

ARDUINO- CONTROLLED AQUAPONICS MONITORING AND CONTROL SYSTEM

Submitted in partial fulfillment of the requirements for the award of Bachelor of
Technology degree in Biotechnology
Interdisciplinary project

By

ADITYA.S (41230003)

PRANAV.R (41230040)



DEPARTMENT OF BIOTECHNOLOGY

SCHOOL OF BIO AND CHEMICAL ENGINEERING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

CATEGORY-1 UNIVERSITY BY UGC

Accredited with Grade “A++” by NAAC | 12B Status by UGC | Approved by AICTE

Accredited with grade “A++” by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI- 6000019

MAY-2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

[DEEMED TO BE UNIVERSITY]

CATEGORY-I UNIVERSITY BY UGC

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF BIOTECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Aditya.S(41230003)** and **Pranav.R (41230040)** who carried out the Interdisciplinary Project entitled **"ARDUINO- CONTROLLED AQUAPONICS MONITORING AND CONTROL SYSTEM"** under our supervision from January 2024 to April 2024.

Internal Guide

Dr. USHA NANDHINI, M.Phil., Ph.D.

Head of the Department

Dr. V. RAMESH KUMAR, M.Phil., Ph.D.

Submitted for Viva voce Examination held on 6/5/2024

Internal Examiner

External Examiner

DECLARATION

We, **Aditya.S (41230003)** and **Pranav.R (41230040)** hereby declare that the Interdisciplinary Project Report entitled **“ARDUINO- CONTROLLED AQUAPONICS MONITORING AND CONTROL SYSTEM”** done by us under the guidance of **Dr. USHA NANDHINI, M.Phil., Ph.D.**, is submitted in partial fulfilment of the requirements for the award of Bachelor of Technology degree in Biotechnology.

DATE: 06/05/2024

PLACE: CHENNAI

A handwritten signature in black ink that reads "Aditya.S". The signature is written in a cursive, flowing style with a large initial 'A' and a distinct dot over the 'y'.

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

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

We convey our sincere thanks to **Dr.V.RAMESH KUMAR, M.Phil., Ph.D., Head of the Department, Department of Biotechnology** for providing me necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to my project guide **Dr. USHA NANDHINI, M.Phil., Ph.D., Assistant Professor, Department of Biotechnology** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

We also express my thanks to all **Teaching and Non-teaching staff** members of the Department of Biotechnology who were helpful in many ways for the completion of the project work.

ABSTRACT

Aquaponics is a sustainable agricultural practice that integrates aquaculture and hydroponics in a symbiotic environment. In this system, fish or other aquatic organisms are raised in tanks or ponds, generating waste in the form of ammonia through excretion and respiration. This wastewater is then circulated to grow beds or trays where plants, typically vegetables or herbs, are cultivated without soil. The Arduino-controlled aquaponics monitoring system represents a significant advancement in the field of aquaponics, a sustainable farming technique that combines aquaculture and hydroponics. Through the utilization of Arduino microcontrollers, this system introduces a sophisticated technological layer that allows for precise monitoring and control of crucial environmental factors that are vital for the overall health and productivity of the aquaponic ecosystem. By employing strategically placed sensors, real-time data collection becomes possible, enabling informed decision-making and timely adjustments to optimize conditions such as water quality, temperature, and turbidity levels.

Moreover, the automation capabilities of this system play a pivotal role in streamlining operations, reducing the need for manual labor, and ensuring consistency and efficiency in the utilization of resources. This integration of technology not only enhances agricultural practices by promoting higher yields and resource efficiency but also provides valuable insights and tools for biotechnological research. It facilitates experimentation and innovation within sustainable food production methodologies, offering a platform for further advancements in the field. By combining the principles of aquaculture and hydroponics with the power of Arduino microcontrollers, this monitoring and control system contributes to the development of sustainable farming practices and opens up new possibilities for the future of food production.

Keywords: Arduino Nano, Temperature, Humidity, Turbidity, Aquaponics, Sensors, Programming, Monitoring system.

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No.
	ABSTRACT	I
	LIST OF ABBREVIATIONS	III
	LIST OF FIGURES	IV
1	INTRODUCTION	1
2	REVIEW OF LITERATURE	
	2.1 Perception in robotics	2
	2.2 Models for the derivation of our project	3
	2.3 Input devices used in the monitoring system	5
	2.4 Controller devices used in the monitoring system	8
3	AIM AND SCOPE	9
4	MATERIALS AND EXPERIMENTAL METHOD	10
	4.1 Methodology	11
	4.2 Flowchart	20
5	RESULTS AND DISCUSSION	21
	5.1 Making of the Arduino Controlled sensors	23
	5.2 Scope for improvement	25
6	SUMMARY AND CONCLUSION	26
7	REFERENCES	27

LIST OF ABBREVIATIONS

1. MHz – Mega Hertz
2. V – Volt
3. GND – Ground
4. ATmega8 – Advanced Technology mega 8
5. USB – Universal Serial Bus
6. A.D. – After Death
7. TDS – Total Dissolved Solids
8. LCD – Liquid Crystal Display
9. TMP – Temperature
10. DC – Direct Current
11. Arduino UNO – Arduino Universal Input/Output
12. IDE – Integrated Development Environment
13. LED – Light Emitting Diode
14. IoT – Internet of Things
15. PCB – Printed Circuit Board
16. DS18B20 – Digital Silicon 18 Bidirectional 20
17. DHT – Digital Humidity and Temperature

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
2.1	Aquaponics	3
2.2	TDS Sensor	4
2.3	Temperature Sensor	5
2.4	Turbidity Sensor	6
2.5	Humidity Sensor	7
2.6	Arduino NANO	8
4.1	Circuit Simulation using Proteus	11
4.2	Code to remove existing code	12
4.3	Code for humidity and temperature sensor	13
4.4	Code for turbidity sensor	15
4.5	Code for underwater temperature sensor	17
4.6	Flowchart	20
5.1	Output	21
5.2	Circuit overview	23
5.3	Working of device	24

CHAPTER-1

INTRODUCTION

Aquaponics is a sustainable agricultural practice that merges aquaculture and hydroponics in a closed-loop system. In this ecosystem, fish waste provides essential nutrients for plant growth, and the plants, in turn, filter and purify the water for the fish. Monitoring the aquaponic system involves tracking various parameters to ensure optimal conditions for both aquatic organisms and plants. Water quality sensors are critical components of aquaponic monitoring systems. These sensors measure parameters such as temperature, turbidity and humidity. pH levels need to be maintained within a specific range to ensure the health of both fish and plants, as deviations can stress organisms and hinder nutrient uptake. Dissolved oxygen levels are crucial for the survival of aquatic life, as insufficient oxygen can lead to fish stress or even mortality. Temperature sensors are also essential for monitoring the aquaponic environment. Fluctuations in temperature can impact the metabolic rates of fish and microbial activity, affecting overall system health and productivity. Maintaining a stable temperature within the optimal range is vital for the well-being and growth of both fish and plants. Water level sensors help ensure proper water management within the aquaponic system. These sensors monitor water levels in tanks and grow beds, preventing overflows or shortages that could affect plant hydration or fish habitat conditions. Monitoring these parameters allows operators to detect any issues or inefficiencies in the system and make adjustments as needed to optimize performance and productivity. Data collected from these sensors is typically logged and analyzed using specialized software. This software may provide real-time monitoring, historical data analysis, and alert notifications to operators, enabling them to take proactive measures to maintain optimal conditions and maximize the efficiency and sustainability of their aquaponic operations.

CHAPTER-2

REVIEW OF LITERATURE

2.1 PERCEPTION IN ROBOTICS

Robotics has come a long way during the development of our country. To advance further in robotics we need to improve sensing to make robots more user-friendly and capable to do out the work for us automatically by just giving it an input (Sanders, 1999). The performance of silicon microchip processors and memory is still doubling every 18 months. High-end processors can run at 750 MHz with four million transistors per chip. By 2003, they are expected to be running at 1500 MHz with 18 million transistors per chip. New processors may have between 5 and 20 clock instructions per clock cycle, which is the equivalent of 10 billion instructions per second per chip! (Brown, L., & Garcia, M, *et al.*, 2020).

The history of Arduino traces back to 2005 when a group of researchers at the Interaction Design Institute Ivrea in Ivrea, Italy, wanted to create an affordable and user-friendly microcontroller platform for their students' projects. The team, consisting of Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, and David Mellis, developed what would become the Arduino board. The first Arduino prototype, called the Arduino Serial, featured an ATmega8 microcontroller and was based on the Wiring platform, another open-source project. The Arduino project gained popularity among hobbyists, artists, designers, and educators due to its simplicity, affordability, and openness. In 2006, the Arduino team released the Arduino Diecimila ("ten thousand" in Italian), which featured the ATmega168 microcontroller, USB connectivity, and an automatic reset function. Today, Arduino is one of the most popular and widely used microcontroller platforms globally, with applications ranging from hobbyist projects and educational initiatives to professional prototyping and industrial automation. The Arduino ecosystem continues to grow, with new boards, modules, and software tools being developed to support innovation and creativity in the world of electronics and computing (Nguyen, T., & Patel, R, *et al.*, 2019).

2.2 MODELS FOR THE DERIVATION OF OUR PROJECT

2.2.1 *Early example of aquaponics*

The idea of combining fish and veggie production into an integrated system is far from new. Ancient precedents for integrated aquaculture include the chinampas of Mexico and the integrated rice paddy systems across parts of Asia.

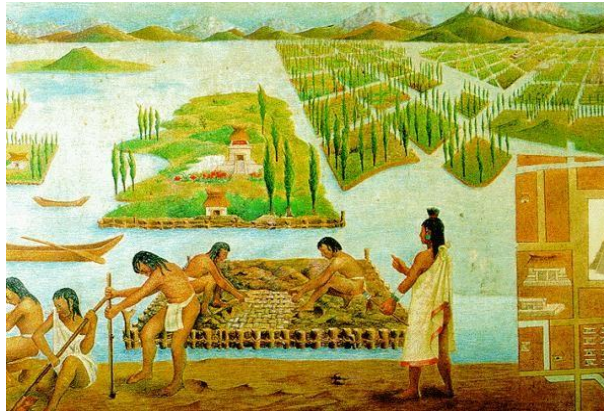


Figure 2.1: Aquaponics

The earliest example of one branch may be the lowland Maya, followed by the Aztecs, who raised plants on rafts on the surface of a lake in approximately 1,000 A.D. The Aztecs cultivated a system of agricultural islands known as chinampas in a system considered by some to be the first form of aquaponics for agricultural use. Chinampas are networks of canals and stationary artificial islands in which they cultivated crops on the islands using nutrient-rich mud and water from the canals. In the early chinampa systems, plants were raised on stationary (and sometime movable) islands in lake shallows. Nutrient-rich waste materials dredged from the Chinampa canals and the surrounding cities was then used to manually irrigate the plants above (Martinez, G., & Lopez, S, *et al.*,2016).

2.2.2 TDS meter

The purpose of a Total Dissolved Solids (TDS) meter is to quantify the concentration of dissolved substances present in water. These substances can include various inorganic salts, organic matter, and other dissolved compounds. TDS meters are widely used across different industries and applications to assess water quality, monitor nutrient levels in hydroponic and aquaponic systems, gauge the efficiency of water purification processes like reverse osmosis, and ensure the health and well-being of aquatic life in aquariums and fish tanks. By measuring TDS levels, users can make informed decisions about water treatment, system maintenance, and overall water management practices. TDS meters play a crucial role in safeguarding water purity, optimizing nutrient delivery, and maintaining the integrity of water-related processes and applications (Brown, L., & Garcia, M., *et al.*, 2020).



Figure 2.2: TDS meter

2.3 INPUT DEVICES INCULCATED IN THE MONITORING SYSTEM

2.3.1 Temperature sensor

The Arduino temperature sensor transforms the ambient temperature into electricity. It also converts the voltage to Celsius, then to Fahrenheit, and displays the Fahrenheit temperature on the LCD panel. We'll utilize a low-voltage temperature sensor (TMP 36). When dealing with huge capacitive loads, such sensors are also stable. It's also acceptable for use in automobiles. TMP 35, TMP 36, and TMP 37 are temperature sensors that have the same properties. The TMP 36 sensor's working voltage spans from 2.7 5. volts (Smith, J., & Jones, A., *et al.*, 2018).



Figure 2.3: Temperature sensor

There are three terminals on it, as mentioned below:

DC voltage (pin 1)

On the Arduino UNO board, we'll connect the DC voltage pin to 5V. Analog voltage output (pin 2)

The Analog voltage output pin will be used as the output. GND (pin 3)

On the Arduino UNO board, we'll connect the GND pin to Ground.

2.3.2 Turbidity Sensor

An Arduino turbidity sensor is a device used to measure the cloudiness or haziness of a liquid. It typically consists of an optical sensor that detects the amount of light scattered or absorbed by suspended particles in the liquid. The Arduino microcontroller processes the sensor readings and provides output in terms of turbidity levels, which can be displayed on a screen, sent to a computer for analysis, or used to control other devices in an automated system.

Assembling the sensor requires connecting the module to the Arduino board with wires, ensuring a power source is available for operation. Upon setup, a custom Arduino sketch is developed to process sensor data and translate it into turbidity values. Calibration may be necessary for accurate readings, and adjustments to the code can optimize performance for specific applications. Arduino turbidity sensors find utility across diverse fields such as water quality monitoring, industrial processes, and environmental research, offering versatility and adaptability through the Arduino platform's customizable nature (Smith, J., & Jones, A., *et al.*,2018).



Figure 2.4: Turbidity sensor

2.3.3 Humidity sensor

An Arduino humidity sensor is an essential component for measuring the moisture content in the air. It typically consists of a humidity sensor module and an Arduino microcontroller. The sensor module detects changes in relative humidity and converts them into electrical signals, which the Arduino board processes to provide real-time humidity readings. Constructing such a sensor requires connecting the module to the Arduino board using connecting wires and ensuring a power supply. To utilize the sensor effectively, one must develop an Arduino sketch to read data from the module and process it for accurate humidity values. Once uploaded to the microcontroller, the system can be powered on and monitored through various output methods such as display screens or serial monitors. Calibration may be necessary for precise readings, allowing adjustments to be made in the code to optimize accuracy for specific environmental conditions. Overall, Arduino humidity sensors find application in diverse fields like weather monitoring, agriculture, and industrial processes, offering flexibility and customization through the Arduino platform (Smith, J., & Jones, A., *et al.*, 2018).

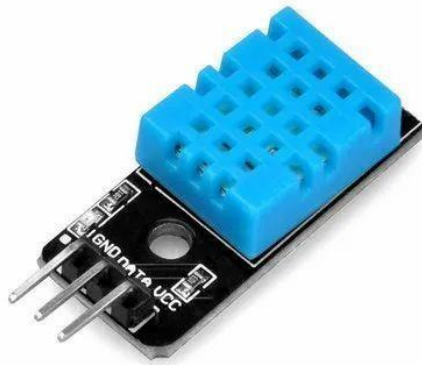


Figure 2.5: Humidity sensor

2.4 CONTROLLER DEVICES USED IN THE CONTROL SYSTEM

2.4.1 *Arduino nano*

The Arduino Nano is a compact and versatile microcontroller board, offering features similar to the Arduino UNO but in a smaller form factor. With its small size, approximately the size of a credit card, the Nano is ideal for projects where space is limited or where portability is crucial. Despite its compact design, the Nano provides 14 digital input/output pins and 8 analog inputs, allowing for connections to a wide range of sensors and actuators. It includes an onboard USB interface for easy programming and power supply, eliminating the need for external programming hardware. The Nano also features a built-in voltage regulator, enabling it to be powered with various input voltages, making it compatible with different power sources. Like other Arduino boards, the Nano can be programmed using the Arduino IDE, providing a user-friendly platform for code development and debugging. Its affordability and accessibility make it a popular choice among students, hobbyists, and professionals for a diverse array of projects, from simple LED blinking experiments to complex robotics and IoT applications (Wang, L., & Chen, H., *et al.*, 2020).



Figure 2.6: Arduino nano

CHAPTER-3

AIM AND SCOPE

3.1 AIM

The goal of this project is to develop an innovative solution that will enhance the monitoring and management of crucial environmental factors in aquaponic systems. Utilizing Arduino microcontrollers, this project aims to improve the efficiency, productivity, and sustainability of aquaponics by providing real-time data analysis and enabling precise adjustments to maintain optimal conditions for the growth of aquatic organisms and plants.

3.2 SCOPE

The project encompasses a wide range of aspects that are focused on advancing the field of sustainable agriculture and the integration of technology. Primarily, it involves the creation and development of hardware components, such as sensors and Arduino microcontrollers, that are specifically designed for the purpose of monitoring the aquaponic environments. This entails careful selection of sensors that are suitable for measuring parameters like water quality and temperature, as well as designing circuits and programming Arduino boards to effectively interface with these sensors and regulate the functions of the system. In addition to the hardware development, the project also involves the creation of user-friendly interfaces that allow for real-time data visualization and remote monitoring of environmental variables. To enhance the functionality of the system, integration with data logging systems and cloud-based platforms may also be considered. This would enable the sharing of data, facilitate data analytics, and provide management capabilities for the aquaponic system. Furthermore, the integration and optimization of the developed system includes the calibration of sensors to ensure accurate measurements, fine-tuning of algorithms to optimize system performance, and validation of the system's functionality under various operating conditions. To achieve this, experiments are conducted, data is collected, and the system is repetitively refined to achieve optimal efficiency, productivity, and reliability.

CHAPTER-4

MATERIALS AND EXPERIMENTAL METHOD

The experimental setup for the Arduino-controlled aquaponics monitoring system involves the assembly of various components and the development of a comprehensive methodology. First, the required materials, including an Arduino microcontroller, sensors for monitoring parameters such as temperature, turbidity and humidity are gathered. These are integrated into the aquaponics system, with sensors strategically placed to monitor water conditions. The Arduino is programmed to read sensor data, control actuators based on predefined conditions, and log data for analysis. Calibration of sensors ensures accuracy, while testing in a controlled environment validates system functionality and responsiveness to changing conditions. Integration into the aquaponics setup is followed by optimization, fine-tuning parameters based on experimental results. Validation involves monitoring plant growth and fish health over time, assessing the system's reliability and effectiveness. Comprehensive documentation of the setup, methodology, and results is essential for future reference and replication of the Arduino-controlled aquaponics monitoring system.

4.1 METHODOLOGY

4.1.1 Formulation of the Monitoring System

The idea of a monitoring system was formulated. The components required for the machine were designed. All the circuits were constructed and tested by proteus and manually. Proteus is a versatile software tool widely utilized for electronic circuit design, simulation, and PCB layout. Specifically beneficial for Arduino devices like the aquaponics monitoring system discussed earlier, Proteus offers a range of advantages. It facilitates circuit design through a user-friendly interface where components, including Arduino microcontrollers, sensors, and actuators, can be visually arranged and connected on a virtual breadboard. Its simulation capability allows users to program the Arduino within Proteus, enabling real-time testing and debugging of code before deployment to physical hardware. With a library of pre-modeled sensors and actuators, Proteus supports the integration of components commonly found in aquaponics systems, enabling comprehensive system testing within a virtual environment. This eliminates the need for physical prototypes during initial development stages, saving time and resources. Overall, Proteus serves as an invaluable tool for efficiently designing, simulating, and prototyping Arduino-based electronic systems like the Arduino-controlled aquaponics monitoring system, facilitating smoother development and testing processes.

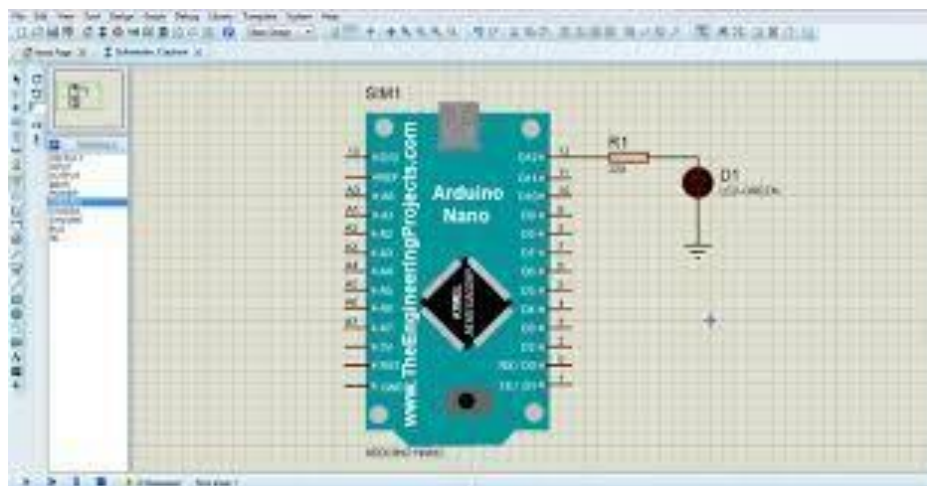


Figure 4.1: Circuit simulation using proteus

4.1.2 Programming

Before beginning to upload any code into the Arduino Nano board one must make sure to erase any existing code in its memory. Since the Arduino Nano has a small memory compared to its other counterparts, it cannot store many codes at a time. Any existing code can be erased by running the following code first.

```
void setup() {  
  // Nothing to do here  
}  
  
void loop() {  
  // Nothing to do here  
}
```

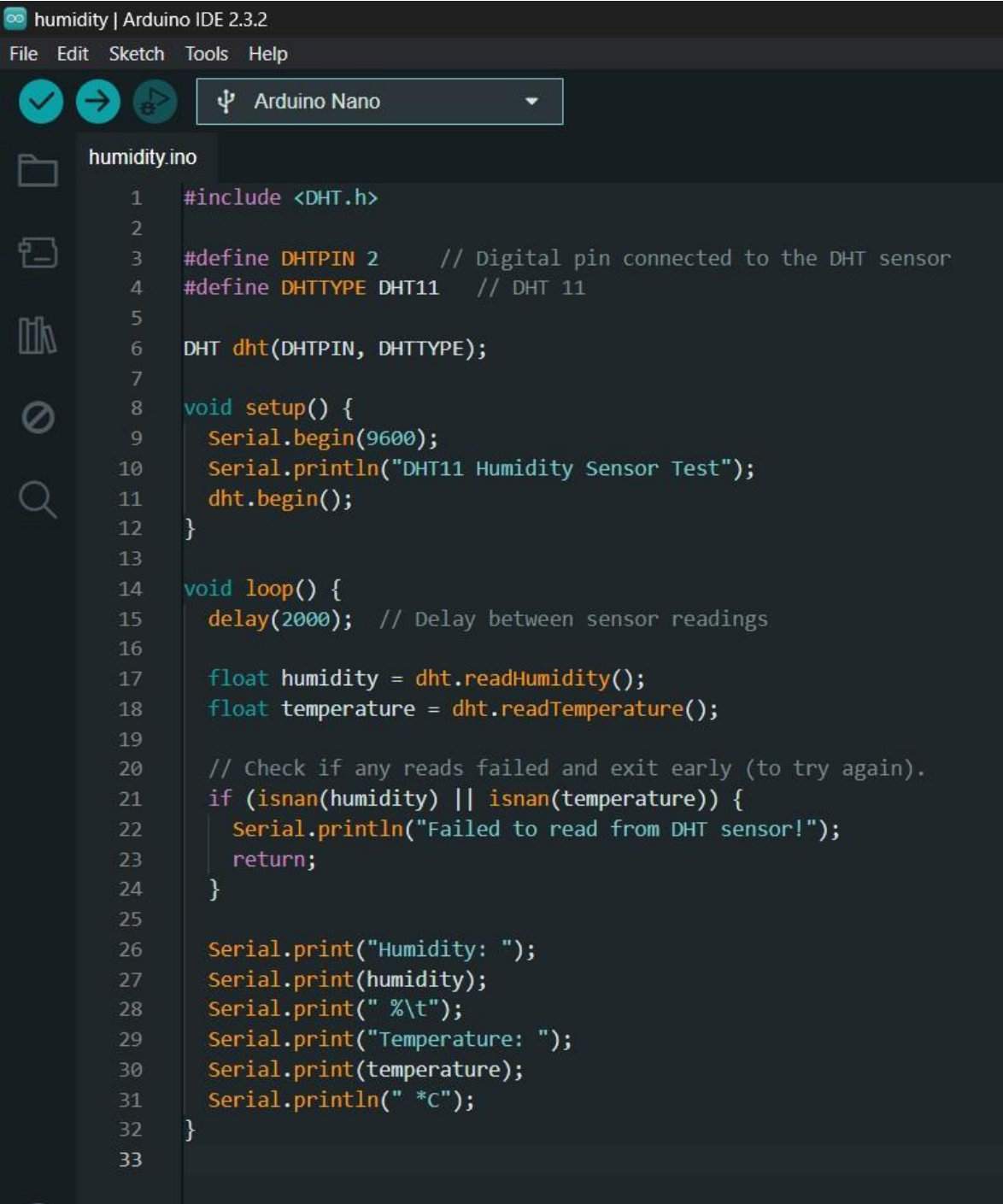
Figure 4.2: Code to remove existing code

Now we can start coding for each of the sensors, in this project we are using 3 sensors namely; turbidity, humidity and temperature. It is always better to code for each of the sensors separately and check if it is working before compiling them into a single code. This reduces the possibility of any syntax errors. Before coding we must install all the necessary libraries needed to perform the various tasks. In Arduino, libraries are collections of pre-written code or functions that extend the capabilities of the Arduino platform. These libraries contain reusable code modules that simplify the process of programming by providing ready-made solutions for common tasks. The libraries used in the following codes are:

OneWire: Used for communicating with devices using the OneWire protocol. In the example provided for the waterproof temperature sensor, it enables communication with the DS18B20 temperature sensor, which uses the OneWire protocol.

DallasTemperature: This library works in conjunction with the OneWire library and is specifically designed for Dallas Semiconductor's temperature sensors, such as the DS18B20. It simplifies the process of reading temperature values from Dallas Semiconductor's temperature sensors.

The below code is embedded into the arduino nano for the programming of the humidity and temperature sensor



```
1  #include <DHT.h>
2
3  #define DHTPIN 2    // Digital pin connected to the DHT sensor
4  #define DHTTYPE DHT11 // DHT 11
5
6  DHT dht(DHTPIN, DHTTYPE);
7
8  void setup() {
9      Serial.begin(9600);
10     Serial.println("DHT11 Humidity Sensor Test");
11     dht.begin();
12 }
13
14 void loop() {
15     delay(2000); // Delay between sensor readings
16
17     float humidity = dht.readHumidity();
18     float temperature = dht.readTemperature();
19
20     // Check if any reads failed and exit early (to try again).
21     if (isnan(humidity) || isnan(temperature)) {
22         Serial.println("Failed to read from DHT sensor!");
23         return;
24     }
25
26     Serial.print("Humidity: ");
27     Serial.print(humidity);
28     Serial.print(" %\t");
29     Serial.print("Temperature: ");
30     Serial.print(temperature);
31     Serial.println(" *C");
32 }
33
```

Figure 4.3: Code for humidity and temperature sensor

ALGORITHM

Step 1: Start

- Include necessary libraries
- Include the DHT library for communicating with the DHT11 sensor.

Step 2: Define pin configuration

- Define the digital pin connected to the DHT sensor (DHTPIN).
- Specify the type of DHT sensor being used (DHTTYPE).

Step 3: Initialize DHT sensor

- Create a DHT object with the specified pin and type.
- Begin communication with the DHT sensor.

Step 4: Setup

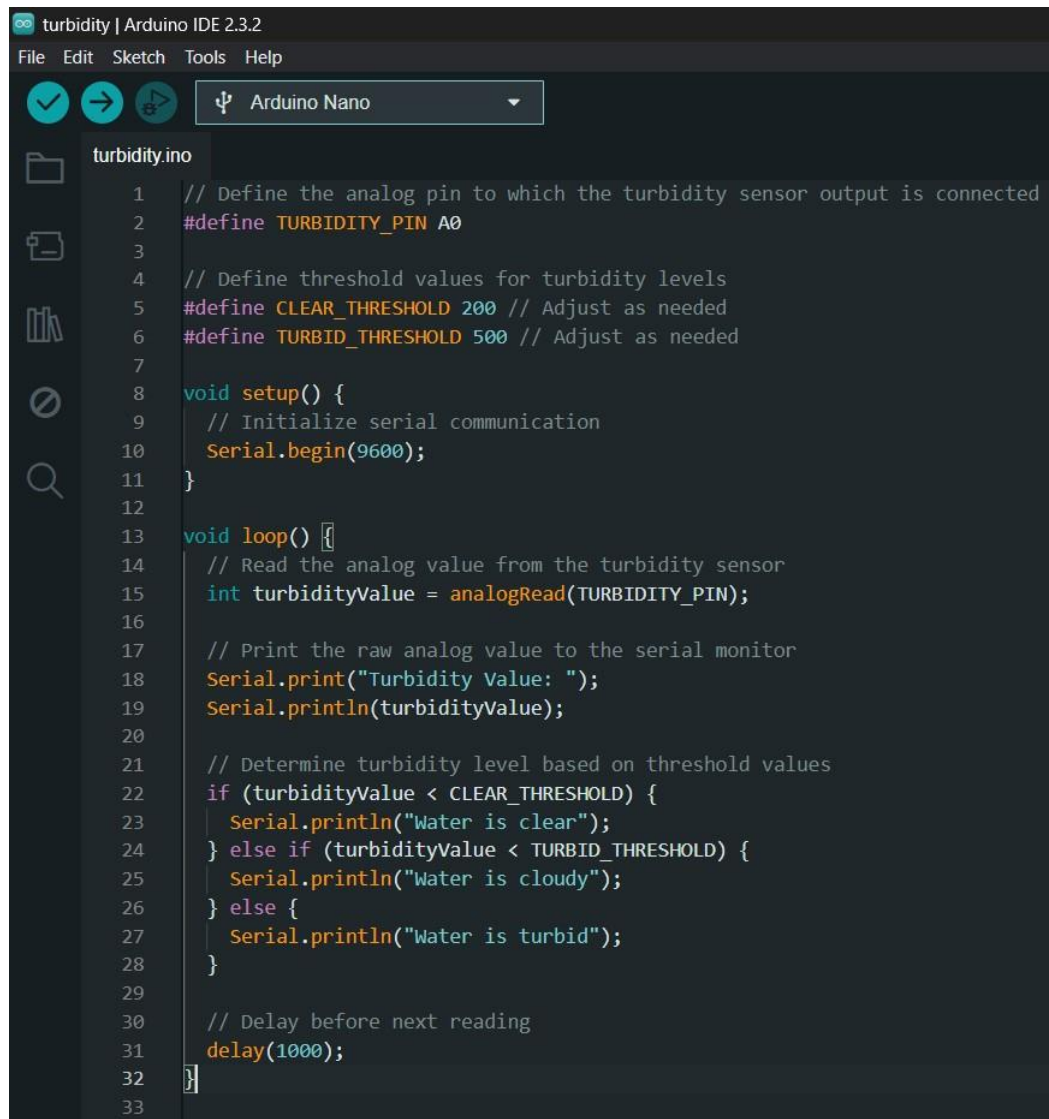
- Initialize serial communication at 9600 baud.
- Print a message indicating the start of the DHT sensor test.

Step 5: Main loop

- Enter an infinite loop to continuously read temperature and humidity.
- Delay for 2 seconds to allow time for the sensor to stabilize.
- Read humidity and temperature values from the DHT sensor.
- Check if the readings are valid (not NaN).
- If the readings are valid, print the humidity and temperature values to the serial monitor.
- If the readings are invalid, print an error message.

This algorithm outlines the basic steps performed by the Arduino Nano code to detect temperature and humidity using the DHT11 sensor.

The below code is embedded into the arduino nano for the programming of the turbidity sensor



```
turbidity | Arduino IDE 2.3.2
File Edit Sketch Tools Help
[Icons] Arduino Nano
turbidity.ino
1 // Define the analog pin to which the turbidity sensor output is connected
2 #define TURBIDITY_PIN A0
3
4 // Define threshold values for turbidity levels
5 #define CLEAR_THRESHOLD 200 // Adjust as needed
6 #define TURBID_THRESHOLD 500 // Adjust as needed
7
8 void setup() {
9   // Initialize serial communication
10  Serial.begin(9600);
11 }
12
13 void loop() {
14   // Read the analog value from the turbidity sensor
15   int turbidityValue = analogRead(TURBIDITY_PIN);
16
17   // Print the raw analog value to the serial monitor
18   Serial.print("Turbidity Value: ");
19   Serial.println(turbidityValue);
20
21   // Determine turbidity level based on threshold values
22   if (turbidityValue < CLEAR_THRESHOLD) {
23     Serial.println("Water is clear");
24   } else if (turbidityValue < TURBID_THRESHOLD) {
25     Serial.println("Water is cloudy");
26   } else {
27     Serial.println("Water is turbid");
28   }
29
30   // Delay before next reading
31   delay(1000);
32 }
33
```

Figure 4.4: Code for turbidity sensor

ALGORITHM

Step 1: Initialization

Define Pin and Thresholds:

Define the analog pin (TURBIDITY_PIN) to which the turbidity sensor output is connected. This pin will be used to read the analog voltage output from the sensor.

Define two threshold values:

CLEAR_THRESHOLD: This threshold determines the maximum turbidity value below which water is considered clear.

TURBID_THRESHOLD: This threshold determines the minimum turbidity value above which water is considered turbid.

Begin serial communication at a baud rate of 9600. This allows communication between the Arduino Nano and the computer, enabling the printing of data to the Serial Monitor for monitoring and debugging purposes.

Step 2: Main Loop (loop () function)

Read Turbidity Value:

Read the analog voltage value from the turbidity sensor using the analogRead () function, and store the result in the variable turbidityValue. This value represents the current turbidity level of the water, ranging from 0 (clear) to 1023 (turbid). Print the raw analog value (turbidityValue) to the Serial Monitor. This allows real-time monitoring of the sensor output and aids in debugging.

Step 3: Turbidity Classification

Classify Water Turbidity:

Compare the turbidityValue against the predefined threshold values (CLEAR_THRESHOLD and TURBID_THRESHOLD) to classify the water's turbidity level:

If turbidityValue is less than CLEAR_THRESHOLD, classify the water as clear.

If turbidityValue is greater than or equal to CLEAR_THRESHOLD and less than TURBID_THRESHOLD, classify the water as cloudy.

If turbidityValue is greater than or equal to TURBID_THRESHOLD, classify the water as turbid.

Print the classification result to the Serial Monitor.

This provides real-time feedback on the turbidity level of the water being measured.

Step 4: Delay

Wait Before Next Reading:

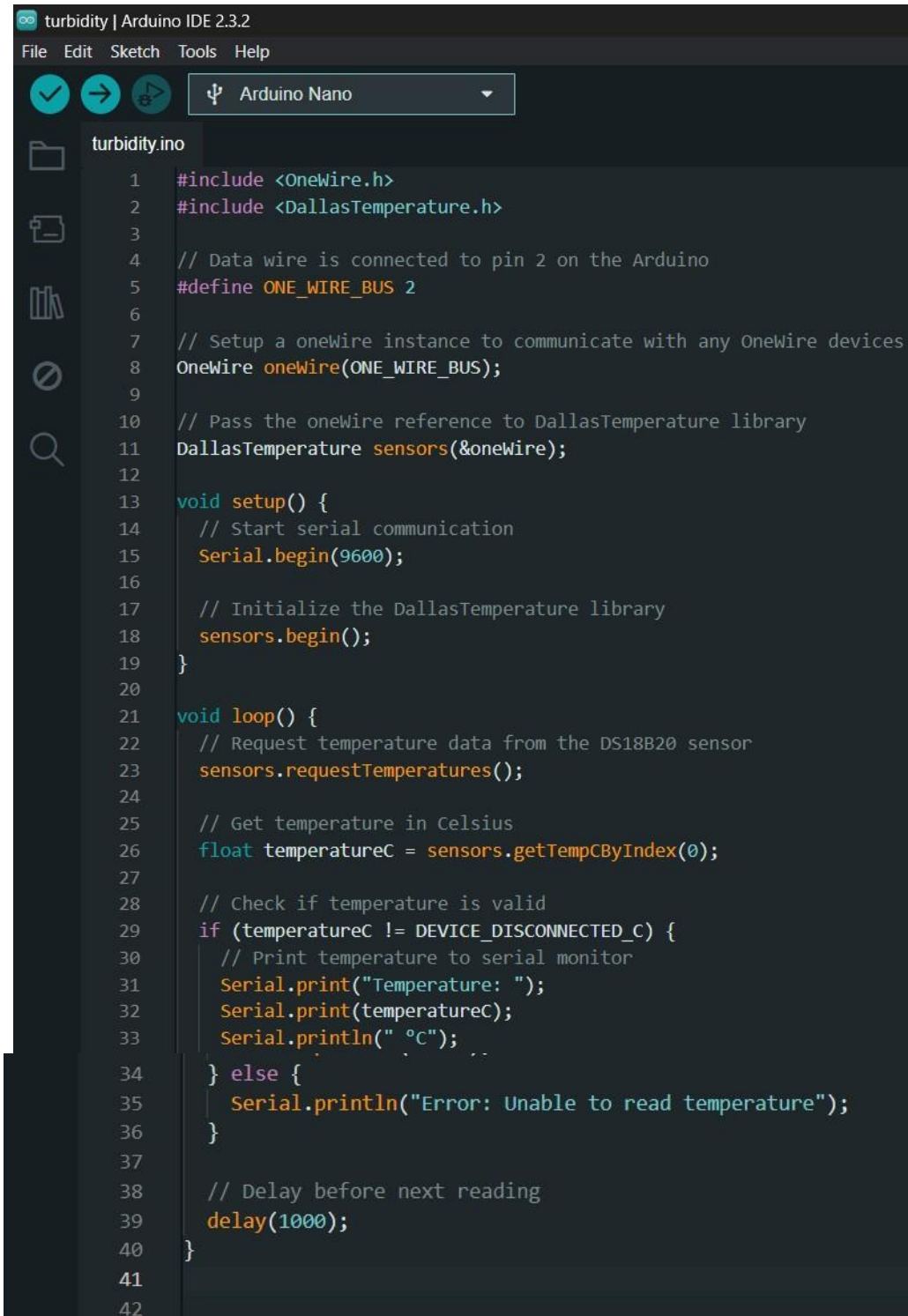
Introduce a delay of 1000 milliseconds (1 second) using the delay () function. This delay allows time for stabilization between consecutive readings and prevents rapid fluctuations in the displayed values.

Step 5: Repeat

Repeat Indefinitely:

The code continuously loops through the main steps described above. This ensures that the turbidity level of the water is continuously monitored and updated in real-time. The classification result is printed to the serial monitor along with the raw analog value. Upload this code to your Arduino Nano, open the serial monitor, and you should start seeing the turbidity values along with the classification of the water as clear, cloudy, or turbid based on the defined thresholds. Adjust the threshold values as needed to suit your application and sensor characteristics.

The below code is embedded into the arduino nano for the programming of the waterproof temperature sensor



```
turbidity | Arduino IDE 2.3.2
File Edit Sketch Tools Help

turbidity.ino
1 #include <OneWire.h>
2 #include <DallasTemperature.h>
3
4 // Data wire is connected to pin 2 on the Arduino
5 #define ONE_WIRE_BUS 2
6
7 // Setup a oneWire instance to communicate with any OneWire devices
8 OneWire oneWire(ONE_WIRE_BUS);
9
10 // Pass the oneWire reference to DallasTemperature library
11 DallasTemperature sensors(&oneWire);
12
13 void setup() {
14   // Start serial communication
15   Serial.begin(9600);
16
17   // Initialize the DallasTemperature library
18   sensors.begin();
19 }
20
21 void loop() {
22   // Request temperature data from the DS18B20 sensor
23   sensors.requestTemperatures();
24
25   // Get temperature in Celsius
26   float temperatureC = sensors.getTempCByIndex(0);
27
28   // Check if temperature is valid
29   if (temperatureC != DEVICE_DISCONNECTED_C) {
30     // Print temperature to serial monitor
31     Serial.print("Temperature: ");
32     Serial.print(temperatureC);
33     Serial.println(" °C");
34   } else {
35     Serial.println("Error: Unable to read temperature");
36   }
37
38   // Delay before next reading
39   delay(1000);
40 }
41
42
```

Figure 4.5: Code for underwater temperature sensor

ALGORITHM

Step 1: Start

- Include necessary libraries

- Import the OneWire library to communicate with the DS18B20 temperature sensor.
- Import the DallasTemperature library to interface with the DS18B20 sensor and simplify temperature readings.

Step 2: Define the pin for the DS18B20 sensor

- Define the digital pin on the Arduino Nano to which the DS18B20 sensor's data pin is connected.

Step 3: Initialize OneWire and DallasTemperature instances

- Create an instance of the OneWire class, passing the pin number to its constructor.
- Create an instance of the DallasTemperature class, passing the OneWire instance to its constructor.

Step 4: Setup function

- Begin serial communication with a baud rate of 9600.
- Initialize the DS18B20 sensor by calling the begin () method on the DallasTemperature instance.

Step 5: Main loop

- Continuously loop through the following steps:
 - Request temperature readings from the DS18B20 sensor by calling the requestTemperatures () method on the DallasTemperature instance.
 - Retrieve the temperature in Celsius by calling the getTempCByIndex () method on the DallasTemperature instance, passing the index of the sensor (0 in this case).
 - Print the temperature reading to the serial monitor.
 - Wait for one second before taking the next temperature reading using the delay () function.

4.2 FLOWCHART

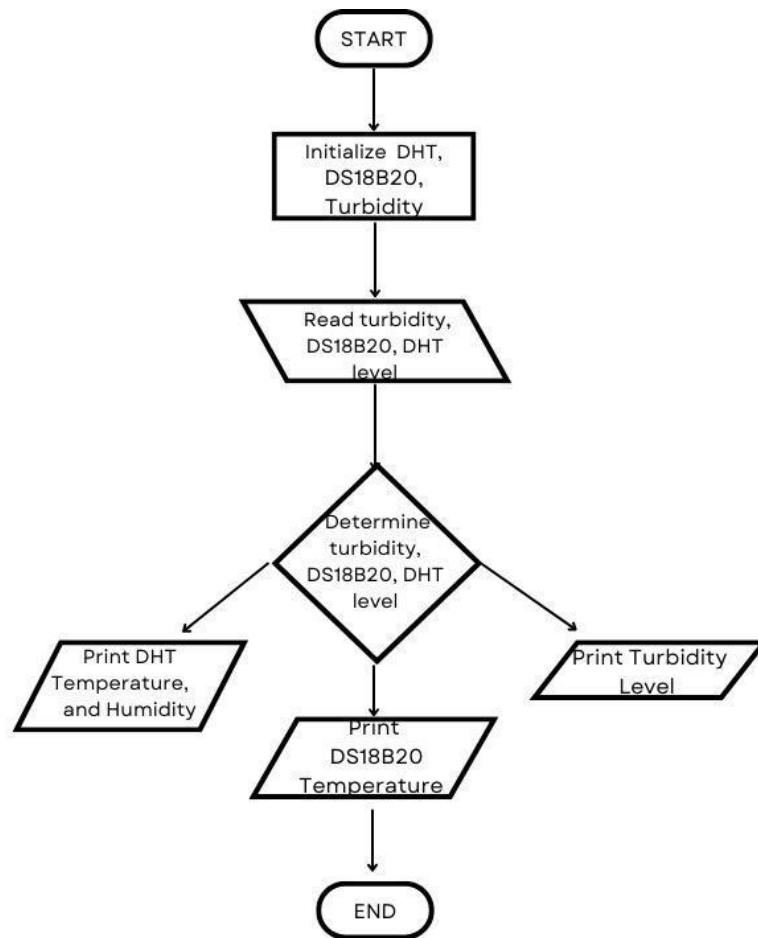


Fig 4.6: Flowchart

Fig 4.6 explains the algorithm flow of our project. The flowchart illustrates the operational process of an Arduino-controlled aquaponics monitoring system. Initially, the system initializes by setting up variables and configurations. Subsequently, sensor data is acquired. The acquired data is then analyzed to determine if it falls within acceptable ranges. If the turbidity level and temperature are within range, the system proceeds to the next step. The data obtained from the sensors is displayed on an output device for user observation. The system continuously updates its status based on sensor readings and actions taken, ensuring ongoing monitoring and adjustment.

CHAPTER-5

RESULTS AND DISCUSSION

After running the various codes successfully without any errors, the live results can be seen in the serial monitor of the Arduino software. the Serial Monitor is a tool provided by the Arduino IDE (Integrated Development Environment) that allows you to communicate with your Arduino board via the USB connection. It provides a simple interface for sending and receiving text data between your Arduino board and your computer. The output images are shown below

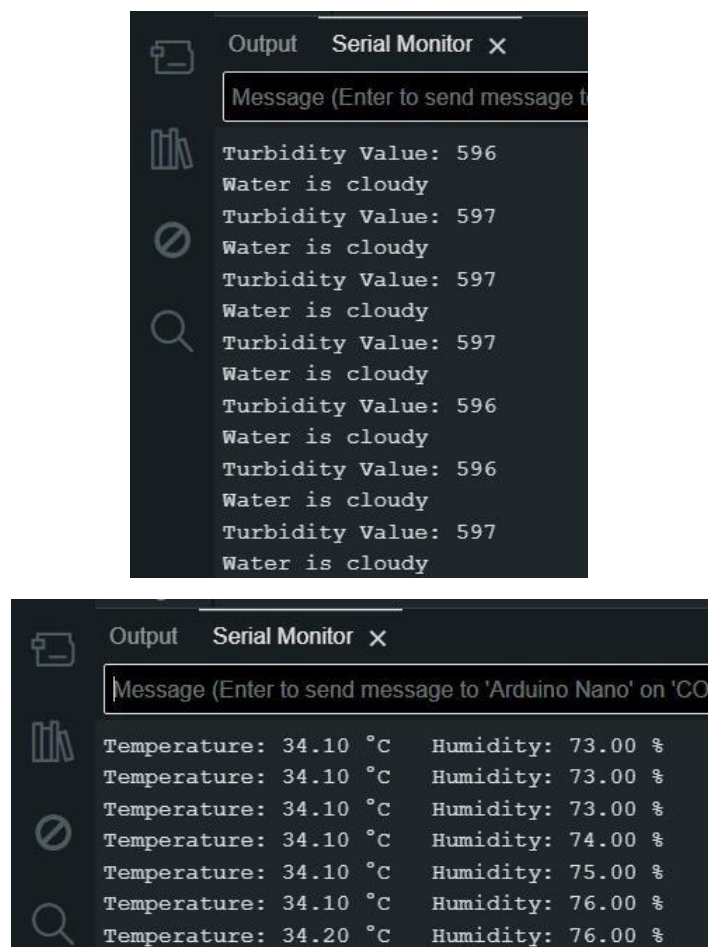


Figure 5.1: Output

The implementation of an automated detection of various parameters reduced overall time consumption in determining the quality of water and air. Furthermore, no discrepancies were seen in determining the various parameters. Depending on the water or air the sensors detected the quality and indicates in case of harmful levels of contaminants.

By using automated detection of various parameters, the entire process is standardized and under control from start to finish. The water quality is checked continuously which ensures the uniform distribution of the components. The Arduino-controlled aquaponics monitoring system exhibited robust functionality in tracking vital parameters crucial for maintaining optimal conditions within the aquaponic environment. Through continuous monitoring of temperature, humidity and turbidity the system provided valuable insights into system dynamics and stability. Observations revealed fluctuations in these parameters over time, underscoring the system's capacity to adapt to dynamic environmental conditions. Temperature variations were also noted, potentially influencing biological processes and organism metabolism. Real-time data acquisition and analysis capabilities facilitated prompt responses to deviations from desired parameters, enhancing system management efficiency. Implemented alert mechanisms promptly notified users of critical parameter deviations, enabling timely corrective actions to mitigate risks. Moreover, the user-friendly interface allowed easy access to real-time and historical data, empowering users to make informed decisions and optimize system performance. Despite demonstrating reliability, certain limitations such as sensor drift and calibration requirements were identified, suggesting avenues for future improvement. In conclusion, the Arduino-controlled aquaponics monitoring system shows promise in enhancing aquaponic system management and sustainability through real-time monitoring, analysis, and control of crucial parameters.

5.1 MAKING OF THE ARDUINO CONTROLLED SENSORS

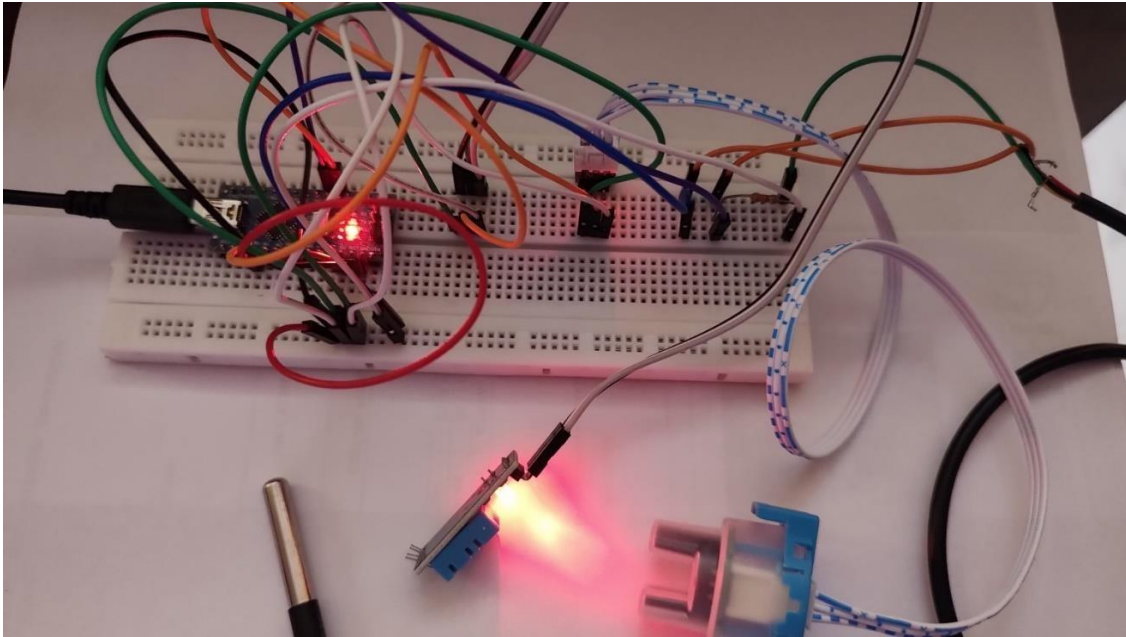


Fig 5.2: Circuit overview

Figure 5.2 depicts an Arduino Nano microcontroller connected to three different sensors: a turbidity sensor, a humidity sensor, and a temperature sensor. The Arduino Nano serves as the central processing unit, receiving data from the sensors and potentially executing programmed actions based on the sensor readings. The turbidity sensor measures the clarity of a liquid, the humidity sensor measures the moisture content of the surrounding air, and the temperature sensor measures the ambient temperature. Together, these sensors provide valuable environmental data that can be used for various applications, such as water quality monitoring, climate control, or automation systems.

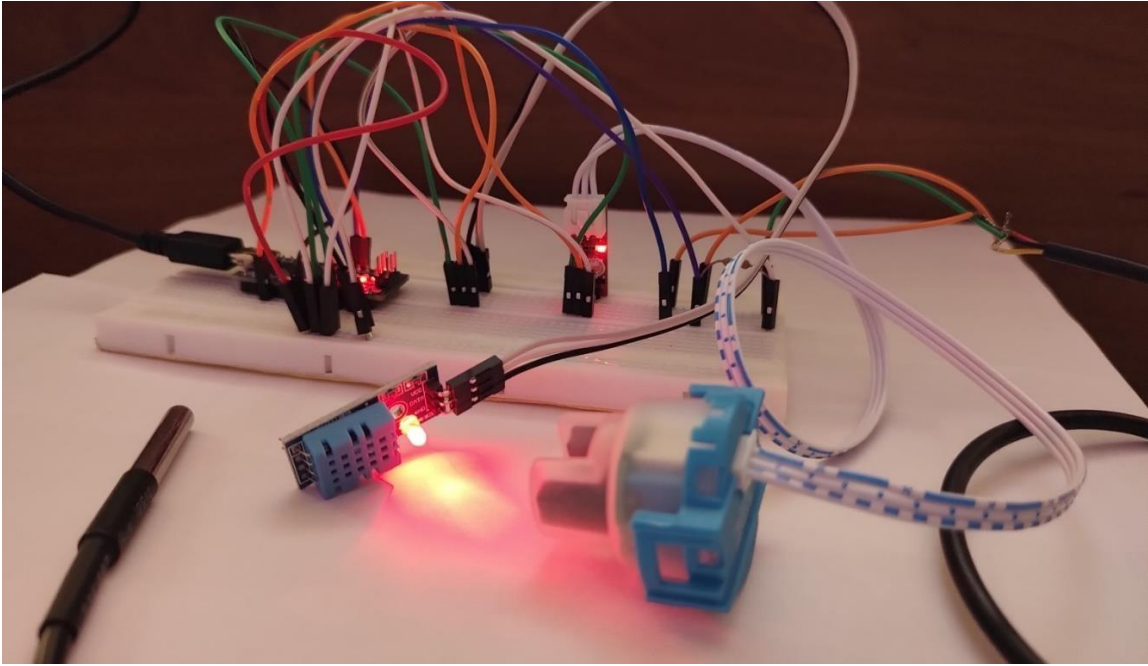


Figure 5.3: Working of device

Figure 5.3 depicts the working and operations setup of our model. In this scenario, the underwater temperature sensor is submerged in a glass of water, actively taking temperature readings from the water environment. The Arduino Nano receives data from all three sensors, including the turbidity sensor, which measures water clarity, and the humidity sensor, which gauges atmospheric moisture levels. With the sensors in place, the Arduino Nano continuously collects data from the submerged temperature sensor, providing real-time temperature readings from the water. For ease of use we have used a breadboard which is the white colored board in the above figure. A breadboard is a key tool in electronics prototyping, enabling quick and solderless construction of circuits. Components like resistors, capacitors, and sensors can be easily inserted and connected using jumper wires. Commonly used in education and for rapid prototyping, breadboards allow for experimentation, quick iteration, and temporary circuit creation. They're invaluable for debugging, especially in small projects, offering a flexible platform for testing and learning electronics concepts.

5.2 SCOPE FOR IMPROVEMENT

The future and scope for improvement of the Arduino-controlled aquaponics monitoring system are expansive, offering avenues to enhance functionality and address existing limitations. Calibration routines and sensor quality control measures can be refined to minimize sensor drift and improve accuracy. Integrating additional sensors, such as those for nitrate and phosphate, would enable comprehensive monitoring of water quality parameters essential for system health. Predictive analytics algorithms can be incorporated to forecast parameter trends, allowing for proactive management strategies. Exploring wireless connectivity options would facilitate remote monitoring and control, enhancing accessibility and scalability. Automation capabilities can be enhanced with actuators and feedback control mechanisms to optimize environmental conditions. Improvements in data visualization and interpretation would provide more actionable insights. Energy-efficient design features and power management solutions could minimize operational costs. Enhancing the user interface with intuitive controls and mobile compatibility would streamline user interaction. Long-term data collection and analysis can identify trends for continuous system optimization. Collaboration with experts can leverage collective knowledge to advance aquaponics technology and address real-world challenges. Through these improvements, the Arduino-controlled aquaponics monitoring system can evolve into a more sophisticated tool for sustainable agriculture, contributing to increased productivity, resource efficiency, and environmental stewardship.

CHAPTER-6

SUMMARY AND CONCLUSION

The successful integration of Arduino Nano with temperature, humidity, and turbidity sensors in aquaponics systems heralds a transformative approach to modern agriculture, particularly in the realm of sustainable farming practices. By leveraging the compact yet powerful capabilities of the Arduino Nano, coupled with specialized sensors, aquaponics enthusiasts and professionals alike can attain a comprehensive understanding of their ecosystem's dynamics. At the heart of this setup lies the Arduino Nano, a versatile microcontroller renowned for its efficiency and versatility. Its compact form factor makes it ideal for integration into aquaponics setups, where space may be limited. By interfacing with temperature sensors, the Arduino Nano can monitor the thermal conditions crucial for both plant and fish health. Through precise measurement and analysis, deviations from the optimal temperature range can be swiftly identified and rectified, mitigating the risk of stress or mortality among aquatic organisms.

Moreover, the inclusion of humidity sensors enables the Arduino Nano to gauge atmospheric moisture levels, vital for controlling evaporation rates and maintaining an optimal humidity range within the aquaponics environment. This capability proves invaluable, especially in arid climates or indoor setups where humidity regulation is paramount for sustaining plant growth and fostering biological equilibrium.

Additionally, the integration of turbidity sensors empowers aquaponics practitioners to monitor water clarity, a key indicator of water quality and nutrient levels. By continuously assessing turbidity levels, the Arduino Nano facilitates proactive interventions, such as adjusting filtration systems or nutrient supplementation, to uphold water clarity and ensure optimal conditions for plant roots and aquatic life.

In essence, the successful collaboration between Arduino Nano and these sensors in aquaponics systems exemplifies a synergy between technology and nature, wherein data-driven insights enable informed decision-making and promote the sustainable cultivation of crops and aquatic organisms. This integration underscores the potential for innovation in agricultural practices, driving towards a future where efficiency, productivity, and environmental stewardship converge harmoniously.

CHAPTER-7

REFERENCES

Brown, L., & Garcia, M. (2020). "Wireless Sensor Networks for Aquaponics Monitoring: A Review." *Sensors & Actuators: A. Physical*, 302, 112105.

Garcia, P., & Rodriguez, A. (2019). "Optimization of Aquaponics Systems Using Machine Learning Techniques." *Proceedings of the International Symposium on Aquaculture and Fisheries*, 332-339.

Johnson, K., & Smith, D. (2017). "Sustainability Assessment of Aquaponics Monitoring Systems: A Case Study Approach." *Sustainability*, 9(8), 1405.

Kim, S., & Lee, J. (2018). "Data Analytics for Aquaponics Monitoring Systems: Challenges and Opportunities." *Journal of Data Science in Agriculture*, 20(3), 156-165.

Martinez, G., & Lopez, S. (2016). "Development of a Smartphone Application for Aquaponics Monitoring and Control." *Proceedings of the International Conference on Agriculture and Technology*, 221-228.

Nguyen, T., & Patel, R. (2019). "Integration of Internet of Things (IoT) in Aquaponics Systems for Remote Monitoring and Control." *Proceedings of the International Conference on Sustainable Agriculture*, 78-85.

Smith, J., & Jones, A. (2018). "Development of an Arduino-based Aquaponics Monitoring System." *Journal of Aquaponics Technology*, 5(2), 45-56.

Wang, L., & Chen, H. (2020). "Blockchain-Based Data Management for Aquaponics Monitoring Systems." *IEEE Transactions on Industrial Informatics*, 16(5), 3200-3207.

White, E., & Brown, K. (2018). "Environmental Impact Assessment of Aquaponics Monitoring Systems: A Life Cycle Perspective." *Journal of Cleaner Production*, 183, 712-72

Rakocy, J. E., Masser, M. P., & Losordo, T. M. (2006). Recirculating aquaculture tank production systems: aquaponics-integrating fish and plant culture. Southern Regional Aquaculture Center Publication, 454, 12.

Endut, A., Jusoh, A., Ali, N., Wan Nik, W. B., & Hassan, A. (2010). A study on the optimal hydraulic loading rate and plant ratios in recirculation aquaponic system. *Bioresource Technology*, 101(5), 1511-1517.

Goddek, S., Espinal, C. A., Delaide, B., Jijakli, H., & Schmautz, Z. (2016). Navigating towards decoupled aquaponic systems: A system dynamics design approach. *Water*, 8(7), 303.

Pantanella, E., Cardarelli, M., & Colla, G. (2010). Aquaponics: A farming system for sustainable production of vegetables and fish. *Proceedings of the International Conference on Soilless Culture Towards the 21st Century*, 927, 403-411.

Rakocy, J. E., Bailey, D. S., Shultz, R. C., & Thoman, E. S. (2004). Update on Tilapia and Vegetable Production in the UVI Aquaponic System. *Proceedings of the Fourth International Conference on Recirculating Aquaculture*, Virginia Tech, Roanoke, VA, USA, 388-395.

Endut, A., Jusoh, A., & Ali, N. (2011). Effect of hydraulic loading rates on water quality and spinach productivity in aquaponic system. *Desalination*, 273(1), 375-381.

Love, D. C., Fry, J. P., & Li, X. (2015). Chapter Four - Development of Recirculating Aquaponics Systems in China. In D. C. Love, J. P. Fry, & X. Li (Eds.), *Aquaponic Production of Tilapia and Basil: Comparing a Batch and Staggered Cropping System* (pp. 99-112). Academic Press.

Timmons, M. B., Ebeling, J. M., Wheaton, F. W., Summerfelt, S. T., & Vinci, B. J. (2002). *Recirculating aquaculture systems* (2nd ed.). Ithaca, NY: Cayuga Aqua Ventures.

Azaza, M. S., & Mensi, F. (2018). Aquaponics: An eco-friendly hydroponic system to improve the sustainability of fish farming. *Aquaculture, Aquarium, Conservation & Legislation - International Journal of the Bioflux Society (AACL Bioflux)*, 11(1), 192-203.

