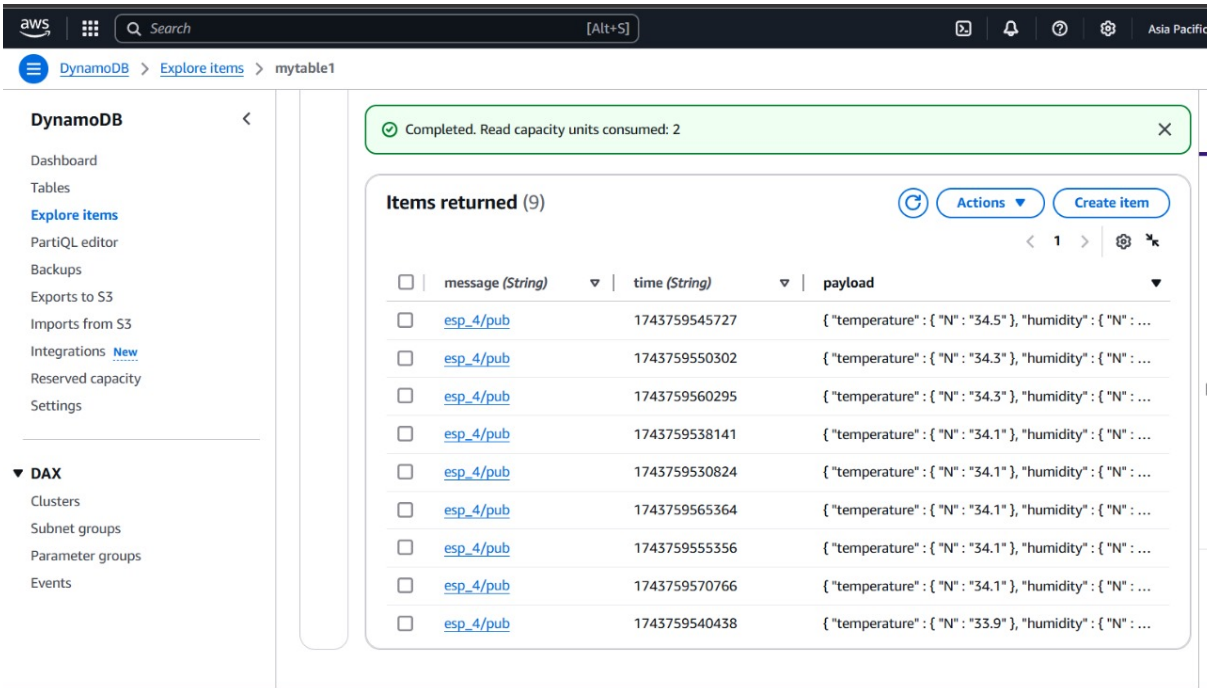


Figure 4.2: Dynamo DataBase

Use case: All field readings are recorded for further analysis and display.

4.3 AWS Lambda

AWS Data Processing and Filtering For example, new MQTT messages might activate the Lambda serverless computing service. Adding information to DynamoDB. functions created using Python or Node.js. JavaScript is used to check sensor data, for as by removing incorrect values like SpO = 0. Employ threshold logic to trigger an alert if the temperature> 38°C. Lambda guarantees lightweight, real-time decision-making. AWS SNS (Simple Notification Service)

4.4 AWS SNS (Simple Notification Service)

System of Warning SNS is used to send email or SMS alerts when sensor readings exceed danger thresholds. Lambda was added for alert delivery. enables multiple alert subscribers, such as a user, doctor, or emergency contact. If the SpO is abnormal or the heart rate drops, an automatic email notice is sent out immediately.

4.5 AWS CloudWatch (Useful Although Optional)

Monitoring and Logging: AWS services, including MQTT traffic and Lambda invocations, are monitored using CloudWatch. capturing error traces in order to debug Lambda functions. ensures the system's stability and lets you keep an eye on data mistakes or abnormalities.

4.6 Identity and Access Management (AWS IAM)

Security IAM roles and policies are configured to: Grant least-privilege access to every component. Secure communication using ESP8266, Lambda, DynamoDB, and SNS. upholds best practices and the security of the system.

CHAPTER 5

Results and Analysis

5.1 Results

5.1.1 Real-time Data Monitoring

The system successfully sent vital health parameters in real time to AWS IoT Core, including temperature, humidity, heart rate, SpO, and distance. The data on the esp$_4/pubtopicwaspublishedu$

5.2 Rule Triggering and Filtering

An SQL-based rule was applied to monitor abnormal values such as:

Parameter	**Expected/Simulated Range**
SpO_2	95–100%
Heart Rate	60–100 mmHg (simulated)
Temperature	$\approx$ 37°C (after scaling)

Table 5.1: Observed Biomedical Parameters

5.3 Email Notification Alerts

When the system detected unusual readings, it sent an alarm to a subject on social networking sites, and the topic sent the notice via email. The actual health information was included in the alert email, as can be seen. The alert email contained the actual health data, as seen in

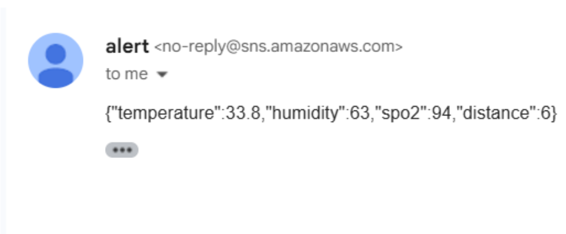


Figure 5.1: Email Notification Alert

5.4 Data Logging in DynamoDB

Every MQTT message was concurrently recorded in DynamoDB for additional analysis; each record contains a date, subject name, and sensor readings.

5.5 LabVIEW: Signal Acquisition and Processing

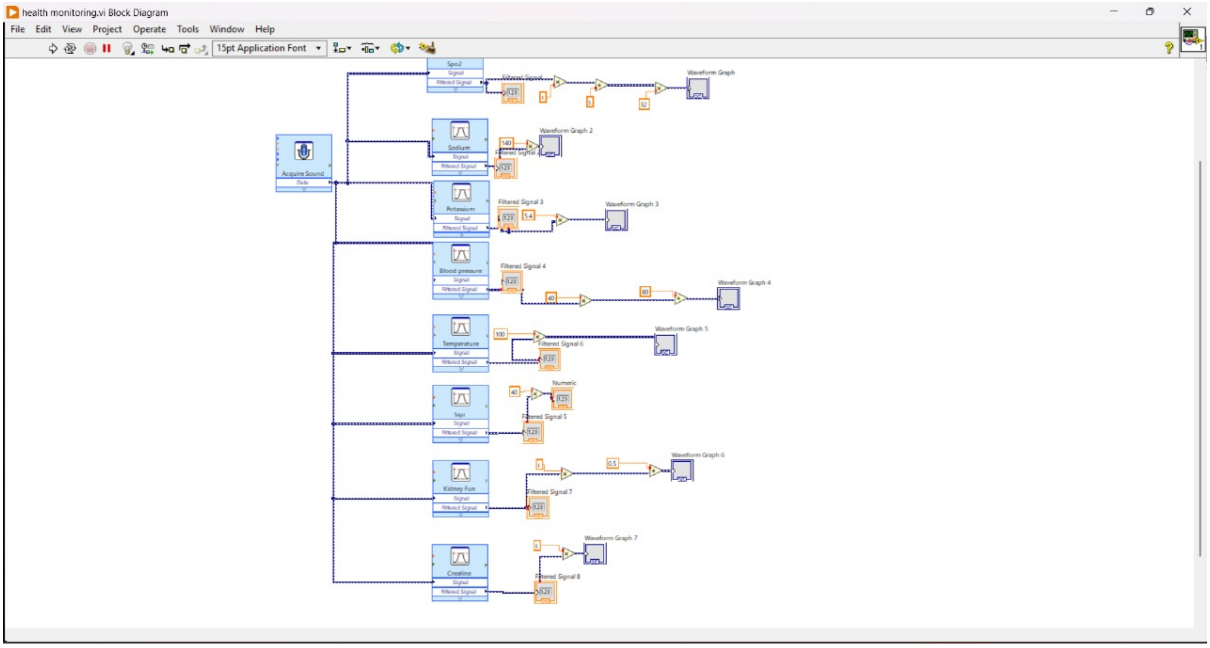


Figure 5.2: Labview Simulation

The LabVIEW VI was developed to gather and filter data from several biological sensors.From Figure 5.2 The input signal is acquired using the ”Acquire Sound” block, which represents analog signal input from many sensor modules. The following is how vital signals are processed: SpO_2 Potassium and Sodium Temperature of Blood Pressure Level of Sugar Function of the Kidneys Level of Creatinine A filter block is applied to each signal in order to eliminate noise and extract pertinent biological information.

5.6 Analysis

The system has the ability to capture and transmit real-time health data at regular intervals. AWS IoT rules successfully screened unusual values in real time. Alerts were triggered within seconds of detecting irregularities. Data that has been preserved can be used for trend analysis and long-term patient monitoring. This cloud-integrated approach provides a scalable, automated, and real-time framework for health monitoring that might be extended to hospitals or remote health systems. for Labview A Waveform Graph that shows data in real time is connected to each parameter. The user may monitor changes and patterns in health parameters over time thanks to this. This real-time graphic helps customers or clinicians keep an eye on developments in real time.

CHAPTER 6
Conclusion and Future Scope

6.1 Conclusion

When cloud computing and the Internet of Things are combined, an efficient platform for real-time environmental and health monitoring is offered. This project successfully demonstrates a full-stack system that collects, processes, and analyzes data remotely using sensors, Amazon Web Services (AWS), and microcontrollers (ESP8266). With the help of AWS IoT Core, DynamoDB, Lambda, and SNS, the solution ensures accurate and timely monitoring by offering scalable data management and quick warning generating. One of the system's main selling points is its ability to effectively handle interference and anomalous data using cloud-side validation and smart rule triggers. By addressing typical problems like signal noise and uneven sensor output, the research increases the reliability of remote monitoring systems. The proposed paradigm, which includes edge computing for offline operation, mobile app compatibility, and AI-based predictions, offers a solid foundation for further advancement. Future developments in this technology might have a big influence on environmental management, smart healthcare, and emergency alarm systems, making homes safer and more connected.

6.2 Scope of the Future

6.2.1 Predictive Monitoring with AI/ML Integration

Utilize AI/ML models in conjunction with services like AWS SageMaker to predict health anomalies or environmental threats. enables the early detection of issues such as elevated temperatures, oxygen deprivation, or rising pollution levels. The process of creating mobile applications Using AWS Amplify or React Native, develop a user-friendly mobile application that offers real-time data access, configurable thresholds, and notification alerts. increases user engagement and mobility.

6.2.2 Data Visualization and Improved Reporting

Make periodic reports and dynamic dashboards with technologies like Amazon QuickSight or Grafana. helps consumers and medical professionals make data-driven decisions and analyze trends.

6.2.3 Offline Processing using Edge Computing

Use ESP32/Raspberry Pi local processing or AWS Greengrass for offline alerting and local decision-making. reduces dependency on the internet and improves reaction time in remote areas. Multiple Users with Role-Based Access Implement secure access control using AWS Cognito or IAM to allow different users (such a doctor, patient, or administrator) to see and manage data in accordance with their responsibilities. enhances data privacy, scalability, and usability in bigger systems.

Learning Outcomes

(PO-1): Apply the knowledge based on curricular and co-curricular activities to solve Electronics and Telecommunication-based project work: The knowledge acquired from subjects like Python, excel, machine learning, etc. that are part of the curriculum and knowledge gained by studying these subjects helped us to understand our project better.

(PO-2): Systematically analyze electronics and telecommunication-related project based on the literature review: We referred to various paper where various applications of SVM, ANN, and dynamic weightage assignment were discussed. These papers helped us to understand and design systems.

(PO-3): Design and develop software code based on the problem specifications of the project: Our aim of the project is to implement a prototype to detect and analyses different parameters to assign weightage and to indicate whether to Buy/Sell a particular signal.

(PO-4): Carry out different experiments to generate data, analyze and interpret the data, and draw valid conclusions related to the project work: Use Excel to study and plan the different parameters of the stock market.

(PO-5): Select and apply appropriate modern tools for the solution of their project problem: We have used various software such as Excel, and Python and integrated the use of libraries as well.

(PO-6): Know the responsibility of the engineer towards society concerning their project work: As Electronics and Telecommunication Engineers, it is our responsibility to analyze the current needs and problems faced by society and if possible, plan, design develop effective solutions for the same.
(PO-7): Understand the impact of engineering solutions related to their project work in societal context for sustainable development: Engineers are increasingly required to play a leadership role in sustainable development, overcoming global challenges, such as depletion of resources, environmental pollution, rapid population growth and damage to ecosystems.

(PO-8): Apply professional ethical principles while project implementation, report writing, and publication. An engineer must always consider the safety of citizens, moral values, and the general well-being of society. We have therefore given proper references for the materials we have referred.

(PO-9): Work effectively as an individual and as a member of the team while the project work is carried out: While working on various aspects of our project, we realized the importance of teamwork and determination needed for the successful completion of the project in the context of helping each other.

(PO-10): Communicate effectively while project report writing and oral/visual presentations: Communication is an excellent skill that is helpful to express and understand ideas and interests, and the presentations have helped us achieve this skill.

(PO-11): Gain knowledge of engineering and management aspects while the project is being implemented: The knowledge of various engineering and management aspects such as performance, feasibility, accuracy, cost-effectiveness, etc. was gained while doing this project.

(PO-12): Engage themselves in independent and lifelong learning: Technology never remains stagnant. It keeps evolving over the years, and as engineers, we realize that we must keep ourselves updated throughout to improve our technical knowledge.

CO PO Mapping

Table 6.1: CO-PO Mapping

Course Outcome	PO1	PO2	P03	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	-	-	-	-	-	-	-	-	-	-
CO2	-	-	3	3	-	-	-	-	-	-	-	-
CO3	-	-	-	-	3	3	-	-	-	-	-	-
CO4	-	-	-	-	-	-	3	3	-	-	-	-
CO5	-	-	-	-	-	-	-	-	3	3	-	-
CO6	-	-	-	-	-	3	-	-	-	-	3	3

ACKNOWLEDGEMENT

We would like to express our sincere gratitude and appreciation to all the people and organizations who have been instrumental in the accomplishment of our significant endeavor. Our profound gratitude goes out to our Head of the Department, **Dr. Megha Kolhekar**, for giving us the chance to work with **Dr. Umesh M** and for the support offered during this project. This initiative has been greatly influenced by your leadership and priceless thoughts.

Our sincere appreciation is extended to our project mentors **Mrs. Akshata Raut** for their unwavering guidance, tolerance, and commitment. Your knowledge, support, and insightful criticism have been crucial to the success of my project. It gives us great pleasure to thank **Dr.Nilashree W** for supporting and being there for the team in any circumstance.

We would like to express our gratitude to the whole faculty of FCRIT's Electronics and Telecommunication Engineering Department for their invaluable instruction, which served as the basis for this project. Your dedication to sharing information and your readiness to answer our questions have been quite helpful. We appreciate the non-teaching staff's efforts in creating a supportive atmosphere for learning and research. Although their behind-the-scenes labor is sometimes overlooked, it is much valued. We appreciate the support and expertise that our classmates and fellow students have given us. We really appreciate your constant support and direction. I appreciate you playing a crucial role in this important turning point in my academic career.

Diksha Bhosale	(3021159)
Pratik Belamkar	(3021107)
Varun Somwanshi	(3021151)
Mudabbir Ahmad	(3021133)

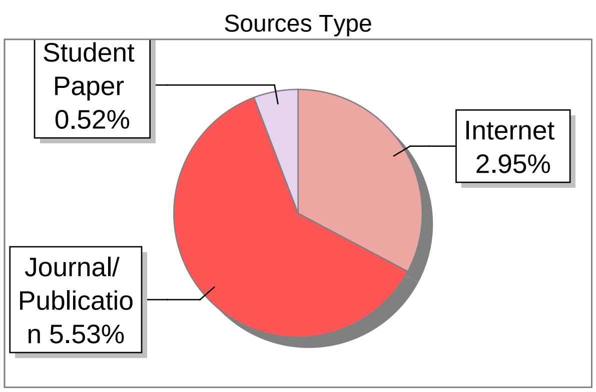
The Report is Generated by DrillBit Plagiarism Detection Software

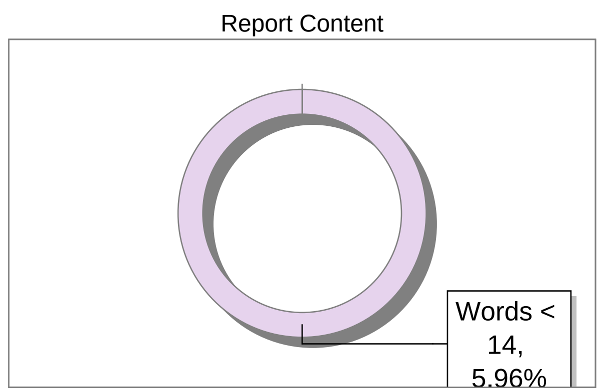
Submission Information

Author Name	Bhosale Diksha; Belamkar Pratik; Somwanshi Varun; Ahmad Mudabbir
Title	A project report on IoT Based Health Monitoring System
Paper/Submission ID	3575546
Submitted by	kalpana.hire@fcrit.ac.in
Submission Date	2025-05-05 10:29:18
Total Pages, Total Words	32, 6881
Document type	Project Work

Result Information

Similarity **9 %**





Exclude Information

Quotes	Not Excluded
References/Bibliography	Not Excluded
Source: Excluded < 14 Words	Not Excluded
Excluded Source	**0 %**
Excluded Phrases	Not Excluded

Database Selection

Language	English
Student Papers	Yes
Journals & publishers	Yes
Internet or Web	Yes
Institution Repository	Yes

A Unique QR Code use to View/Download/Share Pdf File

IOT-BASED PATIENT HEALTH MONITORING SYSTEM

*Note: Sub-titles are not captured in Xplore and should not be used

1st Pratik Belamkar
Student of EXTC Department
Fr. C. Rodrigues Institute of Technology
pratik232013@gmail.com

2nd Diksha Bhosale
Student of EXTC Department
Fr. C. Rodrigues Institute of Technology
bhosalediksha06@gmail.com

3rd Mudabbir Ahmed
Student of EXTC Department
Fr. C. Rodrigues Institute of Technology
mudabbirahmad0077@gmail.com

4th Varun Somwanshi
Student of EXTC Department
Fr. C. Rodrigues Institute of Technology
varunsomwanshi96@gmail.com

5th Akashata Raut
Assistant Professor, EXTC Department
Fr. C. Rodrigues Institute of Technology
akashata.raut@fcrit.ac.in

***Abstract*—The goal of this project is to design and create an Internet of Things (IoT)-based patient health monitoring system that can identify and show critical health metrics. Real-time patient monitoring and data analysis have been made possible by the Internet of Things (IoT), which has completely changed the healthcare industry. This study introduces an Internet of Things (IoT)-based patient health monitoring system that measures vital indicators, including blood oxygen saturation (SpO_2) and heart rate, non-invasively using ordinary and infrared (IR) LEDs. In order to analyze variations in blood volume, the system uses photoplethysmography (PPG) technology, in which ordinary and infrared LEDs shine light onto the patient's skin. A photodiode sensor then records the reflected light. A microcontroller processes the data before it is wirelessly sent via Internet of Things protocols to a cloud-based platform. This data is accessible to healthcare practitioners. ,**

I. Introduction

Patient health monitoring systems are essential for identifying and showing critical health metrics in the medical field. Improvements in the Internet of Things (IoT) have made it easier and more efficient to monitor health in real time. The goal of this project is to create an Internet of Things (IoT)-based patient health monitoring [1] system that continuously monitors vital health indicators including blood oxygen saturation and heart rate.(SpO_2). In order to measure variations in blood volume, the system uses photoplethysmography (PPG) technology, in which ordinary and infrared (IR) LEDs shine light onto the patient's skin. A photodiode sensor then records the reflected light. An ESP8266 microcontroller is used to process the data before it is wirelessly sent to a cloud-based platform. In order to ensure prompt medical intervention, healthcare providers can access this data in real time using a web or mobile application. The suggested method benefits patients with chronic illnesses and the elderly by providing an economical, portable, and effective remote health monitoring option. This solution reducesthe need for frequent hospital visits and improves overall patient outcomes by combining IoT technologies to improve healthcare accessible and provide continuous patient monitoring.

Identify applicable funding agency here. If none, delete this.

II. Literature Review

Health monitoring systems have evolved significantly with advances in IoT, wearable devices, and cloud computing. These systems aim to provide real-time monitoring, early diagnosis, and personalized health care solutions. The literature highlights the growing demand for remote patient monitoring, especially for chronic diseases, elderly care, and post-operative recovery. In the early 2000s, the term "Internet of Things" (IoT) was coined, and the concept of connecting devices to the internet gained traction. Wearable fitness trackers, such as pedometers, became popular, inspiring the medical industry to adopt IoT for health monitoring. IoT allows doctors to monitor patient health remotely using mobile applications and web platforms. During the mid-2000s, advancements in wireless communication protocols (e.g., Bluetooth) enabled the development of more sophisticated health monitoring devices. Researchers began exploring IoT applications in healthcare, leading to the integration of wireless data transmission and cloud-based analytics. IoT-based healthcare has transformed traditional monitoring methods. Wearable sensors track heart rate, SpO_2, and other vital parameters, continuously sending data to a secure cloud platform such as AWS(Amazon Web Services). Real-time alerts notify caregivers if abnormalities are detected, allowing for early intervention before emergencies arise. Further advancements in edge computing have made IoT healthcare systems more responsive. Some devices can process health data locally and send instant alerts before the data even reaches the cloud. This is particularly beneficial for elderly patients or individuals living alone, ensuring rapid assistance in critical situations. Despite these advancements, challenges remain. Power efficiency, data security, and sensor accuracy are ongoing concerns. Researchers are working on improving battery life, encryption methods, and AI-based data processing to enhance the effectiveness of IoT-based healthcare solutions. By integrating IoT, cloud computing, and real-time health monitoring, the reviewed literature underscores the importance of smart healthcare systems in improving patient safety, accessibility, and overall healthcare quality.

III. System Components

A. Sensors and Electronic Components

ordinary and infrared (IR) LEDs – Used for photoplethysmography (PPG) to measure heart rate and SpO_2. Ordinary and IR light are emitted onto the skin, and the variations in reflected light are analyzed to detect blood volume changes. Photodiode Sensor – Detects the intensity of the reflected IR light, helping to measure variations in blood circulation. Resistors (100 Ω and others) – Used to control current flow and prevent damage to LEDs and other circuit components as shown in (fig.1). Trimpot (Trimmer Potentiometer) – A variable resistor used to fine-tune the sensitivity of the photodiode sensor for accurate readings.

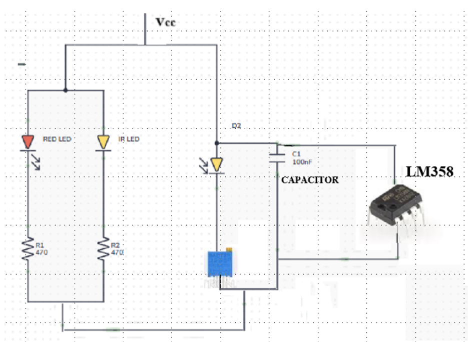


Fig. 1. Circuit Diagram

Capacitor – Helps stabilize the circuit by filtering noise and reducing fluctuations in the sensor signal.

B. Processing Unit

Sensor data must be gathered, examined, and transmitted by the processing unit.

The ESP8266 microcontroller is a high-performance, low-power microcontroller that uses Bluetooth or Wi-Fi to process sensor data and send it wirelessly. Low latency in real-time monitoring is another guarantee provided by the ESP8266. Operational amplifier IC LM358: This device amplifies weak signals from the photodiode sensor so that the microcontroller can read the data more easily.

C. Communication and Data Transmission

These elements allow for IoT-based real-time remote monitoring. The ESP8266's built-in Wi-Fi/Bluetooth module allows wireless data transfer to a connected device or the cloud [3]. AWS is a cloud-based platform that stores and processes patient data so that healthcare providers can access it instantly.

D. Power Supply and Circuit Components

All system components are guaranteed to operate correctly by the power supply. A 5V power supply gives the microcontroller and sensors a steady voltage. Various electronic components in the circuit are connected by connecting wires.

E. Software and Monitoring Tools

The patient data is processed and visualized by these technologies. Waveform outputs from sensors are seen and analyzed using LabVIEW. Real-time patient health data can be accessed remotely by physicians or caregivers through a web or mobile application. IoT protocols, such as MQTT(Message Queuing Telemetry Transport), HTTP (Hypertext Transfer Protocol), or WebSockets, are used to transmit data between the ESP8266 and the cloud in a secure and effective manner.

F. Output and Display Components

These elements show outcomes and offer feedback. Waveform signals from the sensor are debugged and verified using a digital oscilloscope [2]. Real-time health data can be displayed on a small screen by integrating an LCD (Liquid Crystal Display) or OLED (Organic Light Emitting Diode) display (optional).

IV. Design Methodology

In order to guarantee effective data collection, processing, and transmission, the Internet of Things-based Patient Health Monitoring System adheres to a systematic design methodology. Sensor integration, signal processing, data transfer, and cloud-based monitoring are some of the phases that make up the technique.

A. System Architecture

There are three primary layers in the system: ordinary and infrared LED sensors, photodiodes, and an ESP8266 microcontroller are all part of the data acquisition layer, which measures critical parameters [4]. The data processing and transmission layer uses an operational amplifier (LM358) to process sensor signals before sending the data to the cloud over Bluetooth or Wi-Fi. Data is stored in a cloud platform by the Cloud / User Interface Layer. enabling real-time readings to be accessed by medical professionals via a mobile or online application.

B. Hardware Implementation

The following actions are involved in the hardware setup: The patient's skin is illuminated by ordinary and infrared LEDs. Reflected light is detected by a photodiode sensor, which then transforms it into an electrical signal. The weak sensor signal is amplified by the operational amplifier, IC LM358 [7]. After processing the amplified signal, the ESP8266 microcontroller wirelessly sends the data.

C. Signal Processing

The system extracts heart rate and SpO_2 levels and analyzes changes in blood volume using Photoplethysmography (PPG). Capacitors filter noise to improve signal clarity, while a trimpot is used to alter sensitivity. Waveform outputs are created from the processed signals and can be seen using LabVIEW software or a digital oscilloscope.

D. Data Transmission & Cloud Integration

The ESP8266 sends processed data wirelessly using Wi-Fi or Bluetooth. Data is transmitted using IoT protocols such as MQTT or HTTP to a cloud-based platform (AWS, Firebase, or a dedicated server). The system ensures secure and real-time data access for healthcare professionals via a web/mobile application

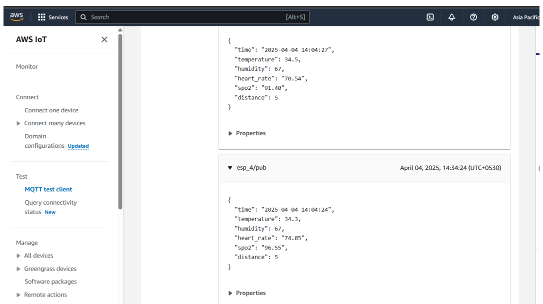


Fig. 2. Monitoring

E. Remote Monitoring & User Interface

A web or mobile application displays real-time patient health data. Alerts/notifications can be triggered if abnormal health parameters are detected [3]. The system supports remote access, allowing doctors or caregivers to monitor patients from anywhere.

F. Testing & Validation

The system is tested using a digital oscilloscope to verify the PPG waveform output. Data accuracy is compared against standard medical devices. Experimental results confirm the reliability and efficiency of the system in real-time health monitoring [5].

V. Hardware Respresentation

The waveform output of the IoT-based patient health monitoring system, likely showcasing variations in heart rate or SpO_2 levels over time. The waveform is a result of photoplethysmography (PPG) signal processing, where light absorption changes due to blood flow are analyzed. This graph helps in real-time monitoring and visualization of vital health parameters.

VI. Software

it visually represents real-time health monitoring data, likely showing variations in heart rate (BPM) or SpO_2 levels using PPG technology. The waveform indicates blood volume changes detected by IR and ordinary LEDs..

VII. Results

The IoT-based Patient Health Monitoring System successfully measured heart rate and SpO_2 using PPG technology with ordinary and IR LEDs. Real-time data transmission via ESP8266 enabled remote monitoring through a web/mobile application. The system's accuracy was validated against medical devices, showing reliable and consistent readings. The extracted waveform graph confirmed effective noninvasive health tracking, making it suitable for home-based and elderly care applications.

Fig. 3. Text alert

VIII. Conclusion

The IoT-based Patient Health Monitoring System provides a cost-effective, portable, and real-time solution for monitoring vital health parameters like heart rate and SpO_2. Using PPG technology with ordinary and IR LEDs, the system accurately detects changes in blood volume, processes the data via ESP8266, and transmits it wirelessly to a cloud-based platform for remote access. The results confirm the system's reliability and efficiency, making it beneficial for home healthcare, elderly care, and chronic disease management.

References

[1] V. Ramya, B. Palaniappan, and A. Kumari, "Embedded Patient Monitoring System," *International Journal of Embedded Systems and Applications*, vol. 1, no. 2, pp. 51–63, 2011.

[2] M. J. Jayashree, A. Sam Sunny, and Anu John, "Smart Hospital Technology," *An International Journal on Health Informatics*, vol. 3, no. 3, pp. 9–16, 2014.

[3] Bharat Prajapati and Jignesh Patel, "An Intelligent Real Time IoT Based System (IRTBS) for Monitoring ICU Patient," in *Proc. Int. Conf. on Information and Communication Technology for Intelligent Systems*, pp. 390–396, Springer, Cham, 2017.

[4] H. T. Yew, M. F. Ng, S. Z. Ping, S. K. Chung, A. Chekima, and J. A. Dargham, "IoT Based Real-Time Remote Patient Monitoring System," in *Proc. 2020 16th IEEE Int. Colloquium on Signal Processing and its Applications (CSPA)*, Feb. 2020, pp. 176–179.

[5] A. I. Abboudi, A. I. Alhammadi, K. M. Albastaki, N. u. M. Khanum, and A. Jarndal, "Design and Implementation of Portable Emergency Ventilator for COVID-19 Patients," in *2022 Advances in Science and Engineering Technology Int. Conferences (ASET)*, Dubai, UAE, 2022, pp. 1–4.

[6] N. Huda, S. Khan, R. Abid, S. B. Shuvo, M. M. Labib, and T. Hasan, "A low-cost, low-energy wearable ECG system with cloud-based arrhythmia detection," *medRxiv*, Sep. 02, 2020.

[7] C. W. Kerechanin, P. N. Cytcgusm, J. A. Vincent, D. G. Smith, and D. S. Wenstrand, "Development of field portable ventilator systems for domestic and military emergency medical response," *John Hopkins APL Technical Digest*, vol. 25, no. 3, 2004.

[8] Ashvini Kamble and Sonali Bhutad, "IoT-based Patient Health Monitoring System with Nested Cloud Security," in *Proc. 2018 4th Int. Conf. on Computing Communication and Automation (ICCCA)*, 2018.

AGNEL CHARITIES'

FR.C.RODRIGUES INSTITUTE OF TECHNOLOGY

(Autonomous Institute & Permanently Affiliated to University of Mumbai)

IEI ELECTRICAL STUDENTS' CHAPTER

IN ASSOCIATION WITH

The Institution of Engineers (India)

Belapur, Navi Mumbai

Certificate of Participation

This is to certify that

Mr/Mrs. Varun Somwanshi

has participated in 'Avishkar Competition' organized by IEI Electrical Students Chapter a National Level Project presentation competition on 19th of April 2025 at Fr.C.Rodrigues Institute of Technology, Vashi

Staff Coordinator
Divya Sajeesh

Bindu S

Head of Department
Dr. Bindu S.

AGNEL CHARITIES'

FR.C.RODRIGUES INSTITUTE OF TECHNOLOGY

(Autonomous Institute & Permanently Affiliated to University of Mumbai)

IEI ELECTRICAL STUDENTS' CHAPTER

IN ASSOCIATION WITH

The Institution of Engineers (India)

Belapur,Navi Mumbai

Certificate of Participation

This is to certify that

Mr/Mrs. Diksha Bhosle

has participated in 'Avishkar Competition' organized by IEI Electrical Students Chapter a National Level Project presentation competition on 19th of April 2025 at Fr.C.Rodrigues Institute of Technology, Vashi

Staff Coordinator
Divya Sajeesh

Bindu S

Head of Department
Dr. Bindu S.

AGNEL CHARITIES'

FR.C.RODRIGUES INSTITUTE OF TECHNOLOGY

(Autonomous Institute & Permanently Affiliated to University of Mumbai)

IEI ELECTRICAL STUDENTS' CHAPTER

IN ASSOCIATION WITH

The Institution of Engineers (India)

Belapur,Navi Mumbai

Certificate of Participation

This is to certify that

Mr/Mrs. Mudabbir Ahmad

has participated in 'Avishkar Competition' organized by IEI Electrical Students Chapter a National Level Project presentation competition on 19th of April 2025 at Fr.C.Rodrigues Institute of Technology, Vashi

Staff Coordinator
Divya Sajeesh

Bindus

Head of Department
Dr. Bindu S.

AGNEL CHARITIES'

FR.C.RODRIGUES INSTITUTE OF TECHNOLOGY

(Autonomous Institute & Permanently Affiliated to University of Mumbai)

IEI ELECTRICAL STUDENTS' CHAPTER

IN ASSOCIATION WITH

The Institution of Engineers (India)

Belapur,Navi Mumbai

Certificate of Participation

This is to certify that

Mr/Mrs. Pratik Belamkar

has participated in 'Avishkar Competition' organized by IEI Electrical Students Chapter a National Level Project presentation competition on 19th of April 2025 at Fr.C.Rodrigues Institute of Technology, Vashi

Staff Coordinator
Divya Sajeesh

Bindus

Head of Department
Dr. Bindu S.

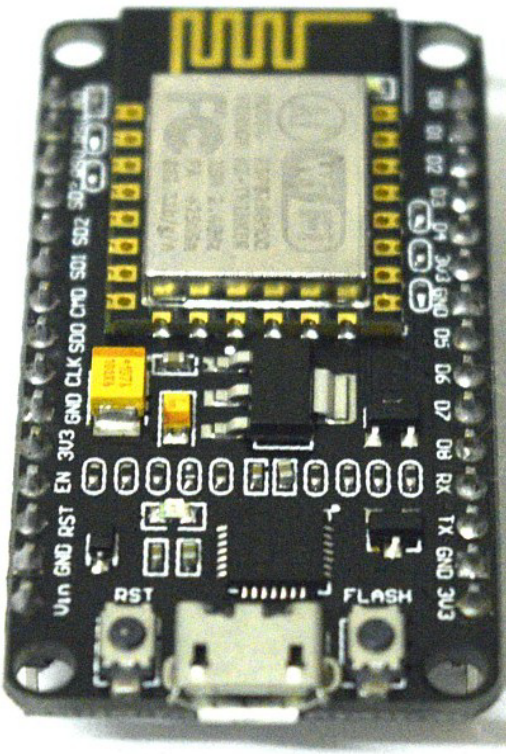
EINSTRONIC

TURN ON THE FUTURE

INTRODUCTION TO

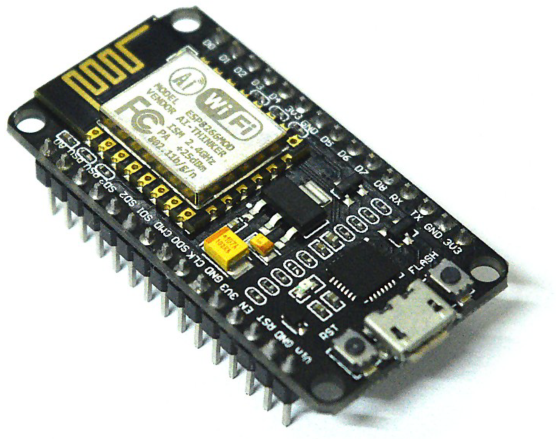
NodeMCU ESP8266

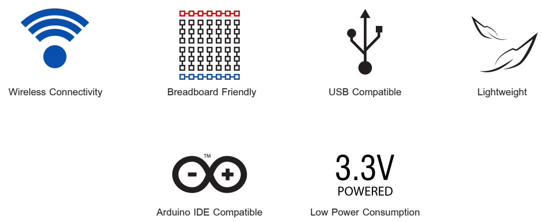
DEVKIT v1.0 **JULY 2017**



NodeMCU ESP8266 ESP-12E WiFi Development Board

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the DevKit. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson, and spiffs.



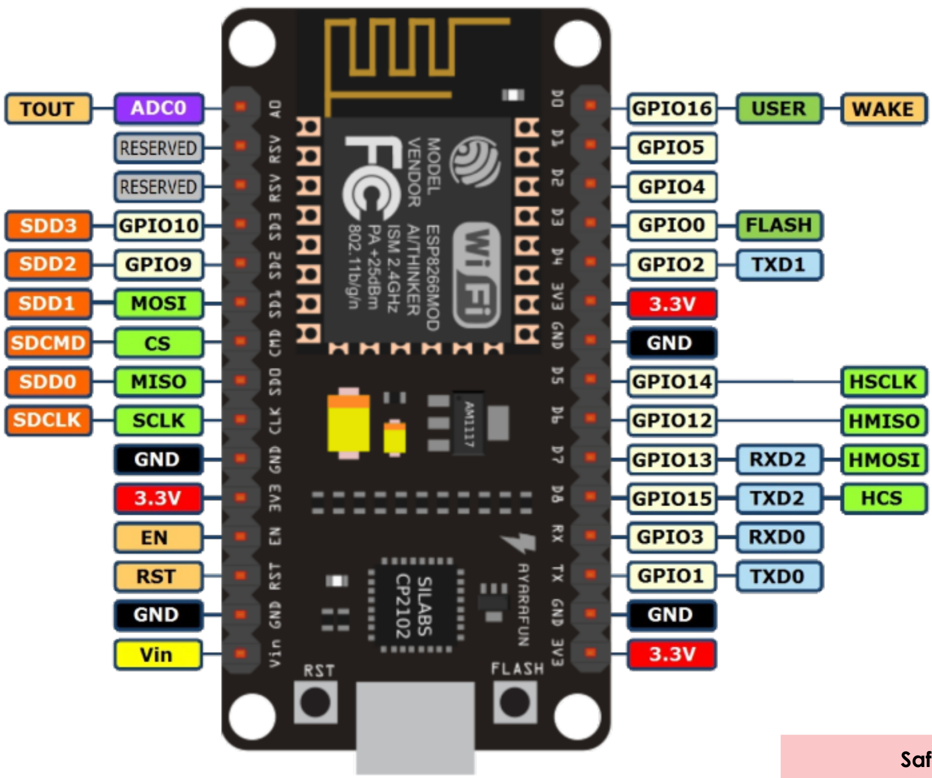


Features

- Version : DevKit v1.0
- Breadboard Friendly
- Light Weight and small size.
- 3.3V operated, can be USB powered.
- Uses wireless protocol 802.11b/g/n.
- Built-in wireless connectivity capabilities.
- Built-in PCB antenna on the ESP-12E chip.
- Capable of PWM, I2C, SPI, UART, 1-wire, 1 analog pin.
- Uses CP2102 USB Serial Communication interface module.
- Arduino IDE compatible (extension board manager required).
- Supports Lua (alike node.js) and Arduino C programming language.

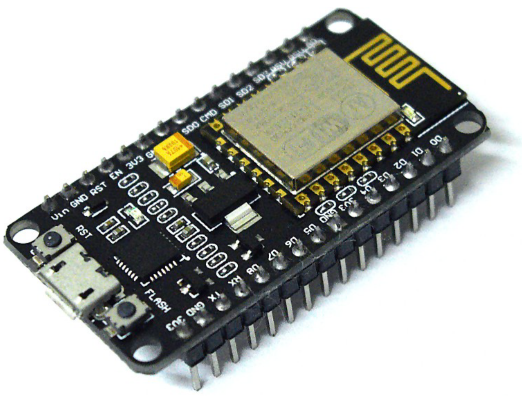
PINOUT DIAGRAM

NodeMCU ESP8266 v1.0

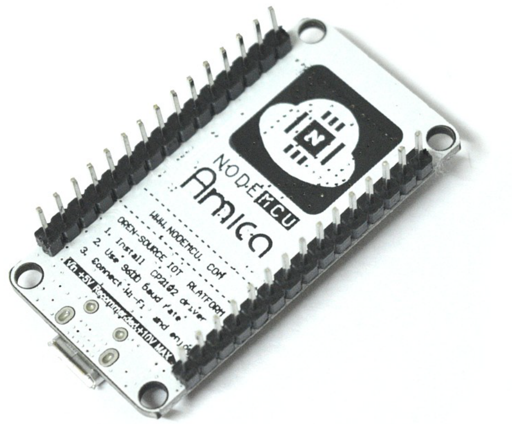


Source
https://iotbytes.wordpress.com/nodemcu-pinout/

Safety Precaution
All GPIO runs at 3.3V !!



Front View



Front View

Specifications of ESP-12E WiFi Module

Wireless Standard	IEEE 802.11 b/g/n
Frequency Range	2.412 - 2.484 GHz
Power Transmission	802.11b : +16 ± 2 dBm (at 11 Mbps) 802.11g : +14 ± 2 dBm (at 54 Mbps) 802.11n : +13 ± 2 dBM (at HT20, MCS7)
Receiving Sensitivity	802.11b : -93 dBm (at 11 Mbps, CCK) 802.11g : -85 dBm (at 54 Mbps, OFDM) 802.11n : -82 dBm (at HT20, MCS7)
Wireless Form	On-board PCB Antenna
IO Capability	UART, I2C, PWM, GPIO, 1 ADC
Electrical Characteristic	3.3 V Operated 15 mA output current per GPIO pin 12 - 200 mA working current Less than 200 uA standby current
Operating Temperature	-40 to +125 ºC
Serial Transmission	110 - 921600 bps, TCP Client 5
Wireless Network Type	STA / AP / STA + AP
Security Type	WEP / WPA-PSK / WPA2-PSK
Encryption Type	WEP64 / WEP128 / TKIP / AES
Firmware Upgrade	Local Serial Port, OTA Remote Upgrade
Network Protocol	IPv4, TCP / UDP / FTP / HTTP
User Configuration	AT + Order Set, Web Android / iOS, Smart Link APP

EINSTRONIC
TURN ON THE FUTURE

Related Sites

NodeMCU official site
http://nodemcu.com/index_en.html

NodeMCU Documentation
https://nodemcu.readthedocs.io/en/master/

NodeMCU Firmware (GitHub)
https://github.com/nodemcu/nodemcu-firmware

Project tagged with NodeMCU, HACKADAY.IO
https://hackaday.io/projects?tag=NodeMCU

ESP8266 Getting started, by ACROBOTIC industries
http://learn.acrobotic.com/tutorials/post/esp8266-getting-started

Quick Start to Nodemcu (ESP8266) on Arduino IDE
by Magesh Jayakumar
http://www.instructables.com/id/Quick-Start-to-Nodemcu-ESP8266-on-Arduino-IDE/

GETTING STARTED WITH PLATFORMIO AND ESP8266 NODEMCU
by Brandon Cannaday
https://www.losant.com/blog/getting-started-with-platformio-esp8266-nodemcu

Programming ESP8266 ESP-12E NodeMCU V1.0 With Arduino IDE Into Wireless Temperature Logger
by Shin Teo
http://www.instructables.com/id/ESP8266-NodeMCU-v10-ESP12-E-with-Arduino-IDE/

For more details, we can be reached at the addresses below.
Terms & Condition apply.

CONTACT INFORMATION

https www.einstronic.com
010 - 2181014 (Henry - Owner)
einstronics@gmail.com
facebook.com/einstronic

HC-SR04 Ultrasonic Sensor Module

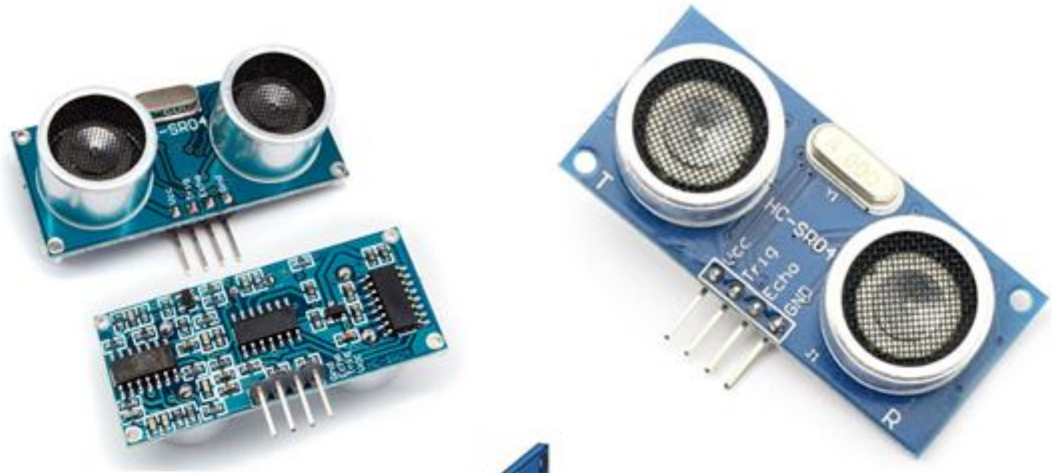
HC-SR04 Ultrasonic Sensor is a very affordable proximity/distance sensor that has been used mainly for object avoidance in various robotics projects. It has also been used in turret applications, water level sensing, and even as a parking sensor.

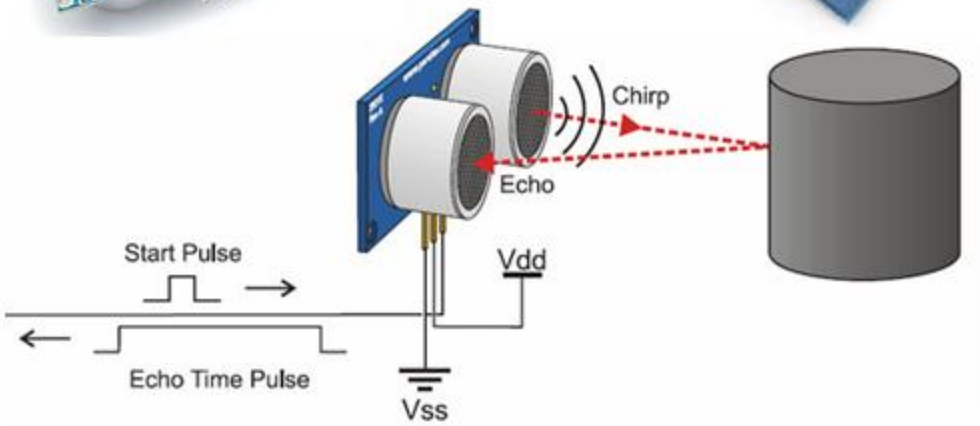
SKU: [SSR1012]

Brief Data:

- Power Supply: 3.3~5 VDC
- Quiescent Current : <2mA
- Working Current: 2.8mA @ 5V
- Effective Angle: <15°
- Ranging Distance : 2cm － 400 cm or 1″ － 13ft
- Connector: 4-pins header with 2.54mm pitch.
- Dimension: 45mm x 20mm x 15mm
- Weight 8.5g.

HC-SR04 Ultrasonic Sensor Module





User Guide: Ultrasonic Sensor V1.0

Table of Contents

1.0 Introduction:

Ultrasonic is an excellent way of figuring outwhat's in the immediate vicinity of your Arduino.The basics of using ultrasound are likethis: you shoot out a sound, wait to hear itecho back, and if you have your timing right,you'll know if anything is out there and howfar away it is. This is called echolocation and it'show bats and dolphins find objects in the darkand underwater, though they use lower frequencies thanyou can use with your Arduino.Figure-1 show the working principal of ultrasonic ranging concept.

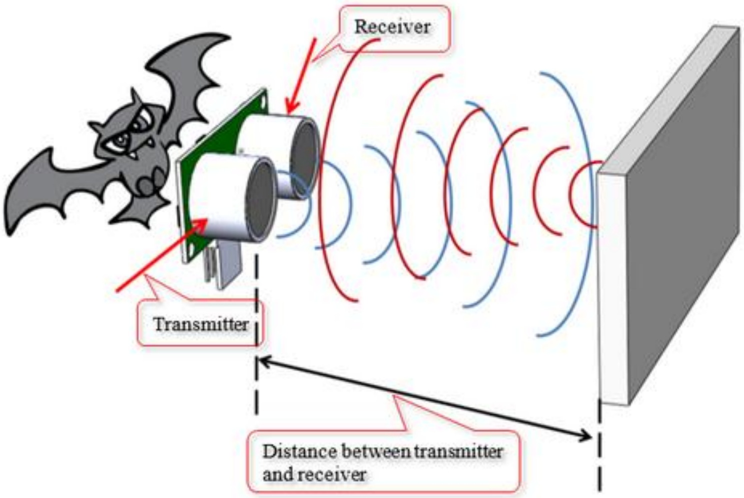


Figure-1

HC-SR04 Ultrasonic Sensor is a very affordable proximity/distance sensor that has been used mainly for object avoidance in various robotics projects. It has also been used in turret applications, water level sensing, and even as a parking sensor.

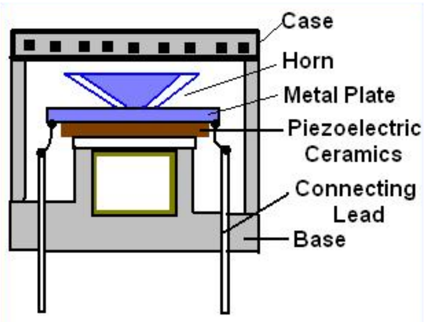
This module is the second generation of the popular HC-SR04 Low Cost Ultrasonic Sensor. Unlike the first generation HC-SR04 that can only operate between 4.8V~5V DC, this new version has wider input voltage range, allow it to work with controller operates on 3.3V.HC-SR04 ultrasonic sensor provides a very low-cost and easy method of distance measurement. It measures distance using sonar, an ultrasonic (well above human hearing) pulse (~40KHz) is transmitted from the unit and distance-to-target is determined by measuring the time required for the echo return. This sensor offers excellent range accuracy and stable readings in an easy-to-use package. An on board 2.54mm pitch pin header allows the sensor to be plugged into a solderless breadboard for easy prototyping.

2.0: Module Specification

Electrical Parameters	Value
Operating Voltage	3.3Vdc ~ 5Vdc
Quiescent Current	<2mA
Operating Current	15mA
Operating Frequency	40KHz
Operating Range	2cm ~ 400cm (1in ~ 13ft)
Sensitivity	-65dB min
Sound Pressure	112dB
Effective Angle	15°
Connector	4-pins header with 2.54mm pitch
Dimension	45mm x 20mm x 15mm
Weight	9g

2.1: Sensor Element Construction

Piezoelectric crystals are used for sensor elements. Piezoelectric crystals will oscillate at high frequencies when electric energy is applied to it. The Piezoelectric crystals will generate electrical signal when ultrasound wave hit the sensor surface in reverse.



Construction of Ultrasonic Sensors

3.0: Ultrasonic Real Application

3.1 Car Parking Reverse Sensors

The main purpose is the distance range detection, which is widely used parking sensor for car. The sensor is used forcalculating the distance, or direction of an object from the time it takes for a soundwave to travel to the object and echo back. The effective detective range is 0.3m ~ 3.0m. Refer to Figure-2.

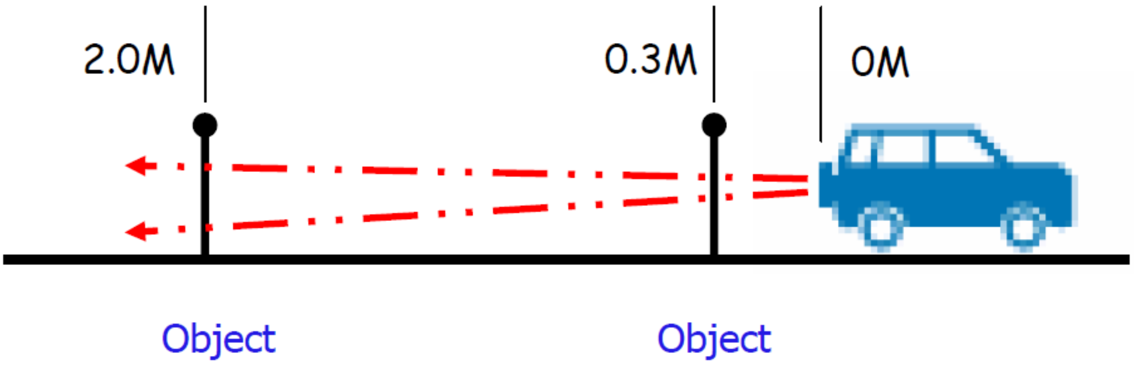


Figure-2.

3.2: Liquid Level Detection

Ultrasonic sensors are widely used forliquid level detection. In such cases,place a pipe on top of the sensor head as shown Figure-3. By detecting the liquid level inside thepipe, a wavy surface or bubbles which can disturb stable reading can be prevented.

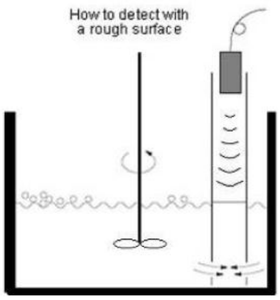
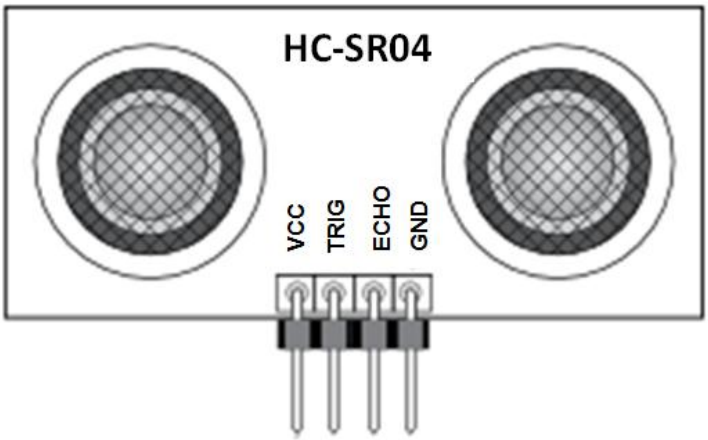


Figure-3.

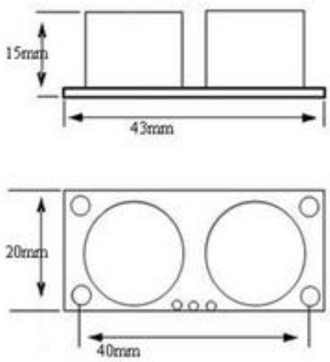
4.0: Pins Assignment and Dimension

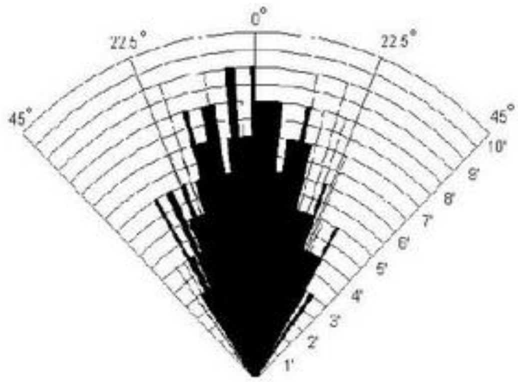
4.1 Pin Assignment



VCC	3.3v ~ 5V
TRIG	Triggering Input Pin. 10uS TTL Pulses
ECHO	TTL Logic Output Pin. Proportional to distance
GND	Ground Pin

4.2 Mechanical Dimension





Practical test of performance,
Best in 30 degree angle

4.3 Timing Diagram

The timing diagram, Figure-4 is shown below. You only need to supply a short 10uS pulse to “Trigger Input” pin to start the ranging.The module will send out 8-cycles burst of ultrasound at 40KHz and raise its “Echo” pin, refer to Figure-5. The echo is a distance object that is pulse width and the range in proportion. You can calculate the range through the time interval between sending trigger signal and receiving echo signal.

Formula: uS / 58 = centimeters or uS / 148 =inch

or: the range = high level time * sound velocity (340m/s) / 2;

Suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.

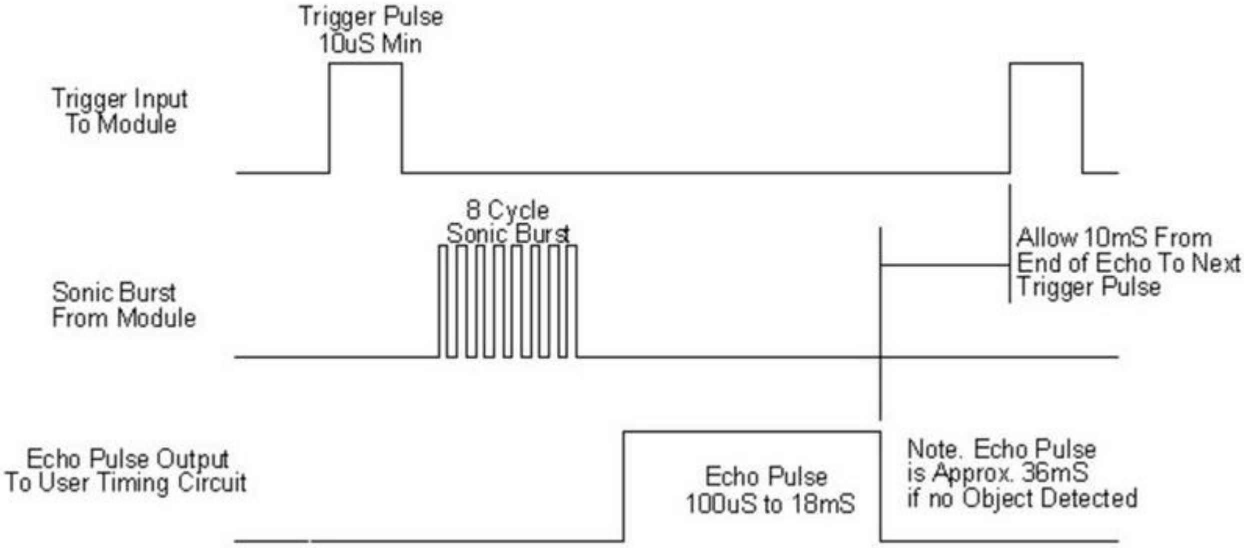


Figure-4: Timing Diagram

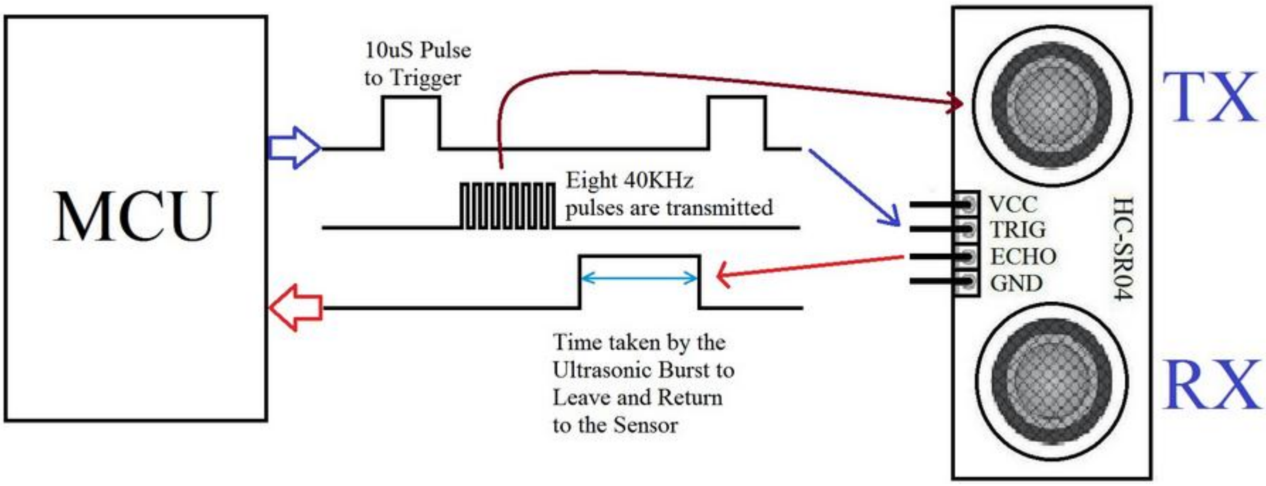
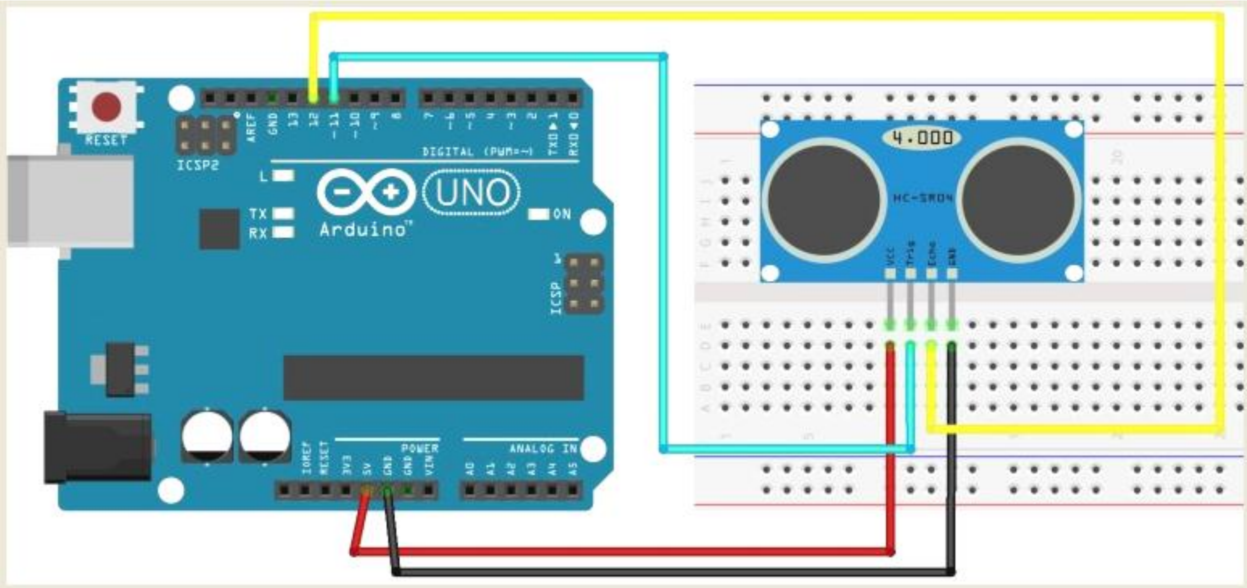


Figure-5: Microcontroller Interfacing

Please make sure the surface of object to be detected should have at least 0.5m^2areafor better performance.

4.4 Arduino Application Examples

Connect the circuit to Arduino as shown in below:



Ultrasonic Sensor HC-SR04	Arduino
VCC	5V
Trig	Pin 11
Echo	Pin 12
GND	GND

Upload the below sketch to Arduino Board. Open up the Serial monitor and place some object in front of the sensor module and observe the distance displayed.

```
/*============================================================================
//   Author       : Handson Technology
//   Project      : HC-SR04 Ultrasonic Sensor with Arduino Uno
//   Description  : HC-SR04 Distance Measure with Arduino and display
//                  on Serial Monitor.
//   Source-Code  : HC-SR04.ino
//============================================================================
*/

int trig=11;
int echo=12;
int duration;
float distance;
float meter;
void setup()
{
  Serial.begin(9600);
  pinMode(trig, OUTPUT);
```

```
  digitalWrite(trig, LOW);
  delayMicroseconds(2);
  pinMode(echo, INPUT);
  delay(6000);
  Serial.println("Distance:");
}
void loop()
{
  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);

  duration = pulseIn(echo, HIGH);

  if(duration>=38000){
      Serial.print("Out range");
      }

  else{
      distance = duration/58;
      Serial.print(distance);
      Serial.print("cm");
      meter=distance/100;
      Serial.print("\t");
      Serial.print(meter);
      Serial.println("m");
      }
  delay(1000);
}
```

5.0:HandsOn Technology Products Quality Commitments

HandsOn Technology wish to be perceived as simple and affordable by our customers. However the joy over a low price is never greater than the disappointment over poor quality products. All our parts are original genuine parts with proper data specifications from manufacturers. This is to ensure you always get the high quality genuine original part as stated in our products information.

5.1 WARRANTY

- Product warranty is valid for 6 months.
- Warranty only applies to manufacturing defect.
- Damaged caused by misuse is not cover under warranty.
- Warranty does not cover freight cost for both ways.

HandsOn Technology provides a multimedia and interactive platform for everyone interested in electronics. From beginner to diehard, from student to lecturer. Information, education, inspiration and entertainment. Analog and digital, practical and theoretical; software and hardware.

Hands*On* Technology support Open Source Hardware (OSHW) Development Platform.

Learn : Design : Share

handsontec.com

The Face behind our product quality…

In a world of constant change and continuous technological development, a new or replacement product is never far away – and they all need to be tested.
Many vendors simply import and sell wihtout checks and this cannot be the ultimate interests of anyone, particularly the customer. Every part sell on Handsotec is fully tested. So when buying from Handsontec products range, you can be confident you're getting outstanding quality and value.

We keep adding the new parts so that you can get rolling on your next project.

Breakout Boards & Modules

Connectors

Electro-Mechanical Parts

Engineering Material

Mechanical Hardware

P

Electronics Components

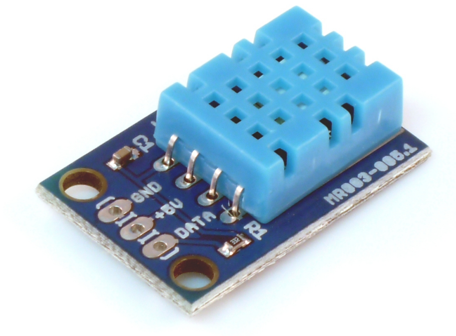
Power Supply

ARDUINO

Arduino Board & Shield

Tools & Accessory

Name: **DHT11 Humidity and Temperature Digital Sensor**
Code: **MR003-005.1**



This board is a breakout board for the DHT11 sensor and gives a digital output that is proportional to temperature and humidity measured by the sensor. Technology used to produce the DHT11 sensor grants high reliability, excellent long-term stability and very fast response time.

Each DHT11 element is accurately calibrated in the laboratory. Calibration coefficient is stored in the internal OTP memory and this value is used by the sensor's internal signal detecting process.
The single-wire serial interface makes the integration of this sensor in digital system quick and easy.
Sensor physical interfacing is realized through a 0.1" pitch 3-pin connector: +5V, GND and DATA. First two pins are power supply and ground and they are used to power the sensor, the third one is the sensor digital output signal.
Its small physical size (1.05"x0.7") and its very light weight (just 0.1oz) make this board an ideal choice to implementing small robots and ambient monitoring systems.

CONNECTIONS

DATA	Serial data output
GND	Ground
+5V	Power supply (+5V)

Tab.1 – Connections

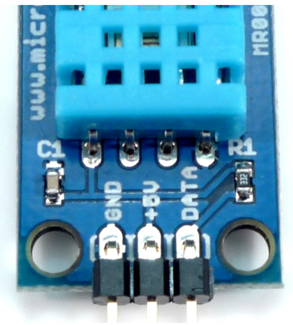


Fig. 1 - Signals

CHARACTERISTICS

Supply voltage	+5V
Supply current (running)	0.5mA typ. (2.5mA max.)
Supply current (stand-by)	100uA typ. (150uA max.)
Temperature range	0 / +50°C ±2°C
Humidity range	20-90%RH ±5%RH
Interface	Digital
Dimensions	1.05” x 0.7” (connectors excluded)
Weight	0.1 oz (2.7g)

Tab.2 - Characteristics

SENSOR UTILIZATION

The single-wire bus needs a 5Kohm pull-up resistor and the connection with the system is realized as showed in Fig.2.

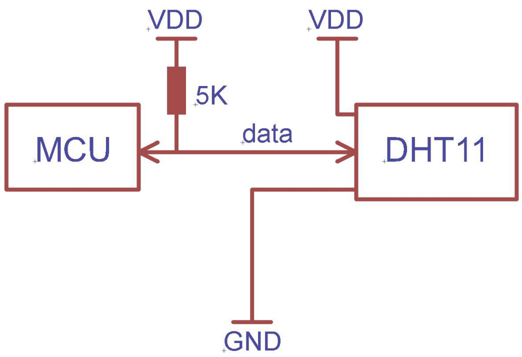


Fig. 2 - System

MICROBOT di Prosseda Mirko – Strada Chiesuola 117, 04010 Borgo Carso (LT), Italy
web: http://www.microbot.it email: info@microbot.it tel: +39-347-2159275

When power is supplied to the sensor, you haven't to send any instruction to the sensor in within one second in order to pass the start-up status.

After DHT11 is powered up, it goes in low power standby mode and it waits to recognize a Start Signal on the DATA line. Start Signal consists in a low voltage level on DATA line for a minimum of 18mS to ensure the DHT11 detects it, then followed by a pull-up voltage for about 40us.

Now the microcontroller has to wait the DHT11 transmission.

Once DHT11 detects the Start Signal, it will send out a low voltage level response signal, which lasts 80us. Then it sets the voltage level from low to high and keeps it for 80us.

Now data transmission will start. Every bit of data begins with the 50us low voltage level and then switch to high voltage level; high voltage level duration depends on the bit value that have to be transmitted: a 1 bit has an high voltage level duration of 27uS, a 0 bit has an high voltage level duration of 70uS.

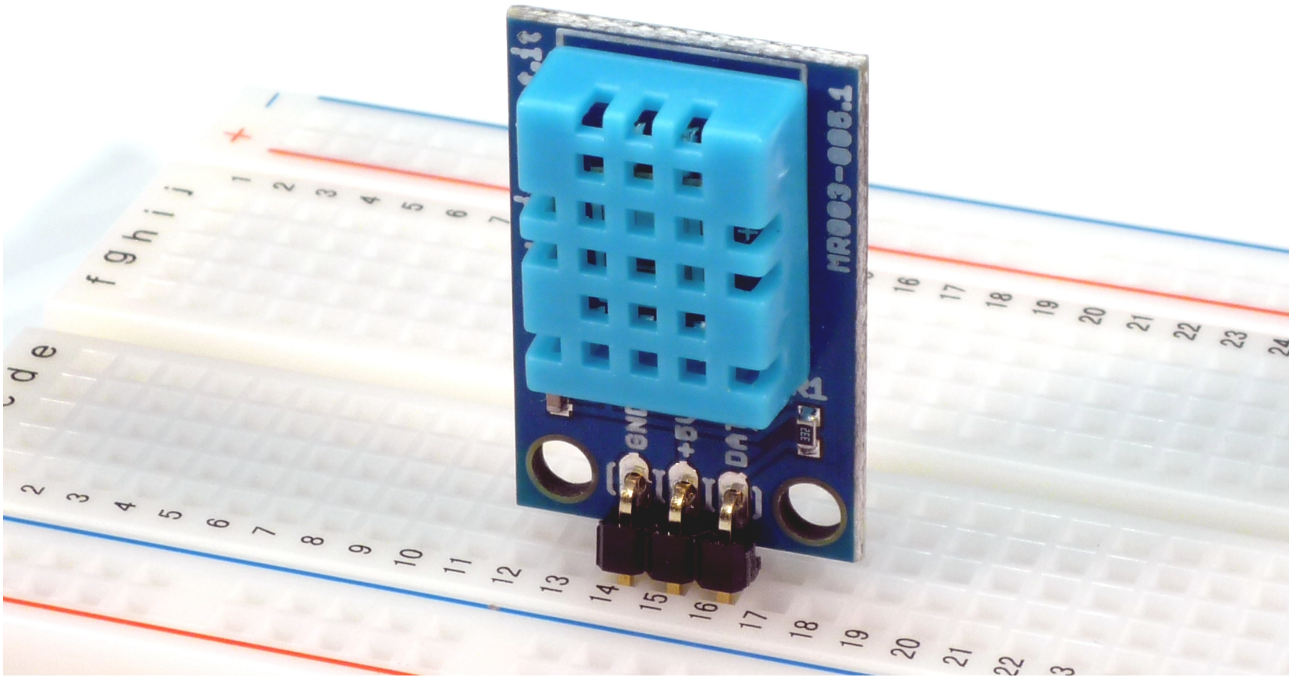
When the last bit data is transmitted, DHT11 pulls down the voltage level and keeps it for 50us, then it leaves the line pulled-up and goes back in the stand-by mode.

To make another sensor read it needs to repeat this cycle, sending again the Start Signal after a minimum of one second from the previous cycle.

A complete data transmission is 40bit, so a communication process is about 4mS. DHT11 sensor sends higher data bit first (MSB) in the following format:

Data = 8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal T data + 8bit check-sum.

If the data transmission is right, the check-sum will be equal to the last 8bit of the sum of the four byte transmitted.



MICROBOT di Prosseda Mirko – Strada Chiesuola 117, 04010 Borgo Carso (LT), Italy
web: http://www.microbot.it email: info@microbot.it tel: +39-347-2159275