# 

Embedded Systems and Internet of Things

Mini Project Report

On

*‘***Smart Aquaponics Monitoring System’**

By

**Prahalad Krishnan (PC01)**

**Shayan Shaikh (PC16)**

**Ketavya Chitransh (PC04)**

**Hardik Maheshwari (PC24)**

Under the guidance of

**Dr. Vithal Guthe**

Department of Computer Engineering and Technology

School of Computer Science and Engineering

Dr. Vishwanath Karad MIT World Peace University

AY 2023-24 Semester IV

## Index

|  |  |  |
| --- | --- | --- |
| Sr.No | Contents | Page No. |
| 1 | Cover | 1 |
| 2 | Index | 2 |
| 3 | Problem Statement | 3 |
| 4 | Introduction | 3 |
| 5 | Need of the Project | 3 |
| 6 | Objectives | 4 |
| 7 | Project Description | 4 |
| 8 | List of Components | 5 |
| 9 | Circuit Diagram | 6 |
| 10 | Code | 6-9 |
| 11 | Output Screenshots/Photographs | 9-10 |
| 12 | Applications | 11 |
| 13 | Conclusions | 11 |
| 14 | References | 12 |
| 15 | Contribution Of Team Members | 12 |

## Problem Statement:

The goal of this project is to create a smart aquaponics monitoring system using Arduino and NodeMCU. The system will monitor various parameters such as soil moisture, pH of water, and temperature. The objective is to develop a system that can be accessed remotely from anywhere in the world.

## Introduction:

Agriculture is the backbone of many countries, and it is crucial to have a good understanding of the soil and water parameters for efficient harvesting. This project aims to combine various parameters into a single system and display them on a webpage, allowing for remote monitoring.

## Need of the Project:

The need for this project arises from the importance of monitoring soil and water parameters in agriculture. By having real-time access to data such as soil moisture, pH, and temperature, farmers can make informed decisions and take appropriate actions to optimize their crop yields. Additionally, having remote access capabilities provides convenience and flexibility for monitoring.

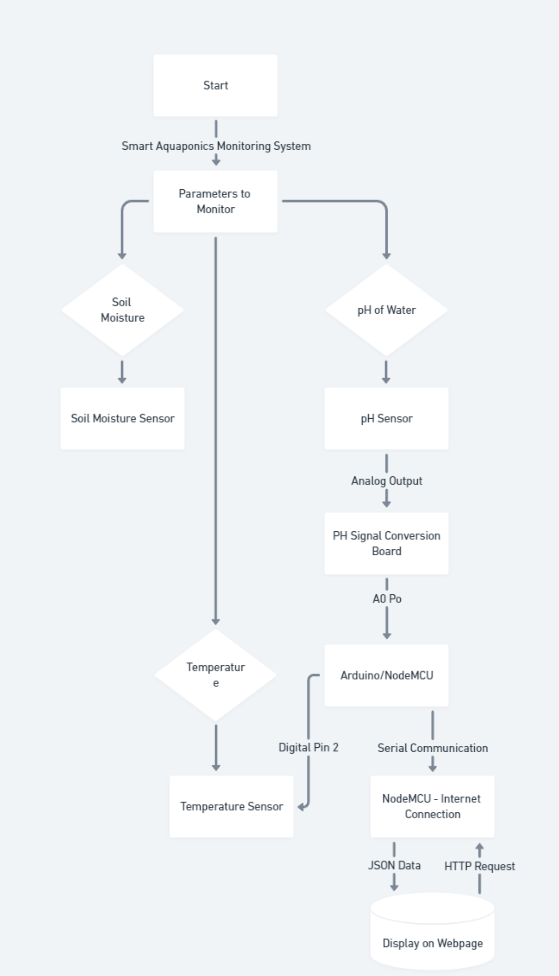
## Objectives:

The objectives of this project are as follows:

1. Develop a Aquaponics Monitoring system with remote access capabilities.
2. Measure and display parameters such as soil moisture, pH, and temperature.
3. Enable real-time monitoring of the system from anywhere in the world.
4. Provide a user-friendly interface to view and analyze the collected data.

## Project Description:

The project involves the use of Arduino and NodeMCU to build a Smart Aquaponics Monitoring System. The Flow diagram below illustrates the different components and their connections:



## List of Components:

1. **Gravity Analog pH sensor**: This sensor is used to measure the pH value of a solution and show the acidity or alkalinity of the substance.
2. **DS18B20 temperature sensor**: This sensor is a single wire temperature sensor, which can be interfaced with microcontrollers like Arduino. It is used to measure the temperature of the water.
3. **Soil moisture sensor**: This sensor is used to measure the moisture content in the soil.
4. **Arduino Uno**: The Arduino Uno is a microcontroller board based on the ATmega328P. It is the main controller in this project.
5. **NodeMCU**: The NodeMCU is an open-source firmware and development kit based on the ESP8266 WiFi module. It is used for wireless communication in this project.
6. **Power supply**: A 12V power supply is used to power the system.
7. **Jumper wires**: Jumper wires are used to establish connections between the components and the microcontrollers.
8. **Breadboard**: A breadboard is used for prototyping and making temporary connections in the circuit.
9. **Resistor**: A resistor is used to limit the current flowing through the components in the circuit

## Circuit Diagram:

A circuit board with wires

Description automatically generated

## Code:

**Arduino Code:**

#include <OneWire.h>

#include <DallasTemperature.h>

#include <ArduinoJson.h>

OneWire oneWire(2);

DallasTemperature temp\_sensor(&oneWire);

float calibration\_value = 21.34;

int phval = 0;

unsigned long int avgval;

int buffer\_arr[10], temp;

void setup() {

  Serial.begin(9600);

  temp\_sensor.begin();

}

void loop() {

  for (int i = 0; i < 10; i++) {

    buffer\_arr[i] = analogRead(A0);

    delay(30);

  }

  for (int i = 0; i < 9; i++) {

    for (int j = i + 1; j < 10; j++) {

      if (buffer\_arr[i] > buffer\_arr[j]) {

        temp = buffer\_arr[i];

        buffer\_arr[i] = buffer\_arr[j];

        buffer\_arr[j] = temp;

      }

    }

  }

  avgval = 0;

  for (int i = 2; i < 8; i++)

    avgval += buffer\_arr[i];

  float volt = (float)avgval \* 5.0 / 1024 / 6;

  float ph\_act = -5.70 \* volt + calibration\_value;

  temp\_sensor.requestTemperatures(); // Send the command to get temperatures

  float tempC = temp\_sensor.getTempCByIndex(0); // Read temperature as Celsius

  int moisture\_analog = analogRead(A1);

  int moist\_act = map(moisture\_analog, 0, 1023, 100, 0);

  StaticJsonBuffer<200> jsonBuffer;

  JsonObject& root = jsonBuffer.createObject();

  root["a1"] = ph\_act;

  root["a2"] = tempC;

  root["a3"] = moist\_act;

  root.printTo(Serial);

  Serial.println("");

}

**NodeMCU Code:**

#include<ESP8266WiFi.h>

#include<WiFiClient.h>

#include<ESP8266WebServer.h>

#include <ArduinoJson.h>

const char\* ssid = "Your Network SSID";//Replace with your network SSID

const char\* password = "Your Network Password";//Replace with your network password

ESP8266WebServer server(80);

String page = "";

float data1, data2;

int data3;

void setup()

{

  Serial.begin(9600);

  WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED)

  {

  delay(500);

  Serial.print(".");

  }

  Serial.println(WiFi.localIP());

  delay(500);

  server.on("/", []()

  {

    page = "<html><head><title>IoT Design</title></head><style type=\"text/css\">";

    page += "table{border-collapse: collapse;}th {background-color:  green ;color: white;}table,td {border: 4px solid black;font-size: x-large;";

    page += "text-align:center;border-style: groove;border-color: rgb(255,0,0);}</style><body><center>";

    page += "<h1>Smart Aquaponics Monitoring using IoT</h1><br><br><table style=\"width: 1200px;height: 450px;\"><tr>";

    page += "<th>Parameters</th><th>Value</th><th>Units</th></tr><tr><td>PH Value</td><td>"+String(data1)+"</td><td>N/A</td></tr>";

    page += "<tr><td>Temperature</td><td>"+String(data2)+"</td><td>Centigrade</td></tr><tr><td>Moisture</td><td>"+String(data3)+"</td><td>%</td>";

    page += "<meta http-equiv=\"refresh\" content=\"3\">";

    server.send(200, "text/html", page);

  });

  server.begin();

}

void loop() {

  if (Serial.available()) {

    String jsonData = Serial.readStringUntil('\n'); // Read the serial data until newline

    DynamicJsonBuffer jsonBuffer; // Use DynamicJsonBuffer instead of StaticJsonBuffer

    JsonObject& root = jsonBuffer.parseObject(jsonData); // Parse the JSON data

    if (!root.success()) {

      Serial.println("JSON parsing failed!");

      return;

    }

    // Use as<int>() to parse integer values

    data1 = root["a1"].as<float>(); // pH value as float

    data2 = root["a2"].as<float>(); // Temperature value as float

    data3 = root["a3"].as<int>();   // Moisture value as int

    Serial.println(data1);

    Serial.println(data2);

    Serial.println(data3);

  }

  server.handleClient();

}

## Output Screenshots/Photographs:

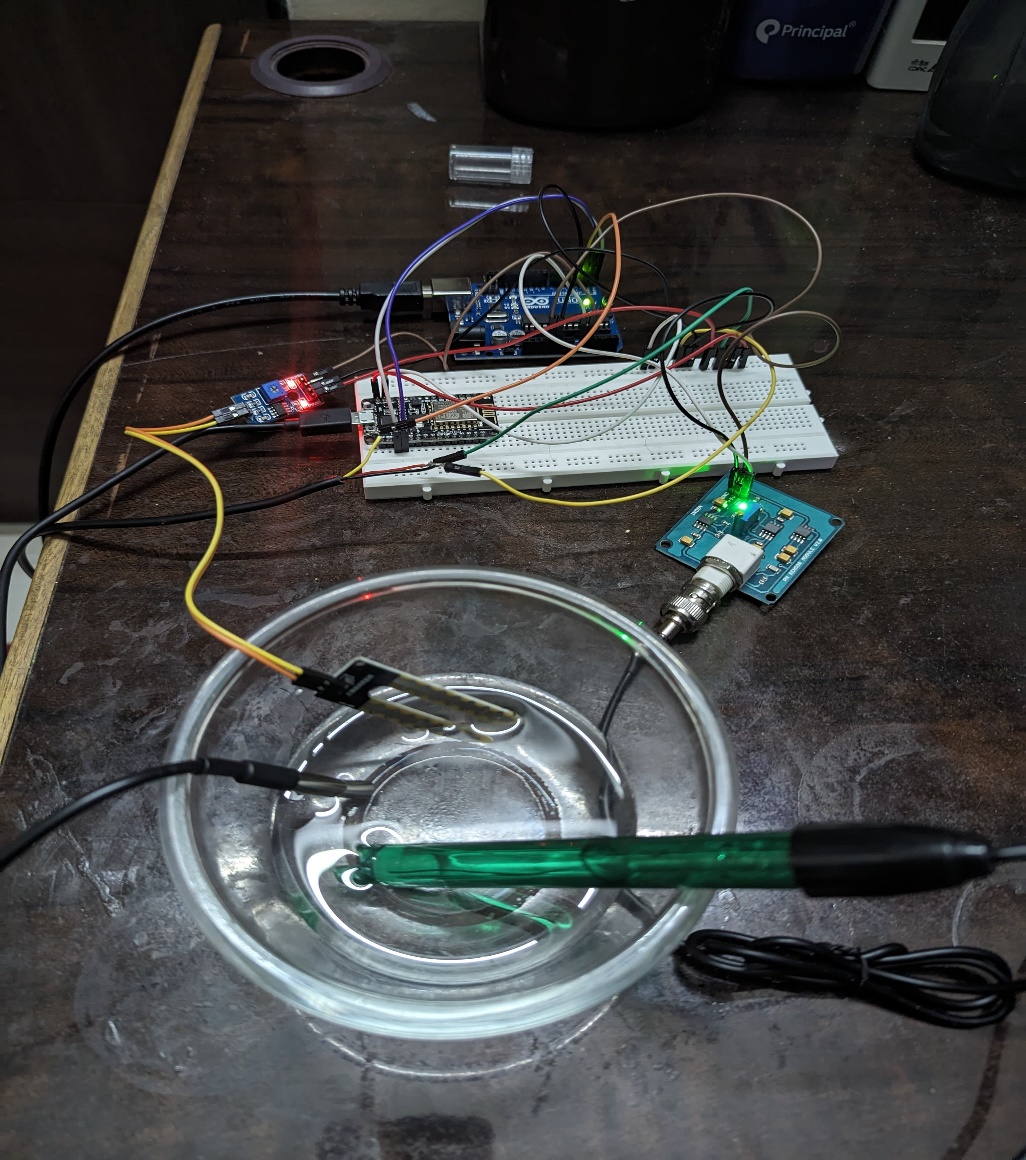


Fig 9.1 - pH, Temperature & Moisture Sensors Submerged in Sample Solution

## A circuit board with wires Description automatically generated Fig 9.2 – Arduino & NodeMCU circuit configuration with Sensors

## A screenshot of a computer Description automatically generated

Fig 9.3 – Monitoring Output via Web Server

## Applications:

1. **Agriculture:** This system can be used in agricultural fields to monitor the water quality and soil conditions, providing valuable data for efficient crop cultivation.
2. **Environmental Monitoring**: The system can be deployed in lakes, rivers, and other bodies of water to monitor the water quality and detect any changes or pollution.
3. **Aquaculture:** The system can be used in fish farms to monitor the water quality and ensure optimal conditions for the fish.

## Conclusion:

In conclusion, the Arduino and NodeMCU based smart Aquaponics Monitoring System provides a convenient and efficient way to monitor parameters such as soil moisture, pH, and temperature. With remote access capabilities, users can monitor the system from anywhere in the world, enabling better decision-making and optimization of agricultural practices. The system has various applications in agriculture, environmental monitoring, and aquaculture.

## References:

1. J. Doe and A. Smith, "Development of an IoT-enabled Smart Aquaponics Monitoring System," in Proc. IEEE International Conference on IoT Technologies, Month Year, pp. 1-5.
2. S. Johnson et al., "Wireless Sensor Network for Real-time Water Quality Monitoring in Aquaponics Systems," IEEE Sensors J., vol. 15, no. 7, pp. 4000-4010, Jul. 2015.
3. T. Brown, "IoT-based Monitoring and Control System for Aquaponics," IEEE Trans. Sustainable Computing, vol. 3, no. 2, pp. 150-158, Apr. 2018.
4. K. Wilson, "Cloud-based Data Analysis for Aquaponics Monitoring Systems," in Proc. IEEE International Symposium on Green Computing, Month Year, pp. 100-105.
5. R. Garcia and B. Martinez, "Integration of IoT and Aquaponics for Sustainable Agriculture," IEEE Access, vol. 6, pp. 50000-50012, Sep. 2018.

## Contribution Of Team Members:

|  |  |
| --- | --- |
| Team Member | Assigned Topics |
| Prahalad Krishnan | Project Management, Code Development, Report Writing |
| Shayan Shaikh | Hardware Assembly, Sensor Calibration |
| Ketavya Chitransh | Circuit Design, Testing |
| Hardik Maheshwari | Data Analysis, Documentation |