e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
Volume:02/Issue:08/August-2020 Impact Factor- 5.354 www.irjmets.com

AUTOMATED HYDROPONIC PLANT GROWTH SYSTEM USING IOT

Suhan M *1, Surya Murali *2, Shriram Bharadwaj T*3, Dr. P. Janarthanan*4

*1,2,3 Undergraduate, Department of Computer Science and Engineering, Sri Venkateswara College of Engineering, Kancheepuram, Tamilnadu, India.

*4Associate Professor, Department of Computer Science and Engineering, Sri Venkateswara College of Engineering, Kancheepuram, Tamilnadu, India.

ABSTRACT

This paper, an integrated Internet of Things (IoT) framework is proposed for controlling and managing the hydroponics garden. The problems of controlling those resources will be solved with the increasing trend of IoT and automation. This system aims to provide the optimal environment for plants to grow, a system for continuous monitoring of pH, water level, air temperature, and relative humidity. In addition, this device provides a regulated irrigation of water and the intake of nutrient solutions by using simple mechanisms. Information is processed, controlled, implemented, and exchanged internally via the Internet collected by the sensors and the use of cloud-based technology as the backend. Managing the resources in hydroponics will be easier to set up and more successful, based on the study's success and results.

KEYWORDS: IoT, sensors, web management, plants, hydroponics.

I. INTRODUCTION

Tending crops and vegetables is not a simple assignment, further so with hydroponics. There are a lot of variables and outside circumstances that could influence the success or failure of the yield. The traditional way of determining which factors led to yield quality is through observation and manual measurements which are both ineffective and insufficient. In the age where information is readily available, the accurate collection and presentation of data are essential in maintaining the integrity of the farming system. These data are facts and statistics collected together for reference or analysis. Information, in general, is a prerequisite to produce backed-up and evidential outcomes that could significantly improve yield quality. Smart farming is becoming a part of the technological advancements brought about by globalization. Clever farming includes the establishment of data and information technologies into machines, equipment, and sensors for use in horticultural stock systems. Hydroponics cultivation grows more creative and seen to be the destiny of agriculture. Farms with an automated system produce a higher quality of crops by sensing its controlling parameters. This paper presents an automated system for the farming of hydroponic style. The objective is to atomize the crop monitoring during the growth process using the network of sensors and actuators. This system assists in monitoring and command of numerous real issues at fields such as water level, pH, electrical conductivity, water temperature, and relative humidity. These physical issues are managed by implementing determinations through messaging systems. From the test outcomes, it is observed that the recommended method shows tiniest variations in sensor states which helps in automated control. This work essentially accumulates data about environmental components from antique sensors and using remote conventions transfers it to the database. A system results in inappropriate decisions about various nutrient values and informs the farmer through messaging systems. A model is developed which appears in sending final decisions to the end-users.

II. WHAT IS HYDROPONICS

Hydroponics is a type of hydroculture, which is plant growth in a less medium soil, or an aquatic environment. Hydroponic growth uses nutrient-mineral solutions to feed the plants in water, without soil. Hydroponic gardening requires growing plants rather than soil, in a nutrient solution. The benefit of hydroponics is that you can prevent many of the problems affecting plants grown in the soil, such as cutworms and soil-borne diseases that can destroy your farm. This means they can stop herbicides and pesticides. You also have greater control over the nutrients which feed your plants. To ensure optimum growth, it is easier to vary the nutrients which the plant receives at different stages of its development.

A porous growing aggregate is used, instead of soil. This can include sand, vermiculite, gravel, cocoon coir, clay or perlite balls. This enables more free circulation of air and nutrients enabling better delivery of oxygen and food to each plant. Nutrients and water are fed directly to the roots to allow the plant to spend more of its energy growing above the ground, rather than moving through the ground to compete for nutrients. Since the roots are smaller, the plants may grow closer together and therefore save space. This can mean that it is increasing faster and producing higher yields within a smaller area.

III. LITERATURE SURVEY

The major disadvantage of Hydroponics is its cost that is higher than the natural soil method as it needs soilless culture structure including controlling and observing equipment and technical ability to cope up with it. It also requires constant supervision i.e. water-based microorganism can be easily introduced or in case of power outage setup must be handled manually. The above listed are of lesser importance in contrast to cultivation on an industrial scale. There have been many technological progressions in the scope of agriculture in modern years. Example of which is the hydroponic farming method of cultivating crops using nutrient-enriched water without soil. While hydroponic agriculture has various benefits, the processes are more complex than those of conventional farming. Chances of method crash are higher due to combined factors of growing. Outcomes revealed that the sensors were correctly calibrated and defined based on standard tools. Additionally, with the usage of uncomplicated mechanisms, this method provides controlled irrigation of 4 water and nutrient solution intake. Throughout the data collected by the sensors and the use of cloud-based technology. Hydroponics is a procedure to develop the plant without utilizing the soil. This method guarantees the plant receives all nutrients wanted from the water solution. There are several varieties of hydroponics methods. Aeroponic practices need a few nutrients and air. The crops must stay on a tray that holds nutrients and water at the ground, with the roots attaching in the air. The roots are misted with the nutrient-rich water every few minutes and a timer manages the nutrient pump. This method can be used solely for high economic investments as it is tough to install up and maintain. Nutrient-rich water is pumped via small pipes and drips on to the head of the plants, with a timer managing the submersible pump. Ebb and Flow System, In an ebb and flow system, a plant clench tray is momentarily overwhelmed with nutrient-rich water which is then drained back into a reservoir. A timer manages a submersible pump to execute the tasks many times a day, with perlite or gravel utilized to produce the plants with a level of endurance. This system is suitable for home use. Nutrient Film Technique System, In a nutrient film technique system, plants are 5 sustained in tiny molded baskets in a tray with their roots attaching into a nutrient solution. This process is intended to offer a constant flow of nutrient solution to the tray above the plant roots. The solution then drains off into a reservoir. This method is perfect for plants such as tomatoes and cucumbers. Water culture System, In a water production system, a Styrofoam stand including nutrient-rich water carries the plants. An air pump provides air to a bubbling rock that aids in the discharge of the nutrient solution and the supply of oxygen to the plant roots. This system can be solely utilized for some varieties of plants such as leaf lettuce for example. A wick method comprises no moving parts, with increasing mediums such as per-liter or rock-wool used in the tray. A nutrient solution is released into the tray through the wicks. The foremost benefit of this practice is that it demands no timers or pumps. In the future, we propose to practice many numerous uncomplicated equipment and easy algorithms to make our system less complicated and less expensive to grow a plant without any human intervention even in the indoor environment.

IV. METHODOLOGY

The system proposed for smart, hydroponic farming is built on a modest scale. The goal is to track and supervise the plant growth safe. The whole job is broken down into different subsystems each of these plays a significant part in the process of atomization. The clean water testing subsystem shall monitor the pH value of water and water Niveau. Sensors are helped to recognize and use information which It quantifies crop variations. The nutrient-rich solution with an acceptable solution Quantity, very important in a given range which is guaranteed by pH Button. Button. The pH sensor also controls the acid release and pH adjustment bases. Those factors are constantly monitored by the flow sensor or water level. The rising mean should be, at any rate, do not become dry. The module for environmental monitoring makes use of Raspberry Pi performs two essential functions; first, it investigates power loss, loss of the equipment, and second, it controls the temperature and Moisture with a camera. The machine gathers all of the information and stores Output in base MySQL. The farmers will be able to get the sensor data back for live lectures. In this project, a fully automated system for hydroponic plant growth is proposed. The key utilization behind this automated system is to develop any suitable plant, indoors, and without humans Intervention.

The goal is to provide the major requirements for automation in such hydroponic systems as:

- Manipulating the river water with the Ph.
- Turn the UV lights on / off to provide day/night conditions for the plant.
- Ground irrigation.
- Supply to the plant of the necessary nutrients, depending on the plant level.

The dataset is acquired and the user must send it to the system before planting the seed into the system. The data set includes details such as the type of nutrients and the number of nutrients needed by the plant each day, the necessary pH level of the plant grown, the level of water needed by the plant, and the optimum daytime and night time required by the plant.

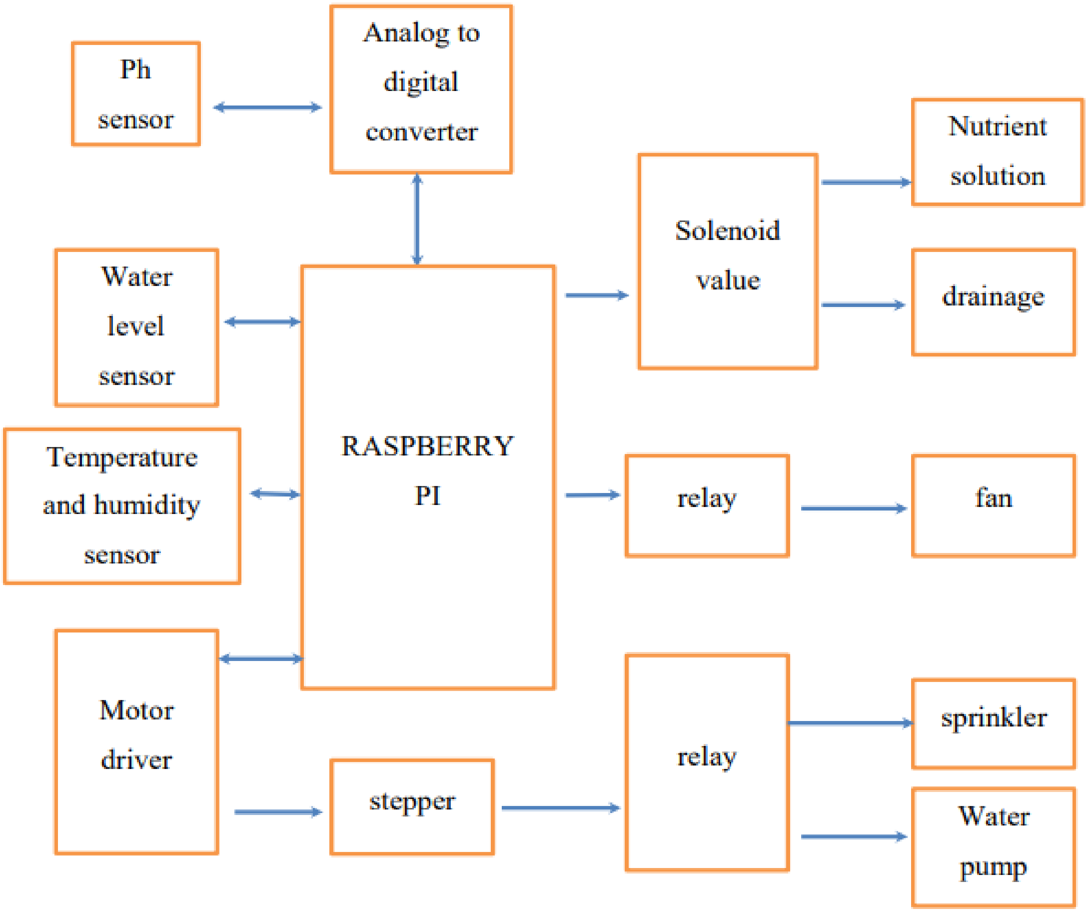


Figure:1 Architecture Diagram.

V. MODELING AND ANALYSIS

The modeling of the system and the nutrients requirements are shown below.

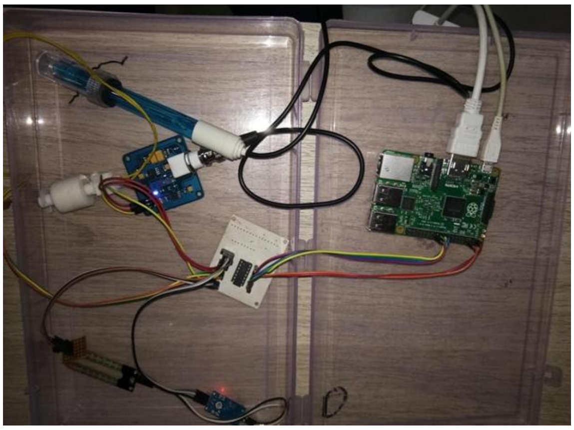


Figure-2: RPI Connections.

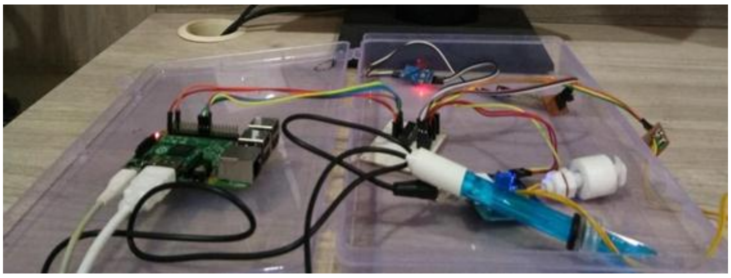


Figure-3: RPI Sensors.

Modular design defines the structure of the overall system. Modularity is a general system concept typically defined as the degree to which a system's components may be separated and recombined. The overall system consists of the following modules:

1. Configuration, to assign pins and ports
2. Hydroponics, to create Hydroponics Controller
3. Hydroponics_server, to create a server to launch webapp
4. Main, webapp for controlling the hydroponics system
5. Web App, control the hardware

The configuration module is the first module that is run because a number of important pins and ports are assigned to the respective variables. The pump and lights are assigned to pins 13 and 11, pump default setting is assigned as false, lights default setting is assigned to true, and the time for which the pump should be run is also assigned in this module. The time that the lights should be turned on and off is also specified in this module. Lastly, the port that should be used by the Hydroponics Server is also specified.

The variables that are assigned in this module are:

-PUMP_PIN : Pump control pin

-LIGHTS_PIN : Lights control pin

-PUMP_DEFAULT_ON : Default setting of pump

-LIGHTS_DEFAULT_ON : Default setting of lights

-PUMP_TIME : Time of running of pump once started

-LIGHTS_ON : At what hour lights turn on

-LIGHTS_OFF : At what hour lights turn off

-PORT : Hydroponics Server port

The class Hydroponics Controller is the main control of the hydroponics system which contains all the important variables and the respective conditions in which the raspberry pi has to work so that the process takes place correctly. An object for the Hydroponics Controller class is created so that the GPIO mode can be set up in the correct mode. All the other variables and conditions are assigned and loops are run to check the working of all the variables.

```python
class HydroponicsController(object):

    def __init__(self, **kwargs):
        self.pump_pin = kwargs["pump_pin"]
        self.lights_pin = kwargs["lights_pin"]
        self.pump_default_on = kwargs["pump_default_on"]
        self.lights_default_on = kwargs["lights_default_on"]

        gpio.setmode(gpio.BOARD)
        gpio.setup(self.pump_pin, gpio.OUT)
        gpio.setup(self.lights_pin, gpio.OUT)

    def pump_on(self):
        if self.pump_default_on:
            gpio.output(self.pump_pin, gpio.LOW)
        else:
            gpio.output(self.pump_pin, gpio.HIGH)

    def pump_off(self):
        if self.pump_default_on:
            gpio.output(self.pump_pin, gpio.HIGH)
        else:
            gpio.output(self.pump_pin, gpio.LOW)

    def run_pump(self, pump_time):
        """Run the pump for `pump_time` seconds."""
        self.pump_on()
        time.sleep(pump_time)
        self.pump_off()
```

Figure-4: Python code of Hydroponics controller.

MockHydroponicsController is another part of the Hydroponics moule where a mock setting of the Hydroponics controller class is created but the difference is that instead of accessing the GPIO using the GPIO BOARD, the methods that print to console are added. This class has the same interface as the HydroponicsController and an object is created to access it.

Hydroponics Server is used to start up a server to run the hydroponics controller. This server runs as a RPyC service and can thus be controlled by another program like the webapp. Before running the program, the server must be run and the script must be executed so that the server can start. The web application does not control the hardware directly because we want only one instance of the Hydroponicscontroller running at the same time, but the web server creates a new thread for every connection.

```
import datetime
import logging
logging.basicConfig()

from apscheduler.jobstores.base import JobLookupError
from apscheduler.schedulers.background import BackgroundScheduler
import rpyc
from rpyc.utils.server import ThreadedServer

from hydroponics import HydroponicsController, MockHydroponicsController
from config import (PUMP_PIN, LIGHTS_PIN, PUMP_DEFAULT_ON, LIGHTS_DEFAULT_ON,
                    PUMP_TIME, LIGHTS_ON, LIGHTS_OFF, PORT)


def CustomizedHydroponicsService(hydroponics_controller, scheduler):
    state = {"paused": False, "resume_time": None}

    class HydroponicsService(rpyc.Service):
        def exposed_is_paused(self):
            return state["paused"]

        def exposed_get_resume_time(self):
            return state["resume_time"]

        def exposed_resume(self):
            """Resume regular operation of the hydroponics system."""

            state["paused"] = False
            try:
                scheduler.remove_job("resume")
            except JobLookupError:
```

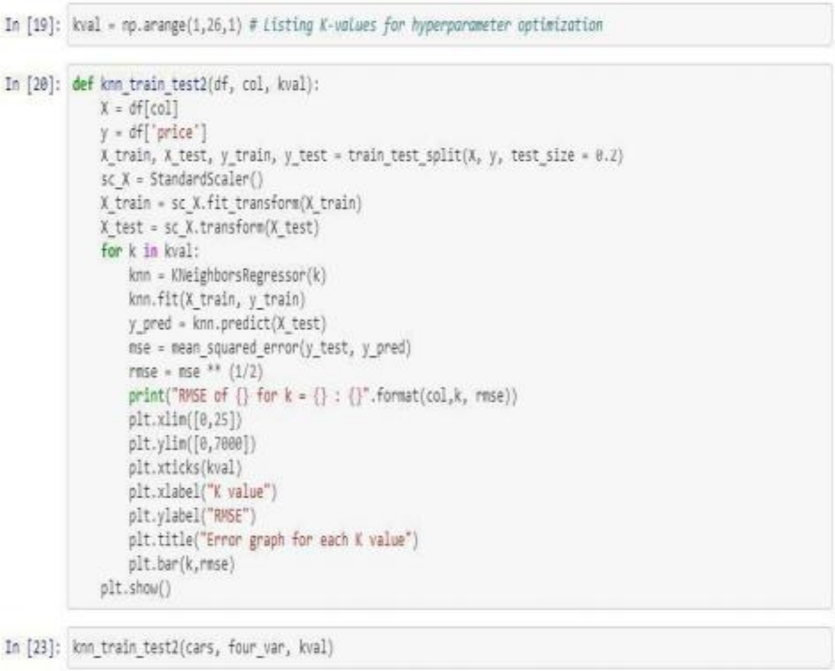
Figure-5: Python code of Hydroponics server.

The main function is used to run the whole Hydroponics system that has been developed. In the main, the HydroponicsController function is called using kwargs and the job is scheduled to be performed at the right times. The pumps are first given a job to be turned on once every 1 hour and the lights on time and light off time are also specified. The control of the program comes from main and the control is shifted to the specific jobs that have been scheduled so that they can take place in the right times that have been specified by the user. Import time statements are used so that correct time can be kept track of so that all the sub systems can work in a systematic manner.

```python
import time

from apscheduler.schedulers.background import BackgroundScheduler

from config import (PUMP_PIN, LIGHTS_PIN, PUMP_DEFAULT_ON, LIGHTS_DEFAULT_ON,
                    PUMP_TIME, LIGHTS_ON, LIGHTS_OFF)
from hydroponics import HydroponicsController


if __name__ == '__main__':
    scheduler = BackgroundScheduler()
    kwargs = {"pump_pin": PUMP_PIN,
              "lights_pin": LIGHTS_PIN,
              "pump_default_on": PUMP_DEFAULT_ON,
              "lights_default_on": LIGHTS_DEFAULT_ON}

    with HydroponicsController(**kwargs) as h:
        scheduler.add_job(h.run_pump, 'interval', hours=1, args=(PUMP_TIME,))
        scheduler.add_job(h.lights_on,  'cron', hour=LIGHTS_ON)
        scheduler.add_job(h.lights_off, 'cron', hour=LIGHTS_OFF)
        scheduler.start()

        try:
            while True:
                time.sleep(10)
        finally:
            scheduler.shutdown()
```

Figure-6: Python code of Main function.

The Flask web application serving an interface for entering quiet mode is done in this module. Flask is a lightweight WSGI web application framework. It is designed to make it fast and easy to get started, with the capability to scale up to complex applications. It started out as a simple wrapper and has become one of the most common web application frameworks in Python. Flask gives recommendations but it does not implement any dependencies or layout of projects. It is up to the developer to select the software they want to use and the libraries. There are several Group plugins that make it easy to add new features. The web application communicates with an EPyC server which controls the hardware in operation. The server must be started using sudo python hydroponics server.py and then the web application must be launched.



```
In [19]: kval = np.arange(1,26,1) # Listing K-values for hyperparameter optimization

In [20]: def knn_train_test2(df, col, kval):
             X = df[col]
             y = df['price']
             X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)
             sc_X = StandardScaler()
             X_train = sc_X.fit_transform(X_train)
             X_test = sc_X.transform(X_test)
             for k in kval:
                 knn = KNeighborsRegressor(k)
                 knn.fit(X_train, y_train)
                 y_pred = knn.predict(X_test)
                 mse = mean_squared_error(y_test, y_pred)
                 rmse = mse ** (1/2)
                 print("RMSE of {} for k = {} : {}".format(col,k, rmse))
                 plt.xlim([0,25])
                 plt.ylim([0,7000])
                 plt.xticks(kval)
                 plt.xlabel("K value")
                 plt.ylabel("RMSE")
                 plt.title("Error graph for each K value")
                 plt.bar(k,rmse)
             plt.show()

In [23]: knn_train_test2(cars, four_var, kval)
```

Figure-7: Python code of Web Application.

Table-1: Nutrients Requirements

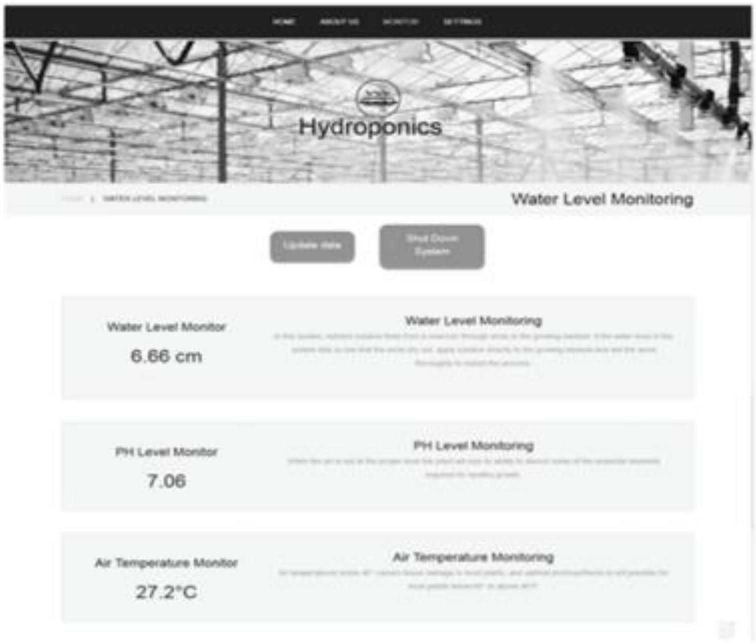
PLANT	***pH***	***GF***	***EG***	***PPM***
Bean	6.0	20-40	2-4	1400-2800
Beetroot	6.0-6.5	8-50	0.8-5	1260-2500
Broccoli	6.0-6.5	28-35	2.8-3.5	1960-2450
Brussels Sprout	6.5-7.5	25-30	2.5-3.0	1750-2100
Cabbage	6.5-7.0	25-30	Z.5-3.0	1750-2100
Capsicum	6.0-6.5	18-22	1.8-2.2	1260-1540
Cauliflower	6.0-7.0	5-20	0.5-2.0	1050-1400
Cucumber	5.8-6.0	17-25	1.7-2.5	1190-1750
Garlic	6.0	14-18	1.4-1.8	980-1260
Lettuce	5.5-6.5	8-12	0.8-1.2	560-840
Onions	6.0-6.7	14-18	1.4-1.8	980-1260
Bell Peppers	6.0-6.5	20-25	2.0-2.5	1400-1750
Potato	5.0-6.0	20-25	2.0-2.5	1400-1750
Pumpkin	5.5-7.5	18-24	1.8-2.4	1260-1680
Radish	6.0-7.0	16-22	1.6-2.2	840-1540
Spinach	5.5-6.6	18-25	1.8-2.5	1260-1610
Sweet Corn	6.0	16-24	1.6-2.4	840-1680
Sweet Potato	5.5-6.0	20-25	2.0-2.5	1400-1750
Tomato	5.5-6.5	20-50	2.0-5.0	1400-5500

	Temperature (°C)	Relative Humidity (%)	Temperature (°C)	Relative Humidity (%)
Initial Data	29.9	65.02	30.6	63.35
Data at 5 minutes	29.98	64.7	30.53	63.3
Data at 10 minutes	30.1	63.76	30.4	63.6
Data at 15 minutes	30.2	64.2	30.45	62.75
Data at 20 minutes	30.38	63.6	30.53	62.25

Figure-8: Effect of the Fan and the Sprinkler to Relative Humidity and Air Temperature.

Samples	ph 7	ph 10	ph 4
Sample 1	6.80	9.94	3.87
Sample 2	6.81	9.90	3.82
Sample 3	6.70	9.91	3.80
Sample 4	6.68	9.92	3.78
Sample 5	6.77	9.91	3.77
Sample 6	6.76	9.92	3.79
Sample 7	6.75	9.90	3.85
Sample 8	6.78	9.93	3.77
Sample 9	6.79	9.91	3.77
Sample 10	6.78	9.91	3.76
Average	**6.762**	**9.915**	**3.798**
Percent Error	**3%**	**1%**	**5%**
Margin of Error	**± 0.238**	**± 0.085**	**±0.202**

Figure-9: pH Sensor Reliability Test



Water Level Monitoring

Water Level Monitor
6.66 cm

Water Level Monitoring

PH Level Monitor
7.06

PH Level Monitoring

Air Temperature Monitor
27.2°C

Air Temperature Monitoring

Figure-10: Monitoring Page

VI. CONCLUSION

The proposed system enables an easy method of farming and growing plants that ensures perfect balance and better yield. The vegetables grown can be perfect based on the number of nutrients that the specific plant needs. The system gives a better yield than most farming techniques and can be controlled effectively using an RPI controller.

FUTURE WORK

The system can be further modified using an air conditioner and a tent which can help control more aspects of the system and can yield better results for certain plants that need a particular environment to grow. The air conditioner can be used to control the temperature and a dehumidifier can be used to control the humidity based on the needs of the plants.

VII. REFERENCES

[1] DiCola, Tony. "Raspberry Pi Analog to Digital Converters." https://learn.adafruit.com/raspberry-pi-analog-to-digitalconverters/mcp3008#software-spi

[2] "Raspberry Pi ADC: MCP3008 Analog to Digital Converter." https://pimylifeup.com/raspberry-pi-adc/

[3] Monk, Simon. "Adafruit's Raspberry Pi Lesson 10. Stepper Motors." https://learn.adafruit.com/adafruits-raspberry-pi-lesson-10-steppermotors/hardware-uln2803

[4] Nedelkovski , Dejan. "Control High Voltage Devices – Arduino Relay Tutorial." http://howtomechatronics.com/tutorials/arduino/controlhigh-voltage-devices-arduino-relay-tutorial/

[5] Kartha, Vivek. "Interfacing HC-SR04 Ultrasonic Sensor with Raspberry Pi." https://electrosome.com/hc-sr04-ultrasonic-sensor-raspberry-pi/

[6] Griffiths, Mark. "The design and implementation of a hydroponics control system." Oulu University of Applied Sciences. Autumn 2014.