

SMART AUTOMATED PESTICIDE SPRAYING BOT

A Project Report Phase – II

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

by

SANDU PRAVALLIKA (39130394)
PULAVARTHI SWADARSHANA (39130359)



**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 - 2023



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 COMMUNICAITON ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Final Year Project Report is the bonafide work of **SANDU PRAVALLIKA** (39130394) and **PULAVARTHI SWADRASHANA** (39130359) who carried out the project entitled "SMART AUTOMATED PESTICIDE SPRAYING BOT" under my supervision from NOVEMBER 2022 to APRIL 2023.

Internal Guide

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

Head of the Department

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

Submitted for Viva voce Examination held on 23.4.2023

Internal Examiner

External Examiner

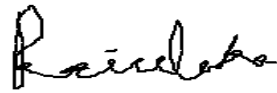
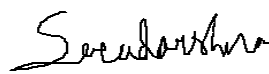
DECLARATION

We, **SANDU PRAVALLIKA** (REG NO: 39130394) and **PULAVARTHI SWADARSHANA**(REG NO: 39130359), hereby declare that the Project Report on "**SMART AUTOMATED PESTICIDE SPRAYING BOT** " done by us under the guidance of **Dr. T. VINO, M.E., Ph.D.**, Professor at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements of the award of Bachelor of Engineering degree in **Electronics and Communication Engineering**.

DATE: 23.4.2023

SIGNATURE OF THE CANDIDATES

PLACE: Chennai

1. 
2. 

ACKNOWLEDGEMENT

We are 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.

We convey our thanks to **Dr. N.M.NANDHITHA, M.E., Ph.D., Professor & Dean, School of Electrical and Electronics** and **Dr. T. RAVI, M.E., Ph.D., Professor and Head, Department of Electronics and Communication Engineering** for providing us necessary support and details at the right time during the progressive reviews.

We would like to express my sincere and deep sense of gratitude to our project Guide **Dr. T. VINO, M.E., Ph.D., Professor, Department of Electronics and Communication Engineering**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of our project work.

We wish to express our 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

By using an Arduino board, as the brains for the device, along with motors an automated vehicle can be designed. However, by complementing it with even one input sensor such as an MEMS sensor, the vehicle's functionality increases significantly. In the prototype, we have added various sensor modules like the MEMS sensor HC-SR04, Bluetooth module HC-05 and Ultrasonic sensors to allow the prototype to function in different modes. Instead of designing a new vehicle for each independent mode, we have decided to compile it all into one device.

There are several different methods of communicating with the prototype which would allow us to change the modes such as Ultrasonic, Wi-Fi and Bluetooth. To reduce the physical interaction with the vehicle and make it as automated as possible, we would be changing the modes by giving instructions to the vehicle via Bluetooth. Thus being able to switch to whatever mode needed effortlessly and conveniently. In order to be able to give commands to the prototype, we developed a simple Android application to control the speed of the motors as well as change modes with ease. Along with this, we incorporated the Google speech-to-text feature into the application as an added mode using the in-built microphone of the Android phone being used to run the application.

TABLE OF CONTENTS

Chapter No.	TITLE	Page No.
	ABSTRACT	v
	LIST OF FIGURES	viii
1	INTRODUCTION	1
2	LITERATURE SURVEY	
	2.1 Inferences from Literature Survey	10
	2.2 Open problems in Existing System	16
3	AIM AND SCOPE OF THE PRESENT INVESTIGATION	
	3.1 Project Objectives	18
	3.2 Scope of The Investigation	19
	3.3 Requirement Analysis	20
	3.4 Software Requirement Specification Document	22
4	DESCRIPTION OF PROPOSED SYSTEM	
	4.1 Proposed Methodology or process model	23
	4.2 Arduino Uno	24
	4.3 Ultrasonic Sensor	27
	4.4 Battery	31
5	IMPLEMENTATION DETAILS	
	5.1 Embedded C	38
	5.2 Embedded Systems Programming	40
	5.3 Uploading	47
6	CIRCUIT AND ITS WORKING	
	6.1 Circuit	53
	6.2 Android Application Development	56
7	RESULTS AND CONCLUSION	
	7.1 Result	58
	7.2 Summary	60
	7.4 Conclusion	61

LIST OF FIGURES

Figure No.	DESCRIPTION	Page No.
1.1	Working of IoT Enables Care Devices	8
1.2	IoT Controlled Green House Environment	9
2.1	Pesticide Spraying In Agriculture	16
2.2	Visual Examination of the System	17
3.1	Model of Phases in Project Management	19
4.1	Block Diagram of Proposed Methodology	23
4.2	Arduino Uno R3	24
4.3	ATMega328P and Arduino Uno Pin Mapping	25
4.4	Ultrasonic Sensor	27
4.5	HC-05 Module	30
5.1	Arduino Software Languages	49
6.1	Circuit Diagram	53
6.2	Application Interface	54
6.3	User Interface of Application	56
6.4	Application Backend	57
7.1	Intial Build	59
7.2	Finished Prototype	61

LIST OF TABLES

Table No.	DESCRIPTION	Page No.
4.1	Characterstics of ATMEGA328P	29
4.2	Male Connector Side	31
4.3	Specifications	31

CHAPTER-1

INTRODUCTION

Introduction

Internet of Things (IoT) technology has brought a revolution to each and every field of common man's life by making everything smart and intelligent. IoT refers to a network of things that make a self-configuring network. The development of Intelligent Smart Farming IoT-based devices is day by day turning the face of agriculture production by not only enhancing it but also making it cost-effective and reducing wastage. The aim/objective of this report is to propose IoT based Smart Farming System assisting farmers in getting Live Data (Temperature, Soil Moisture) for efficient environment monitoring which will enable them to increase their overall yield and quality of products. The IoT-based Smart Farming System being proposed via this report is integrated with Arduino Technology mixed with different Sensors and a Wi-Fi module producing a live data feed that can be obtained online from Things speak.com. The product being proposed is tested on Live Agriculture Fields giving high accuracy of over 98% in data feeds.

INTERNET OF THINGS:

Internet of Things (IoT) is a concept where each device is assigned to an IP address and through that IP address, anyone makes that device identifiable on the internet. The mechanical and digital machines are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. Basically, it started as the "Internet of Computers." Research studies have forecast explosive growth in the number of "things" or devices that will be connected to the Internet. The resulting network is called the "Internet of Things" (IoT). The recent developments in technology that permit the use of wireless controlling environments like Bluetooth and Wi-Fi have enabled different devices to have the capability of connecting with each other. Using a WIFI shield to act as a Micro web server for the Arduino eliminates the need for wired connections between the Arduino board and computer which reduces cost and enables it to work as a standalone device. The Wi-Fishield needs

a connection to the internet from a wireless router or wireless hotspot and this would act as the gateway for the Arduino to communicate with the internet. With this in mind, an internet-based home automation system for remote control and observing the status of home appliances is designed.

The concept of “Home Automation” has been in existence for several years. “Smart Home”, and “Intelligent Home” are terms that followed and is been used to introduce the concept of networking appliances within the house. Home Automation Systems (HASS) includes centralized control and distance status monitoring of lighting, security system, and other appliances and systems within a house. Home automation systems enable energy efficiency, improve the security systems, and certainly the comfort and ease of users. In the present emerging market, the home automation system is gaining popularity and has attracted the interest of many users. Our system comes with its own challenges. Mainly being, in the present day, end-users especially the elderly and disabled, even though hugely benefited, aren’t seen to accept the system due to the complexity and cost factors.

IoT as a term has evolved a long way as a result of the convergence of multiple technologies, machine learning, embedded systems, and commodity sensors. IoT is a system of interconnected devices assigned a UIDS, enabling data transfer and control of devices over a network. It reduced the necessity of actual interaction in order to control a device. IoT is an advanced automation and analytics system that exploits networking, sensing, big data, and artificial intelligence technology to deliver complete systems for a product or service. These systems allow greater transparency, control, and performance when applied to any industry or system.

Features of IoT:

- A. Intelligence
- B. Connectivity
- C. Dynamic Nature
- D. Enormous Scale
- E. Sensing
- F. Heterogeneity

Intelligence:

IoT comes with the combination of algorithms and computation, software & hardware that makes it smart. Ambient intelligence in IoT enhances its capabilities which facilitate the things to respond in an intelligent way to a particular situation and supports them in carrying out specific tasks. In spite of all the popularity of smart technologies, intelligence in IoT is only concerned with a means of interaction between devices, while user and device interaction are achieved by standard input methods and graphical user interface

Connectivity:

Connectivity empowers the Internet of Things by bringing together everyday objects. The connectivity of these objects is pivotal because simple object-level interactions contribute to collective intelligence in the IoT network. It enables network accessibility and compatibility in the things. With this connectivity, new market opportunities for the Internet of things can be created by the networking of smart things and applications

Dynamic Nature:

The primary activity of the Internet of Things is to collect data from its environment, this is achieved with the dynamic changes that take place around the devices. The state of these devices changes dynamically, for example sleeping and waking up, connected and/or disconnected as well as the context of devices including temperature, location, and speed. In addition to the state of the device, the number of devices also changes dynamically with a person, place, and time

Enormous Scale:

The number of devices that need to be managed and that communicate with each other will be much larger than the devices connected to the current Internet. The management of data generated from these devices and their interpretation for application purposes becomes more critical. Gartner (2015) confirms the enormous scale of IoT in the estimated report where it stated that 5.5 million new things will get connected every day and 6.4 billion connected things will be in use worldwide in 2016, which is up by 30 percent from 2015. The report also forecasts that the number of connected devices will reach 20.8 billion by 2020.

Sensing:

It wouldn't be possible without sensors that will detect or measure any changes in the environment to generate data that can report on their status or even interact with the environment. Sensing technologies provide the means to create capabilities that reflect a true awareness of the physical world and the people in it. The sensing information is simply the analog input from the physical world, but it can provide a rich understanding of our complex world.

Heterogeneity:

Heterogeneity in the Internet of Things is one of the key characteristics. Devices in IoT are based on different hardware platforms and networks and can interact with other devices or service platforms through different networks. IoT architecture should support direct network connectivity between heterogeneous networks. The key design requirements for heterogeneous things and their environments in IoT are scalabilities, modularity, extensibility, and interoperability.

Security:

IoT devices are naturally vulnerable to security threats. As we gain efficiencies, novel experiences, and other benefits from the IoT, it would be a mistake to forget about the security concerns associated with it. There is a high level of transparency and privacy issues with IoT. It is important to secure the endpoints, the networks, and the data that is transferred across all of it means creating a security paradigm.

Advantages of IoT

- A. Communication
- B. Automation and Control
- C. Information
- D. Monitor
- E. Time
- F. Money
- G. Automation of daily tasks leads to better monitoring of devices
- H. Efficient and saves time
- I. Saves money
- J. Better Quality of Life

Communication

IoT encourages communication between devices, also famously known as Machine-to-Machine (M2M) communication. Because of this, the physical devices are able to stay connected, and hence the total transparency is available with lesser inefficiencies and greater quality.

Automation and Control

Due to physical objects getting connected and controlled digitally and centrally with wireless infrastructure, there is a large amount of automation and control in the workings. Without human intervention, the machines are able to communicate with each other leading to faster and timely output.

Information

It is obvious that having more information helps make better decisions. Whether it is mundane decisions such as needing to know what to buy at the grocery store or if your company has enough widgets and supplies, knowledge is power and more knowledge is better.

Monitor

The second most obvious advantage of IoT is monitoring. Knowing the exact quantity of supplies or the air quality in your home can further provide more information that could not have previously been collected easily. For instance, knowing that you are low on milk or printer ink could save you another trip to the store in the near future. Furthermore, monitoring the expiration of products can and will improve safety.

Time

As hinted in the previous examples, the amount of time saved because of IoT could be quite large. And in today's modern life, we all could use more time.

Money

The biggest advantage of IoT is saving money. If the price of the tagging and monitoring equipment is less than the amount of money saved, then the Internet of Things will be very widely adopted. IoT fundamentally proves to be very helpful to people in their daily routines by making the appliances communicate with each other in an effective manner thereby saving and conserving energy and cost. Allowing the data to be communicated and shared between devices and then

translating it in our required way, makes our systems efficient.

Automation of daily tasks leads to better monitoring of devices

The IoT allows you to automate and control the tasks that are done on a daily basis, avoiding human intervention. Machine-to-machine communication helps to maintain transparency in the processes. It also leads to uniformity in the tasks. It can also maintain the quality of service. We can also take necessary action in case of *emergencies*.

Efficient and saves time

The machine-to-machine interaction provides better efficiency, hence; accurate results can be obtained fast. This results in saving valuable time. Instead of repeating the same tasks every day, it enables people to do other creative jobs.

Saves Money

Optimum utilization of energy and resources can be achieved by adopting this technology and keeping the devices under surveillance. We can be alerted in case of possible bottlenecks, breakdowns, and damages to the system. Hence, we can save money by using this technology.

Disadvantages of IoT:

- A. Compatibility
- B. Complexity
- C. Privacy / Security
- D. Safety
- E. Lesser Employment of Menial Staff
- F. Technology Takes Control of Life

Compatibility

Currently, there is no international standard of compatibility for the tagging and monitoring equipment. I believe this disadvantage is the easiest to overcome. The manufacturing companies of this equipment just need to agree to a standard, such as Bluetooth, USB, etc. This is nothing new or innovative needed.

Complexity

As with all complex systems, there are more opportunities for failure. With the Internet of Things, failures could skyrocket. For instance, let's say that both you and your spouse each get a message saying that your milk has expired, and both of you stop at a store on your way home, and you both purchase milk. As a result,

you and your spouse have purchased twice the amount that you both need. Or maybe a bug in the software ends up automatically ordering a new ink cartridge for your printer each and every hour for a few days, or at least after each power failure when you only need a single replacement.

Privacy / Security

With all of this IoT data being transmitted, the risk of losing privacy increases. For instance, how well encrypted will the data be kept and transmitted? Do you want your neighbors or employers to know what medications you are taking or your financial situation.

Safety

Imagine if a notorious hacker changes your prescription. Or if a store automatically ships you an equivalent product that you are allergic to, a flavor that you do not like, or a product that is already expired. As a result, safety is ultimately in the hands of the consumer to verify any and all automation. As all the household appliances, industrial machinery, public sector services like water supply and transport, and many other devices are connected to the Internet, a lot of information is available on it. This information is prone to attack by hackers. It would be very disastrous if private and confidential information is accessed by unauthorized intruders.

Lesser Employment of Menial Staff

The unskilled workers and helpers may end up losing their jobs in the effect of automation of daily activities. This can lead to unemployment issues in society. This is a problem with the advent of any technology and can be overcome with education. With daily activities getting automated, naturally, there will be fewer requirements for human resources, primarily, workers and less educated staff. This may create an Unemployment issue in society.

Technology Takes Control of Life

Our lives will be increasingly controlled by technology and will be dependent on it. The younger generation is already addicted to technology for every little thing. We have to decide how much of our daily lives are we willing to mechanize and be controlled by technology.

Application Grounds of IoT:

- A. Wearables
- B. Smart Homes
- C. Health Care
- D. Agriculture
- E. Industrial Automation
- F. Government and Safety

Wearables

Wearable technologies are a hallmark of IoT applications and are one of the earliest industries to have deployed IoT at its services. Fit Bits, heart rate monitors, smartwatches, and glucose monitoring devices reflect the successful applications of IoT.

Smart homes

This area of application is concerned to this particular project, so a detailed application is discussed further. *Jarvis*, an AI home automation employed by Mark Zuckerberg, is a remarkable example in this field of application.

Health Care

IoT applications have turned the reactive medical-based system into the proactive wellness-based system. IoT focuses on creating systems rather than equipment. IoT creates a future of medicine and healthcare which exploits a highly integrated network of sophisticated medical devices. The integration of all elements provides more accuracy, more attention to detail, faster reactions to events, and constant improvement while reducing the typical overhead of medical research.

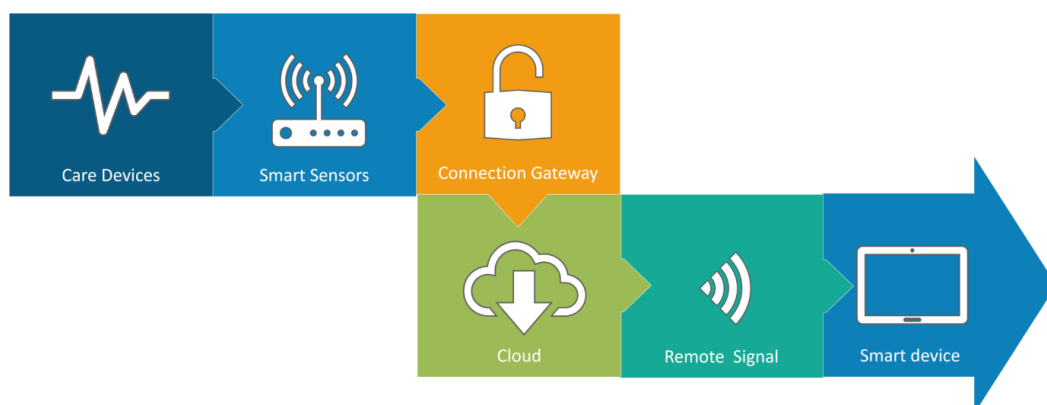


Fig 1.1 Working of IoT enables care devices

Agriculture

A greenhouse farming technique enhances the yield of crops by controlling environmental parameters. However, manual handling results in production loss, energy loss, and labor cost, making the process less effective. A greenhouse with embedded devices not only makes it easier to be monitored but also, enables us to control the climate inside it. Sensors measure different parameters according to the plant requirement and send them to the cloud. It, then, processes the data and applies a control action.

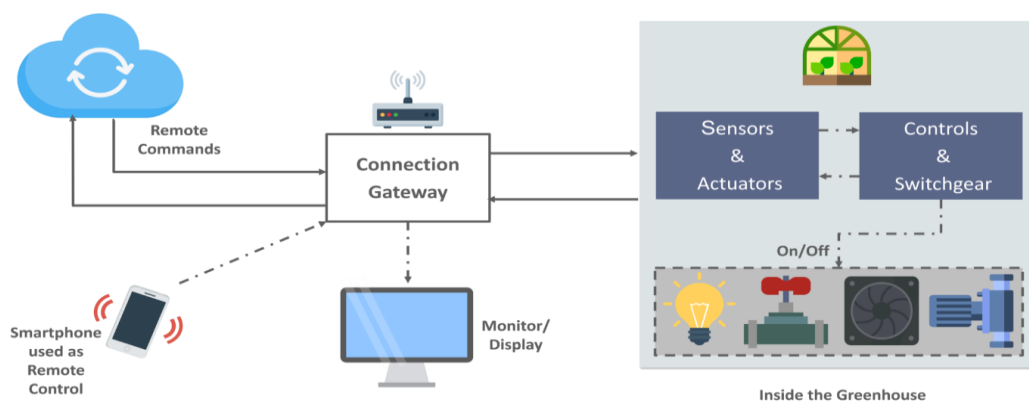


Fig 1.2 IoT Controlled Greenhouse Environment

Industrial Automation

For a higher return on investment, this field requires both fast developments and quality products. This vitality thus coined the term IIOT. This whole schematic is re-engineered by IoT applications. Following are the domains of IoT applications in industrial automation

- 1) Factory Digitalization
- 2) Product flow Monitoring
- 3) Inventory Management
- 4) Safety and Security
- 5) Quality Control

CHAPTER -2

LITERATURE SURVEY

The following shows a survey done for dual home automation for with and without internet on Realtime which includes an instant feedback mechanism that drives the appliances. This system is reliable to integrate on multiple platforms with the help of ESP32 master which serves as standard firmware. Further, the study was extended to multiple operation platforms like Linux, Windows, and IOS are discussed as follows.

Satyendra K. Vishwakarma, Prashant Upadhyaya, Babita Kumari, and Arun Kumar Mishra, IJSR “Smart Energy Efficient Home Automation System Using IoT”, IJSR Transaction on Computer Science Engineering 2021.

This paper presents a step-by-step procedure of a smart home automation controller. It uses IoT to convert home appliances to smart and intelligent devices, with the help of design control. An energy-efficient system is designed that accesses the smart home remotely using IoT connectivity. The proposed system mainly requires, Node MCU as the microcontroller unit, IFTTT to interpret voice commands, Adafruit a library that supports MQTT acts as an MQTT broker, and Arduino IDE to code the microcontroller. This multimodal system uses Google Assistant along with a web-based application to control the smart home. The smart home is implemented with the main control unit that is connected to the 24-hour available Wi-Fi network. To ensure, that the Wi-Fi connection does not turn off, the main controller is programmed to establish an automatic connection with the available network and connected to the auto power backup.

Shardha Somani, Parikshit Solunke, Shaunak Oke, Parth Medhi. Prof. P. P. Laturkar, IEEE “IoT Based Smart Security and Home Automation”, IEEE Transactions on Augmented Reality, 2021

This paper focuses on a system that provides features of Home Automation relying on IoT to operate easily, in addition to that it includes a camera module and provides home security. The android application basically converts smartphones into a remote for all home appliances. Security is achieved with motion sensors if movement is sensed at the entrance of the house; a notification is sent that

contains a photo of the house entrance in real-time. This notification will be received by the owner of the house via the internet such that the app can trigger a notification.

Tui-Yi Yang, Chu-Sing Yang, IJEEE “A Dynamic Distributed Energy Management Algorithm of Home Sensor Network for Home Automation System”, IJEEE Transactions on Electrical and Electronics Engineers, 2017

This paper proposes an optimization of home power consumption based on PLC (Power Line Communication) for an easy to access home energy consumption. This also proposes a Zigbee and PLC-based renewable energy gateway to monitor the energy generation of renewable energies. ACS and DDEM algorithms are proposed for the design of intelligent distribution of power managementsystems to make sure ongoing power supply of home networks. To provide efficient power management the power supply models of home sensor networks are classified into groups viz. main supply only, main supply and backup battery, rechargeable battery power, and non-rechargeable battery power. Devices with particular features are assigned to these groups. It targets establishing a real-timeprocessing scheme to address variable sensor network topologies.

Tushar Chaurasia, and Prashant Kumar Jain “Enhance Smart Home Automation System based on Internet of Things”, Transactions on Artificial Neural Networks, 2019

This paper proposes a system that develops a model to reduce the computation overhead in existing smart home solutions that use various encryption technologies like AES, ECHD, hybrid, etc. these solutions use an intermediate gateway for connecting various sensor devices. The proposed model provides a method for automation with sensor-based learning. The system uses a temperature sensor for development but other sensors can also be used as per requirement. These smart home devices with sensors can configure themselves autonomously and can operate without human intervention.

This work minimizes encryption decryption and focuses on authentication and automation of smart home devices with learning. The system bypasses the local gateway mentioned in the existing system to provide better security for smart home

devices and sensor data and save computation overhead. The real-time broker cloud is directly connected with the smart home and manages all incoming and outgoing requests between users and devices. The main purpose to use a real-time broker cloud saves time for cryptographic operations.

Suraj, Ishi Kool, Dharmendra Kumar, Shovan Barman “Visual Machine Intelligence for Home Automation”, MDPI Transactions on Applied Science, 2020

The paper presents a vision-based machine intelligence system to sense the on/off state of common home appliances. The proposed method of sensing the state of appliances results in a novel home automation system. The accessibility of the suite of devices in the home over a remote network is facilitated by the IP Addressing methods in the IoT. This project uses two boards viz. Raspberry Pi and Intel Galileo Gen 2. The communication between the User devices, Raspberry Pi, and the Intel Galileo boards happens over a wireless network. The UDP protocol is deployed to facilitate the wireless communication of the nodes present in the home automation network. A Pi Cam and a USB Logitech camera are attached to the rotating shaft of two different servo motor capture snapshots that are passed as inputs to the Machine Learning based models trained using dlib-C++ to detect the state of the operation of the appliances. The proposed method uses visual modality to automate the appliances, as privacy concerns may emerge while using the images from some specific places, as a counter to this issue, an SPDT switch is added to the Raspberry Pi which when turned off ensures that even if the images are taken from the webcams, they are just passed as inputs to the machine learning models and are not displayed on the website when the users access the website on the server address obtained from Raspberry Pi.

Vikram N, Harish KS, Nihal M S, Raksha Umesh, and, Shetty Ashik Ashok Kumar, “A Low-Cost Home Automation System using Wi-Fi-based Wireless sensor Network incorporating Internet of things”, Procedia Computer Science Transactions on Mobile Device, 2015

This paper illustrates a methodology to provide a low-cost Home Automation System (HAS) using Wireless Fidelity (Wi-Fi). This crystallizes the concept of internetworking of smart devices. A Wi-Fi-based Wireless Sensor Network (WSN)

is designed for the purpose of monitoring and controlling the environmental, safety, and electrical parameters of a smart interconnected home. The different sections of the HAS are; the temperature and humidity sensor, gas leakage warning system, fire alarm system, burglar alarm system, rain-sensing, switching and regulation of load & voltage, and current sensing. The primary requirement of HAS to monitor and control devices is accomplished using a Smartphone application. The application is developed using Android Studio based on the JAVA platform and the User Interface is exemplified. The primary focus of the paper is to develop a solution cost-effective and flexible in the control of devices and implements a wide range of sensors to capture various parameters.

Mrs. Paul Jasmin Rani, Jason Baktha kumar, Praveen Kumar B, Praveen Kumar U. Santhosh Kumar. “Voice Controlled Home Automation System using Natural Language Processing and Internet of Things”, INCC Transactions on National Security Science and Computation, 2018

The paper focuses on the construction of a fully functional voice-based home automation system that uses the Internet of Things, Artificial Intelligence, and Natural Language Processing (NLP) to provide a cost-effective, efficient way to work together with home appliances using various technologies such as GSM, NFC, etc. it implements a seamless integration of all the appliances to a central console, i.e., the mobile device. The prototype uses Arduino MK1000, known as Genuine MK1000. The NLP in this project gives the user the freedom to interact with the home appliances with his/her own voice and normal language rather than complicated computer commands. The appliances are connected to the mobile device through an Arduino Board that establishes the concept of the Internet of Things. The Arduino Boards are interfaced with the appliances and programmed in such a way that they respond to mobile inputs.

Jonathan J. Hull, Berna Erol, Jamey Graham, Qifa Ke, Hidenobu Kishi, Jorge Moraleda, Daniel G. Van Olsth Research Gate “Paper-Based Augmented Reality”, Research Gate Transaction on Camera and Paper documentation, 2007

A new method for augmenting paper documents with electronic information is described that does not modify the format of the paper document in any way.

Applicable to both commercially printed documents as well as documents that are output from PC's, the technique we call Paper-Based Augmented Reality substantially improves the utility of paper. We describe the recognition technology that makes this possible as well as several applications. An implementation on a camera phone is discussed that lets users retrieve data and access links from paper documents to electronic data. Recognition is performed at 4 frames per second on a Treo 700w and support is provided for several user applications, including "clickable paper" – printed web pages whose appearance is unchanged but that can be navigated with a camera phone. Linking patches of text will increase a lot of security reasons.

Mikko Kytö, Barrett Ens, Thammathip Piumsomboon, Gun A. Lee, and Mark Billinghurst, Research Gate "Pinpointing: Precise Head- and Eye-Based Target Selection for Augmented Reality", Research Gate Transactions on Augmented Reality in Computer Science Engineering, 2018

Head and eye movement can be leveraged to improve the user's interaction repertoire for wearable displays. Head movements are deliberate and accurate and provide the current state-of-the-art pointing technique. Eye gaze can potentially be faster and more ergonomic but suffers from low accuracy due to calibration errors and the drift of wearable eye-tracking sensors. This work investigates precise, multimodal selection techniques using head motion and eye gaze. A comparison of speed and pointing accuracy reveals the relative merits of each method, including the achievable target size for robust selection. We demonstrate and discuss example applications for augmented reality, including compact menus with deep structure, and a proof-of-concept method for online correction of calibration drift.

Subramani Roy Chowdhary, A. Divija, G. V N Vijayalakshmi, P. Vamsi, JARTMS "TOUCHLESS HOME AUTOMATION USING AUGMENTED REALITY", JARTMS Transactions on Augmented Reality and Web Interface, 2021.

As human life is heading towards a busy schedule it becomes necessary to automate our home appliances. Human error is something that cannot be completely erased. With the busy schedule in hand, there is defiantly a possibility

of missing something that may be trivial to us but can result in a catastrophe. For these reasons, Our Project home automation can increase efficiency, security, and reliability. AR has recently evolved for the automation of various electrical appliances by popping virtual objects into the real world.

Smart Agriculture Monitoring System Using Internet of Things (IoT) Dr. Tanuj Manglani Department of EE Arya College of Engineering and Research Centre, Jaipur, Rajasthan, India Aman Vaishnav Department of CSE Arya College of Engineering and Research Centre. IEEE 2021

In this paper presented the smart agriculture monitoring system using internet of things (IoT). The proposed model is implemented with the help of different sensors and IoT equipments. This proposed design helps the farmer to continuously monitor the crop to increase the production of the crops in the land. Presently assuming the framework mode set into manual mode, microcontroller unit will send the information to the android gadget. From the client's android cell phone client at that point get a pop-up message from the framework. What's more, in that notice client will see the sensor information. In the wake of checking the current condition, the client will at that point make an appropriate move. In another manner on the off chance that the client set the activity mode programmed, the framework will take sensors information and figure them. In the event that the worth decided a best for water system, water pump will be turned on. Then again, assuming the worth isn't ideal for water system, the framework turned off. For IoT-based control of agricultural advancement conditions, sensor centers that can exist in various zones, entryways that can connect with the web, or little terminals that interlock with their frames through network interfaces.

“Sensor Based Smart Agriculture with IoT Technologies” A Review Dr.M.Pyingkodi , Dr.K.Thenmozhi ,K.Nanthini , M.Karthikeyan , Dr.Suresh Palarima IEEE 2020

IoT sensor-based agriculture is now widely recognized as the new age of farming. It will shortly exchange traditional farming practices also enhance agricultural output It is already in use in large farms and industrialized nations, but if properly implemented in countries such as India, China, and Africa, it may quickly solve world hunger. This contemporary farming approach has the potential to usher in a

new era of green revolution and pave the way for progress in a nation like India, where farming is the major sector and market is based on it.

OPEN PROBLEMS IN EXISTING SYSTEM:

Water acts as a great essential life source. It is a well-known fact that life began with water and water cleanliness is a very important aspect of life to survive on earth. But, the byproducts of science laid their monstrous footsteps as pollutants. Most of these pollutants are toxic and are affecting adversely the water resources (wells, lakes, rivers, and sea), living organisms in the water, and all dependent organisms. Also, due to the carelessness in the use & maintenance of water bodies, millions of tons of plastics and other floating wastes are dumped into the water daily. Most of the time, the water bodies are cleaned manually with human labor which requires a lot of time and cost. Manual pest spray might increase in a lot of medical complexes hence it is mandatory to know the amount of bug activity in each field.



Fig 2.1 : Pesticide Spraying In Agriculture

- There is no proper tracking of water source utility
- Automatic Pump and Station AMC is not monitored
- Physical emergency overflow cut off is still in practice
- Unwanted distribution of water flow is not monitored
- Terrain distribution of water sources is not precise
- Bug Control mechanism and effective methodologies are not properly discussed

- Lots of unwanted pesticide automatic sprayers were implemented which is much harmful
- There is no proper data maintained during pesticide spray

MAJOR DISADVANTAGES TO OVERCOME:

Gestures have the power to act as effective tools of communication and also in Real-time Automation. Augmented Reality is that it creates unique digital experiences that blend the best of the digital and physical worlds. Augmented Reality presents information in neatly summarized digital snippets. AR can help users take quick decisions without cognitive overload. Augmented Reality is the integration of digital information with the user's environment in real-time. Our proposed system can show the live status of a machine to the operator through his AR glasses or tablet. In our augmented reality application, the position of the label is determined by images, text labels, or by QR codes. This methodology will make the user benefit by showing the live data from the unit when the camera of the label scans the corresponding card. This in turn makes it possible to control your environment with augmented reality Augmented Reality would be the one and only choice to handle impossible situations that occurs due to fire, flood, and other Natural Calamities.



Fig 2.1 : Visual Examination of the System

CHAPTER - 3

REQUIREMENT ANALYSIS

PROJECT OBJECTIVES:

- A.** Design of an Independent Smart Agricultural Management System (SAMS)
- B.** Wireless Control of Home appliances (Switch, Cloud, and, Bluetooth)
- C.** Monitoring status of appliances
- D.** Secure connection channels between application and ESP32 Master
- E.** Controlled by any device capable of Wi-Fi (Android, iOS, PC)
- F.** An extensible platform for future enhancement

Design of an independent (SAMS):

To formulate the design of an interconnected network of home appliances to be integrated into the HAS. The objective to account for every appliance and its control to be automated and integrated into the network further formulated into the HAS.

Wireless control of home appliances (Switch, Cloud, and, Bluetooth):

To develop the application that would include features of the switch and/or voice modes to control the applications.

Monitoring status of appliances:

Being able to view the status of home appliances on the application, in order to have a better HAS.

Secure connection channels between application and ESP32 Master:

Use of secure protocols over Wi-Fi so that other devices are prevented to achieve control over the HAS. Secure connections are obtained by SSL over TCP and SSH.

Controlled by any device capable of Wi-Fi (Android, iOS, PC):

To achieve flexibility in control of the home appliances, a device capable of Wi-Fi connectivity will be able to obtain secure control.

SCOPE OF THE INVESTIGATION:

The aim is to design a prototype that establishes wireless remote control over a network of home appliances. The application is designed to run on an android device providing features like switch-mode control, voice command control, and a provision to view the status of the devices on the application itself. Considering its wide range of applications, the following are the scope of this prototype. The system can be implemented in homes, small offices, and malls as well, being in charge of control of the electrical appliances. For remote access to appliances on the internet or intranet. The appliances in the above-mentioned environment can be controlled intra-network or can be accessed via the internet. The development of a technology-friendly environment. The system incorporates the use of technology and making SAS. By the use of day-to-day gadgets, we can utilize them from a different perspective.

Management of any project can be briefly disintegrated into several phases. Our project has been decomposed into the following phases:

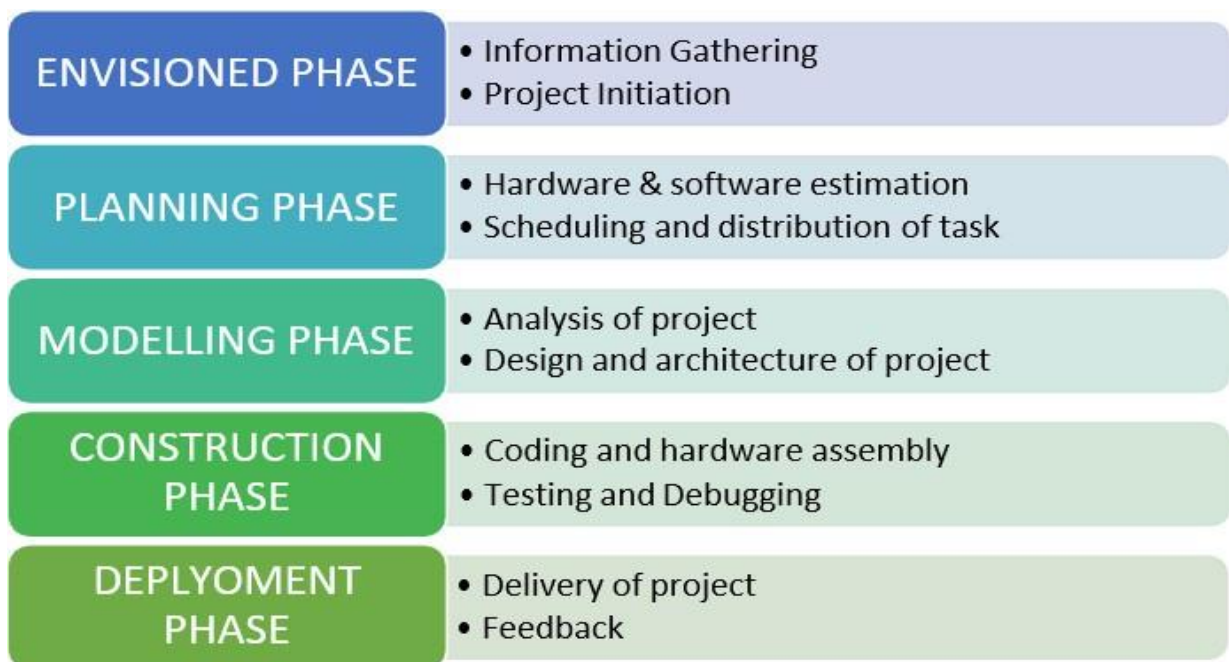


Fig 3.1 : Model of Phases in Project Management

REQUIREMENT ANALYSIS:

IoT Technologies and Safety:

Several communication protocols and technologies cater to and meet the specific functional requirements of the IoT system.

A. Bluetooth

B. Zigbee

C. Z-Wave

D. Wi-Fi

E. Cellular

F. NFC

G. LoRaWAN

Bluetooth

Bluetooth is a short-range IoT communication protocol/technology that is profound in many consumers product markets and computing. It is expected to be key for wearable products, in particular, again connecting to the IoT albeit probably via a smartphone in many cases. The new Bluetooth Low-Energy (BLE) – or Bluetooth Smart, as it is now branded – is a significant protocol for IoT applications. Importantly, while it offers a similar range to Bluetooth it has been designed to offer significantly reduced power consumption.

Zigbee

ZigBee is similar to Bluetooth and is majorly used in industrial settings. It has some significant advantages in complex systems offering low-power operation, high security, robustness, and high and is well-positioned to take advantage of wireless control and sensor networks in IoT applications. The latest version of ZigBee is the recently launched 3.0, which is essentially the unification of the various ZigBee wireless standards into a single standard.

Z-Wave

Z-Wave is a low-power RF communications IoT technology that is primarily designed for home automation for products such as lamp controllers and sensors

among many other devices. A Z- Wave uses a simpler protocol than some others, which can enable faster and simpler development, but the only maker of chips is Sigma Designs compared to multiple sources for other wireless technologies such as ZigBee and others.

Wi-Fi

Wi-Fi connectivity is one of the most popular IoT communication protocols, often an obvious choice for many developers, especially given the availability of Wi-Fi within the home environment within LANs. There is a wide existing infrastructure as well as offering fast of data transfer and the ability to handle high quantities of data. Currently, the most common Wi-Fi standard used in homes and many businesses is 802.11n, which offers a range of hundreds of megabits per second, which is fine for file transfers but maybe too power-consuming for many IoT applications.

Cellular

Any IoT application that requires operation over long distances can take advantage of GSM/3G/4G cellular communication capabilities. While cellular is clearly capable of sending high quantities of data, especially for 4G, the cost and also power consumption will be too high for many applications. But it can be ideal for sensor-based low-bandwidth-data projects that will send very low amounts of data over the Internet.

NFC

NFC (Near Field Communication) is an IoT technology. It enables simple and safe communications between electronic devices, specifically smartphones, allowing consumers to perform transactions in which one does not have to be physically present. It helps the user to access digital content and connect electronic devices. Essentially it extends the capability of contactless card technology and enables devices to share information at a distance that is less than 4cm.

SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

IoT Software:

A. Data Collection

B. Device Integration

C. Real-Time Analytics

D. Application and Process Extension

A. Data Collection

This software manages sensing, measurements, light data filtering, light data security, and aggregation of data. It uses certain protocols to aid sensors in connecting with real-time, machine-to-machine networks. Then it collects data from multiple devices and distributes it in accordance with settings. It also works in reverse by distributing data over devices.

B. Device Integration

Software supporting integration binds (dependent relationships) all system devices to create the body of the IoT system. It ensures the necessary cooperation and stable networking between devices. These applications are the defining software technology of the IoT network because, without them, it is not an IoT system.

C. Real-Time Analysis

These applications take data or input from various devices and convert it into feasible actions or clear patterns for human analysis. They analyze information based on various settings and designs in order to perform automation-related tasks or provide the data required by the industry.

D. Application and Process Extension

These applications extend the reach of existing systems and software to allow a wider, more effective system. They integrate predefined devices for specific purposes such as allowing certain mobile devices or engineering instruments access. It supports improved productivity and more accurate data collection.

CHAPTER - 4

PROPOSED METHODOLOGY

METHODOLOGY

The block diagram gives the functionality of the overall project. The Arduino unit is the microcontroller or the main controlling unit of the system. The user uses the mobile application in setting commands for the functioning of the appliances. The mobile application interprets the command form in the user in voice or switch mode and sends a signal to the Arduino unit, over a wireless network established by Wi-Fi communication. Hence the Wi-Fi module (actually inbuilt into Arduino), helps the microcontroller establish Wi-Fi communication with a device and take commands from an application over the wireless network. And the final appliances are physically connected. There is a power supply unit that powers the microcontroller, the relay as well as the final appliances. There is also a display unit that displays the status of the application.

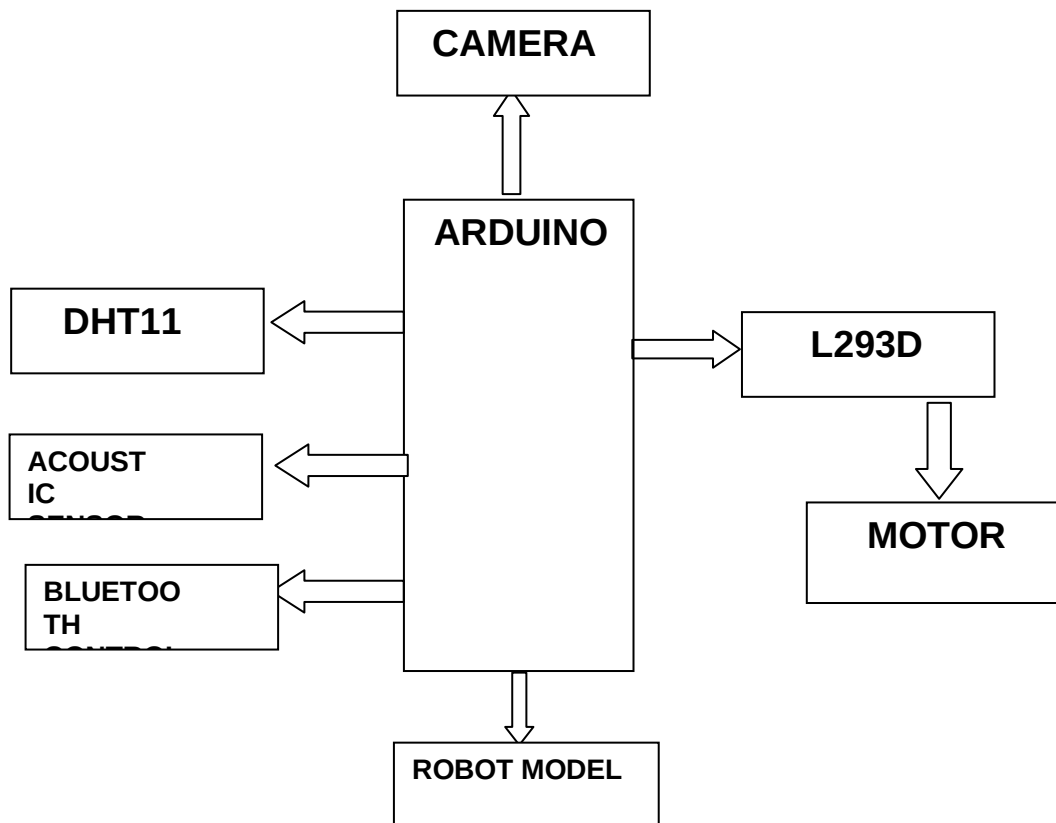


Fig 4.1: Block Diagram of Proposed Methodology

- The method that using is an Autonomous bot.
- This serves as a helping hand to the farmers by replacing the Manual farming technique with the modern farming technique. In our proposed method, the bot will control by using WiFi which is technique.
- It reduces the workload of the farmers by sprinkling pesticides on their own.
- This system consists of two major units Input unit and a Spray Control Unit
- Input Unit - The purpose of the input unit is to give the power supply all over the circuit and it will act as a storage device too for saving the pesticide liquid.
- Spray and Control Unit - The main component of the circuit is Cp2102 WiFi Controller and it is also known as the Heart of the robot. A driver circuit is connected to the Controller, which is used for the movement of the robot. A pesticide pump is also used for transferring liquid pesticide to the sprayer head from the storage tank.

ARDUINO UNO:



Fig 4.2 : Arduino Uno R3

Overview

A versatile microcontroller which is widely used throughout for various electronic projects is the Arduino Uno dependent on the ATmega 328P. It has 14 computerized i/o pins , 6 analogous inputs, a Universal Serial Bus, and a reset button. Comprises of all that anticipated to assist the microcontroller; interface Personal Computer as well as provide power to via a battery or a connector. There are various applications for which an Arduino board can be use. Such as:

1. Modelling prototypes and systems of electronic nature
2. Endless possibilities of DIY (Do-It-Yourself) Undertakings
3. Simple beginner level usage
4. Undertakings requiring multiple I/O pins

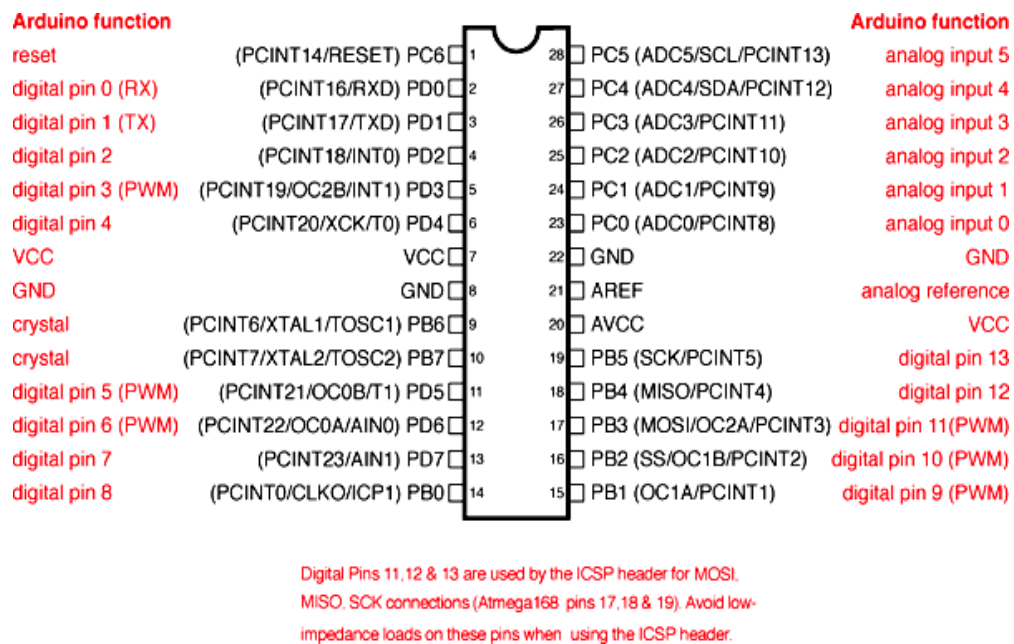


Fig 4.3: ATmega328P and Arduino Uno Pin Mapping

The Arduino Uno is AVR RISC centered board, which operates sandwiched between 1.8 to 5.5 Volts, conglomerates as much as nearly 32 kb memory, three versatile timers/counters with compare modes, a 2-wire serial interface, programmable watchdog timer, 1 KB EEPROM, 2 KB SRAM, and power saving modes. To program the Arduino Uno board, the IDE is to be used. Arduino can be used to communicate with a computer, another Arduino board or supplementary microcontrollers. There are two LEDs on the Arduino board which will flash to indicate when data is being transmitted

Table 4.1: Characteristics of AT Mega328P

Parameter	Value
CPU type	8-bit AVR
Performance	20 MIPS @ 20 MHz
SRAM	2 KB
Pin count	28 / 32 pin
Max. operating frequency	20 MHz
No. of touch channels	16
Max. I/O pins	23

ULTRA SONIC SENSOR

Ultrasonic sensors work on a principle similar to sonar which evaluates distance of a target by interpreting the echoes from ultrasonic sound waves. This ultrasonic module measures the distance accurately which provides 0cm - 400cm with a gross error of 3cm. Its compact size, higher range and easy usability make it a handy sensor for distance measurement and mapping. The module can easily be interfaced to micro controllers where the triggering and measurement can be done using two pin. The sensor transmits an ultrasonic wave and produces an output pulse that corresponds to the time required for the burst echo to return to the sensor. By measuring the echo pulse width, the distance to target can easily be calculated.

FEATURES

- Ideal for use in Beam Detection Systems
- Sensitivity: >-65dB
- Nominal Frequency: 40kHz
- Maximum Driving Voltage: 20Vrms
- Operating Temperature: -20oC to +70oC
- Dimensions: 16.2mm Diameter, 12.2mm High excluding pins, Pins - 10mm High

PIN DESCRIPTION

With the sensor oriented as shown alongside, locate Pin-1 as the 1st pin on the left hand side.



Fig 4.4: Ultrasonic Sensor

Table 4.2 : Male Connector Side

Pin No.	Signal
1	VCC (5V supply)
2	Trigger Pulse Input
3	Echo Pulse Output
4	GND (0V)

Table 4.3 : Specifications

Parameter	Specification
Dimensions	45 x 20 x 15 mm
Pin-out Pitch	2.54mm male berg
Interface	VCC, GND, SDA,SCL

OPERATION

Time interval between sending trigger signal and receiving echo signal.

Formula: $\mu\text{S} / 58 = \text{centimeters}$ or $\mu\text{S} / 148 = \text{inch}$;

NOTE

- The module is not suggested to connect directly to live supply, if connected, the GND terminal should be connected to the module first, otherwise, it will affect the normal work of the module.
- When testing objects, the range of area should not be less than 0.5 square meters and the surface plane as smooth as possible, otherwise, it will affect the results of measuring.

DC MOTOR

It is a turning electrical gadget that transforms electrical energy (Direct Current) into mechanical energy. A loop of coil inside the motor delivers a field of magnetisation that makes revolving movement when a voltage (DC in nature) provided to its ends. There is also a shaft made of iron present within the motor and is engulfed by a loop of wire. The aforementioned shaft consists of two immovable magnets (North and South) on the two sides which results in both a repulsive and attracting ultrasonic, thereby delivering torque.

Type of motors

The following are examples of direct current motors:

- Brushed Motor
- Brushless Motor
- Gear Motors (Planetary)
- Gear Motors (Spur)
- Stepper Motors
- Coreless Brushless Motors
- Servo Motor
- Gear head Motor

Gear Motors – common electrical motors transform electrical energy into mechanical energy. A gear motor is an amalgamation of a motor and gearbox. The addition of a gear head to a motor reduces the speed and increases the torque output. The most important parameters in regards to gear motors are:

- Speed (rpm rating)
- Torque (lb/in)
- Efficiency (%)

In order to select the appropriate gear motor for our application we must analyse:

- Load
- Speed

Bluetooth Module HC-05

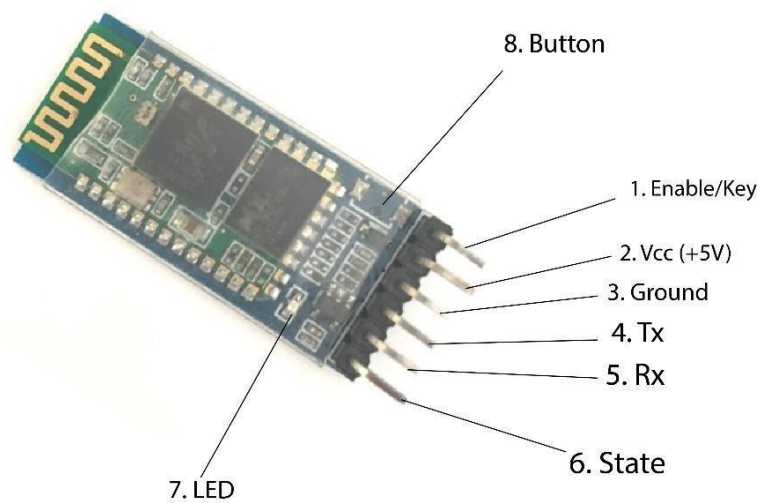


Fig 4.5: HC-05 Module

BATTERY

In electricity, a battery is a device consisting of one or more electrochemical cells that convert stored chemical energy into electrical energy. A common power source for many household and industrial applications. There are two types of batteries: primary batteries (disposable batteries), which are designed to be used once and discarded, and secondary batteries (rechargeable batteries), which are designed to be recharged and used multiple times. Batteries come in many sizes from miniature cells used to power hearing aids and wristwatches to battery banks the size of rooms that provide standby power for telephone exchanges and computer data centers.

A battery is a device that converts chemical energy directly to electrical energy. It consists of a number of voltaic cells; each voltaic cell consists of two half-cells connected in series by a conductive electrolyte containing anions and cations. One half-cell includes electrolyte and the electrode to which anions migrate, i.e., the anode or negative electrode; the other half-cell includes electrolyte and the electrode to which cations migrate, i.e., the cathode or positive electrode. In the redox reaction that powers the battery, cations are reduced at the cathode, while anions are oxidized at the anode. The electrodes do not touch each other but are electrically connected by the electrolyte. Some cells use two half-cells with different electrolytes. A separator between half-cells allows ions to flow, but prevents mixing of the electrolytes.

- Primary batteries irreversibly transform chemical energy to electrical energy. When the initial supply of reactants is exhausted, energy cannot be readily restored to the battery by electrical means.
- Secondary batteries can be recharged; that is, they can have their chemical reactions reversed by supplying electrical energy to the cell, restoring their original composition.
- Some types of primary batteries used, for example, for telegraph circuits, were restored to operation by replacing the components of the battery consumed by the chemical reaction. Secondary batteries are not indefinitely rechargeable due to

dissipation of the active materials, loss of electrolyte and internal corrosion.

PRIMARY BATTERIES

Primary batteries can produce current immediately on assembly. Disposable batteries are intended to be used once and discarded. These are most commonly used in portable devices that have low current drain, are used only intermittently, or are used well away from an alternative power source, such as in alarm and communication circuits where other electric power is only intermittently available.

Disposable primary cells cannot be reliably recharged, since the chemical reactions are not easily reversible and active materials may not return to their original forms. Battery manufacturers recommend against attempting recharging primary cells.

SECONDARY BATTERIES

Secondary batteries must be charged before use; they are usually assembled with active materials in the discharged state. Rechargeable batteries or secondary cells can be recharged by applying electric current, which reverses the chemical reactions that occur during its use. Devices to supply the appropriate current are called chargers or rechargers. The oldest form of rechargeable battery is the lead–acid battery. This battery is notable in that it contains a liquid in an unsealed container, requiring that the battery be kept upright and the area be well ventilated to ensure safe dispersal of the hydrogen gas produced by these batteries during overcharging. The lead–acid battery is also very heavy for the amount of electrical energy it can supply. Despite this, its low manufacturing cost and its high surge current levels make its use common where a large capacity is required or where the weight and ease of handling are not concerns.

BATTERY CHARGERS

We supply several related chargers that are very easy to use. All of the units have an internal computer-on-a-chip that makes possible the exact performance required for lead-acid batteries, whether they are manually-filled, sealed electrolyte, or Gell-cells. There are no switches to set. All battery chargers are fully regulated and automatic; continuously shorting the battery leads will not harm the chargers.

1. CONNECTING THE CHARGER- The sequence of connections is very important for the charger's computer operation. First, connect the color-coded leads to the battery. Then, plug the charger into the AC outlet. If you mistakenly connect the battery backwards (negative to positive) or short the charger leads together, no damage or sparks occur—only a RED light comes on to indicate the possible mistake.

2. INITIAL – When the charger is initially plugged in, the unit's computer sends intermittently, a small current into the battery and then monitors the battery's response to determine its charge state. A RED light may come on briefly if the battery was excessively discharged, but it should go off within the first few minutes.

3. BULK CHARGE MODE – When the charger determines that the battery can safely take a charge, it goes into the bulk mode where the full capacity of the charger is applied to the battery until it is 80% charged. In this mode, the YELLOW light comes on.

4. ABSORPTION CHARGE MODE- Once the battery is 80% charged, the computer within the charger tapers the current from 100% to 33% of charger capacity until the battery reaches full charge. The light will remain YELLOW during this phase.

5. FLOAT CHARGE- After full charge has been reached, a GREEN light comes on, indicating the battery is ready for use. The computer switched to the 4 th part of the cycle, that of a temperature compensated float charger. The unit is now in its maintenance mode, keeping the battery in its ideal charged state. Once the battery has reached this ideal state, all of our chargers maintain this condition without causing “boil-off “, heating, or permitting the battery's lead plates to sulfate. This means you can leave your

batteries connected to the charger overnight, over the weekend, or indefinitely, knowing that when you take them to the field, they are in the ideal state.

The black and white figure of charging characteristics above is for illustration. How long your batteries take to charge depends on which model is used and how depleted the battery was. The example shown above is for a 15 Amp-Hr battery that has been depleted 10 Amps being charged by a 1.5 Amp-Hr charger. The process will take about 12 hours.

BATTERY TYPES:

- **LEAD-ACID BATTERIES:** Our lead-acid batteries use a gelled-electrolyte in a sealed, high impact plastic case. The construction of these high power-to-weight batteries eliminates the need to ever water or worry about spilling acid. The units are completely rechargeable and weigh 4 and 8 lbs for the 6 and 12 VDC versions, respectively. The 6 V version is rated at 12 amp-hours and the 12 V version at 10 amp-hours. See our battery product page for additional battery sizes. Please contact us with questions regarding appropriate chargers, batteries, or charging techniques for your application.
- **DISPOSABLE BATTERIES:** For individuals desiring the convenience of disposable batteries, all of our 6 volt traps are available with a 4-D Cell Battery Holder (Model 1.50) that will hold 4 D-cells – enough capacity for 1 or 2 night's operations, depending on trap and battery type.
- **SOLAR-POWERED BATTERY CHARGERS:** Our solar-powered battery chargers are available in 6 or 12 VDC units depending upon voltage and amperage requirements of your traps. Solar Charger Model SA is for 6 volt traps; it is capable of replenishing the power used during the previous night during the following day. The Solar Charger Model CD is for 12 volt traps and can recharge a single battery which has been used to power a backlight trap for 10 or more hours. Both units complete with power cords, stands, and instructions. Both are protected from reverse-polarity connection.

POWER SUPPLY

Power supply is a reference to a source of electrical power. A device or system that supplies electrical or other types of energy to an output load or group of loads is called a power supply unit or PSU.

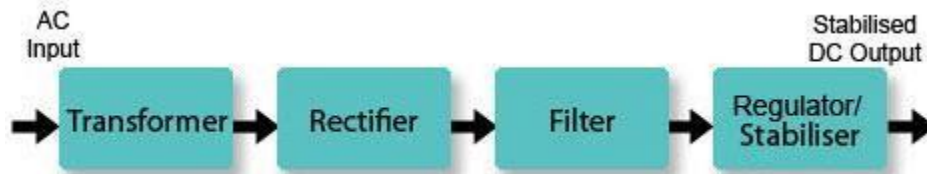


Fig:4.6 Power Supply

Basic power supply the input power transformer has its primary winding connected to the mains (line) supply. A secondary winding, electromagnetically coupled but electrically isolated from the primary is used to obtain an AC voltage of suitable amplitude, and after further processing by the PSU, to drive the electronics circuit it is to supply.

The transformer stage must be able to supply the current needed. If too small a transformer is used, it is likely that the power supply's ability to maintain full output voltage at full output current will be impaired. With too small a transformer, the losses will increase dramatically as full load is placed on the transformer.

As the transformer is likely to be the most costly item in the power supply unit, careful consideration must be given to balancing cost with likely current requirement. There may also be a need for safety devices such as thermal fuses to disconnect the transformer if overheating occurs, and electrical isolation between primary and secondary windings, for electrical safety.

THE RECTIFIER STAGE

Rectifier circuit is used, to convert the AC input is converted to DC. The full wave bridge rectifier uses four diodes arranged in a bridge circuit to give full wave rectification without the need for a centre-tapped transformer. An additional advantage is that, as two diodes are conducting at any one time, the diodes need only half the reverse breakdown voltage capability of diodes

used for half and conventional full wave rectification. The bridge rectifier can be built from separate diodes or a combined bridge rectifier can be used.

The current paths on positive and negative half cycles of the input. It can be seen that on each half cycle, opposite pairs of diodes conduct, but the current through the load remains in the same polarity for both half cycles.

FILTER

A typical power supply filter circuit can be best understood by dividing the circuit into two parts, the reservoir capacitor and the low pass filter. Each of these parts contributes to removing the remaining AC pulses, but in different ways.

Electrolytic capacitor used as a reservoir capacitor, so called because it acts as a temporary storage for the power supply output current. The rectifier diode supplies current to charge a reservoir capacitor on each cycle of the input wave. The reservoir capacitor is large electrolytic, usually of several hundred or even a thousand or more microfarads, especially in mains frequency PSUs. This very large value of capacitance is required because the reservoir capacitor, when charged, must provide enough DC to maintain a steady PSU output in the absence of an input current; i.e. during the gaps between the positive half cycles when the rectifier is not conducting.

The action of the reservoir capacitor on a half wave rectified sine wave. During each cycle, the rectifier anode AC voltage increases towards V_{pk} . At some point close to V_{pk} the anode voltage exceeds the cathode voltage, the rectifier conducts and a pulse of current flows, charging the reservoir capacitor to the value of V_{pk} .

Once the input wave passes V_{pk} the rectifier anode falls below the capacitor voltage, the rectifier becomes reverse biased and conduction stops. The load circuit is now supplied by the reservoir capacitor alone.

Of course, even though the reservoir capacitor has large value, it discharges as it supplies the load, and its voltage falls, but not by very much. At some point during the next cycle of the mains input, the rectifier input voltage rises above the voltage on the partly discharged capacitor and the reservoir is re-charged to the peak value V_{pk} again.

REGULATOR

Voltage regulator ICs are available with fixed or variable output voltages. They are also rated by the maximum current they can pass. Negative voltage regulators are available, mainly for use in dual supplies. Most regulators include some automatic protection from excessive current and overheating.

The LM78XX series of three terminal regulators is available with several fixed output voltages making them useful in a wide range of applications. One of these is local on card regulation, eliminating the distribution problems associated with single point regulation. The voltages available allow these regulators to be used in logic systems, instrumentation, HiFi, and other solid state electronic equipment. Although designed primarily as fixed voltage regulators these devices can be used with external components to obtain adjustable voltages and current.

1. Positive regulator

- input pin
- ground pin
- output pin
- It regulates the positive voltage

2. Negative regulator

- Ground pin
- Input pin
- Output pin

CHAPTER-5

SOFTWARE IMPLEMENTATION

EMBEDDED C

Embedded C is the most popular embedded software language in the world. Most embedded software is written in Embedded C. It is a set of language extensions for the C Programming language by the C Standards committee to address commonality issues that exist between C extensions for different embedded systems.

Historically, embedded C programming requires nonstandard extensions to the C language in order to support exotic features such as fixed-point arithmetic, multiple distinct memory banks, and basic I/O operations. The C programming language is perhaps the most popular programming language for programming embedded systems.

Most C programmers are spoiled because they program in environments where not only there is a standard library implementation, but there are frequently a number of other libraries available for use. The cold fact is, that in embedded systems, there rarely are many of the libraries that programmers have grown used to, but occasionally an embedded system might not have a complete standard library, if there is a standard library at all. Few embedded systems have capability for dynamic linking, so if standard library functions are to be available at all, they often need to be directly linked into the executable. Oftentimes, because of space concerns, it is not possible to link in an entire library file, and programmers are often forced to "brew their own" standard C library implementations if they want to use them at all. While some libraries are bulky and not well suited for use on microcontrollers, many development systems still include the standard libraries which are the most common for C programmers.

C remains a very popular language for micro-controller developers due to the code efficiency and reduced overhead and development time. C offers low-level control and is considered more readable than assembly. Many free C compilers are available for a wide variety of development platforms.

The compilers are part of an IDEs with ICD support, breakpoints, single-stepping and an assembly window. The performance of C compilers has

improved considerably in recent years, and they are claimed to be more or less as good as assembly, depending on who you ask. Most tools now offer options for customizing the compiler optimization. Additionally, using C increases portability, since C code can be compiled for different types of processors.

BASIC CONCEPTS OF EMBEDDED C AND EMBEDDED PROGRAMMING

Embedded C, even if it's similar to C, and embedded languages in general requires a different kind of thought process to use. Embedded systems, like cameras or TV boxes, are simple computers that are designed to perform a single specific task. They are also designed to be efficient and cheap when performing their task.

For example, they aren't supposed to use a lot of power to operate and they are supposed to be as cheap as possible. As an embedded system programmer, you will have simple hardware to work with. You will have very little RAM, ROM and very little processing power and stack space. The reason why most embedded systems use Embedded C as a programming language is because Embedded C lies somewhere between being a high level language and a low level language. Embedded C, unlike low level assembly languages, is portable.

It can run on a wide variety of processors, regardless of their architecture. Unlike high level languages, Embedded C requires less resources to run and isn't as complex. Some experts estimate that C is 20% more efficient than a modern language like C++. Another advantage of Embedded C is that it is comparatively easy to debug.

EMBEDDED C COMPILERS

There are a variety of different compilers on the market, manufactured by different companies that use Embedded C. One of the more popular ones is the Keil compiler. Because of this, Embedded C is also sometimes known as Keil C.

Embedded C has several keywords that are not present in C (learn more about the concept of keywords in this course). These keywords are associated with operations needed by microprocessors. You will need to be familiar with all of them to be able to write Embedded C programs.

Unsigned char data a;

Here, the unsigned char declaration is like a normal C declaration. We just added the data keyword, which tells the microcontroller to store the unsigned char a in the internal data memory.

bdata: The bdata keyword lets you store a declared variable in the bit addressable memory. Take a look at this example:

Unsigned char bdata a;

This is similar to the data declaration we showed you above. You have to access bdata variables in a different way, however.

Using: This keyword lets you execute a function by letting it access a register bank. There are three possible values: 1, 2 and 3.

EMBEDDED SYSTEMS PROGRAMMING

- Embedded systems programming is different from developing applications on a desktop computers. Key characteristics of an embedded system, when compared to PCs, are as follows:
- Embedded devices have resource constraints (limited ROM, limited RAM, limited stack space, less processing power)
- Components used in embedded system and PCs are different; embedded systems typically use smaller, less power consuming components. Embedded systems are more tied to the hardware.

EMBEDDED SYSTEMS USING DIFFERENT TYPE OF LANGUAGES:

- Machine Code
- Low level language, i.e., assembly
- High level language like C, C++, Java, Ada, etc.
- Application level language like Visual Basic, scripts, Access, etc.

DIFFERENCE BETWEEN C AND EMBEDDED C

- Though C and embedded C appear different and are used in different contexts, they have more similarities than the differences. Most of the constructs are same; the difference lies in their applications.

- C is used for desktop computers, while embedded C is for microcontroller based applications. Accordingly, C has the luxury to use resources of a desktop PC like memory, OS, etc. While programming on desktop systems, we need not bother about memory. However, embedded C has to use with the limited resources (RAM, ROM, I/Os) on an embedded processor. Thus, program code must fit into the available program memory. If code exceeds the limit, the system is likely to crash.
- Compilers for C(ANSI C) typically generate OS dependant executables. Embedded C requires compilers to create files to be downloaded to the microcontrollers/microprocessors where it needs to run. Embedded compilers give access to all resources which is not provided in compilers for desktop computer applications.
- Embedded systems often have the real-time constraints, which is usually not there with desktop computer applications.
- Embedded systems often do not have a console, which is available in case of desktop applications.

So, what basically is different while programming with embedded C is the mindset; for embedded applications, we need to optimally use the resources, make the program code efficient.

The Arduino Integrated Development Environment or Arduino Software (IDE) contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them.

WRITING SKETCHES

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text.

The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information.

The bottom right hand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

NB: Versions of the Arduino Software (IDE) prior to 1.0 saved sketches with the extension.pde. It is possible to open these files with version 1.0, you will be prompted to save the sketch with the.ino extension on save.

FILE

- **Verify**
Checks your code for errors compiling it.
- **New**
Creates a new instance of the editor, with the bare minimum structure of a sketch already in place.
- **Open**
Allows loading 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 saving 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 boot loader on the board; you will need to use Tools > Burn Boot loader 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. Please note that this command will NOT burn the fuses. To do so Tools -> Burn Bootloader command must be executed.

- **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 Boot loader

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 new ATmega microcontrollers (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 set 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 File 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.

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, boot loaders, 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.

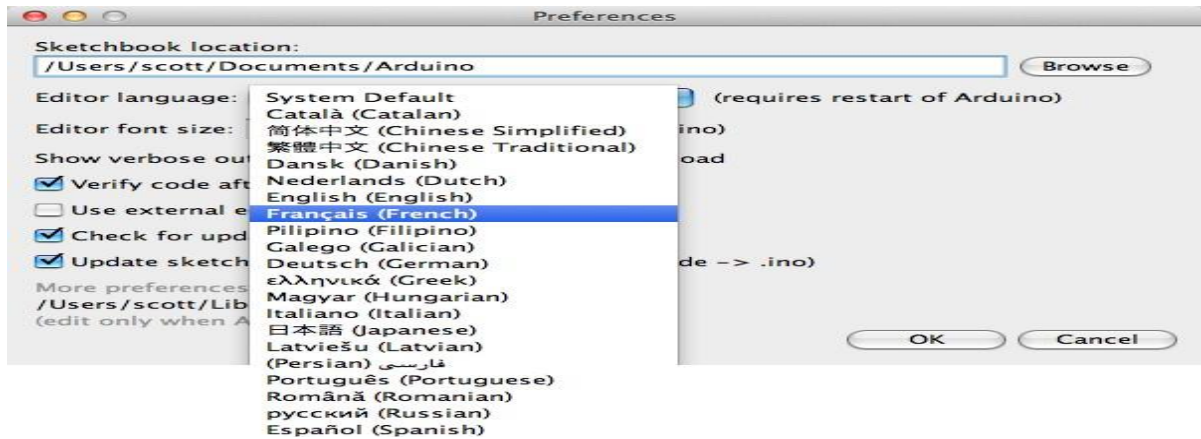


Fig 5.1:Arduino Software Languages

Since version 1.0.1, the Arduino Software (IDE) has been translated into 30+ different languages. By default, the IDE loads in the language selected by your operating system. (Note: on Windows and possibly Linux, this is determined by the locale setting which controls currency and date formats, not by the language the operating system is displayed in.)

If you would like to change the language manually, start the Arduino Software (IDE) and open the Preferences window. Next to the Editor Language there is a dropdown menu of currently supported languages. Select your preferred language from the menu, and restart the software to use the selected language. If your operating system language is not supported, the Arduino Software (IDE) will default to English.

You can return the software to its default setting of selecting its language based on your operating system by selecting System Default from the Editor Language drop-down. This setting will take effect when you restart the Arduino Software (IDE).

Similarly, after changing your operating system's settings, you must restart the Arduino Software (IDE) to update it to the new default language.

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 boot loader 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 boot loader. 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 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.

- **ArduinoYùn**
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.
- **ArduinoDiecimila 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 Micro**
An ATmega32u4 running at 16 MHz with auto-reset, 12 Analog In, 20 Digital I/O and 7 PWM.

- **ArduinoEsplora**
An ATmega32u4 running at 16 MHz with auto-reset.
- **Arduino Mini w/ ATmega328**
An ATmega328 running at 16 MHz with auto-reset, 8 Analog In, 14 Digital I/O and 6 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.
- **ArduinoFio**
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 boot loader burned (4 KB) includes codes to initialize the on-board Bluetooth module, 6 Analog In, 14 Digital I/O and 6 PWM..
- **Lily Pad Arduino USB**
An ATmega32u4 running at 8 MHz with auto-reset, 4 Analog In, 9 Digital I/O and 4 PWM.
- **Lily Pad 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.
- **Arduino NG or older w/ ATmega168**
An ATmega168 running at 16 MHz without auto-reset. Compilation and upload is equivalent to Arduino Diecimila or Duemilanove w/ ATmega168, but the bootloader burned has a slower timeout (and blinks the pin 13 LED three times on reset); 6 Analog In, 14 Digital I/O and 6 PWM.
- **Arduino Robot Control**
An ATmega328 running at 16 MHz with auto-reset.

- Arduino Robot Motor

An ATmega328 running at 16 MHz with auto-reset.

CHAPTER 6

CIRCUIT AND ITS WORKING

CIRCUIT

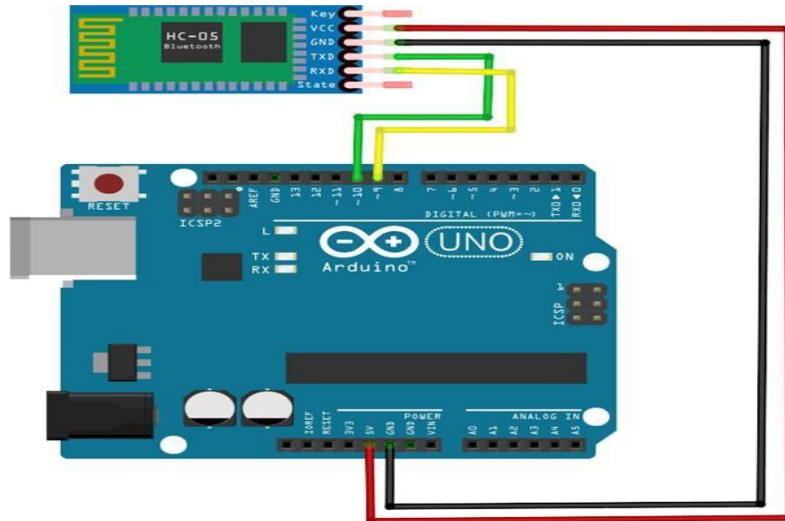


Fig 6.1: Circuit Diagram

The components used in the making of this prototype have been specifically considered with their respective ratings and functionality in mind. The sensors, motors, mounting brackets, Arduino, Motor driver shield etc. have been packaged and mounted on the chassis to effectively use the space at offer, making the vehicle compact and easy to handle.

All the connections seen above are the basic signal and power pins used for each component connected to the Arduino. Li-ion Battery pack (3000mAh) cells are used to supply the peak load from all the sensors and components working simultaneously.

The maximum rated rpm at the highest configured speed for each motor is 255rpm and can be varied to 0rpm is necessary. This alteration in speed can be achieved directly through a slider control within the Application. Direct access buttons have been configured to switch between the different modes of operation and directly communication to the Bluetooth module and update the Code loop. The circuit is connected to ensure no reconfiguration or unexpected load is shared to any sensor and damage the circuit continuity.

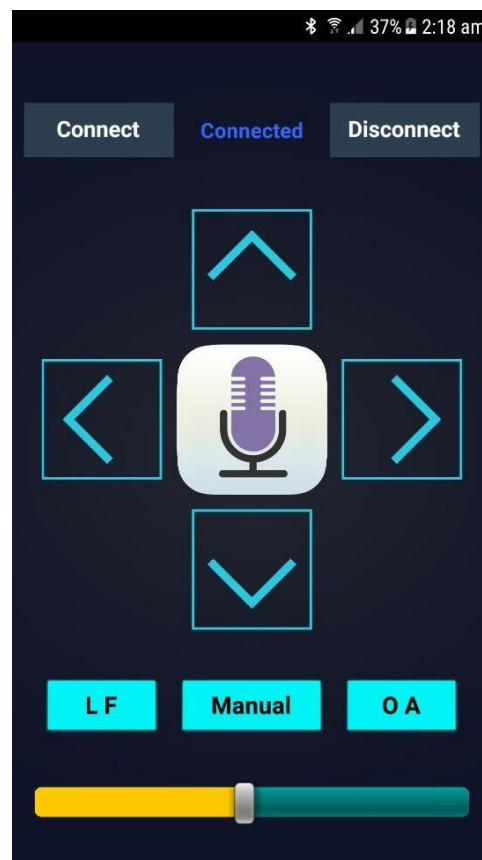


Fig 6.1:Application Interface

WORKING

Bluetooth is the key to this entire integration and prototype. Once the power is in and all the sensors are available, the vehicle can be connected to the application by first connecting the Bluetooth module to the device of control Smartphone and then navigating within the app for the final step. Once completed the interface show the connected notification right in middle.

Once connected the vehicle can operate in 4 distinct modes.

- Voice Controlled Mode
- Bluetooth Control (RC) Mode
- Line-Following Mode
- Obstacle Avoiding Mode

Each mode can be switched between using the dedicated button control to toggle between the same. Once a particular mode is under use, the vehicle will solely follow the instruction set of that particular logic unless ultrasonic to switch into manual mode. Manual mode is the default mode for the vehicle. Input from the application is fed to the module which instructs the Arduino to send out appropriate signals to control the movement of all for motors individually. The Arduino board is the main controller for all the functions. Every input signal from the Bluetooth module is sent to the Arduino and decoded based on the specific mode it is configured to work in that instance. The signals are processed on FIFO logic. Any change in the Application or while execution of a specific command will complete the loop and go into default manual mode to avoid and unexpected movement or uncontrolled manoeuvres.

Android Application Development

A. Front End

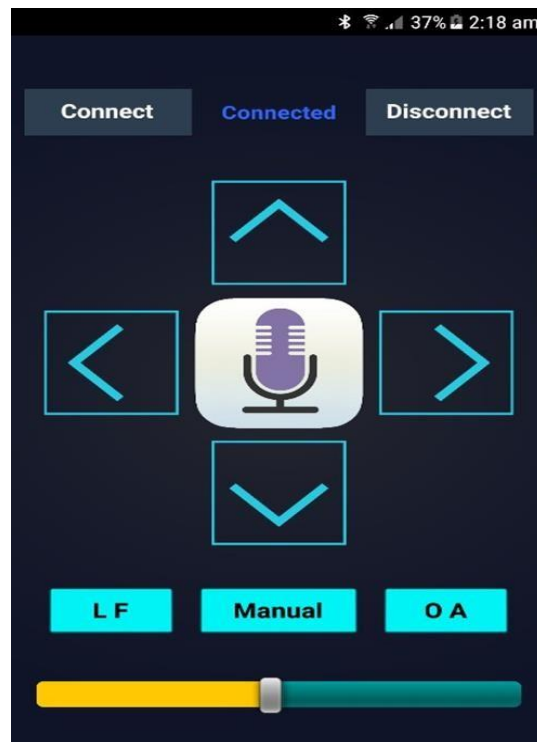


Fig 6.2: User Interface of Application

The buttons Connect and Disconnect pertain to establishing communication between the phone and the Bluetooth module attached to the prototype. In between these two buttons, the status of the connection is presented. If the communication is established and a success, the status bar will read “Connected” else will read “Disconnected”.

The arrow keys are present to control the vehicle manually by tapping said buttons. In the centre is the icon with a microphone on it. This is the Voice control mode which utilises the Google speech recognition software and interprets specific commands such as “Go Forward”, “Go Backward”, “Turn Left”, and “Turn Right” which can be given into the microphone of the phone by anyone.

On the bottom of the application, there are three buttons which are as follows:

1. LF – Switch to Line Following mode
2. Manual – Switch to Remote Control (RC) mode
3. OA – Switch to Obstacle Avoiding mode

Apart from these buttons, there is a slider below the mode changing buttons. This slider allows the user to change the speed at which the prototype will function in RC mode. This varies from 0 RPM to 255 RPM.

B.Back End

Using the MIT App developer site, we constructed the back end of the application. As the site is easy to use and accessible, the backend is in block structure. The backend of the application though immense, is in the form of blocks that can be connected to one another simple by dragging and clicking. This is the feature provided by the MIT App Inventor which allows anyone to create Android applications.

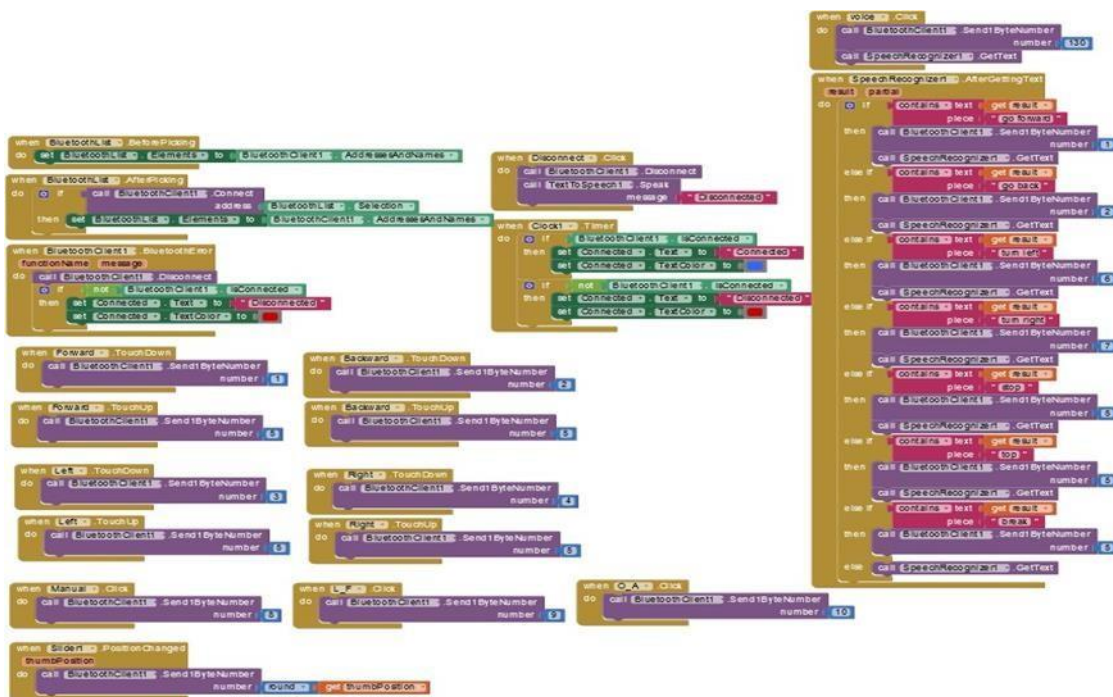


Fig 6.3: Application Backend

CHAPTER-7

RESULTS AND DISCUSSION

RESULTS:

The hardware design built has been the primary focus to ensure the robustness and durability of the prototype. Base chassis is a Plastic Composite selected to mount all the electronic components without any electrical interference from a board. Precision cut holes were drilled out in the chassis to pass along the wiring between the components placed on the bottom and top of the board. Securing the connection were also a part of the design evaluation.

Extensive use of strong adhesive and mounts to ensure all the components do not move around and cause any damage to other components around it. Packaging the vehicle in this small chassis involved multiple iteration and changes to ensure the level of durability we aimed for. Motor Driver, Arduino Uno and all the individual components, i.e. the MEMS Sensor, Bluetooth HC-05 Module and the Ultrasonic Sensor, have been either connected or soldered with the help of a soldering iron.

Every connection made is to ensure the continuity of the circuit is maintained for all the operating modes in one sequence hence eliminating the need for multiple wires or connection for individual modes. Each sensor terminal is distinguished with distinct positive, negative, and Signal connection to make troubleshooting easy and quick. In case of any distress signal from the Arduino or the Motor Driver Shield, the connection terminals can be quickly checked to eliminate the issue.

Mounting of motors are measured setup to provide tight manoeuvring control and agile movement of the vehicle in varied environments. Ground clearance and width of the vehicle has been considered as an essential part of the design to make the package as compact as possible. For varied condition and requirements the dimension, load carrying capacity, speed and other such factors can be modified and adjusted for the necessary application.



Fig 7.1: Initial Build

The desired mode of functionality for the vehicle are as follows:

- Bluetooth RC mode (Manual)
- Path Following mode
- Obstacle avoiding mode
- Voice Control Mode

All the modes have delivered the desired functionality in varied environments and terrain as tested. The application developed to interact with the module and control its functionality, built with the intention to keep the UI simple and intuitive. Direct and easy connectivity to the module with low signal communication via Bluetooth ensures .

SUMMARY:

It is evident from this project work that an individual control home automation system can be cheaply made from low-cost locally available components and can be used to control multifarious home appliances ranging from the security lamps, and the television to the air conditioning system and even the entire house lighting system. And better still, the components required are so small and few that they can be packaged into a small inconspicuous container. The designed home automation system was tested a number of times and certified to control different home appliances used in the lighting system, air conditioning system, home entertainment system, and many more. Hence, this system is scalable and flexible. All the relevant data are updated to the cloud where we monitor the environmental parameters on the cloud web server for the enhancement of the smart agricultural plant monitoring system. The system is highly secured and stabilized by the web server protocols.

FURTHER ENHANCEMENT AND FUTURE SCOPE

Looking at the current situation we can build a cross-platform system that can be deployed on various platforms like iOS, and Windows. The limitation to control only several devices can be removed by extending the automation of all other home appliances. The prototype can include sensors to implement automatic control of the home appliances like; an LDR that can sense daylight and switch lamps accordingly, a PIR to detect motion and be used for security purposes making an alarm buzz, or a DHT11 sensor that senses ambient temperature and humidity of the atmosphere and switch fan/air conditioner accordingly.

CONCLUSION

The prototype built to inculcate various different functionalities such as Obstacle Avoidance, Line Following, Bluetooth control, Voice Control showcases various environments such vehicle can be deployed in. Automating manual process by using the base of this vehicle principle can open a wide variety of use cases this implementation can be deployed in.

Minimising physical interaction with the vehicle and automating the process of switching between modes serves as a boon for manual processes that can now be controlled remotely. The performance of the prototype built also highlights the importance of durability and robustness to help improve the longevity and functionality of such vehicle in different scenarios.

The concept of this prototype can be implemented in with various logics to better suit the controllability as per the defined use case, operational hours, frequency of interaction, degree of automation etc. The model showcases the added benefits of inculcating modes of operation as an intuitive, easy to control, and wireless option to increase the functionality of such vehicles which can be deployed in controlled environments to help in improving and reducing manual efforts for various applications.

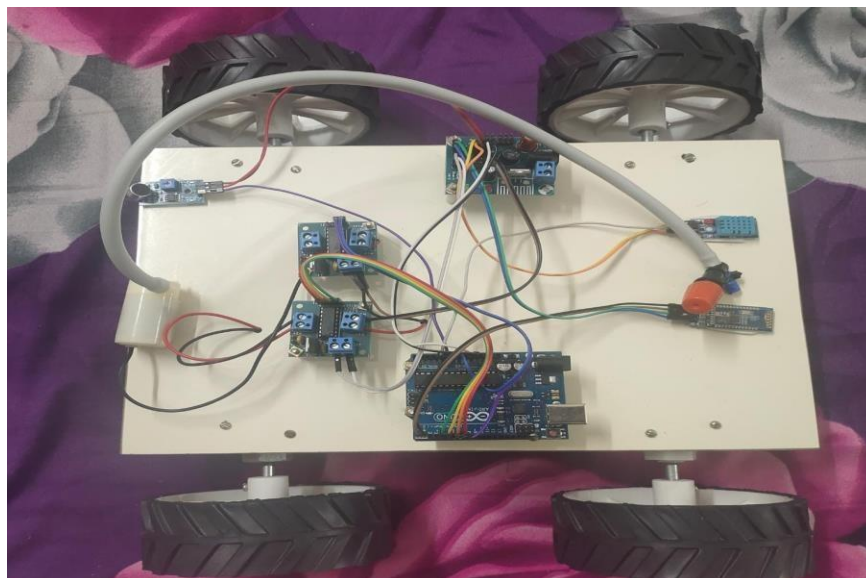


Fig 7.2: Finished Prototype