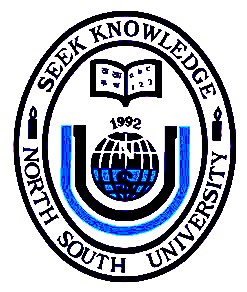
**Department of Electrical and Computer Engineering**

**North South University**



**Senior Design Project**

**Automated Hydroponic System Using IOT and Cloud Computing**

**Ramiza Maliha ID # 1430592042**

**Md. Nahid Rahman Ontu ID # 1431123042**

**Syed Tahmid Abrar ID # 1430651042**

**Faculty Advisor:**

**Mr. Zunayeed Bin Zahir**

**Lecturer, ECE Department**

**Summer, 2019**

**Declaration**

This is to declare that no part of this report or the project has been previously submitted elsewhere for the fulfillment of any other degree or program. Proper acknowledgement has been provided for any material that has been taken from previously published sources in the bibliography section of this report.

.........................................................

Ramiza Maliha

ECE Department

North South University, Bangladesh

........................................................

Md. Nahid Rahman Ontu

ECE Department

North South University, Bangladesh

.........................................................

Syed Tahmid Abrar

ECE Department

North South University, Bangladesh

**Approval**

The Senior Design Project entitled “**Automated Hydroponic System using IoT and Cloud Computing** ” by Ramiza Maliha (ID#1430592042), Md. Nahid Rahman Ontu (ID#1431123042) and Syed Tahmid Abrar (ID#1430651042) has been accepted as satisfactory and approved for partial fulfillment of the requirement of BS in CSE degree program on Summer, 2019.

|  |
| --- |
| **Supervisor’s Signature** |
|  |
| **Mr. Zunayeed Bin Zahir Lecturer**  Department of Electrical and Computer Engineering  North South University  Dhaka, Bangladesh. |
| **Chairman’s Signature** |
| **Dr. Kazi Mohammad Abdus Salam**  **Professor & Chairman**  Department of Electrical and Computer Engineering  North South University  Dhaka, Bangladesh. |

**Acknowledgement**

First of all, we would like to express our profound gratitude to our honorable course instructor, **Mr. Zunayeed Bin Zahir,** for his constant and meticulous supervision, valuable suggestions, his patience and encouragement to complete the thesis work.

We would also like to thank the ECE department of North South University for providing us with the opportunity to have an industrial level design experience as part of our curriculum for the undergraduate program.

Last but not the least, we would like to thank our families and everybody who supported us and provided with guidance for the completion of this project.

**Abstract**

In this report, we have explored the idea of automation in regards of hydroponic system. The purpose of this project was to make this process more manageable and accessible for the masses so that the people of Bangladesh can re-establish the slowly falling agricultural sector with advance new technology. As the unemployment rate is increasing and people are looking for new ways to stabilize the growing economy that people of Bangladesh has worked hard for, the most lucrative offer can come from the most back dated sector of our country, thus the hydroponic system. Hydroponic system can be the emancipator of our agricultural system, albeit the venture is quite risky due to the fact that we lack both knowledge and resources. Since, hydroponic system present itself as something of a sacrilege, which more often than not, people tends to avoid. Furthermore, with the backward technology we have mixed with the lack of understanding, most of the enthusiast either gives up half way or are still struggling. Ergo, this project is aiming to help the entrepreneurs to invest in agriculture so that we can not only reinvent our agriculture sector with new generation of young people, but also provide the country with adequate food, medicinal plant, foreign plants etc. We are integrating internet of things (IoT) and cloud computing to ease the hydroponic system, which requires constant maintenance, so that the new comers has more access.

Table of Contents

*[Chapter 1](#_Toc19090866)* *[Project Overview](#_Toc19090867)* [12](#_Toc19090867)

[1.1 Introduction 13](#_Toc19090868)

[1.2 Hydroponic system basics 14](#_Toc19090869)

[1.2.1 How Hydroponic system works 14](#_Toc19090870)

[1.2.2 Applications of hydroponic system 15](#_Toc19090871)

[1.2.3 Pros and Cons 15](#_Toc19090872)

[1.3 Our proposed project 16](#_Toc19090873)

[1.3.1 Description of the idea 16](#_Toc19090874)

[1.3.2 Difficulty 17](#_Toc19090875)

[1.4 Motivation 17](#_Toc19090876)

[1.5 Summary 18](#_Toc19090877)

[Chapter 2](#_Toc19090878) *[Related work](#_Toc19090879)* [19](#_Toc19090879)

[2 Related Work 20](#_Toc19090880)

*[Chapter 3](#_Toc19090881)* *[Theory](#_Toc19090882)* [23](#_Toc19090882)

[3. Theory 24](#_Toc19090883)

*[Chapter 4](#_Toc19090884)* *[Structure of the system](#_Toc19090885)* [27](#_Toc19090885)

[4.1 Introduction 28](#_Toc19090886)

[4.2 NFT System (Hydroponic system) 28](#_Toc19090887)

[4.3 Automated IoT based System 29](#_Toc19090888)

[4.4 Remote monitoring using an App 30](#_Toc19090889)

[Chapter 5 Technical Exposition 31](#_Toc19090890)

[5.1 Introduction 32](#_Toc19090891)

[5.2 Node MCU with ESP8266 WiFi Module 32](#_Toc19090892)

[5.2.1 NodeMCU Lua V3 ESP8266 WiFi Module with CH340G 33](#_Toc19090893)

[5.2.2 Pin Configuration 34](#_Toc19090894)

[5.3 Arduino Uno R3 35](#_Toc19090895)

[5.3.1 Description and Configuration 36](#_Toc19090896)

[5.4 Ultrasonic Ranging Module HC - SR04 37](#_Toc19090897)

[5.5 pH sensor with BNC connector and signal conversion module 39](#_Toc19090898)

[5.6 Temperature and Humidity module (DHT22 AM2302) 41](#_Toc19090899)

[5.7 Grow light 42](#_Toc19090900)

[5.8 Oxygen Pump 42](#_Toc19090901)

[5.9 DC Fan (12V) 42](#_Toc19090902)

[5.10 Servo Motor MG996R (180 Degree) 43](#_Toc19090903)

[5.11 Submersible pump (SP2500) 43](#_Toc19090904)

[5.12 Micro Submersible pump (6V) 43](#_Toc19090905)

[5.13 Relay 44](#_Toc19090906)

[5.14 Servers and Application 45](#_Toc19090907)

[5.14.1 Thingspeak 46](#_Toc19090908)

[5.14.2 Arduino IDE 48](#_Toc19090910)

[5.14.3 Android Studio 49](#_Toc19090911)

[5.15 Summary 49](#_Toc19090912)

[Chapter 6](#_Toc19090913) [Design Implementation 50](#_Toc19090914)

[6.1 Introduction 51](#_Toc19090915)

[6.2 Required Equipment 51](#_Toc19090916)

[6.2.1 Required Equipment for design implementation 51](#_Toc19090917)

[6.2.2 Required Equipment and component for internal circuitry 52](#_Toc19090918)

[6.3 Controlling the pH value 52](#_Toc19090919)

[6.4 Controlling the amount of solution in the main container 53](#_Toc19090921)

[6.5 Controlling the humidity and temperature 53](#_Toc19090922)

[6.6 Getting the notification in our mobile application when the reservoirs are empty 54](#_Toc19090923)

[6.7 Controlling the servo motor and the grow light 54](#_Toc19090924)

[6.8 Controlling the concentration of the nutrients in the main container 54](#_Toc19090925)

[6.9 Controlling the main submersible pump and the oxygen pump 55](#_Toc19090926)

[Chapter 7](#_Toc19090927) *[Compliance with standards](#_Toc19090928)* [56](#_Toc19090928)

[7.1 Introduction 57](#_Toc19090929)

[7.2 Compliance with IEEE standards 57](#_Toc19090930)

[7.3 Compliance with US standards 57](#_Toc19090931)

[7.4 Compliance with European standards 57](#_Toc19090932)

[Chapter 8](#_Toc19090933) *[Design Impact](#_Toc19090934)* [58](#_Toc19090934)

[8.1 Introduction 59](#_Toc19090935)

[8.2 Economic impact 59](#_Toc19090936)

[8.3 Environmental impact 59](#_Toc19090937)

[8.4 Social and Health impact 62](#_Toc19090938)

[Chapter 9](#_Toc19090939) *[Discussion and Budget](#_Toc19090940)* [63](#_Toc19090940)

[9.1 Discussion 64](#_Toc19090941)

[9.2 Budget 64](#_Toc19090942)

[Chapter 10](#_Toc19090943) *[Result and Conclusion](#_Toc19090944)* [65](#_Toc19090944)

[10.1 Result 66](#_Toc19090945)

[10.2 Conclusion 66](#_Toc19090946)

***[Bibliography](#_Toc19090947)*** [68](#_Toc19090947)

[Bibliography 69](#_Toc19090948)

***[APPENDIX](#_Toc19090949)*** [71](#_Toc19090949)

[APPENDIX A 72](#_Toc19090950)

[APPENDIX B 74](#_Toc19090951)

[APPENDIX C 76](#_Toc19090952)

[APPENDIX D 78](#_Toc19090953)

[APPENDIX E 80](#_Toc19090954)

[APPENDIX F 81](#_Toc19090955)

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Fig No.** | **Figure Caption** | **Page No.** |
| 1.1 | Cultivable land of Bangladesh since 1970 according World Bank. (x-axis: ‘year’ and y-axis ‘percentage of arable land’) | 12 |
| 1.2 | Working of Nutrient Film Technique (NFT) | 14 |
| 4.1 | System Block Diagram | 28 |
| 4.2 | NFT System Structure | 28 |
| 4.3 | System Architecture | 29 |
| 4.4 | Customized App called ‘MINT’ | 30 |
| 5.1 | NodeMcu Lua V3 ESP8266 WiFi with CH340G | 34 |
| 5.2 | NodeMCU with pin configuration | 35 |
| 5.3 | Arduino Uno R3 with pin configuation | 37 |
| 5.4 | Working of a Ultrasonic sonar sensor | 37 |
| 5.5 | Ultrasonic Sonar Sensor: Front View (left); BackView(right) | 39 |
| 5.6 | Signal Converter board | 40 |
| 5.7 | pH sensor with BNC and signal converter | 40 |
| 5.8 | Temperature and Humidity Sensor DHT22 | 41 |
| 5.9 | Grow Light | 42 |
| 5.10 | Oxygen pump | 42 |
| 5.11 | DC Fan | 42 |
| 5.12 | Servo Motor | 43 |
| 5.13 | SP2500 submersible pump | 43 |
| 5.14 | Micro Submersible | 44 |
| 5.15 | 1 channel relay | 44 |
| 5.16 | Internal working of a relay | 45 |
| 5.17 | ThingSpeak Architecture | 46 |
| 5.18 | Data Analysis in ThingSpeak | 47 |
| 5.19 | Arduino IDE (screenshot) | 48 |
| 5.20 | Android Studio(Screen Shot) | 49 |
| 6.1 | pH sensor circuit | 53 |
| 6.2 | Controlling the humidity and temperature | 54 |
| 6.3 | Sonar Sensor circuit | 55 |
| 10.1 | Completed Project | 66 |

**List of Tables**

|  |  |  |
| --- | --- | --- |
| **Tab. No.** | **Table Caption** | **Page No.** |
| 1.1 | Imported items in Bangladesh with costing | 17 |
| 1.2 | Exported items in Bangladesh with costing | 17 |
| 5.1 | Ultrasonic Sensor Pin Configuration | 38 |
| 5.2 | pH sensor Pin Configuration | 40 |
| 5.3 | DHT22 module Pin Configuration | 41 |
| 9.1 | BUDGET | 64 |

# *Chapter 1*

# *Project Overview*

### Introduction

For a delta like Bangladesh, it is to be expected that agriculture is still the most important sector that contributes about 19.6% to the national GDP. This sector is responsible for the employment of about 63% of the population. However, agriculture in Bangladesh is heavily dependent on the sporadic weather changing, and the entire harvest can be wiped out in a matter of hours when cyclones hit the country [4]. However, according to the World Bank, the total arable land in Bangladesh is well below 54% (2011).

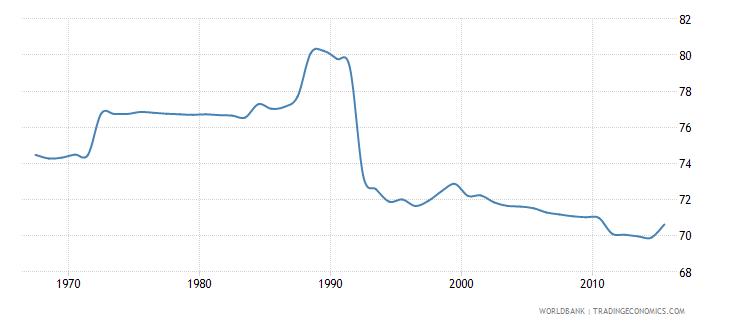


Fig 1.1: Cultivable land of Bangladesh since 1970 according World Bank. (x-axis: ‘year’ and y-axis ‘percentage of arable land’)

For a country with such like Bangladesh arable land is decreasing at an increasing rate because of certain factors such as over population. On the other hand, soil that was once fertile and good for growing crops can become redundant if used to grow crops all year around. Soil pollution and water pollution has become the repercussion of fertilizer and pesticides usage. In fact, Bangladesh has too little space for a huge population and the main economic backbone is breaking slowly which started to create an unimaginable collateral damage to everyone our lives. One of the more lucrative solutions to this problem is hydroponic system.

## 1.2 Hydroponic system basics

A prominent subset of hydro-culture, also known as aquatic horticulture is hydroponics which is a method of soilless farming, using mineral [nutrient](https://en.wikipedia.org/wiki/Nutrient" \o "Nutrient) [solutions](https://en.wikipedia.org/wiki/Solution" \o "Solution) in a water solvent. Terrestrial plants, especially the soft stem plants, may be grown with only their [roots](https://en.wikipedia.org/wiki/Root" \o "Root) exposed to the mineral solution, or the roots may be supported by an inert medium, such as [perlite](https://en.wikipedia.org/wiki/Perlite" \o "Perlite) or [gravel](https://en.wikipedia.org/wiki/Gravel" \o "Gravel). The nutrients in hydroponics can come from an array of different sources; these can include but are not limited to byproduct from fish waste, duck manure, or [normal nutrients](https://en.wikipedia.org/wiki/Chemical_fertilizer" \o "Chemical fertilizer).[4]

There are many different types of hydroponic system, based on architectural and technique. However, in this project we are working with Nutrient film technique (NFT) which is one of the most popular hydroponic systems available.

### 1.2.1 How Hydroponic system works

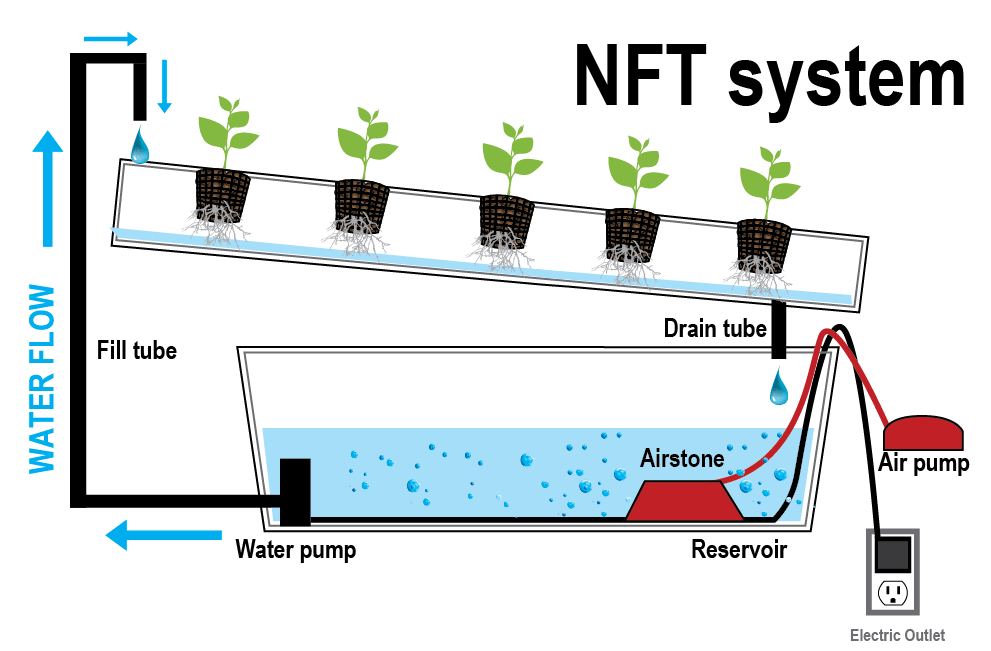


Fig 1.2: Working of Nutrient Film Technique (NFT)

Nutrient film technique (NFT) is a hydroponic technique where in a very shallow stream of water containing all the dissolved nutrients required for plant growth is re-circulated past the bare roots of plants in a watertight channel. For this process we need the following things under control:

1. PH level of the water
2. Electric Conductivity of the water
3. Temperature and Humidity
4. Water level of in the pipe
5. Water temperature etc.

### 1.2.2 Applications of hydroponic system

Hydroponic System can be a great advancement of our agricultural sector.

* We can apply this system to produce plants that are foreign to our country.
* We can produce rare medicinal plants all year around.
* We can apply this method to produce more food crops.
* We can make sure that there are plants in a super city like Dhaka and stop pollution of different sorts.
* We can apply this method to produce ornamental plants that are always in demand.
* This can help the garden hobbyist to get themselves a garden in this cramped up city.

### 1.2.3 Pros and Cons

Pros of Hydroponic System[15]

* Growing anywhere. We needn’t fertile ground.
* Water can be reused, so hydroponics systems require 20 times less water than soil based gardening.
* Reduces nutrient leaching into the environment (and you save money)
* No pesticides, if we provide a sterile environment to the plant.
* Hydroponics systems need 20% less space for growing.
* You can have complete control over nutrient balance.
* Less labor. Hard tasks such as tilling, cultivating, fumigation and mulching are not required for hydroponic growing.
* Growing year round; season isn’t a problem.
* Crops grow faster in hydroponic culture than in a soil based growing medium.
* It is a proven technology that ensures high yields.
* Harvesting is easier. Growing is a relaxing hobby, stress-relieving and a great way to spend quality family time.

Cons of Hydroponic System [15]

* Installing a hydroponic system expensive.
* More supervision is required as the system is very sensitive.
* Mistakes and system malfunctions affect plants faster, without soil acting as a buffer.
* Hydroponic gardens are affected by power outages.
* It requires the use of better water.
* Waterborne diseases spread quickly.

## 1.3 Our proposed project

As we go through the cons list we can see that hydroponic system is very sensitive to malfunction. So we proposed to make this system as less sensitive as possible and make sure that we can remotely maintain our hydroponic system. The system is self-sufficient in a sense that it measures and controls all the sensitive aspects that will be discussed below according to the sensor reading and lets the user know if he needs to act upon some issue via an app.

### 1.3.1 Description of the idea

We are going to create a fully automated system that has tree back up reservoirs containing extra nutrient, water and pH buffer that supplies its content to the main tank when it is needed. This need is determined by the use of sensors that is attached to each of the reservoir so that the most sensitive characteristics of the system can be detected and neutralize if any anomaly occurs, In addition to that, the user can monitor and will be notified if he/she needs to manually add something to the system, the rarity of which will depend mostly on the architecture of the hydroponic system and the backup reservoirs.

### 1.3.2 Difficulty

The most difficult part is to calibrate the whole system. The calibration of the nutrient and pH buffer that is added to the main tank needs to be done through trial and error. In addition to that, some of the parts are a bit expensive and unavailable. Getting an industrial sensor is very difficult and thus inaccessible. On the other hand, illiteracy about this topic is damping the hope it brings. Getting the right nutrient solution is also a challenge. Chemicals and good architecture is very difficult to manage in a limited time with limited budget.

## 1.4 Motivation

|  |  |
| --- | --- |
| **BD imported items** | **Value** |
| [Cereals](https://tradingeconomics.com/bangladesh/imports/cereals) | $1.52B |
| [Edible vegetables and certain roots and tubers](https://tradingeconomics.com/bangladesh/imports/edible-vegetables-certain-roots-tubers) | $704.76M |
| [Oil seed, oleagic fruits, grain, seed, fruits](https://tradingeconomics.com/bangladesh/imports/oil-seed-oleagic-fruits-grain-seed-fruit) | $592.23M |
| [Fertilizers](https://tradingeconomics.com/bangladesh/imports/fertilizers) | $1.26B |

Table 1.1: Imported items in Bangladesh with costing

|  |  |
| --- | --- |
| **BD exported items** | **Value** |
| [Cereal, flour, starch, milk preparations and products](https://tradingeconomics.com/bangladesh/exports/cereal-flour-starch-milk-preparations-products) | $84.84M |
| [Vegetable, fruit, nut food preparations](https://tradingeconomics.com/bangladesh/exports/vegetable-fruit-nut-food-preparations) | $70.47M |
| [Vegetable, fruit, nut food preparations](https://tradingeconomics.com/bangladesh/exports/vegetable-fruit-nut-food-preparations) | $70.47M |
| [Pharmaceutical products](https://tradingeconomics.com/bangladesh/exports/pharmaceutical-products) | $70.12M |

Table 1.2: Exported items in Bangladesh with costing

For a developing country this small glimpse of our economy can show us how an agricultural based country like ours is still very much dependent on imported goods. Furthermore, we import 1.26B USD worth of fertilizer. Using hydroponic methods can save millions in this sector alone. We can stop importing foods and that can be grown using this method and instead we can export and grow our economy. This will affect the life of Bangladeshi people like a butterfly effect. The social standard, the living standard and the economical standard of our country can improve beyond our imagination. We will have entrepreneur venturing in social business that not only have positively impact their life but also improves their surroundings.

Meanwhile, we keep up with our Paris Agreement that is there to save the environment. Plant is the only thing that can save our planet. This technique will let us grow more plants with less effort, less time and less space with double the benefit.

## 1.5 Summary

In this chapter, we have briefly described what hydroponic system is and how it can affect Bangladesh with a nudge towards the right direction. The impact and motivation of our project and the difficulties that may occur during the project runtime are also explained in this chapter.

# Chapter 2

# *Related work*

## 2 Related Work

The true source of inspiration for a technique like hydroponic system lies with the construction of the hanging garden of Babylon that was constructed by the king Nebuchadnezzar [12]. However the term was coined around the late 1930s by a professor called Professor William Frederick Gericke of California university, in his book ‘Complete Guide to Soilless Gardening’ [6][18]. Such, an old technique was bound to go through a lot of modification and still there are people who are constantly researching and altering certain factor to get a better result.

Bangladesh, being a river rein country, is very much prone to flood that usually have an immensely negative impact on the agricultural sector. In addition to that, Bangladesh has a deficiency when cultivable land is concern. Thus, the crops produce is not enough to provide for the massive population that exists here. These are issues were analyzed extensively by Shantanu K. Saha et.al. [18]. He further deduced the importance of soilless cultivation a.k.a. Hydroponic system for a country that is in possession of a huge number of water bodies. The paper also discussed the effect of flood and how it can be used to our advantages using a technique called the Hydroponic System. It is also clearly pointed out by Dhananjay S. Jagtap et.al. [11], how hydroponic system can not only conserve landmass but also water. It can grow crops faster but without any kind of contamination. The crops are rather more nutritious. Since it is also easy and time saving, it can turn out to be very economically profitable with huge impact on society.

There are primarily two types of hydroponic system, as discussed by Renu P. et.al. [13]. One that requires a medium and the one that requires nothing but a chemical solution as a source for nutrient, however, we are only working with the latter one in this paper. Among the various solution based hydroponic system such as –the wick system, ebb and flow, drip system, deep water culture etc. that has been discussed in much details by Shreya T. et.al. [23], the one known as the nutrient film technique (NFT) is thought to the most popular technique of all. According to Dhananjay S. Jagtap et.al. [11], any terrestrial plant is suitable for this process to grow. However, the structures are not able to support big woody plants.

Usually for a hydroponic system, certain factors are in need of constant catering. Some of these factors are pH level and nutrient level, water temperature and humidity etc. that allows various research work on finding out the optimum level for crops in different system. Such research works consists of work by Henrique Sánchez et.al.[19] where he concentrate on Ocimum basilicum (a.k.a basil leaf) and introduces a new source of nutrient, vermin compost powder , and observed the result. The resultant plants were healthy and nutritious without much complication. Meanwhile, W.K. Gebremedhin et.al. [8], worked with barley fodder in a semi hydroponic environment as he inspected the yield and chemical composition of barley with each observation and compared it. In another work, brackish water was used for lettuce cultivation to observe how the change affects the hydroponics system [9].

In a similar manner, this fields offers a huge occupation of sorts to the engineers working with automation using internet of things (IoT) and cloud computing. Such research work includes the work by Shreya T. et.al. [23], Rakshitha M. et.al. [12], P Sihombing [20], Dhananjay S. Jagtap et.al. [11], Dr. S. Umamaheswari [25], Dania E. et.al. [7], Vaibhav P. et.al. [13] etc. Most of the automation works are based on arduino. Furthermore a pH sensor and EC (a.k.a. electric conductivity) sensor is used to read and control the pH and nutrient level of the controlled environment. A water level indicator is also used to control the water level in the system. Most of these sensors and indicators are connected to relay and pump, thus making the whole process automated [7][11][13][20][23][25].To further automate the system, Vaibhav P. et.al. [13] and Shreya T. et.al. [23] used arduino as nodes and make use of cloud computing to preserve the data for outside monitoring. Other than automation there are research like Cesar G. Valdivia et.al.[13] who worked with turbulent kinetic energy in NFT system.

Hydroponic system needs a constant monitoring of pH and nutrient level. Anything out of ordinary can pose harm to the plants. In order to control pH level (which usually ranges from 5.5 – 6.5 pH scale) two buffer solutions is used according to the pH meter. For nutrient, each plant as their own preference of nutrient measurement, but most of the nutrient base is same. The usually contain about 17 different nutrients such as - C, H, O, N, P, K, S, Mg, Ca, Fe, Mn, Cu, Zn, B, Cl, Mo and Ni essential. These are used in combination with other solution having cations like Ca2+ , Mg2+ and K+ and the major nutrient anions like NO3−,SO4−, SO2− and H2PO4− [3]. In other papers they discuss these nutrients in details for individual plants like basil leaf [7], lettuce [21], Arabidopsis thaliana [24] etc.

# *Chapter 3*

# *Theory*

## 3. Theory

Hydroponics is the act of growing plants without the use of soil. When growing plants in soil, water serves to make nutrients in the soil soluble and provides oxygenation for the plant's roots. In Hydroponics, however, the nutrients are instead made available directly in the water. In the most abstract sense, a hydroponics system requires a nutrient rich water supply, a means of oxygenating the water supply, and a means to transport the water to the plants. Since nutrients can be expensive, typical hydroponics systems are recirculating – the runoff from transporting water to the plants returns to the original water source. In the following, we discuss the main parameters that should be considered in a hydroponics system. This description is not intended to be complete, but rather gives an overview of the parameters that are most likely to affect a hydroponics system.

Water chemistry refers to the makeup of a water source. In hydroponics, we chiefly care about three factors: pH, concentration of Macro-nutrients, and oxygenation. pH is important due to nutrient lockout. The various nutrients within a water source are more easily absorbed by a plant's roots based on the pH of the water source – too basic or too acidic, and the plant will not be able to absorb the nutrients from the water, even if they are present. The desired pH range will vary depending on the plants in the hydroponics system, but typical values are between 5.5 and 6.5.

Macro-nutrients refer to Nitrogen, Phosphorus, and Potassium (fertilizers typically list these as relative concentrations: 10-10-10 means the nutrients are balanced) – plants use these three in greater concentrations than other nutrients like Iron and Magnesium, which are called Micro-nutrients. The particular concentrations your system will need depends heavily on the plants you plan on growing. Leafy vegetables like lettuce and cucumbers need a higher concentration of Nitrogen, whereas others (like tomatoes) require a more balanced mixture. In any case, hydroponics systems use special nutrient mixes designed for hydroponics – regular fertilizers are not water soluble enough for use in hydroponics, and will cause buildups that can be difficult to get rid of. As the plants use up the nutrients in the water, more fertilizer will need to be added. The common way of detecting when it is time to add more is with an Electrical Conductivity (EC) Meter. These meters measure how conductive a solution is. This is important for hydroponics because we typically use distilled water (or de-ionized water) and add nutrient solution. Since the nutrient solution is the only source of ions, EC is a reasonable measure of nutrient concentration. EC is not a direct measure of nutrient concentration, however, and if other sources of ions make their way into the water source, then EC will not be a very accurate measure of nutrient concentration.

Finally, oxygenation is a critical factor in the water source. If the water has low dissolved oxygen, the roots of a plant can 'drown'. In container gardening, this manifests as over watering, and the symptoms are very similar to under watering. Since hydroponics systems lack soil to retain water, the major concern is under oxygenated water. Luckily, we don't need to worry about over oxygenating the water. For the majority of setups, a small air pump with an air stone is sufficient – check the manufacturer's suggested gallon rating when choosing an air pump. Oxygen can also be introduced when the runoff returns to the water source if there is a waterfall (or the water free falls through air).

Watering cycles do not seem to be well researched, but the common wisdom states a 15-on-45-off cycle. This can be done all day, or the cycle can be reduced at night. However, certain hydroponics setups require constant water flow – Nutrient Film Technique (NFT) is an example of constant water flow. For more traditional systems, the cycling allows the water to drain completely from the roots of the plant – allowing the roots to come into contact with air directly – before the next cycle begins. This can also help prevent mold buildup on the roots. In any setup, drainage is very important. It is not typically an issue with drip-line or NFT systems, since the water can immediately drain back to the reservoir, but ebb-and-flow systems must typically employ a siphon or external pump so that the grow-bed can fill up before draining completely.

Finally, lighting is important for photosynthesis. For outdoor hydroponics systems, the available lighting is usually sufficient. However, indoor systems almost always lack enough sunlight for proper plant growth. Furthermore, indoor lights typically do not emit the proper wavelengths that plants use for photosynthesis. There are many commercially available lights that do produce the light plants need, and some of the newer lights use LEDs, which produce much less heat than older bulbs. The bulb should be placed directly over the plants, but care should be taken that the bulb is not too close so that any heat produced will not affect the plants. Common lighting cycles seem to match the sun cycle – about 12 on, 12 off. While it is very important that the plants receive enough light, it is also important that the reservoir gets no light – algae can grow in the reservoir when introduced to light. While algae is not directly harmful to the plants (though algae buildup on roots can be detrimental), the biggest impact is to the nutrient concentration. Since algae use nutrients just like your plants do, if algae begins to grow in your reservoir, you will need to add nutrients more often to feed your plants. Since this can become expensive, it is vital that as little light as possible is permitted to enter the reservoir.

There are many more parameters that can be monitored in a hydroponics system, and here we presented the most obvious ones. If the suggestions described above are followed, there is a strong chance that the system will work well. Tweaking additional parameters (such as humidity) may optimize plant growth, but is not necessary for creating a strong, productive hydroponics system.[22]

# *Chapter 4*

# *Structure of the system*

## 4.1 Introduction

In our system architecture, there are basically three part

* NFT System (Hydroponic system)
* Automated IoT based System
* Remote monitoring using an App

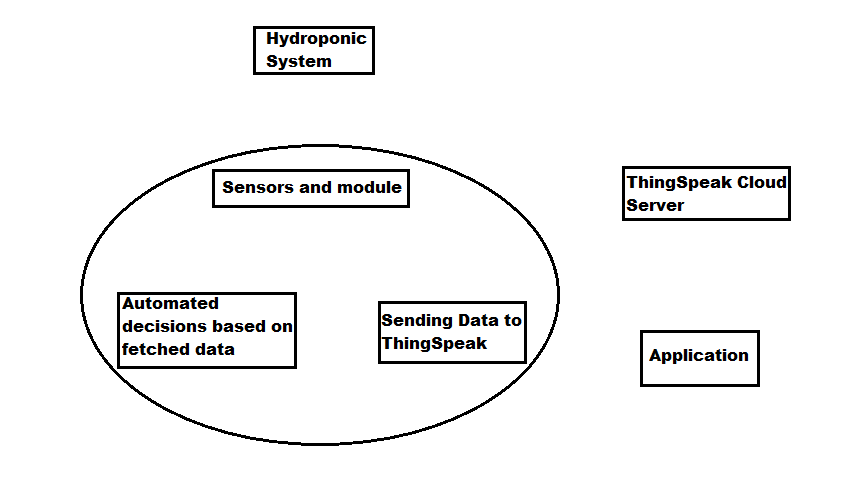


Fig 4.1: System Block Diagram

## 4.2 NFT System (Hydroponic system)

In the agricultural part, the architecture of the system depends on the owner. However, in our prototype we applied the following architecture where the water circulates all the pipe and returns to the tank. This process occurs over and over again.



Fig 4.2: NFT System Structure

There is a one big container that contains all the nutrient solution for the plant. A pump and an oxygen pump is continuously working inside the container.

## 4.3 Automated IoT based System

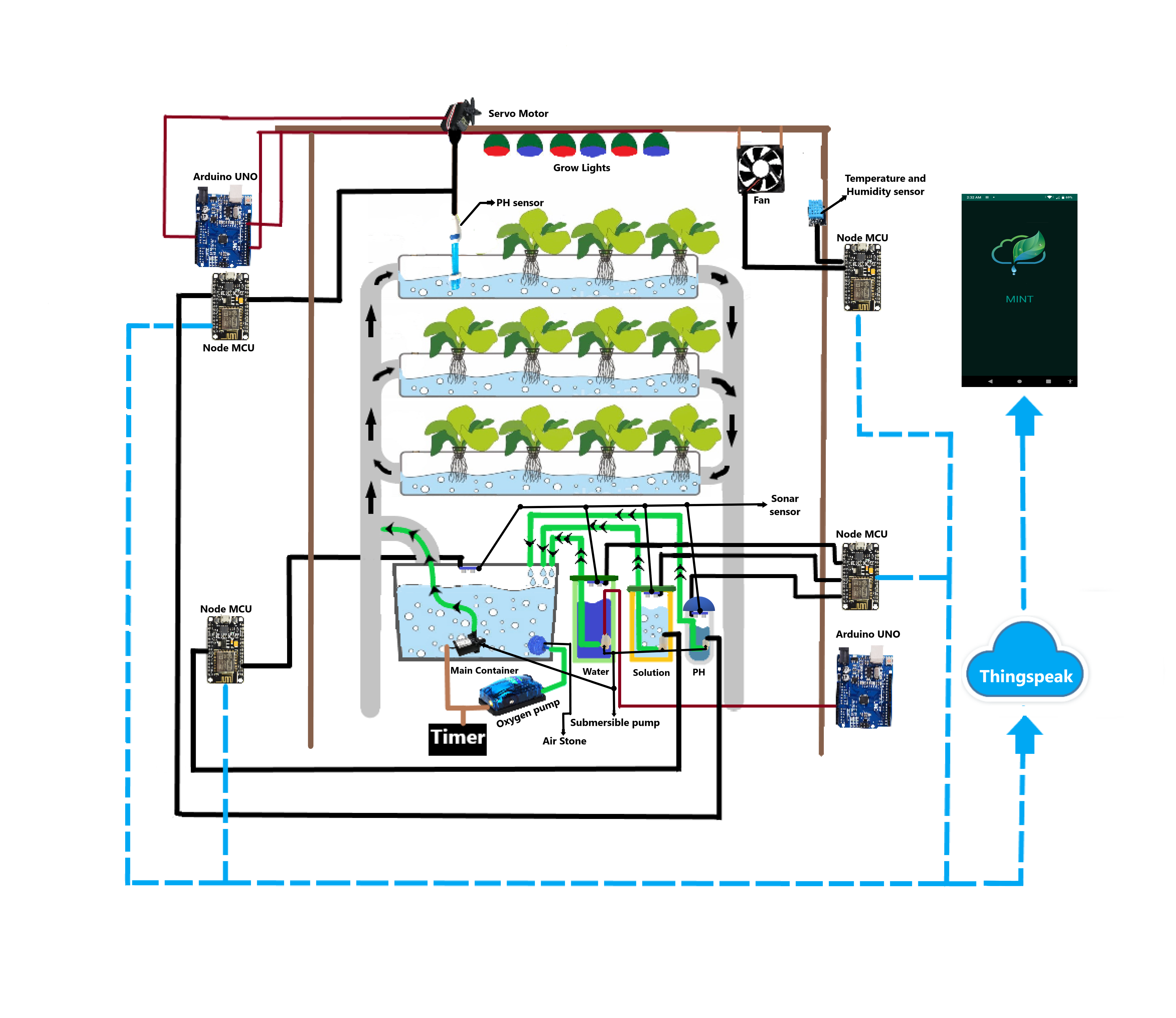
In this part we used different sensors to get data from the agriculture part of the whole system. These data is sent to ThingSpeak cloud server via a WiFi module. There are three reservoirs in

Fig 4.3: System Architecture

this portion along with the major container. These three reservoirs contain nutrient solution, pH down solution and water. Sonar sensors are attached on the lead of all the containers. Inside the reservoirs, there are small submersible pumps connected to the main container with the pipes. There are grow lights on the top of structure for the plants to grow indoors. A servo motor is also attached on the top which is connected with to the pH sensor. An exhaust fan is connected at the back of the structure and a DHT22 sensor is also attached on one side. All these sensors, small submersible pumps and other electrical equipment are connected to four Node MCUs in total and two Arduino Uno. The Node MCUs are used to upload the collected data to the cloud server. The Node MCUs are used to upload the collected data to the cloud server.

## 4.4 Remote monitoring using an App

An app was made that can fetch data from ThingSpeak cloud server. The main objective of the app is to update the users knowledge about his/her garden so that they know when to add more of the solution, water or pH down solution to the system. The app also includes information about different plants and their required nutrient level and pH level.

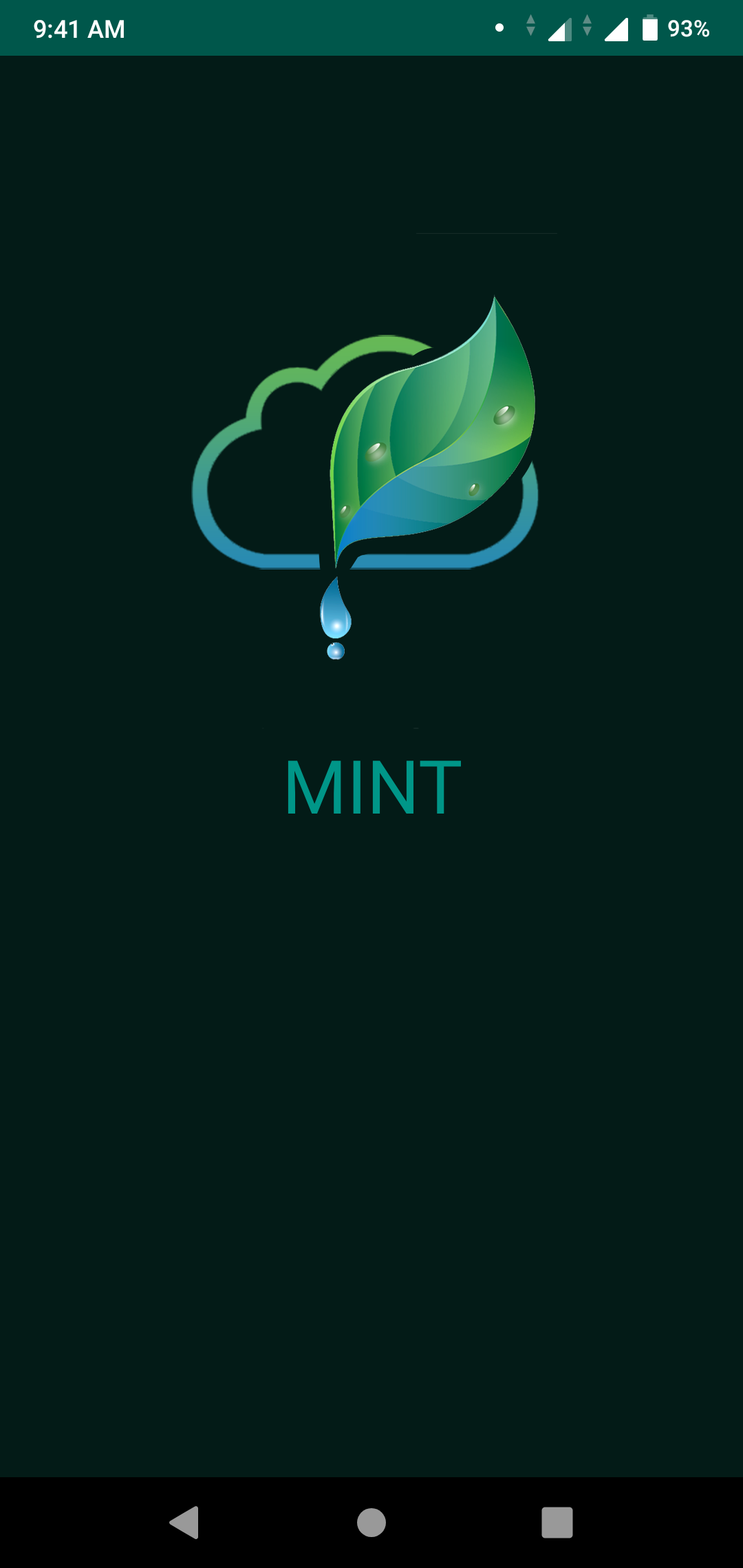
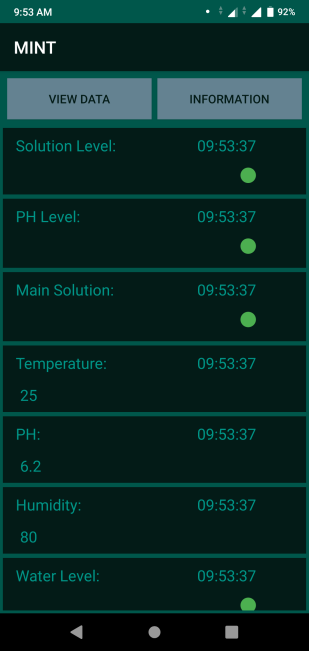


Fig 4.4: Customized App called ‘MINT’

The app that we made is called ‘MINT’. This app is mainly a information based app. The first screen on the very left if the splash screen. The screen in the middle is the data screen. This screen updates in every five minutes. The last screen contains the information about EC and pH level of different plants.

# Chapter 5

***Technical Exposition***

## 5.1 Introduction

This chapter describes the components that were used to implement this project. This chapter especially focuses on the automation using IoT and cloud computing part of the project. For this we needed the following equipment, server and applications.

* Node MCU with ESP8266 WiFi Module
* Arduino Uno R3
* Ultrasonic Ranging Module (hc sr04)
* pH sensor with BNC connector and signal conversion module
* Temperature and Humidity module (DHT22 AM2302)
* Grow light
* Oxygen pump
* DC Fan (12V)
* Servo Motor MG996R (180 Degree)
* Submersible pump (SP2500)
* Micro Submersible pump (6V)
* Relay
* Applications

# 5.2 Node MCU with ESP8266 WiFi Module

In this article, we discuss in details about NodeMCU (Micro Controller Unit). NodeMcu came out after ESP8266, which is a WiFi module, was produced in 2013. It is open source platform and Hong Kong was the first one to commit the first file of nodemcu-firmware to GitHub. Afterward, an open source Hardware was started to manufacture. In summer 2015 the creators abandoned the firmware project and a group of independent contributors took over. By summer 2016 the NodeMCU included more than 40 different modules.

As *Arduino.cc*began developing some new MCU boards based on non-AVR processors like the ARM/SAM MCU and the one used in the Arduino. They needed to modify the Arduino IDE so that it would be relatively easy to change the IDE to support alternate tool chains to allow Arduino C/C++ to be compiled for these new processors. They did this with the introduction of the Board Manager and the SAM Core. A "core" is the collection of software components required by the Board Manager and the Arduino IDE to compile an Arduino C/C++ source file for the target MCU's machine language. Even though there are multiple boards the following features usually stay common: It has the following specification:

Memory: 128kBytes

Developer: ESP8266 Opensource Community

Operating system: XTOS

Storage: 4MBytes

CPU: ESP8266 (LX106)

Power: USB

# 5.2.1 NodeMCU Lua V3 ESP8266 WiFi Module with CH340G

In our project we used NodeMCU ESP8266 WiFi module as an IOT platform that made our circuit simple and effective. It is an open-source firmware and development kit that works as an IOT platform within a few Lua script lines. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266 and uses hardware which is based on the ESP-12 module.. It uses many open source projects, such as lua-cjson and SPIFFS. In another word, it has an ESP8266 chip on board so that we can easily write and use the WiFi module to our advantage, all the while performing simple Arduino code.

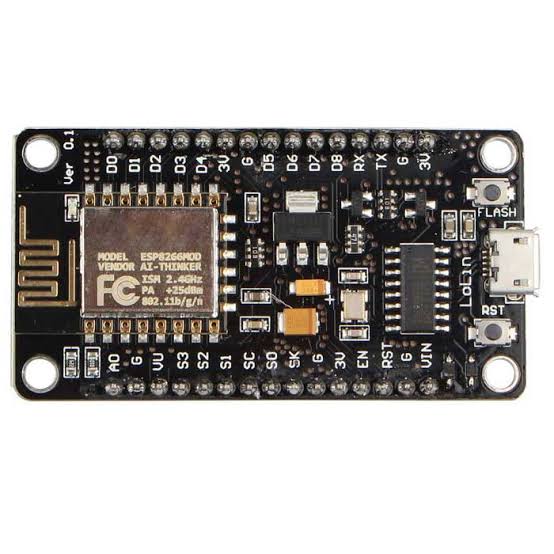


Fig 5.1 NodeMcu Lua V3 ESP8266 WiFi with CH340G

ThingSpeak allows one to aggregate, visualize and analyze live data streams in the cloud. Some of the key capabilities of ThingSpeak include the ability to:

* Easily configure devices to send data to ThingSpeak using popular IoT protocols.
* Visualize sensor data in real-time.
* Aggregate data on-demand from third-party sources.
* Use the power of MATLAB to make sense of IoT data.
* Run IoT analytics automatically based on schedules or events.
* Prototype and build IoT systems without setting up servers or developing web software.

# 5.2.2 Pin Configuration

As the ESP8266 chip is set on the board, and with the help of Arduino we are now able to directly upload codes on the chip using C/C++ etc. The board contains about 30 pins with multiple GND(Ground) and power outlet of ~3.3V. It has one pin that supplies ~5V and about 8 digital/GPIO pins and 1 analog pin. Other pins includes enabler, reset, flash, transmitter(TX) and receiver(RX) pin. There is an antenna for WiFi connection.

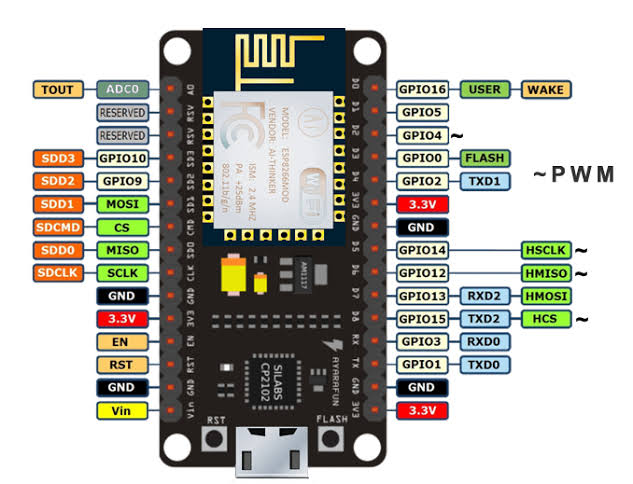


Fig 5.2: NodeMCU with pin configuration.

## 5.3 Arduino Uno R3

Arduino is an open-source computer hardware and software company, project and user community that designs and manufactures microcontroller-based kits for building digital devices and interactive objects that can sense and control objects in the physical world. The project is based on microcontroller board designs, manufactured by several vendors, using various microcontrollers. These systems provide sets of digital and analog I/O pins that can be interfaced to various expansion boards ("shields") and other circuits. The boards feature serial communications interfaces, including USB on some models, for loading programs from personal computers. For programming the microcontrollers, the Arduino platform provides an integrated development environment (IDE) based on the Processing project, which includes support for the C, C++ and Java programming languages.

The first Arduino was introduced in 2005, aiming to provide an inexpensive and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats, and motion detectors.

### 5.3.1 Description and Configuration

The microcontroller we used in our system is Arduino Uno R3. Arduino boards are available commercially in preassembled form, or as do-it-yourself kits. The hardware design specifications are openly available, allowing the Arduino boards to be manufactured by anyone. Adafruit Industries estimated in mid-2011 that over 300,000 official Arduinos had been commercially produced, and in 2013 that 700,000 official boards were in users' hands.

An Arduino board consists of an Atmel 8-, 16- or 32-bit AVR microcontroller with complementary components that facilitate programming and incorporation into other circuits. An important aspect of the Arduino is its standard connectors, which lets users connect the CPU board to a variety of interchangeable add-on modules known as *shields*. Some shields 28 communicate with the Arduino board directly over various pins, but many shields are individually addressable via an I²C serial bus—so many shields can be stacked and used in parallel. Official Arduinos have used the megaAVR series of chips, specifically the ATmega8, ATmega168, ATmega328, ATmega1280, and ATmega2560. A handful of other processors have been used by Arduino compatibles. Most boards include a 5 V linear regulator and a 16 MHz crystal oscillator (or ceramic resonator in some variants), although some designs such as the Lily Pad run at 8 MHz and dispense with the onboard voltage regulator due to specific form-factor restrictions. An Arduino's microcontroller is also pre-programmed with a boot loader that simplifies uploading of programs to the on-chip flash memory, compared with other devices that typically need an external programmer. This makes using an Arduino more straightforward by allowing the use of an ordinary computer as the programmer. Currently, optiboot boot loader is the default boot loader installed on Arduino UNO R3.

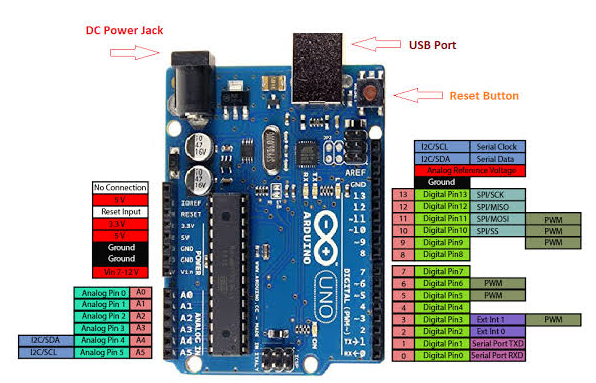


Fig 5.3 Arduino Uno R3 with pin configuation

## 5.4 Ultrasonic Ranging Module HC - SR04

Ultrasonic ranging module HC - The HC-SR04 is an inexpensive distance sensor. It is based on a pair of ultrasonic transducers with a straightforward TTL (Transistor–transistor logic) level interface. It claims to be able to measure a range of 2cm to 4m. However, in practical use it accurately detects up to 80cm. The module is operational at a voltage of +5v and a current of <15mA. The module consists of an ultrasonic transmitter, a receiver and a control circuit. [2]

The basic principle of work:

(1) Using IO trigger for at least 10us high level signal,

(2) The module is built to send eight 40 Hz and detect if there is any pulse signal that bounced back.

(3) If the receiver receives any signal, through high level then the time of high output IO duration is the time from sending ultrasonic signal to receiving ultrasonic signal.[2]

Fig 5.4: Working of a Ultrasonic sonar sensor

We use the following formula to calculate the distance:

We take the speed of sound as 440 m/s (approximately).

Important Features:

* Operating voltage: +5V
* Theoretical  Measuring Distance: 2cm to 450cm
* Practical Measuring Distance: 2cm to 80cm
* Accuracy: 3mm
* Measuring angle covered: <15°
* Operating Current: <15mA
* Operating Frequency: 40Hz

|  |  |  |
| --- | --- | --- |
| **Pin Number** | **Pin Name** | **Description** |
| 1 | VCC | The Vcc pin powers the sensor, typically with +5V |
| 2 | Trigger | Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave. |
| 3 | Echo | Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor. |
| 4 | Ground | This pin is connected to the Ground of the system. |

Table 5.1: Ultrasonic Sensor Pin Configuration

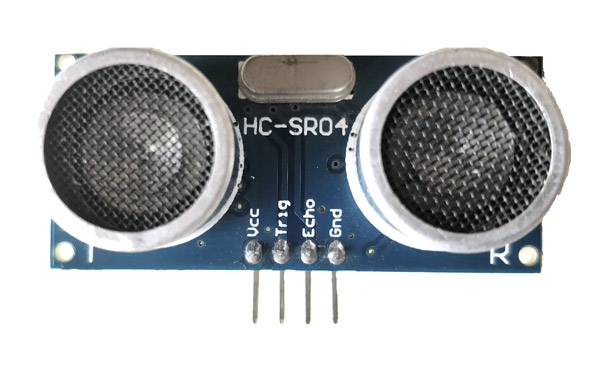
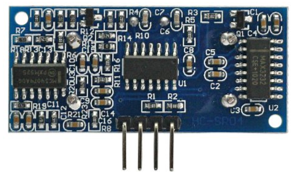


Fig 5.5: Ultrasonic Sonar Sensor: Front View (left); BackView(right)

## 5.5 pH sensor with BNC connector and signal conversion module

pH is the Hydrogen ion (H+) concentration index in a logarithmic scale. It determines whether a particular liquid is acidic or alkaline. The range of the scale is from 0-14 where 7 is regarded as neutral with 14 being the most alkaline and 0 being the most acidic.

The pH sensor is used to measure the acidity or alkalinity of any liquid it is dipped into. For this project we are using it to check the pH value of our solution for the plants. The sensor that we used can be easily calibrated as it can automatically identify two standard buffer solutions (4.0 and 7.0). The sensor comes with a BNC (Bayonet Neill–Concelman) connector that we connect to the signal conversion. It gives hardware filtered output signal, thus, low jitter. An on board regulator also allows the sensor to be working in a range of 3.3v to 5v. The accuracy of the sensor is about 0.1 at 25C. The sensor sends a pulse that we convert into a pH reading using an equation. To increase the precision of our reading we take multiple pulse of a short time and take their average. The result is used as an input in the equation[5]. We used the following equation to calculate the pH value from the received pulse:

Important Features:

Signal Conversion Board (Transmitter) V2

* Supply Voltage: 3.3~5.5V
* Output Voltage: 0~3.0V
* Probe Connector: BNC
* Signal Connector: PH2.0-3P
* Measurement Accuracy: ±0.1 at 25℃

pH Probe

* Detection Range: 0~14
* Temperature Range: 5~60°C
* Zero Point: 7±0.5
* Response Time: <2min
* Internal Resistance: <250MΩ

|  |  |  |
| --- | --- | --- |
| **Pin** | **Label** | **Description** |
| 1 | - | Power GND(0V) |
| 2 | + | Power VCC(3.3~5.5V) |
| 3 | A | Analog Signal Output(0~3.0V) |
| 4 | BNC | pH Probe Connector |

Table 5.2: pH sensor Pin Configuration

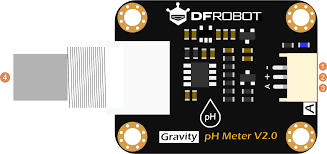


Fig 5.6: Signal Converter board Fig 5.7: pH sensor with BNC and signal converter

## 5.6 Temperature and Humidity module (DHT22 AM2302)

In order to measure the temperature and humidity, DHT22 sensor is used. The DHT22 sensor is made of two parts, a capacitive humidity sensor and a thermistor. There is a chip inside that acts as an analog to digital converter and produces a digital signal with the temperature and humidity. The sensor is cheap and easily available.[1]

Features:

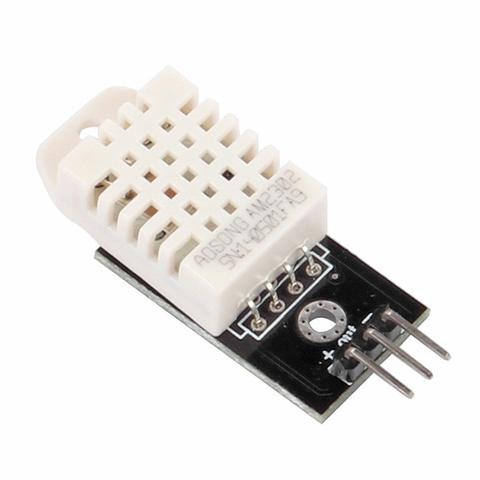
* Operating Voltage: 3.5V to 5.5V
* Operating current: 0.3mA (measuring) 60uA (standby)
* Output: Serial data
* Temperature Range: -40°C to 80°C
* Humidity Range: 0% to 100%
* Resolution: Temperature and Humidity both are 16-bit
* Accuracy: ±0.5°C and ±1%

Fig 5.8: Temperature and Humidity Sensor DHT22

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pin |  | Label |  | Description |
| 1 |  | Vcc(+) |  | Power supply 3.5V to 5.5V |
| 2 |  | Data(out) |  | Outputs both Temperature and Humidity through serial Data |
| 3 |  | Ground(-) |  | Connected to the ground of the circuit |

Table 5.3: DHT22 module Pin Configuration

## 5.7 Grow light

Grow lights are usually used to ensure healthy growth of plants when kept indoors. The color, intensity and duration vary from plant to plant. However, most plants needs at least 8-12 hours of light if kept indoors. Most of the time florescent light with full spectrum of natural light is good to use, albeit the color red and blue are more frequently used than the others. Sometimes, green are mixed with red and blue are also used.[9] Fig 5.9: Grow Light

## 5.8 Oxygen Pump

An oxygen pump is used to supply oxygen to the water so that the plants can acquire the extra oxygen it needs to grow better. Using oxygen pump enables better and healthier plants

Fig 5.10: Oxygen Pump

## 5.9 DC Fan (12V)

The DC Fan is used as an exhaust fan that is activated based on the temperature and humidity of the system.

Important Feature:

Voltage rating: 12V DC

Current Rating: 0.20A

Connector: 2 pin

Dimension: 80mm x 80mm x 25mm (Approx. 3" x 3.1") Fig 5.11: DC Fan

## 5.10 Servo Motor MG996R (180 Degree)

Servo MG996 is the successor of MG995, servo motor with full rotation (180 degrees). [16]

Important Features:

Metal Gear

Rotational angle: 180o

Stall torque: 9.4kg/cm (4.8v); 11kg/cm (6.0v)

Operating speed: 0.19sec/60degree (4.8v); 0.15sec/60degree (6.0v)

Operating voltage: 4.8 – 6V

Gear Type: All Metal Gears

Temperature range: 0- 55oC Fig 5.12: Servo Motor

Dead band width: 1us

Wire length: 32cm

Aluminum 6061-T6

Weight: 55g

## 5.11 Submersible pump (SP2500)

The submersible pump is operational at 220V-240V AC current. It uses a stainless steel shaft and durable magnetic rotor. Its low operating noise and energy efficient with little maintenance. It works with a power of 18W that can pump water up to 1.5 meter height. Fig 5.13: SP2500 submersible pump

## 5.12 Micro Submersible pump (6V)

It is a small submersible pump for adding solution in a controlled manner from a backup reservoir. [17]

Features:

Voltage: DC 2.5-6V

Working Current: 130-220mA

Power: 0.4-1.5W

Maximum lift: 40-110cm / 15.75"-43.4"

Flow rate: 80-120L/H

Driving mode: brushless dc design, magnetic driving

Outside diameter of water outlet: 7mm

Outside diameter of water inlet: 6.8mm

Diameter: Approx. 24mm / 0.95"

Length: Approx. 56mm Fig 5.14: Micro Submersible pump

Height: Approx. 33mm

Material: Engineering plastic

## 5.13 Relay

As we have used motors that take 12V to power up, we used relays for switching purpose. A relay is an electrically operated switch. Usually relays use an electromagnet to mechanically trigger a switch, but other operating principles are also used, such as solid-state relays. Relays are used where it is necessary to control a circuitry by a low-power signal (with complete electrical isolation between control and controlled circuits), or where several circuits must be controlled by one signal. The first relays were used in long distance telegraph circuits as amplifiers. Relays were used extensively in telephone exchanges and early computers to perform logical operations.[10]

Fig: 5.15: 1 channel relay

Relays are switches that open and close circuits electromechanically or electronically. Relays control one electrical circuit by opening and closing contacts in another circuit. As relay diagrams show, when a relay contact is normally open (NO), there is an open contact when the relay is not energized. When a relay contact is Normally Closed (NC), there is a closed contact when the

relay is not energized. In either case, applying electrical current to the contacts will change their state. Relays are generally used to switch smaller currents in a control circuit and do not usually control power consuming devices except for small motors and Solenoids that draw low amps. Nonetheless, relays can “control” larger voltages and amperes by having an amplifying effect because a small voltage applied to a relays coil can result in a large voltage being switched by the contacts. Protective relays can prevent equipment damage by detecting electrical abnormalities, including overcurrent, undercurrent, overloads and reverse currents. In addition, relays are also widely used to switch starting coils, heating elements, pilot lights and audible alarms.[10]

Fig 5.16: Internal working of a relay

## 5.14 Servers and Application

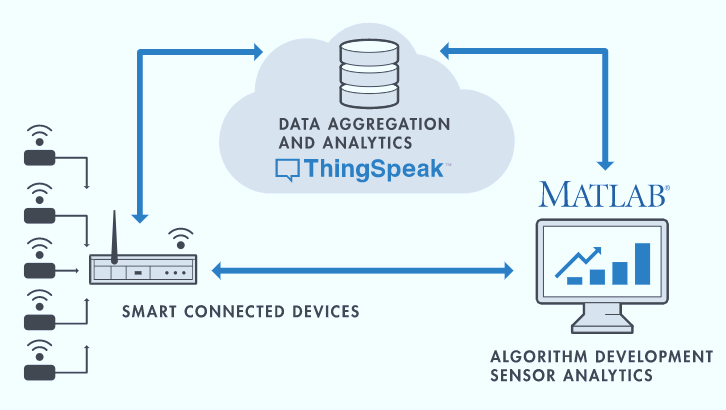
We are using the following application:

* Thingspeak
* Arduino IDE
* Android Studio

## 5.14.1 Thingspeak

For IoT, we have used ThingSpeak cloude server. Internet of Things (IoT) is used to describe an emerging trend where a large number of embedded devices (things) are connected to the Internet. These connected devices communicate with people and other things and often provide sensor data to cloud storage and cloud computing resources where the data is processed and analyzed to gain important insights. Cheap cloud computing power and increased device connectivity is enabling this trend.

## It is an IoT analytics platform service that allows one to aggregate, visualize and analyze live data streams in the cloud. ThingSpeak provides instant visualizations of data posted by devices used to ThingSpeak. With the ability to execute MATLAB code in ThingSpeak, online analysis and processing of the data is possible .ThingSpeak is often used for prototyping and proof of concept IoT systems that require analytics.

Fig 5.17: ThingSpeak Architecture

ThingSpeak allows one to aggregate, visualize and analyze live data streams in the cloud. Some

of the key capabilities of ThingSpeak include the ability to:

* Easily configure devices to send data to ThingSpeak using popular IoT protocols.
* Visualize sensor data in real-time.
* Aggregate data on-demand from third-party sources.
* Use the power of MATLAB to make sense of IoT data.
* Run IoT analytics automatically based on schedules or events.
* Prototype and build IoT systems without setting up servers or developing web software

In one ThingSpeak Channel we can create about 8 different fields that can fetch 8 different data. They store and graph these data. Exporting previous data or importing new data is also possible.

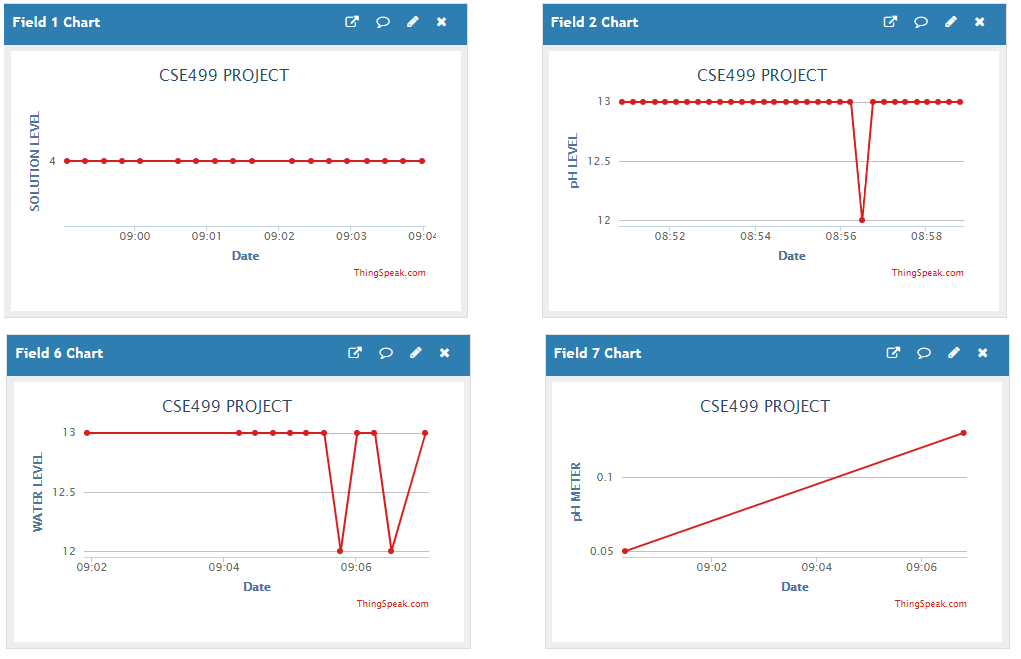
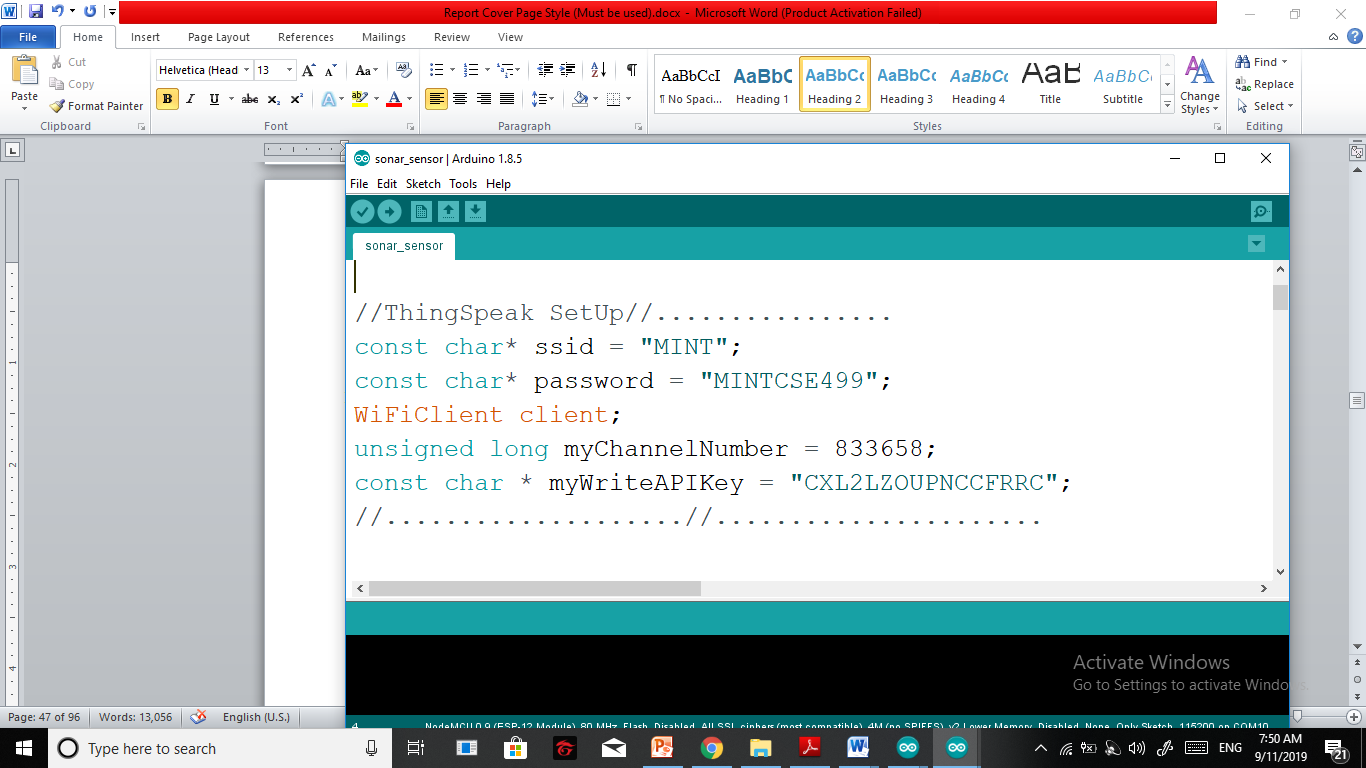


Fig 5.18: Data Analysis in ThingSpeak

Data can be both send and fetched from here and this process can be customized according to the user. The following is the code to set up thinkspeak to connect to nodemcu.



## 5.14.2 Arduino IDE

We used the Arduino IDE (integrated development environment) to write, compile and upload code in Arduino Uno and NodeMCU. We used C++ language for this purpose. The Arduino IDE is a cross-platform application that is written in the programming language Java. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.

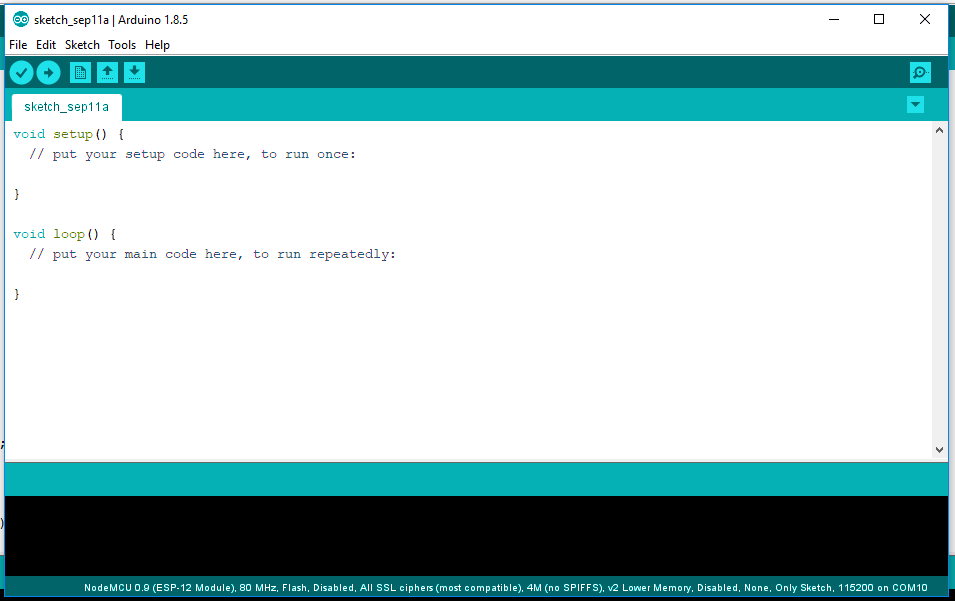


Fig 5.19: Arduino IDE (screenshot)

## 5.14.3 Android Studio

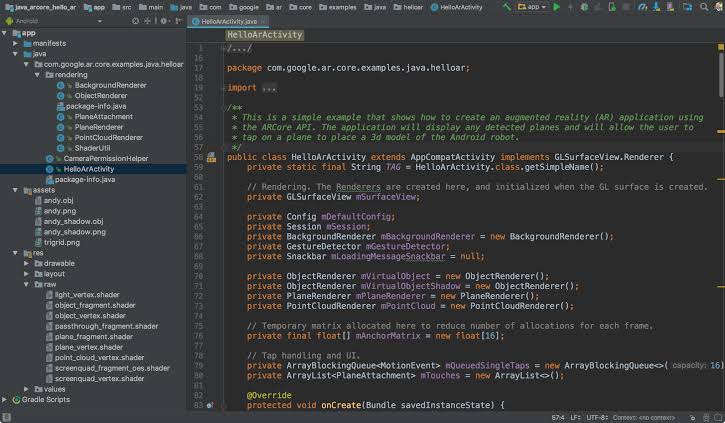
We used Android Studio to create a customized app called MINT. Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems.

Fig 5.20: Android Studio(Screen Shot)

## 5.15 Summary

In conclusion, in this chapter all the crucial technical aspects were discussed in detailed. The next phase is to bring these components to together to finish our project. These components are used according to the given data. This will be discussed about in details in the next chapter.

# Chapter 6

# Design Implementation

## 6.1 Introduction

There are mainly seven separate parts of our project. These are:

* Controlling the pH value
* Controlling the amount of solution in the main container
* Controlling the humidity and temperature
* Getting the notification in our mobile application when the reservoirs are empty
* Controlling the servo motor and the grow light
* Controlling the concentration of the nutrients in the main container
* Controlling the main submersible pump and the oxygen pump

## 6.2 Required Equipment

We require the following things to implement our project:

## 6.2.1 Required Equipment for design implementation

1. Soldering Iron

2. Wire cutter

3. Power Cables/ Data Cables with adapter

4. Screw and bolts

5. Glue gun

6. Flexible pipe

7. Digital Multi meter

8. Thread

9. PVC pipe

10. Hard Board

11. PVC board

## 6.2.2 Required Equipment and component for internal circuitry

1. Node MCU with ESP8266 WiFi Module
2. Arduino Uno R3
3. Ultrasonic Ranging Module (hc sr04)
4. pH sensor with BNC connector and signal conversion module
5. Temperature and Humidity module (DHT22 AM2302)
6. Grow light
7. Oxygen pump
8. DC Fan (12V)
9. Servo Motor MG996R (180 Degree)
10. Submersible pump (SP2500)
11. Micro Submersible pump (6V)
12. Relay

## 6.3 Controlling the pH value

There are different ranges of pH which are suitable for different types of hydroponic plants to grow. The pH range suitable for the plants we grew is from 5.5 to 6.5. To maintain this pH level, we use the pH sensor and a submersible pump which is inside the pH solution reservoir. These two are connected to a Node MCU to upload the pH value to the cloud server and also to control the submersible pump based on the pH value. For hydroponic system, the pH of the solution actually increases. So, when the pH value goes above the suitable range, the submersible pump is turned on by using a relay and the pH down solution is added to the main container from the pH solution reservoir to bring the pH value within an appropriate range.

## 

Fig 6.1:pH sensor circuit

## 6.4 Controlling the amount of solution in the main container

A sonar sensor is attached to the top of the lead of the main container to measure the amount of solution present in it. This sensor is connected to a Node MCU which is also connected to the submersible pump inside the solution reservoir. The Node MCU uploads the sonar sensor value in the cloud server and also controls the submersible pump. When the amount of solution in the reservoir goes below a certain level, the submersible pump is turned on using a relay and the main container is refilled with the solution from the solution reservoir.

## 6.5 Controlling the humidity and temperature

A DHT22 sensor is used to measure the temperature and humidity of the system. This sensor is connected to a Node MCU to upload the data to the cloud server and control the exhaust fan. The exhaust fan is turned on when the humidity goes above 88.

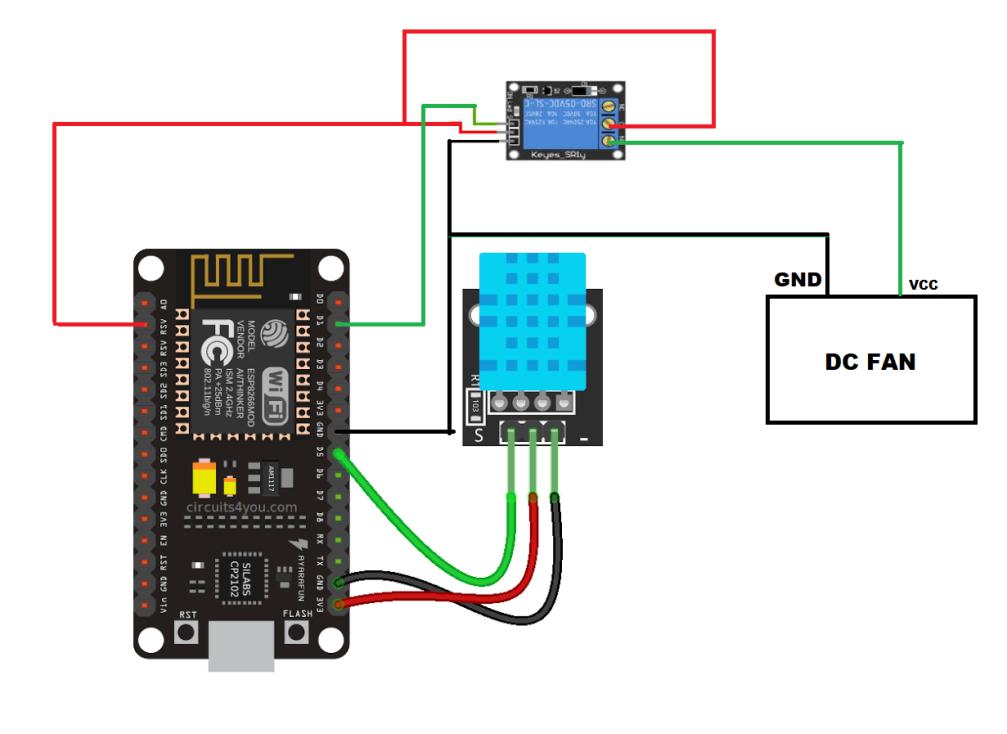


Fig 6.2: Controlling the humidity and temperature

## 6.6 Getting the notification in our mobile application when the reservoirs are empty

Sonar sensors are attached on the lead of the reservoirs to measure the amount of solution or water that is present in the reservoir. All these data are uploaded to the cloud server by using a Node MCU. By our mobile application, we can get all the real time data from the cloud server. It also gives us notifications when the reservoirs get empty and are needed to be refilled. Circuit is according to Fig 6.3.

## 6.7 Controlling the servo motor and the grow light

The servo motor is used to dip the pH sensor in the solution for taking the reading and picking it up again after a certain interval of time. It is done because the crystal at the bottom of the pH sensor can get damaged if it is kept in the solution for a long time. The grow light is used to provide sufficient amount of light for the plants to grow in indoor. The grow light is kept on for 9 hours a day. The grow lights and the servo motor are controlled by an Arduino Uno.

## 6.8 Controlling the concentration of the nutrients in the main container

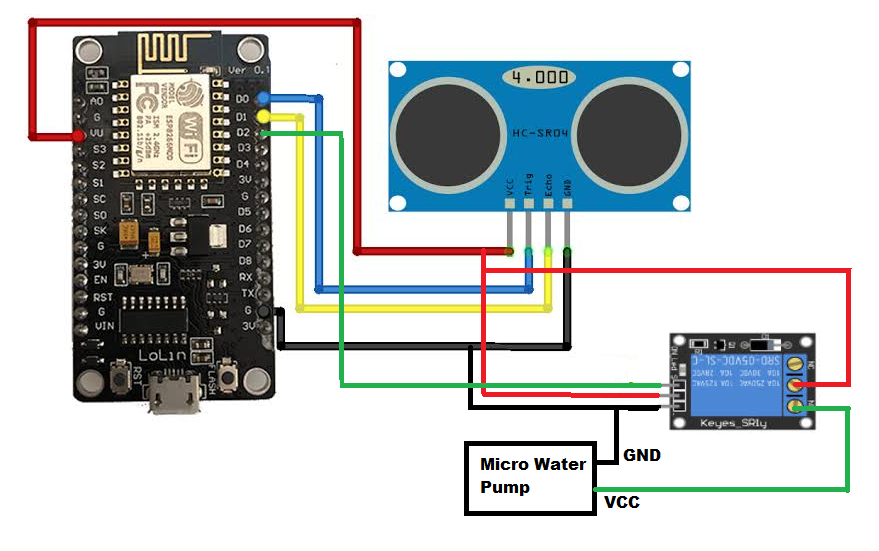
After somedays, the concentration of the nutrients in the main container increases above an appropriate level. To keep the concentration within the range, water is added by using a submersible pump which is in the water reservoir after every seven days and this is done by using an Arduino Uno.

Fig 6.3: Sonar Sensor circuit

## 6.9 Controlling the main submersible pump and the oxygen pump

The big submersible pump in the main container is used to supply the solution to the whole structure. The air stone and the air pump are used to add oxygen to the solution. These two pumps are controlled by using a timer. These are turned on after every one hour and are kept on for thirty minutes.

# Chapter 7

# *Compliance with standards*

## 7.1 Introduction

There are several international standards which a system should meet, among which the IEEE, US and European standards are noteworthy to mention. The compliance of our designed system to all these standards is discussed in this chapter.

## 7.2 Compliance with IEEE standards

There are a few distinct guidelines put forward by IEEE Standards affiliation. The equipment we have used for our project are also used in most of the IEEE standard projects like Arduino Uno, sonar sensor, submersible pump, Node MCU etc.. Our system is beneficial for the environment and is also requires low electricity. Except for a water pump all the other equipment requires only 5V worth of electricity. Beside, as per the Paris Agreement this system is rather lucrative.

## 7.3 Compliance with US standards

In case of US, there is no such guideline that dictates the standard use of IoT in Hydroponic system. There are, however, standard for the plants grown, which does not affect our project as it does not hamper any pants life or healthy growth. Our system is human friendly and is beneficial to both the humanity and the environment.

## 7.4 Compliance with European standards

There are no European protocol our system, albeit there are some protocol that dictates the use of IoT devices which can only make our devise ore durable. And turn our prototype into reality.

# Chapter 8

# *Design Impact*

## 8.1 Introduction

The different ways in which our designed project leaves an impact, and how its manufacturability and sustainability may be is discussed in this chapter.

## 8.2 Economic impact

With a technology that can make a lucrative idea like hydroponic system that solves a lot of the problem of our country, easy and accessible is itself a business plan that can have major effect on our economy. Nevertheless, the idea of pulling our agricultural sector from the abyss of technological backwardness is quiet thought provoking. The following economic effect can be expected from commercial hydroponic:

1. Less labor intensive due to **[automated controls](https://powerhousehydroponics.com/future-of-food-plant-factory-technology/" \t "_blank" \o "The Future of Food: Plant Factory Technology)** and harvests

2. Controllable operating costs through conserved **[water](https://powerhousehydroponics.com/innovative-water-saving-methods-in-hydroponics/" \t "_blank" \o "Innovative Water Saving Methods in Hydroponics)** and **[energy](https://powerhousehydroponics.com/energy-saving-methods-hydroponics/" \t "_blank" \o "Energy Saving Methods in Hydroponics)** consumption

3. Economies of scale yield more crops in less time

4. Producing better quality crops adds value to the produce

5. **[Repurposing abandoned buildings](https://powerhousehydroponics.com/urban-renovation-hydroponic-retrofits-empty-buildings/" \t "_blank" \o "Urban Renovation: Hydroponic Retrofits in Unused Buildings)** improves local marketplace

6. Urban food production cuts down on **[shipping costs](https://powerhousehydroponics.com/5-ways-hydroponics-improves-food-supply-chain/" \t "_blank" \o "5 Ways Hydroponics Improves Food Supply Chain)**

7. Greater market acceptance means people are willing to pay for it

8. Technology is available today to automate and maximize efficiency

## 8.3 Environmental impact

Some of the environmental impacts are given below:

1. Less Land Is Needed :

With traditional farming, we can be as compact as we did like, but it remains a fact that every plant you put into the ground occupies a fixed space that doesn’t change throughout the maturation period of the plant.

This is different when using hydroponics. Grow pots can be placed closer together, without causing any problems in the growth rate of plants. Nutrients in the growing solution are distributed differently in water compared to soil.

In fact, commercial hydroponics tends to consume only 1/5 of the land needed for the same amount of plants being cultivated on farmland.

2. Reduced Fossil Fuel Use with Flexible Growing Locations :

With hydroponics, commercial growers have the option of establishing their greenhouses in locations that are closer to their intended markets or distribution channels. Essentially, this cuts down the need for expensive transportation. This improves the profit of the hydroponics grower and also cuts down the pollution associated with the transportation of goods. And while this may not sound like a big deal, the combined effect of many hydroponics growers over time does create a difference.

The flexibility of hydroponics helps the environment by reducing emissions and the overall carbon footprint associated with food production and market transport.

3. Less Water is Used:

Crops need to be constantly hydrated, and they will easily dry out if there isn’t sufficient watering. The problem with soil is that once the water is applied, it is absorbed by the soil and the excess sinks past the roots. Therefore, we have to continually replenish the supply of water to keep plants hydrated and alive. With hydroponics, there is a central nutrient reservoir or water bank, and the water in this water bank is either circulated or fed directly to the plants. There is some loss to evaporation, which is normal, but much of the water is retained for days and weeks. Thus, water is recycled.

4. Potentially Better for The Earth and Soil :

Soil erosion is just one of the problems associated with conventional farming. In some countries, old forests, which are part of the Earth’s “lungs”, are cut down and cleared to turn large areas into agricultural land. The amount of farmable land available for commercial agriculture has been steadily shrinking since the Second World War. Land has become so scarce that we are touching nature’s reserves of forests, just so we can continue growing food for the Earth’s ever increasing human population. Obviously, you still need a little land when you engage in hydroponics. But, unlike conventional farming, the amount of land needed for hydroponics is drastically reduced. So much more can be done with hydroponics in a small space, compared to traditional farming. And, because it doesn’t require soil, hydroponic growing practices can be used in areas with harsh climate and little fertile soil, like in the desert. Additionally, there are different kinds of hydroponics setups. You can combine aquaculture and hydroponics and come up with an aquaponics system that produces not just flowers, fruits, and vegetables, but also fish.

5. Reduced Use of Pesticides and Herbicides

In conventional farming, two kinds of sprays are applied: herbicides and pesticides. Herbicides are sprayed to kill off unwanted plants (these are generally called weeds), while pesticides are applied to control pest insects. The spraying of pesticides and herbicides do great damage to the environment. Pesticides can be swept away by rain, contaminating nearby ecosystems. And, pesticides not only poison the target insect, but can potentially poison other animals as well. However, with hydroponics, there is no weed growth because it takes time for plants to grow into hydroponic grow medium. Hydroponic grow medium does not function exactly like soil. There is no need to use herbicides. There also isn’t much need for pesticide use. For the most part, crops will be protected by closed greenhouses. These greenhouses can be fortified to resist insect pest attacks. Less use of herbicides and pesticides is good for the environment as this will mean drastically reduced runoff, which affects both soil and bodies of water.

One of the looming problems that have affected us all is pesticide runoff. Pesticide molecules have been found in the meat of fish and other animals, which mean the runoff from agricultural lands have become so widespread that it has invaded animals and territories that it shouldn’t be present in.

## 8.4 Social and Health impact

The social and health impacts are given below:

1. Plants improve our **[air quality](http://www.powerhousegrowers.com/?s=air+quality" \t "_blank" \o "Air Quality | PowerHouse Growers)** by filtering toxins that get trapped indoors.

2. The improved air quality lowers risk of respiratory disorders, as well as chronic headaches and eye irritation.

3. Plants teach us how to be attentive and responsive to their needs. This improves our levels of empathy and compassion.

4. **[Research in environmental psychology](https://www.sciencedirect.com/science/article/pii/S0272494410001027" \t "_blank)** has shown that a connection with plants improves human mental wellbeing as well as productivity levels.

5. An indoor garden can provide you with **[fresh kitchen ingredients](https://powerhousehydroponics.com/grow-your-own-microgreens-for-better-health/" \t "_blank" \o "Grow Your Own Microgreens for Better Health)**.

6. Growing your own food reduces our risk of chemical ingestion which you find in foods that have been transported to your local grocery store.

7. Gardening is an excellent **[conversation starting point](https://powerhousehydroponics.com/green-towers-team-creates-solutions-for-urban-food-production/" \t "_blank" \o "GreenTowers Team Creates Solutions for Urban Food Production)** in gaining more attention about the need for improving our environmental impact.

8. In an indoor environment, gardening is a lot less maintenance and plants are relatively safe from pests and disease.

9. There are countless **[automated indoor gardening systems](https://powerhousehydroponics.com/s_automated/" \t "_blank" \o "Automated Systems | PowerHouse Hydroponics)** for you to choose from which are cost effective and low-maintenance

# Chapter 9

# *Discussion and Budget*

## 9.1 Discussion

In conclusion, we can state that we have developed a system that can minimize the risk of the system and also be affordable for the masses, so that the future generation is inspired to take new initiative to enter the agricultural sector. The novelty of this project is that it is flexible so any architecture of hydroponic system is possible to automate and maintain using the system we designed. We hope to further improve this project before deployment.

## 9.2 Budget

The budget for the system we developed is given below. This estimation does not include the cost for the architecture of the agriculture part of the project as it is per the design of the client.

|  |  |  |
| --- | --- | --- |
| **Item** | **Price(BDT)** | **Total Cost (BDT)** |
| NodeMCU x 4 | 445 | 1780 |
| Arduino UNO R3 x 2 | 450 | 900 |
| pH sensor x 1 | 2680 | 2680 |
| DHT11 x1 | 118 | 118 |
| Sonar Sensor x4 | 90 | 360 |
| Relay x5 | 80 | 400 |
| Micro water pump x 3 | 190 | 480 |
| Wires x 6 (20pieces each) | 20 | 120 |
| DC Fan x1 | 47 | 47 |
| TOTAL |  | **6885** |

Table 9.1: BUDGET

# Chapter 10

***Result and Conclusion***

## 10.1 Result

The system is able to send data to the thingSpeak in real time as shown in Fig 5.18. The app ‘MINT’ is also able receive and display the data in real time which is shown in Fig 4.4. The problem of a sensor dying is solved by using null as a value. Thus, we have handles some constrains efficiently.



Fig 10.1: Completed Project

The finished project is given above. We developed a casing for the display for a finishing look.

## 10.2 Conclusion

In this project we have implemented an automated system which works based on IoT and cloud computing and completely maintains a hydroponic system. It minimizes the risk of venturing into hydroponics and also makes it affordable. Seven sensor data is sent every fifteen seconds to thingspeak server, which also provides a visual representation. The system is low maintenance and gives ample opportunity to a lot of different ventures. It consumes less power.

We want to make the system more compact and easy to install. We want to get more durable, industrial level sensor to use and develop this project. We are also thinking about involving image processing techniques to detect disease real time and also collect data on the plants growth.

***Bibliography***

# Bibliography

[1] 101, C. (2018, April 19). Retrieved September 9, 2019, from COMPONENT 101: https://components101.com/sensors/dht22-pinout-specs-datasheet

*[2] SONAR SENSOR*. (2017, September 18). Retrieved September 9, 20119, from COMPONENT 101 : https://components101.com/ultrasonic-sensor-working-pinout-datasheet

*[4] Bangladesh - Agriculture*. (n.d.). Retrieved from Nations Encyclopedia: https://www.nationsencyclopedia.com/economies/Asia-and-the-Pacific/Bangladesh-AGRICULTURE.html

[5] DFRobot. (n.d.). *Gravity: Analog pH Sensor/Meter Kit V2 – DFRobot*. Retrieved September 9, 2019, from DFRobot: https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/SEN0161-V2\_Web.pdf

[6] Domingues, D. S., Takahashi, H. W., Camara, C. A., & Nixdor, S. L. (2012, February 11). Automated system developed to control pH and concentration of nutrient solution evaluated in hydroponic lettuce production. *Computers and Electronics in Agriculture, 84*, 53-61.

[7] Eridani, D., Wardhani, O., & Widianto, E. D. (2017). Designing and Implementing the Arduino-based Nutrition Feeding Automation System of a Prototype Scaled Nutrient Film Technique (NFT) Hydroponics using Total Dissolved Solids (TDS) Sensor. *4th Int. Conf. on Information Tech., Computer, and Electrical Engineering (ICITACEE)* (pp. 170-175). Semarang, Indonesia: IEEE.

[8] Gebremedhin, W., Deasi, B., & Mayekar, A. (2015, April-June). Nutritional Evaluation of Hydroponically Grown Barley Fodder. *Journal of Agricultural Engineering and Food Technology, 2*(2), 86-89.

*[9] Grow Light*. (n.d.). Retrieved from Gardeners: https://www.gardeners.com/how-to/gardening-under-lights/5080.html

*[10] How Relays Work* . (n.d.). Retrieved September 2019, from Galco: http://www.galco.com/comp/prod/relay.htm.

[11]Jagtap, D. S., Chaudhari, J. M., Babar, S. S., & Dhumal, A. P. (2018, June). Automated Hydroponics System. *International Research Journal of Engineering and Technology, 05*(06), 706-709.

[12] M., R., L, S. H., Roopa, R, T., & Prasad, A. S. (2018). Automation of Hydroponics System using Android Application and Ubidots Platform. *NCESC.* *06.* International Journal of Engineering Research & Technology (IJERT).

[13] Palandea, V., Zaheera, A., & Georgea, K. (2017). Fully Automated Hydroponic System for Indoor Plant Growth. *International Conference on Identification, Information and Knowledge in the Internet* (pp. 482-486). Elsevier.

[14] Pandey, R., Jain, V., & Singh, K. P. (2009). Hydroponics Agriculture: Its Status,Scope and Limitations.

*[15] Pros and Cons of hydroponics*. (n.d.). Retrieved September 2019, from Ecoponicss: http://easyponic.com/pros-and-cons-hydroponics/

[16] RoBoDoc. (n.d.). *Micro Water Pump (6v)*. Retrieved from RoBoDoc: https://www.robodocbd.com/product/detail/6V-Micro-Water-Pump-Motor

[17] RoBoDoc. (n.d.). *Servo Motor (180 Degree)*. Retrieved September 2019, from RoBoDoc: https://www.robodocbd.com/Product/detail/Servo-Motor-MG996R--180-Degree-

[18] Saha, S. K. (2010). Soilless Cultivation for Landless People: An Alternative Livelihood Practice through Indigenous Hydroponic Agriculture in Flood-prone Bangladesh. *Ritsumeikan Journal of Asia Pacific Studies*, 139-152.

[19] Sánchez, H. (2019, May). *Cultivating Ocimum basilicum in a hydroponics deep-water culture (DWC) system using dried vermin compost powder as the nutrient source.* Research Gate.

[20]Sihombing, P., Karina, N. A., Tarigan, J. T., & Syarif, M. I. (2018). android, Automated hydroponics nutrition plants systems using arduino uno microcontroller based on. *International Conference on Computing and Applied Informatics.* *978.* IOP Publishing.

[21] Soares, H. R., Júnior, J. A., Silva, Ê. F., MonteiroRolim, M., & Silva, G. F. (2018, June 12). Water and physiological relationships of lettuce cultivated in hydroponics with brackish waters. *Revista Ciência Agronômica,, 50*(2), 216-222.

[22] Teleimperium. (2015, May 5). *A brief Theory on Hydroponics*. Retrieved September 2019, from HACKADAY.IO: https://hackaday.io/project/5403-teleimperium/log/17433-a-brief-theory-on-hydroponics

[23] Tembe, S., Khan, S., & Acharekar, R. (2018, February). IoT based Automated Hydroponics System. *International Journal of Scientific & Engineering Research*, 67-71.

[24] Tocquin, P., Corbesier, L., Havelange, A., Pieltain, A., Kurtem, E., Bernier, G., et al. (2003, January 30). ] A novel high efficiency, low maintenance, hydroponic system for synchronous growth and flowering of Arabidopsis thaliana. *BMC Plant Biology*.

[25] Umamaheswari, D. S., Preethi, A., Pravin, E., & Dhanusha, R. (2016). Integrating Scheduled Hydroponic System. *IEEE International Conference on Advances in Computer Applications (ICACA)* (pp. 333-337). IEEE.

[26] Valdivia, C. G. (2019, June). Turbulent Kinetic Energy Distribution of Nutrient Solution Flow in NFT Hydroponic Systems Using Computational Fluid Dynamics.

***APPENDIX***

## APPENDIX A

**Main Solution Level:**

**#**include <ESP8266WiFi.h>

#include <WiFiClient.h>

#include <ThingSpeak.h> // Include library file for MQTT

#include "Ultrasonic.h" // Include library file for ultrasonic (HC-SR04)

const char\* ssid = "MINT";

const char\* password = "MINTCSE499";

WiFiClient client;

unsigned long myChannelNumber = 833658;

const char \* myWriteAPIKey = "CXL2LZOUPNCCFRRC";

#define pump D2

int minVal= 20;

Ultrasonic ultrasonic(D5,D6); // Assign Trig PIN 14(D5),Assign Echo PIN 12(D6)

int GIu\_Ultrasonic\_Dist\_CM;

//=====================Basic Setup ============================

void setup(){

Serial.begin(9600); // Setup Debug uart port if you want ?

delay(10);

pinMode(pump,OUTPUT);

// Connect to WiFi network

Serial.println();

Serial.println();

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, password);

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

{

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected");

// Print the IP address

Serial.println(WiFi.localIP());

ThingSpeak.begin(client);

}

void loop()

{

GIu\_Ultrasonic\_Dist\_CM=ultrasonic.Ranging(CM);

Serial.print(GIu\_Ultrasonic\_Dist\_CM);

Serial.println(" cm" );

if(GIu\_Ultrasonic\_Dist\_CM <= 20)

{

pinMode(pump,HIGH);

Serial.println("Motor is off");

delay(5000);

}

else

{

pinMode(pump,LOW);

Serial.println("Motor is on");

}

delay(5000);

ThingSpeak.writeField(myChannelNumber, 3, GIu\_Ultrasonic\_Dist\_CM, myWriteAPIKey);

delay(100);

}

//==================================================================

## APPENDIX B

**Reservoir Solution Level (nutrient solution, water and pH):**

#include <ESP8266WiFi.h>

#include <WiFiClient.h>

#include <ThingSpeak.h>

//ThingSpeak SetUp//.................................................................//

const char\* ssid = "MINT";

const char\* password = "MINTCSE499";

WiFiClient client;

unsigned long myChannelNumber = 833658;

const char \* myWriteAPIKey = "CXL2LZOUPNCCFRRC";

//...............................................................................................//

#define trigPinpH D0

#define echoPinpH D1

#define trigPinSol D2

#define echoPinSol D3

#define trigPinWat D4

#define echoPinWat D5

// defines variables

long duration;

int distancepH;

int distanceSol;

int distanceWat;

void setup() {

pinMode(trigPinpH, OUTPUT);

pinMode(echoPinpH, INPUT);

pinMode(trigPinSol, OUTPUT);

pinMode(echoPinSol, INPUT);

pinMode(trigPinWat, OUTPUT);

pinMode(echoPinWat, INPUT);

Serial.begin(9600);

// Connect to WiFi network

Serial.println();

Serial.println();

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, password);

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

{

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected");

// Print the IP address

Serial.println(WiFi.localIP());

ThingSpeak.begin(client);

}

void loop() {

// \*\*\*\* 1st sensor \*\*\*\*

digitalWrite(trigPinpH, LOW);

delayMicroseconds(2);

digitalWrite(trigPinpH, HIGH);

delayMicroseconds(10);

digitalWrite(trigPinpH, LOW);

duration = pulseIn(echoPinpH, HIGH);

distancepH= duration\*0.034/2;

Serial.print("Distance 1 : ");

Serial.println(distancepH);

ThingSpeak.writeField(myChannelNumber, 2, distanceWat, myWriteAPIKey);

delay(200);

// \*\*\*\* 2nd sensor \*\*\*\*

digitalWrite(trigPinSol, LOW);

delayMicroseconds(2);

digitalWrite(trigPinSol, HIGH);

delayMicroseconds(10);

digitalWrite(trigPinSol, LOW);

duration = pulseIn(echoPinSol, HIGH);

distanceSol= duration\*0.034/2;

Serial.print("Distance 2 : ");

Serial.println(distanceSol);

ThingSpeak.writeField(myChannelNumber, 1, distanceSol, myWriteAPIKey);

delay(200);

// \*\*\*\* 3nd sensor \*\*\*\*

digitalWrite(trigPinWat, LOW);

delayMicroseconds(2);

digitalWrite(trigPinWat, HIGH);

delayMicroseconds(10);

digitalWrite(trigPinWat, LOW);

duration = pulseIn(echoPinWat, HIGH);

distanceWat= duration\*0.034/2;

Serial.print("Distance 3 : ");

Serial.println(distanceWat);

ThingSpeak.writeField(myChannelNumber, 6, distanceWat, myWriteAPIKey);

delay(200);

}

## APPENDIX C

**Temperature and Humidity:**

#include <DHT.h>

#include <ESP8266WiFi.h>

#include <WiFiClient.h>

#include <ThingSpeak.h>

#define DHTPIN D1

#define fan D2

int maxhum=88;

int maxTemp=28;

#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

const char\* ssid = "MINT";

const char\* password = "MINTCSE499";

WiFiClient client;

unsigned long myChannelNumber = 833658;

const char \* myWriteAPIKey = "CXL2LZOUPNCCFRRC";

uint8\_t temperature, humidity;

void setup()

{

Serial.begin(9600);

pinMode(fan,OUTPUT);

dht.begin();

delay(10);

// Connect to WiFi network

Serial.println();

Serial.println();

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, password);

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

{

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected");

// Print the IP address

Serial.println(WiFi.localIP());

ThingSpeak.begin(client);

}

void loop()

{

temperature = dht.readTemperature();

humidity = dht.readHumidity();

Serial.print("Temperature Value is :");

Serial.print(temperature);

Serial.println("C");

Serial.print("Humidity Value is :");

Serial.print(humidity);

Serial.println("%");

ThingSpeak.writeField(myChannelNumber, 4, temperature, myWriteAPIKey);

delay(10);

ThingSpeak.writeField(myChannelNumber, 5, humidity, myWriteAPIKey);

if(isnan(humidity)||isnan(temperature))

{

Serial.println("Failed to read from DHT11 sensor");

return;

}

if(humidity>maxhum||temperature>maxTemp)

{

pinMode(fan, LOW);

delay(1000);

Serial.println("ON");

}

else

{

pinMode(fan, HIGH);

delay(1000);

}

delay(5000); // ThingSpeak will only accept updates every 15 seconds.

}

## APPENDIX D

**pH level:**

#include<ESP8266WiFi.h>

#include<WiFiClient.h>

#include<ThingSpeak.h>

#define SensorPin 0 // the pH meter Analog output is connected with the Arduino’s Analog

#define pump D2

// Thingspeak Connection

const char\* ssid = "MINT";

const char\* password = "MINTCSE499";

WiFiClient client;

unsigned long myChannelNumber = 833658;

const char \* myWriteAPIKey = "CXL2LZOUPNCCFRRC";

void setup()

{

Serial.begin(9600);

Serial.println("Ready"); //Test the serial monitor

pinMode(pump,OUTPUT);

delay(10);

Serial.println();

Serial.println();

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, password);

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

{

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected");

// Print the IP address

Serial.println(WiFi.localIP());

ThingSpeak.begin(client);

}

void loop()

{

//for(int i=0; i < 28; i++){

//delay(900000);

pinMode(pump,HIGH);

float s = analogRead(SensorPin)/100.0;

Serial.println(s);

ThingSpeak.writeField(myChannelNumber, 7, s , myWriteAPIKey);

delay(5000);

if(s >= 6.5){

Serial.println("Motor on");

pinMode(pump,LOW);

delay(3000);

}

//}

//delay(75600000);

}

## APPENDIX E

**Light and servomotor:**

#include <Servo.h>

#define RELAY1 7

#define RELAY2 4

Servo myservo;

int pos = 0;

void setup() {

Serial.begin(9600);

pinMode(RELAY1, OUTPUT);

pinMode(RELAY2, OUTPUT);

myservo.attach(2);

}

void loop() {

digitalWrite(RELAY1,0);

digitalWrite(RELAY2,0);// Turns ON Relays 1

Serial.println("Light ON");

for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees// in steps of 1 degree

myservo.write(pos); // tell servo to go to position in variable 'pos'

delay(10); // waits 15ms for the servo to reach the position

}

delay(10000);

//delay(32400000);

for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees

myservo.write(pos); // tell servo to go to position in variable 'pos'

delay(10); // waits 15ms for the servo to reach the position

}

digitalWrite(RELAY1,1); // Turns Relay Off

digitalWrite(RELAY2,1);

Serial.println("Light OFF");

delay(10000);

//delay(54000000);

}

## APPENDIX F

**Application:**

**Splash.XML**

<?xml version="1.0" encoding="utf-8"?>

<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="#031B17"

tools:context=".Splash">

<ImageView

android:id="@+id/imageView"

android:layout\_width="353dp"

android:layout\_height="254dp"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

android:layout\_marginEnd="8dp"

android:layout\_marginBottom="360dp"

android:contentDescription="@string/app\_name"

android:visibility="visible"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.34"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="1.0"

app:srcCompat="@mipmap/logo"

tools:visibility="visible" />

<TextView

android:layout\_width="165dp"

android:layout\_height="114dp"

android:layout\_marginStart="0sp"

android:paddingTop="0dp"

android:gravity="center\_horizontal"

android:text="@string/app\_name"

android:textColor="@color/Text"

android:textSize="36sp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.502"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/imageView"

app:layout\_constraintVertical\_bias="0.0" />

</android.support.constraint.ConstraintLayout>

**Splash.java**

package com.example.myapplication1;

import android.content.Intent;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

public class Splash extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

try

{

this.getSupportActionBar().hide();

}

catch (NullPointerException e){}

setContentView(R.layout.activity\_splash);

Thread myThread = new Thread(){

@Override

public void run() {

try {

sleep(3000);

Intent intent = new Intent(getApplicationContext(),MainActivity.class);

startActivity(intent);

finish();

} catch (InterruptedException e) {

e.printStackTrace();

}

}

};

myThread.start();

}

}

**MainAvtivity.XML**

<?xml version="1.0" encoding="utf-8"?>

<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/colorPrimaryDark"

android:padding="10px">

<ScrollView

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<TableLayout

android:id="@+id/tableLayout"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginStart="0dp"

android:layout\_marginTop="0dp"

android:layout\_marginEnd="0dp"

android:layout\_marginBottom="0dp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent">

<TableLayout android:layout\_margin="5dp">

<TableRow>

<Button

android:id="@+id/ViewData"

android:layout\_width="0px"

android:layout\_height="match\_parent"

android:layout\_weight="4"

android:layout\_column="1"

android:layout\_marginBottom="5dp"

android:layout\_marginRight="10px"

android:paddingLeft="40px"

android:paddingRight="40px"

android:background="@color/button"

android:text="@string/DataView"

android:textColor="@color/colorPrimary" />

<Button

android:id="@+id/info"

android:layout\_width="0px"

android:layout\_column="2"

android:layout\_marginLeft="10px"

android:layout\_marginBottom="5dp"

android:layout\_weight="4"

android:background="@color/button"

android:paddingLeft="40px"

android:paddingRight="40px"

android:text="@string/ViewInfo"

android:textColor="@color/colorPrimary" />

</TableRow>

</TableLayout>

<TableLayout

android:id="@+id/solutionLevel"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="5dp"

android:background="@color/colorPrimary"

android:baselineAligned="false">

<TableRow

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<TextView

android:id="@+id/S.L.Name"

android:layout\_width="0dp"

android:layout\_weight="2"

android:layout\_column="1"

android:paddingLeft="15dp"

android:paddingTop="8dp"

android:paddingBottom="8dp"

android:text="@string/SolutionL"

android:textColor="@color/background"

android:textSize="18sp" />

<TextView

android:layout\_width="0px"

android:layout\_weight="1"

android:layout\_column="2"

android:padding="8dp"

android:text=""

android:textColor="@color/background"

android:textSize="14sp" />

<TextView

android:id="@+id/S.L.Time"

android:layout\_width="135dp"

android:layout\_column="3"

android:padding="8dp"

android:textColor="@color/background"

android:textSize="18sp" />

</TableRow>

<TableRow>

<TextView

android:id="@+id/S.L.Data"

android:layout\_weight="2"

android:layout\_column="1"

android:paddingLeft="18dp"

android:paddingTop="5dp"

android:paddingBottom="8dp" android:textColor="@color/background"

android:textSize="18sp" />

<ImageView

android:id="@+id/color1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_column="3"

android:padding="5dp"

android:src="@drawable/circle" />

</TableRow>

</TableLayout>

<TableLayout

android:id="@+id/pHlevel"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="5dp"

android:background="@color/colorPrimary">

<TableRow

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<TextView

android:id="@+id/PH\_Name"

android:layout\_width="0dp"

android:layout\_weight="2"

android:layout\_column="1"

android:paddingLeft="15dp"

android:paddingTop="8dp"

android:paddingBottom="8dp" android:text="@string/PHlevel"

android:textColor="@color/background"

android:textSize="18sp" />

<TextView

android:layout\_width="0px"

android:layout\_weight="1"

android:layout\_column="2"

android:padding="8dp"

android:text=""

android:textColor="@color/background"

android:textSize="14sp" />

<TextView

android:id="@+id/PH\_Time"

android:layout\_width="135dp"

android:layout\_column="3"

android:padding="8dp"

android:textColor="@color/background"

android:textSize="18sp" />

</TableRow>

<TableRow>

<TextView

android:id="@+id/PH\_Data"

android:layout\_weight="2"

android:layout\_column="1"

android:paddingLeft="15dp"

android:paddingTop="8dp"

android:paddingBottom="8dp" android:textColor="@color/background"

android:textSize="18sp" />

<ImageView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_column="3"

android:padding="5dp"

android:src="@drawable/circle" />

</TableRow>

</TableLayout>

<TableLayout

android:id="@+id/mainSolution"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="5dp"

android:background="@color/colorPrimary">

<TableRow

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<TextView

android:id="@+id/mainSolutionName"

android:layout\_width="0dp"

android:layout\_weight="2"

android:layout\_column="1"

android:paddingLeft="15dp"

android:paddingTop="8dp"

android:paddingBottom="8dp" android:text="@string/MainSolution"

android:textColor="@color/background"

android:textSize="18sp" />

<TextView

android:layout\_width="0px"

android:layout\_weight="1"

android:layout\_column="2"

android:padding="8dp"

android:text=""

android:textColor="@color/background"

android:textSize="14sp" />

<TextView

android:id="@+id/mainSolutionTime"

android:layout\_width="135dp"

android:layout\_column="3"

android:padding="8dp"

android:textColor="@color/background"

android:textSize="18sp" />

</TableRow>

<TableRow>

<TextView

android:id="@+id/mainSolutionData"

android:layout\_weight="2"

android:layout\_column="1"

android:paddingLeft="15dp"

android:paddingTop="8dp"

android:paddingBottom="8dp" android:textColor="@color/background"

android:textSize="18sp" />

<ImageView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_column="3"

android:padding="5dp"

android:src="@drawable/circle" />

</TableRow>

</TableLayout>

<TableLayout

android:id="@+id/temperature"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="5dp"

android:background="@color/colorPrimary">

<TableRow

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<TextView

android:id="@+id/temperatureName"

android:layout\_width="0dp"

android:layout\_weight="2"

android:layout\_column="1"

android:paddingLeft="15dp"

android:paddingTop="8dp"

android:paddingBottom="8dp" android:text="@string/Temp"

android:textColor="@color/background"

android:textSize="18sp" />

<TextView

android:layout\_width="0px"

android:layout\_weight="1"

android:layout\_column="2"

android:padding="8dp"

android:text=""

android:textColor="@color/background"

android:textSize="14sp" />

<TextView

android:id="@+id/temperatureTime"

android:layout\_width="135dp"

android:layout\_column="3"

android:padding="8dp"

android:textColor="@color/background"

android:textSize="18sp" />

</TableRow>

<TableRow>

<TextView

android:id="@+id/temperatureData"

android:layout\_weight="2"

android:layout\_column="1"

android:paddingLeft="20dp"

android:paddingTop="5dp"

android:paddingBottom="8dp" android:textColor="@color/background"

android:textSize="18sp" />

</TableRow>

</TableLayout>

<TableLayout

android:id="@+id/PH"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="5dp"

android:background="@color/colorPrimary">

<TableRow

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<TextView

android:id="@+id/PHName"

android:layout\_width="0dp"

android:layout\_weight="2"

android:layout\_column="1"

android:paddingLeft="15dp"

android:paddingTop="8dp"

android:paddingBottom="8dp" android:text="@string/PH"

android:textColor="@color/background"

android:textSize="18sp" />

<TextView

android:layout\_width="0px"

android:layout\_weight="1"

android:layout\_column="2"

android:padding="8dp"

android:text=""

android:textColor="@color/background"

android:textSize="14sp" />

<TextView

android:id="@+id/PHTime"

android:layout\_width="135dp"

android:layout\_column="3"

android:padding="8dp"

android:textColor="@color/background"

android:textSize="18sp" />

</TableRow>

<TableRow>

<TextView

android:id="@+id/PHData"

android:layout\_weight="2"

android:layout\_column="1"

android:paddingLeft="20dp"

android:paddingTop="5dp"

android:paddingBottom="8dp" android:textColor="@color/background"

android:textSize="18sp" />

</TableRow>

</TableLayout>

<TableLayout

android:id="@+id/Humidity"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="5dp"

android:background="@color/colorPrimary">

<TableRow

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<TextView

android:id="@+id/HumidityName"

android:layout\_width="0dp"

android:layout\_weight="2"

android:layout\_column="1"

android:paddingLeft="15dp"

android:paddingTop="8dp"

android:paddingBottom="8dp" android:text="@string/Humi"

android:textColor="@color/background"

android:textSize="18sp" />

<TextView

android:layout\_width="0px"

android:layout\_weight="1"

android:layout\_column="2"

android:padding="8dp"

android:text=""

android:textColor="@color/background"

android:textSize="14sp" />

<TextView

android:id="@+id/HumidityTime"

android:layout\_width="135dp"

android:layout\_column="3"

android:padding="8dp"

android:textColor="@color/background"

android:textSize="18sp" />

</TableRow>

<TableRow>

<TextView

android:id="@+id/HumidityData"

android:layout\_weight="2"

android:paddingLeft="20dp"

android:paddingTop="5dp"

android:paddingBottom="8dp"

android:layout\_column="1" android:textColor="@color/background"

android:textSize="18sp" />

</TableRow>

</TableLayout>

<TableLayout

android:id="@+id/WaterLT"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="5dp"

android:background="@color/colorPrimary">

<TableRow

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<TextView

android:id="@+id/WaterLTName"

android:layout\_width="0dp"

android:layout\_weight="2"

android:layout\_column="1"

android:paddingLeft="15dp"

android:paddingTop="8dp"

android:paddingBottom="8dp" android:text="@string/WaterT"

android:textColor="@color/background"

android:textSize="18sp" />

<TextView

android:layout\_width="0px"

android:layout\_weight="1"

android:layout\_column="2"

android:padding="8dp"

android:text=""

android:textColor="@color/background"

android:textSize="14sp" />

<TextView

android:id="@+id/WaterLTime"

android:layout\_width="135dp"

android:layout\_column="3"

android:padding="8dp"

android:textColor="@color/background"

android:textSize="18sp" />

</TableRow>

<TableRow>

<TextView

android:id="@+id/WaterLTData"

android:layout\_weight="2"

android:layout\_column="1"

android:paddingLeft="15dp"

android:paddingTop="8dp"

android:paddingBottom="8dp" android:textColor="@color/background"

android:textSize="18sp" />

<ImageView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_column="3"

android:padding="5dp"

android:src="@drawable/circle" />

</TableRow>

</TableLayout>

</TableLayout>

</ScrollView>

</android.support.constraint.ConstraintLayout>

**MainActivity.java**

package com.example.myapplication1;

import android.app.AlertDialog;

import android.content.Intent;

import android.os.Handler;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.ImageView;

import android.widget.TextView;

import com.android.volley.Request;

import com.android.volley.RequestQueue;

import com.android.volley.Response;

import com.android.volley.VolleyError;

import com.android.volley.toolbox.JsonObjectRequest;

import com.android.volley.toolbox.Volley;

import org.json.JSONArray;

import org.json.JSONException;

import org.json.JSONObject;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

public class MainActivity extends AppCompatActivity {

Button click;

Button click1;

public static TextView SolutionData;

public static TextView SolutionLTime;

public static TextView PHLTime;

public static TextView PHLData;

public static TextView TempTime;

public static TextView TempData;

public static TextView MainSolution;

public static TextView MainSoutionTime;

public static TextView PHData;

public static TextView PHTime;

public static TextView WLD;

public static TextView WLT;

public static TextView HumiData;

public static TextView HumiTime;

public static final String time = "com.example.myapplication1.time";

private String field3="";

ImageView circleColor;

TextView solutionL1;

private String solutionL;

private String ShowTime;

private String PH\_L;

private String TemperatumeL;

private String Humidity;

private String PH\_M;

private String WaterTemperature;

private RequestQueue mQueue;

public static int field11;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

click = (Button) findViewById(R.id.ViewData);

click1=findViewById(R.id.info);

circleColor=findViewById(R.id.color1);

SolutionData=findViewById(R.id.S\_L\_Data);

SolutionLTime=findViewById(R.id.S\_L\_Time);

PHLTime=findViewById(R.id.PH\_Time);

PHLData=findViewById((R.id.PH\_Data));

MainSolution=findViewById(R.id.mainSolutionData);

MainSoutionTime=findViewById(R.id.mainSolutionTime);

TempData=findViewById(R.id.temperatureData);

TempTime=findViewById(R.id.temperatureTime);

PHData=findViewById((R.id.PHData));

PHTime=findViewById(R.id.PHTime);

HumiData=findViewById(R.id.HumidityData);

HumiTime=findViewById(R.id.HumidityTime);

WLD=findViewById(R.id.WaterLTData);

WLT=findViewById(R.id.WaterLTime);

mQueue = Volley.newRequestQueue(this);

click1.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent=new Intent(MainActivity.this,Information.class);

startActivity(intent);

}

});

click.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

fetchData process = new fetchData();

process.execute();

HumiData process1 = new HumiData();

process1.execute();

pH process3 = new pH();

process3.execute();

pHLevel process4 = new pHLevel();

process4.execute();

Temp process5 = new Temp();

process5.execute();

WaterLevel process6 = new WaterLevel();

process6.execute();

new Handler().postDelayed(new Runnable() {

@Override

public void run() {

click.performClick();

}

}, 5000000);

}

});

}

public void goToInfo()

{

Intent i=new Intent(MainActivity.this,Information.class);

startActivity(i);

}

}

**FetchData.java**

package com.example.myapplication1;

import android.os.AsyncTask;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.net.HttpURLConnection;

import java.net.MalformedURLException;

import java.net.URL;

public class fetchData extends AsyncTask<Void,Void,Void> {

String data ="";

@Override

protected Void doInBackground(Void... voids) {

try {

URL url = new URL("https://api.thingspeak.com/apps/thinghttp/send\_request?api\_key=W9RYQDQSJ4Z40VOG\n");

HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();

InputStream inputStream = httpURLConnection.getInputStream();

BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(inputStream));

data = bufferedReader.readLine();

} catch (MalformedURLException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

return null;

}

@Override

protected void onPostExecute(Void aVoid) {

super.onPostExecute(aVoid);

if(data.length()<2)

MainActivity.SolutionData.setText(this.data);

}

}

**pHLevel.java**

package com.example.myapplication1;

import android.os.AsyncTask;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.net.HttpURLConnection;

import java.net.MalformedURLException;

import java.net.URL;

public class pHLevel extends AsyncTask<Void,Void,Void> {

String data="";

@Override

protected Void doInBackground(Void... voids) {

try {

URL url = new URL("https://api.thingspeak.com/apps/thinghttp/send\_request?api\_key=N9D6D08J8DUQTS54\n");

HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();

InputStream inputStream = httpURLConnection.getInputStream();

BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(inputStream));

data = bufferedReader.readLine();

} catch (MalformedURLException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

return null;

}

@Override

protected void onPostExecute(Void aVoid) {

super.onPostExecute(aVoid);

if(data.length()<2)

MainActivity.PHLData.setText(this.data);

}

}

**MainSolution.java**

package com.example.myapplication1;

import android.os.AsyncTask;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.net.HttpURLConnection;

import java.net.MalformedURLException;

import java.net.URL;

public class MainSolution extends AsyncTask<Void,Void,Void> {

String data ="";

@Override

protected Void doInBackground(Void... voids) {

try {

URL url = new URL("https://api.thingspeak.com/apps/thinghttp/send\_request?api\_key=ZH5H49X6PL56L2RQ\n");

HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();

InputStream inputStream = httpURLConnection.getInputStream();

BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(inputStream));

data = bufferedReader.readLine();

} catch (MalformedURLException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

return null;

}@Override

protected void onPostExecute(Void aVoid) {

super.onPostExecute(aVoid);

if(data.length()<2)

MainActivity.MainSolution.setText(this.data);

}

}

**Temp.java**

package com.example.myapplication1;

import android.os.AsyncTask;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.net.HttpURLConnection;

import java.net.MalformedURLException;

import java.net.URL;

public class Temp extends AsyncTask<Void,Void,Void> {

String data ="";

@Override

protected Void doInBackground(Void... voids) {

try {

URL url = new URL("https://api.thingspeak.com/apps/thinghttp/send\_request?api\_key=1AOXHS2IED2IWI3I\n");

HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();

InputStream inputStream = httpURLConnection.getInputStream();

BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(inputStream));

data = bufferedReader.readLine();

} catch (MalformedURLException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

return null;

}@Override

protected void onPostExecute(Void aVoid) {

super.onPostExecute(aVoid);

if(data.length()<2)

MainActivity.TempData.setText(this.data);

}

}

**HumiData.java**

package com.example.myapplication1;

import android.os.AsyncTask;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.net.HttpURLConnection;

import java.net.MalformedURLException;

import java.net.URL;

public class HumiData extends AsyncTask<Void,Void,Void> {

String data ="";

@Override

protected Void doInBackground(Void... voids) {

try {

URL url = new URL("https://api.thingspeak.com/apps/thinghttp/send\_request?api\_key=84TTGIFLET8TOJBA\n");

HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();

InputStream inputStream = httpURLConnection.getInputStream();

BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(inputStream));

data = bufferedReader.readLine();

} catch (MalformedURLException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

return null;

}@Override

protected void onPostExecute(Void aVoid) {

super.onPostExecute(aVoid);

if(data.length()<2)

MainActivity.HumiData.setText(this.data);

}

}

**WaterLevel.java**

package com.example.myapplication1;

import android.os.AsyncTask;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.net.HttpURLConnection;

import java.net.MalformedURLException;

import java.net.URL;

public class WaterLevel extends AsyncTask<Void,Void,Void> {

String data ="";

String dataParsed = "";

String singleParsed ="";

@Override

protected Void doInBackground(Void... voids) {

try {

URL url = new URL("https://api.thingspeak.com/apps/thinghttp/send\_request?api\_key=ZH5H49X6PL56L2RQ\n\n");

HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();

InputStream inputStream = httpURLConnection.getInputStream();

BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(inputStream));

data = bufferedReader.readLine();

} catch (MalformedURLException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

return null;

}

@Override

protected void onPostExecute(Void aVoid) {

super.onPostExecute(aVoid);

if(data.length()<2)

MainActivity.WLD.setText(this.data);

}

}

**Information.XML**

<?xml version="1.0" encoding="utf-8"?>

<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/colorPrimaryDark"

tools:context=".information">

<ScrollView

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<TableLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<TableRow

android:layout\_width="match\_parent">

<ImageView

android:layout\_weight="4"

android:layout\_gravity="fill"

android:layout\_height="565sp"

android:layout\_marginTop="0dp"

android:padding="0dp"

android:src="@mipmap/pic1" />

</TableRow>

<TableRow

android:layout\_width="match\_parent">

<ImageView

android:layout\_weight="4"

android:layout\_gravity="center"

android:layout\_height="565sp"

android:layout\_marginTop="0dp"

android:padding="0dp"

android:src="@mipmap/pic2" />

</TableRow>

<TableRow

android:layout\_width="match\_parent">

<ImageView

android:layout\_height="280dp"

android:layout\_gravity="center"

android:layout\_marginTop="0dp"

android:layout\_weight="4"

android:padding="0dp"

android:src="@mipmap/pic3" />

</TableRow>

<TableRow

android:layout\_width="match\_parent">

<ImageView

android:layout\_height="450dp"

android:layout\_marginTop="0dp"

android:layout\_weight="4"

android:padding="0dp"

android:src="@mipmap/pic4" />

</TableRow>

<TableRow

android:layout\_width="match\_parent">

<ImageView

android:layout\_height="565sp"

android:layout\_marginTop="0dp"

android:layout\_weight="4"

android:padding="0dp"

android:src="@mipmap/pic5" />

</TableRow>

<TableRow

android:layout\_width="match\_parent">

<ImageView

android:layout\_height="250dp"

android:layout\_marginTop="0dp"

android:layout\_weight="4"

android:padding="0dp"

android:src="@mipmap/pic6" />

</TableRow>

<TableRow

android:layout\_width="match\_parent">

<ImageView

android:layout\_height="370dp"

android:layout\_marginTop="0dp"

android:layout\_weight="4"

android:padding="0dp"

android:src="@mipmap/pic7" />

</TableRow>

</TableLayout>

</ScrollView>

</android.support.constraint.ConstraintLayout>

**Information.java**

package com.example.myapplication1;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

public class Information extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_information);

}

}