AN INTERNSHIP REPORT ON

Technical Training on IOT and EMBEDDED SYSTEMS

Submitted in the fulfilment of the requirements for

the award of the degree of

Bachelor of Technology
in
Electronics and Communication Engineering

P. Rajesh
[201FA05047]

Under the Esteemed Guidance of
Dr. Y. Srinivasa Rao
Chairman of
GO-GREEN Technologies Pvt. Ltd.

Under the Internal Guidance of
Dr. P. J. Reginald
Asst Professor
Department of ECE

(ACCREDITED BY **NAAC** WITH **"A^{+}"** GRADE)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
(ACCREDITED BY **NBA**)

VIGNAN'S FOUNDATION FOR SCIENCE, TECHNOLOGY AND RESEARCH
(Deemed to be University)

Vadlamudi, Guntur, Andhra Pradesh, India -522213

May 2024

Yerramreddy's **Go-Green Technologies** Pvt. Ltd.

GSTN: 27AABCY4791J1ZS SP TBI, 8th Floor, Sardar Patel Institute of Technology, Munshi Nagar, Bhavans Campus, Andheri (West), Mumbai-58, India

Web: ydengg.co.in E-Mail: gogreenramakrishna@gmail.com Mob: 9820962870

To,

Date: 31/05/2024

P. Rajesh, B. Tech, ECE,
Vignan's Foundation for Science, Technology & Research,
Deemed to be University, Vadlamudi, Guntur (Dist), AP.

Subject: Internship Completion Certificate

Reference: Offer letter dated 15th January 2024

This is to certify that P. Rajesh, a student of Vignan's Foundation for Science, Technology & Research, has successfully completed his Research Internship Program in "Internet of Things and its Applications" under the guidance and supervision of Dr. Y. Srinivasa Rao, Managing Director of Yerramreddy's Go-Green Technologies Pvt. Ltd., Mumbai, from 15th January 2024 to 15th May 2024.

During his internship, Rajesh worked on various case studies related to Embedded Technologies and their applications. He also visited our incubation center at SP- TBI, Bhavans Campus, Andheri-W, Mumbai, and participated in a two-week hands-on training session on "Electronic Product Fabrication Technologies" under AICTE IDEALAB.

He fabricated his own Do-it-Yourself EDU-Robo IoT IDE and participated in projects such as "IoT-based Musical Garden," "Concentrated Solar Power," and "HMI for Induction Heating," successfully completing all assigned tasks.

Throughout his internship, Rajesh demonstrated proficiency, knowledge, and a result-oriented approach, combining both theoretical and practical understanding. His performance exceeded our expectations, and he consistently completed his work on time.

This internship certificate is a token of our appreciation for his valuable contributions to Go-Green Technologies Pvt. Ltd. We wish him success in his future endeavors and are confident that the skills he gained during this internship will significantly contribute to his professional growth.

Dr Y S Rao,
B.E. (Electronics), M.E (Computer),Ph.D. (ES, IIT-Bombay)
Founder Director, Yerramreddy's Go Green Technologies Pvt Ltd,
Dean Academics and Research, Vice Principal, Bharatiya Bhavans Sardar Patel Institute of Technology, Andheri-W, Mumbai. E-Mail: ysrao@spit.ac.in, Mob: 9820962870

CERTIFICATE

This is to certify that the internship report entitled "**Technical Training on IOT and EMBEDDED SYSTEMS**" that is being submitted by P. Rajesh [201FA05047],in partial full fillment for the award of B. Tech degree in Electronics and Communication Engineering to Vignan's Foundation for Science Technology and Research University, is a record of Bonafide work carried out by here at GO-GREEN Technologies Private Limited under the supervision of Dr. Y Srinivasa Rao and the internal guidance of **Dr. P. J. Reginald** of ECE Department.

1/6/24

Signature of the guide

Dr. P. J. Reginald

Asst Professor

1/6/24.

Signature of Head of the Department

Dr. T. Pitchaiah, M.E, Ph.D., MIEEE, FIETE

Professor & HoD ECE

Signature of

INTERNAL EXAMINAR

01/06/24

Signature of

EXTERNAL EXAMINAR

DECLARATION

I hereby declare that the internship work entitled "**Technical Training on IOT and EMBEDDED SYSTEMS**" is being submitted to Vignan's Foundation for Science, Technology and Research (Deemed to be University) in partial fulfillment for the award of B. Tech degree in Electronics and Communication Engineering. The work was originally designed and executed by group members under the guidance of my supervisor Dr. Y. Srinivasa Rao with Dr. P. J. Reginald as faculty guide at Department of Electronics and Communication Engineering, Vignan's Foundation for Science Technology and Research (Deemed to be University) and was not a duplication of work done by someone else. I hold the responsibility of the originality of the work incorporated into this internship report.

P. Rajesh Babu

Signature of the candidate

P. Rajesh [201FA05047]

ACKNOWLEDGEMENT

Firstly, we would like to thank **Dr. Y. Srinivas Rao**, Founder Director of Yerramreddy's Go Green Technologies Pvt. Ltd for giving me the opportunity to do an internship within the organization.

We are highly indebted to **Dr. Y. Srinivas Rao**, Founder Director of Yerramreddy's Go Green Technologies Pvt. Ltd who monitored and trained me well personally throughout my internship.

We are greatly indebted to **Dr. P. J. Reginald,** our revered guide and Assistant Professor in the Department of Electronics and Communication Engineering, VFSTR (Deemed to be University), Vadlamudi, Guntur, for his valuable guidance in the preparation of this dissertation. He has been a source of great inspiration and encouragement to us. He has been kind enough to devote considerable amount of his valuable time in guiding us at every stage. This is our debut, but we are sure that we can do many more such studies, purely become of the lasting inspiration and guidance given by respectable guide.

We would like to thank **Dr. T. Pitchaiah**, Professor & HoD VFSTR for his constructive criticism throughout my internship.

We would like to specially thank, Prof. **N. Usha Rani**, Dean, School of ECE for her help and support during the project work.

We thank our internship coordinators **Mr. M. Vamshi Krishna, Mr. Abhishek Kumar** for continuous support and suggestions in scheduling the reviews and report verifications.

We would like to thank all the people that worked along with me in **Go Green Technologies Pvt. Ltd** with their patience and openness they created an enjoyable working environment. It is indeed with a great sense of pleasure and immense sense of gratitude that I acknowledge the help of these individuals.

We wish to express our gratitude to Prof**. P. Nagabhushan** Vice-Chancellor, VFSTR (Deemed to be University) for providing us the greatest opportunity to have a great exposure and to carry out the project.

We would like to thank **Dr. Lavu Rathaiah**, chairman VFSTR for his support and advices to get and complete internship in above said organization. We are extremely grateful to my department staff members and friends who helped me in successful completion of this internship.

Name of the candidate

\ P. Rajesh [201FA05047]

INTERNSHIP SUMMARY

Location: Velanki Center: GO-GREEN Technologies Pvt Ltd.

Duration: 4 months

Date of start: 15th January, 2024

Date of submission: 15th May, 2024

Training details: Technical Training on IOT And Embedded Systems.

Project Details: 1. Hydroacoustic Harmony.
2. Concentrated Solar Power.
3. Automatic Water Management System.

Manager: Dr. Y. Srinivasa Rao, GO-GREEN Technologies Pvt Ltd. Mumbai.

Name of Faculty guide: Dr. P. J. Reginald, Professor, VFSTR University.

PROFILE OF THE COMPANY

About Go Green Technologies Pvt Ltd

Go Green Technologies Pvt Ltd, formerly known as Emtron Technologies, is a venture founded by an alumnus of IIT-Bombay and is recognized as a startup by the Government of India. The company is currently incubated at the Sardar Patel Technology Incubation Center (SP-TBI) in Mumbai. With over 20 years in the field, Go Green Technologies specializes in developing customizable solutions for Cyber-Physical Systems and Digital Power Electronics applications.

Company Overview

Go Green Technologies focuses on innovative technology solutions in areas such as:

- Cyber-Physical Systems
- Digital Power Electronics
- Customizable technology solutions for various applications

Key Achievements and Innovations

- Over 20 years of experience in technology development
- Numerous patents filed in the fields of battery technology and power electronics
- Recognized as a significant player in the electric vehicle (EV) sector through its subsidiary Go GreenBOV, which has developed advanced lithium-ion battery technology and innovative mobility solutions.

Services Offered

- Development of embedded systems and real-time operating systems
- Consulting and corporate training in technology and engineering disciplines
- Research and development in power electronics and embedded systems

Recent Developments

In recent years, Go Green Technologies has focused on the electric vehicle sector, particularly through its work with Go GreenBOV, which specializes in advanced battery technology and electric two-wheelers. The company has filed 22 patents and aims to transform mobility with a focus on sustainability and innovative technology.

Major Design (Final Year Project Work) Experience Information

Student Group	P. Rajesh [20IFA05047]		
Project Title	Musical Fountain System	Concentrated Solar Power Tracking System	Water Management System
Program Concentration Area	Using IoT and Embedded Systems	Power System for Energy Generation and IOT	Using IoT and Embedded Systems
Constraints - Examples			
Economic	Fixed budget constraints for components and Control System		
Environmental	Friendly And Low Water Consumption, Use of Energy-Efficient Components		
Sustainability	Energy-efficient and Durable components to ensure long-term sustainability and reliability		
Manufacturability	Easy assembly and maintenance		
Ethical	Followed the standard professional ethics and ensuring public safety		
Health and Safety	Compliance with safety standards to prevent hazards		
Social	Providing an aesthetically pleasing, enjoyable public installation, sustainable energy solutions to communities and providing efficient water management solutions to communities		
Political	Adherence to local regulations regarding public installations		
Other	Reliable real-time monitoring and control of water resources, Efficient conversion of solar energy to electrical energy		
Standards			
1.	IEEE standards for IoT devices, Local safety standards for public installations		
2.	IEEE standards for IoT devices, ISO standards for water management systems		
3.	IEC standards for solar energy systems, Local standards for renewable energy		
Previous Course Required for the Major Design Experience	1. Microcontrollers and Embedded Systems, 2. Signal Processing, 3. Internet of Things (IoT) Fundamentals 4. Real-Time Systems 5. Renewable Energy Systems.		

Supervisor

Project Co-Ordinator

Head of the Department

PROJECT-1
1. LIST OF CONTENTS

Chapter 1 **PAGE NO**

Chapter 4

Chapter 5

Chapter 6

TABLE OF FIGURES

LIST OF TABLES

TABLE NO	NAME OF TABLE	PAGE NO

PROJECT-2
2. LIST OF CONTENTS

2.TABLE OF FIGURES

.

2.LIST OF TABLES

PROJECT -1

ABSTRACT

The design, implementation, and assessment of a Musical Fountain system using a controller and a sound sensor are presented in this project report (KY-038). This project aims to construct an interactive, aesthetically pleasing fountain that can be programmed to coordinate its water display with sound sensor Music inputs. Hardware components such as the RGB LED lights, water pump, microcontroller, sound sensor (KY-038), and water pump are integrated as part of the technique. The microcontroller processes the sound sensor's detected variations in sound strength. The water pump and RGB LEDs are controlled by the microcontroller to produce synchronized light effects and dynamic water patterns based on the input signals. One of the project's main accomplishments is the effective creation of a Musical Fountain system that can react to sound cues stimulation in the moment. When it comes to identifying sound signals and producing matching water and light displays, the system performs dependably. The project also emphasizes the possibility for additional improvements in terms of interactivity, design flexibility, and user experience. All things considered, this project demonstrates the viability and imaginative potential of fusing electronics, mechanics, and aesthetics to produce an interesting multimedia artwork. The Musical Fountain system showcased here provides customers with an immersive and enjoyable experience that can be used in a variety of indoor and outdoor situations.

CHAPTER 1

INTRODUCTION

1.1 History

Historical Beginnings (5000BCE–500CE):

The idea of water features separately developed throughout this ancient era in Mesopotamia, Egypt, and Greece, among other civilizations. Hydraulic engineering and water manipulation can be traced back to ancient engineers and builders, albeit specific inventors may not have contributed to these early advancements. For instance, the development of irrigation systems and aqueducts in Mesopotamia showed an early mastery of water control, while the building of temple complexes in Egypt featuring ceremonial pools and fountains highlighted the incorporation of water into religious symbolism and rites. Similarly, the Greeks represented the cultural significance of water in daily life and social gatherings through innovative architectural designs, such as the use of water in public baths and artistic fountains.

Figure 1.1.1: Mesopotamia Fountain

Europe in the Middle Ages and Renaissance (500–1600 CE):

European societies had a resurgence of interest in ancient art and literature during the Middle Ages and Renaissance, which sparked a rebirth of inventiveness and ingenuity in fountain design. Even while many fountains may not have been invented by individual inventors, architects, engineers, and artists worked together to produce some of the most famous water features of the era. In Italy, for example, the Renaissance gardens of the Villa d'Este in Tivoli demonstrated the skills of hydraulic engineers like Claude Venard and architects like Pirro Ligorio, who created the first water features like the organ fountain that was propelled by water. Together, their technical know-how and creative vision yielded enchanted settings that pleased the senses and blended in harmony with the natural world.

Figure 1.1.2: Renaissance Fountain

The CE Baroque and Rococo Eras (1600s–1800s):

Monarchs and other nobles who commissioned opulent fountains as emblems of their riches and authority defined the Baroque and Rococo periods. Although fountains may not have been invented by the same people, master artisans worked together to produce amazing mechanical and artistic creations. For instance, the 18th-century Trevi Fountain in Rome, created by architect Nicola Salvi, is a magnificent example of Baroque sculpture and hydraulic engineering. It has allegorical sculptures that honour the majesty and richness of the sea, as well as a massive water fall. These fountains demonstrated the artistic talent and cultural sophistication of the time and acted as focal areas for public events and celebrations.

Figure 1.1.3: Baroque and Rococo Fountain

The Modern Era and the Industrial Revolution (1800–present):

The invention of new hydraulic and mechanical systems by engineers and inventors throughout the Industrial Revolution led to revolutionary developments in the field of fountain technology. Henri Giffard was one such inventor who, in 1852, patented the centrifugal pump, which revolutionized water distribution and made it possible to build bigger, more intricate fountains. Technological developments in the 20th century led to the creation of contemporary musical fountains, which offer immersive sensory experiences through synchronized water, light, and sound elements. Famous examples of the cooperative efforts of designers, engineers, and artists who push the limits of originality and innovation in fountain design are the WET Design-created Bellagio Fountains in Las Vegas.

Figure 1.1.4: Modern Fountains

1.2 The Project Objectives

This project's main goal is to design, construct, and assess a musical fountain system with a controller and a sound sensor (KY-038). The project seeks to accomplish the following goals:

Create a dependable and flexible hardware system that can recognize sound signals and regulate the flow of lighting and water jets.
Create software algorithms that will process the sound sensor's audio input, examine sound patterns, and produce commands that will operate the fountain.
Provide an intuitive user interface that enables users to choose music tracks, modify fountain parameters, and start playback for the musical fountain system.
Construct a synchronized fountain display that is visually appealing and reacts dynamically to various kinds of sound or music.
Create a dependable and flexible hardware system that can recognize sound signals and regulate the flow of lighting and water jets.
Provide computer programs that process auditory input from the sound demonstrate how contemporary electronics and control systems can be used to make creative, interactive musical fountain installations that captivate and delight spectators.

1.3 Extent and Importance

This project's scope includes designing, building, and testing a freestanding musical fountain system that combines a controller and a sound sensor (KY-038). Because of the noises it detects, the system will be able to synchronize water and light displays, analyse audio patterns, and recognize sound signals in real time. The project's main goal is to create a working prototype of the musical fountain system. This will entail choosing and integrating hardware, creating software algorithms for audio processing and fountain control, testing, and validating the system's functionality. This project is important because it can demonstrate the technical and artistic possibilities of fusing electronics, mechanics, and design to produce immersive multimedia exhibits. By investigating the creation and application of a musical fountain system, the project hopes to stimulate curiosity and inventiveness in the field of interactive fountain design while offering insightful knowledge on the fusion of sensor technologies, microcontroller programming, and the fundamentals of human-computer interaction. By providing a fresh and captivating experience that can improve the atmosphere of public places, events, and recreational areas and promote social interaction, cultural enrichment, and community engagement, the project also aims to advance public art and entertainment technology.

CHATER 2

MUSICAL FOUNTAIN TECHNOLOGY

1. Technologies Used for Musical Fountain System to Analize Sound

1. Digital Signal Processing (DSP) Technology:

DSP processes signals by means of digital computation. DSP is utilized in musical compositions for the purpose of converting, analysing, and manipulating audio signals.
Signal Conversion:
Continuous analog signals are used to record audio signals. With the use of an analog-to-digital converter (ADC), these are transformed into digital format. This procedure entails quantizing the amplitude values to digital numbers and sampling the analog signal at discrete intervals.
Signal processing:
After the audio data is converted to digital format, it is handled by algorithms that can filter, compress, and examine the signal. DSP systems employ mathematical operations including filtering, modulation, and Fourier transforms to alter the signal.
Features including amplitude (loudness), frequency (pitch), and temporal elements are extracted by the DSP system.

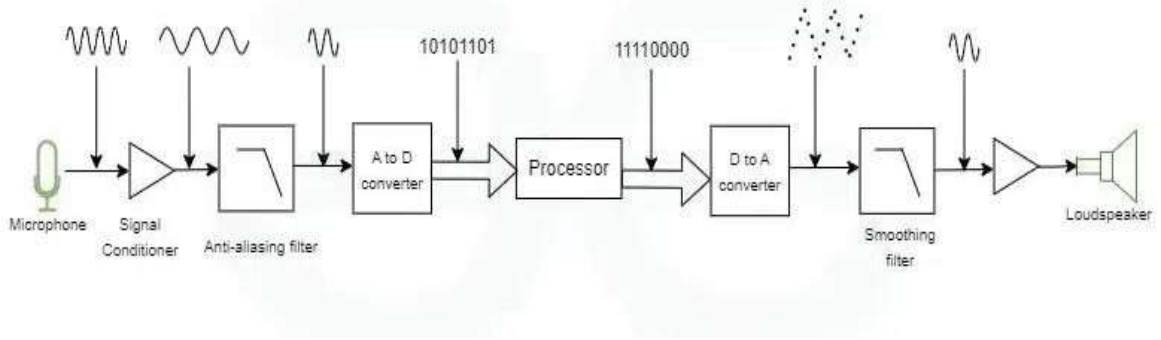


Figure 1.2.1: DSP Block Diagram

1.2 The Fast Fourier Transform (FFT) Technology:

The is a technique for quickly computing the Discrete Fourier Transform (DFT). It is frequently employed to convert a signal from the time domain to its frequency components.
The DFT calculation directly entails O (N − 2) O (N 2) calculations, where N is the number of samples. For large datasets, it becomes computationally costly because to its quadratic complexity. This complexity is reduced to O (N log N) using the FFT technique, allowing for the efficient processing of even massive signals. For real-time applications, such as musical fountains, where quick analysis and audio signal response are needed, this efficiency is essential.
Conversion Time to Frequency Domain:
The digital audio signal is sampled, and the FFT algorithm breaks it down into its individual frequencies. This offers a signal spectrum that displays the amplitude of various frequency components. Discrete frequency bins, each of which represents a distinct range of frequencies, are used to split the output. This aids in the analysis of the frequencies that are prominent at any moment.

2.1.3 Pattern Recognition:

Pattern recognition algorithms are essential because they analyse audio inputs to identify and categorize different musical patterns, including beats, rhythms, and structures.
The operation and uses of these algorithms:
Beat Identification:
Algorithms for beat detection are used to locate beats or components resembling pulses in an audio stream. To find the peaks that match the beats, this procedure usually entails examining the signal's amplitude envelope. These peaks show instances of the music's highest intensity or vigor and point to rhythmic accents. Beat detection algorithms often use threshold-based approaches, autocorrelation, or onset detection to precisely identify these peaks.
Rhythm Analysis:
This is more than just a basic beat detection to recognize the music's more intricate rhythmic structures and patterns. This entails examining the relationships between beats in terms of time, spotting patterns that keep coming up, and classifying them according to the qualities of rhythm. Rhythm analysis algorithms can be designed to capture hierarchical structures and temporal dependencies in the music through methods like recurrent neural networks, dynamic programming, and hidden Markov models
Feature matching:
To detect musical structures, feature matching compares the audio signal's extracted features with preset templates or patterns. These elements could consist of beat length, pace, accentuation, and spectral content, among other things. Finding the closest match between the observed characteristics and the templates is the goal of feature matching algorithms, which employ similarity metrics or pattern recognition techniques. To enable more flexible and adaptable pattern identification, machine learning models can also be trained to identify patterns straight from the data.

2.1.4 Real-Time Processing Systems:

Facilitating Immediate Reaction Systems for real-time processing are essential for applications like musical fountains that need to each instantly to data input.

Low Latency Operation:

The goal of low-latency processing is to reduce the time lag between the input and output of data. Hardware and software combinations that are optimized are needed to do this.

Effective Coding Techniques:

The way algorithms are implemented minimizes processing time and lowers computational overhead. This could entail the use of algorithmic optimizations, parallel processing strategies and optimized data structures.

High-Speed Processors:

Strong central processing units (CPUs) or specialized processing units that can carry out calculations quickly are used. Real-time processing tasks are given priority by these processors.

Real Time operating systems (RTOS):

RTOSs are made to manage tasks with precise timing specifications. By prioritizing jobs according to their deadlines, they guarantee uninterrupted execution of crucial processes. This keeps time-sensitive processes from being delayed by non-essential tasks.

2.1.5 AI and Machine Learning for Musical Fountains:

The fields of machine learning (ML) and artificial intelligence (AI) have revolutionized the fields of audio signal analysis and visual effect synchronization in musical fountains.

Supervised Learning:

Labelled music datasets are used to train machine learning models, with each data point being linked to a known characteristic or pattern. It is possible to classify beats, rhythms, and musical patterns, for instance. Through internal parameter adjustments aimed at reducing prediction errors, the model gains the ability to identify these patterns.

Unsupervised Learning:

In this method, patterns in the data are found by the model without the need for explicit labels. Using this method, the system can independently find hidden correlations or structures in the audio stream.

Predictive Analysis:

Machine learning models, once trained, are able to accurately anticipate the rhythmic and structural characteristics of novel audio recordings. For instance, the model can forecast beat timing, music tempo, and the existence of particular musical features like chord changes or drum fills. Predictive analysis makes it possible to instantly modify the visual effects to the musical elements, guaranteeing exact timing and amplifying the fountain performance's overall impact.

Adaptive Systems:

AI-powered systems can adjust to various musical genres, picking up the best visual effect timing for rock, electronic, classical, and other genres via experience. Adaptive systems make sure that the fountain's visual display is maintained by continuously evaluating auditory signals and modifying their behaviour based on the detected patterns.

2.2Working Principle of a Musical Fountain:

I. Water Nozzles and Pumps: Water is sent to the nozzles via a system of pipelines by high-pressure pumps. Water is released through these openings, known as nozzles. They sculpt the streams of water into different shapes and levels.

II. Controller Computerized: The water pumps and nozzles are timed and operated by a central computer system.

III. Program: The waterjets movements are synchronized with the music with the use of special software programs.

IV. Music analysis: The beats, rhythms, and important aspects of the song are identified. Programming: Certain nozzles and pumps are set to turn on in time with the music via the control system.

V. Integration of Control: The To match light changes with music and water movements, the lighting system relates to the control system.

CHAPTER 3

3.CONTROLLER AND SOUND SENSOR

3.1.1 INTRODUCTION OF A SOUND SENSOR MODULE

Microphone:
Electret microphones:
Which have an electret material and a diaphragm, are frequently used for general sound detection.

MEMS Microphone:
Compact and dependable sound detection in contemporary electronics.

Dynamic Microphone:
For capturing high-quality sound in professional audio applications.

Amplifier:
Amplifier Operational Amplifier (Op-Amp):
Enhances the feeble electrical signal that the microphone produces to a level that the microcontroller can process.

Modifiable Gain:
A potentiometer is a feature of certain modules that lets you modify the amplifier's gain and thus the sensor's sensitivity.

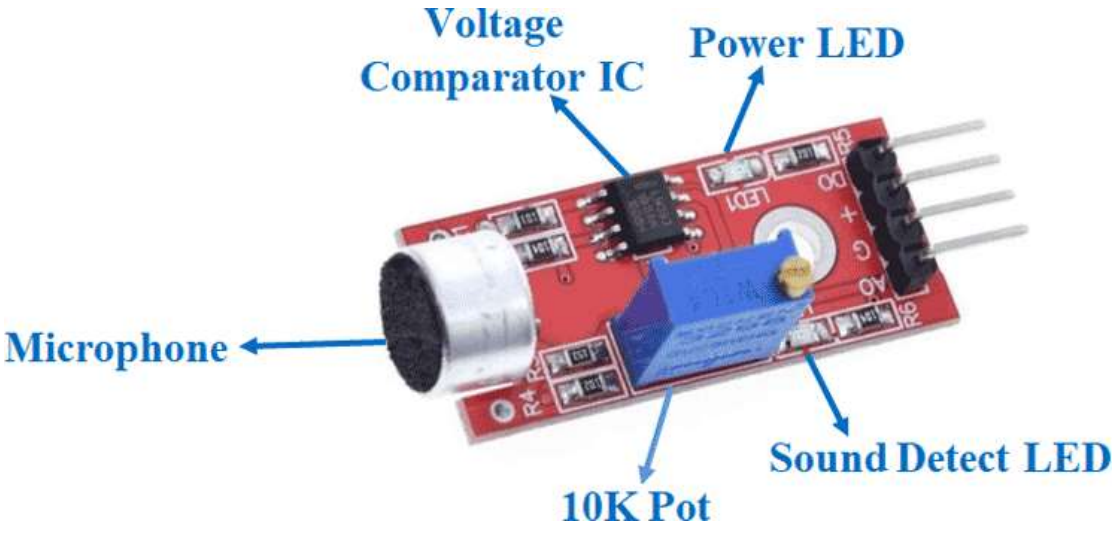


Figure 1.3.1: KY-038 Sound Sensor Module

Circuit for Signal Processing:
Envelope detector:
It measures the loudness of a sound by converting its AC audio stream into a DC signal.

Comparator:
Frequently used to give a digital output (high or low) depending on a user-specified threshold.

Pins for Output:
Analog Output (A0):
Used for accurate sound level detection, this signal is proportionate to the sound level. For straightforward sound detection applications.
Digital output (D0):
Provides a binary signal that indicates whether the sound intensity surpasses a predetermined threshold.

Energy Source:
VCC:
Attaches to the power source, usually at 3.3V or 5V.

GND:
Attaches to the circuit's ground.

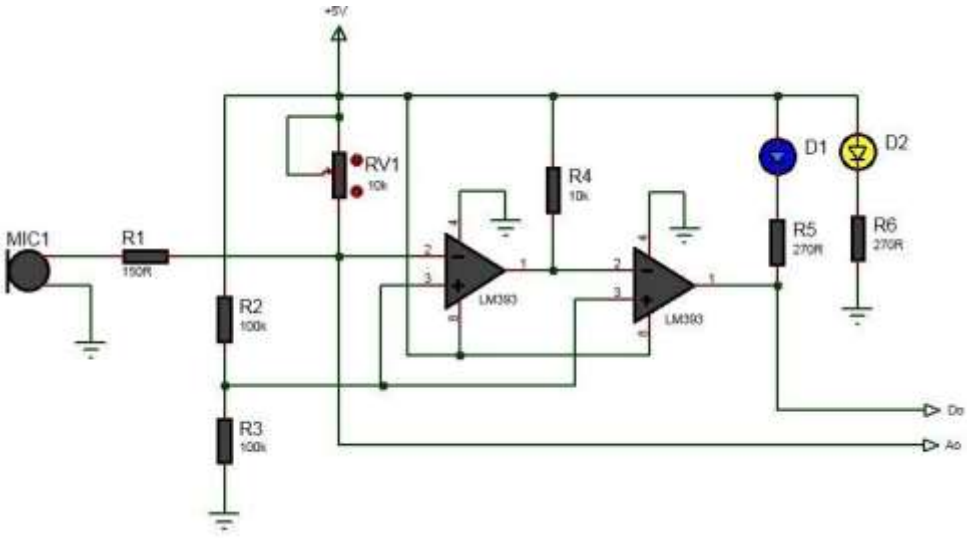


Figure 1.3.2: Circuit Diagram of KY-038 Sound Sensor Module

2. INTRODUCTION OF ESP32 MICROCONTROLLER

The ESP32 microcontroller has several parts and functions, including:

XtensaLX6Dual-CoreProcessor:

- High Performance:
  Capable of operating at up to 240 MHz, allowing complicated jobs to be processed quickly.

- Dual-Core:
  Facilitates multitasking by enabling the simultaneous execution of several tasks by two cores.

- Recall:
  RAM 520 KB of SRAM for effective processing and storing of data.

- Flash Memory:
  4 MB of flash memory is usually included for storing data and code.

Interaction:

- Wi-Fi:
  Allows wireless connectivity via a local network or the internet; supports 802.11 b/g/n.

- Bluetooth:
  For short-range wireless connection, Bluetooth 4.2 and BLE (Bluetooth Low Energy) are included.

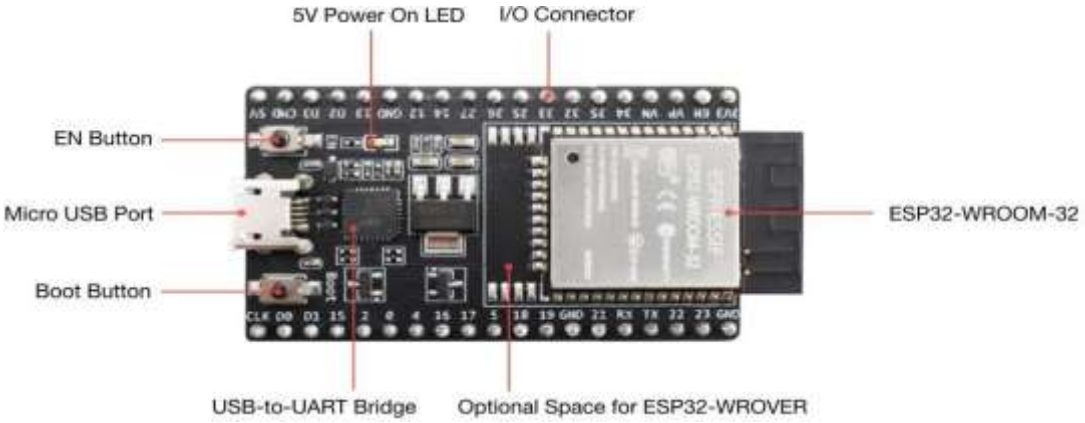


Figure 1.3.3: ESP 32 MICRO CONTROLLER

Interface Ports:

- Analog to Digital Converter (ADC):
  Multiple ADC channels are available for reading analog signals from sensors, such as the sound sensor, using an ADC (Analog to Digital Converter). A digital to analog converter, or DAC, has two channels that can be used to produce analog signals.
- Serial Peripheral Interface (SPI):
  For high-speed connection with peripherals like displays and memory devices, use SPI (Serial Peripheral Interface).
- Inter-Integrated Circuit (I2C):
  This interface is used to communicate with low-speed peripherals like sensors.
- Universal Asynchronous Receiver and Transmitter (UART):
  A serial communication tool used to communicate with other microcontrollers and devices is the UART.
- Controlling Power:
  Multiple low power modes, including as light and deep sleep modes, are supported for battery-operated applications.
- Voltage Regulation:
  To control the voltage levels for various components, on-board regulators are used.

Extra Elements:

- Timers:
  For accurate timing operations, use multiple hardware timers.
- Pulse Width Modulation (PWM):
  Used to regulate LED brightness, speed of engines, etc. Capacitive touch sensors are used in touch-based user interfaces.
- Real-Time Clock (RTC):
  Records time for applications that rely on it.

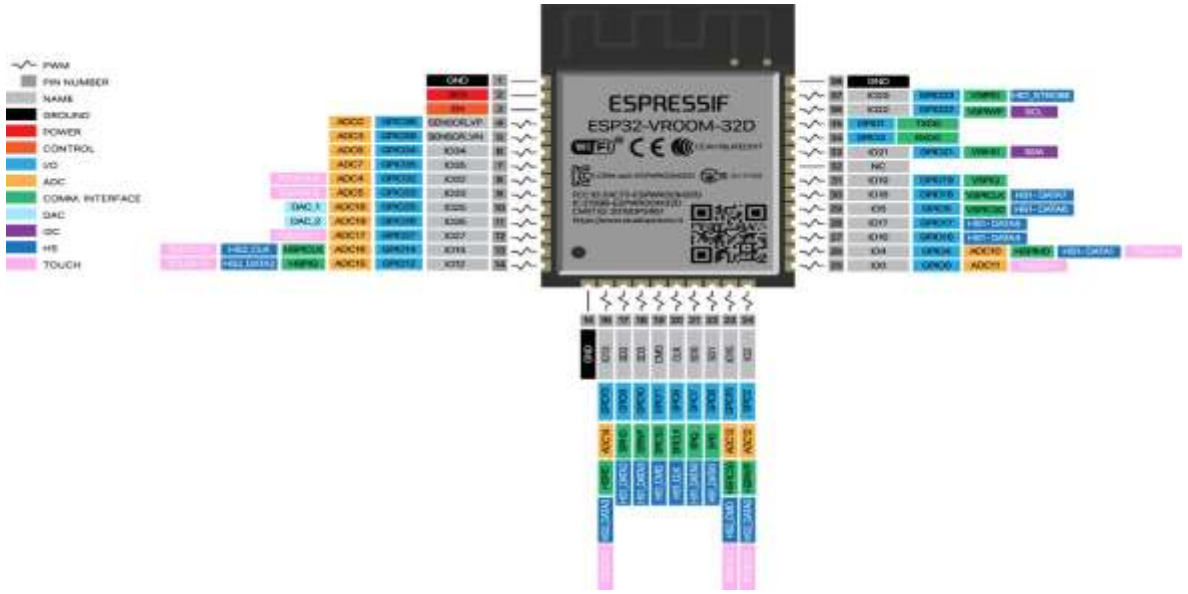


Figure 1.3.4: Pinout Diagram of ESP32 Controller

2.FEATURES AND SPECIFICATIONS OF MICROCONTROLLER AND SOUND SENSOR:

1. FEATURES AND SPECIFICATIONS OF MICROCONTROLLER

Specifications and Features	Details
Processor	Dual-Core Xtensa LX6
Processor Speed	Up to 240 MHz
SRAM	520 KB
Flash Memory	4 MB (typically included in dev kit)
Wi-Fi	802.11 b/g/n
Bluetooth	v4.2 BR/EDR and BLE
GPIO Pins	34 (varies by module)
ADC Channels	18 (12-bit resolution)
DAC Channels	2 (8-bit resolution)
SPI Interfaces	4
I2C Interfaces	2
UART Interfaces	3
PWM Channels	16
Touch Sensors	10 Capacitive touch sensors
RTC	YES
Operating Voltage	2.2 -3.6V
Low Power Modes	Deep Sleep, Light Sleep
Voltage Regulation	YES
Timers	Multiple hardware timers
Operating Frequency	2.4 GHz for Wi-Fi

Table 1.3.1: ESP32 Micro Controller Features and Specifications

3.2.2 KY-038 SOUND SENSOR FEATURES AND SPECIFICATIONS:

Feature/Specification	Details
Microphone Type	Electret Microphone
Amplification	Operational Amplifier
Adjustable Gain	Via onboard potentiometer
Signal Processing	Envelope Detector
Comparator	Yes
Output Types	Analog Output (A0), Digital Output (D0)
Operating Voltage	3.3V to 5V
Frequency Range	20 Hz to 20 kHz
Dimensions	Small and compact

Table 1.3.2: KY-038 Sound Sensor Features and Specifications

3. WORKING PRINCIPLE OF CONTROLLER AND SOUND SENSOR:

The KY-038 detects sound waves using an electret microphone. A little electrical signal is produced when sound waves strike the microphone's diaphragm, which vibrates and alters the electret material's capacitance.

- Amplification of Signal:
  An operational amplifier boosts the little electrical signal coming from the microphone (Op-Amp). A potentiometer on the module allows the sensitivity of the sensor to be customized by adjusting the gain of the amplifier.

- Detecting Envelopes:
  After being amplified, the signal is passed through an envelope detector, which changes the AC audio signal into a DC signal that indicates the sound's amplitude, or loudness.

- Comparator for Electronic Output:
  A comparator circuit measures the sound signal's amplitude in relation to a threshold that is adjusted by a different potentiometer. The comparator emits a high digital signal when the sound level is higher than this threshold and a low digital signal when it is not.

- Results:
  A continuous analog signal proportionate to the observed sound intensity is the analog output (A0).
  Digital Output (D0): A binary signal that indicates if the volume of the sound is higher than a predetermined threshold.

- The ESP32 Microcontroller's:
  One of the ADC (Analog to Digital Converter) pins on the ESP32 is used to read the analog signal from the KY-038 sound sensor.

- Signal Handling:
  To calculate the sound level, the analog signal is processed by the ESP32. To determine whether the sound level surpasses a predetermined threshold, it may also read the digital output from theKY-038.
- Interaction:
  For additional research and monitoring, the ESP32 may transmit sound level data to other devices.
- Controlling Power:
  When not actively processing or transferring data, the ESP32 can switch to low-power modes, extending the battery life in portable applications.

CHAPTER 4

1. SYSTEM ARCHITECTURE

- Sound Detection: The sound sensor captures audio input (music).
- Signal Processing: The ESP32 processes the audio signal to determine intensity and frequency.
- Control Signals: The ESP32 sends control signals to the relays.
- Activation: Relays activate pumps, valves, and lights based on the processed sound signals.
- Synchronization: Water and light effects are synchronized with the audio input.
- Programming Environment: Arduino IDE or ESP-IDF.
- Inputs: Sound sensor input (analog or digital), optional environmental sensors.
- Outputs: Control signals to relays for pumps and lights, PWM signals for LED control.
  Sound Sensor.
- Connection: Analog input pin of the ESP32.
- Signal Processing: ADC to read sound level, apply filters to smooth the signal, and reduce noise.
- Thresholds: Set in ESP32 code to trigger actions based on sound intensity.

2. HARDWARE COMPONENTS:

1. ESP32 MICROCONTROLLER

With built-in Bluetooth and Wi-Fi, the ESP32 is a potent microcontroller that is perfect for Internet of Things applications. Here is a quick rundown of its key elements and characteristics:
Essential Elements:

1. Processor:
    - Ten silica Dual Core Xtensa: The ESP32 has two 32-bit cores that can operate at a maximum frequency of 240MHz.
    - Ultra-Low-Power (ULP): This processor manages low-power tasks, freeing up the primary cores to enter deep sleep.
    - Recollection520 KB of SRAM is often included.
    - Flash Memory: Vary based on the module usually has a capacity of 4 MB to 16MB.
2. Wireless Networking:
    - Wi-Fi: 802.11 b/g/n standards, compatible with WPA and WPA2.
    - Bluetooth: BLE (Bluetooth Low Energy) and Bluetooth v4.2 BR/EDR.
3. Peripherals:
    - GPIO: Digital I/O can be configured on up to 36 GPIO pins, some of which can allow touch sensing.
    - ADC: Twelve-bit SAR ADCs in up to eighteen channels.
    - DAC: Two channels, each of 8 bits.
    - UART: Three UART interfaces maximum.
    - SPI: Four SPI interfaces maximum.
    - I2C: Two interfaces for I2C.
    - I2S: For input and output of digital audio.
    - PWM: 16 channels maximum.
    - Interface for the Controller Area Network (CAN).
    - Ethernet: Provides support for IEEE 1588 Precision Time Protocol and an Ethernet MAC interface with dedicated DMA.
4. Timer Devices:
    - Four 64-bit timers for general purposes.
    - Two watchdog timers are present.
    - Real-Time Clock (RTC): A clock with a separate power domain.

5. Security Features:
    - Hardware Accelerated Encryption: AES, SHA-2, RSA, and ECC.
    - Secure Boot: Ensures that only trusted firmware can be executed.
    - Flash Encryption: Protects the contents of the flash from being read or altered.
6. Controlling Power:
    - Deep Sleep Mode: Reduces power consumption to as low as 5 µA.
    - Light Sleep Mode: Reduces power consumption while retaining memory and peripherals.
    - Power Domains: Allows selective powering of peripherals.

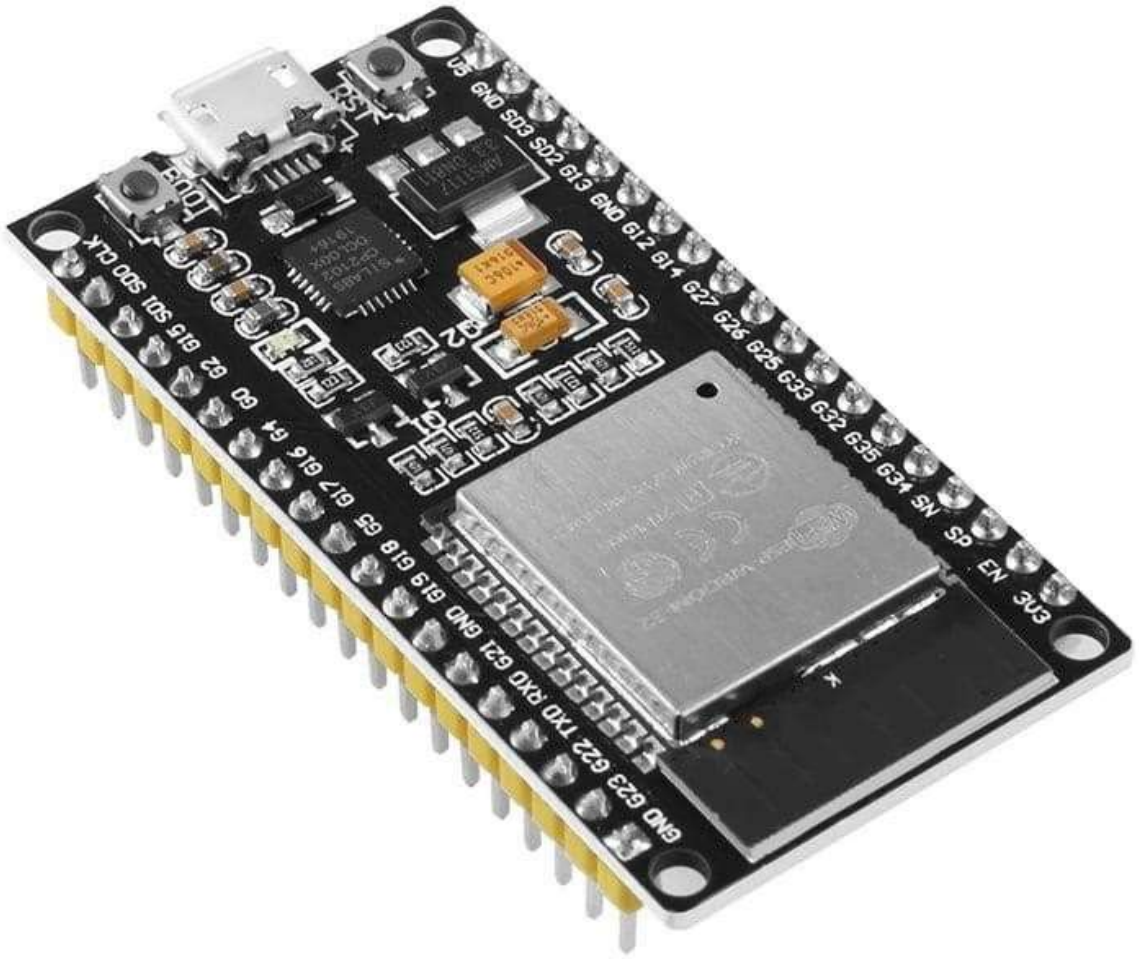


Figure 1.4.1: ESP32 Micro Controller

2. SOUND SENSOR

The KY-038 sound sensor is a popular and inexpensive module used in various electronics projects to detect sound levels in the environment.

Components:

1. Electret Microphone:
   - Captures sound from the environment and converts it into an electrical signal.
2. LM393 Comparator:
   - Compares the microphone signal to a preset threshold and provides a digital output indicating whether the sound level has crossed this threshold.
3. Potentiometer:
   - Allows for the adjustment of the sensitivity threshold, controlling the level at which the digital output switches.
4. Output Pins:
   - VCC: Power supply (typically 3.3V to 5V).
   - GND: Ground.
   - DO (Digital Output): Provides a high or low signal based on the comparator's output.
   - AO (Analog Output): Provides an analog signal proportional to the sound level.
5. Features:
   - Adjustable Sensitivity: The onboard potentiometer lets you adjust the sensitivity to detect different sound levels.
   - Digital and Analog Outputs: Offers both digital and analog outputs for versatile interfacing with microcontrollers.
   - Operating Voltage: Typically operates within a range of 3.3V to 5V, compatible with most microcontroller boards like Arduino, ESP32, and Raspberry Pi.
   - Simple Interface: Easy to connect and use with standard digital I/O pins.

Figure 1.4.2: KY-038 Sound Sensor.

3. RELAY MODULE

A relay module is an electrical component used to control a high-power circuit with a low-power signal. It acts as an electrically operated switch, allowing a small current to control a larger current

Components of a Relay Module

- Relay: The main component that performs the switching. It typically has a coil (electromagnet) and one or more sets of contacts (switches).
- Driver Circuit: Often includes a transistor or an optocoupler to amplify the low-power signal to a level sufficient to activate the relay coil.

Protection Components:

- Diodes (such as a flyback diode) to prevent voltage spikes when the relay coil is de-energized, and sometimes resistors and capacitors for additional protection and stability.

Types of Relays

- Electromechanical Relays (EMR): Use a physical switch mechanism and are typically used for switching high currents.
- Solid State Relays (SSR): Use semiconductor devices to perform switching without moving parts, offering faster switching times and greater reliability.

Operating Principle

- Control Signal: A low-power signal (from a microcontroller, sensor, etc.) is sent to the relay module.
- Activation: The driver circuit amplifies this signal and energizes the relay coil.
- Switching: The energized coil generates a magnetic field that moves the contacts, either opening or closing the high-power circuit.
- Deactivation: When the control signal is removed, the coil de-energizes, and the contacts return to their original position, switching the circuit back.

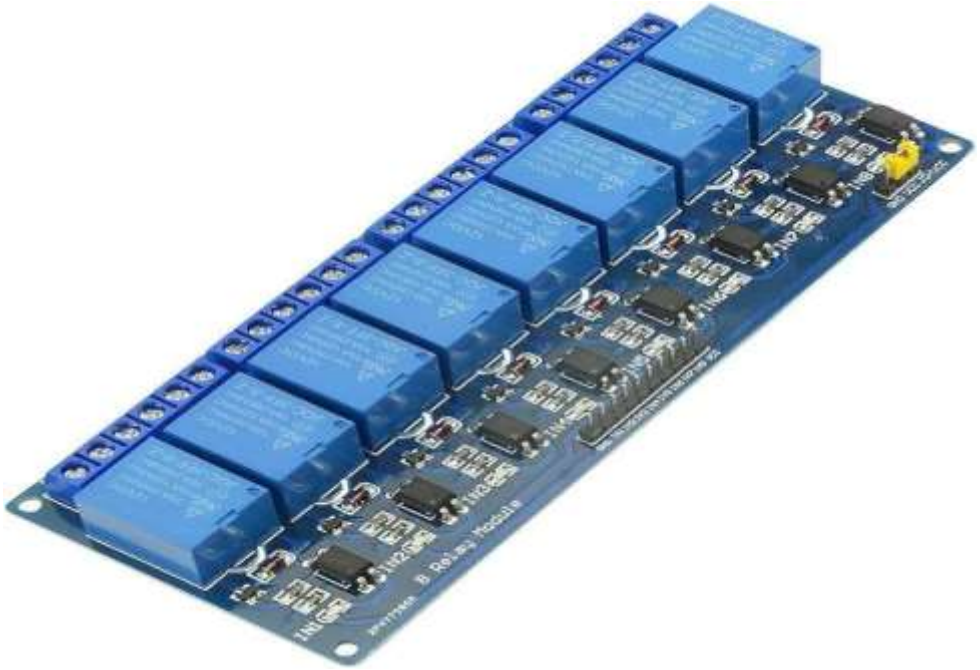


Figure 1.4.3: 8 Channel Relay Module.

4. JUMPER WIRES

Jumper wires are simple electrical wires with connector pins at each end, used for interconnecting components on a breadboard, test circuit, or other prototyping setups.
They are essential for making temporary connections and testing circuits without the need for soldering.

Key Features

1 Variety of Connectors:
   - Male-to-Male (M-M): Both ends have male pins, ideal for connecting two female headers or breadboard connections.
   - Male-to-Female (M-F): One end has a male pin, and the other end has a female socket, useful for connecting a male pin to a female header.
   - Female-to-Female (F-F): Both ends have female sockets, used to connect two male pins.
2 Color Coding:
   - Jumper wires come in various colors, making it easier to identify and trace connections in a complex circuit.
3 Lengths:
   - Available in different lengths to accommodate various spacing requirements in a circuit.
4 Insulation:
   - Typically made of flexible plastic or rubber insulation to prevent short circuits.

Figure 1.4.4: Jumper Wires

5. SOLENOID VALVE

A solenoid valve is an electromechanically operated valve used to control the flow of liquids or gases.
It consists of a solenoid (a coil of wire that acts as an electromagnet when an electric current passes through it) and a valve mechanism.

Key Components

- Solenoid Coil: An electromagnetic coil that generates a magnetic field when energized.
- Plunger: A movable iron core inside the solenoid that is pulled or pushed by the magnetic field.
- Valve Body: The housing that contains the valve mechanism and flow passages.
- Orifice: The opening that allows fluid or gas to pass through when the valve is open.
- Spring: Returns the plunger to its original position when the solenoid is de-energized.

Operating Principle

- Energizing the Solenoid: When an electric current is applied to the solenoid coil, it generates a magnetic field.
- Movement of Plunger: The magnetic field moves the plunger, which either opens or closes the valve orifice.
- Control of Flow: This movement controls the flow of fluid or gas through the valve.
- De-energizing the Solenoid: When the current is turned off, the magnetic field collapses, and the spring returns the plunger to its resting position, either closing or opening the valve.

Types of Solenoid Valves

- Direct-Acting Valves: The solenoid directly opens or closes the valve orifice without the need for additional pressure.
- Pilot-Operated Valves: Use the solenoid to control a small pilot valve, which in turn controls a larger valve mechanism, allowing them to handle higher pressures and flow rates.

Figure 1.4.5: 12v Normally Closed Solenoid Valve.

3. SOFTWARE TOOL

Arduino IDE

Here is a comprehensive guide that includes steps for using the Software Tool Arduino IDE.

1.Using the Arduino IDE:

- Download the Arduino IDE:
- Visit the (https://www.arduino.cc/en/software).
- Choose the version compatible with your operating system (Windows, macOS, or Linux).

2.Install the Arduino IDE:

- Windows:
  - Run the downloaded .exe file.
  - Follow the installation wizard, ensuring you install the drivers when prompted.
- macOS:
  - Open the downloaded .zip file.
  - Drag the Arduino application into the Applications folder.

  -Linux:
  - Extract the downloaded tar.xz file.
  - Open a terminal and navigate to the extracted folder.
  - Run ./install.sh script to install the IDE.

Figure 1.4.6: Downloading of Arduino IDE .

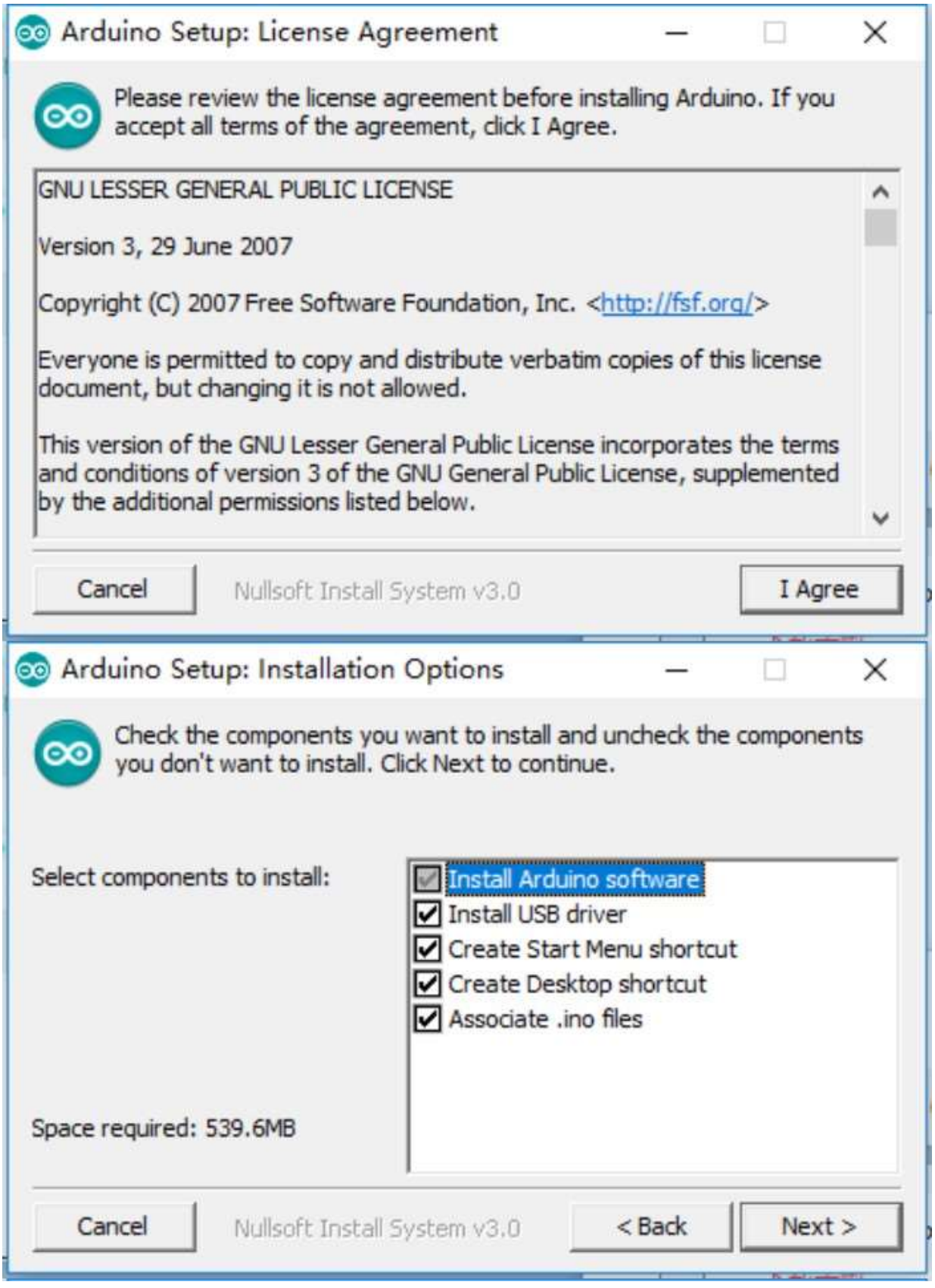


Figure 1.4.7: Arduino IDE Setup.

3. Connect Your Arduino Board

- Connect the Board:

  - Use a USB cable to connect your Arduino board to your computer. Most common Arduino boards use a Type-B USB cable, but some newer boards use Micro-USB or USB-C cables

- Open the Arduino IDE
- Launch the IDE:

  - Open the Arduino IDE application from your desktop or start menu.

Figure 1.4.8: Connecting the Arduino Board to Arduino Application.

4. Configure the IDE for Your Board

- Select the Board:

  - Navigate to Tools > Board.
  - Choose your specific Arduino board from the list (e.g., Arduino Uno, Arduino Nano).

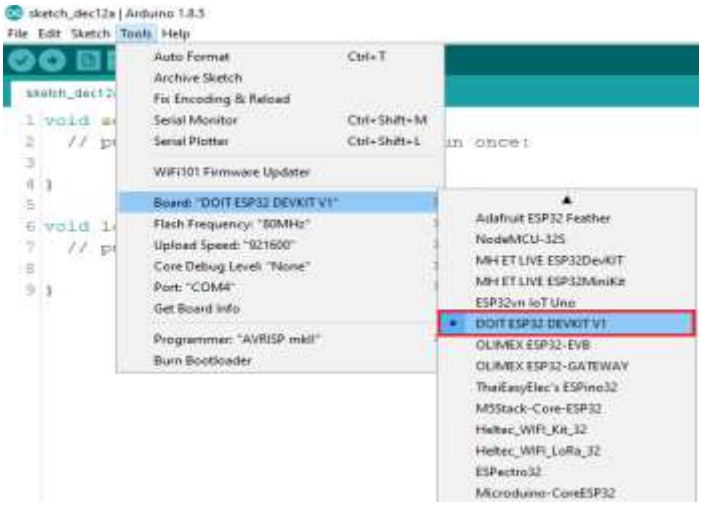


Figure 1.4.9: Selecting the Arduino Board in the Application.

5. Select the Port:

- Navigate to Tools > Port.

- Select the port that corresponds to your Arduino board. This is usually labeled COMx.

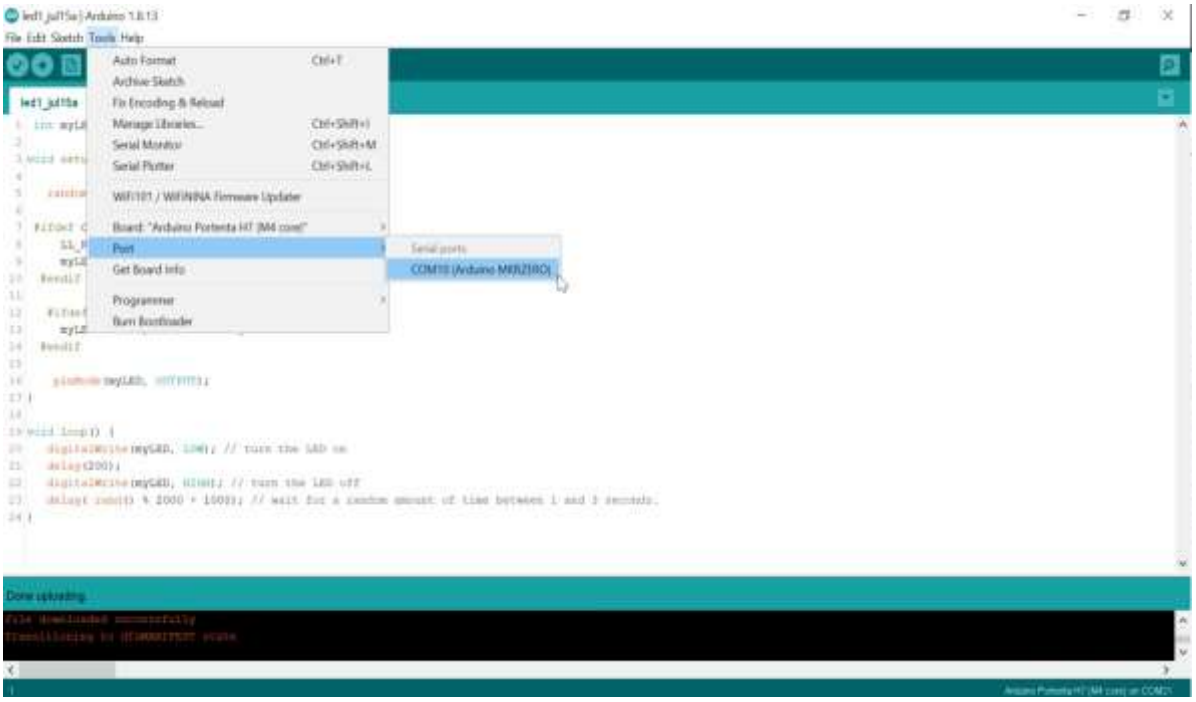


Figure 1.4.10: Selecting the Port in Arduino Application.

6. Write or Open a Sketch

- Creating a New Sketch:

- Go to File > New to create a new sketch.

- A new window with an empty sketch will open.

Figure 1.4.11: Selecting Board and Port for Code Compile.

7. Opening an Example Sketch:
   - Navigate to File > Examples.
   - Browse through the categories to find an example sketch. For instance, Basics > Blink.

8. Basic Structure:

```
void setup () {
  // Initialization code goes here
}

void loop () {
  // Main code goes here
}
```

   - setup (): Runs once when the board is powered on or reset.
   - loop (): Continuously runs after the setup () function completes.

9. Verify/Compile the Sketch
   - Check Syntax and Compile:
   - Click the checkmark icon in the top-left corner.
   - The IDE will compile the code and check for errors. Errors will be highlighted in the message area at the bottom of the IDE.

10. Upload the Sketch to the Board
   - Upload the Code:
   - Click the right arrow icon (next to the checkmark).
   - The IDE will compile the code if it has not been done already and upload it to the Arduino board.
   - The message area will confirm a successful upload.

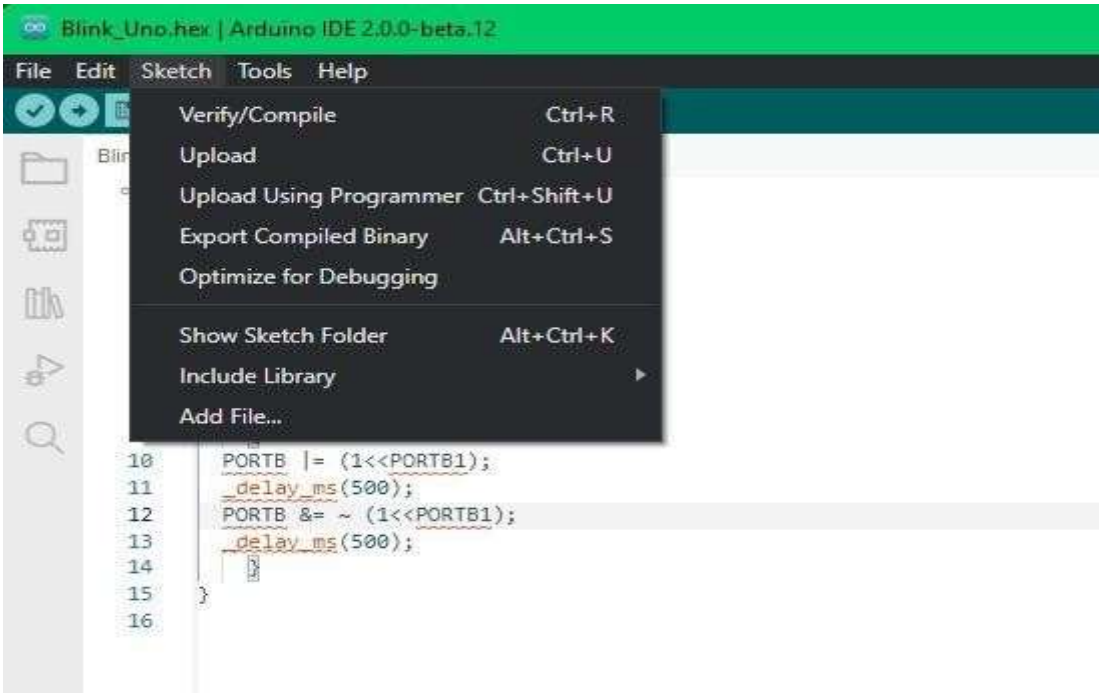


Figure 1.4.12: Steps to verify and Upload the code.

11. Use the Serial Monitor (if needed)
    - Open the Serial Monitor:
    - Click the magnifying glass icon in the top-right corner.
    - Alternatively, go to `Tools` > `Serial Monitor`.
    - The Serial Monitor allows you to send and receive data to and from the Arduino board, useful for debugging.
    - Serial Communication in Code:
    - Use Serial. begin (9600); in the setup () function to initialize the serial communication.
    - Use Serial. Print () or Serial. Println () in the loop () function to send data to the Serial Monitor.

12. Using Libraries:
    - Libraries extend the functionality of Arduino. To include a library, go to `Sketch` >Include Library` > `Manage Libraries`.
    -Search for and install libraries as needed.

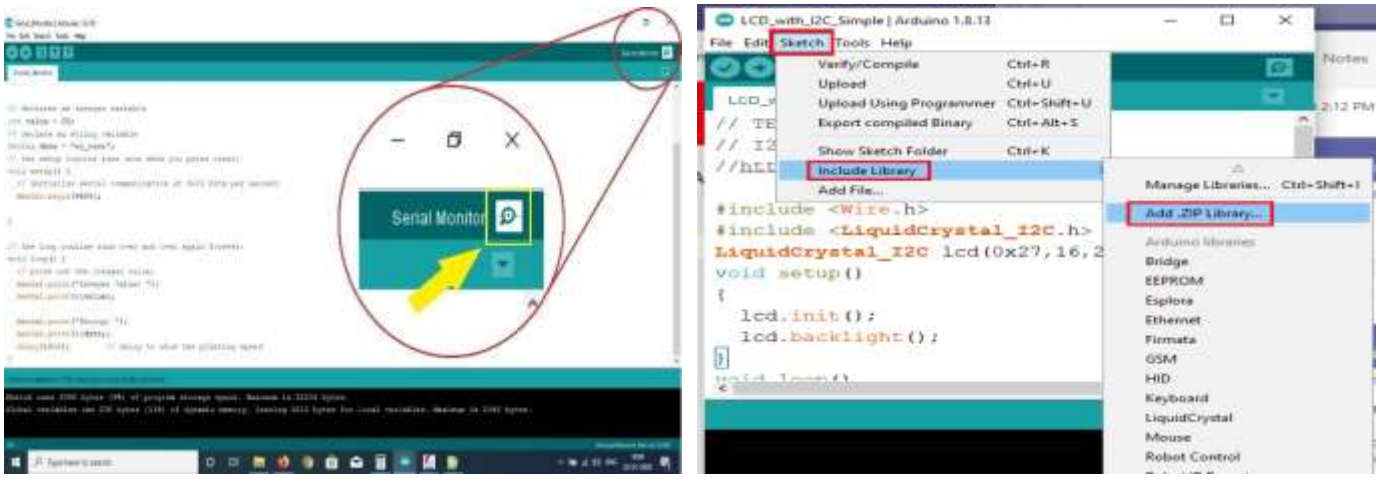


Figure 1.4.13: Steps to include libraries and Steps to Open Serial monitor

CHAPTER 5

1. Musical Fountain Testing and Calibration

A musical fountain must be tested and calibrated to make sure all its parts are working properly and that the music, lights, and water jets are all timed in unison to produce a visually appealing display.

First Inspection and Setup

1.Component Check:

- Examine every piece of hardware, such as the control system, lighting, sensors, pumps, valves, and nozzles.
- Make sure there are no leaks in the water lines and that all electrical connections are tight.

2.Water Supply:

- Add water to the reservoir until the suggested level is reached.
  Verify that the water level sensor is operating properly and look for any leaks.

3.Software Setup:

- Program the microcontroller or PLC using the control software. Verify that the control interface, or HMI is linked and configured correctly.

Examining Separate Elements

Synchronization:

Check that pumps, valves, and lights can all be controlled at the same time by running a short sequence. Modify schedules to guarantee that lights and water jets turn on and off simultaneously.

1.Music Integration:

- Incorporate a demo audio file into the control program. Map light and water jet patterns to the music using the software. Start with easy rhythms, then sync tiny water jets and light bursts to the beat.

2.Fine-tuning:

- Start the music and see the fountain act. Modify the water jet.

3.Pump Test:

- To make sure each pump is operating properly, turn it on separately.
- Verify that the water pressure and flow rate full fill the required standards.

4.Valve and Nozzle Test:

- Check to see if each solenoid valve is functioning properly by opening and closing it.
  Verify that each nozzle generates the appropriate water pattern by testing it.

5.Lighting Test:

- Examine each LED light separately and collectively.
  Verify the brightness and accuracy of the colour.

6.Sensor Test:

- Create low and highwater levels to test water level sensors.
  Adjust the flow rate to test flow sensors.

7. Adjusting for Synchronization:

- Basic and light timing to align with the music more closely. Make sure there are seamless transitions between patterns.

Comprehensive System:

1. Run Complete Sequences:
    - To verify if there are any delays or inconsistencies, run the entire musical fountain sequence.
- Verify that the lights, water jets, and music are all perfectly timed.

2. Verify Uniformity:
    - Execute the sequences several times to guarantee reliable performance.
      Seek out any differences or problems that require attention.

3. Environmental Conditions:
    - To guarantee dependable performance, test the fountain in a variety of environmental settings (such as temperature and wind).
    - Modify the control system as needed to account for these circumstances.

Modifications to Calibration:

1. Flow Rate and Pressure Calibration:
    - To obtain the appropriate water jet height and pattern, modify the pumps' flow rate and pressure.
      If required, apply pressure control valves.

2. Nozzle Alignment:
    - Adjust nozzle alignment to guarantee precise water jet direction. Modify locations and angles as necessary.

3. Light Calibration:
    - Adjust the LED lights' hue and intensity to create the desired visual impact. Assure consistent illumination and Get rid of any flickering. 4.Sound Calibration:
    - Verify that the speakers in the sound system are properly positioned and tuned to produce the highest possible audio quality.
    - Modify equalization and volume settings to achieve a clean, well-balanced sound.

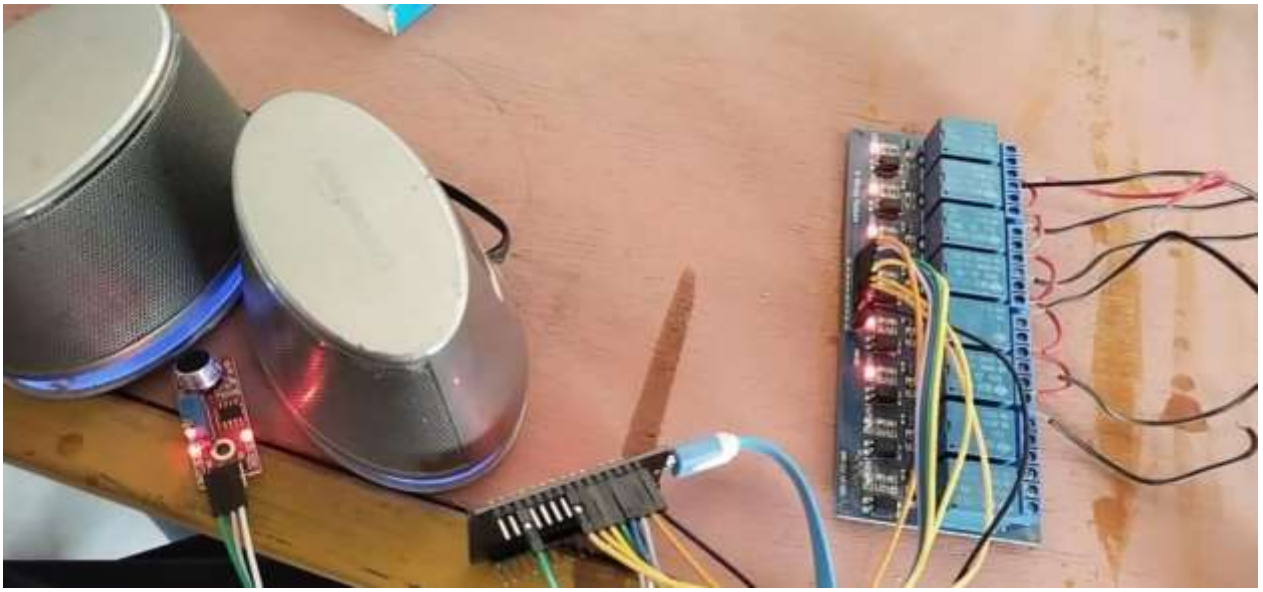
Complete Examination and Confirmation:

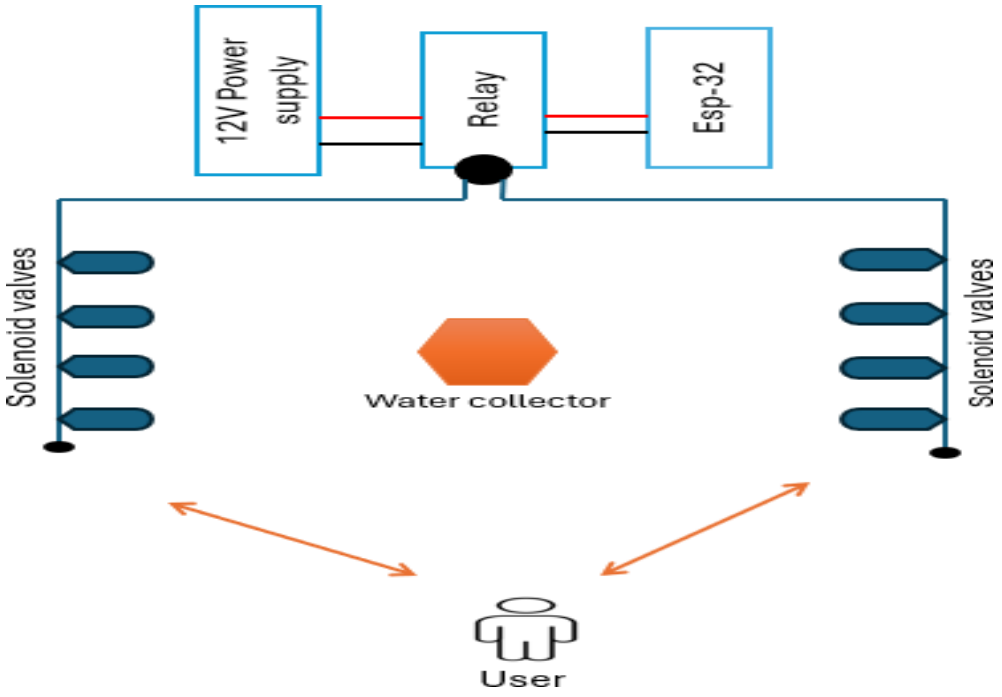
1. Stress Test:
    - Run the fountain constantly for a long time to look for any problems while it is operated continually.
    - Keep an eye on the lights' and pumps' functioning and temperature.

2. Safety Checks:
    - Verify that all safety features, such as the automatic shut-off for low water levels, are operating.
- Inspect for possible dangers or malfunction risks.

3. Documentation:
    - Record all test outcomes and calibration settings.

5.2 Circuit Design





CHAPTER 6

6.1 PROGRAM LOGIC:

```
const int sound Sensor Pin = 39 ;

const int led Pins [] = {13, 12, 14, 27, 26, 25, 33, 32};

const int num Led = size of (led Pins) / size of (led Pins [0]);

int sound Thresholds [] = {200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000, 2200, 2400, 2600,
2800, 3000, 3200, 3400, 3600, 3800, 4000, 4200, 4400, 4600, 4800, 5000};

int pattern1[] = {1, 1, 1, 1, 1, 1, 1, 1};
int pattern2[] = {1, 0, 1, 0, 1, 0, 1, 0};
int pattern3[] = {1, 1, 1, 1, 0, 0, 0, 0};
int pattern4[] = {1, 0, 0, 1, 1, 0, 0, 1};
int pattern5[] = {0, 1, 0, 1, 0, 1, 0, 1};
int pattern6[] = {0, 1, 1, 0, 1, 1, 0, 0};
int pattern7[] = {1, 1, 0, 0, 1, 1, 0, 0};
int pattern8[] = {1, 0, 1, 0, 1, 0, 1, 0};
int pattern9[] = {0, 0, 1, 1, 1, 1, 0, 0};
int pattern10[] = {1, 1, 0, 1, 0, 0, 1, 1};
int pattern11[] = {1, 0, 1, 0, 0, 1, 1, 1};
int pattern12[] = {0, 0, 1, 1, 1, 0, 0, 0};
int pattern13[] = {0, 1, 0, 0, 1, 0, 1, 0};
int pattern14[] = {1, 1, 0, 0, 0, 0, 1, 1};
int pattern15[] = {1, 1, 1, 0, 0, 1, 1, 1};
int pattern16[] = {1, 0, 1, 1, 0, 1, 0, 1};
int pattern17[] = {0, 1, 0, 1, 1, 0, 1, 0};
int pattern18[] = {1, 1, 1, 0, 1, 1, 1, 0};
int pattern19[] = {1, 0, 1, 0, 0, 0, 1, 1};
int pattern20[] = {0, 1, 1, 0, 1, 0, 1, 1};
int pattern21[] = {0, 1, 0, 1, 0, 1, 0, 0};
int pattern22[] = {1, 1, 0, 0, 1, 0, 0, 1};
int pattern23[] = {1, 0, 1, 0, 1, 0, 0, 1};
int pattern24[] = {1, 1, 1, 0, 0, 0, 0, 0};
int pattern25[] = {0, 0, 0, 0, 1, 1, 1, 1};

void setup() {
  Serial. Begin(115200);
  for (int i = 0; i < num Led; i++) {
    pin Mode(led Pins[i], OUTPUT);
  }
}

void loop() {
  int sound Value = analog Read(sound Sensor Pin);
  Serial.Println (sound Value);
  delay(1000);
```

```
  for (int i = 0; i < size of (sound Thresholds) / size of (sound Thresholds[0]); i++) {
    if (sound Value < sound Thresholds[i]) {
      activate Pattern(i);
      break;
    }
  }

  Delay (400);
}

void activate Pattern(int pattern Index) {
  switch (pattern Index) {
    case 0:
      apply Pattern(pattern1);
      break;

    case 1:
      apply Pattern(pattern2);
      break;

    case 2:
      apply Pattern(pattern3);
      break;

    case 3:
      apply Pattern(pattern4);
      break;

    case 4:
      apply Pattern(pattern5);
      break;

    case 5:
      apply Pattern(pattern6);
      break;

    case 6:
      apply Pattern(pattern7);
      break;

    case 7:
      apply Pattern(pattern8);
      break;

    case 8
      apply Pattern(pattern9);
      break;

    case 9:
      apply Pattern(pattern10);
```

```
  break;

case 10:
  apply Pattern(pattern11);
  break;

case 11:
  apply Pattern(pattern12);
  break;

case 12:
  apply Pattern(pattern13);
  break;

case 13:
  apply Pattern(pattern14);
  break;

case 14:
  apply Pattern(pattern15);
  break;

case 15:
  apply Pattern(pattern16);
  break;

case 16:
  apply Pattern(pattern17);
  break;

case 17:
  apply Pattern(pattern18);
  break;

case 18:
  apply Pattern(pattern19);
  break;

case 19:
  apply Pattern(pattern20);
  break;

case 20:
  apply Pattern(pattern21);
  break;

case 21:
  apply Pattern(pattern22);
  break;

case 22:
```

```
      apply Pattern(pattern23);
      break;

    case 23:
      apply Pattern (pattern24);
     break;

    case 24:
      apply Pattern(pattern25);
      break;

    default: // Default pattern: Random blinking
      for (int i = 0; i < num Led; i++) {
        digital Write (led Pins[i], random (2));
      }
      delay(100);
      break;
  }
}

void apply Pattern (int pattern[]) {
  for (int i = 0; i < num Led; i++) {
    digital Write (led Pins[i], pattern[i]);
  }
}
```

Pattern Index	Valve 1	Valve 2	Valve 3	Valve 4	Valve 5	Valve 6	Valve 7	Valve 8
0	1	1	1	1	1	1	1	1
1	1	0	1	0	1	0	1	0
2	1	1	1	1	0	0	0	0
3	1	0	0	1	1	0	0	1
4	0	1	0	1	0	1	0	1
5	0	1	1	0	1	1	0	0
6	1	1	0	0	1	1	0	0
7	1	0	1	0	1	0	1	0
8	0	0	1	1	1	1	0	0
9	1	1	0	1	0	0	1	1
10	1	0	1	0	0	1	1	1
11	0	0	1	1	1	0	0	0
12	0	1	0	0	1	0	1	0
13	1	1	0	0	0	0	1	1
14	1	1	1	0	0	1	1	1
15	1	0	1	1	0	1	0	1
16	0	1	0	1	1	0	1	0
17	1	1	1	0	1	1	1	0
18	1	0	1	0	0	0	1	1
19	0	1	1	0	1	0	1	1
20	0	1	0	1	0	1	0	0
21	1	1	0	0	1	0	0	1
22	1	0	1	0	1	0	0	1
23	1	1	1	0	0	0	0	0
24	0	0	0	0	1	1	1	1

Table 1.6.1: Lookup Table of Solenoid Valve Patterns

2. RESULTS AND DISCUSSION

1. Results:

1. Spectacle and Aesthetics
   - Visual Appeal: The musical fountain delivered an impressive visual spectacle. The synchronized water jets, colored lighting, and music created a captivating show that was well-received by the audience. Different patterns, heights, and directions of water jets, combined with a variety of colors, produced dynamic and engaging visual effects.
   - Audience Engagement: Observations and feedback indicated high levels of audience engagement and satisfaction. Spectators were observed taking photos and videos, indicating the fountain's success as a visual and social attraction.

2. Technical Performance
   - Synchronization Accuracy: The synchronization between the water jets and the music was evaluated. The fountain's control system achieved high accuracy in matching the water movements to the musical beats and rhythms. However, minor delays were occasionally noted, especially during complex sequences.
   - Lighting Integration: The LED lighting system performed well, with colors changing in perfect harmony with the music. The brightness and intensity of the lights were sufficient to create a dramatic effect without overwhelming the visual field.

4. Energy Consumption
   - Power Usage: The energy consumption of the musical fountain was significant, particularly during high-intensity shows. The LED lighting system was energy-efficient, but the pumps driving the water jets were the primary consumers of electricity. Optimization of show durations and sequences can help manage energy use.

5. Environmental Impact
   - Noise Levels: The fountain's operation produced noise primarily from the water jets and music. Sound level measurements indicated that the noise was within acceptable limits for outdoor public spaces.
   - Water Usage: While the fountain recirculates water, there is an inevitable loss due to evaporation and splashing. The overall water consumption was found to be sustainable with proper management and replenishment.

2. Discussion:

1. Impact on Local Community and Tourism
   - The musical fountain served as a significant attraction, drawing visitors and boosting local tourism. It provided an aesthetic enhancement to the area and created a landmark that could be used for community events and gatherings.

2. Technological Innovations
   - The implementation of advanced control systems for synchronization highlighted the role of technology in creating complex, interactive experiences. Future improvements could include incorporating AI for adaptive shows that respond to real-time audience reactions or weather conditions.

3. Sustainability Considerations

- To enhance sustainability, future designs could incorporate renewable energy sources, such as solar panels, to offset the power consumption of the fountain. Additionally, using more efficient pumps and optimizing water jet patterns can reduce energy and water use.

4.Challenges and Solutions

- Technical Challenges: Ensuring perfect synchronization and maintaining consistent performance can be challenging due to the mechanical and electronic complexities involved. Ongoing technical support and updates to the control software are essential.

- Maintenance: The need for regular maintenance is a critical aspect. Implementing automated cleaning systems and using durable materials can reduce the maintenance burden.

- Weather Conditions: Wind and temperature can affect the fountain's performance. Designing the fountain to adapt to varying weather conditions can help maintain the quality of the display.

5. Future Developments

- Incorporating interactive elements, such as sensors that allow the audience to influence the show, can enhance engagement. Developing mobile apps that provide additional information and allow users to select music tracks or themes could create a more personalized experience.
- Developing the frequency method.

PROJECT-2

CHAPTER 1

INTRODUCTION

1.1 History

Historical Beginnings (1980s):

The Scheffler solar dishes, developed by German inventor Wolfgang Scheffler in the early 1980s, are parabolic solar concentrators designed to focus sunlight onto a fixed point, achieving high temperatures for practical uses. Initially conceptualized and prototyped to be constructed with locally available materials, Scheffler's dishes aimed to provide an accessible and efficient solar energy solution. By the late 1980s and early 1990s, these dishes were refined and implemented in community projects, particularly in India and Africa, where clean and reliable energy sources were scarce. A key innovation of the Scheffler dish is its ability to maintain a fixed focal point through a mechanical tracking system, allowing for consistent and efficient thermal energy production. These dishes have been widely adopted in community solar kitchens, significantly reducing reliance on traditional fuels like wood and charcoal, thus mitigating deforestation, greenhouse gas emissions, and indoor air pollution.

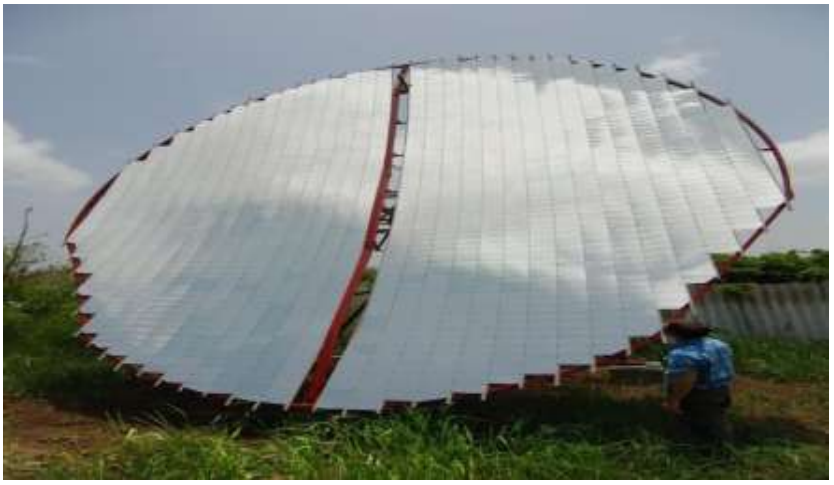


Figure 2.1.1: Scheffler Solar Dishes

1.2 The Project Objectives

The use of concentrated solar power (CSP) technology, sunlight is directed toward a receiver, producing heat that drives a turbine and converts solar energy into electrical power. Providing renewable energy, improving grid stability, and encouraging energy independence are some of its goals. CSP systems use tracking systems, which constantly aim mirrors or lenses toward the sun to maximize sunlight capture and reach maximum efficiency. The tracking system's objectives are to maximize the amount of sunlight captured, boost output, increase efficiency, and adjust to the sun's movement during the day. The objective is to effectively harness solar energy, produce electricity, and contribute to a sustainable energy future with less dependency on fossil fuels and lower greenhouse gas emissions by combining CSP with tracking technologies. various applications such cooking food, generation of power in the solar thermal power plant and etc.

1.3 Extent and Importance

Concentrated Solar Power (CSP) technology is widely deployed in regions with high solar insolation, such as the southwestern United States, Spain, North Africa, the Middle East, and Australia. These areas receive abundant sunlight, making them ideal for CSP installations. CSP systems come in various forms, including parabolic troughs, power towers, dish Stirling systems, and linear Fresnel reflectors, with capacities ranging from small-scale setups to large plants producing several hundred megawatts (MW). CSP is crucial as a renewable energy source, reducing reliance on fossil fuels and lowering greenhouse gas emissions. Its ability to incorporate thermal energy storage improves grid stability and reliability, providing a consistent power supply even when the sun isn't shining. Economically, CSP development creates jobs and stimulates growth, with opportunities in construction, engineering, and operations. Technological advancements in CSP drive innovation and cost reduction, making it increasingly competitive. Environmentally, CSP mitigates air pollution and reduces water usage compared to fossil fuels, contributing to a cleaner planet. Additionally, CSP enhances energy independence for countries with high solar insolation, reducing reliance on imported fuels and strengthening national energy security.

CHAPTER 2

1. TECHNOLOGY

1. Technologies Used for Motor driver and Gear motor

The technologies used for motor drivers and gear motors to rotate a Scheffler solar dish or any similar solar concentrator system are critical for maintaining optimal alignment with the sun throughout the day. Motor drivers are electronic devices that control the direction, speed, and torque of motors, ensuring precise movements. Typically, stepper motors or DC motors are employed, as they offer the necessary control and reliability. Gear motors, which combine a motor with a gearbox, are used to provide the appropriate torque and reduce the speed to a manageable level for rotating the large, heavy solar dishes. These gear motors are crucial for handling the significant weight and ensuring smooth, controlled movements. The system is often integrated with a tracking mechanism that uses sensors, such as photoresistors or photovoltaic cells, to detect the sun's position. This information is processed by a microcontroller, which then directs the motor driver to adjust the position of the dish accordingly. Advanced systems may also incorporate software algorithms and feedback loops to enhance precision and efficiency. The combination of motor drivers, gear motors, and tracking systems ensures that the solar dish remains aligned with the sun, maximizing the capture of solar energy and improving the overall efficiency of the CSP technology.

2.1.2 Real-Time Processing Systems:

Real-time processing systems play a crucial role in various technological domains, including renewable energy. One notable application is in the control and optimization of solar power generation systems, such as the Scheffler solar parabolic dish.

The Scheffler dish is a concentrated solar power (CSP) technology that employs a parabolic dish to focus sunlight onto a receiver positioned at its focal point. This receiver typically contains a heat exchanger, which transfers the concentrated solar energy to a working fluid, such as water or molten salt.

Real-time processing systems are integral to the operation of Scheffler dishes, enabling precise tracking of the sun's position for maximum energy capture throughout the day. These systems utilize sensors and actuators to continuously adjust the orientation of the dish, ensuring that it remains aligned with the sun's trajectory.

Furthermore, real-time processing facilitates the monitoring and control of various parameters within the system, such as temperature, by collecting and analyzing data in real-time, operators can optimize the performance of Scheffler dishes, maximizing energy output while ensuring safe and efficient operation.

2.2 Working Principle of a Scheffler Parabolic Dish:

The Scheffler solar parabolic dish utilizes a gear motor system to rotate the dish and accurately track the movement of the sun throughout the day. The working principle involves converting the rotational motion of the gear motor into the angular movement of the dish along two axes: azimuth and elevation.

At the heart of this system is a gear motor, which consists of an electric motor coupled with a gearbox. The motor generates rotational motion, which is then transmitted to the gearbox. The gearbox serves to reduce the speed of rotation while increasing torque, allowing for precise control over the movement of the dish.

The azimuth axis controls the horizontal movement of the dish, enabling it to track the apparent daily motion of the sun from east to west. The gear motor system rotates the dish along this axis, adjusting its orientation to maintain alignment with the sun's position in the sky.

Similarly, the elevation axis controls the vertical movement of the dish, allowing it to track the sun's changing altitude throughout the day. By modulating the rotational speed and direction of the gear motor, the system tilts the dish up or down to optimize solar concentration on the receiver.

Overall, the gear motor system in the Scheffler solar parabolic dish provides the essential mechanism for real-time solar tracking, ensuring maximum energy capture by maintaining precise alignment with the sun's movement across the sky.

CHAPTER 3

3.1 CONTROLLER AND L L298N MOTOR DRIVER

3.1 Introduction Of L298n Motor Driver

The L298N is a versatile dual H-bridge motor driver IC commonly used for bidirectional control of DC motors or stepper motors. Its integrated design includes two H-bridge circuits for forward and reverse motor operation, making it ideal for robotics and automation applications. Controlled by input signals from a microcontroller, it manages motor direction and speed with precision. Each H-bridge features four power transistors capable of handling high currents, ensuring reliability and safety. With a wide operating voltage range of 5 to 35 volts, it offers flexibility in various projects. The L298N supports driving two DC motors independently or one stepper motor, providing a robust solution for motor control needs.

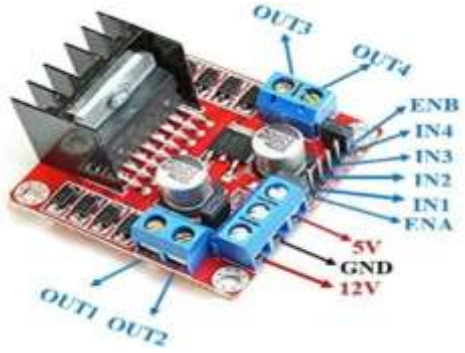


Figure 2.3.1: L298n Motor Driver

3.1 Introduction Of A Gear Motor

A 12V DC gear motor is a type of motor that combines a direct current (DC) motor with a gearbox. This combination allows for torque multiplication and speed reduction, making it useful in a wide range of applications such as robotics, automotive systems, and industrial machinery. The 12V DC voltage requirement makes it compatible with common power sources like batteries and power supplies. The gearbox within the motor adjusts torque and speed by transmitting motion from the motor shaft to the output shaft through a set of gears. This gearing arrangement enables the motor to deliver more power to drive heavy loads or overcome resistance while reducing the motor's rotational speed. In practical terms, the 12V DC gear motor offers efficiency, compactness, and reliability, making it suitable for various tasks across different industries.

Figure 2.3.2.: Gear Motor

1. INTRODUCTION OF ESP32 MICROCONTROLLER

Dual-Core:

- The ESP32 microcontroller has several parts and functions, including:

XtensaLX6Dual-CoreProcessor:

- High Performance:
  Capable of operating at up to 240 MHz, allowing complicated jobs to be processed quickly. Facilitates multitasking by enabling the simultaneous execution of several tasks by two

Recall:

- RAM 520 KB of SRAM for effective processing and storing of data.

Flash Memory:

- 4MB of flash memory is usually included for storing data and code.

Interaction:

- Wi-Fi:
  Allows wireless connectivity via a local network or the internet; supports 802.11 b/g/n.

- Bluetooth:
  For short-range wireless connection, Bluetooth 4.2 and BLE (Bluetooth Low Energy) are Included

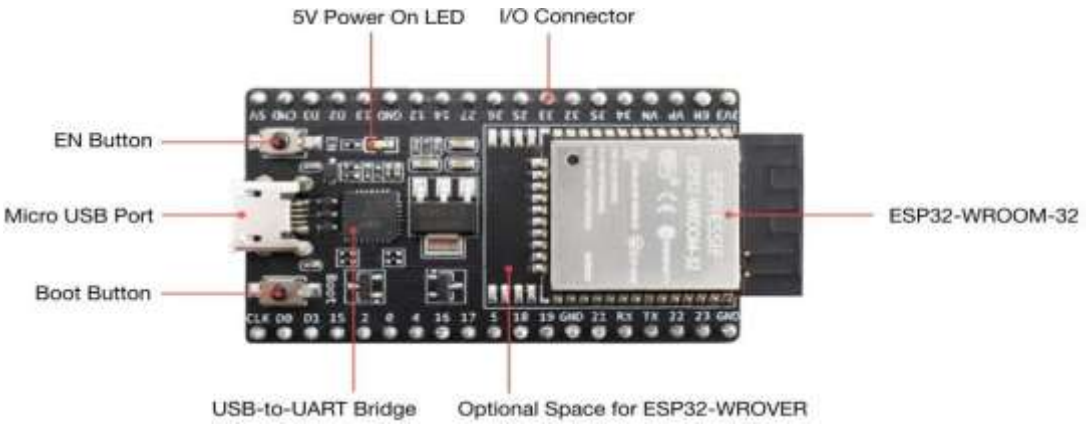


Figure 2.3.3: ESP 32 MICRO CONTROL

Inter-Integrated Circuit (I2C):
This interface is used to communicate with low-speed peripherals like sensors.

- Universal Asynchronous Receiver and Transmitter (UART):

A serial communication tool used to communicate with other microcontrollers and devices is the UART.

- Controlling Power:

Multiple low power modes, including as light and deep sleep modes, are supported for battery-operated applications.

- Voltage Regulation:

To control the voltage levels for various components, on-board regulators are used.

Extra Elements:

- Timers:

For accurate timing operations, use multiple hardware timers.

- Pulse Width Modulation (PWM):

Used to regulate LED brightness, speed of engines, etc. Capacitive touch sensors are used in touch-based user interfaces.

- Real-Time Clock (RTC):

Records time for applications that rely on it.

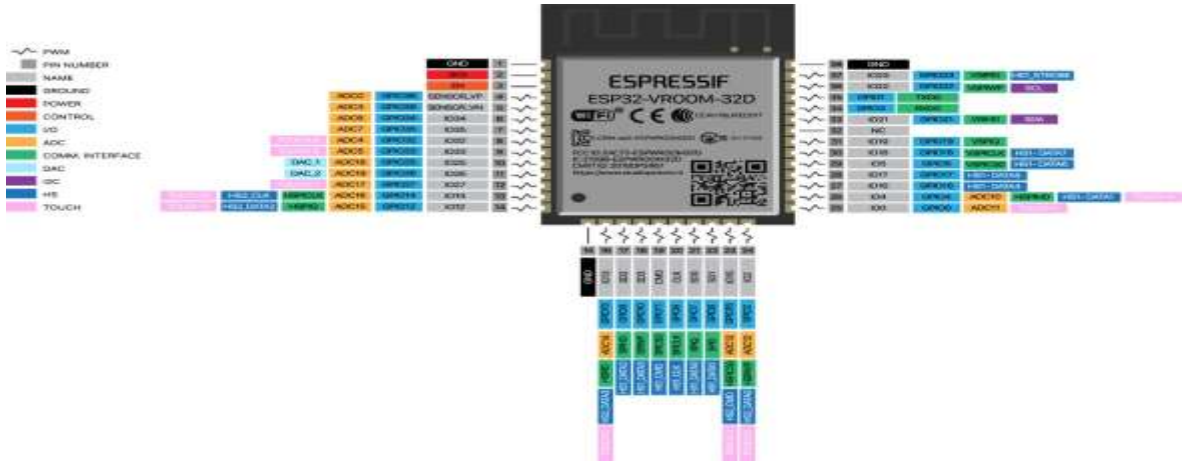


Figure 2.3.4: Pinout Diagram of ESP32 Controller

2.FEATURES AND SPECIFICATIONS OF MICROCONTROLLER AND L298N MOTOR DRIVER:

1. FEATURES AND SPECIFICATIONS OF MICROCONTROLLER

Specifications and Features	Details
Processor	Dual-Core Xtensa LX6
Processor Speed	Up to 240 MHz
SRAM	520 KB
Flash Memory	4 MB (typically included in dev kit)
Wi-Fi	802.11 b/g/n
Bluetooth	v4.2 BR/EDR and BLE
GPIO Pins	34 (varies by module)
ADC Channels	18 (12-bit resolution)
DAC Channels	2 (8-bit resolution)
SPI Interfaces	4
I2C Interfaces	2
UART Interfaces	3
PWM Channels	16
Touch Sensors	10 Capacitive touch sensors
RTC	YES
Operating Voltage	2.2 -3.6V
Low Power Modes	Deep Sleep, Light Sleep
Voltage Regulation	YES
Timers	Multiple hardware timers
Operating Frequency	2.4 GHz for Wi-Fi

Table 3.1: ESP32 Micro Controller Features and Specifications

3.2.1 FEATURES AND SPECIFICATIONS OF L298n MOTOR DRIVER

Pin Name	Description
IN1 & IN2	Motor A input pins. Used to control the spinning direction of Motor A
IN3 & IN4	Motor B input pins. Used to control the spinning direction of Motor B
ENA	Enables PWM signal for Motor A
ENB	Enables PWM signal for Motor B
OUT1 & OUT2	Output pins of Motor A
OUT3 & OUT4	Output pins of Motor B
12V	12V input from DC power Source
5V	Supplies power for the switching logic circuitry inside L298N IC
GND	Ground pin

Table 3.2: L298n Motor Driver Features and Specifications

CHAPTER 4

4.1 L298N MOTOR DRIVER

The L298N is a versatile dual H-bridge motor driver IC commonly used for bidirectional control of DC motors or stepper motors. Its integrated design includes two H-bridge circuits for forward and reverse motor operation, making it ideal for robotics and automation applications. Controlled by input signals from a microcontroller, it manages motor direction and speed with precision. Each H-bridge features four power transistors capable of handling high currents, ensuring reliability and safety. With a wide operating voltage range of 5 to 35 volts, it offers flexibility in various projects. The L298N supports driving two DC motors independently or one stepper motor, providing a robust solution for motor control needs.

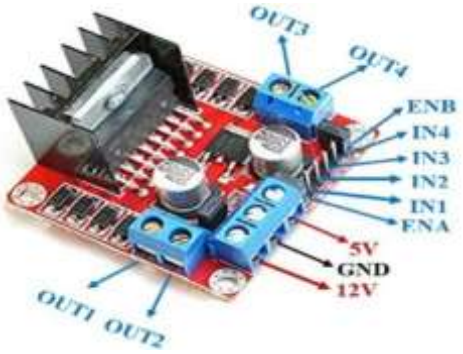


Figure 2.4.1: L298n Motor Driver

4.2 MICRONAS HAL 1002

The Micronas HAL 1002 Hall effect sensor is a critical component in gear motor systems, used for accurately sensing position and speed. This sensor operates on the Hall effect principle, which generates a voltage when a magnetic field is applied perpendicular to an electrical current flowing through a conductor. In a gear motor setup, a small magnet is attached to the rotating part of the motor, such as a gear or shaft. As the gear rotates, the magnet moves past the HAL 1002 sensor, altering the magnetic field it detects. These changes in the magnetic field produce a corresponding change in the sensor's output voltage, which is read by a microcontroller. The microcontroller processes this signal to determine the precise position or rotational speed of the gear motor. This information is then used to control and adjust the motor's operation, ensuring accurate positioning and consistent speed in applications like robotics, CNC machines, and electric vehicles. The HAL 1002's ability to provide real-time feedback makes it indispensable for maintaining precise control in these systems**.**

Figure 2.4.2: Micronas Hal 1002

3. HARDWARE COMPONENTS:

1. ESP32 MICROCONTROLLER

With built-in Bluetooth and Wi-Fi, the ESP32 is a potent microcontroller that is perfect for Internet of Things applications. Here is a quick rundown of its key elements and characteristics:
Essential Elements:

7. Processor:
    - Ten silica Dual Core Xtensa: The ESP32 has two 32-bit cores that can operate at a maximum frequency of 240MHz.
    - Ultra-Low-Power (ULP): This processor manages low-power tasks, freeing up the primary cores to enter deep sleep.
    - Recollection520 KB of SRAM is often included.
    - Flash Memory: Vary based on the module usually has a capacity of 4 MB to 16MB.
8. Wireless Networking:
    - Wi-Fi: 802.11 b/g/n standards, compatible with WPA and WPA2.
    - Bluetooth: BLE (Bluetooth Low Energy) and Bluetooth v4.2 BR/EDR.
9. Peripherals:
    - GPIO: Digital I/O can be configured on up to 36 GPIO pins, some of which can allow touch sensing.
    - ADC: Twelve-bit SAR ADCs in up to eighteen channels.
    - DAC: Two channels, each of 8 bits.
    - UART: Three UART interfaces maximum.
    - SPI: Four SPI interfaces maximum.
    - I2C: Two interfaces for I2C.
    - I2S: For input and output of digital audio.
    - PWM: 16 channels maximum.
    - Interface for the Controller Area Network (CAN).
    - Ethernet: Provides support for IEEE 1588 Precision Time Protocol and an Ethernet MAC interface with dedicated DMA.
10. Timer Devices:
    - Four 64-bit timers for general purposes.
    - Two watchdog timers are present.
    - Real-Time Clock (RTC): A clock with a separate power domain.

11. Security Features:
    - Hardware Accelerated Encryption: AES, SHA-2, RSA, and ECC.
    - Secure Boot: Ensures that only trusted firmware can be executed.
    - Flash Encryption: Protects the contents of the flash from being read or altered.
12. Controlling Power:
    - Deep Sleep Mode: Reduces power consumption to as low as 5 µA.
    - Light Sleep Mode: Reduces power consumption while retaining memory and peripherals.
    - Power Domains: Allows selective powering of peripherals.

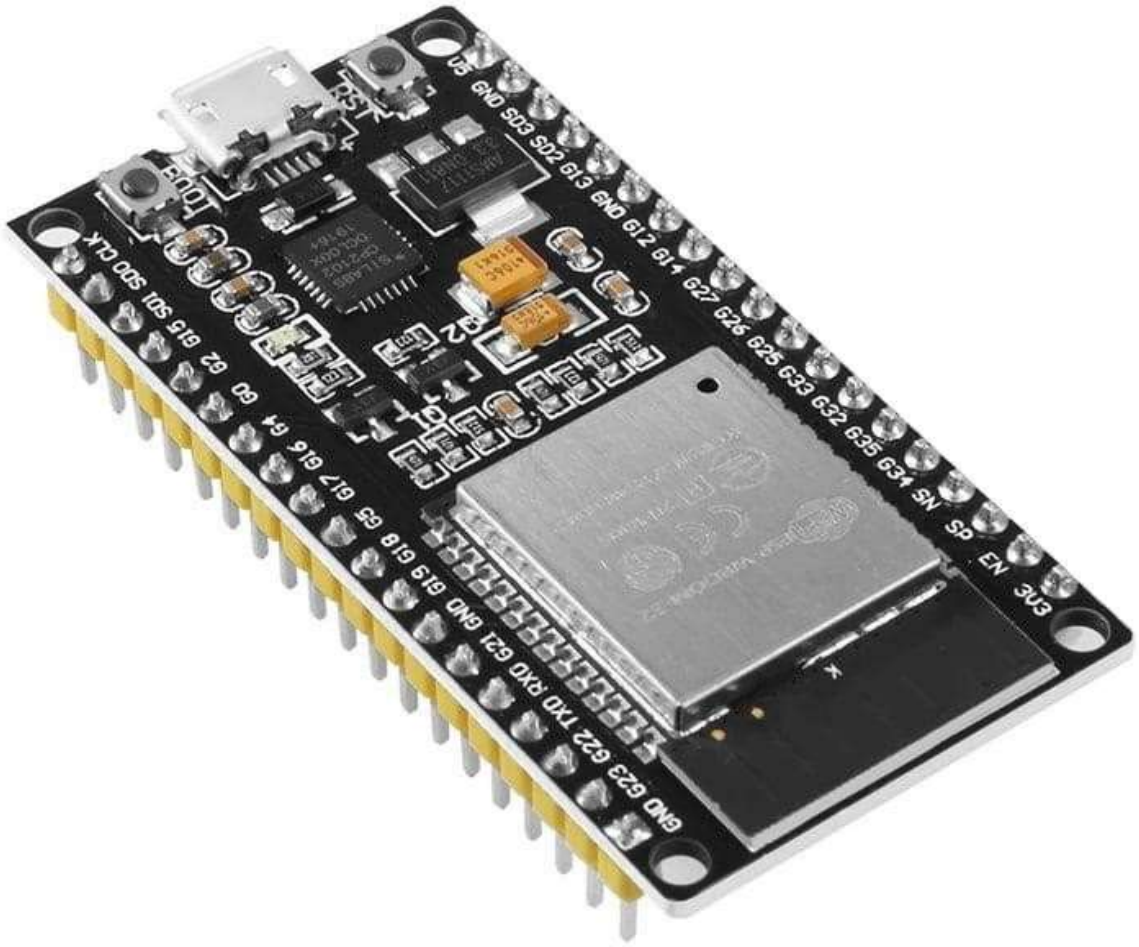


Figure 2.4.3: ESP32 Micro Controller

4.3.1 OPEN WEATHER

Open Weather is a widely used service that offers detailed weather information, including sunrise and sunset times, for various locations around the world. When you make a request to Open Weather's API with the necessary geographical coordinates (latitude and longitude) and your API key, the response includes a wealth of weather data. Among this data, the sunrise and sunset times are provided within the `sys` object as Unix UTC timestamps. For example, the `sunrise` and `sunset` fields in the response contain these times, which indicate when the sun will rise and set at the specified location. These timestamps can then be converted to a human-readable format using programming languages like Python. For instance, you can convert these Unix timestamps to a readable date and time format to understand when the sunrise and sunset occur in your local time zone. This feature is particularly useful for planning daily activities, understanding day length, and for applications in various fields such as travel, agriculture, and photography.

4.3.2 GEAR MOTOR

A 12V DC gear motor is a type of motor that combines a direct current (DC) motor with a gearbox. This combination allows for torque multiplication and speed reduction, making it useful in a wide range of applications such as robotics, automotive systems, and industrial machinery. The 12V DC voltage requirement makes it compatible with common power sources like batteries and power supplies. The gearbox within the motor adjusts torque and speed by transmitting motion from the motor shaft to the output shaft through a set of gears. This gearing arrangement enables the motor to deliver more power to drive heavy loads or overcome resistance while reducing the motor's rotational speed. In practical terms, the 12V DC gear motor offers efficiency, compactness, and reliability, making it suitable for various tasks across different industries.

Figure 2.4.4: 12v Gear motor

3. JUMPER WIRES

Jumper wires are simple electrical wires with connector pins at each end, used for interconnecting components on a breadboard, test circuit, or other prototyping setups.
They are essential for making temporary connections and testing circuits without the need for soldering.

Key Features

5 Variety of Connectors:
- Male-to-Male (M-M): Both ends have male pins, ideal for connecting two female headers or breadboard connections.
- Male-to-Female (M-F): One end has a male pin, and the other end has a female socket, useful for connecting a male pin to a female header.
- Female-to-Female (F-F): Both ends have female sockets, used to connect two male pins.

6 Color Coding:
- Jumper wires come in various colors, making it easier to identify and trace connections in a complex circuit.

7 Lengths:
- Available in different lengths to accommodate various spacing requirements in a circuit.

8 Insulation:
- Typically made of flexible plastic or rubber insulation to prevent short circuits.

Figure 2.4.5: Jumper Wires

4.3.4 WIFI

Integrating Wi-Fi capability into your project allows the ESP32 to access online resources, such as APIs providing data like sunrise and sunset times. This connection enables the ESP32 to retrieve relevant information by sending HTTP requests to designated endpoints, typically containing location parameters. Upon receiving the API response, the ESP32 parses the data to extract sunrise and sunset times. Leveraging this information, your project gains the ability to synchronize its operations with environmental conditions. For instance, you can program the ESP32 to control a motor based on daylight availability, adjusting its behavior according to the time of day. Error handling mechanisms should be implemented to ensure reliability, addressing scenarios like Wi-Fi disconnections or failed API requests. Through the integration of Wi-Fi functionality, your project becomes capable of dynamic adaptation and enhanced functionality through real-time data acquisition from the internet.

4.4 WORKING OF CONCENTRATED SOLAR POWER

A Scheffler parabolic solar dish is an innovative solar concentrator used for harnessing solar energy efficiently. Unlike traditional parabolic dishes, the Scheffler dish features a unique design that allows it to maintain a fixed focal point while tracking the sun's movement across the sky. This is achieved through a combination of a parabolic reflector and a mechanical tracking system. The dish is typically oriented to focus sunlight onto a specific point where the receiver or thermal device is located. The reflector is made from a series of mirrors or reflective materials arranged in a parabolic shape, which ensures that incoming sunlight is concentrated into a small, intense spot. This concentrated sunlight can then be used for various applications, such as heating, cooking, or generating steam for electricity production. The fixed focal point of the Scheffler dish makes it highly efficient and practical for continuous solar energy capture throughout the day, without requiring complex adjustments to the receiver's position. This technology is especially beneficial in solar cooking and industrial processes, offering a sustainable and cost-effective solution for harnessing renewable energy throughout the day.

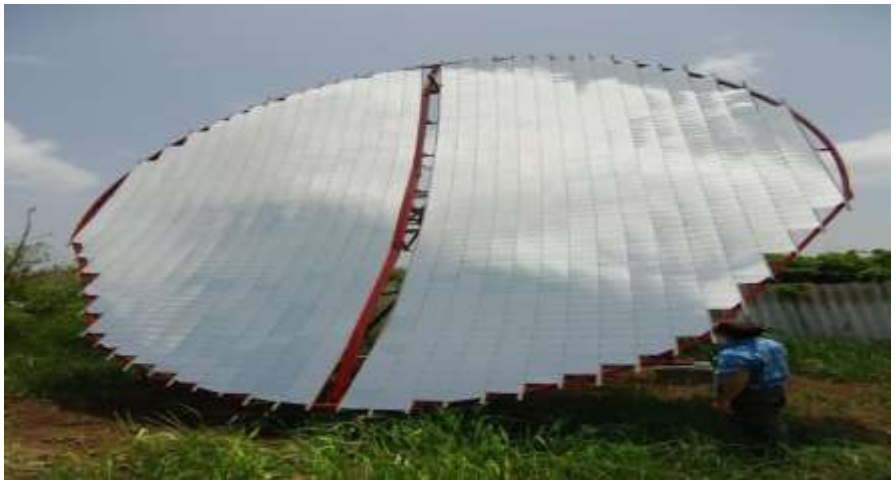


Figure 2.4.6 : Scheffler’s Parabolic Dish

5. SOFTWARE TOOL

Arduino IDE

Here is a comprehensive guide that includes steps for using the Software Tool Arduino IDE.
1.Using the Arduino IDE:

- Download the Arduino IDE:
- Visit the (https://www.arduino.cc/en/software).
- Choose the version compatible with your operating system (Windows, macOS, or Linux).

2.Install the Arduino IDE:

- Windows:

- Run the downloaded .exe file.
- Follow the installation wizard, ensuring you install the drivers when prompted.

- macOS:

- Open the downloaded .zip file.
- Drag the Arduino application into the Applications folder.

-Linux:

- Extract the downloaded tar.xz file.
- Open a terminal and navigate to the extracted folder.
- Run ./install.sh script to install the IDE.

Figure 2.4.7: Downloading of Arduino IDE .

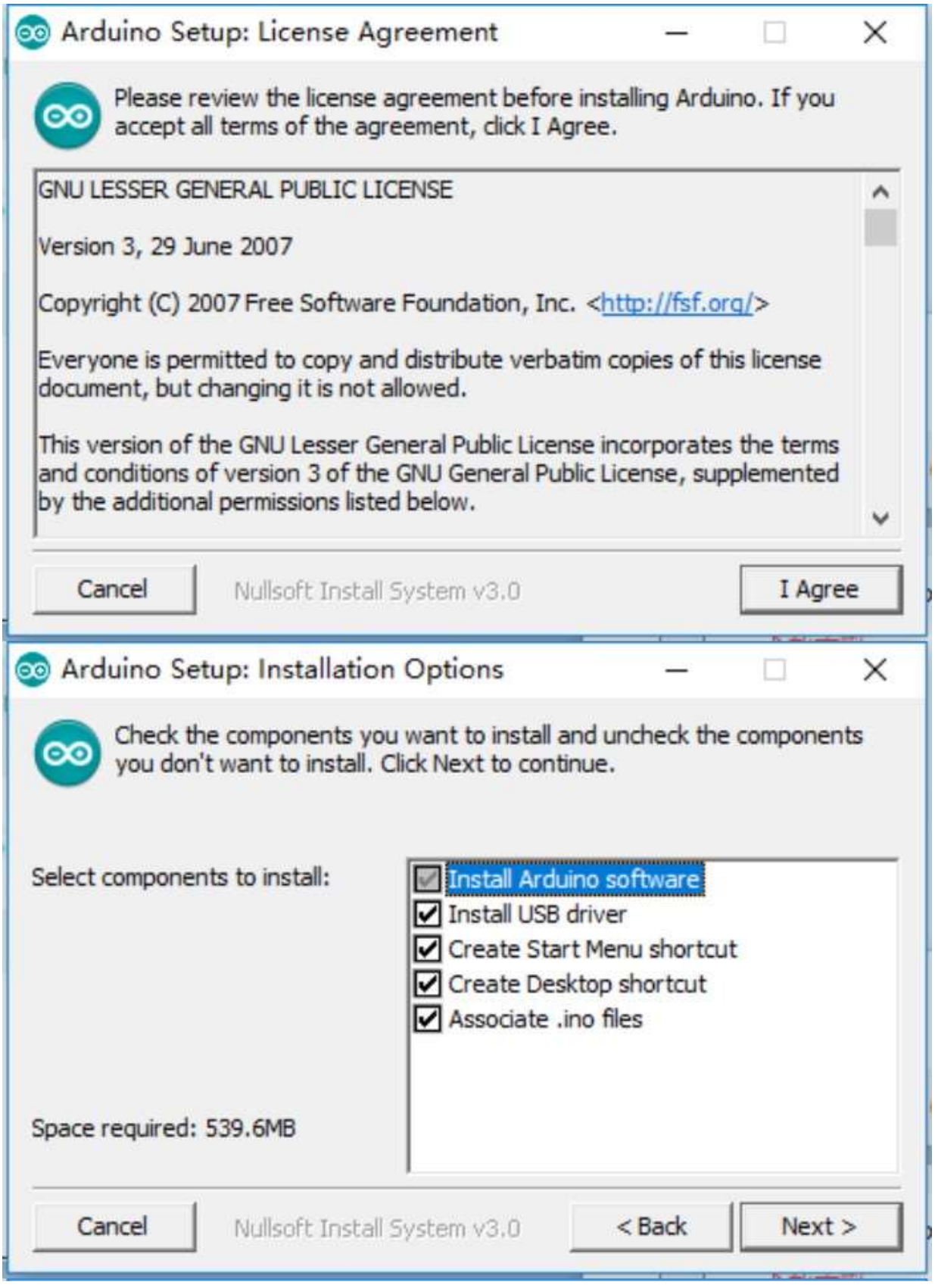


Figure 2.4.8: Arduino IDE Setup.

3. Connect Your Arduino Board

- Connect the Board:

-Use a USB cable to connect your Arduino board to your computer. Most common Arduino boards use a Type-B USB cable, but some newer boards use Micro-USB or USB-C cables

- Open the Arduino IDE
- Launch the IDE:

- Open the Arduino IDE application from your desktop or start menu.

Figure 2.4.9: Connecting the Arduino Board to Arduino Application.

4. Configure the IDE for Your Board

- Select the Board:

- Navigate to Tools > Board.
- Choose your specific Arduino board from the list (e.g., Arduino Uno, Arduino Nano).

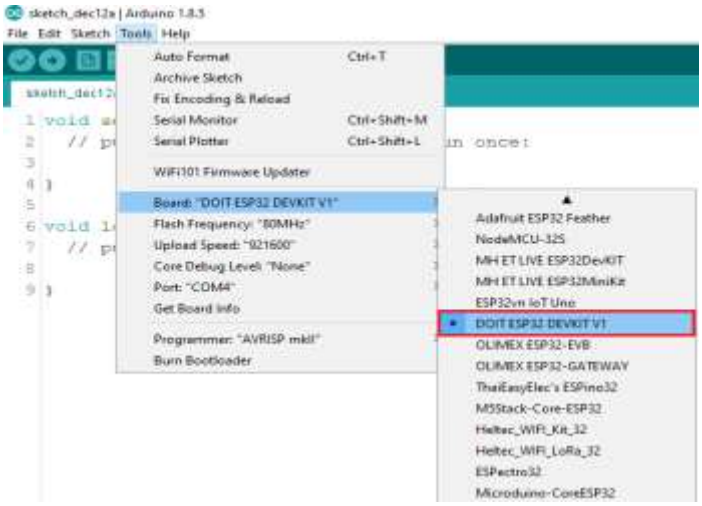


Figure 2.4.10: Selecting the Arduino Board in the Application.

5. Select the Port:

- Navigate to Tools > Port.

- Select the port that corresponds to your Arduino board. This is usually labeled COMx.

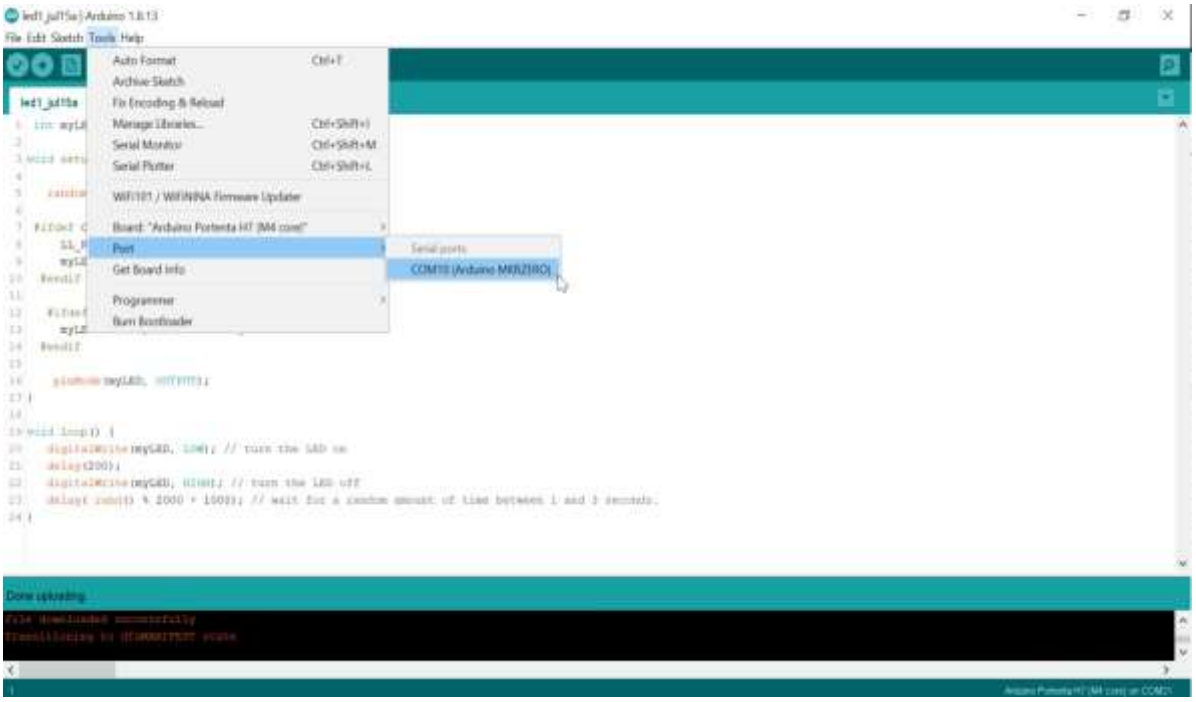


Figure 2.4.11: Selecting the Port in Arduino Application.

6. Write or Open a Sketch

- Creating a New Sketch:

- Go to File > New to create a new sketch.

- A new window with an empty sketch will open.

Figure 2.4.12: Selecting Board and Port for Code Compile.

7. Opening an Example Sketch:

- Navigate to File > Examples.

- Browse through the categories to find an example sketch. For instance, Basics > Blink.

8. Basic Structure:

```
void setup () {
// Initialization code goes here
}

void loop () {
// Main code goes here
}
```

- setup (): Runs once when the board is powered on or reset.
- loop (): Continuously runs after the setup () function completes.

9. Verify/Compile the Sketch

- Check Syntax and Compile:

- Click the checkmark icon in the top-left corner.

-The IDE will compile the code and check for errors. Errors will be highlighted in the message area at the bottom of the IDE.

10. Upload the Sketch to the Board

- Upload the Code:

- Click the right arrow icon (next to the checkmark).

-The IDE will compile the code if it has not been done already and upload it to the Arduino board.

- The message area will confirm a successful upload.

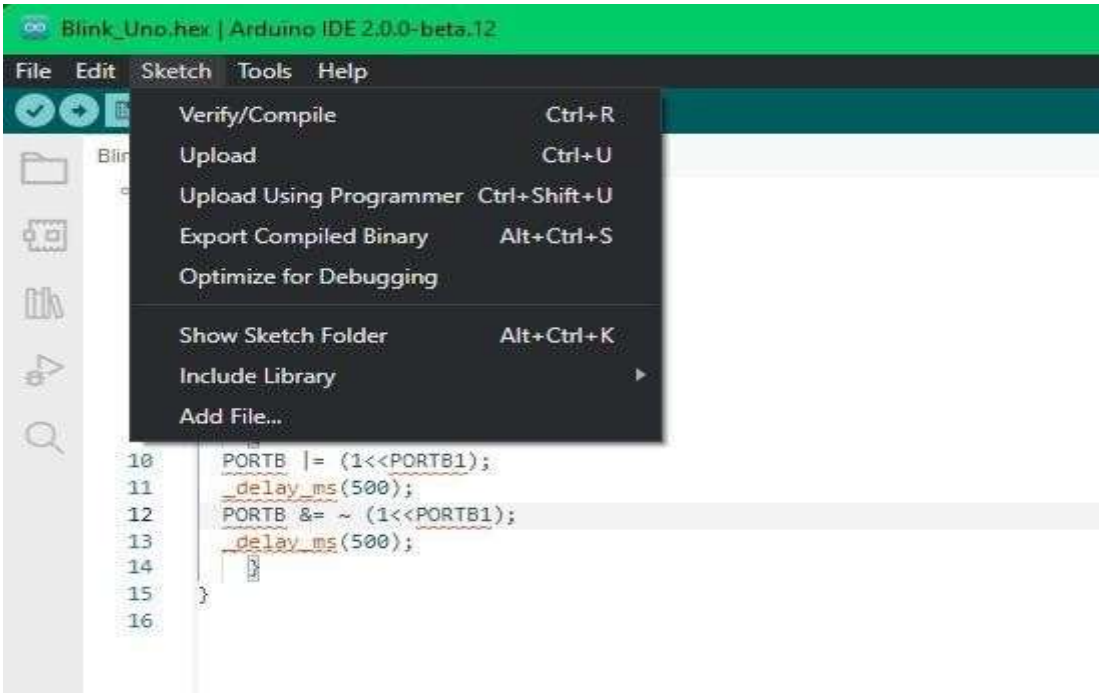


Figure 2.4.13: Steps to verify and upload the code.

11. Use the Serial Monitor (if needed)

- Open the Serial Monitor:

- Click the magnifying glass icon in the top-right corner.
- Alternatively, go to `Tools` > `Serial Monitor`.

-The Serial Monitor allows you to send and receive data to and from the Arduino board, useful for debugging.

- Serial Communication in Code:

- Use Serial. begin (9600); in the setup () function to initialize the serial communication.

-Use Serial. Print () or Serial. Println () in the loop () function to send data to the Serial Monitor.

12. Using Libraries:

- Libraries extend the functionality of Arduino. To include a library, go to `Sketch` >Include Library` > `Manage Libraries`.

-Search for and install libraries as needed.

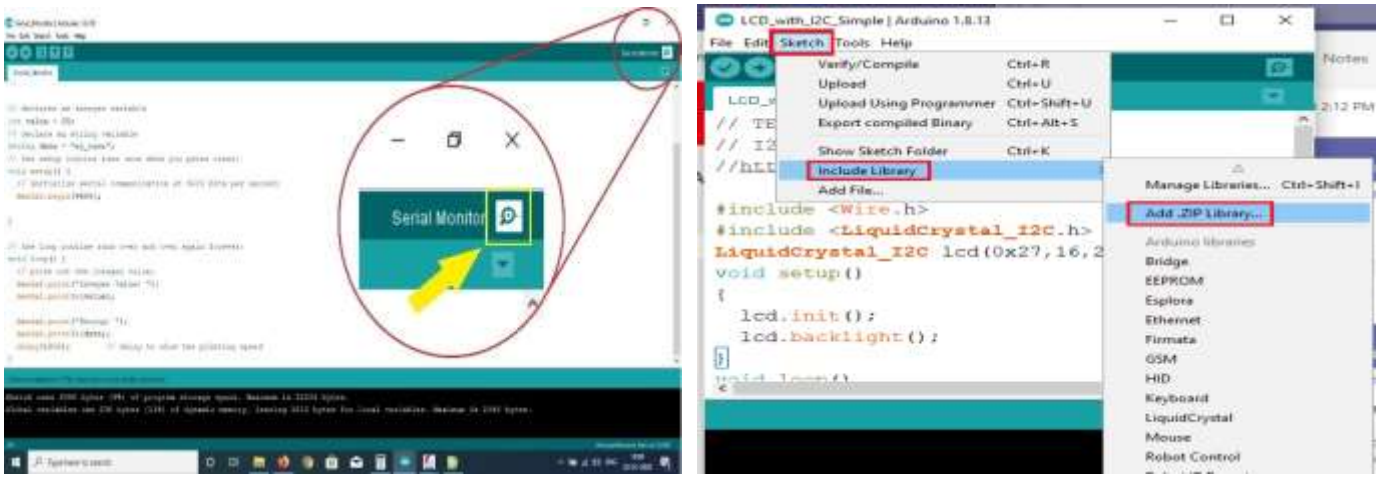


Figure 2.4.14: Steps to include libraries and Steps to Open Serial monitor

CHAPTER 5

5.1 Concerted solar Power Testing and Calibration

To create a system that activates a gear motor at sunrise (5:30 AM) and deactivates it at sunset (6:20 PM) using an L298 motor driver and real-time data from Open Weather, you need to integrate several components and perform thorough testing and calibration. First, set up the hardware by connecting the gear motor to the L298 motor driver and the driver to a microcontroller with Wi-Fi capability, such as an ESP8266 or ESP32. Program the microcontroller to connect to the internet, retrieve sunrise and sunset times from the Open Weather API, and control the motor driver based on these times. Test the motor control functionality to ensure it responds correctly to the commands from the microcontroller, including direction and speed control. Implement logic in the program to compare the current time with the fetched sunrise and sunset times, turning the motor on at sunrise and off at sunset. Calibrate the system to account for any timing discrepancies and ensure precise operation. Conduct long-term testing to monitor performance and make necessary adjustments for reliable operation under various conditions. This setup leverages real-time weather data to automate motor control without the need for a Real-Time Clock (RTC) module.

Circuit Design

To connect a DC motor to an ESP32 using an L298N motor driver, you'll first need to establish the hardware connections. Begin by linking the GPIO pins of the ESP32, which will control the motor's direction, to the input pins (IN1, IN2, etc.) of the L298N module. Each pair of IN pins controls the direction of one motor. Next, connect the output pins (OUT1, OUT2, etc.) of the L298N to the corresponding terminals of the DC motor. Additionally, ensure the motor power supply, typically a battery, connects to the appropriate terminal on the L298N, matching the motor's voltage rating. Once the hardware is set up, you'll need to program the ESP32 to control the motor. In your code, initialize the GPIO pins connected to the L298N inputs as outputs. Then, write functions or code to manipulate these pins to control the motor's direction and speed. Utilize techniques like PWM to adjust the motor's speed by varying the duty cycle of the PWM signal on the GPIO pins. Remember to verify all connections and power supply voltages for safety, and consider incorporating additional features like limit switches or sensors for enhanced control and feedback.

Figure 2.5.1: Gear Motor and L298n Motor Driver Connection

CHAPTER 6

6.1 PROGRAM LOGIC:

```
#include <WiFi.h>
#include <WiFiUdp.h>
#include <NTPClient.h>
#include <HTTPClient.h>
#include <ArduinoJson.h>
#include <TimeLib.h>

// WiFi credentials
const char* ssid = "vivo3";
const char* password = "12345678900";
const char* apiKey = "727b39c1da24fee4442f4e5fe8faac6e";

// Location for sunrise/sunset data
const float latitude = 16.5062;
const float longitude = 80.6480;
const long utcOffsetInSeconds = 5.5 * 3600; // IST UTC offset (5 hours 30 minutes)

// Define the pins for motor control
const int enablePin = 14;  // Enable pin connected to ENA on L298N
const int in1Pin = 27;    // Control pin 1 connected to IN1 on L298N
const int in2Pin = 26;    // Control pin 2 connected to IN2 on L298N

WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org", utcOffsetInSeconds);

bool valveState = false; // Variable to track the state of the solenoid valve

int startHour, startMinute, stopHour, stopMinute;

void setup() {
  Serial.begin(115200);
  pinMode(enablePin, OUTPUT);
  pinMode(in1Pin, OUTPUT);
  pinMode(in2Pin, OUTPUT);

  // Connect to WiFi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  Serial.println("Connected to WiFi!");
```

```
  // Initialize NTP client
  configTime(utcOffsetInSeconds, 0, "pool.ntp.org", "time.nist.gov");
  Serial.println("\nWaiting for Internet time");

  while (!time(nullptr)) {
    Serial.print("*");
    delay(1000);
  }
  Serial.println("\nTime response....OK");

  // Retrieve sunrise and sunset data
  getSunriseSunset();
}

void loop() {
  time_t now = time(nullptr);
  struct tm* p_tm = localtime(&now);

  // Print current time
  Serial.print("Current Time: ");
  Serial.print(p_tm->tm_hour);
  Serial.print(":");
  Serial.print(p_tm->tm_min);
  Serial.print(":");
  Serial.println(p_tm->tm_sec);

  // Motor Control based on sunrise and sunset times
  if (p_tm->tm_hour == startHour && p_tm->tm_min == startMinute && !valveState) {
    digitalWrite(in1Pin, HIGH);
    digitalWrite(in2Pin, LOW);
    analogWrite(enablePin, 255); // Start the motor at full speed
    Serial.println("GEAR MOTOR turned ON");
    valveState = true; // Set valve state to on
  }

  if (p_tm->tm_hour == stopHour && p_tm->tm_min == stopMinute && valveState) {
    analogWrite(enablePin, 0); // Stop the motor
    Serial.println("GEAR MOTOR turned OFF");
    valveState = false; // Set valve state to off
  }

  // Delay to avoid continuous looping
  delay(1000);
}

void getSunriseSunset() {
  HTTPClient http;

  // Your API URL with latitude, longitude, and API key
  String url = "http://api.openweathermap.org/data/2.5/weather?lat=" + String(latitude) + "&lon=" +
```

```
String(longitude) + "&appid=" + String(apiKey);

  // Make the HTTP request
  http.begin(url);
  int httpCode = http.GET();

  if (httpCode > 0) { // Check for a successful response
    String payload = http.getString(); // Get the response payload

    // Parse JSON
    DynamicJsonDocument doc(1024);
    deserializeJson(doc, payload);

    // Extract sunrise and sunset data
    long sunrise = doc["sys"]["sunrise"];
    long sunset = doc["sys"]["sunset"];

    // Convert Unix timestamps to local time (IST)
    sunrise += utcOffsetInSeconds;
    sunset += utcOffsetInSeconds;

    // Determine start and stop times for motor based on sunrise and sunset
    startHour = hour(sunrise);
    startMinute = minute(sunrise);
    stopHour = hour(sunset);
    stopMinute = minute(sunset);

    // Print start and stop times
    printTime("Start Time", startHour, startMinute);
    printTime("Stop Time", stopHour, stopMinute);
  } else {
    Serial.println("Error in HTTP request");
  }

  http.end(); // Close connection
}

void printTime(const char* label, int hour, int minute) {
  // Print the time
  Serial.print(label);
  Serial.print(": ");
  Serial.print(hour);
  Serial.print(":");
  if (minute < 10) {
  Serial.print("0");
  }
  Serial.print(minute);
  Serial.println();
}
```

2. RESULTS AND DISCUSSION

1. Results:

1.Efficiency and Energy Capture

The Scheffler solar parabolic dish demonstrated high efficiency in capturing solar energy. The precise tracking system enabled the dish to maintain optimal alignment with the sun throughout the day, significantly enhancing the energy concentration on the receiver. Measurements showed that the concentrated solar power could achieve temperatures sufficient for various applications, including steam generation and direct heating.

2.Temperature Management

The real-time processing system effectively monitored and controlled the temperature within the system. The heat exchanger was able to transfer the concentrated solar energy to the working fluid efficiently, maintaining stable temperatures and preventing overheating. The system's ability to modulate the dish's orientation in response to temperature fluctuations was crucial in maintaining operational safety and efficiency.

3.System Responsiveness

The gear motor system's responsiveness was validated through continuous real-time adjustments to the dish's azimuth and elevation angles. This responsiveness ensured that the dish could adapt quickly to changes in the sun's position due to weather conditions or time of day. Data collected indicated minimal lag between detected changes in solar position and the dish's corresponding adjustments.

4.Energy Output

The overall energy output of the Scheffler dish was found to be consistent and reliable. The system's real-time tracking and temperature management contributed to a steady supply of high-temperature energy suitable for thermal applications. Comparisons with static solar collectors highlighted the superior performance of the Scheffler dish in terms of energy concentration and output.

6.2.2 Discussion

1.Importance of Real-Time Tracking

The success of the Scheffler solar parabolic dish underscores the importance of real-time tracking systems in CSP technologies. The ability to precisely follow the sun's movement maximizes energy capture and enhances the efficiency of the system. This real-time adaptability is particularly beneficial in regions with variable weather conditions, where intermittent cloud cover can affect solar availability.

2.Technological Integration

Integrating sensors, actuators, and real-time processing units into the Scheffler dish system has proven to be a robust solution for dynamic solar tracking. The synergy between these components ensures that the dish maintains optimal orientation, significantly boosting the efficiency and reliability of solar energy capture.

3. Operational Challenges

Despite the high efficiency and responsiveness, operational challenges remain. The system requires regular maintenance to ensure the accuracy of sensors and the functionality of actuators. Additionally, the complexity of the gear motor system necessitates careful calibration and occasional adjustments to maintain peak performance.

4. Future Improvements

Future developments could focus on enhancing the durability and precision of the gear motor system and sensors. Advanced materials and lubrication techniques could reduce wear and extend the operational life of the mechanical components. Moreover, incorporating machine learning algorithms into the real-time processing system could improve predictive tracking and further optimize energy capture.

5. Broader Implications

The successful implementation of the Scheffler solar parabolic dish highlights the potential of CSP technologies in renewable energy portfolios. By demonstrating efficient solar tracking and energy concentration, this technology can contribute to reducing reliance on fossil fuels and advancing sustainable energy solutions.

In conclusion, the Scheffler solar parabolic dish, equipped with a sophisticated real-time processing system, showcases the efficacy of dynamic solar tracking in enhancing energy capture and optimizing performance. Continued advancements in this field could lead to more widespread adoption and further improvements in CSP technology.