A

MAJOR PROJECT REPORT ON

SMART AUTOMATIC AQUARIUM MONITORING SYSTEM USING IOT

Submitted in partial fulfillment of the requirement for the award of degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

SUBMITTED BY

G. LOKESH REDDY	218R1A04E9
G. RAHUL	218R1A04F0
G. RAHUL	218R1A04F2
G. SAMUL JASHWANTH	218R1A04F3

Under the Esteemed Guidance of

Dr. K. Sravan Abhilash
Associate Professor.

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

CMR ENGINEERING COLLEGE

UGC AUTONOMOUS

(Approved by AICTE, Affiliated to JNTU Hyderabad, Accredited by NBA)

Kandlakoya(V), Medchal(M), Telangana – 501401

(2024-2025)

CMR ENGINEERING COLLEGE

UGC AUTONOMOUS

(Approved by AICTE, Affiliated to JNTU Hyderabad, Accredited by NBA) Kandlakoya (V), Medchal Road, Hyderabad - 501 401

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that Major project work entitled **"SMART AUTOMATIC AQUARIUM MONITORING SYSTEM USING IOT"** is being Submitted by **G. LOKESH** bearing Roll No: **218R1A04E9, G. RAHUL** bearing Roll No**: 218R1A04F0, G. RAHUL** bearing Roll No: **218R1A04F2, G. SAMUEL** bearing Roll No: **218R1A04F3** in BTech IV-II semester, Electronics and Communication Engineering is a record bonafide work carried out by then during the academic year 2024-25. The results embodied in this report have not been submitted to any other University for the award of any degree.

INTERNAL GUIDE	**HEAD OFTHE DEPARTMENT**
Dr. K. SRAVAN ABHILASH	**Dr. SUMAN MISHRA**
Associate Professor	**Professor**

EXTERNAL EXAMINER

ACKNOWLEDGEMENTS

We sincerely thank the management of our college **CMR Engineering College** for providing required facilities during our project work. We derive great pleasure in expressing our sincere gratitude to our Principal **Dr. A. S. Reddy** for his timely suggestions, which helped us to complete the project work successfully. It is the very auspicious moment we would like to express our gratitude to **Dr. SUMAN MISHRA,** Head of the Department, ECE for his consistent encouragement during the progress of this project.

We take it as a privilege to thank our major project coordinator **Dr. T. SATYANARAYANA**, Associate Professor, Department of ECE for the ideas that led to complete the project work and we also thank him for his continuous guidance, support and unfailing patience, throughout the course of this work. We sincerely thank our project internal guide **DR.K. SRAVAN ABHILASH,** Associate Professor of ECE for guidance and encouragement in carrying out this project work.

DECLARATION

We hereby declare that the Major project entitled **"SMART AUTOMATIC. AQUARIUM MONITORING SYSTEM USING IOT"** is the work done by us in campus at **CMR ENGINEERING COLLEGE,** Kandlakoya during the academic year 2024-2025 and is submitted as major project in partial fulfillment of the requirements for the award of degree of **BACHELOR OF TECHNOLOGY in ELECTRONICS AND COMMUNICATION ENGINEERING FROM JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD.**

G. LOKESH REDDY	**(218R1A04E9)**
G. RAHUL	**(218R1A04F0)**
G. RAHUL	**(218R1A04F2)**
G. SAMUEL JASHWANTH	**(218R1A04F3)**

ABSTRACT

The Smart Automatic Aquarium Monitoring System (SAAMS) leverages the Internet of Things (IoT) to provide a comprehensive and automated solution for maintaining optimal conditions in aquariums. This system is designed to monitor and regulate key environmental parameters such as water temperature, pH levels, oxygen levels, and lighting. Utilizing an array of sensors and a central control unit, the system continuously gathers data and makes real-time adjustments to ensure a healthy habitat for aquatic life. By integrating IoT technologies, SAAMS provides aquarium owners with remote access and control, enabling them to monitor the status of their aquarium from anywhere at any time through a dedicated mobile application. The core functionality of SAAMS includes automated feeding schedules, water quality monitoring, and alert notifications. The system is equipped with sensors that detect changes in water quality, triggering automated responses such as water changes or chemical adjustments to maintain a stable environment. Additionally, the system includes an automated feeding mechanism that dispenses food at scheduled intervals, reducing the risk of overfeeding or underfeeding. The integration of real-time alerts ensures that owners are promptly informed of any critical changes, such as drastic temperature shifts or low oxygen levels, allowing for timely intervention.

The SAAMS offers a user-friendly interface that provides detailed analytics and historical data, helping owners understand trends and make informed decisions about aquarium management. This system is particularly beneficial for hobbyists, aquarists, and commercial aquarium operators who require consistent and reliable monitoring. By automating routine tasks and providing actionable insights, SAAMS not only enhances the well-being of aquatic organisms but also simplifies the maintenance process, making it accessible even to those with limited experience in aquarium care. The system represents a significant advancement in the integration of IoT in aquaculture, promising improved efficiency, reduced manual labor, and enhanced aquatic ecosystem management.

CONTENTS

CHAPTER	**PAGE NO.**
CERTIFICATE	i
ACKNOWLEDGEMENTS	ii
DELARATION	iii
ABSTRACT	iv
CONTENTS	v
LIST OF THE FIGURES	vii
LIST OF THE TABLES	ix
CHAPTER-1	
1. INTRODUCTION	**1**
1.1 HISTORY AND FUTURE	3
1.2 REAL TIME SYSTEMS	3
1.3 OVERVIEW OF EMBEDDED SYSTEM ARCHITECTURE	6
CHAPTER-2	
2. LITERATURE SURVEY	**9**
2.1 SMART AQUARIUM BASED MICRO CONTROLLER BY BUILD PRIJO SEMBODO AND NOVENDRA GEOFANDA PRATAMA	9
2.2 SMART AQUARIUM DESIGN USING RASPBERRY PI AND IN ANDRIOD BASED BY KHAIRUNISA MARDENI YUDA	10
2.3 EMBEDDED INTRODUCTION	12
CHAPTER-3	
3. SOFTWARE REQUIREMETS	22
3.1 SOFTWARE TOOLS	22
CHAPTER-4	
4. HARDWARE REQUIREMENTS & BLOCK DIAGRAM	**31**
4.1 POWER SUPPLY	31
4.2 AURDINO UNO	32
4.2.1 Summary	33
4.2.2 Schematic & Reference Design	34
4.2.3 Power	34
4.2.4 Vin	34
4.2.5 Memory	34
4.2.6 Input and Output	35
4.2.7 Schematic & Reference Design	35
4.2.8 Power	35
4.2.9 Communication	36

4.2.10 Power Supply 36
4.3 DISPLAY 43
4.4 WIFI 45
4.5 WIFI ALLIANCE 47
4.6 DHY-11 47
4.7 PH SENSOR 49
4.8 TEMPERATURE SENOR 50
4.9 INTERNET OF THINGS -IOT 52
4.10 CHARACTERISTICS 55
4.11 IOT ARCHITECTURE 56
4.12 FUTER TECHIONOLOGICAL DEVELOPMENS FOR IOT 59

CHAPTER-5

5. BLOCK DIAGRAM & WORKING 63

5.1 BLOCK DIAGRAM 63
5.2 WORKING 64

CHAPTER-6

6.RESULT ANALYSIS AND REPORT 68

6.1 RESULTS 68
6.2 ADVANTAGES 71

CHAPTER-7

7. CONCLUSION AND FUTURE SCOPE 74

7.1 CONCLUSION 74
7.2 FUTURE SCOPE 75

REFERENCES 78

APENDEX 80

CODE 82

LIST OF THE FIGURES

Figure no.	Figure name	Page no.
Fig: 2.1	Emdedded OS	12
Fig: 2.2	Embedded system	13
Fig: 2.3	Blocks of embedded system	14
Fig: 2.4	Embedded system hard ware	15
Fig: 2.5	Embedded design-process-steps	16
Fig: 2.6	Hard ware and software of embedded	18
Fig: 2.7	Applications of embedded systems	20
Fig: 2.8	Logic gates	20
Fig: 3.1	Mini b-cable	22
Fig: 3.2	Opening arduino	24
Fig: 3.3	Launch arduino	25
Fig: 3.4	Launch arduino ide	26
Fig: 3.5	Launch arduino ide tool bar	26
Fig: 3.6	Arduino IDE	27
Fig: 3.7	Program uploading	28
Fig: 3.8	Sketch-nov299	29
Fig: 4.1	Arduino UNO	32
Fig: 4.2	Block diagram of power supply	37
Fig: 4.3	Circuit diagram of power supply	37
Fig: 4.4	Pin diagram of lcd	38
Fig: 4.5	Buzzer	45
Fig: 4.6	Dht11 sensor	49
Fig: 4.7	Ph sensor	50
Fig: 4.8	Temperature sensor	50
Fig: 4.9	Vision of IOT	53
Fig: 4.10	Characteristics of internet of things	55
Fig: 4.11	Future technological developments for IOT	59
Fig: 5.1	Description of components	63
Fig: 5.2	Block diagram of smart aquarium	63
Fig: 6.1	Smart automatic aquarium system	68
Fig: 6.2	Morning time feeder ON	68
Fig: 6.3	Feeder ON & OFF in mobile application	69
Fig: 6.4	Feeder TIM	69

Fig: 6.5	PH value detector	70
Fig: 6.6	PH value, temperature and humidity output In mobile application	70

LIST OF TABLES

Table no	List of table names	Page no
Table 2.1	Design parameters & functions of an embedded systems	19
Table 4.1	Pin description of lcd	39
Table 4.2	Command operation of lcd	41
Table 4.3	Command list of lcd	42

CHAPTER 1
INTRODUCTION

As the trend of keeping pets increases, people are keeping all sorts of animals at home and it is not a new concept in any way. The strong connection between pets and their owners is evident from a report by Micheal Gross . All of these animals require special care and sometimes humans cannot attend to their needs and these days there are many people fighting to protect the ethical rights of animals like PETA . Out of these animals, fish require the utmost care because their environment is completely different from land animals, so they need specific conditions like a temperature range, pH, suitable oxygen and CO2 levels. Normally aquariums have oxygen pumps, heaters, and filters. This is not enough or equivalent to the natural habitat. Many scientists have worked on the effects of meteorological and hydrological diversity with respect to the spatiotemporal scales . Maintaining these conditions is very hard manually, so automating this process will greatly reduce the fish death rate and will create great convenience for the owners.

The main cause of death for fish in aquariums and fish farms is the inability to take care. This is not only true for this project but in fact automation is one of the most productive way of doing things with ease . This project is designed to decrease the labor time and can be controlled from anywhere, such as a mobile phone or PC etc. IoT is the technology that enables communication between devices; this minimizes human interaction with the machine, automates normal or routine tasks and even makes them faster as the machine can also communicate with the other machines it is dependent upon.

This makes an entire network of smart machines that are independent of human beings and this will also monitor the breakdown of the product or mistakes that happen as the IoT based machines are constantly monitoring through sensors. Internet of things is a blend of many concepts that are put together to make an autonomous product that is easy to use and is diverse enough to perform the end task . IoT is helping manufacturers make better products and diagnose problems much more easily.Internet of things is a blend of many concepts that are put together to make an autonomous product that is easy to use and is diverse enough to perform the end task.

Introduction of Embedded System

An Embedded System is a combination of computer hardware and software, and perhaps additional mechanical or other parts, designed to perform a specific function. A good example is the microwave oven. Almost every household has one, and tens of millions of them are used every day, but very few people realize that a processor and software are involved in the preparation of their lunch or dinner.

This is in direct contrast to the personal computer in the family room. It too is comprised of computer hardware and software and mechanical components (disk drives, for example). However, a personal computer is not designed to perform a specific function rather; it is able to do many different things. Many people use the term general-purpose computer to make this distinction clear. As shipped, a general-purpose computer is a blank slate; the manufacturer does not know what the customer will do wish it. One customer may use it for a network file server another may use it exclusively for playing games, and a third may use it to write the next great American novel.

Frequently, an embedded system is a component within some larger system. For example, modern cars and trucks contain many embedded systems. One embedded system controls the anti-lock brakes, other monitors and controls the vehicle's emissions, and a third displays information on the dashboard. In some cases, these embedded systems are connected by some sort of a communication network, but that is certainly not a requirement.

At the possible risk of confusing you, it is important to point out that a general-purpose computer is itself made up of numerous embedded systems. For example, my computer consists of a keyboard, mouse, video card, modem, hard drive, floppy drive, and sound card-each of Which is an embedded system? Each of these devices contains a processor and software and is designed to perform a specific function. For example, the modem is designed to send and receive digital data over analog telephone line. That's it and all of the other devices can be summarized in a single sentence as well.

If an embedded system is designed well, the existence of the processor and software could be completely unnoticed by the user of the device. Such is the case for a

This could be done by replacing the combination with a custom integrated circuit that performs the same functions in hardware. However, a lot of flexibility is lost when a design is hard-cooled in this way. It is much easier, and cheaper, to change a few lines of software than to redesign a piece of custom hardware.

1.1 HISTORY AND FUTURE

Given the definition of embedded systems earlier is this chapter; the first such systems could not possibly have appeared before 1971. That was the year Intel introduced the world's first microprocessor. This chip, the 4004, was designed for use in a line of business calculators produced by the Japanese Company BusCom. In 1969, Busicom asked Intel to design a set of custom integrated circuits-one for each of their new calculator models. The 4004 was Intel's response rather than design custom hardware for each calculator, Intel proposed a general- purpose circuit that could be used throughout the entire line of calculators. Intel's idea was that the software would give each calculator its unique set of features.

The microcontroller was an overnight success, and its use increased steadily over the next decade. Early embedded applications included unmanned space probes, computerized traffic lights, and aircraft flight control systems. In the 1980s, embedded systems quietly rode the waves of the microcomputer age and brought microprocessors into every part of our kitchens (bread machines, food processors, and microwave ovens), living rooms (televisions, stereos, and remote controls), and workplaces (fax machines, pagers, laser printers, cash registers, and credit card readers).

It seems inevitable hat the number of embedded systems will continue to increase rapidly. Already there are promising new embedded devices that have enormous market potential; light switches and thermostats that can be central computer, intelligent air-bag systems that don't inflate when children or small adults are present, pal-sized electronic organizers and personal digital assistants (PDAs), digital cameras, and dashboard navigation systems. Clearly, individuals who possess the skills and desire to design the next generation of embedded systems will be in demand for quite some time.

1.2 REAL TIME SYSTEMS

One subclass of embedded is worthy of an introduction at this point. As commonly defined, a real-time system is a computer system that has timing constraints. In other words, a real-time system is partly specified in terms of its ability to make certain calculations or decisions in a timely manner. These important calculations are said to have deadlines for completion. And, for all practical purposes, a missed deadline is just as bad as a wrong answer.

The issue of what if a deadline is missed is a crucial one. For example, if the real-time system is part of an airplane's flight control system, it is possible for the lives of the passengers and crew to be endangered by a single missed deadline. However, if instead the system is involved in satellite communication, the damage could be limited to a single corrupt data packet. The more severe the consequences, the more likely it will be said that the deadline is "hard" and thus, the system is a hard real-time system. Real-time systems at the other end of this discussion are said to have "soft" deadlines.

However, if instead the system is involved in satellite communication, the damage could be limited to a single corrupt data packet. The more severe the consequences, the more likely it will be said that the deadline is "hard" and thus, the system is a hard real-time system. Real-time systems at the other end of this discussion are said to have "soft" deadlines.

All of the topics and examples presented in this book are applicable to the designers of real-time system who is more delight in his work. He must guarantee reliable operation of the software and hardware under all the possible conditions and to the degree that human lives depend upon three system's proper execution, engineering calculations and descriptive paperwork.

Application Areas: Nearly 99 per cent of the processors manufactured end up in embedded systems. The embedded system market is one of the highest growth areas as these systems are used in very market segment- consumer electronics, office automation, industrial automation, biomedical engineering, wireless communication, Data communication, telecommunications, transportation, military and so on.

Consumer appliances: At home we use a number of embedded systems which include digital camera, digital diary, DVD player, electronic toys, microwave oven, remote controls for TV and air-conditioner, VCO player, video game consoles, video recorders etc Today's high- tech car has about 20 embedded systems for transmission control, engine spark control, air- conditioning, At home we use a number of camera, digital diary, DVD player. **Office automation:** The office automation products using embedded systems are copying machine, fax machine, key telephone, modem, printer, scanner etc.

Industrial automation: Today a lot of industries use embedded systems for process control. These include pharmaceutical, cement, sugar, oil exploration, nuclear energy, electricity generation and transmission. The embedded systems for industrial use are designed to carry out specific tasks such as monitoring the temperature, pressure, humidity, voltage, current etc., and then take appropriate action based on the monitored levels to control other devices or to send information to a centralized monitoring station. In hazardous industrial environment, where human presence has to be avoided, robots are used, which are programmed to do specific jobs. The robots are now becoming very powerful and carry out many interesting and complicated tasks such as hardware assembly.

Medical electronics: Almost every medical equipment in the hospital is an embedded system. These equipment include diagnostic aids such as ECG, EEG, blood pressure measuring devices, X-ray scanners; equipment used in blood analysis, radiation, colonoscopy, endoscopy etc. Developments in medical electronics have paved way for more accurate diagnosis of diseases.

Computer networking: Computer networking products such as bridges, routers, Integrated Services Digital Networks (ISDN), Asynchronous Transfer Mode (ATM), X.25 and frame relay switches are embedded systems which implement the necessary data communication protocols. For example, a router interconnects two networks. The two networks may be running different protocol stacks. The router's function is to obtain the data packets from incoming pores, analyze the packets and send them towards the destination after doing necessary protocol conversion. Most networking equipment's, other than the end systems (desktop computers) we use to access the networks, are embedded systems.

Telecommunications: In the field of telecommunications, the embedded systems can be categorized as subscriber terminals and network equipment. The subscriber terminals such as key telephones, ISDN phones, terminal adapters, web cameras are embedded systems. The network equipment includes multiplexers, multiple access systems, Packet Assemblers Dissemblers (PADs). The subscriber terminals such as key telephones, ISDN phones, terminal adapters, web cameras are embedded systems.

1.3 OVERVIEW OF EMBEDDED SYSTEM ARCHITECTURE

Every embedded system consists of custom-built hardware built around a Central Processing Unit (CPU). This hardware also contains memory chips onto which the software is loaded. The software residing on the memory chip is also called the ‘firmware’. The embedded system architecture can be represented as a layered architecture as shown in Fig. The operating system runs above the hardware, and the application software runs above the operating system. The same architecture is applicable to any computer including a desktop computer. However, there are significant differences. It is not compulsory to have an operating system in every embedded system.

For small appliances such as remote-control units, air conditioners, toys etc., there is no need for an operating system and you can write only the software specific to that application. For applications involving complex processing, it is advisable to have an operating system. In such a case, you need to integrate the application software with the operating system and then transfer the entire software on to the memory chip. Once the software is transferred to the memory chip, the software will continue to run fora long time you don’t need to reload new software.

Now, let us see the details of the various building blocks of the hardware of an embedded system. As shown in Fig. the building blocks are:

- Central Processing Unit (CPU)
- Memory (Read-only Memory and Random Access Memory)

- Input Devices
- Output devices
- Communication interfaces
- Application-specific circuitry

The operating system runs above the hardware, and the application software runs above the operating system. The same architecture is applicable to any computer including a desktop computer. However, there are significant differences. It is not compulsory to have an operating system in every embedded system.

- **Central Processing Unit (CPU):**

The Central Processing Unit (processor, in short) can be any of the following: microcontroller, microprocessor or Digital Signal Processor (DSP). A micro-controller is a low- cost processor. Its main attraction is that on the chip itself, there will be many other components such as memory, serial communication interface, analog-to digital converter etc. So, for small applications, a micro-controller is the best choice as the number of external components required will be very less. On the other hand, microprocessors are more powerful, but you need to use many external components with them. D5P is used mainly for applications in which signal processing is involved such as audio and video processing.

- **Memory:**

The memory is categorized as Random Access 11emory (RAM) and Read Only Memory (ROM). The contents of the RAM will be erased if power is switched off to the chip, whereas ROM retains the contents even if the power is switched off. So, the firmware is stored in the ROM.

- **Input devices:**

Unlike the desktops, the input devices to an embedded system have very limited capability. There will be no keyboard or a mouse, and hence interacting with the embedded system is no easy task. Many embedded systems will have a small keypad-you press one key to give a specific command. A keypad may be used to input only the digits. So, for small applications, a micro-controller is the best choice as the number of external components required will be very less. On the other hand, microprocessors are more powerful, but you need to use many external components with them. D5P is used mainly for applications in which signal processing is involved such as audio and video processing.

- **Output devices:**

The output devices of the embedded systems also have very limited capability. Some embedded systems will have a few Light Emitting Diodes (LEDs) to indicate the health status of the system modules, or for visual indication of alarms. A small Liquid Crystal Display (LCD) may also be used to display some important parameters.

- **Communication interfaces:**

The embedded systems may need to, interact with other embedded systems at they may have to transmit data to a desktop. To facilitate this, the embedded systems are provided with one or a few communication interfaces such as RS232, RS422, RS485, Universal Serial Bus (USB), IEEE 1394, Ethernet etc.

- **Application-specific circuitry:**

Sensors, transducers, special processing and control circuitry may be required fat an embedded system, depending on its application. This circuitry interacts with the processor to carry out the necessary work.

CHAPTER 2
LITERATURE SURVEY

2.1 SMART AQUARIUM BASED MICROCONTROLLER BY BUDI PRIJO SEMBODO AND NOVENDRA GEOFANDA PRATAMA

The researchers Budi Prijo Sembodo et al.[1] have created a smart aquarium system that has an Arduino-based feeding system that controls the servo motor as an open and close system as the exit of fish feed into the aquarium.The servo motor can automatically deliver fish feed that was controlled by Arduino and has a feed output of 12.5 grams that was adjusted to the state of the 7 decorative fishes and the daily feed demands of the ornamental fishes in the aquarium. With a 3-second delay, the servo motorcan feed fish automatically according to the daily meal demands in the morning at 07.00 Western Indonesian Time and at night at 19.00 Western Indonesian Time.

If the light intensity is below 028.7 Lux, which was after half-past 6 p.m., this LDR (Light Dependent Resistor) type light sensor will send a signal to the relay, which was turned on the 220-volt lamp, and if the light intensity was above 203.4 Lux, which was after 6 a.m., the 220- volt lamp will be turned off.

The Arduino controls the water pump as an automated drainer and water filler based on the turbidity of the water in the aquarium by receiving a signal from the Arduino. The drain water pump can drain 30 liters of water in two minutes, while the filling water pump can drain 28 liters in four minutes.

➢ Smart system for maintaining aquascape environment using internet of things-based light and temperature controller by Daniel Patricko Hutabarat, Rudy Susanto, Bryan Prasetya, Barry Linando, Senanayake Mudiyanselage Namal AroshaThe main objective of the researcher's Daniel Patricko Hutabarat et al. was to create a smart system based on an internet of things (IoT) application for a plant aquarium.In this study, the parameters to be controlled by the system are light intensity and temperature. The hardware used to develop this system is the ESP32 as the microcontroller, BH1750FVI as the light sensor, high power led (HPL) light-emitting diodes (LED) lamp as the light source, DS18B20 as a temperature sensor, the heater, and the 220 VAC fan that is used to raise and lower the temperature. The hardware used to develop this system is the ESP32 as the microcontroller, BH1750FVI as the light sensor, high power led (HPL) light-emitting diodes (LED) lamp as the light source, This study also developed an application that is used by the user to provide input to the system. The developed application is then installed on the user's smartphone and used to connect the user to the system via the internet. The ease of adding and removing devices used on the system is a capability that is also being developed in this smart system. The developed system can produce light intensity with an accuracy rate of 96% and always manage to keep the temperature within the predetermined range.

2.2 SMART AQUARIUM DESIGN USING RASPBERRY PI AND ANDROID BASED BY KHAIRUNISA, MARDENI, YUDA IRAWAN

The researchers Khairunisa et al. designed a smart aquarium device in order to feed fishes automatically, namely using Android-Based Raspberry Pi.This aquarium performed a variety of tasks, including automatic fish feeding over the internet network and management of the aquarium's ornamental lights. It employed a servo motor to operate the fish feeding valve and a relay as an on/off aquarium ornamental light to move the fish feeding valve.

If the user forgets to feed the fish, fish feed devices can feed them on a regular basis. The fish feeding valve is rotated by the servo motor, and the feeding operation is completed automatically.

- IoT Based Automatic Aquarium Monitoring System for Freshwater Fish by Mohammad Fahmi Suhaimi, Nurul Huda Mat Tahir, Safuan Naim Mohamad, Suzanna Ridzuan.The authors Mohammad Fahmi Suhaimi et al.created a project that was based on a computer-controlled system that detects physical changes in the water and keeps it in optimal condition. The aquarium will perform all the operations automatically including temperature control, pH control, turbidity control, feeding, and water level control. The aquarium's status was continuously transferred to the database via the IoT monitoring system, which users may check over the internet. The pH sensor module and the temperature module were then used to collect data for the freshwater fish monitoring system. To handle the data gathered from the sensor, the Arduino ESP8266 was employed as a controller. In the electrical box, the Arduino circuit and all sensor modules for this project are wired.The IoT platform used for this project is thinger.io. Also, when sensors detected any problem, they sent a notification to the IoT platform, which was marked as an alert and monitored automatically.

- **Aquarium Monitoring System Based on Internet of Things by Wen-Tsai Sung, Shuo-Chen Tasi, and Sung-Jung Hsiao**

The authors Wen-Tsai Sung et al.[5] created a remote monitoring system using IoT technology for the aquarium environment. The main control development platform for this system was a MediaTek LinkIt 7697, and remote monitoring components include temperature, illuminance, water level, and passive infrared sensor modules.The system uses a wireless sensor network to communicate and calculate the obtained physical sensing signals (WSN). The program was compiled and sent to the LinkIt 7697's built-in Wi-Fi communication module using BlocklyDuino, a web-based visual programming editor for Arduino. The results were shown on the Cloud Sandbox platform by the back-end computer. This platform allowed users to view the environmental data collected by each sensor in real-time. The system must keep the fish in the tank in a comfortable climate.

To identify the six main concerns of the people in handling technology like IoT. Among the six topics they considered the most of them are related to the Big data and Security. This gives us a prediction of what IoT has to change in order to secure data and handle every need of their users. In spite of the business intrigue that the IoT presents for huge information investigation, the difficulties looked by the restricted security of the present IoT gadgets is the real worry forthe overall population [8].

So as to offer advantages to the two individuals and the things in IoT, information mining innovations are incorporated with IoT advancements for enhancement of framework and expanding basic decision-making capacities. Data mining includes extraction of helpful information for finding new methods and possibly valuable examples from information and to make new calculations for the extraction of concealed data . It is anticipated that the accessibility of web is all over the place and online for every one of the general populations. With the progressions in numerous highlights to non-critical failure and viable power utilization of hubs and handset, IoT has encouraged internetwork, diverse gadgets and accessibility of information from anyplace.

Data mining includes extraction of helpful information for finding new methods and possibly valuable examples from information and to make new calculations for the extraction of concealed data . It is anticipated that the accessibility of web is all over the place and online for every one of the general populations. With the progressions in numerous highlights to non-critical failure and viable power utilization of hubs and handset.

In Lee et al The depiction is that there is yet an opportunity to get better in the field of IoT. This makes it very hard going after for associations to choose decisions in the light of IoT determination/utilization. It is perceived that there are three characterizations of IoT applications: watching and control, gigantic data and business assessment, and, information sharing and joint exertion. IoT can be utilized to show adventure openings and theory evaluation with NPV and real options. An exploration was directed in USA that analyzed five challenges in realizing IoT applications for different endeavor

2.3 EMBEDDED INTRODUCTION

An embedded system is a combination of computer hardware and software designed for a specific function or functions within a larger system. The systems can be programmable or with fixed functionality. Industrial machines, consumer electronics, agricultural and process industry devices, automobiles, medical equipment, cameras, household appliances,

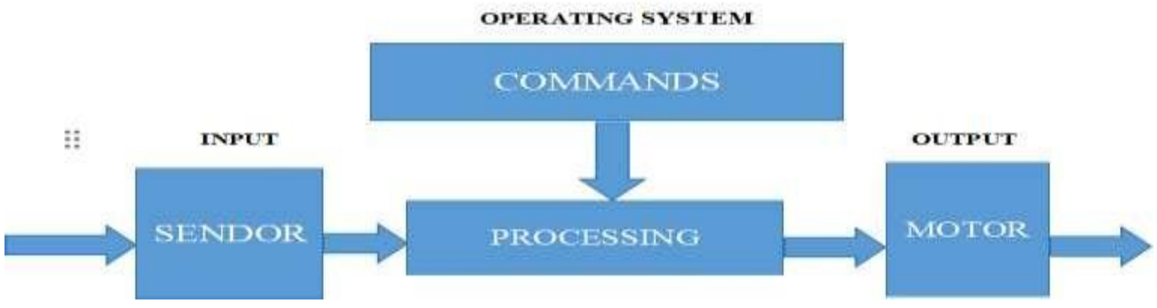


FIG: 2.1 Embedded OS

While embedded systems are computing systems, they can range from having no user interface (UI) -- for example, on devices in which the system is designed to perform a single task -- to complex graphical user interfaces (GUIs), such as in mobile devices. User interface scan include buttons, LEDs and touchscreen sensing. Some systems use remote user interfaces as well.

History Of Embedded Systems

Embedded systems date back to the 1960s. Charles Stark Draper developed an integrated circuit (IC) in 1961 to reduce the size and weight of the Apollo Guidance Computer, the digital system installed on the Apollo Command Module and Lunar Module. The first computer to use ICs, it helped astronauts collect real-time flight data.

In 1965, Autonoetic, now a part of Boeing, developed the D-17B, the computer used in the Minuteman I missile guidance system. It is widely recognized as the first mass-produced embedded system. When the Minuteman II went into production in 1966, the D-17B was replaced with the NS-17 missile guidance system, known for its high-volume use of integrated circuits. In 1968, the first embedded system for a vehicle was released; the Volkswagen 1600 used a microprocessor to control its electronic fuel injection system.

Also, in 1971, Intel released what is widely recognized as the first commercially available processor, the 4004. The 4-bit microprocessor was designed for use in calculators and small electronics, though it required eternal memory and support chips. The 8-bit Intel 8008, released in 1972, had 16 KB of memory; the Intel 8080 followed in 1974 with 64 KB of memory. The 8080's successor, x86 series, was released in 1978 and is still largely in use today.

In 1987, the first embedded operating system, the real-time VxWorks, was released by Wind River, followed by Microsoft's Windows Embedded CE in 1996. By the late 1990s, the first embedded Linux products began to appear.

Characteristics Of Embedded Systems

The main characteristic of embedded systems is that they are task specific. They perform a single task within a larger system. For example, a mobile phone is *not* an embedded system, it is a combination of embedded systems that together allow it to perform a variety of general-purpose tasks. The embedded systems within it perform specialized functions. For example, the GUI performs the singular function of allowing the user to interface with the device. In short, they are programmable computers, but designed for specific purposes, not general ones.

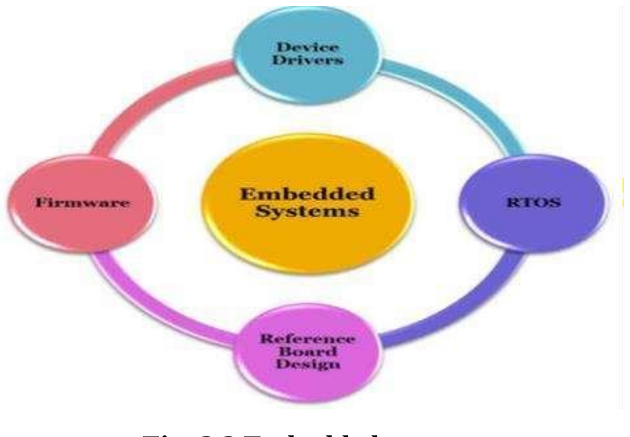


Fig: 2.2 Embedded systems

The hardware of embedded systems is based around microprocessors and microcontrollers. Microprocessors are very similar to microcontrollers and generally refer to a CPU that is integrated with other basic computing components such as memory chips and digital signal processors (DSP). Microcontrollers have those components built into one chip.

Additionally, embedded systems can include the following characteristics:

- comprised of hardware, software and firmware;
- embedded in a larger system to perform a specific function as they are built for specialized tasks within the system, not various tasks;
- either microprocessor-based or microcontroller-based -- both are integrated circuits that give the system compute power;
- often used for sensing and real-time computing in internet of things (IoT) devices -- devices that are internet-connected and do not require a user to operate;
- vary in complexity and in function, which affects the type of software, firmware and hardware they use.
- often required to perform their function under a time constraint to keep the larger system functioning properly.

Fig: 2.3 Blocks of embedded systems

Embedded systems vary in complexity, but generally consist of three main elements:

- **Hardware:** The hardware of embedded systems is based around microprocessors and microcontrollers. Microprocessors are very similar to microcontrollers, and generally refer to a CPU that is integrated with other basic computing components such as memory chips and digital signal processors (DSP). Microcontrollers have those components built into one chip.

- **Software:** Software for embedded systems can vary in complexity. However, industrial grade microcontrollers and embedded IoT systems generally run very simple software that requires little memory.
- **Firmware:** Embedded firmware is usually used in more complex embedded systems to connect the software to the hardware. Firmware is the software that interfaces directly with the hardware.

Why Embedded?

An embedded system is a computer system with a particular defined function within a larger mechanical or electrical system. They control many devices in common use. They consume low power, are of a small size and their cost is low per-unit.

Modern embedded systems are often based on micro-controllers. A micro-controller is a small computer on a single integrated circuit which contains a processor core, memory, and programmable input and output peripherals. As Embedded system is dedicated to perform specific tasks therefore, they can be optimized to reduce the size and cost of the product and increase the reliability and performance.

Fig 2.4: Embedded systems hardware

Embedded Systems has brought about a revolution in Science. It is also a part of a Internet of Things (IoT) – a technology in which objects, animals or people are provided with unique identifiers and the ability to transfer data over a network without requiring humanto-human or human-to-computer interaction. Well this is just one good thing about IoT. We can monitor Pollution Levels, we can control the intensity of street lights as per the season and weather requirements, IoT can also provide the parents with real-time information about their baby's breathing, skin temperature, body position, and activity level on their smartphones and many other applications which can make our life easy. Well this is just one good thing about IoT. We can monitor Pollution Levels, we can control the intensity of street lights as per the season and weather requirements.

Design Approaches

A system designed with the embedding of hardware and software together for a specific function with a larger area is an embedded system design. In embedded system design, a microcontroller plays a vital role. Micro-controller is based on Harvard architecture, it is an important component of an embedded system. External processor, internal memory, and i/o components are interfaced with the microcontroller. It occupies less area and less power consumption. The application of microcontrollers is MP3 and washing machines.

Critical Embedded Systems (CES) are systems in which failures are potentially catastrophic and, therefore, hard constraints are imposed on them. In the last years the amount of software accommodated within CES has considerably changed. For example, in smart cars the amount of software has grown about 100 times compared to previous years. This change means that software design for these systems is also bounded to hard constraints (e.g., high security and performance). Steps in the Embedded System Design Process

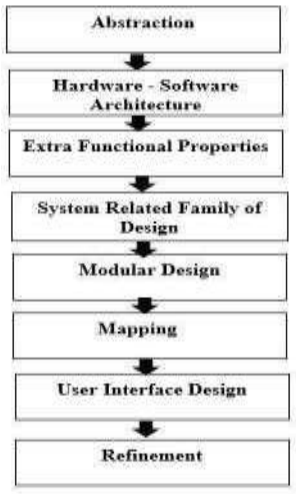

Abstraction
Hardware - Software Architecture
Extra Functional Properties
System Related Family of Design
Modular Design
Mapping
User Interface Design
Refinement

Fig: 2.5 Embedded design-process-steps

The different steps in the embedded system design flow/flow diagram include the following:

Specification: The first step in the process, where you define the requirements that the system must meet.

Hardware and software partitioning: You divide the system into hardware and software components

- **Hardware and software design**: You design approach the hardware and software independently
- **Hardware and software integration**: You integrate the hardware and software, and decide how and when to resolve bugs
- **Software testing**: You test the software to detect vulnerabilities
- **User interface design**: You design the interface between the CPU software and the digital interface logic, and between the digital and analog sides of the interface .

Abstraction

In this stage the problem related to the system is abstracted.

- **Hardware – Software Architecture**

Proper knowledge of hardware and software to be known before starting any design process.

- **Extra Functional Properties**

Extra functions to be implemented are to be understood completely from the main design.

- **System Related Family of Design**

When designing a system, one should refer to a previous system-related family of design.

- **Modular Design**

Separate module designs must be made so that they can be used later on when required.

- **Mapping**

Based on software mapping is done. For example, data flow and program flow are mapped into one.

User Interface Design

In user interface design it depends on user requirements, environment analysis and function of the system. For example, on a mobile phone if we want to reduce the power consumption of mobile phones, we take care of other parameters, so that power consumption can be reduced. For example, on a mobile phone if we want to reduce the power consumption of mobile phones, we take care of other parameters, so that power consumption can be reduced.

Every component and module must be refined appropriately so that the software team can understand.

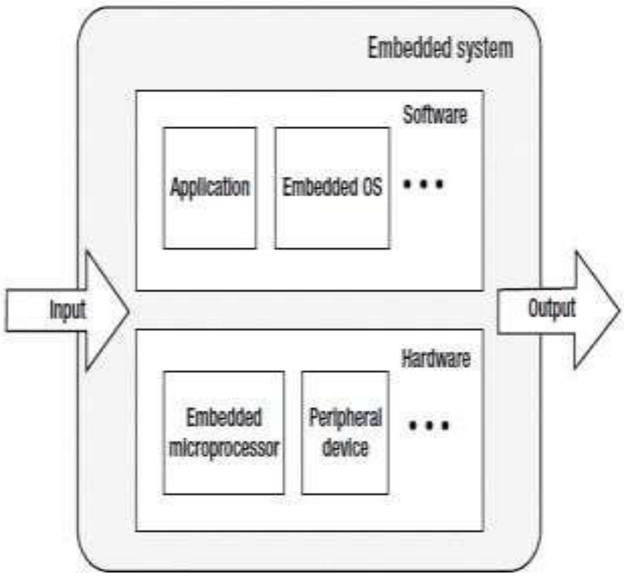


Fig: 2.6 Hardware and software of embedded system

Architectural description language is used to describe the software design.

- Control Hierarchy
- Data structure and hierarchy
- Software Procedure.

In user interface design it depends on user requirements, environment analysis and function of the system. For example, on a mobile phone if we want to reduce the power consumption of mobile phones, we take care of other parameters, so that power consumption can be reduced. WHO has identified formulations for their local preparation.

Embedded systems are used in a variety of technologies across industries. Some examples include:

Automobiles: Modern cars commonly consist of many computers or embedded systems, designed to perform different tasks within the vehicle. Some of these systems perform basic utility function and others provide entertainment or user-facing functions. Some embedded systems in consumer vehicles include cruise control, backup sensors, suspension control, navigation systems and airbag systems.

Mobile phones. These consist of many embedded systems, including GUI software and hardware, operating systems, cameras, microphones and USB I/O modules.

Table 2.1: Design parameters and functions of an embedded system

Design Metrics / Design Parameters of an Embedded System	**Function**
Power Dissipation	Always maintained low
Performance	Should be high
Process Deadlines	The process/task should be completed within a specified time.
Manufacturing Cost	Should be maintained.
Engineering Cost	It is the cost for the edit-test-debug of hardware and software.
Size	Size is defined in terms of memory RAM/ROM/Flash Memory/Physical Memory.
Prototype	It is the total time taken for developing a system and testing it.
Safety	System safety should be taken like phone locking, user safety like engine breaks down safety measure must be taken
Maintenance	Proper maintenance of the system must be taken, in order to avoid system failure.
Time to market	It is the time taken- for the product/system developed to be launched into the market.

Most logic gates are made from silicon, although some utilize gallium arsenide or other semiconductor materials. The semiconductor material is doped for organization into layers. The doped layers drive power capabilities and typical impedances at input or outputs of each gate. Logic gates used together must employ the same, or complementary, material properties. Knowledge of material properties for logic gates will drive selection of parts within design blocks Data bits build into digital words used to communicate with other design blocks within the system. Digital bits and words do this with logic gates in an organized fashion using dedicated address, data, or control signal nodes. Logic gates are the physical devices that enable processing of many 1’s and 0’s.

- **Industrial machines.** They can contain embedded systems, like sensors, and can be embedded systems themselves. Industrial machines often have embedded automation systems that perform specific monitoring and control functions.

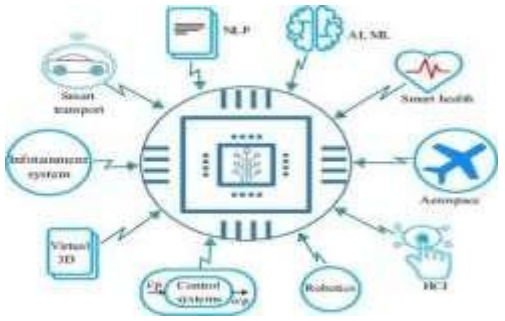


Fig: 2.7 Applications of embedded systems

- **Medical equipment.** These may contain embedded systems like sensors and control mechanisms. Medical equipment, such as industrial machines, also must be very user-friendly, so that human health isn't jeopardized by preventable machine mistakes. This means they'll often include a more complex OS and GUI designed for an appropriate UI.

COMBINATION OF LOGIC DEVICES

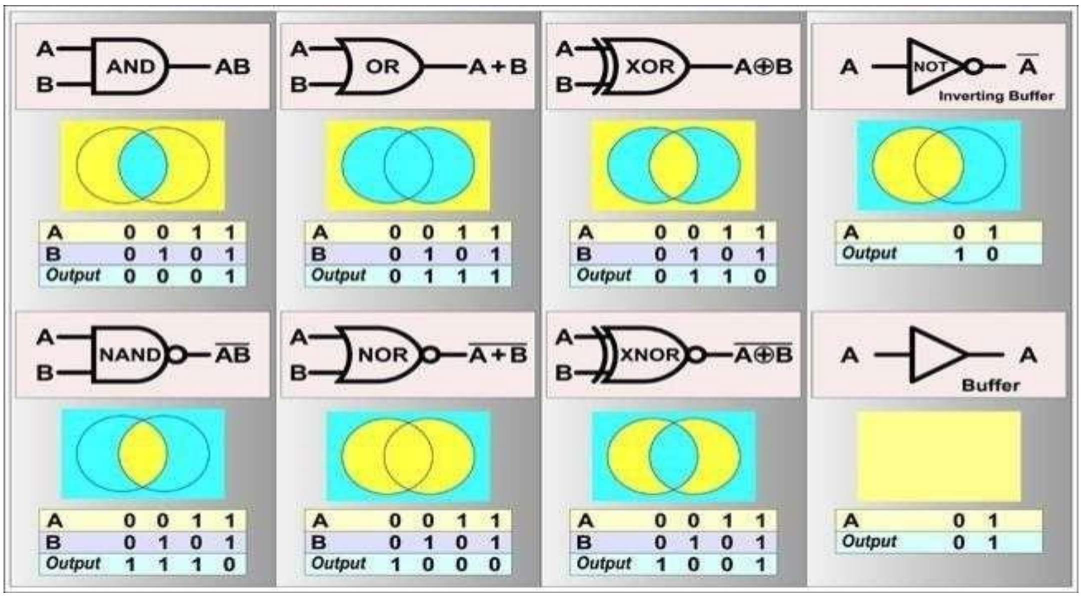


Fig: 2.8 Logic gates

Logic gates are physical devices that use combinational logic to switch an electrical one ("1") or zero ("0") to downstream blocks in digital design. Combinational logic uses those bits to send or receive data within embedded systems. Data bits build into digital words used to communicate with other design blocks within the system. Digital bits and words do this with logic gates in an organized fashion using dedicated address, data, or control signal nodes. Logic gates are the physical devices that enable processing of many 1's and 0's.

Logic families are collections of integrated circuits containing logic gates that perform functions needed by embedded systems to communicate with one another to drive the design. Logic gates are organized into families relative to the type of material and its operational characteristics.

Most logic gates are made from silicon, although some utilize gallium arsenide or other semiconductor materials. The semiconductor material is doped for organization into layers. The doped layers drive power capabilities and typical impedances at input or outputs of each gate. Logic gates used together must employ the same, or complementary, material properties. Knowledge of material properties for logic gates will drive selection of parts within design blocks.

CHAPTER 3
SOFTWARE REQUIREMENTS

3.1 SOFTWARE TOOL

- **Arduino Software**

Arduino IDE (Integrated Development Environment) is required to program the Arduino Uno board. Download it here.

- **Programming Arduino**

Once Arduino IDE is installed on the computer, connect the board with computer using USB cable. Now open the Arduino IDE and choose the correct board by selecting Tools>Boards>Arduino/Genuino Uno, and choose the correct Port by selecting Tools>Port. Arduino Uno is programmed using Arduino programming language based on Wiring. To get itstarted with Arduino Uno board and blink the built-in LED, load the example code by selecting Files>Examples>Basics>Blink. Once the example code (also shown below) is loaded into your IDE, click on the 'upload' button given on the top bar. Once the upload is finished, you should see the Arduino's built-in LED blinking. Below is the example code for blinking:

- **Arduino – Installation**

After learning about the main parts of the Arduino UNO board, we are ready to learn how to set up the Arduino IDE. Once we learn this, we will be ready to upload our program on the Arduino board.In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB

Fig 3.1: Mini-b cable

Step 1: First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image
In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the followingimage, In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, Diecimila you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

Step 2: Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

He project will be programmed using the Arduino IDE, an open-source software environment for writing and uploading code to the Arduino microcontroller. It allows easy interaction with hardware components through simple code and libraries.
Open the Arduino IDE, write the code to control the hardware components. The code will be responsible for gathering data from sensors, processing it, and taking appropriate actions based on that data.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

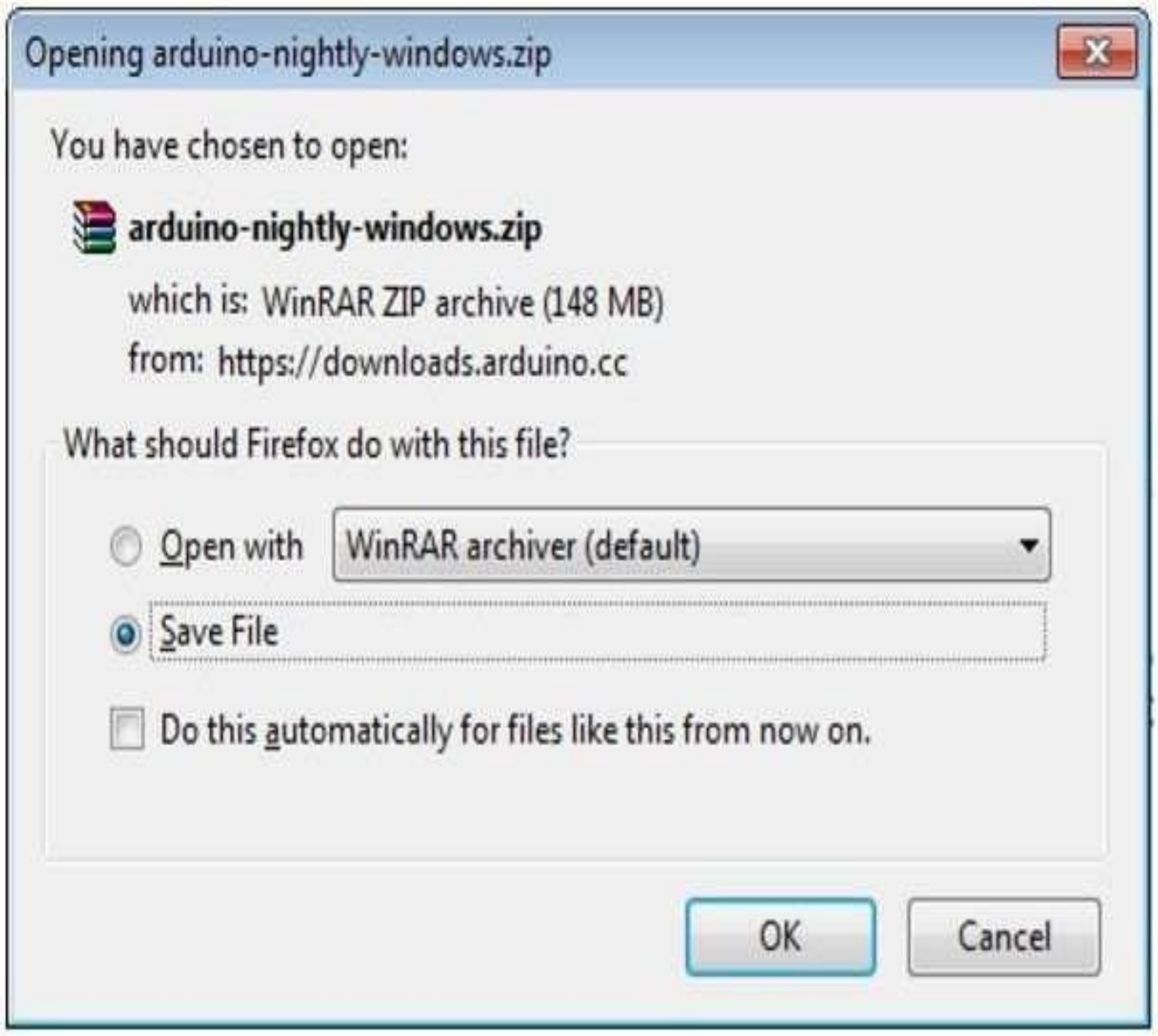
He project will be programmed using the Arduino IDE, an open-source software environment for writing and uploading code to the Arduino microcontroller. It allows easy interaction with hardware components through simple code and libraries.

Open the Arduino IDE, write the code to control the hardware components. The code will be responsible for gathering data from sensors, processing it, and taking appropriate actions based on that data.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your

Step 3: Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection


Opening arduino-nightly-windows.zip

You have chosen to open:

arduino-nightly-windows.zip

which is: WinRAR ZIP archive (148 MB)

from: https://downloads.arduino.cc

What should Firefox do with this file?

Open with WinRAR archiver (default)

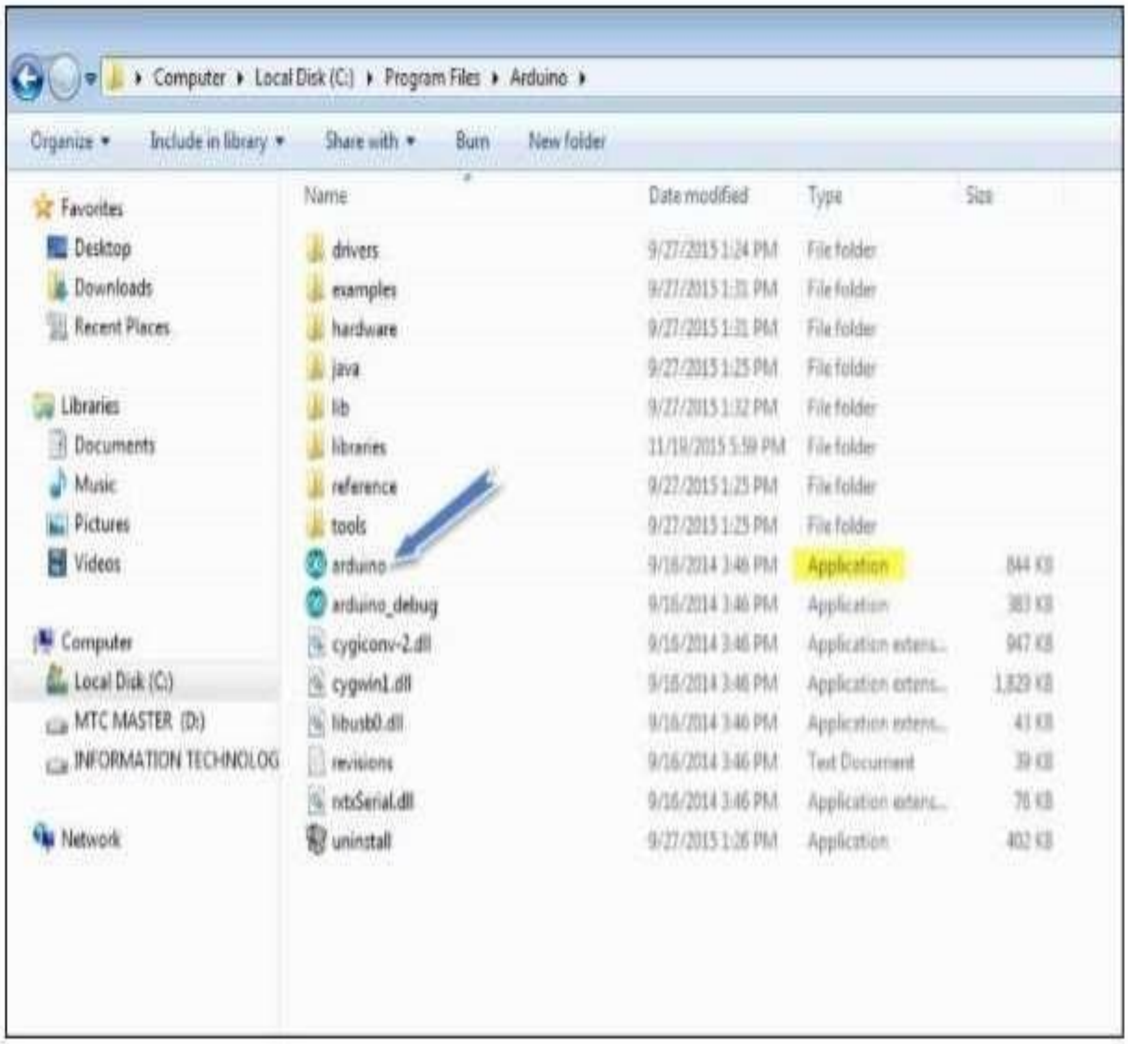
[✓ Save File]

Do this automatically for files like this from now on.

Fig 3.2: Opening Arduino

Step 4: **Launch Arduino IDE**.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Doubleclick the icon to start the IDE



Name	Date modified	Type	Size
drivers	9/27/2015 1:24 PM	File folder	
examples	9/27/2015 1:31 PM	File folder	
hardware	9/27/2015 1:31 PM	File folder	
java	9/27/2015 1:25 PM	File folder	
lib	9/27/2015 1:32 PM	File folder	
libraries	11/19/2015 5:59 PM	File folder	
reference	9/27/2015 1:25 PM	File folder	
tools	9/27/2015 1:25 PM	File folder	
arduino	9/16/2014 3:46 PM	Application	844 KB
arduino_debug	9/16/2014 3:46 PM	Application	383 KB
cygiconv-2.dll	9/16/2014 3:46 PM	Application extens...	947 KB
cygwin1.dll	9/16/2014 3:46 PM	Application extens...	1,829 KB
libusb0.dll	9/16/2014 3:46 PM	Application extens...	43 KB
revisions	9/16/2014 3:46 PM	Text Document	39 KB
rxtxSerial.dll	9/16/2014 3:46 PM	Application extens...	76 KB
uninstall	9/27/2015 1:26 PM	Application	402 KB

Fig 3.3: Launch Arduino IDE

Step 5: Open your first project.

Once the software starts, you have two options:

- Create a new project.

➢ Open an existing project example.

To open an existing project example, select File -> Example -> Basics -> Blink.

Here, we are selecting just one of the examples with the name Blink. It turns the LED on and off with some time delay. You can select any other example from the list.

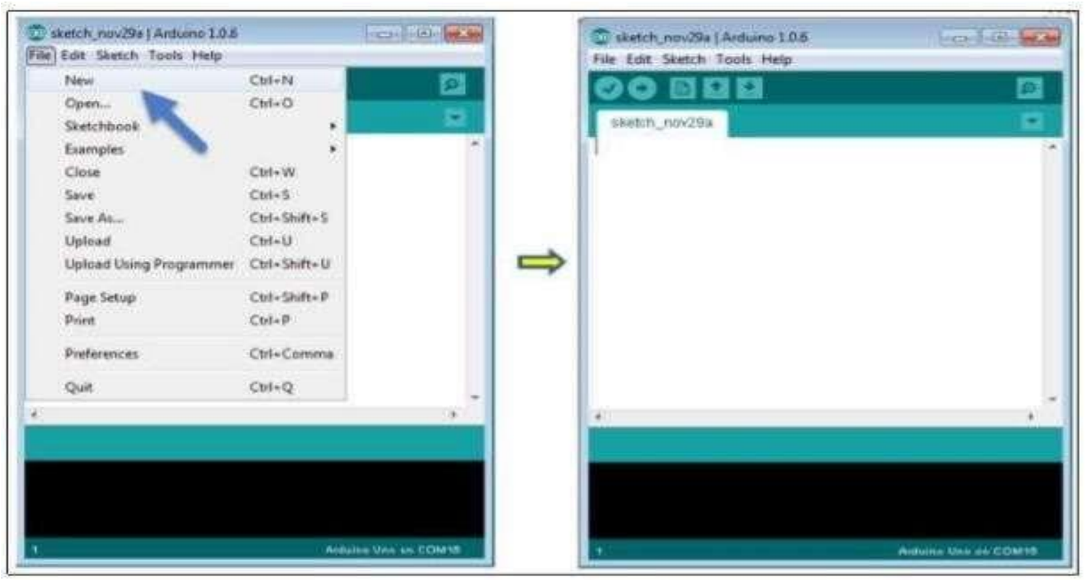


Fig 3.4 : Launch Arduino IDE

Step 6: Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

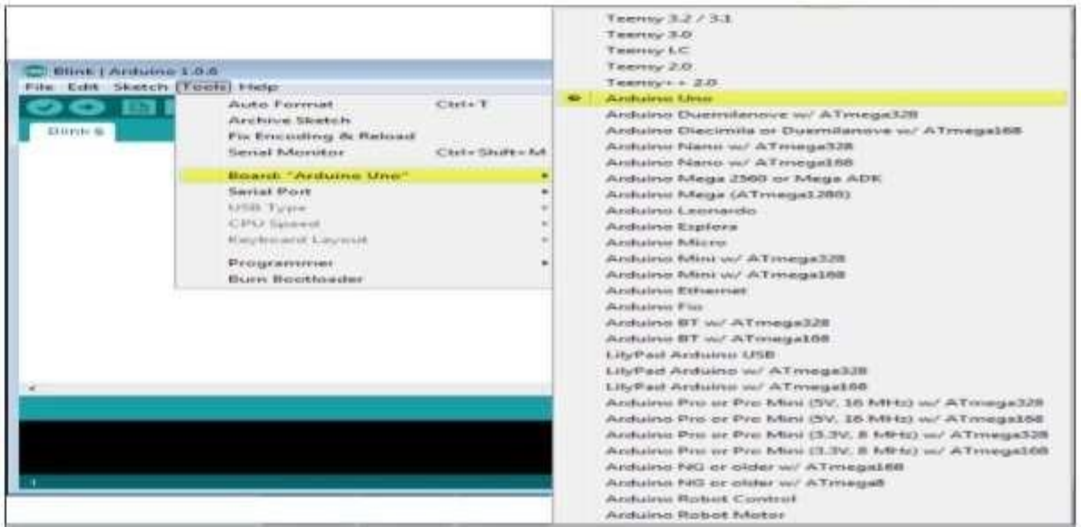

Blink | Arduino 1.0.6
File Edit Sketch Tools Help
Blink §
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Board: "Arduino Uno"
Serial Port
USB Type
CPU Speed
Keyboard Layout
Programmer
Burn Bootloader
Teensy 3.2 / 3.1
Teensy 3.0
Teensy LC
Teensy 2.0
Teensy++ 2.0
Arduino Uno
Arduino Duemilanove w/ ATmega328
Arduino Diecimila or Duemilanove w/ ATmega168
Arduino Nano w/ ATmega328
Arduino Nano w/ ATmega168
Arduino Mega 2560 or Mega ADK
Arduino Mega (ATmega1280)
Arduino Leonardo
Arduino Esplora
Arduino Micro
Arduino Mini w/ ATmega328
Arduino Mini w/ ATmega168
Arduino Ethernet
Arduino Fio
Arduino BT w/ ATmega328
Arduino BT w/ ATmega168
LilyPad Arduino USB
LilyPad Arduino w/ ATmega328
LilyPad Arduino w/ ATmega168
Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328
Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega168
Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328
Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega168
Arduino NG or older w/ ATmega168
Arduino NG or older w/ ATmega8
Arduino Robot Control
Arduino Robot Motor

Fig 3.5 : Launch Arduino IDE TOOLS BAR

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7: Select your serial port.

Select the serial device of the Arduino board. Go to Tools -> Serial Port menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

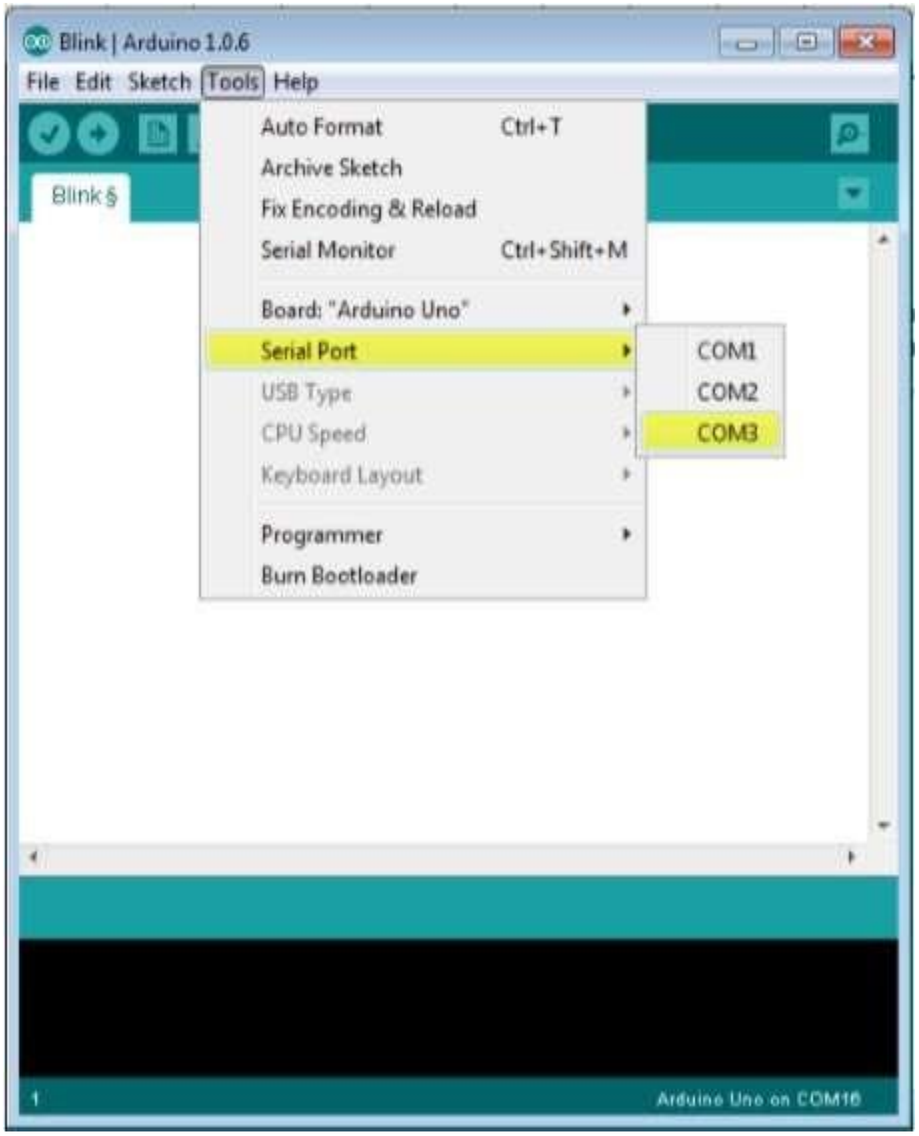


Fig 3.6 : Launch Arduino IDE

Step 8: Upload the program to your board

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar

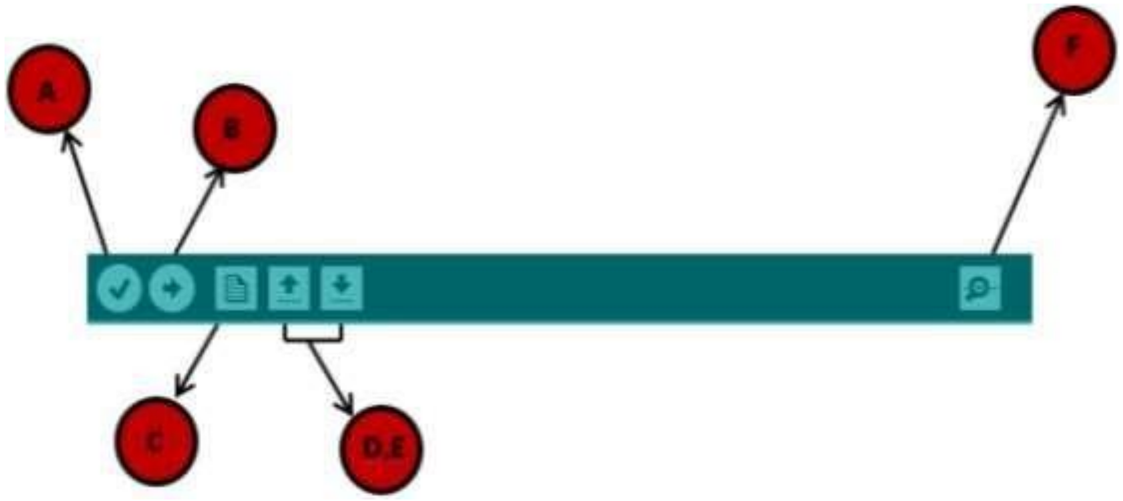


Fig 3.7: Program Uploading

A- Used to check if there is any compilation error.

B- Used to upload a program to the Arduino board.

C- Shortcut used to create a new sketch.

D- Used to directly open one of the example sketch.

E- Used to save your sketch.

F- Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note: If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message

"Done uploading" will appear in the status bar.

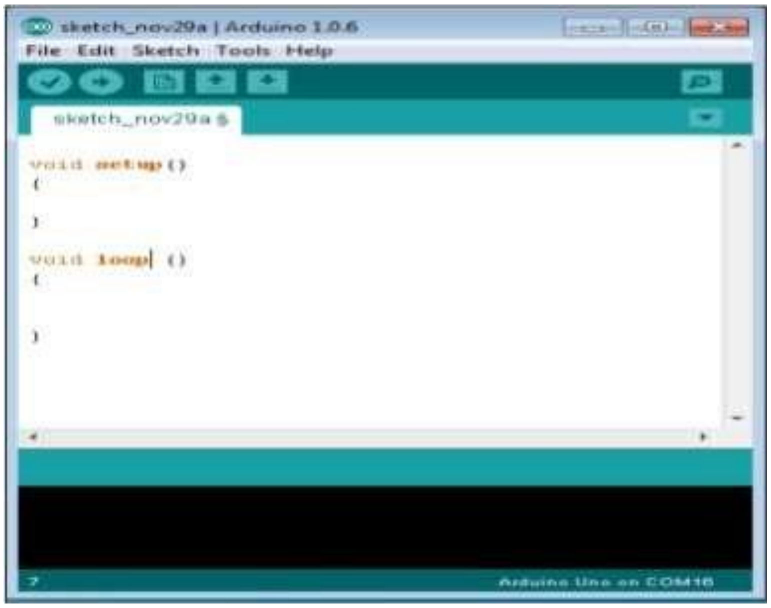
we will study in depth, the Arduino program structure and we will learn more new terminologies used in the Arduino world. The Arduino software is open-source. The source code for the Java environment is released under the GPL and the C/C++ microcontroller libraries are under the LGPL.

Sketch: The first new terminology is the Arduino program called "sketch".

Structure Arduino programs can be divided in three main parts: Structure, Values (variables and constants), and Functions. In this tutorial, we will learn about the Arduino software program, step by step, and how we can write the program without any syntax or compilation error.

Let us start with the Structure. Software structure consist of two main functions:

- Setup( ) function
- Loop( ) function



```
void setup ()
{
}
void loop ()
{

}
```

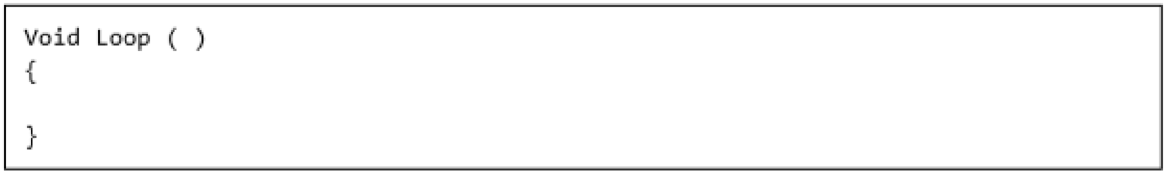
Fig 3.8: SKETCH_NOV29a

```
Void setup ( )
{

}
```

PURPOSE: The setup() function is called when a sketch starts. Use it to initialize the variables, pin modes, start using libraries, etc. The setup function will only run once, after each power up or reset of the Arduino board.

INPUT: - OUTPUT: - RETURN:



```
Void Loop ( )
{

}
```

CHAPTER 4
HARDWARE REQURIEMENTS

4.1 POWER SUPPLY :

he power supply is a crucial component of our smart automatic aquarium monitoring system, providing the necessary power to the system's sensors, pumps, and other components. It is designed to deliver a stable and efficient output, ensuring reliable operation of the system.

In terms of specifications, the power supply has an input range of 100-240V AC, allowing it to be used in various countries and regions. The output is 5V DC, 2A, which is suitable for powering the system's low-voltage components. The power supply also has a high efficiency rating of over 80%, minimizing heat generation and reducing energy consumption.

To ensure safe and reliable operation, the power supply has several built-in protection features. These include overvoltage protection (OVP), undervoltage protection (UVP), overcurrent protection (OCP), and short-circuit protection (SCP). These features help prevent damage to the power supply and the system's components in case of abnormal operating conditions.

The power supply is also designed to meet various international safety and environmental standards. It is certified to UL, CE, and RoHS standards, ensuring compliance with regulations in North America, Europe, and other regions.

In terms of physical design, the power supply has a compact and lightweight enclosure, making it easy to install inside the aquarium cabinet or mount externally. The enclosure is also designed to provide good heat dissipation, ensuring reliable operation even in high-temperature environments**.**

To ensure safe and reliable operation, it's essential to follow proper safety precautions when using the power supply. Avoid touching electrical components or wiring to prevent electric shock. Keep the power supply away from flammable materials to prevent fire hazards. Ensure good ventilation and avoid blocking airflow to prevent overheating.

4.2. AURDINO UNO

Placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.

- Stronger RESET circuit.
- At mega 16U2 replace the 8U2. "Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards. . It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board.

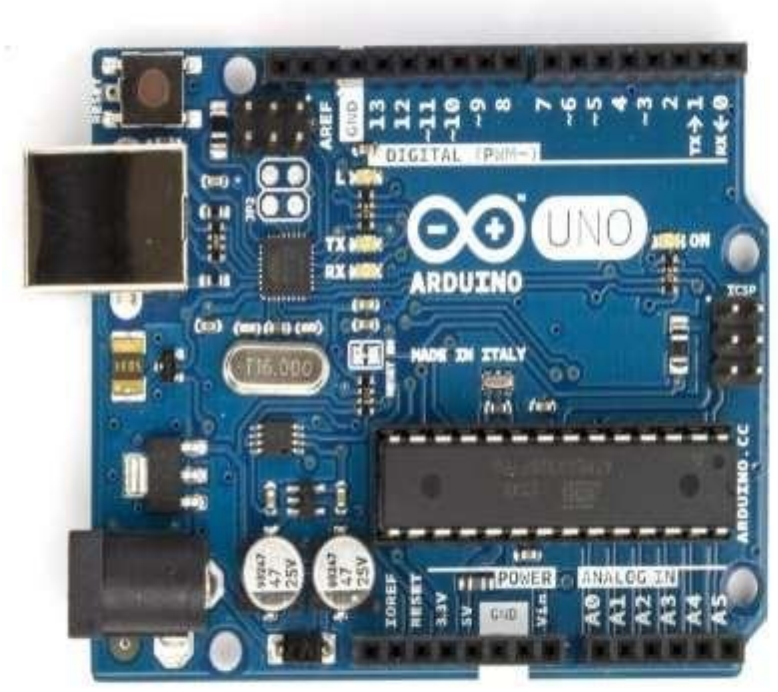


Fig 4.1 : Arduino Uno

4.2.1 Summary:

- Microcontroller ATmega328
- Operating Voltage 5V
- Input Voltage (recommended) 7-12V
- Input Voltage (limits) 6-20V
- Digital I/O Pins 14 (of which 6 provide PWM output)
- Analog Input Pins 6
- DC Current per I/O Pin 40 mA
- DC Current for 3.3V Pin 50 mA
- Flash Memory 32 KB (ATmega328) of which 0.5 KB used by bootloader
- SRAM 2 KB (ATmega328)
- EEPROM 1 KB (ATmega328)
- Clock Speed 16 MHz

Overview The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter. Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode. Revision 3 of the board has the following new features:

- 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.

4.2.2 Schematic & Reference Design:

EAGLE files: arduino-uno-Rev3-reference-design.zip (NOTE: works with Eagle 6.0 and newer) Schematic: arduino-uno-Rev3-schematic.pdf Note: The Arduino reference design can use an Atmega8, 168, or 328, Current models use an ATmega328, but an Atmega8 is shown in the schematic for reference. The pin configuration is identical on all three processors

4.2.3 Power:

The Arduino Uno can be powered via the USB connection or with an external power supply. The PowerSource is selected automatically.External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. The power pins are as follows:

4.2.4 VIN:

The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.This pin outputs a regulated 5V from the regulator on the board.
The bord power with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it. 3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

4.2.5 Memory:

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

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

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attach Interrupt() function for details.

4.2.7 Schematic & Reference Design

EAGLE files: arduino-uno-Rev3-reference-design.zip (NOTE: works with Eagle 6.0 and newer) Schematic: arduino-uno-Rev3-schematic.pdf Note: The Arduino reference design can use an Atmega8, 168, or 328, Current models use an ATmega328, but an Atmega8 is shown in the schematic for reference. The pin configuration is identical on all three processors

4.2.8 Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The PowerSource is selected automatically.External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack.
The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. The power pins are as follows:

4.2.9 Communication:

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required.

4.2.10.POWER SUPPLY

The input to the circuit is applied from the regulated power supply. The a.c. input i.e., 230V from the mains supply is step down by the transformer to 12V and is fed to a rectifier. The output obtained from the rectifier is a pulsating d.c voltage. So in order to get a pure d.c voltage, the output voltage from the rectifier is fed to a filter to remove any a.c components present even after rectification. Now, this voltage is given to a voltage regulator to obtain a pure constant dc voltage.The output from the transformer is fed to the rectifier. It converts A.C. into pulsating D.C. The rectifier may be a half wave or a full wave rectifier.

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

- TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library. There are a couple of other pins on the board:
- AREF.Reference voltage for the analog inputs. Used with analogReference().
- Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board. See also the mapping between Arduino pins and ATmega328 ports. The mapping for the Atmega8, 168, and 328 is identical.

However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board.

The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

Typically used to add a reset button to shields which block the one on the board. See also the mapping between Arduino pins and ATmega328 ports. The mapping for the Atmega8, 168, and 328 is identical. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0

he RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer

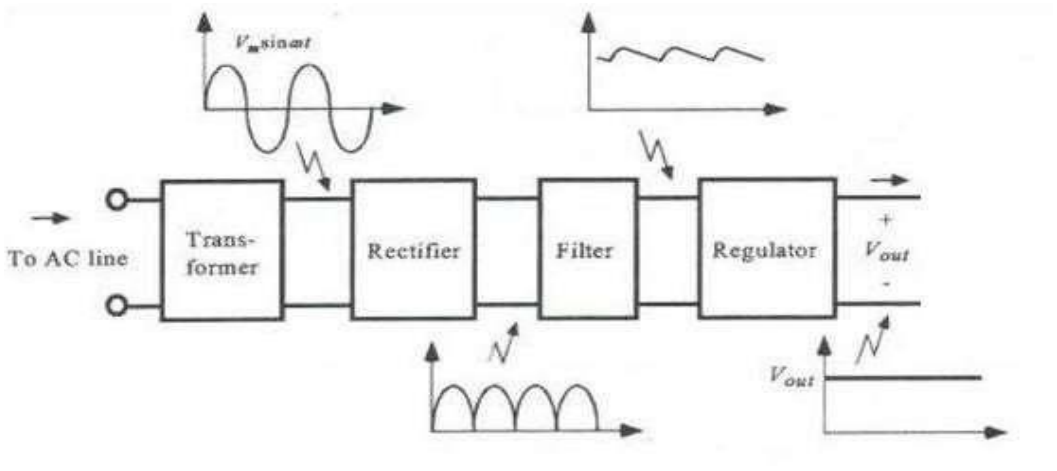


Fig 4.2 Block Diagram of Power supply

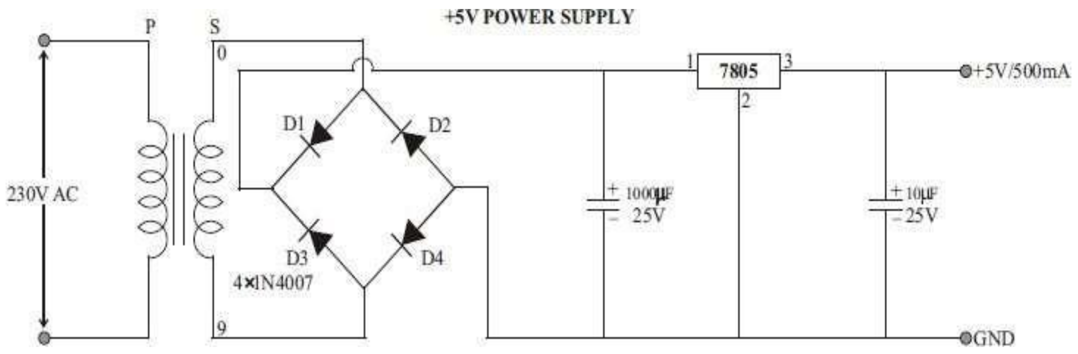


Fig 4.3 Circuit Diagram of Power supply

The input to the circuit is applied from the regulated power supply. The a.c. input i.e., 230V from the mains supply is step down by the transformer to 12V and is fed to a rectifier. The output obtained from the rectifier is a pulsating d.c voltage. So in order to get a pure d.c voltage, the output voltage from the rectifier is fed to a filter to remove any a.c components present even after rectification. Now, this voltage is given to a voltage regulator to obtain a pure constant dc voltage.The output from the transformer is fed to the rectifier. It converts A.C. into pulsating D.C. The rectifier may be a half wave or a full wave rectifier.

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

4.2.12 Pin Description:

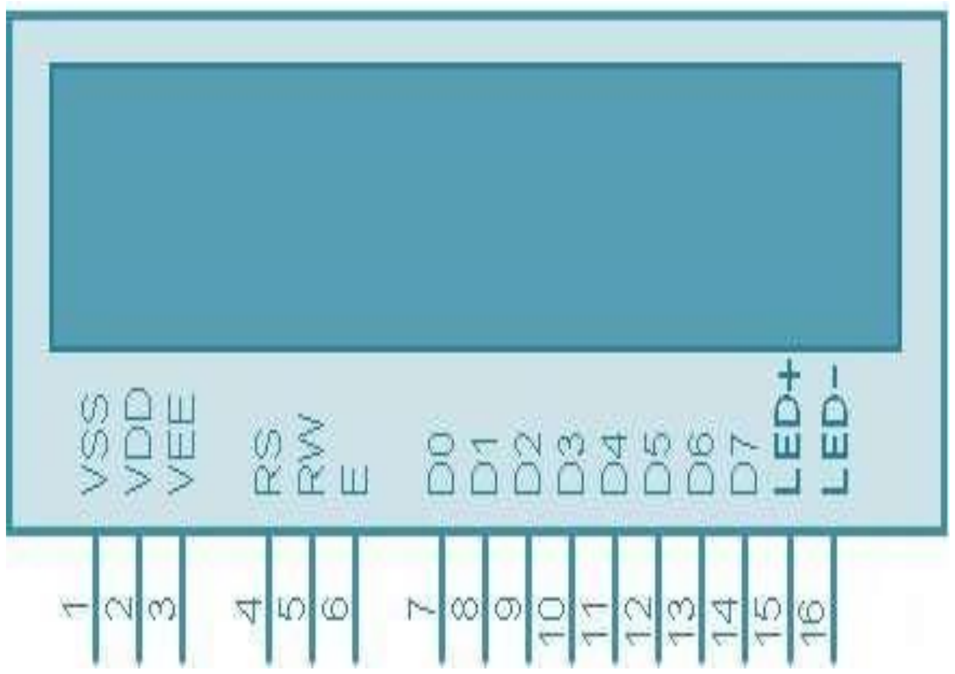


Fig 4.4 Pin Diagram of LCD

Table 4.1 Pin Description of LCD

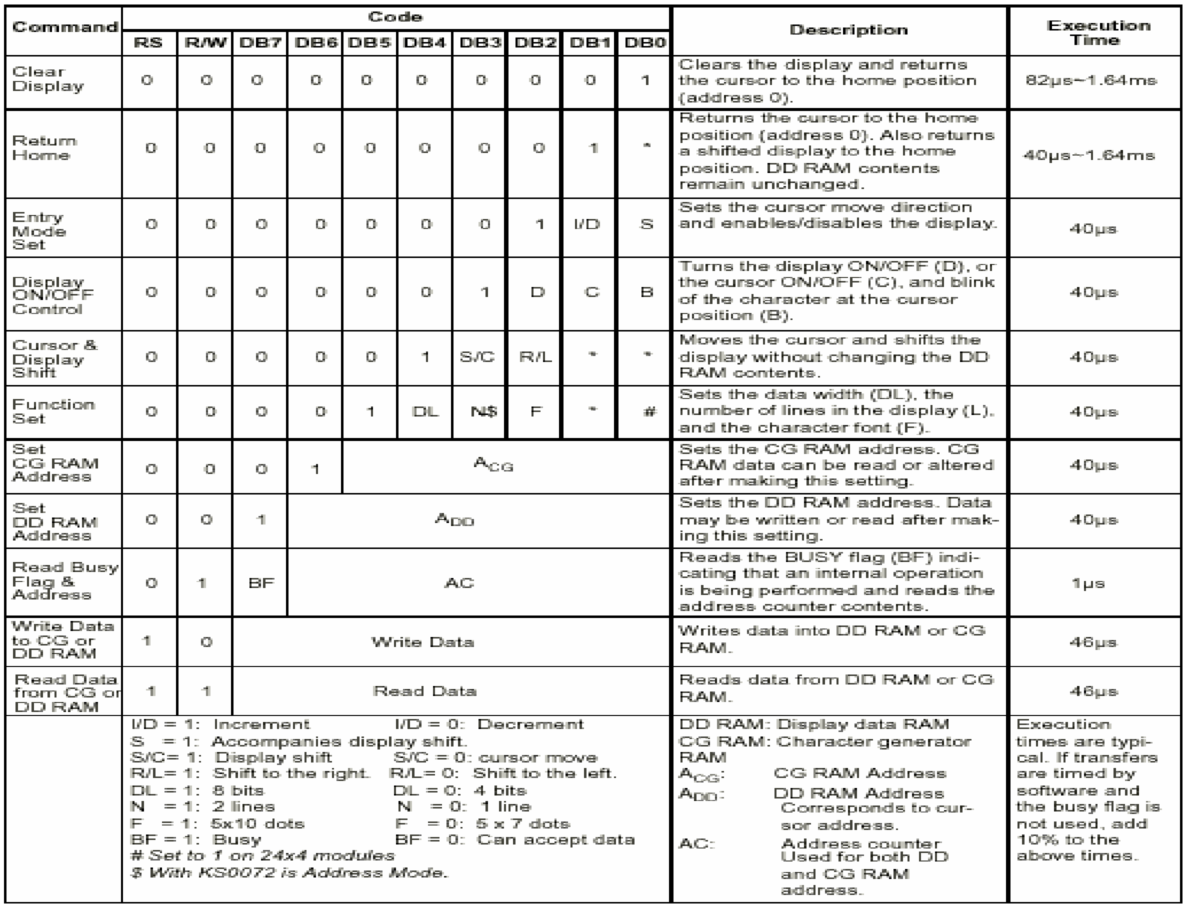
PIN NO.	NAME	DESCRIPTION
1	**VSS**	Power supply (GND)
2	**VCC**	Power supply (+5V)
3	**VEE**	Contrast adjust
4	**RS**	0 = Instruction input 1 = Data input
5	**R/W**	0 = Write to LCD module 1 = Read from LCD module
6	**EN**	Enable signal
7	**D0**	Data bus line 0 (LSB)
8	**D1**	Data bus line 1
9	**D2**	Data bus line 2

10	**D3**	Data bus line 3
11	**D4**	Data bus line 4
12	**D5**	Data bus line 5
13	**D6**	Data bus line 6
14	**D7**	Data bus line 7 (MSB)
15	**LED+**	Back Light VCC
16	**LED-**	Back Light GND

Although looking at the table you can make your own commands and test them. Below is a brief list of useful commands which are used frequently while working on the LCD.The LCD Smart Automatic Aquarium Monitoring System operates by continuously collecting data from various sensors such as the temperature sensor (like the DHT11 or DS18B20), pH sensor, and TDS (Total Dissolved Solids) sensor.

These sensors monitor essential parameters like water temperature, pH levels, and water quality. The microcontroller, often an Arduino or Raspberry Pi, processes this data and compares it to predefined safe thresholds.LED is on, when the pin is LOW, it's off. The Uno has 6 analog inputs, labeled A0 through A5, each which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts.

Table 4.2 Command operation of LCD



Command	Code										Description	Execution Time
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Clears the display and returns the cursor to the home position (address 0).	82μs~1.64ms
Return Home	0	0	0	0	0	0	0	0	1	*	Returns the cursor to the home position (address 0). Also returns a shifted display to the home position. DD RAM contents remain unchanged.	40μs~1.64ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Sets the cursor move direction and enables/disables the display.	40μs
Display ON/OFF Control	0	0	0	0	0	0	1	D	C	B	Turns the display ON/OFF (D), or the cursor ON/OFF (C), and blink of the character at the cursor position (B).	40μs
Cursor & Display Shift	0	0	0	0	0	1	S/C	R/L	*	*	Moves the cursor and shifts the display without changing the DD RAM contents.	40μs
Function Set	0	0	0	0	1	DL	N$	F	*	#	Sets the data width (DL), the number of lines in the display (L), and the character font (F).	40μs
Set CG RAM Address	0	0	0	1	A_{CG}						Sets the CG RAM address. CG RAM data can be read or altered after making this setting.	40μs
Set DD RAM Address	0	0	1	A_{DD}							Sets the DD RAM address. Data may be written or read after making this setting.	40μs
Read Busy Flag & Address	0	1	BF	AC							Reads the BUSY flag (BF) indicating that an internal operation is being performed and reads the address counter contents.	1μs
Write Data to CG or DD RAM	1	0	Write Data								Writes data into DD RAM or CG RAM.	46μs
Read Data from CG or DD RAM	1	1	Read Data								Reads data from DD RAM or CG RAM.	46μs
	I/D = 1: Increment, I/D = 0: Decrement S = 1: Accompanies display shift. S/C= 1: Display shift, S/C = 0: cursor move R/L= 1: Shift to the right. R/L= 0: Shift to the left. DL = 1: 8 bits, DL = 0: 4 bits N = 1: 2 lines, N = 0: 1 line F = 1: 5x10 dots, F = 0: 5 x 7 dots BF = 1: Busy, BF = 0: Can accept data *# Set to 1 on 24x4 modules* *$ With KS0072 is Address Mode.*										DD RAM: Display data RAM CG RAM: Character generator RAM A_{CG}: CG RAM Address A_{DD}: DD RAM Address Corresponds to cursor address. AC: Address counter Used for both DD and CG RAM address.	Execution times are typical. If transfers are timed by software and the busy flag is not used, add 10% to the above times.

Although looking at the table you can make your own commands and test them. Below is a brief list of useful commands which are used frequently while working on the LCD.
The LCD Smart Automatic Aquarium Monitoring System operates by continuously collecting data from various sensors such as the temperature sensor (like the DHT11 or DS18B20), pH sensor, and TDS (Total Dissolved Solids) sensor.

These sensors monitor essential parameters like water temperature, pH levels, and water quality. The microcontroller, often an Arduino or Raspberry Pi, processes this data and compares it to predefined safe thresholds. If any parameter falls outside the acceptable range, the system can take corrective actions by controlling actuators such as water pumps, heaters, or aerators via a relay module.If any parameter falls outside the acceptable range, the system can take corrective actions by controlling actuators such as water pumps, heaters, or aerators via a relay module.

Table 4.3 Command List of LCD

No.	Instruction	Hex	Decimal
1	Function Set: 8-bit, 1 Line, 5x7 Dots	0x30	48
2	Function Set: 8-bit, 2 Line, 5x7 Dots	0x38	56
3	Function Set: 4-bit, 1 Line, 5x7 Dots	0x20	32
4	Function Set: 4-bit, 2 Line, 5x7 Dots	0x28	40
5	Entry Mode	0x06	6
6	Display off Cursor off (clearing display without clearing DDRAM content)	0x08	8
7	Display on Cursor on	0x0E	14
8	Display on Cursor off	0x0C	12
9	Display on Cursor blinking	0x0F	15
10	Shift entire display left	0x18	24
12	Shift entire display right	0x1C	30
13	Move cursor left by one character	0x10	16
14	Move cursor right by one character	0x14	20
15	Clear Display (also clear DDRAM content)	0x01	1
16	Set DDRAM address or cursor position on display	0x80+add	128+add
17	Set CGRAM address or set pointer to CGRAM location	0x40+add	64+add

Sending Commands to LCD

To send commands we simply need to select the command register. Everything is same as we have done in the initialization routine. But we will summarize the common steps and put them in a single subroutine. Following are the steps:

- move data to LCD port
- select command register
- select write operation
- send enable signal
- wait for LCD to process the command
- Sending Data to LCD
- To send data move data to LCD port
- select data register
- select write operation Bottom of Form.

4.3 DISPLAY

often called a **notebook**, is a small, portable personal computer (PC) with a "clamshell" form factor, typically having a thin LCD or LED computer screen mounted on the inside of the upper lid of the clamshell and an alphanumeric keyboard on the inside of the lower lid. The clamshell is opened up to use the computer. Laptops are folded shut for transportation, and thus are suitable for mobile use.[1] Its name comes from lap, as it was deemed to be placed on a person's lap when being used.

Although originally there was a distinction between laptops and notebooks (the former being bigger and heavier than the latter), as of 2014, there is often no longer any difference.[2] Today, laptops are commonly used in a variety of settings, such as at work, in education, for playing games, Internet surfing, for personal multimedia, and general home computer use.

Laptops combine all the input/output components and capabilities of a desktop computer, including the display screen, small speakers, a keyboard, data storage device, optical disc drive, pointing devices (such as a touchpad or trackpad), a processor, and memory into a single unit. Laptops are folded shut for transportation, and thus are suitable for mobile use.[1] Its name comes from lap, as it was deemed to be placed on a person's lap

Design elements, form factor and construction can also vary significantly between models depending on intended use. Examples of specialized models of laptops include rugged notebooks for use in construction or military applications, as well as low production cost laptops such as those from the One Laptop per Child (OLPC) organization, which incorporate features like solar charging and semi-flexible components not found on most laptop computers. Portable computers, which later developed into modern laptops, were originally considered to be a small niche market, mostly for specialized field applications, such as in the military, for accountants, or for traveling sales representatives. As the portable computers evolved into the modern laptop, they became widely used for a variety of purposes.

❖ BUZZER

General description

A buzzer or beeper is an audio signaling device, which may be mechanical, electromechanical, or piezoelectric. Typical uses of buzzers and beepers include alarm devices, timers and confirmation of user input such as a mouse click or key stroke. Buzzer is an integrated structure of electronic transducers, DC power supply, widely used in computers, printers, copiers, alarms, electronic toys, automotive electronic equipment, telephones, timers and other electronic products for sound devices. Active buzzer 5V Rated power can be directly connected to a continuous sound, this section dedicated sensor expansion module and the board in combination, can complete a simple circuit design, to "plug and play."

❖ PRODUCT DESCRIPTION

A buzzer or beeper is an audio signalling device, which may be mechanical, electromechanical, or piezoelectric. Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke. It generates consistent single tone sound just by applying D.C voltage. Using a suitably designed resonant system, this type can be used where large sound volumes are needed.

Typical uses of buzzers and beepers include alarm devices, timers and confirmation of user input such as a mouse click or key stroke. Buzzer is an integrated structure of electronic transducers, DC power supply, widely used in computers, printers, copiers, alarms, electronic toys, automotive electronic equipment, telephones, timers and

At Future Electronics we stock many of the most common types categorized by Type, Sound Level, Frequency, Rated Voltage, Dimension and Packaging Type. They transmit at frequencies of 2.4 GHz or 5 GHz. This frequency is considerably higher than the frequencies used for cell phones, walkie-talkies and televisions. The higher frequency allows the signal to carry more data They use 802.11 networking standards

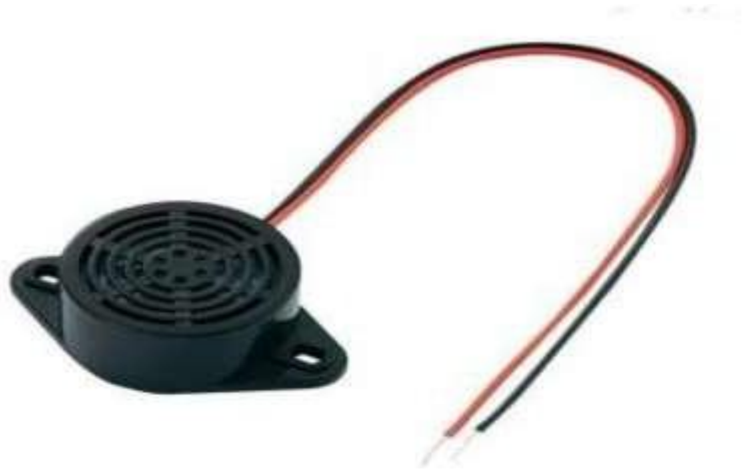


Fig 4.5: Buzzer

FEATURES

Input supply: 5 VDC

- Current consumption: 9.0 mA max.
- Oscillating frequency: 3.0 ±0.5 KHz
- Sound Pressure Level: 85dB min.

4.4 WIFI

A wireless network uses radio waves, just like cell phones, televisions and radios do. In fact, communication across a wireless network is a lot like two-way radio communication. The Wi-Fi connectivity in this project can allow users to interact with the system in several useful ways. For example, you could connect the Smart Aquarium to a cloud-based platform like Thing m Speak, Blynk, or Adafruit IO, where the sensor data is uploaded and visualized in real-time. This allows the user to monitor the aquarium conditions from their smartphone, tablet, or computer without having to be physically present by the tank. Additionally, the system could send notifications to the user if the temperature or humidity goes out of range, if the water quality drops, or if the aquarium's lighting needs adjustment.

❖ How does it work?

A wireless router receives the signal and decodes it. The router sends the information to the Internet using a physical, wired Ethernet connection. The process also works in reverse, with the router receiving information from the Internet, translating it into a radio signal and sending it to the computer's wireless adapter .The radios used for Wi-Fi communication are very similar to the radios used for walkie-talkies, cell phones and other devices. They can transmit and receive radio waves, and they can convert 1s and 0s into radio waves and convert the radio waves back into 1s and 0s. But Wi-Fi radios have a few notable differences from other radios: They transmit at frequencies of 2.4 GHz or 5 GHz. This frequency is considerably higher than the frequencies used for cell phones, walkie-talkies and televisions. The higher frequency allows the signal to carry more data.They use 802.11 networking standards

A small device known as a wireless transmitter, or hub, is required; this device receives information from the internet via your home broadband connection. This transmitter (often referred to as a Wireless Access Point, or WAP) then converts this information into radio waves and emits it, effectively creating a small, local area around itself, within which your devices can receive these radio signals if they are fitted with the correct kind of wireless adapter. This area is often termed a Wireless Local Area Network, or WLAN for short. The radio signals aren't very strong, which is why the Wi-Fi signal doesn't travel very far; it will travel far enough to cover throughout the average home and to the street directly outside, for example, but not much further. One wireless hub is usually enough to enable you to connect to the internet in any room in your home, though the signal will be stronger the nearer the hub you are.

By adding Wi-Fi connectivity to the Smart Aquarium project, users can transform a simple automated system into a highly interactive and flexible IoT-based solution. Whether it's adjusting the water temperature, controlling lighting, or receiving alerts when conditions are not ideal, Wi-Fi integration enhances both convenience and efficiency, providing aquarium enthusiasts with a seamless experience.

This allows the user to monitor the aquarium conditions from their smartphone, tablet, or computer without having to be physically present by the tank. Additionally, the system could send notifications to the user if the temperature or humidity goes out of range, if the water quality drops, or if the aquarium's lighting needs adjustment.

4.5 The Wi-Fi Alliance

The Wi-Fi Alliance, the organization that owns the Wi-Fi registered trademark term specifically defines Wi-Fi as any "*wireless local area network (WLAN) products that are based on the Institute of Electrical and Electronics Engineers' (IEEE) 802.11 standards.*" Initially, When you send information back to the internet – by clicking on a link or sending an email, for example – the process works in reverse; your device sends information via a radio signal to the wireless transmitter, which converts the signal and communicates it back via the broadband connection.

Wi-Fi was used in place of only the 2.4GHz 802.11b standard; however the Wi-Fi Alliance has expanded the generic use of the Wi-Fi term to include any type of network or WLAN product based on any of the 802.11 standards, including 802.11b, 802.11a, dual-band and so on, in an attempt to stop confusion about wireless LAN interoperability.

❖ Wi-Fi Support in Applications and Devices

Wi-Fi is supported by many applications and devices including video game consoles, home networks, PDAs, mobile phones, major operating systems, and other types of consumer electronics. Any products that are tested and approved as "Wi-Fi Certified" (a registered trademark) by the Wi-Fi Alliance are certified as interoperable with each other, even if they are from different manufacturers. Forexample, a user with a Wi-Fi Certified product can use any brand of access point with any other brand of client hardware that also is also "Wi-Fi Certified".

4.6 DHT 11

Humidity is the measure of water vapour present in the air. The level of humidity in air affects various physical, chemical and biological processes. In industrial applications, humidity can affect the business cost of the products, health and safety of the employees. So, in semiconductor industries and control system industries measurement of humidity is very important. Humidity measurement determines the amount of moisture present in the gas that can be a mixture of water vapour, nitrogen, argon or pure gas etc… Humidity sensors are of two types based on their measurement units. They are a relative humidity sensor and Absolute humidity sensor. DHT11 is a digital temperature and humidity sensor.

DHT11 is a low-cost digital sensor for sensing temperature and humidity. This sensor can be easily interfaced with any micro-controller such as Arduino, Raspberry Pi etc… to measure humidity and temperature instantaneously. DHT11 humidity and temperature sensor is available as a sensor and as a module. The difference between this sensor and module is the pull-up resistor and a power-on LED. DHT11 is a relative humidity sensor. To measure the surrounding air this sensor uses a thermistor and a capacitive humidity sensor.

Working Principle of DHT11 Sensor DHT11 sensor consists of a capacitive humidity sensing element and a thermistor for sensing temperature. The humidity sensing capacitor has two electrodes with a moisture holding substrate as a dielectric between them. Change in the capacitance value occurs with the change in humidity levels. The IC measure, process this changed resistance values and change them into digital form.

For measuring temperature this sensor uses a Negative Temperature coefficient thermistor, which causes a decrease in its resistance value with increase in temperature. To get larger resistance value even for the smallest change in temperature, this sensor is usually made up of semiconductor ceramics or polymers. The temperature range of DHT11 is from 0 to 50 degree Celsius with a 2-degree accuracy. Humidity range of this sensor is from 20 to 80% with 5% accuracy. The sampling rate of this sensor is 1Hz .i.e. it gives one reading for every second.

The main advantage of using the DHT11 in this Smart Aquarium project is its low cost, ease of use, and reliability in basic environmental monitoring. Although its accuracy is limited compared to more expensive sensors, it is perfectly suited for general temperature and humidity sensing in an aquarium environment. However, if higher precision is required, especially for larger, more sensitive aquariums, sensors like the DHT22 may be considered.

The library simplifies reading the temperature and humidity data, which is then processed by the Arduino to make decisions about controlling the environmental factors. The integration of additional sensors like water temperature and pH requires careful calibration and the proper handling of sensor data to ensure accuracy, However, if higher precision is required, especially for larger, more sensitive aquariums.

Applications:

This sensor is used in various applications such as measuring humidity and temperature values in heating, ventilation and air conditioning systems. Weather stations also use these sensors to predict weather conditions. The humidity sensor is used as a preventive measure in homes where people are affected by humidity. Offices, cars, museums, greenhouses and industries use this sensor for measuring humidity values and as a safety measure. It’s compact size and sampling rate made this sensor popular among hobbyists. Some of the sensors which can be used as an alternative to DHT11 sensor are DHT22, AM2302, SHT71. Which of the specification of the DHT11 sensor was helpful for your application.

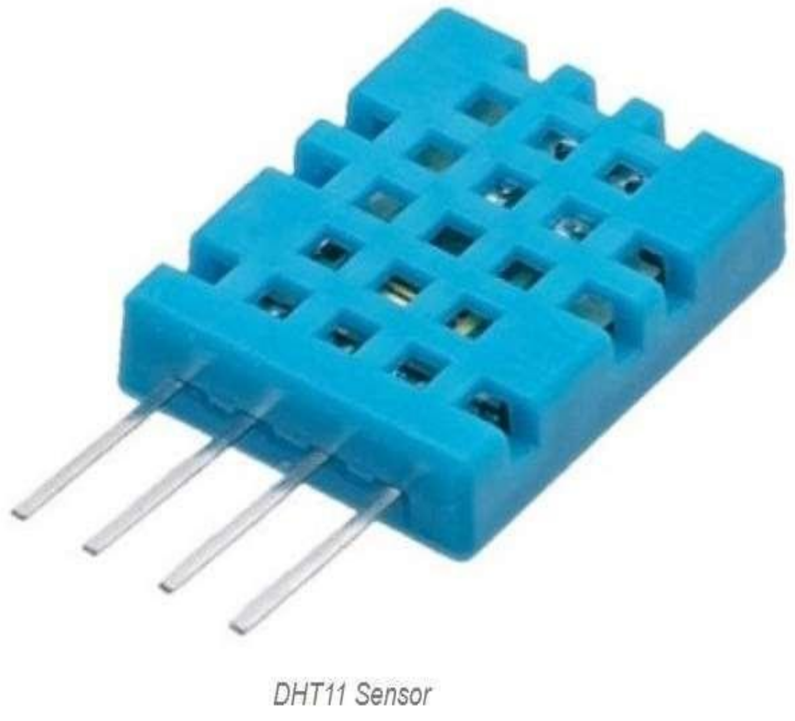


Fig 4.6 DHT11 Sensor

4.7 pH Sensor

pH sensor is used to measure hydrogen ion concentration in a solution. Glass pH electrode is widely used in pH sensors. The electrode is main part of measuring the pH in a solution. It works on the principle of voltmeter and use potential difference to check solution voltages and compare them with existing ones. It works on the principle of voltmeter and use potential difference to check solution voltages.

The electrode is main part of measuring the pH in a solution. It works on the principle of voltmeter and use potential difference to check solution voltages and compare them with existing ones. The ideal value for a solution should be pH=7 and if it is more than 7 it will a basic solution and if pH is less than 7 then solution will be acidic.

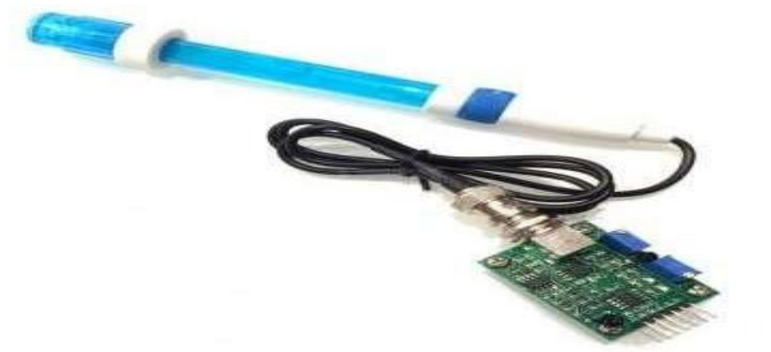


Fig 4.7 PH Sensor

4.8 Temperature Sensor

Temperature sensor plays an important role in many applications like in case of fish aquarium it is necessary to check the temperature. Temperature sensors are usually thermocouple or RTD. We have used thermistor base temperature sensor which is capable of monitoring water temperature. It works on the inverse time characteristics phenomena. The resistance of thermistor decreases when temperature increases and gives the signal of rise in temperature.

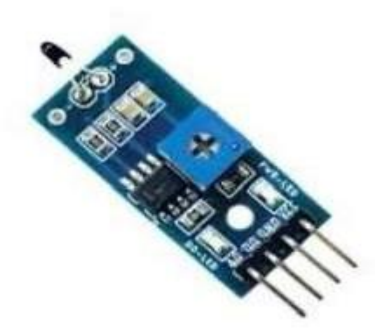


Fig 4.8 Temperature Sensor

❖ Internet of Things-IOT

The IOT concept was coined by a member of the Radio Frequency Identification (RFID) development community in 1999, and it has recently become more relevant to the practical world largely because of the growth of mobile devices, embedded and ubiquitous communication, cloud computing and data analytics. Imagine a world where billions of objects can sense, communicate and share information, all interconnected over public or private Internet Protocol (IP) networks.

These interconnected objects have data regularly collected, analysed and used to initiate action, providing a wealth of intelligence for planning, management and decision making. This is the world of the Internet of Things (IOT). Internet of things common definition is defining as: Internet of things (IOT) is a network of physical objects. The internet is not only a network of computers, but it has evolved into a network of device of all type and sizes , vehicles, smart phones, home appliances, toys, cameras, medical instruments and industrial systems, animals, people, buildings, all connected ,all communicating & sharing information based on stipulated protocols in order to achieve smart reorganizations, positioning, tracing, safe & control & even personal real time online monitoring , online upgrade, process control & administration.

We define IOT into three categories as below: Internet of things is an internet of three things People to people, People to machine /things, Things /machine to things /machine, Interacting through internet. Internet of Things Vision: Internet of Things (IoT) is a concept and a paradigm that considers pervasive presence in the environment of a variety of things/objects that through wireless and wired connections and unique addressing schemes are able to interact with each other and cooperate with other things/objects to create new applications/services and reach common goals. In this context the research and development challenges to create a smart world are enormous. A world where the real, digital and the virtual are converging to create smart environments that make energy, transport, cities and many other areas more intelligent.

4.9 Internet of Things-IOT:

The IOT concept was coined by a member of the Radio Frequency Identification (RFID) development community in 1999, and it has recently become more relevant to the practical world largely because of the growth of mobile devices, embedded and ubiquitous communication, cloud computing and data analytics. Imagine a world where billions of objects can sense, communicate and share information, all interconnected over public or private Internet Protocol (IP) networks. These interconnected objects have data regularly collected, analysed and used to initiate action, providing a wealth of intelligence for planning, management and decision making. This is the world of the Internet of Things (IOT).

Internet of things common definition is defining as: Internet of things (IOT) is a network of physical objects. The internet is not only a network of computers, but it has evolved into a network of device of all type and sizes , vehicles, smart phones, home appliances, toys, cameras, medical instruments and industrial systems, animals, people, buildings, all connected ,all communicating & sharing information based on stipulated protocols in order to achieve smart reorganizations, positioning, tracing, safe & control & even personal real time online monitoring , online upgrade, process control & administration .

We define IOT into three categories as below: Internet of things is an internet of three things People to people,   People to machine /things, Things /machine to things /machine, Interacting through internet. Internet of Things Vision: Internet of Things (IoT) is a concept and a paradigm that considers pervasive presence in the environment of a variety of things/objects that through wireless and wired connections and unique addressing schemes are able to interact with each other and cooperate with other things/objects to create new applications/services and reach common goals. In this context the research and development challenges to create a smart world are enormous. A world where the real, digital and the virtual are converging to create smart environments that make energy, transport, cities and many other areas more intelligent. In this context the research and development challenges to create a smart world.

Internet of Things is refer to the general idea of things, especially everyday objects, that are readable, recognisable, locatable, addressable through information sensing device and/or controllable via the Internet, irrespective of the communication means (whether via RFID, wireless LAN, wide area networks, or other means). Everyday objects include not only the electronic devices we encounter or the products of higher technological development such as vehicles and equipment but things that we do not ordinarily think of as electronic at all - such as food , clothing ,chair, animal, tree, water etc.

Fig 4.9: Vision of IOT

Internet of Things is a new revolution of the Internet. Objects make themselves recognizable and they obtain intelligence by making or enabling context related decisions thanks to the fact that they can communicate information about themselves. They can access information that has been aggregated by other things, or they can be components of complex services. This transformation is concomitant with the emergence of cloud computing capabilities and the transition of the Internet towards IPv6 with an almost unlimited addressing capacity. The goal of the Internet of Things is to enable things to be connected anytime, anyplace, with anything and anyone ideally using any path/network and any service.

Internet of things (IoT) is a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies. With the Internet of Things the communication is extended via Internet to all the things that surround us.2G/3G/4G,GSM,GPRS,RFID, WI-FI, GPS, microcontroller, microprocessor etc. These are considered as being the enabling technologies that make “Internet of Things” applications possible. Enabling technologies for the Internet of Things are considered in and can be grouped into three categories: (1) technologies that enable “things” to acquire contextual information, (2) technologies that enable “things” to process contextual information, and (3) technologies to improve security and privacy.

The first two categories can be jointly understood as functional building blocks required building “intelligence” into “things”, which are indeed the features that differentiate the IoT from the usual Internet. The third category is not a functional but rather a de facto requirement, without which the penetration of the IoT would be severely reduced The Internet of Things is not a single technology, but it is a mixture of different hardware & software technology. These are considered as being the enabling technologies that make “Internet of Things” applications possible. The Internet of Things provides solutions based on the integration of information technology, which refers to hardware and software used to store, retrieve, and process data and communications technology which includes electronic systems used for communication between individuals or groups.

There is a heterogeneous mix of communication technologies, which need to be adapted in order to address the needs of IoT applications such as energy efficiency, speed, security, and reliability.These are considered as being the enabling technologies that make “Internet of Things” applications possible. In this context, it is possible that the level of diversity will be scaled to a number a manageable connectivity technologies that address the needs of the IoT applications, are adopted by the market, they have already proved to be serviceable, supported by a strong technology alliance. Examples of standards in these categories include wired and wireless technologies like Ethernet, WI-FI, Bluetooth, ZigBee, GSM, and GPRS. [1, 2] The key enabling technologies for the Internet of Things is presented in Figure 2.

4.2. CHARACTERISTICS:

The fundamental characteristics of the IoT are as follows

Interconnectivity:

With regard to the IoT, anything can be interconnected with the global information and communication infrastructure.

Things-related services: The IoT is capable of providing thing-related services within the constraints of things, such as privacy protection and semantic consistency between physical things and their associated virtual things. In order to provide thing-related services within the constraints of things, both the technologies in physical world and information world will change.

Heterogeneity: The devices in the IoT are heterogeneous as based on different hardware platforms and networks. They can interact with other devices or service platforms through different networks.

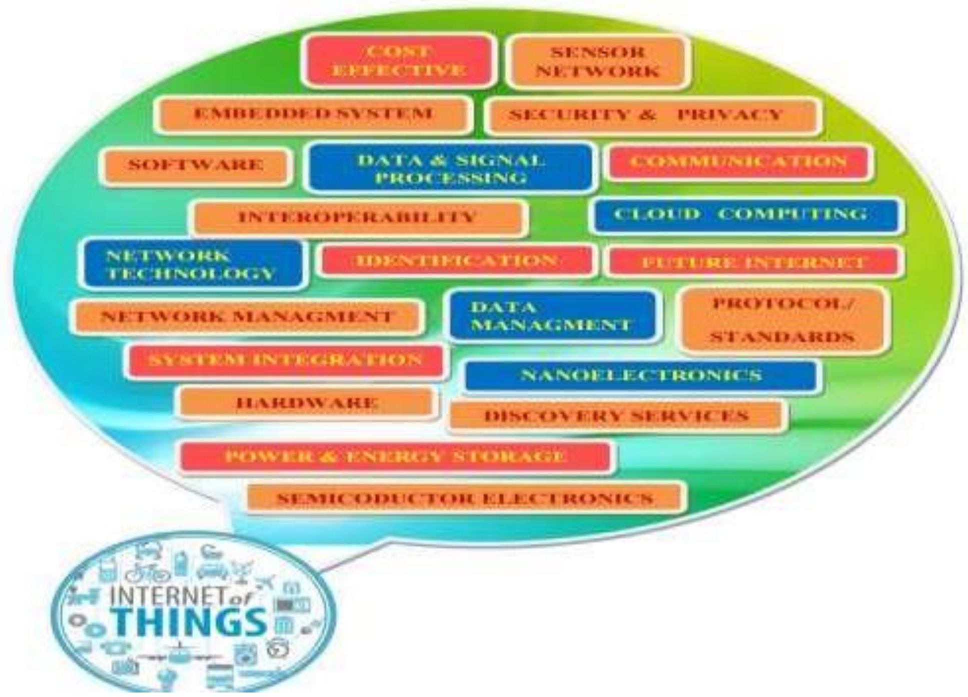


Fig 4.10: Characteristics of Internet of Things

Dynamic changes: The state of devices change dynamically, e.g., sleeping and waking up, connected and/or disconnected as well as the context of devices including location and speed. Moreover, the number of devices can change dynamically.

Enormous scale: The number of devices that need to be managed and that communicate with each other will be at least an order of magnitude larger than the devices connected to the current Internet. Even more critical will be the management of the data generated and their interpretation for application purposes. This relates to semantics of data, as well as efficient data handling.

Safety: As we gain benefits from the IoT, we must not forget about safety. As both the creators and recipients of the IoT, we must design for safety. This includes the safety of our personal data and the safety of our physical well-being. Securing the endpoints, the networks, and the data moving across all of it means creating a security paradigm that will scale.

Connectivity: Connectivity enables network accessibility and compatibility. Accessibility is getting on a network while compatibility provides the common ability to consume and produce data.

4.3 IOT ARCHITECTURE:

IOT architecture consists of different layers of technologies supporting IOT. It serves to illustrate how various technologies relate to each other and to communicate the scalability, modularity and configuration of IOT deployments in different scenarios. Figure 4 shows detailed architecture of IOT. The functionality of each layer is described below

A. Smart device / Sensor layer:

The lowest layer is made up of smart objects integrated with sensors. The sensors enable the interconnection of the physical and digital worlds allowing real-time information to be collected and processed. There are various types of sensors for different purposes. The sensors have the capacity to take measurements such as temperature, air quality, speed, humidity, pressure, flow, movement and electricity etc. In some cases, they may also have a degree of memory, enabling them to record a certain number of measurements. A sensor can measure the physical property and convert it into signal that can be understood by an instrument. Sensors are grouped according to their unique purpose such as environmental sensors, body sensors, home appliance sensors and vehicle telematics sensors, etc. Most sensors require connectivity to the sensor gateways. This can be in the form of a Local Area Network (LAN) such as Ethernet and Wi-Fi connections or Personal Area Network (PAN) such as ZigBee, Bluetooth and Ultra Wideband (UWB). For sensors that do not require connectivity to sensor aggregators, their connectivity to backend servers/applications can be provided using Wide Area Network (WAN) such as GSM, GPRS and LTE. Sensors that use low power and low data rate connectivity, they typically form networks commonly known as wireless sensor networks (WSNs). WSNs are gaining popularity as they can accommodate far more sensor nodes while retaining adequate battery life and covering large areas.

B. Gateways and Networks

Massive volume of data will be produced by these tiny sensors and this requires a robust and high performance wired or wireless network infrastructure as a transport medium. Current networks, often tied with very different protocols, have been used to support machine-to-machine (M2M) networks and their applications. With demand needed to serve a wider range of IOT services and applications such as high speed transactional services, context-aware applications, etc., multiple networks with various technologies and access protocols are needed to work with each other in a heterogeneous configuration. These networks can be in the form of a private, public or hybrid models and are built to support the communication requirements for latency, bandwidth or security. Various gateways (microcontroller, microprocessor...) & gateway networks (WI-FI, GSM, GPRS…) are shown in figure.

C. Management Service Layer

The management service renders the processing of information possible through analytics, security controls, process modelling and management of devices. One of the important features of the management service layer is the business and process rule engines. IOT brings connection and interaction of objects and systems together providing information in the form of events or contextual data such as temperature of goods, current location and traffic data.

Some of these events require filtering or routing to post processing systems such as capturing of periodic sensory data, while others require response to the immediate situations such as reacting to emergencies on patient’s health conditions. The rule engines support the formulation of decision logics and trigger interactive and automated processes to enable a more responsive IOT system.

In the area of analytics, various analytics tools are used to extract relevant information from massive amount of raw data and to be processed at a much faster rate. Analytics such as in memory analytics allows large volumes of data to be cached in random access memory (RAM) rather than stored in physical disks. In-memory analytics reduces data query time and augments the speed of decision making. Streaming analytics is another form of analytics where analysis of data, considered as data-in-motion, is required to be carried out in real time so that decisions can be made in a matter of seconds.

Data management is the ability to manage data information flow. With data management in the management service layer, information can be accessed, integrated and controlled. Higher layer applications can be shielded from the need to process unnecessary data and reduce the risk of privacy disclosure of the data source. Data filtering techniques such as data anonymisation, data integration and data synchronization, are used to hide the details of the information while providing only essential information that is usable for the relevant applications. With the use of data abstraction, information can be extracted to provide a common business view of data to gain greater agility and reuse across domains.

Security must be enforced across the whole dimension of the IOT architecture right from the smart object layer all the way to the application layer. Security of the system prevents system hacking and compromises by unauthorized personnel, thus reducing the possibility of risks.

D. Application Layer

The IoT application covers “smart” environments/spaces in domains such as: Transportation, Building, City, Lifestyle, Retail, Agriculture, Factory, Supply chain, Emergency, Healthcare, User interaction, Culture and tourism, Environment and Energy.

E. iot functional view

The Internet of Things concept refers to uniquely identifiable things with their virtual representations in an Internet-like structure and IoT solutions comprising a number of components such as : (1) Module for interaction with local IoT devices. This module is responsible for acquisition of observations and their forwarding to remote servers for analysis and permanent storage. (2) Module for local analysis and processing of observations acquired by IoT devices. (3) Module for interaction with remote IoT devices, directly over the Internet. This module is responsible for acquisition of observations and their forwarding to remote servers for analysis and permanent storage. (4) Module for application specific data analysis and processing. This module is running on an application server serving all clients. It is taking requests from mobile and web clients and relevant IoT observations as input, executes appropriate data processing algorithms and generates output in terms of knowledge that is later presented to users. (5) User interface (web or mobile): visual representation of measurements in a given context (for example on a map) and interaction with the user, i.e. definition of user queries.

4.4 FUTURE TECHNOLOGICAL DEVELOPMENTS FOR IOT.

The development of enabling technologies such as semiconductor electronics, communications, sensors, smart phones, embedded systems, cloud networking, network virtualization and software will be essential to allow physical devices to operate in changing environments & to be connected all the time everywhere.

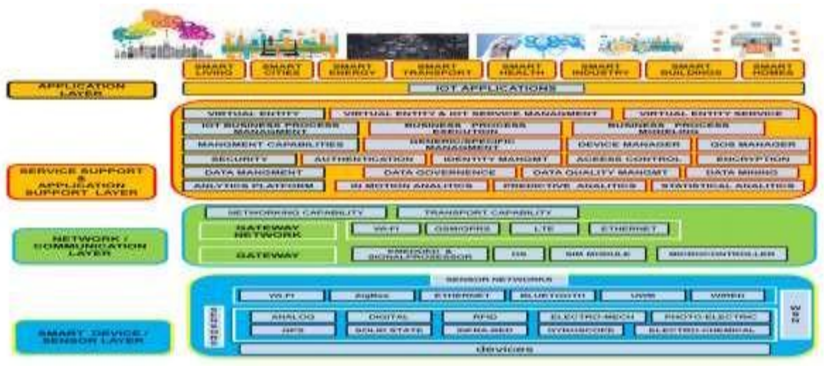


Fig 4.11: Future Technological Developments for IOT

IOT Architecture

The **Internet of Things (IoT)** architecture for a **Smart Aquarium** project is designed to connect various sensors, actuators, and controllers through a network, enabling real-time monitoring and management of the aquarium’s environment from remote locations. The IoT architecture consists of several layers, each serving a specific function to ensure data collection, processing, and communication are handled efficiently. Here's a breakdown of the IoT architecture for a Smart Aquarium project:

1. Perception Layer (Sensing Layer)

The **perception layer** is responsible for gathering data from various physical sensors and devices that monitor the environment of the aquarium. This layer includes:

- **DHT11 Sensor**: For measuring temperature and humidity inside the aquarium, ensuring theconditions remain optimal for fish and plants.
- **Water Temperature Sensor**: Measures the water temperature to maintain a consistent and healthy aquatic environment.
- **Water pH Sensor**: Monitors the pH level of the water to ensure it is within the ideal range for the aquarium's inhabitants.
- **Water Level Sensor**: Tracks water levels to prevent overflow or low water conditions, activating pumps or water dispensers when necessary.
- **Light Sensor**: Used for controlling aquarium lighting to simulate day/night cycles, which is important for both fish and plant health.
- **Water Quality Sensor**: Measures the quality of water, including parameters like turbidity, ammonia, and nitrates, which are vital for maintaining a healthy ecosystem.

These sensors collect environmental data and send it to the **microcontroller** (e.g., an **ESP8266** or **ESP32**) for further processing.

2. Network Layer (Communication Layer)

The **network layer** is responsible for enabling communication between devices and transmitting the collected data to a central location (such as a server or cloud platform). In this layer, various communication protocols are used:

- **Wi-Fi**: The Wi-Fi module (e.g., **ESP8266** or **ESP32**) connects the sensors and the microcontroller to a home Wi-Fi network, enabling the transmission of data to cloud services.
- **MQTT** (Message Queuing Telemetry Transport) or **HTTP** protocols are used to send sensor data from the microcontroller to a cloud platform, ensuring reliable and lightweight communication between devices and the cloud.

The data from the sensors is sent to cloud services like **ThingSpeak**, **Blynk**, or **Adafruit IO** for real-time data logging and visualization. This data can also be transmitted to mobile devices, allowing the user to remotely monitor and control the aquarium.

3. Processing Layer (Data Processing Layer)

The **processing layer** is where data is processed, analyzed, and acted upon based on predefined rules. This layer can be divided into:

- **Local Processing**: The microcontroller (e.g., Arduino, ESP32) reads the sensor data and processes it locally. For example, it can check whether the temperature or humidity is within the set range and activate an actuator if needed (e.g., a fan, heater, water pump, or LED light).
- **Cloud Processing**: Once the data is transmitted to the cloud, it can be analyzed and processed using cloud-based platforms. For instance, the cloud service can store historical data, provide advanced analytics, and send alerts or notifications to the user if the environmental conditions fall outside the desired parameters.

For more complex processing tasks, such as predictive maintenance or advanced analytics, you could implement edge computing techniques where some data analysis occurs closer to the sensors (on the ESP32 or another device), reducing the need to send all data to the cloud.

4. Application Layer (User Interface Layer)

The **application layer** is the topmost layer, providing the user with an interface to interact with the Smart Aquarium system. This layer involves:

- **Web Dashboard**: Using a web application or platform like **ThingSpeak** or **Adafruit IO**, the user can monitor the aquarium's status in real time, viewing parameters like temperature, humidity, water pH, and water quality. The dashboard can display graphical representations (graphs, gauges) for each sensor’s data, making it easy to understand the current status of the aquarium.
- **Mobile App**: The **Blynk** platform or custom-built mobile apps can be used to remotely monitor the aquarium. The user can receive alerts, view current sensor readings, and even control actuators (e.g., turning on/off aquarium lights, adjusting the water pump, or regulating temperature).

- **Notifications & Alerts**: If any environmental parameters fall outside the ideal range (e.g., if water temperature rises above the set threshold), the system can send **push notifications**, **SMS**, or **email alerts** to the user, prompting them.

5. Actuator Layer (Control Layer)

The **actuator layer** is where actions are taken to control the aquarium's environment based on the data received from the sensors and the decisions made by the system. Actuators are devices that physically change the environment of the aquarium. They can include:

- **Heating and Cooling Systems**: If the water temperature is too high or too low, the system can activate a **heating element** or **cooling fan** to bring the temperature back within the acceptable range.
- **Water Pumps and Filtration Systems**: The system can control water pumps to maintain water circulation and filtration, ensuring proper water quality and preventing stagnation.
- **LED Lighting**: The system can control the aquarium's lighting, simulating natural day/night cycles or adjusting light levels based on the time of day.
- **Humidifiers/Dehumidifiers**: If humidity levels are too high or too low, the system can activate a **humidifier** or **dehumidifier** to maintain the ideal range.

CHAPTER 5

BLOCK DAIGRAM AND WORKING

5.1 BLOCK DIAGRAM

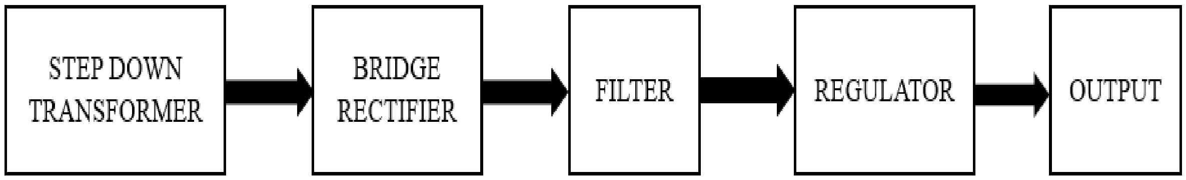


Fig: 5.1 Description of components

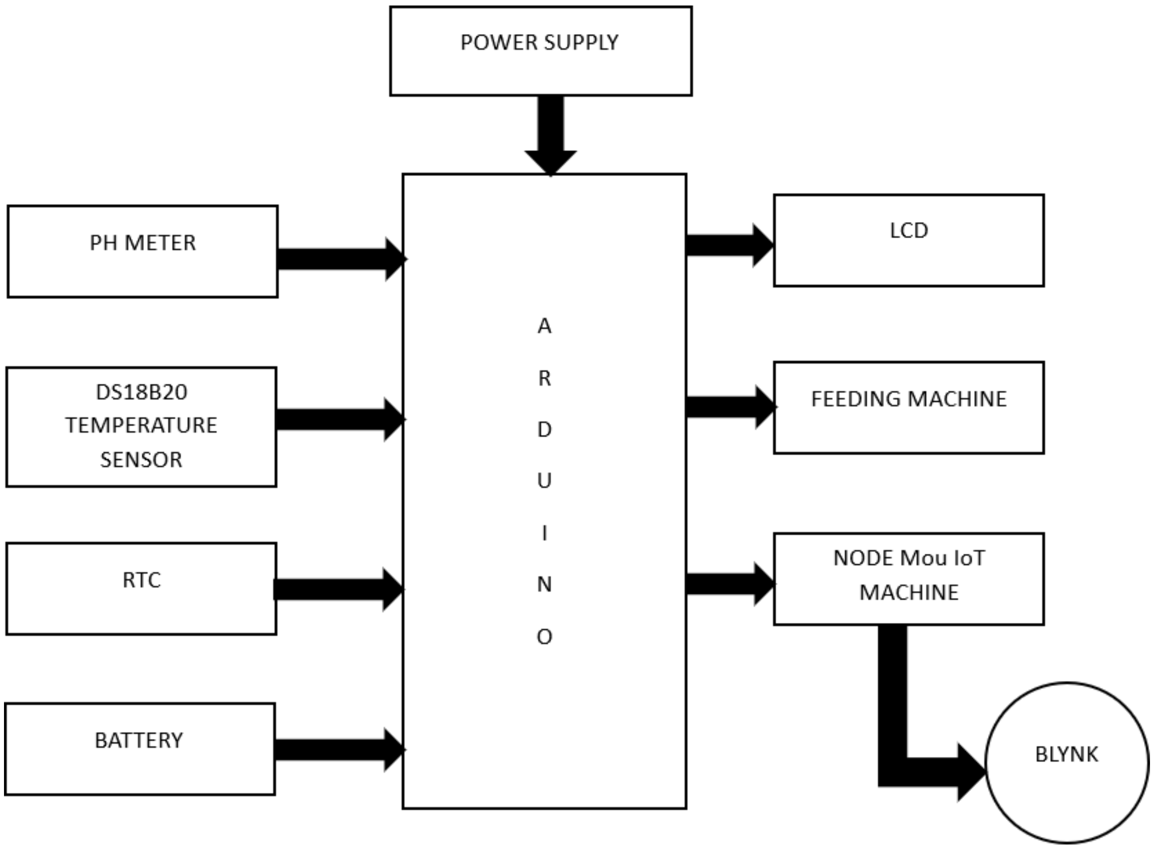


Fig: 5.2 Block diagram of Smart Aquarium

5.2 WORKING

Aquariums are a beautiful addition to homes and offices, offering a serene atmosphere and an opportunity to enjoy aquatic life. However, maintaining an aquarium requires regular attention to parameters such as water quality, temperature, pH, and the health of the aquatic organisms. This is where the concept of a "Smart Automatic Aquarium Monitoring System" using the Internet of Things (IoT) comes into play. IoT allows the creation of a connected system that automates the monitoring and control of various factors within the aquarium. With IoT-based systems, aquarium owners can remotely monitor conditions in real-time and receive alerts if any irregularities occur, ensuring optimal conditions for aquatic life.

The Smart Automatic Aquarium Monitoring System provides a user-friendly, automated, and efficient solution to maintain the delicate balance needed for fish and other aquatic organisms. It enhances the experience for aquarium owners by reducing manual effort and minimizing the chances of human error.

1) Components of the System

The IoT-based aquarium monitoring system generally comprises hardware and software components that work together to collect data, process it, and present it in a way that is useful for the aquarium owner. Below is a list of the key components:

1. Sensors

Sensors play a crucial role in monitoring the conditions inside the aquarium. Various types of sensors are used in this system to collect data about the environment within the tank.

- Temperature Sensor (DHT11/DHT22 or LM35): Monitors the water temperature to ensure it remains within the ideal range for the aquarium's inhabitants.
- pH Sensor: Measures the pH of the water, which is critical for the health of aquatic life. A pH level that is too high or too low can lead to stress or death of fish and plants.
- Water Level Sensor: Tracks the water level in the tank to ensure that it remains consistent. The sensor can detect when water needs to be refilled or if there's a potential water leakage.
- Turbidity Sensor: Measures the clarity of the water. If the water becomes cloudy, it could indicate the presence of pollutants or the need for a water change.
- Ammonia, Nitrite, and Nitrate Sensors: These sensors monitor the levels of toxic

substances in the water, which can accumulate and harm aquatic organisms.
- Light Sensor: Measures the intensity of light in the aquarium, ensuring that the lighting is optimal for both the plants and the fish.

2) Microcontroller (Arduino, Raspberry Pi, ESP8266/ESP32)

The microcontroller acts as the brain of the system. It is responsible for collecting data from the sensors, processing it, and sending it to the cloud or a local server for further analysis. Microcontrollers such as Arduino or Raspberry Pi are commonly used due to their low cost and versatility. ESP8266/ESP32 are also great choices because they come with built-in Wi-Fi, enabling easy integration with cloud platforms for remote monitoring.

3) Actuators

Actuators perform actions based on the data from sensors. These include:

- Water Pump : A water pump may be used to adjust the water level if the level sensor detects a drop in the water level.

- Heater : If the temperature sensor indicates that the water temperature is too low, the heater can be turned on automatically to maintain the optimal temperature range.

- Lights : Automated lighting can be controlled based on the time of day, or on the input from the light sensor to maintain a healthy light cycle for the aquarium inhabitants.

- Water Filter : The system can automatically turn on the water filter when the turbidity sensor detects that the water quality is decreasing.

4) Communication Module

The communication module connects the microcontroller to the cloud or remote server. In an IoT-based system, this could be a Wi-Fi module like the ESP8266 or ESP32, or a GSM module for remote communication over cellular networks. This module allows real-time data collection and remote monitoring through mobile apps or web platforms.

5) Cloud or Local Server

Data from the sensors is transmitted to the cloud or a local server for processing and storage. Cloud services like AWS, Google Cloud, or ThingSpeak can be used to store and analyze the data. This allows the owner to access the data remotely and make decisions based on real-time information.

6) Mobile Application or Web Interface

A mobile app or web interface provides the user with an easy-to-use platform to monitor and control the aquarium. The app can display real-time sensor readings, alerts, and even allow the user to control actuators like the water pump, heater, or lights remotely.

Working Procedure

The working of the Smart Automatic Aquarium Monitoring System can be divided into the following steps:

1. Data Collection

Sensors continuously monitor the parameters of the aquarium, such as temperature, pH, water level, turbidity, and light. The sensors are connected to the microcontroller (e.g., Arduino or ESP32). The microcontroller reads data from these sensors at regular intervals (e.g., every minute) and stores it in the system.

2. Data Processing

The microcontroller processes the data and compares the readings to predefined threshold values. For instance, if the temperature is above 30°C or below 22°C, it might trigger an alarm or activate the water heater or chiller. Similarly, if the pH level goes outside the optimal range (typically between 6.5 and 8.5), the system may alert the owner or activate corrective measures.

3. Data Transmission

The microcontroller sends the processed data to the cloud or a local server via a communication module such as Wi-Fi (using ESP8266/ESP32). The data is then stored on the cloud or local server for further analysis. The cloud platform may be integrated with an analytics tool to monitor the health of the aquarium over time and provide insights on when to perform maintenance or water changes. The cloud platform may be integrated with an analytics tool to monitor the health of the aquarium over time.

4. Remote Monitoring and Control

The aquarium owner can access the data via a mobile app or web interface. The app will display real-time sensor readings and send notifications if any parameter goes beyond the acceptable range. For example, if the temperature is too high, the app may send an alert to the owner, prompting them to take corrective action.

5. Automation

One of the key features of the system is automation. Based on sensor readings, the system can automatically control actuators like heaters, pumps, or filters. For instance, if the water level drops below a certain threshold, the water pump can be activated to refill the aquarium. Similarly, if the water temperature falls below a certain value, the heater will automatically turn on.

6. Alerts and Notifications

The system will notify the user via the app or web interface in case any critical parameter is outside the optimal range. These alerts are important for the early detection of potential issues, preventing catastrophic events like fish death or equipment failure.

7. Historical Data and Analytics

The cloud platform or local server can also store historical data on the aquarium's conditions. This data can be analyzed to identify trends, like water temperature fluctuations or pH changes. By understanding these patterns, the user can optimize aquarium maintenance routines and improve the overall health of the aquatic life.

CHAPTER 6

RESULT ANALYSIS AND REPORT

6.1 RESULT

- The system includes a microcontroller (possibly Arduino) mounted on a PCB with multiple connections, indicating that it is the central processing unit for automation.

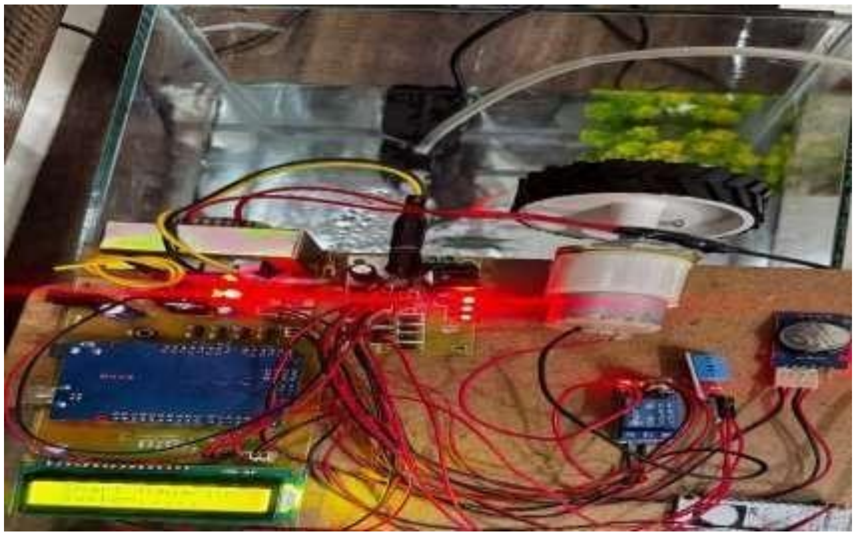


Fig 6.1: Smart automatic aquarium system

- The LCD screen shows different messages related to system operation, such as:
  - "MORNING TIME, FEEDER ON" (Indicating automatic feeding)
  - "TIME: 8:59:55, WATER PH: 7" (Showing real-time monitoring of water quality)

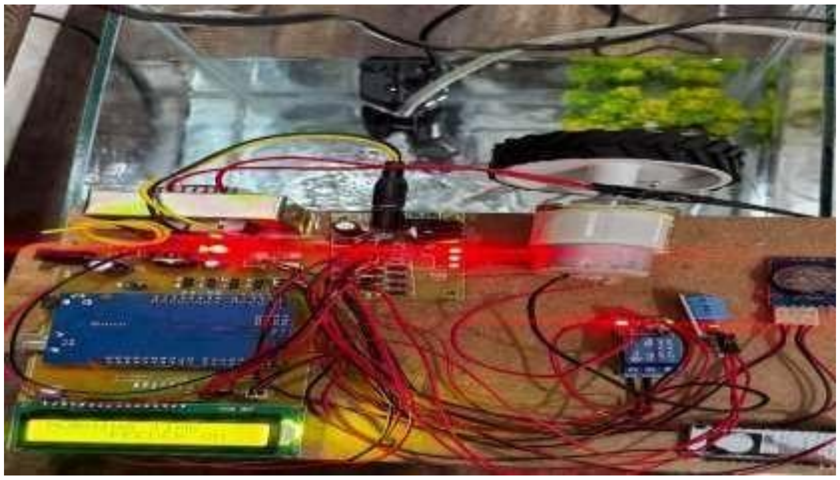


Fig 6.2: Morning Time Fedder On

- Control Buttons: At the top, there are multiple buttons labeled "ON" and "OFF." One of the "ON" buttons is highlighted, indicating that a specific function is active.

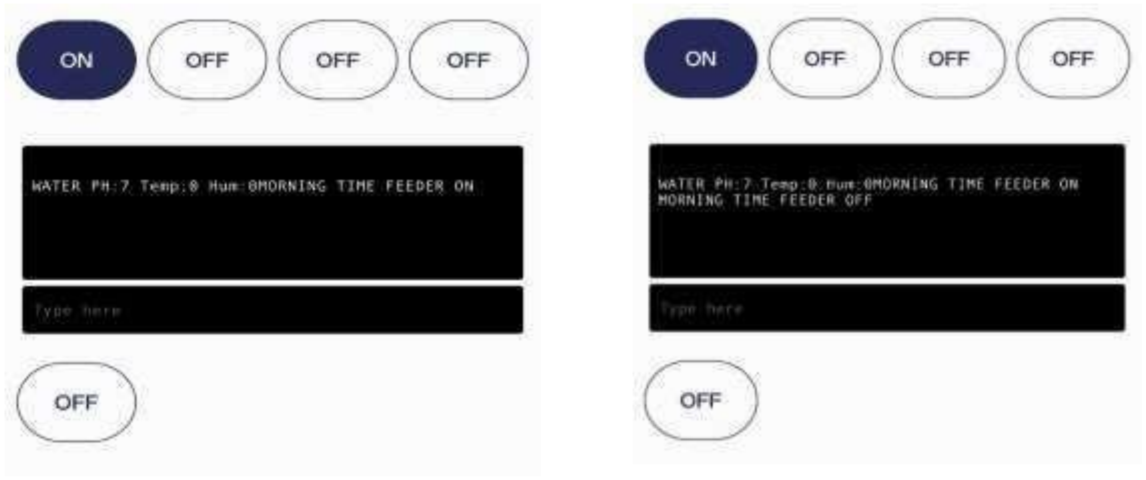

```
WATER PH:7 Temp:0 Hum:0MORNING TIME FEEDER ON
```
```
WATER PH:7 Temp:0 Hum:0MORNING TIME FEEDER ON
MORNING TIME FEEDER OFF
```

Fig 6.3: Fedder on & off in mobile application

- "TIME: 8:59:55, WATER PH: 7" (Showing real-time monitoring of water quality)

  A motorized rotating wheel attached to a container is likely dispensing fish food at scheduled intervals.

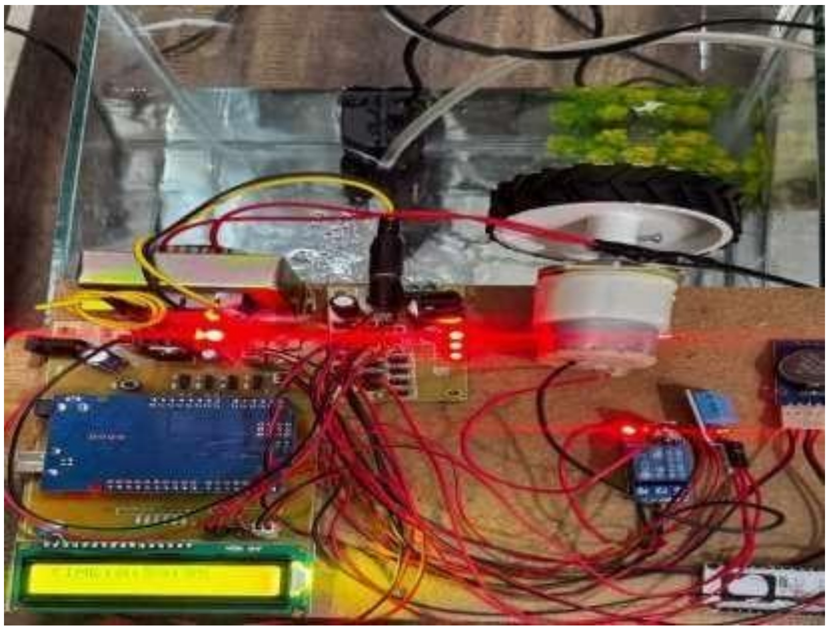


Fig 6.4: Fedder Timer on

- ➢ The system appears to automate fish feeding and monitor the water quality by measuring parameters like pH.
  - The relays may be controlling additional components like an air pump or a heater.
  - The LCD provides real-time updates on the system’s status.

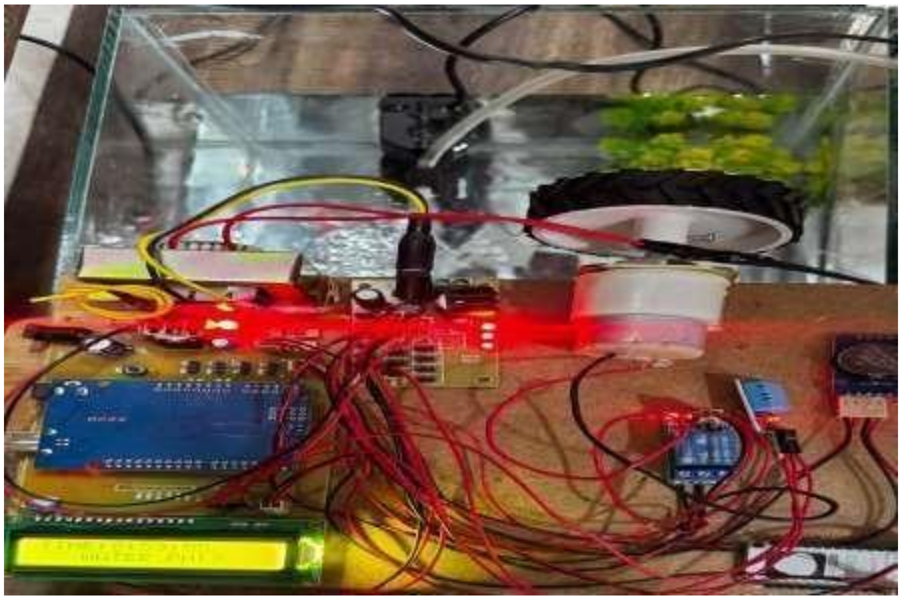
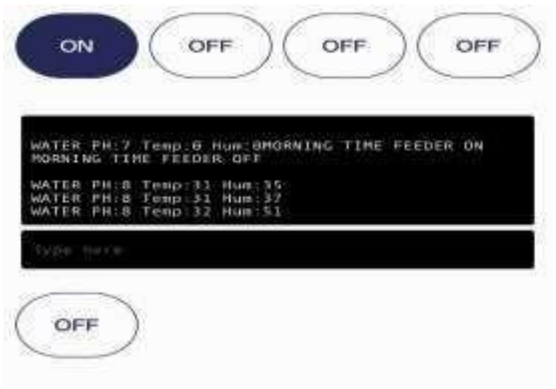


Fig 6.5: pH value detector

- ➢ Data Display Panel*: Below the buttons, there is a black text area displaying sensor data, including:
  - Water pH levels (e.g., 7 or 8)
  - Temperature readings (e.g., 0, 31, or 32)
  - Humidity values (e.g., 0, 35, 37, or 51)



```
WATER PH:7 Temp:0 Hum:0MORNING TIME FEEDER ON
MORNING TIME FEEDER OFF

WATER PH:8 Temp:31 Hum:35
WATER PH:8 Temp:31 Hum:37
WATER PH:8 Temp:32 Hum:51
```

Fig 6.6: pH value, Temperature and Humidity output in mobile application

6.2 ADVANTAGES

Real-Time Monitoring: Allows users to monitor the aquarium's environment (temperature, humidity, pH, etc.) in real-time, from anywhere, anytime.

Automation: Automates routine tasks like temperature control, water filtration, and lighting, reducing the need for manual intervention and ensuring optimal conditions for aquatic life.

Remote Control: Enables remote control of the aquarium's settings (lighting, water pumps, heater, etc.) via a mobile app or web dashboard, even when the user is away.

Proactive Alerts and Notifications: Sends instant alerts to the user if any environmental parameter (temperature, pH, water quality) goes beyond the desired range, allowing for quick corrective actions.

Data Logging and Analysis: Continuously records data over time, allowing users to track trends and identify potential issues or improvements in the aquarium's ecosystem.

Improved Health of Aquarium Life: Ensures the aquarium environment stays within the ideal range, enhancing the health and well-being of fish and plants by preventing harmful fluctuations.

Reduced Human Error: Minimizes the chance of human error in monitoring and maintaining the aquarium, leading to more consistent care and fewer risks to aquatic life.

Integration with Other IoT Devices: The system can be expanded and integrated with other IoT devices (e.g., automated feeders, home automation systems) for a fully synchronized and smart setup.

Convenience and Time-Saving: Saves time and effort by automating the monitoring and control processes, making aquarium care more convenient, especially for busy or traveling users.

Long-Term Insights: Provides users with valuable historical data, helping them make informed decisions for better aquarium maintenance and long-term health of the ecosystem.

Energy Efficiency: The system can optimize energy consumption by automating lighting, heating, and filtration based on real-time data, ensuring that only necessary systems are active at the right times.

Customization: Users can tailor the system to meet the specific needs of their aquarium, such as adjusting temperature and lighting schedules to match the preferences of different fish species or plants.

Scalability: The system is easily scalable, allowing users to add more sensors, actuators, or even integrate with other IoT systems as their aquarium needs grow or change over time.

Increased Convenience for Large Aquariums: For larger or more complex aquariums, the IoT system provides a streamlined way to manage and monitor numerous variables across multiple tanks or sections, reducing the complexity of manual checks.

Cloud-Based Access: Cloud integration enables users to access and manage their aquarium remotely, eliminating the need for a local device, while ensuring data is securely stored and accessible from any device.

Improved Aquarium Aesthetics: The automation of lighting and environmental control contributes to more natural and visually pleasing aquarium setups, mimicking natural day/night cycles and maintaining the right ambiance.

Cost-Effective: By automating several aspects of aquarium care, the system helps users save on costs related to overuse of energy, water treatment products, and equipment maintenance.

Peace of Mind: With automated monitoring and alert systems in place, users can feel confident that their aquarium's ecosystem is being closely managed and protected, even when they are not physically present.

Educational Value: The system provides an educational platform for those learning about IoT technology, sensors, and environmental management, helping users gain hands-on experience in building and maintaining smart devices.

Enhanced Security: The integration of cameras or motion sensors can provide additional security, ensuring that the aquarium is not tampered with and monitoring any unusual events (like power failures or equipment malfunctions).

Integration with Weather Data: The system can integrate with external weather data (e.g., temperature and humidity from local weather stations) to adapt the aquarium's internal environment based on external conditions, improving efficiency and responsiveness.

Customizable Alerts: Users can set custom thresholds for alerts, allowing them to focus on the specific environmental factors that matter most to them, whether it's water temperature, pH, or any other parameter.

Minimized Stress on Aquatic Life: The system helps maintain stable and optimal conditions, reducing stress for fish and plants. Consistent conditions are crucial for promoting natural behavior and healthy growth in the aquarium's inhabitants.

Prevention of Overfeeding: If the system is integrated with an automatic feeding mechanism, it can prevent overfeeding by dispensing food only when required, ensuring a balanced diet for the fish while also preventing water contamination.

24/7 Monitoring: Continuous 24/7 monitoring ensures that no environmental changes go unnoticed, providing users with peace of mind that their aquarium is always in check, even when they are asleep or away from home.

Efficient Resource Management: By automating the control of water pumps, heaters, and filters, the system ensures resources like electricity and water are used efficiently, reducing waste and helping conserve energy.

Automatic Calibration: Some IoT systems can automatically calibrate sensors to maintain accuracy over time, reducing the need for manual intervention and ensuring reliable data for decision-making.

Support for Multiple Aquariums: The system can be extended to monitor and control multiple aquariums simultaneously, making it ideal for businesses, research facilities, or hobbyists managing several tanks at once.

Improved Fish Breeding Conditions: By maintaining optimal environmental conditions, the system can help improve breeding conditions for aquatic life, supporting a more natural and conducive environment for breeding.

Simplified Troubleshooting: If something goes wrong, the system can diagnose issues, alert the user, and provide detailed data logs to help troubleshoot and fix the problem without the need for expert intervention.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

In conclusion, the Smart Automatic Aquarium Monitoring System using IoT offers a transformative approach to managing aquariums, combining technology with convenience to create an efficient and reliable solution for aquarium care. By integrating sensors, actuators, and IoT connectivity, this system ensures real-time monitoring and automation of critical environmental parameters such as temperature, humidity, water quality, and pH levels, ultimately creating a healthy and stable ecosystem for fish and plants.

The advantages, including remote monitoring, automated control, energy efficiency, and proactive alerts, significantly reduce the effort and time required for traditional aquarium management, while also minimizing the risk of human error. Users can maintain optimal conditions and easily troubleshoot any issues from anywhere in the world, making it particularly beneficial for busy individuals or those who frequently travel. Additionally, the system's scalability and flexibility allow for future upgrades and expansions, ensuring it stays relevant as technology advances.

With enhanced monitoring, energy efficiency, and a focus on long-term sustainability, the Smart Aquarium System powered by IoT offers a reliable and smart solution for aquarium enthusiasts and professionals alike, making aquarium management more convenient, cost-effective, and precise. This innovative approach not only simplifies day-to-day operations but also enhances the overall health of the aquarium, promoting a thriving aquatic environment.

The advantages of real-time data collection, remote control, and instant alerts provide peace of mind and enable users to intervene quickly in case of any issues. Whether it's adjusting temperature, controlling lighting, or triggering a water change, the system makes it easier to maintain an optimal aquatic environment. The system also contributes to cost savings by optimizing energy use and reducing waste, offering both financial and environmental benefits,

Moreover, the IoT-based aquarium monitoring system enhances the overall user experience, making it easier for beginners and experts alike to manage their aquariums effectively.

Ultimately, this system not only simplifies aquarium care but also empowers users with the tools and insights they need to create and sustain vibrant aquatic ecosystems. With its combination of automation, convenience, and advanced technology, the **Smart Aquarium Monitoring System using IoT** has the potential to revolutionize the way we care for aquariums, bringing a new level of ease, precision, and efficiency to aquarium management. With its combination of automation, convenience, and advanced technology, the **Smart Aquarium Monitoring System using IoT.**

7.2 FUTURE SCOPE

- Integration with Artificial Intelligence (AI): Future systems could incorporate AI algorithms to predict potential issues in the aquarium environment based on historical data and patterns, allowing for proactive adjustments and automatic decision-making.
- Advanced Water Quality Monitoring: More sophisticated sensors could be introduced to detect additional water quality parameters like ammonia, nitrates, and dissolved oxygen, providing a more comprehensive understanding of the aquarium's health.
- Integration with Smart Home Systems: The system could be integrated with popular smart home platforms like Amazon Alexa, Google Home, or Apple HomeKit for voice-controlled management and synchronization with other smart devices within the home.
- Machine Learning for Customization: The system could learn from the user’s behaviour and the aquarium’s unique conditions, offering personalized suggestions for temperature, lighting, and feeding schedules based on the specific needs of different species.
- Predictive Maintenance: By integrating sensors that monitor the performance of key equipment (like pumps, heaters, and filters), the system could predict when maintenance or replacement is required, minimizing downtime and ensuring the longevity of devices.
- Enhanced Mobile App Features: Future versions of the mobile app could include augmented reality (AR) features, allowing users to visualize and interact with their aquarium's environment more intuitively on their smartphone or tablet.

- Remote Water Change Automation: The system could introduce automation for water changes, allowing for the controlled addition of fresh water or removal of old water without manual intervention.
- Data Sharing and Community Integration: Future systems could allow users to share data and insights with other aquarium enthusiasts or participate in community-driven projects, fostering collaboration and knowledge sharing.
- Integration with Weather Forecasting: The system could adapt the aquarium's settings based on weather forecasts, adjusting temperature and lighting in response to expected external environmental changes (e.g., a cold front or heatwave).
- Solar-Powered Aquariums: Incorporating solar power into the IoT-based system could make it more eco-friendly by using renewable energy to power the sensors and actuators, reducing the carbon footprint of running the system.
- Multi-Aquarium Management: The system could evolve to allow management of multiple aquariums from a single dashboard, ideal for businesses, research centers, or large-scale aquarium setups.
- Cloud-Based Analytics and Insights: More advanced analytics could be provided via cloud computing, offering long-term trend analysis, and recommending actions based on data across multiple aquariums or multiple users.
- AI-Driven Fish and Plant Health Monitoring: AI could analyze fish behavior and plant health to detect early signs of disease or distress, sending alerts to users and suggesting remedies.
- Biometric Sensors for Fish Health: Future systems may use biometric sensors to track the health of the fish in more detail, monitoring factors like heart rate or stress levels to ensure their well-being.
- Automated Breeding Support: The system could include features that monitor and control breeding conditions for fish, ensuring the right parameters are met to foster successful breeding.
- Blockchain for Data Security: Blockchain could be used to secure data related to the aquarium's environment, ensuring that historical records are tamper-proof and that sensitive information is protected.
- Global Connectivity: With 5G technology, the system could offer ultra-fast, global

connectivity for remote users to access and manage their aquariums with near-instantaneous data transmission, enhancing real-time responsiveness.

- Aquarium Ecosystem Simulation: Future IoT systems could offer ecosystem simulation tools that allow users to simulate and test changes to the environment (such as adding new species or adjusting water conditions) before applying them to the real aquarium
- The **integration with smart home systems** like Amazon Alexa or Google Home could provide users with a more seamless, voice-activated way of managing their aquariums. The system could also become **more energy-efficient** with the introduction of solar-powered components and enhanced automation to reduce electricity consumption. As the system becomes more advanced, it might enable features such as **automated water changes** and **automated fish feeding** based on activity levels, ensuring that the ecosystem remains stable and efficient with minimal human intervention.
- **Cloud-based analytics** could provide users with deeper insights into long-term trends, allowing them to make more informed decisions about tank maintenance and species care. Furthermore, with the growth of **5G connectivity**, the system could offer real-time data transmission, enabling immediate responses to changing environmental conditions and improving user experience. For those managing multiple aquariums, future systems might include **multi-tank management**, enabling users to monitor and control several aquariums from a single dashboard.
- As technology advances, the system could also include features such as **voice-controlled feedback**, **smart water conservation tools**, and **real-time fish and plant health monitoring** through advanced sensors. Ultimately, the system will evolve into a more **intelligent, automated, and user-friendly solution**, helping aquarium enthusiasts manage their tanks efficiently while fostering the health and sustainability of aquatic life. With these innovations, the **future scope** of IoT-enabled aquarium monitoring systems promises to revolutionize the way aquariums are maintained, providing a seamless and highly efficient experience for both beginners and seasoned aquarium hobbyists.

REFERENCE

[1] G. G. Raj, S. S. R. Anjaneyulu, and B. A. Babu, "IoT based Smart Aquarium Monitoring System," International Journal of Engineering Research and Technology, vol. 8, no. 7, pp. 613-618, 2023.

This paper discusses the use of IoT technology in aquarium monitoring systems, focusing on sensors, data collection, and control mechanisms for maintaining optimal conditions in aquariums.

[2] S. T. K. R. Reddy, P. S. Kumar, and G. S. R. A. Prasad, "Smart Aquarium Monitoring System using IoT," International Journal of Computer Applications, vol. 179, no. 13, pp. 10-14, 2021.

The study explores various applications of IoT in aquarium systems, discussing remote monitoring, automation of environmental parameters, and user notifications.

[3] H. M. Ali and W. A. S. M. Ali, "IoT-Based Smart Aquarium," International Conference on Smart Computing and Communication Systems (ICCS), 2020.

This paper presents a smart aquarium solution that leverages IoT, with sensors for water quality monitoring and a mobile app for remote control and management.

[4] S. Pandey, R. Singh, and R. Bansal, "IoT based Smart Aquarium Monitoring and Control System," 2018 3rd International Conference on Advances in Computing, Communication, and Control (ICAC3), 2020.

Discusses the architecture of an IoT-based system that monitors and controls environmental factors in aquariums, providing a complete solution for aquarium management.

[5] J. A. Garcia, R. Ruiz, and R. Garcia, "Design of an IoT-based Automated Fish Tank Monitoring System," 2019 IEEE World Conference on Industrial Applications (WCIA), 2019.

This paper presents a detailed design for an automated fish tank monitoring system, focusing on data acquisition, processing, and remote monitoring through IoT.

[6] A. Sharma, P. G. Z. Rana, and M. J. B. Simson, "A Smart Aquarium Monitoring System Using IoT," 2020 IEEE International Conference on Computing and Communications Technologies (ICCCT), 2020.

This article explores the different IoT sensors used in a smart aquarium system.

[7] K. M. M. Rajan and D. S. Sharma, "An IoT-based Aquarium Monitoring System for Water Quality," International Journal of Advanced Research in Computer Science, vol. 9, no. 5, pp. 289-293, 2024.

This paper focuses on the IoT-based monitoring of water quality, temperature, and pH levels, with automated control features to optimize aquarium conditions.

[8] R. D. T. S. Ashok and R. M. R. Sharma, "Development of an IoT-based Smart Fish Tank," 2019 IEEE Conference on Smart Technologies, 2019.

Discusses the integration of IoT technologies in smart fish tanks, highlighting sensor data collection, remote monitoring, and automated control of aquarium parameters.

[9] Patil, Tarun, T. Sri Srujan Hari, and M. Anitha. "Real-Time Smart Aquarium Monitoring System Driven by IoT." *2025 International Conference on Sustainable Energy Technologies and Computational Intelligence (SETCOM)*. IEEE, 2025.

[10] Sheela, M. Sahaya, et al. "Empowering aquarists a comprehensive study on IOT-enabled smart aquarium systems for remote monitoring and control." *Babylonian Journal of Internet of Things* 2024 (2024): 33-43.

[11] Asokan, Anju, et al. "Smart Aquarium and Water Quality Monitoring using IoT." *2024 International Conference on Cognitive Robotics and Intelligent Systems (ICC-ROBINS)*. IEEE, 2024.

[12] Rey, William P., and Kieth Wilhelm Jan D. Rey. "Smart AroTank: an IOT-based Smart Aquarium Management System for Arowana Fish Keepers." *Proceedings of the 2024 10th International Conference on Computing and Data Engineering*. 2024.

APPENDIX:

Appendix A: System Components and Hardware Specifications

1. Microcontroller:
   - Model: ESP8266 / ESP32
   - Function: Wi-Fi module for connecting the system to the internet.
   - Pinout: List of pins used for various sensors and actuators.
2. Sensors:
   - Temperature Sensor: DS18B20
     - Range: -55°C to +125°C
     - Accuracy: ±0.5°C
   - pH Sensor: Analog pH sensor
     - Range: 0-14 pH
     - Accuracy: ±0.1 pH
   - Water Level Sensor: Ultrasonic sensor (HC-SR04)
     - Range: 2cm to 400cm
     - Accuracy: ±3mm
3. Actuators:
   - Water Pump: 12V submersible pump
   - Air Pump: 5V DC air pump
4. Power Supply:
   - Voltage: 12V DC Adapter
   - Backup: Battery (for backup power in case of failure)
5. Other Components:
   - Relay Module: For controlling high voltage devices like pumps and heaters.
   - LCD Display: 16x2 LCD to display system parameters locally.
   - LEDs: For visual indications of system status.

Appendix B: Circuit Diagram

Include the complete circuit schematic diagram of the Smart Aquarium Monitoring System. The diagram should illustrate how the sensors, actuators, microcontroller, and power supply are connected.

Cloud Integration and Data Visualization

1. Cloud Platform Setup:
   - Cloud platforms like ThingSpeak or Blynk are used to store and visualize data. Both offer API keys for easy integration with the IoT devices.
   - Data is sent using HTTP/MQTT requests, where each sensor's data is mapped to a specific channel or field in the platform.
   - Example setup: ThingSpeak channel with fields for temperature, pH, oxygen, and water level.
2. Dashboard and Alerts:
   - The cloud platform provides a web-based dashboard where users can view graphs and charts of the aquarium's environmental parameters.
   - Alerts: Notifications are sent to the user's phone or email when values exceed predefined thresholds (e.g., pH drops below 6.5 or temperature exceeds 28°C).
   - Example of data visualization: A graph plotting the pH level over time, helping users track trends and make adjustments.

Sensors and IoT Communication

1. **Sensor Data Collection**:
   - The sensors collect data continuously or at set intervals (e.g., every minute or every 5 minutes) depending on the system's configuration. The data is then processed and sent to the cloud.
2. **Sensor Communication**:
   - The microcontroller (e.g., ESP32) reads data from the sensors via **Analog-to-Digital Conversion (ADC)** or digital inputs.
   - The sensor data is then formatted and sent to the cloud platform via **Wi-Fi** using **MQTT** or **HTTP** protocols.

CODE

```
 include <SimpleDHT.h>
int pinDHT11 = 2;
SimpleDHT11 dht11(pinDHT11);
include<SoftwareSerial.h>
SoftwareSerial gps(6,7);
  include <Wire.h>
  include "Sodaq_DS3231.h"
  include <LiquidCrystal.h>
const int rs = 13, en = 12, d4 =11, d5 = 10, d6 = 9, d7 = 8;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
char weekDay[][10] = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday",
"Friday", "Saturday" };
char monthDay[][10] = {" ","Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul"
,"Aug","Sep","Oct","Nov","Dec"};
int val;
int bell=5;
void setup ()
{
  pinMode(bell,OUTPUT);digitalWrite(bell,HIGH);
   lcd.begin(16,2);
    lcd.clear();
    Serial.begin(9600);delay(1000);
    gps.begin(9600);///Serial.println("----WELCOME TO PROJECT --- ");delay(1000);
    Wire.begin();
    rtc.begin();
DateTime dt(2023, 12, 14, 8, 59, 50, 0);
 rtc.setDateTime(dt); //Adjust date-time as defined 'dt' above
 lcd.clear();lcd.print(" Smart Aquarium");
 lcd.setCursor(0,1);lcd.print(" System ....");delay(2000);
}
```

```
void loop ()
{
  back:
 // read without samples.
  byte temperature = 0;
  byte humidity = 0;
  int err = SimpleDHTErrSuccess;
  if ((err = dht11.read(&temperature, &humidity, NULL)) != SimpleDHTErrSuccess)
  {
   // Serial.print("Read DHT11 failed, err="); Serial.print(SimpleDHTErrCode(err));
   // Serial.print(","); Serial.println(SimpleDHTErrDuration(err)); delay(1000);
    return;
  }
  //Serial.print((int)temperature); Serial.print("  C, ");
 // Serial.print((int)humidity); Serial.println(" H");
  int temp=((int)temperature);
  int hum=((int)humidity);delay(1500);
 lcd.clear();lcd.print("TEMP:");lcd.print(temp);lcd.print("
HUM:");lcd.print(hum);delay(hum);delay(1000);
  char ph1[]="";
    delay(1000);DateTime now = rtc.now();
    lcd.clear(); lcd.setCursor(0,0);
    lcd.clear();lcd.print("TIME:");
    if(now.hour()>12)
    {
    lcd.print(now.hour()-12, DEC);lcd.print(':');
    }
    else
    {
     lcd.print(now.hour(), DEC);lcd.print(':');
    }
    lcd.print(now.minute(),  DEC);lcd.print(':');
    lcd.print(now.second(), DEC);lcd.print(' ');
   lcd.print("  ");delay(1000);
 if((now.hour()==9)&&(now.minute()==0)&&((now.second()>=0)
)&&(now.second()<=15) )
    {
    Serial.print("MORNING TIME FEEDER ON\r\n");
   lcd.clear();lcd.print("MORNING TIME");delay(1000);
   lcd.setCursor(0,1);lcd.print("      FEEDER ON
");digitalWrite(bell,LOW);delay(3000);delay(3000);
    lcd.setCursor(0,1);lcd.print("      FEEDER
OFF");digitalWrite(bell,HIGH);delay(3000);
     Serial.print("MORNING TIME FEEDER OFF\r\n");
    }
    else if((now.hour()==9)&&(now.minute()==5)&&((now.second()>=0)
)&&(now.second()<=15) )
    {
       Serial.print("AFTERNOON TIME FEEDER ON\r\n");
      lcd.clear();lcd.print("AFTERNOON TIME");delay(1000);
   lcd.setCursor(0,1);lcd.print("      FEEDER ON
");digitalWrite(bell,LOW);delay(3000);delay(3000);
    lcd.setCursor(0,1);lcd.print("      FEEDER
OFF");digitalWrite(bell,HIGH);delay(3000);
     Serial.print("AFTERNOON TIME FEEDER OFF\r\n");
    }
    else if((now.hour()==9)&&(now.minute()==10)&&((now.second()>=0)
)&&(now.second()<=15) )
    {
       Serial.print("EVENING TIME FEEDER ON\r\n");
   lcd.clear();lcd.print("EVENING TIME");delay(1000);
   lcd.clear(); lcd.setCursor(0,1);lcd.print("     FEEDER ON
");digitalWrite(bell,LOW);delay(3000);
    lcd.setCursor(0,1);lcd.print("      FEEDER
OFF");digitalWrite(bell,HIGH);delay(3000);
     Serial.print("EVENING TIME FEEDER OFF\r\n");
    }
    String data1=gps.readString();
 if(data1[9]==',')
 {
  //ph=String(data1[5])+String(data1[6])+String(data1[7])+String(data1[8]);
  ph1[0]=data1[5];
  ph1[1]=data1[6];
  ph1[2]=data1[7];
  ph1[3]=data1[8];
  ph1[4]='\0';
 }
 else
 {
//ph=String(data1[5])+String(data1[6])+String(data1[7])+String(data1[8])+String(data1[
9]);
  ph1[0]=data1[5];
  ph1[1]=data1[6];
  ph1[2]=data1[7];
  ph1[3]=data1[8];
  ph1[4]=data1[9];
 }
 int phval = atoi(ph1);
 if(phval>30)
 {
 lcd.clear();lcd.print("PLEASE DIP PH");
 lcd.setCursor(0,1);lcd.print("METER IN WATER");delay(1000);goto back;
 }
```

```
lcd.setCursor(0,1);lcd.print("  WATER PH:");lcd.print(phval);lcd.print("
");delay(2000);
String iot="\r\nWATER PH:"+String(phval)+" Temp:"+String(temp)+"
Hum:"+String(hum);delay(1000);
 Serial.print(iot);delay(1000);
  while(Serial.available())
  {
    String data1=Serial.readString();
    if(data1[0]=='2')
    {
    lcd.clear();lcd.print("FROM IOT");delay(1000);
   lcd.setCursor(0,1);lcd.print("      FEEDER ON
");digitalWrite(bell,LOW);delay(3000);delay(3000);
    lcd.setCursor(0,1);lcd.print("      FEEDER
OFF");digitalWrite(bell,HIGH);delay(3000);
    }
  }
}
```