UNIVERSITI PUTRA MALAYSIA

SMART ENVIRONMENTAL MONITORING FOR HYDROPONIC SYSTEM

NURSAKINAH BINTI ABD MALIK

SMART ENVIRONMENTAL MONITORING FOR HYDROPONIC SYSTEM

NURSAKINAH BINTI ABD MALIK

183160

BACHELOR OF ENGINEERING

(AGRICULTURAL AND BIOSYSTEMS)

FACULTY OF ENGINEERING

UNIVERSITI PUTRA MALAYSIA

SERDANG SELANGOR

2018/2019

APPROVAL SHEET

This project report here to entitle "SMART ENVIRONMENTAL MONITORING FOR HYDROPONIC SYSTEM" was prepared and submitted by NURSAKINAH BINTI ABD MALIK in partial fulfillment of the requirement for the degree of Bachelor of Engineering (Agricultural and Biosystems) is hereby accepted.

Approved by:

…………………………… Date:…………………………

(Dr.Diyana Jamaludin)

Project Supervisor

Approved by:

…………………………… Date:…………………………

(Prof. Madya Dr. Samsuzana bt. Abd Aziz)

Project Examiner

Approved by:

…………………………… Date:…………………………

(Dr. Hazreen Haizi bt. Harith)

Project Examiner

ACKNOWLEDGEMENT

In the name of Allah, The Most Gracious, The Most Merciful.

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals. I would like to extend my sincere thanks to all of them.

First and foremost, my deep appreciation and gratitude goes to my supervisor, Dr.Diyana binti Jamaludin, who expertly guided me through her advice, guidance and encouragement without which this project would not have been possible. Her unwavering support kept me constantly focused with my work and engaged on what I had done. I would like to thank Mr Ariffin and Mr Amesh their help and guidance in completing my project.

My sincere appreciation and thanks also towards the Master student, Mr Syazwan Hassan for his willingness in numerous ways to help and guide me throughout the completion of the project. Thanks to the lab instructor, Mr Sabri who helped me directly or indirectly in completing my experiment. Special thanks to my bestfriend, Mr Adam Danial Lim who always give me advice on my writing.

I would like express my gratitude towards my family for their kind co-operation and encouragement which help me to able complete this project even at worst time.

My thanks and appreciations also go to my friends and my colleagues in developing the project and people who have willingly helped me out with their abilities.

TABLE OF CONTENTS

ABSTRACT

This project was conducted to developed a smart environmental monitoring for hydroponic system using wireless data computing, observe and analyze smart system as comparison to the timer system in terms of effectiveness and costing. Two systems which are timer and smart systems that consist of microcontroller, actuators and sensors were connected to a NodeMCU to enable sending data to cloud. The control parameters in this hydroponic system are light intensity, temperature, humidity and water level. The control mechanism for smart system is automated self-control based on real-time condition and optimum set point, while timer system response based on the timing setup. The smart system has more investment than timer but smart system is more effective in control the environmental condition according to the optimum set-up and the sensors also able to work as an IoT system.

ABSTRAK

Projek ini dijalankan untuk membangunkan pemantauan alam sekitar pintar untuk sistem hidroponik menggunakan pengkomputeran data tanpa wayar, memerhatikan dan menganalisis sistem pintar sebagai perbandingan dengan sistem pemasa dari segi keberkesanan dan kos. Dua sistem yang sistem pemasa dan pintar yang terdiri daripada mikrokontroler, penggerak dan sensor disambungkan ke NodeMCU untuk membolehkan penghantaran data ke *cloud*. Parameter kawalan dalam sistem hidroponik ini adalah intensiti cahaya, suhu, kelembapan dan paras air. Mekanisme pengawasan untuk sistem pintar adalah kendali kendali otomatis berdasarkan kondisi waktu nyata dan titik set yang optimum, sementara respon sistem pemasa berdasarkan penentuan waktu. Sistem pintar mempunyai lebih banyak pelaburan daripada pemasa tetapi sistem pintar lebih berkesan dalam mengawal keadaan alam sekitar mengikut tetapan yang optimum dan sensor juga dapat berfungsi sebagai sistem IoT.

LIST OF TABLES

LIST OF FIGURES

LIST OF EQUATIONS

CHAPTER 1

INTRODUCTION

1.1 Background of Study

The ancient history of agriculture dates back almost thousands of years. As time goes, its progress has been driven through the implementation of several new systems, methods, techniques, and methods over time. It uses more than one-third of the global workforce. For many nations, agriculture is the cornerstone of an economy and makes an important contribution to underdeveloped countries economic growth. It also steers the process of financial growth in developed countries. (Imran Ali Lakhiar, 2018)

The fast rise in the population, along with the decline in agricultural property, the intensification of global climate change and the exacerbation of water resources, the decline of labour-power and energy crunches pose tremendous difficulties and hurdles to the agricultural sector. The unpredictable climate change which includes the extreme weather is the main hurdle to cope with the need of the produce food constantly. If these things left ignored, food security will be an issue.

More food production is in demand but the other thing to consider is the industry sector because the agriculture is complimentary with industrial sector. As humanity evolves from hunters and gatherers to agrarian societies, various alternatives has been concentrated primarily on enhancing plant output and productivity through either genetic change, cultural or husbandry, management methods, or the development and introduction of plant protection measures.

Accordingly, peoples have begun to explore the opportunities in the last and present centuries by embracing various contemporary methods in agriculture.

Nowadays, more modern practices has been introduced to the farmer such as precision agriculture which opened to other technology for example, cloud data computing, wireless sensor, etc. All of the modern practices aim to make the task at the field has become less time consuming as well as help to increase the productivity of agriculture field. (Verdouw, 2016), (TongKe, 2013).

In this study, smart monitoring system for hydroponic is applied. There are two systems which are timer and smart system. One system is using the timer to control environment and on system will use automated system where it has a microcontroller to receive data from sensor and give signal to the actuators to response towards the current environment.

This smart system helps to minimize human intervention yet the productivity increasing. On the other hand, the system will helps to minimize the sources usually used and wasted due to lacking in monitoring system. To optimize the uses of sources like water, electricity, the system must smartly react to the changes that occurred.

1.2 Problem Statements

Hydroponic farming system can be difficult and tedious same as conventional agriculture practice since users need to manage their crops requirement such as water irrigation, control the temperature and light requirement. One of the efforts taken to face this problem is by implement timer to the system which helps the users to ensure the input required by crops are given continuously. However, this method is less practical because the condition may be affected by the surrounding. So, by implementing an automated system combine with cloud data computing, it can help user to get real-time data and less intervention without worrying the system condition. Furthermore, the system can be integrating with actuators that works based on the real data to provide the optimum condition to the system.

1.3 Objectives of Study

The main objective of this research is to develop a smart environmental monitoring for hydroponic system using wireless data computing. In order to achieve this, the specific objectives need to be evaluated:

- To integrate temperature, relative humidity, light and water level sensors with control board systems
- To compare performance of smart system with timer system
- To determine cost of smart system and timer system

CHAPTER 2

LITERATURE REVIEW

2.1 Monitoring System in Agriculture

Monitoring system is now taking a higher level in line with other technology in the world, monitoring tasks in agriculture is one of the essential parts since plant or living things need more attention. By monitoring, the user can know the output from their input that they have set, therefore this will affect resource being invest in the first place.

Nowadays, the monitoring system has been upgraded using implementation of sensor and wireless network. The system is able to be monitored easily even with tremendous amount of data, the system also can be controlled remotely or automatically from a distance and this technology also known as Internet of Things (IoT).

Internet of Things (IoT) basically is where different devices are connected through internet and enable data transferred to one another. Zedadra et al., (2018) stated that the IoT is a research field where digital and physical entities are interconnected through internet. The quality and the accuracy of IoT data can be ensure by doing validation process of aspects such as communication, computing and software which can be integrate continuously. (Matthews A., 2017), (Braun, Colangelo, & Steckel, 2018). This kind of technology must consists of devices, connectivity and middleware to function as it was defined.

2.1.1 Devices

The 'things' or devices refers to computer-based smart devices that can function to sense and communicate. Smart elements such as any device or system is essential to run the IoT system where these elements will be connect to the internet and allow communication between systems and devices. (Matthew A., 2017). Examples of devices include RFID, sensors, and computer nodes as shown in figure 1.

These devices used must be able to be connecting to the internet such as WiFi to enable data acquisition. However, having a massive data will likely happen when reporting raw data to the cloud where it will be costly as every bit was charged and use more battery power. (Rahman, Ozcelebi, & Lukkien, 2018)

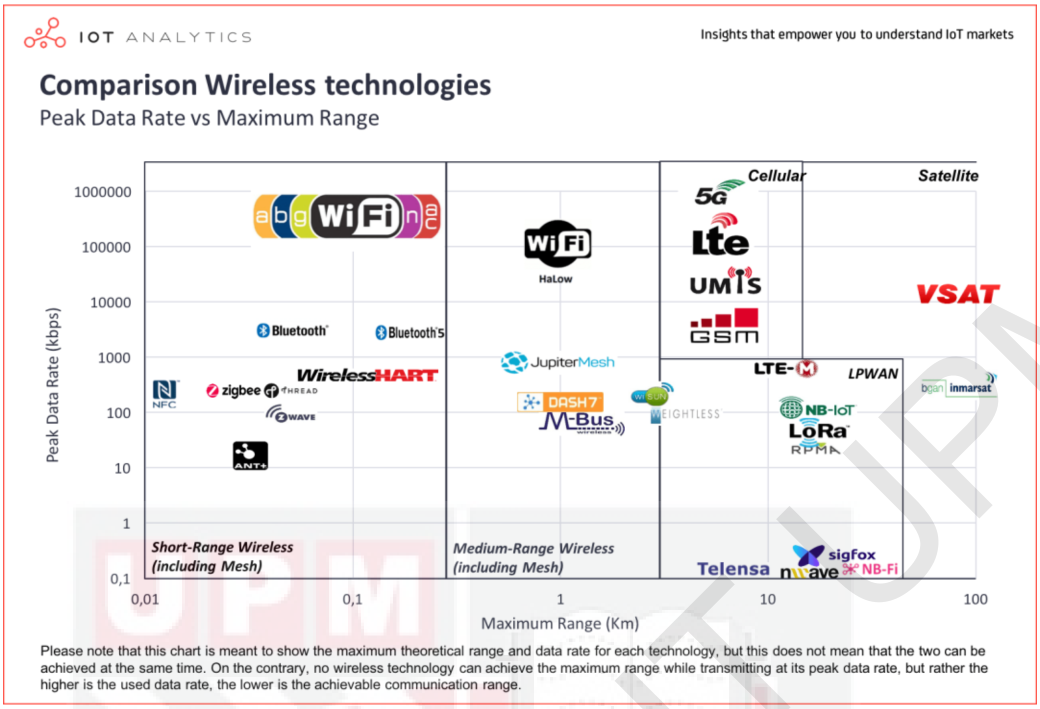
source : https://blogs.synopsys.com/tousbornottousb/2015/06/22/how-will-usb-used-in-iot-part-1/

Figure 1 : Examples of wearable devices and machine-machine applications

2.1.2 Connectivity

One of the crucial elements is connectivity. The system need to be in a network of connectivity then only it will allow the information to be transferred from one side to another side. Connectivity must be available to the internet provider such as telco, 3G or 4G etc. Ozturk, Jaber, & Imran, (2018) stated that the connection used in data networks comes in various communication standards with different strengths and weaknesses. Even though the task is just the same which connecting data to the cloud, but it is quite important to choose which standards is the best.

Figure 2 shows the symbols for the connectivity that will appear when connections are available thus users should know the symbol for connectivity options while in figure 3 shows the differences between those options. The spectrum have two categories which are licensed usually used by Telco operators and class-licensed (or open access) where minimum license administration were required by spectrum authorities. (B. Al Homssi *et al*., 2018).



Source : https://iot-analytics.com/iot-segments/iot-connectivity/

Figure 2 : The symbols of IoT connections

Wireless IoT Connectivity Options

	Current SP Offering				Private Network			Emerging	
Technology	2G	3G	LTE	WiFi	Zigbee	Wireless Hart	802.15.4g	LPWA (LoRa, Ingenu, SigFox, etc.)	NB-IOT EC-GSM
Range	Long	Long	Long	Limited (<200m)	Short	Limited (<250m)	Limited (<1 km)	Long >10 km (rural) >1 km (urban)	Long
Topology	P2P	P2P	P2P	P2P/Mesh	Mesh	Mesh	Mesh	P2P	P2P
Tx Current Consumption (3V)	30mA to 400mA	500 to 1000mA	600 to 1100 mA	19 to 400 mA	34mA	28mA	~ 35mA	<20 mA	
Standby Current Consumption (3V)	0.35 mA	1.2 to 3.5mA	1.5 to 5.5mA	1.1 mA	0.003mA	0.008mA	~.005mA	<0.005mA	
Energy Harvesting	No	No	No	No	Possible	Possible	Possible	Possible	Possible
Operating Life on battery (2000mAh) h=hours; d=days A=active; I=Idle	4-8 h (A) 36 d (I)	2-4 h (A) 20 d (I)	2-3 h (A) 12 d (I)	4-8 h (A) 50 h (I)	60 h (A)	8-10 years	Variable	10-20 years	
Module Cost (est.)	$12	$35-$50	$40-$80	$5-$8	$6-$12	NC	$3	$5	?
Spectrum	Lic.	Lic.	Lic.	Unlic.	Unlic.	Unlic.	Unlic.	Unlic.	Lic.

source : cisco

Figure 3 : The detail comparison of different types of connectivity

2.1.3 Middleware

Middleware refers to the software that functions as the data processor or called as the IoT platform. It joined the applications communicating over interfaces. (Razzaque, Milojevic-Jevric, Palade, & Cla, 2016). Google Cloud, Amazon Web etc. are examples of the IoT platform.

The middleware play crucial role in data networking system for example, data demonstrator, storage of data, so it is important to choose the most suitable platform to perform all the data processing. The criteria as shown in Table 1 can be referred for choosing the most suitable platform.

Table 1 : The explanation of the criteria of suitable platform

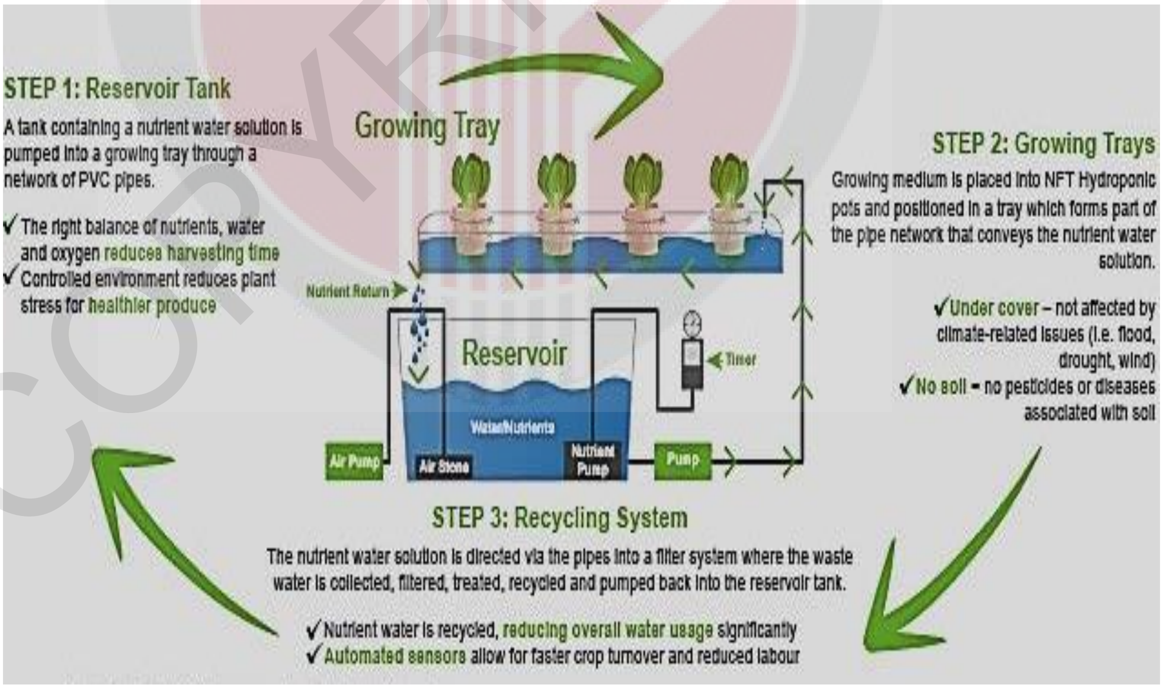
USABILITY	PORTABILITY	SUPPORTABILITY
The usability can be measured by looking at its simplicity of installation, configuration and to learn the task. It is also can be refer to the guidance to the functions and the sample code availability.	Portability is the ability to function with other components such as databases, operating systems, and development frameworks.	This refers to the how active the community of using the platform and the quality and clarity of the documentation.

Middleware acts as a data computing medium where all data recorded will be stored, it is preferable if the middleware can do data analysis, have map for location, and can stored a lot of data.

2.2 Hydroponic System

Hydroponic system is one of the types of urban farming other than greenhouse, vertical farm, rooftop garden, etc. This system is a system where plant grows in nutrient solution instead of soil. The benefits of using hydroponic system are saving space, clean and fresh vegetables, controlled environment and less time of planting yet good quality of crop in certain condition.

In hydroponic system, the crop use less space but the crops can grow very close to each other. The environment can be controlled from contamination that come from the soil and can produce good quality of crop. This system also requires little amount of water since farmer can control the watering and because for all of the benefit stated, this system will give profit towards the farmer. (Boonnam et al., 2017). Figure 4 shows how the hydroponic works.



Source : Front Porch

Figure 4 : The figure shows how hydroponic system works step by step.

2.2.1 Automated Hydroponic System

The fully automated hydroponic system consists of sensors and actuators that will function in pair. The sensor will obtain data that will be an order for the actuators to response or not. This system helps to simplify the monitoring and controlling task for the hydroponic system because less human intervention needed. In Table 2, the comparison between two systems was listed based on crop environment, controlling and monitoring method, and how it will operate. This comparison also shows reasons why automated hydroponic system chosen by many researcher.

Table 2 : Shows the comparison between fully automated with not automated system.

	AUTOMATED HYDROPONIC SYSTEM	**NORMAL HYDROPONIC SYSTEM**
Crop environment	The plant can be place in a greenhouse or any control environment but user can control to maintain the optimal condition (smart environmental condition).	In order to make the plant survive, the environment must suitable for the plant or put the system in a greenhouse.
Control	Using the actuator as a device to do the controlling task. For example, if temperature rises above the optimal temperature, the fan will be switch on automatically.	Usually people install timer and the hydroponic follow the timing in the timer, no matter it is actually needed or no.
Monitor	With the help of sensor, the data can be collect timely and accurate.	No sensor used or the sensor can be implementing into the system.
Operation	Need expertise, since it is related to electronic and coding.	No need expertise in coding.

2.2.2 Recent Research Using Modern Monitoring system in Agriculture

In this section, the recent studies of IoT in agriculture were listed and the aspect that will be listed is parameter that was measured, the connectivity and user interface used and the analysis and discussion.

TITLE	PARAMETER	CONNECTIVITY	USER INTERFACE	ANALYSIS AND DISCUSSION
Internet Of Things For Planting In Smart Farm Hydroponics Style. (Pitakphongmetha et al., 2017)	1. Temperature and humidity using DHT11 2. Water level using ultrasonic sensor module)	Wi-Fi	1. Blynk for android application and Thingspeak as a cloud	• During first week, 48 trees dying because of high temperature • Then, SLAN mesh filter to reduce heat • Ultrasonic sensor monitored water level if the water level is low, the solenoid used for water addition to the system. • The Blynk used to control the greenhouse function automatically.
Fully Automated Hydroponic System For Indoor Plant Growth. (Palande, Zaheer, & George, 2018)	1. Electrical conductivity using EC probes 2. pH value using pH probes 3. Water temperature using water temperature sensor 4. Temperature and humidity using air temperature and humidity sensor	NRF24C01 + radios	1. Domoticz monitor data and control light if needed and it was run by Raspberry Pi 2. MySensors as a sensor network 3. Arduino IDE as a coding software	• Each log updated and demonstrated everyday using Domoticz • The data that shown in the software is the temperature and amount of light exposure. The switch on or off of light also being shown. • Water level and pH value that change throughout times because the plants grow. • The comparison between plant inside the system and outside was made and showed that the plant in system grow better than outside system

IoT Based Hydroponics System using Deep Neural Networks. (Mehra, Saxena, Sankaranarayanan, Tom, & Veeramanikandan, 2018)	Temperature and Humidity using DHT11 Water level using water level sensor Ambient light conditions using LDR	Wi-Fi	1. Raspbian Jessie torun the Raspberry Pi 2. Arduino IDE for the microcontroller 3. Google Firebase as cloud	• Get 5000 real-time data from hydroponic including pH, water level, temperature, humidity, light and water level • The tomato using system have a lot of height than traditional one
IoT and agriculture data analysis for smart farm. (Muangprathub et al., 2019)	Temperature, humidity and soil moisture.	Wi-Fi	1. smart farm 2. Kaset Yim mobile application 3. LINE application	• Data can be displayed in the mobile device • Farmer can control on and off button for watering and notify through LINE • It is a low cost proposed system • More product produce while farmer can do other task since there are more than one crop.

From all these studies that have been conducted, a good user interface or a good middleware is important because user need to read the data easily even though there are many. Wi-Fi was mostly being used in the wireless connection and temperature and humidity is the most popular parameters being measured. Using remotely controlled hydroponic system is helpful but is it better to use fully automated system to lessen human intervention. These studies also showed the advantages using cloud to compute data, easier to monitor the system. One study includes about costing, but is it for a huge and portioned farm, a study on low cost hydroponic should be considered.

CHAPTER 3

METHODOLOGY

3.1 System Integration

The study was carried out using two systems which are a timer system and a smart system. Both of the systems used in this study have implementation of sensors to obtain data and a microcontroller (ESP8266) that was connected to Wi-Fi and enable data sent into the platform (Favoriot). The monitoring data that was taken by the sensors can be trace in the platform as well as in the android interface (Blynk). One same environment was applied for both systems where the compartment used to put both systems was placed in the air-conditioned closed room.

Both of the systems also differ in controlling method where in the timer system, all actuators will follow the setting time to operate. As for the smart system, a set of coding was uploaded into the Arduino that was connected to the four way channel relay module to move the actuator according to the current situation.

3.1.1 Hardware

The sensors used are temperature and humidity sensor (DHT11), light intensity sensor (LDR) and water level sensor. There are three actuators which are NodeMCU, Arduino UNO and Arduino Nano. All of the sensors get powers supply same as the microcontroller. All actuators used were connected to a 12A power supply and the entire three microcontrollers used USB cable to have power supply

3.1.1.1 Microcontrollers

Arduino boards are suitable to be the brain for smart system, it is inexpensive and easy to use because it is open sources making it easy to identify and solved problem regarding Arduino. Microcontrollers were used for both systems to store and run the programming setups respective to the type of system. In timer system, people usually go for the digital timer not the timer based Arduino coding. In this study, the actuators designed for Arduino connection which is the reason why microcontrollers are used for both systems.

The different type of microcontrollers (Figure 5) was used due to number of pins. Different from smart system, only few pins were used in timer system, so Arduino Nano are preferable since it is physically small and no need of huge space. The NodeMCU was used because it is consists of ESP8266 that enable the connection to Wi-Fi and was connected to all sensors, except for LDR that used to detect light for whole environment.

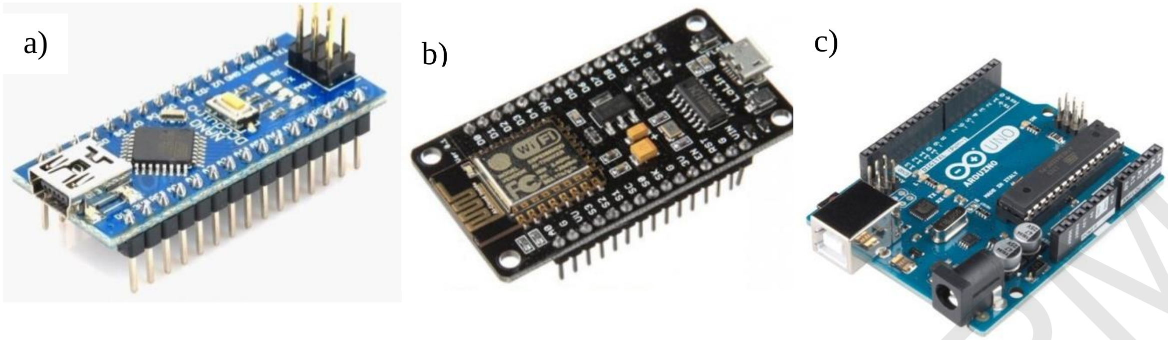


Figure 5 : a) Arduino Nano, b) NodeMCU, c) Arduino UNO

3.1.1.2 Sensors

The Temperature & Humidity Sensor, DHT11 (Figure 6) features a calibrated digital signal output with temperature and moisture sensors. It can guarantee high reliability and outstanding long-term stability. Other than it has quick reaction, anti-interference capacity, and benefits of high cost performance, it is also small in size, require low energy and signal transmission range up to 20 metres. (Nightfire Electronics, 2000). This sensor was placed next to the hydroponic so it can detect the air from the above system.

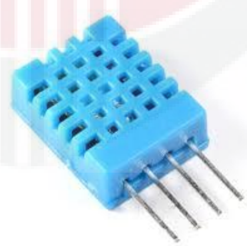


Figure 6 : DHT11

Light intensity sensor, LDR (Figure 7) placed above the system where LED was placed because it possibly detects the light reading outside the compartment which is fault. Different from other sensor, there are three LDR used in this study which two LDR used to obtain data for the cloud and the other one was used in smart system to detects the light existence in the room. This condition will be explained in section 3.2.

Figure 7 : LDR

The water level sensor used in this study is a non-contact type (Figure 8). It is suitable to be used for all types of solution. Since NFT hydroponic system was used, the non-contact water level even preferable because no need to adjust length of wire to avoid water contact. The sensor must be centred with the level of water, so the dimensions of hydroponic container was scaled where from 8 cm of height, the sensor will be placed at 4.5 cm from the top.

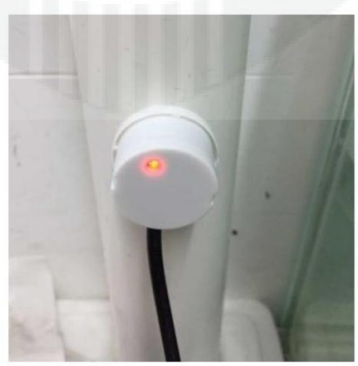


Figure 8 : Non-contact water level sensor

3.1.1.3 Actuators and Power Supply

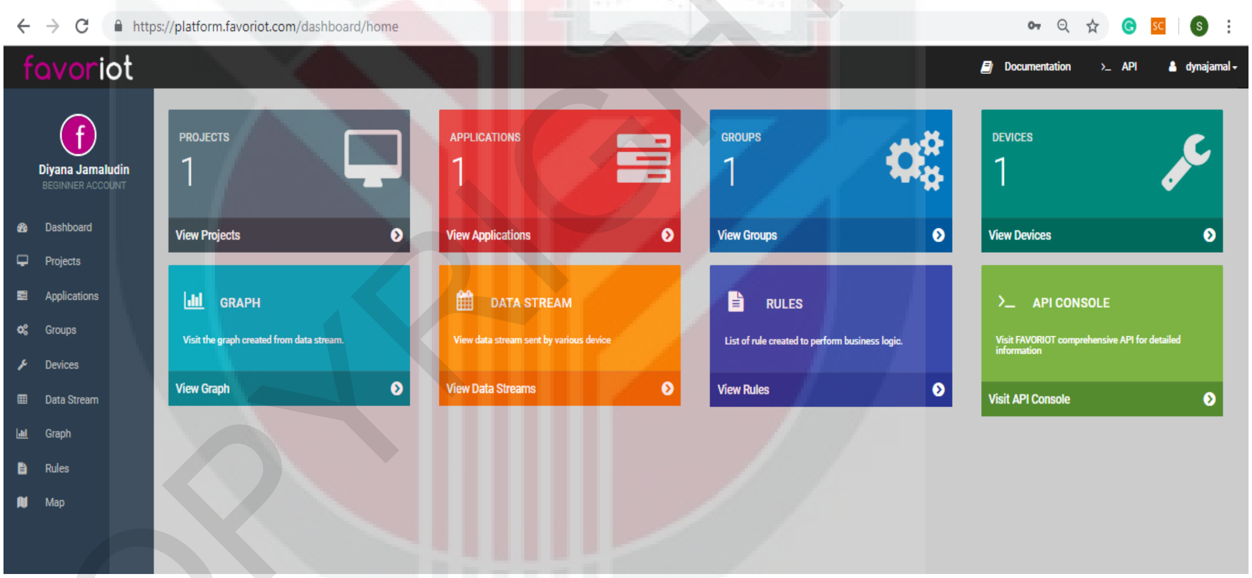
The power supply for the microcontrollers were using USB, the USB wire was plugged to the main power supply. As for the actuators, the 12A power supply was used. The microcontrollers cannot be connected to 12A power supply because the power is too big for the board which not preferable for a long term use. Actuators are fan, pump and LED, all are 5V DC and connected to relay module. The pump and LED have no wires originally, so the wires need to be solder, as for the fan, it already come with wires.

3.1.2 Connectivity

The wireless connectivity used is WiFi. The range of WiFi is limited until 200m but less current consumption compared to LTE, 3G and 2G. The compartment of the hydroponic system was placed within 200m and able to connect to WiFi. Since the WiFi used was "Engineering FASA 2" only limited to only two devices, the Connectify application was installed to make all devices can be connects. In the coding, the name and the password of the WiFi need to be include in so that the ESP8226 able to get access of the internet.

3.1.3 Middleware

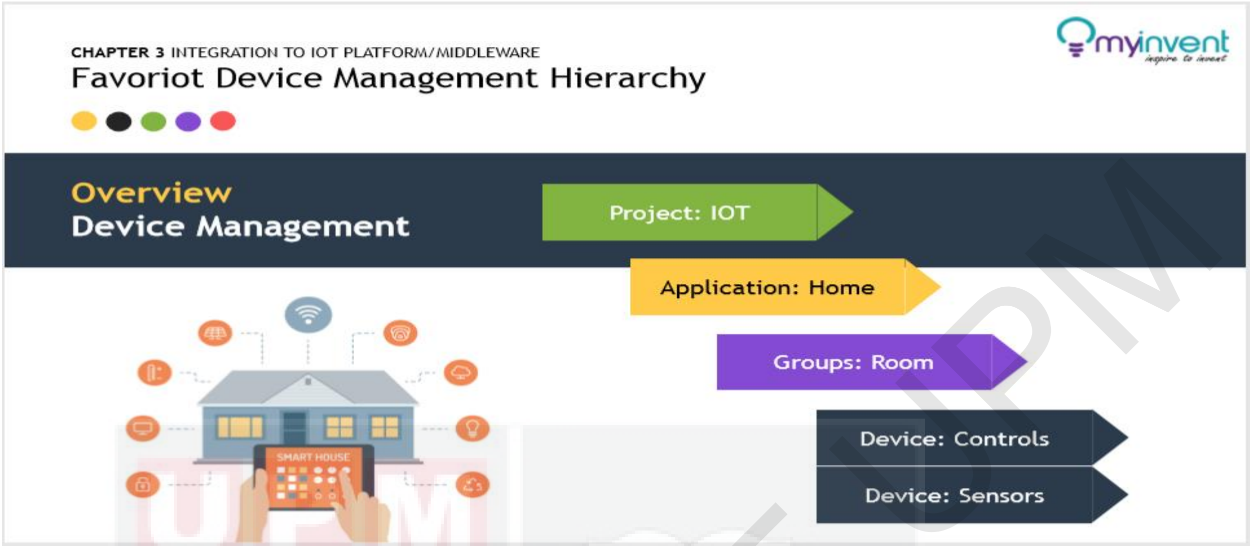
The middleware used in this project is Favoriot (Figure 9), it is an IoT platform designed for any type of IoT projects. The platform is able to support the integration of data from actuators and sensors on the internet. The platform able to collect and store data and it is easier to track all data where the real time data were aggregated into the data streams and the historical data also can be access for data aggregation. (Favoriot, 2019)



Source : myinvent arduiono IoT workshop notes

Figure 9 : The home of the Favoriot platform

The Favoriot is platforms that need to be subscribed according to the project requirement and after subscription, the account can be created where user ID and password needed to login. In this platform, there is a hierarchy (Figure 10) of device management as shown in figure 6 and user needs to create ID for each hierarchy.



Source : myinvent arduiono IoT workshop notes

Figure 10 : The hierarchy in Favoriot and its examples

The device ID will be used in the programming so that data will recognize the device and data can be placed in the data streams. Authenticate token, device ID, WiFi user ID and password are the essential characters that need to be put in the programming and for Favoriot, the programming language must be in JSON format means all data from sensors must be convert into this format. Both protocol MQTT and HTTP can be support in Favoriot, in this study HTTP protocol was used.

3.2 Hydroponic System Setup

There are two hydroponic systems used in this study which one is for timer system and the other is for smart system. These hydroponic systems were placed in the detachable compartment that has four storey. The timer hydroponic system was placed on the first storey while the smart hydroponic system was in the third storey. The hydroponic systems have 10 holes each and Nutrient Film Technique (NFT) type was used in this study.

Both of the systems were installed with same amount, type of sensors as well as the sensors position. The light sensor placed on the top of the hydroponic to sense the presence of light while the temperature and humidity sensor placed next to the hydroponic to sense the cool air going down. The water level sensor was attached externally on the hydroponic system container.

In timer system, the time for actuators to operate are based on time where LED switch on from 5 pm until 8 am, pump switch on once a week for 1 minute and switch on fan at 12 am until 1 am. This timing is based on consideration of the condition in the room and time schedule when the room can be used.

In smart system, the controlling system was set in the optimum condition. The optimum conditions of hydroponic systems are as shown in table 4 and that condition will be set in the Arduino UNO. Figure 11 and Figure 12 show the flowchart for automated system and timer system respectively. The figures show the input and outputs of the system also the decision making that happen in the system when the environment is beyond the setting condition.

Table 3 : The list of optimum condition that was set for smart system

PARAMETER	SETTING CONDITION
Temperature	23˚C
Humidity	55% - 65%
Light	If light in the room was less than 1.98 v, the light must switch on
Water Level	3 liter or 4.5 cm from base.

Smart system:

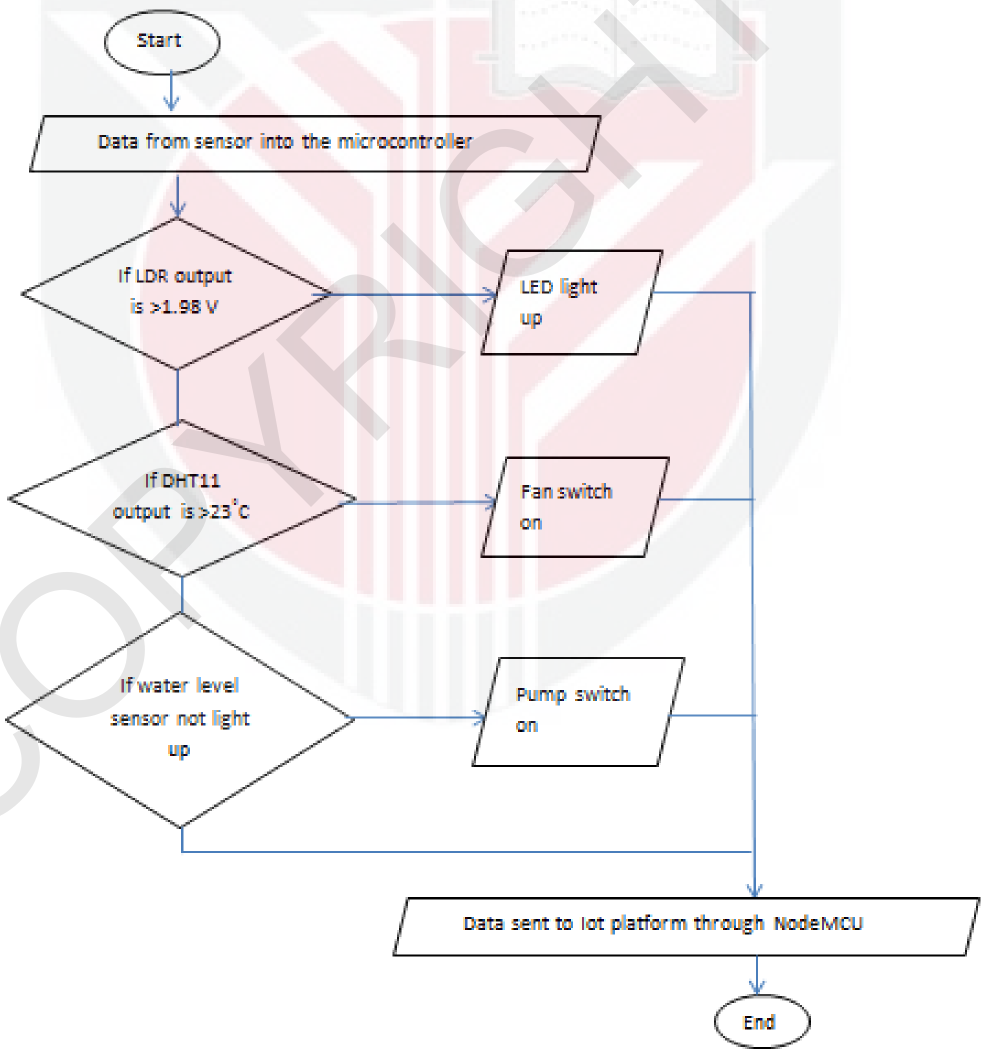


Figure 11 : The flowchart of smart system

Timer system:

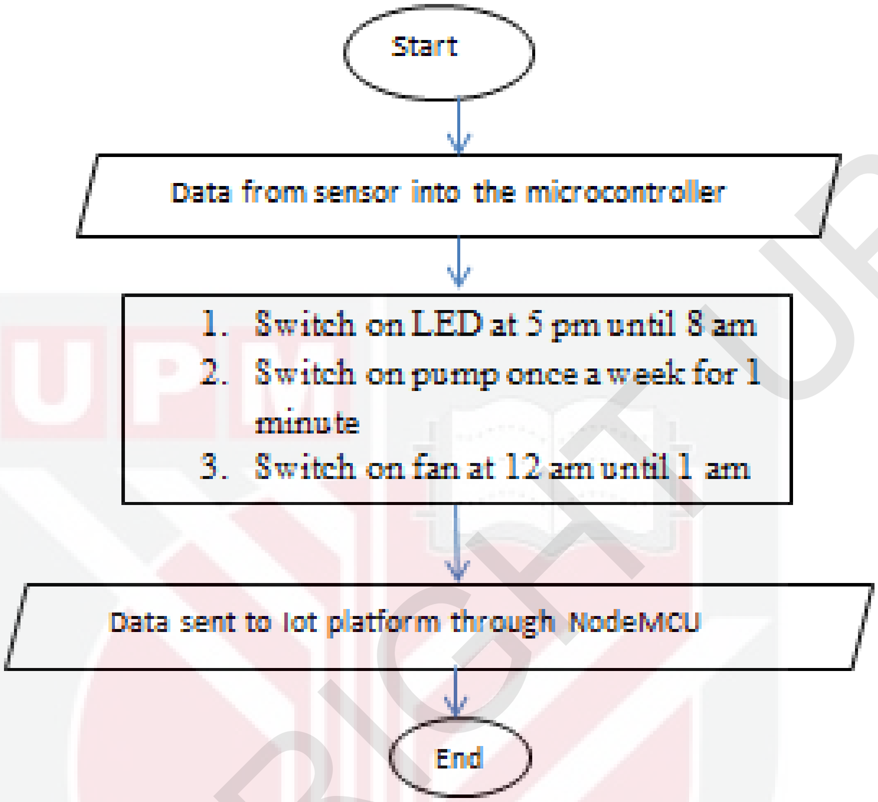


Figure 12 : The flow chart for timer system

3.3 Data Collection

The data that will be collect are result of the systems, reliability of data, cost for timer and smart system and system performance by manipulating environment condition. First, the result of the IoT system and this result will be collected by using monitoring data in the data streams and the dashboard because from this data will shows if the monitoring system using Iot really successful.

Second data collection is reliability of data taken from the DHT11 sensor and this data will be compared with the reading from portable sensor to identify the differences between the reading of the whole environment and the compartment reading. The cost of both systems will only include the hardware used for each system and the data from sensors will be taken when there is manipulation of environment.

The manipulation of environment divided into three parts which are manipulation of temperature, manipulation of presence of light and the manipulation of water level. The manipulation of temperature is where the air conditioner will be switch on and switch off same goes to the presence of light, the light in the room will be on and off. The water level in the hydroponic container was reduced by 150 ml to record the time taken to reach the original water level.

3.4 Data Analysis

T-test will be used to determine the significant difference between the reading of portable sensor and the DHT11. T-test is the parametric tests that function as conducting statistically significant tests in the testing of hypotheses. (Statistics Solution, 2019). The t-test was done in Excel. The percentage formula as shown in equation 1, will be applied to determine the percentage of additional cost of smart monitoring system compared to the timers system.

Equation 1 : Additional cost percentage formula

$$\% \text{ of additional cost} = \frac{smart - timer}{timer} \times 100$$

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Sensor Development and Integration

Figure 13 shows the connections to all sensors except for LDR that used to detect light for whole environment. It is only used in smart system. In table 4, the pins for each sensors are listed, each sensors connected to the positive (GND) and positive (3V3), it was labelled in the board NodeMCU.

Figure 14 shows the connection in the smart system, the sensors in this system was connected or 'jumped' to the breadboard used for NodeMCU connection so it can obtain data to send to the cloud and detect the current condition so the actuators can response. In the Arduino UNO, LDR was at pin A0, DHT11 at A1 and water level sensor A2.

Figure 15 is the connection of each hardware that involved to develop the system, as can be seen, all sensors connected to the cloud or the IoT platform. This can be achieved when the ESP8266 in the NodeMCU are connected to the internet. The labels used in the table are the label used for each sensor indicating '1' for smart system and '2' for timer.

Table 4 : The pins number for each sensor

PINS	SENSORS
D3	LDR1
D4	DHT1
D5	WATER LEVEL SENSOR 1
D6	DHT2
D7	LDR2
D8	WATER LEVEL SENSOR 2

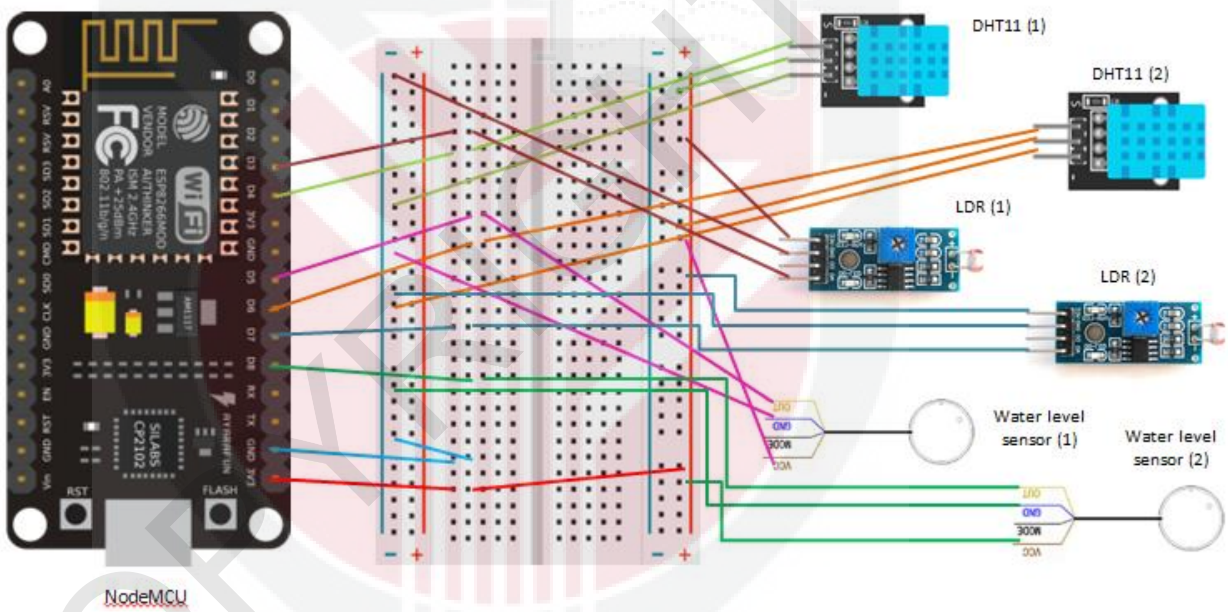


Figure 13 : Connection between sensors and NodeMCU

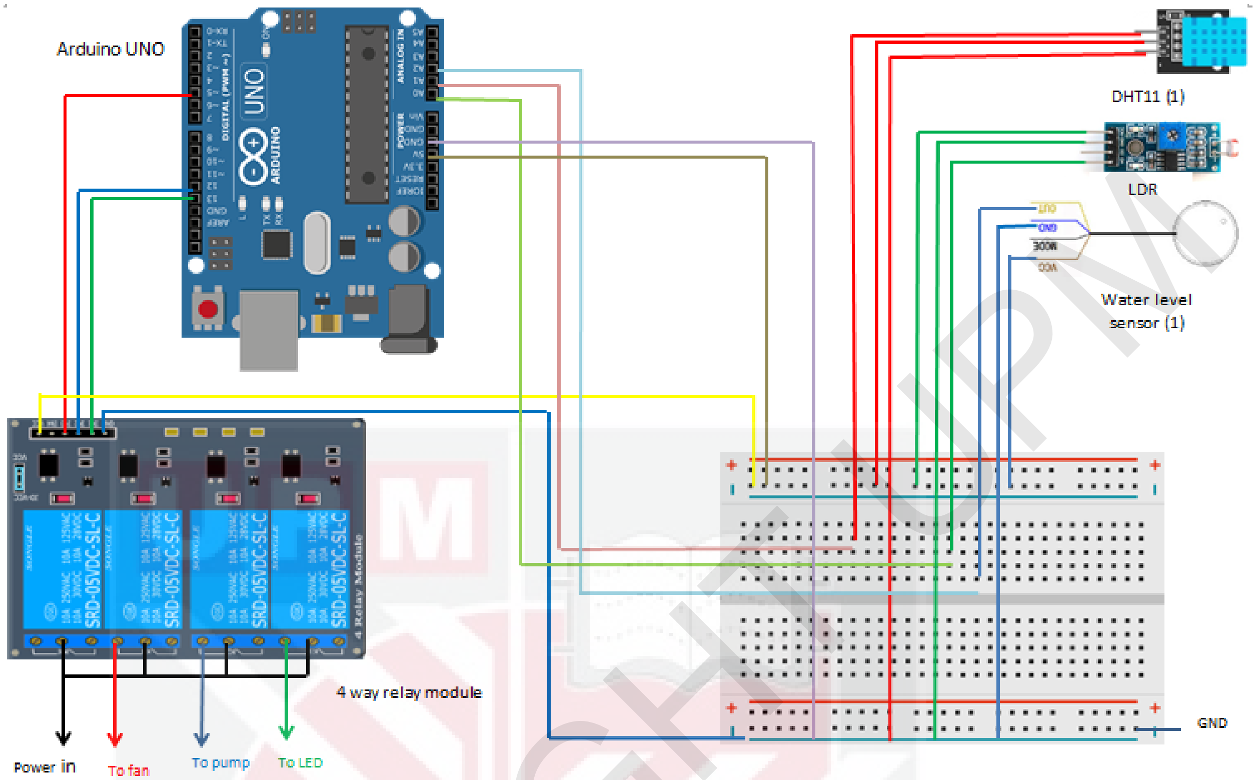


Figure 14 : Connection between sensors, actuators and Arduino UNO

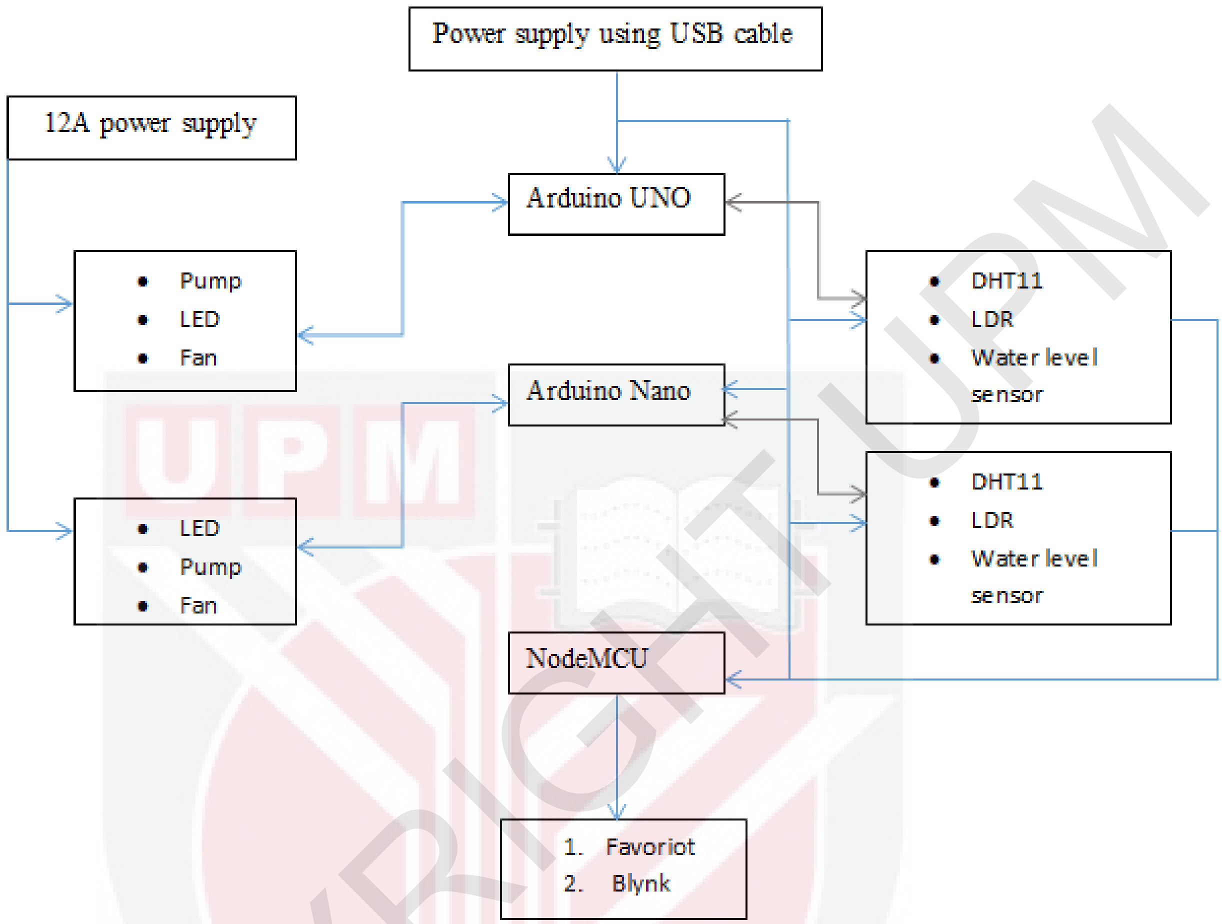


Figure 15 : The flowchart of the connection

Figure 16 is the circuit diagram for the actuators. Once soldered, the wires have positive and negative wires. The positive wires will be connected to the relay module where; int1 is for LED, int2 is for pump and int3 is for fan. The relay will open the switch when needed and it was connected to the power supply, positive side. The negative wires combined into one wire using wire connector that one wire goes to the negative part of the power supply.

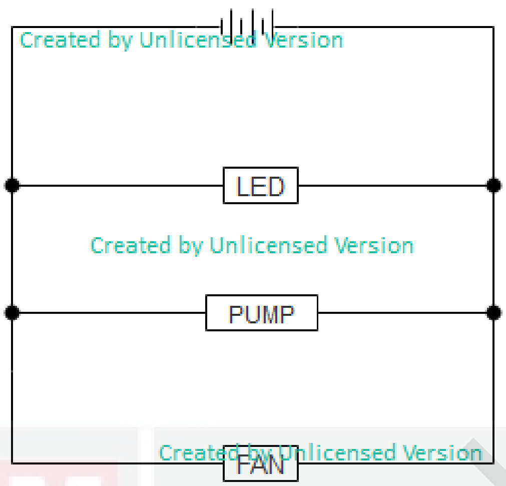


Figure 16 : The circuit for actuators and 12A power supply

4.2 System Connection

Real-time data such as temperature, humidity, light intensity and water level from timer and smart hydroponic systems were sent to Favoriot and can be display in platform. As long as the systems connected to the Wi-Fi, the monitoring data will constantly updated and demonstrated each day.

Figures 17 and 18 show the monitoring data for all parameters at noon (12:00 pm) in graph form. The data at noon was chosen because it is showing the most differences between two systems among the data at other time. Temperature2, light2, humidity2 and water level2 are the names used for the data that coming from timer system whereas temperature, light, humidity and water level refers to the data of smart system. It is easier to observe all the data in graph section in Favoriot where types of graph can be change and able to add more graph by clicking the purple button at the top right.

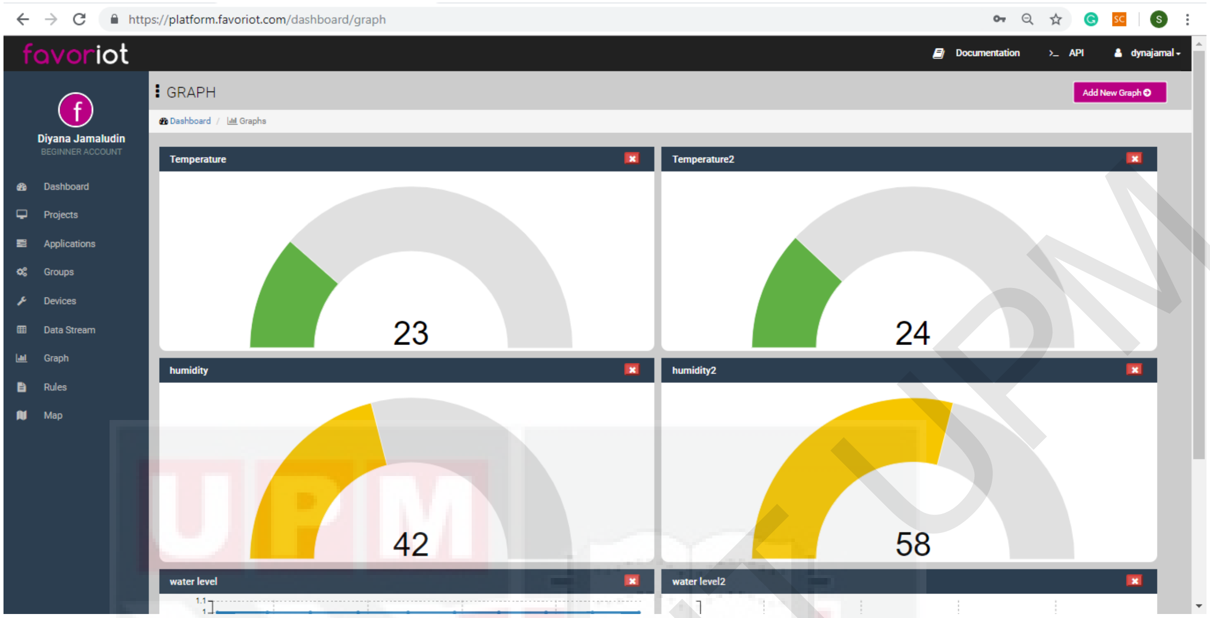


Figure 17 : The monitoring data for temperature and humidity

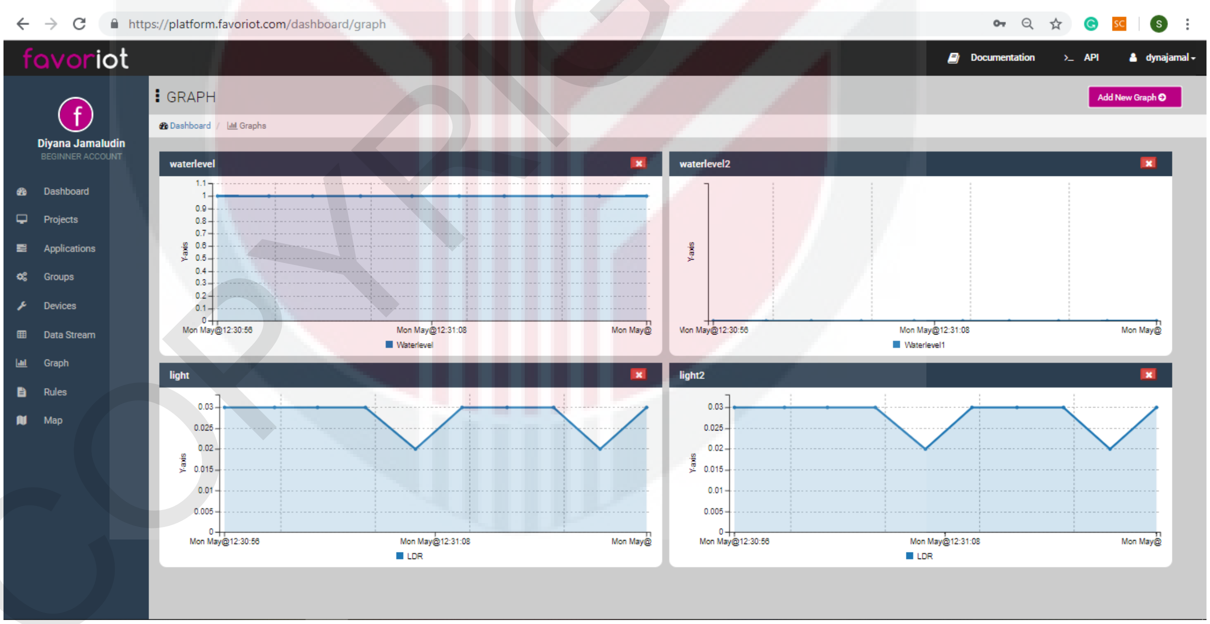


Figure 18 : The monitoring data for water level and light intensity

There are differences that can be observed between the data from both systems as shown in figures 17 and 18. The temperature values differ at 1˚C whereas the humidity values differ at 16%, this is because the fan in timer system are not functioning at this hour while in smart system, the fan was switched on because the temperature was more than setting condition. When fan react based on time, there are chances the temperature of hydroponic environment is high which can cause the plant to die. The results have been shown in productivity of the vegetables that died in the first week due to high temperature, leaving 0% of tree survival. (Pitakphongmetha et al., 2017), only after the cooling system was installed, 22 trees survived from 48 seeds planted.

Water level in smart system was 1 and in timer it was 0 which indicates the water level in smart system was in the desired level and this is because the pump will immediately pump the water when needed. This will minimize the uses of water and avoid the overflow. The room was in bright condition and the timer has not yet reaches the time to switch on LED. So, LDR for both will read same data.

The platform is accessible anytime and anywhere which helps ease the monitoring method. The data easily recorded in the platform also easy to analyzed the environment condition rather than spending hours or tons of effort collecting all data manually. Hence, the system connection is successful to shorten time and fewer tasks conducting the experiment.

4.3 Reliability of Data

The reliability of output data from sensor was tested by conducting change of temperature experiment. In this experiment, two sensors were used which are DHT11 and a portable sensor which then the reading from both was compared. The portable sensor must be trustable, so calibration needs to be carried out.

4.3.1 Calibration of Portable Sensor

The portable sensor used is E-SUN ETP101 Digital 3 in 1 Thermometer Hygro and Clock (Figure 19), This temperature measurement device can display real-time temperature, humidity, time and date. The specifications are as follows;

- Temperature : -10°C to 50°C
- Humidity : 10 % RH to 90 % RH
- Clock : 12/24 hours, with alarm function
- Power : 1.5V AAA single
- Dimensions(W×L×T) : 101 × 108 × 23 mm
- Weight : 130g (with battery)

Figure 19 : E-Sun ETP101 Digital 3 in 1 Thermometer Hygro and Clock

The portable sensor was calibrated and checked with a mercury thermometer. The thermometer used has been tested for ice temperature reading where the temperature for ice is fixed, 0˚C and apparently the thermometer reads 0˚C. Next, the portable sensor and the thermometer both tested to obtain the reading a place that contains ices, to make it fair, the thermometer end was not directly touch the ice since the sensor in the portable device was embedded inside so it is impossible to get to reach zero . As a result, the different between the readings was 1˚C, so the value from portable sensor must be ∓1. This means the value from portable sensor are trusted and the comparison between sensor and portable one can be made.

4.3.2 T-Test Results

. The air-conditioner temperature was manipulated in certain time and data from both sensors were recorded and compared. The data from DHT11 and the portable sensor were recorded under different level temperature where air-conditioner was set at starting point of 25˚C and lowered one degree for each 10 minutes.

Table 5 : The portable sensor reading and the Arduino reading for temperature

PORTABLE SENSOR (˚C)	ARDUINO (˚C)
25.6	24
25.1	25
24.9	24
24.5	24
24.3	24
23.4	23
23	23
22.8	23
22.8	23
22.8	23
22.8	23

Table 6 : The analysis results of t-test for temperature data

t-Test: Two-Sample Assuming Unequal Variances		
	portable sensor (˚C)	*arduino (˚C)*
Mean	23.81818182	23.54545455
Variance	1.167636364	0.472727273
Observations	11	11
Hypothesized Mean Difference	0	
df	17	
t Stat	0.706244121	
P(T<=t) one-tail	0.244801607	
t Critical one-tail	1.739606726	
P(T<=t) two-tail	0.489603214	
t Critical two-tail	2.109815578	

Table 7 : The portable sensor reading and Arduino reading for humidity

PORTABLE SENSOR (%)	ARDUINO (%)
47	38
48	44
46	40
45	38
44	38
44	37
44	37
44	37
44	37
44	37
44	37

Table 8 : The analysis results of t-test for humidity data

t-Test: Two-Sample Assuming Unequal Variances		
	Portable sensor (%)	***Arduino (%)***
Mean	44.90909091	38.18181818
Variance	2.090909091	4.563636364
Observations	11	11
Hypothesized Mean Difference	0	
df	18	
t Stat	8.649204771	
P(T<=t) one-tail	3.96392E-08	
t Critical one-tail	1.734063607	
P(T<=t) two-tail	7.92783E-08	
t Critical two-tail	2.10092204	

By using t-test, the significant difference value can be determine and the value of the significant difference for one-tail must be less than or equal to 0.05 to declare there is significant difference between the data that need to be considered. Table 6 the significant difference is more than 0.05 while in table 8 the significant difference is less than 0.05. So, only data taken for humidity have the significant difference which indicates the condition in the compartment was different from room environment. Thus, the data taken are not reliable.

4.4 Calculation of Cost

The calculation of the cost of both systems was carried out because it will assist decision making by comparing total cost for both. The example of decision making that affected by cost is planning and control, whether to make the things or buy, etc. Costing element is important in the complexity of business and increasingly changes in industry. (College Accounting Coach, 2006). The cost of hardware used in the smart system is listed in the table 9 below. In the same table, the price also compared with the hydroponic system using timer. The prices are based on the shop where the hardware was bought and the price may differ from other shop and the prices listed below only for hardware cost not including other things needed for example wires, screwdriver, etc.

Table 9 : Name of each hardware in system and its price

HARDWARE	PRICE (RM)	
	SMART	TIMER
4 way relay module	15.00	
DHT11	6.50	6.50
LDR	5.60 x 2 = 11.20	5.60
Water level sensor	39.90	39.90
LED	14.95	14.95
Fan	15.00	15.00
Pump	36.60	36.60
Power supply	32.00	
NodeMCU	35.00	35.00
Arduino UNO	45.00	
Timer		24.00 x 3 = 72.00
Total	251.15	225.55

$$\% \text{ of additional cost} = \frac{smart - timer}{timer} \times 100$$

$$\% \text{ of additional cost} = \frac{251.15 - 225.55}{225.55} \times 100$$

$$\% \text{ of additional cost} = 11.35\%$$

By using the formula to find percentage, the percentage of additional cost for smart system was known as 11.35%. In economic perspective, the improvement cost is considered good up until 30% and the smart system have a low cost investment.

By comparison, the cost to use timer in the system is cheaper than smart system however the cost to develop the smart system still consider low. It is definitely a good investment to go for smart system for hydroponic system because can minimize the uses of resources like water, electricity, etc. With smart system, the user can do other tasks without worrying the crops since the system act smartly towards the current condition. This system also believed will need few labor and beneficial for long term.

4.5 Assessment of System Performance by Manipulating Environment Condition

There are three tests were conducted to observe and assess the performance of the automated controlling system which are controlling temperature test, controlling light test and controlling water level test. The results of the assessment were observed from the reading of the sensors and from those data, the decision was made whether the system is functioned as it was set.

4.5.1 Controlling Temperature

The ambient temperature in the enclosed room is influenced by the temperature of the air conditioner. So, the test of controlling temperature was carried out by switching the air conditioner on and vice versa. Other than recording data from DHT11, fan observation was also performed to see if the fan works when the temperature exceeds 23˚C as specified in the setting condition. The portable sensor was used to compare the reading of the sensor.

Table 10 : The sensors reading when switch off air-conditioner

PORTABLE SENSOR	DHT11
Temperature : 29˚C Humidity : 73%	Temperature : 30˚C Humidity : 91%

Table 11 : The sensors reading when switch on air-conditioner

PORTABLE SENSOR	DHT11
Temperature : 24˚C Humidity : 48%	Temperature : 24˚C Humidity : 63%

Table 11 show the reading of the sensor when air conditioner was switched off and table 10 shows the reading when air conditioner was switched off. The enclosed room was left one hour with air conditioner on, this step is to make sure the reading of the sensors are stable enough before taking the next data in table 11. The sensor reading showed a decreasing value of temperature and humidity and fan was in on mode because the temperature has not yet reached 23˚C. After few more hours left with lower temperature, the ambient temperature finally reached below 23˚C, and fan was switched off automatically.

4.5.2 Controlling Light

The atmospheres of the room when the light source is in control where the lights are turned off or switched on are divided into two conditions either dark or bright. Through sensor readings, the dark state is when the reading of the sensor is more than 1.98 V while bright state is when sensor reading is less than 1.98 V. when the light switched off, the LED in the compartment for automated control system should lights up immediately. The reading of the LDR was recorded when the presence of light was manipulated from dark to bright.

Table 12 : The LDRs reading when room is dark

ROOM LIGHT	COMPARTMENT
Dark	Bright

Table 13 : The LDRs reading when room is bright

ROOM LIGHT	COMPARTMENT
Bright	Light

There are two LDR used where one for whole room light and one for the compartment. From table 12, the 0.24 V indicating the compartment is in bright condition because the LED is switch on while from table 13, 1.02 V is the condition where LED was switch off. The room when in dark will have reading of 2.34 V from LDR while in bright, the reading is 0.58 V. Hence, light source control test has proven that the LED works immediately where the LED react with the presence of light in the room.

4.5.3 Controlling Water Level

The water level test was conducted to assess the performance of the water level sensor and pump to pump water into the hydroponic container. 150 ml water was reduced from the original water level which is 3 liter of water and during that process, the power supply was turned off and the water level sensor was not light up. When the power supply was turned on, the pump immediately pump water from the water tank into the hydroponic container until the water level sensor light up again and the process to pump in water only takes 4.97 seconds.

4.6 Problem and Limitation

There are several problem and limitation while running this project which describe as follows;

- The electronic components were damaged because overheated when leave for 24 hours with power supply was switched on. The micro component in the electronic component will be burned when electricity flow excessively and some electronic components was damaged because of wrong connection
- The UPM Wi-Fi connection was stable, however within 4:30 A.M until 5:30 A.M, the connection automatically disturbed and has to reconnect it again by authenticating. This caused the data loss within that range of time or until login is possible again.
- The connectivity used in this project need authentication and only allowable for two devices which makes the input of data always disturbed.
- The enclosed room was difficult to reach the desired condition which makes the fan was switch on for long time and reduced humidity. Also, this problem makes the process of taking data was difficult.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

The temperature, humidity, light intensity and water level sensors was successfully integrated with IoT system where data from those sensors were able to be sent into the cloud. The data were recorded in the cloud and can be monitored anywhere and anytime, this is good for future references and improving process. The comparison between two systems, timer and smart system was observed from the effectiveness of response toward the fluctuations in the environment. From the results, the smart system definitely more effective controlling the hydroponic environment. In terms of costing, the smart system need a higher investment than timer system but to upgrade from timer to smart system only have small percentage of additional cost. In conclusion, all three objectives are achieved.

5.2 Recommendations

There are several factors recommended for future research which describe as follows;

- The connectivity that will be used is an internet that do not need the authenticate for constant and non-disturbance data stream
- If system will be used for 24 hours, it needs a cooling system for the hardware so it will not get overheated.
- The actuator fan that will be used should be as big as the compartment size and was installed with a mist system for moisture. This type of fan able reduces more degree Celcius of temperature and reduces loss of humidity.

REFERENCES

Boonnam, N., Pitakphongmetha, J., Kajornkasirat, S., Horanont, T., Somkiadcharoen, D., & Prapakornpilai, J. (2017). Optimal Plant Growth in Smart Farm Hydroponics System using the Integration of Wireless Sensor Networks into Internet of Things. *Advances in Science, Technology and Engineering Systems Journal*, *2*(3), 1006–1012. https://doi.org/10.25046/aj0203127

Braun, A. T., Colangelo, E., & Steckel, T. (2018). Farming in the Era of Industrie 4.0. *Procedia CIRP*, *72*, 979–984. https://doi.org/10.1016/j.procir.2018.03.176

Mehra, M., Saxena, S., Sankaranarayanan, S., Tom, R. J., & Veeramanikandan, M. (2018). IoT based hydroponics system using Deep Neural Networks. *Computers and Electronics in Agriculture*, *155*(October), 473–486. https://doi.org/10.1016/j.compag.2018.10.015

Muangprathub, J., Boonnam, N., Kajornkasirat, S., Lekbangpong, N., Wanichsombat, A., & Nillaor, P. (2019). IoT and agriculture data analysis for smart farm. *Computers and Electronics in Agriculture*, *156*(November 2018), 467–474. https://doi.org/10.1016/j.compag.2018.12.011

O'Grady, M. J., & O'Hare, G. M. P. (2017). Modelling the smart farm. *Information Processing in Agriculture*, *4*(3), 179–187. https://doi.org/10.1016/j.inpa.2017.05.001

Ozturk, M., Jaber, M., & Imran, M. A. (2018). Energy-Aware Smart Connectivity for IoT Networks: Enabling Smart Ports. *Wireless Communications and Mobile Computing*, *2018*. https://doi.org/10.1155/2018/5379326

Palande, V., Zaheer, A., & George, K. (2018). Fully Automated Hydroponic System for Indoor Plant Growth. *Procedia Computer Science*, *129*, 482–488. https://doi.org/10.1016/j.procs.2018.03.028

Pitakphongmetha, J., Boonnam, N., Wongkoon, S., Horanont, T., Somkiadcharoen, D., & Prapakornpilai, J. (2017). Internet of things for planting in smart farm hydroponics style. *20th International Computer Science and Engineering Conference: Smart Ubiquitos Computing and Knowledge, ICSEC 2016*, (December). https://doi.org/10.1109/ICSEC.2016.7859872

Rahman, L. F., Ozcelebi, T., & Lukkien, J. (2018). Understanding IoT Systems: A Life Cycle Approach. *Procedia Computer Science*, *130*, 1057–1062. https://doi.org/10.1016/j.procs.2018.04.148

Razzaque, M. A., Milojevic-Jevric, M., Palade, A., & Cla, S. (2016). Middleware for internet of things: A survey. *IEEE Internet of Things Journal*, *3*(1), 70–95. https://doi.org/10.1109/JIOT.2015.2498900

TongKe, F. (2013). Smart Agriculture Based on Cloud Computing and IOT. *Journal of Convergence Information Technology*, *8*(2), 210–216.

https://doi.org/10.4156/jcit.vol8.issue2.26

Verdouw, C. (2016). Internet of Things in agriculture. *CAB Reviews: Perspectives in Agriculture, Veterinary Science, Nutrition and Natural Resources*, *11*(035). https://doi.org/10.1079/pavsnnr201611035

Zedadra, O., Guerrieri, A., Jouandeau, N., Spezzano, G., Seridi, H., & Fortino, G. (2018). Swarm intelligence-based algorithms within IoT-based systems: A review. *Journal of Parallel and Distributed Computing*, *122*, 173–187. https://doi.org/10.1016/j.jpdc.2018.08.007

Bassel Al Homssi, A. A.-H. (2018, July 5). *IEEE Xplore Digital Library*. Retrieved from IEEE.org Web site: https://ieeexplore.ieee.org/document/8403749

College Accounting Coach. (2006, July 16th). *Costing and reasons for its importance*. Retrieved from College Accounting Coach Website: http://basiccollegeaccounting.com/2006/07/costing-and-reasons-for-its-importance/

Favoriot. (2019). *Favoriot platform : Favoriot*. Retrieved from Favoriot Web site: https://www.favoriot.com/favoriotplatform

Imran Ali Lakhiar, G. J. (2018). Journal of Sensor. *Monitoring and Control Systems in Agriculture Using Intelligent Sensor Techniques: A Review of the Aeroponic System*, 18.

Matthews, A. (2017, January 13th). *The complex nature of IoT devices and cyber security*. Retrieved from Electronic Specifier Web site: https://wireless.electronicspecifier.com/iot-1/the-complex-nature-of-iot-devices-and-cyber-security

Nightfire Electronics. (2000, October). Temperature and Humidiy Sensor (DHT11). Florida, Ocala, USA.

Statistics Solution. (2019). *T-test : Statistics Solution*. Retrieved from Statistics Solution Web site: https://www.statisticssolutions.com/t-test/

APPENDIX

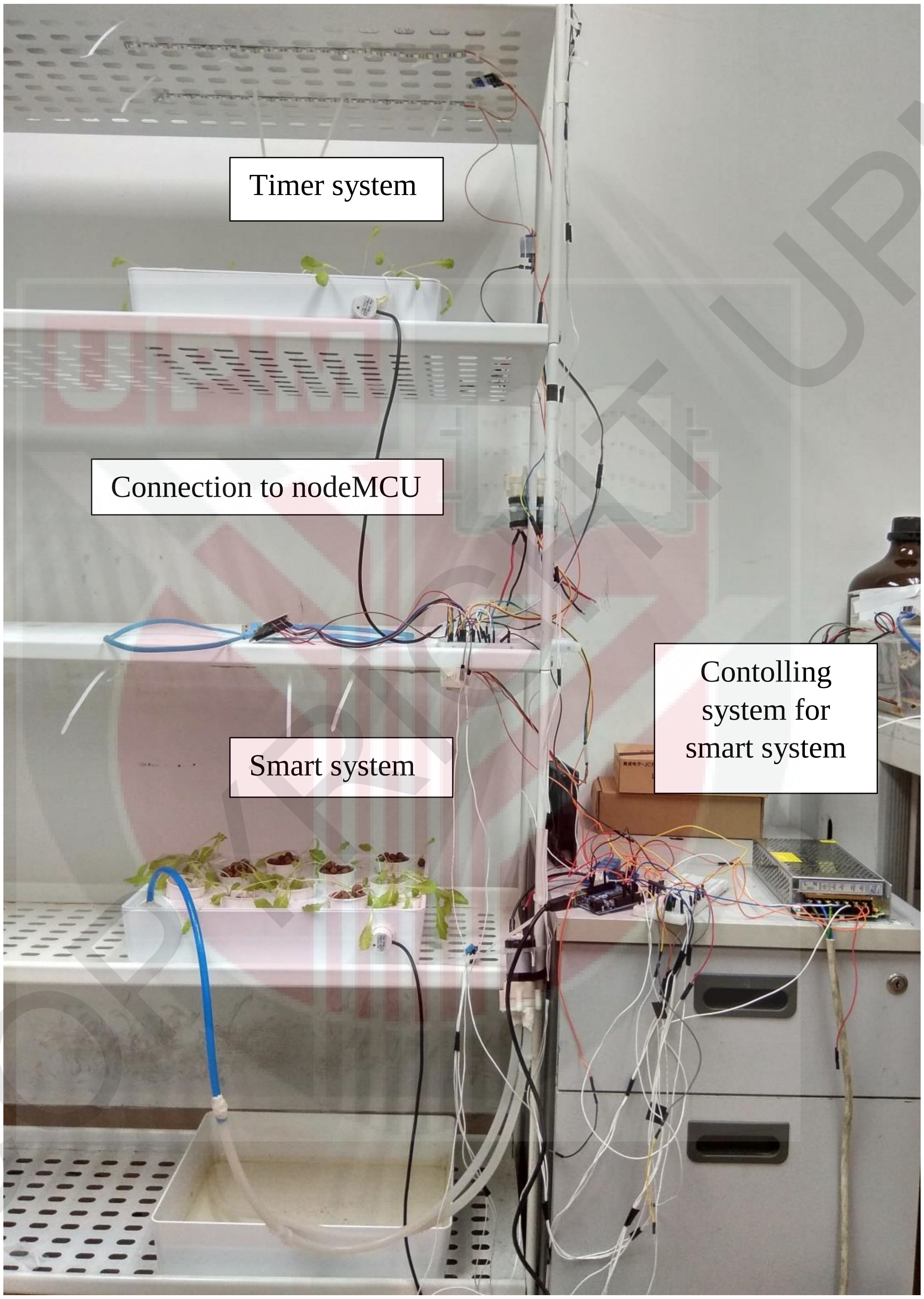


Figure 20 : The whole set up of hydroponic systems

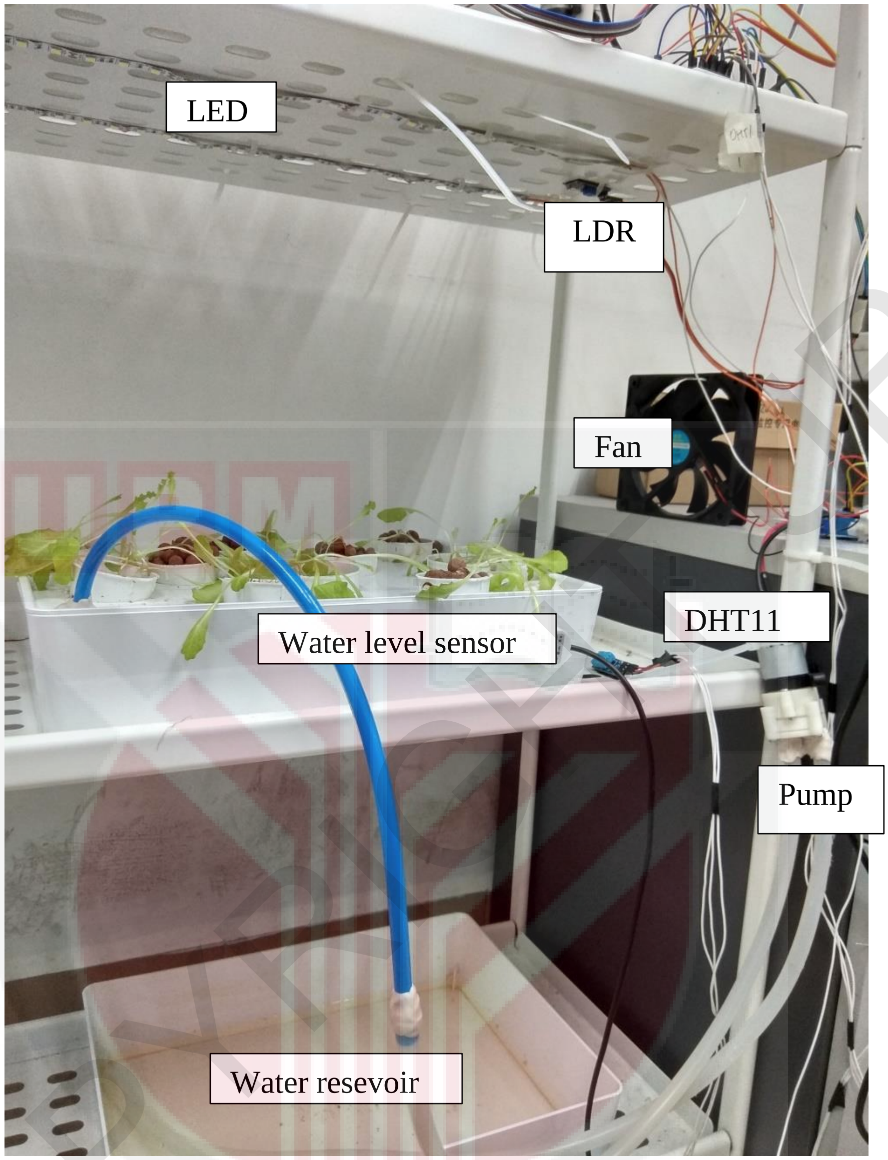


Figure 21 : The smart system

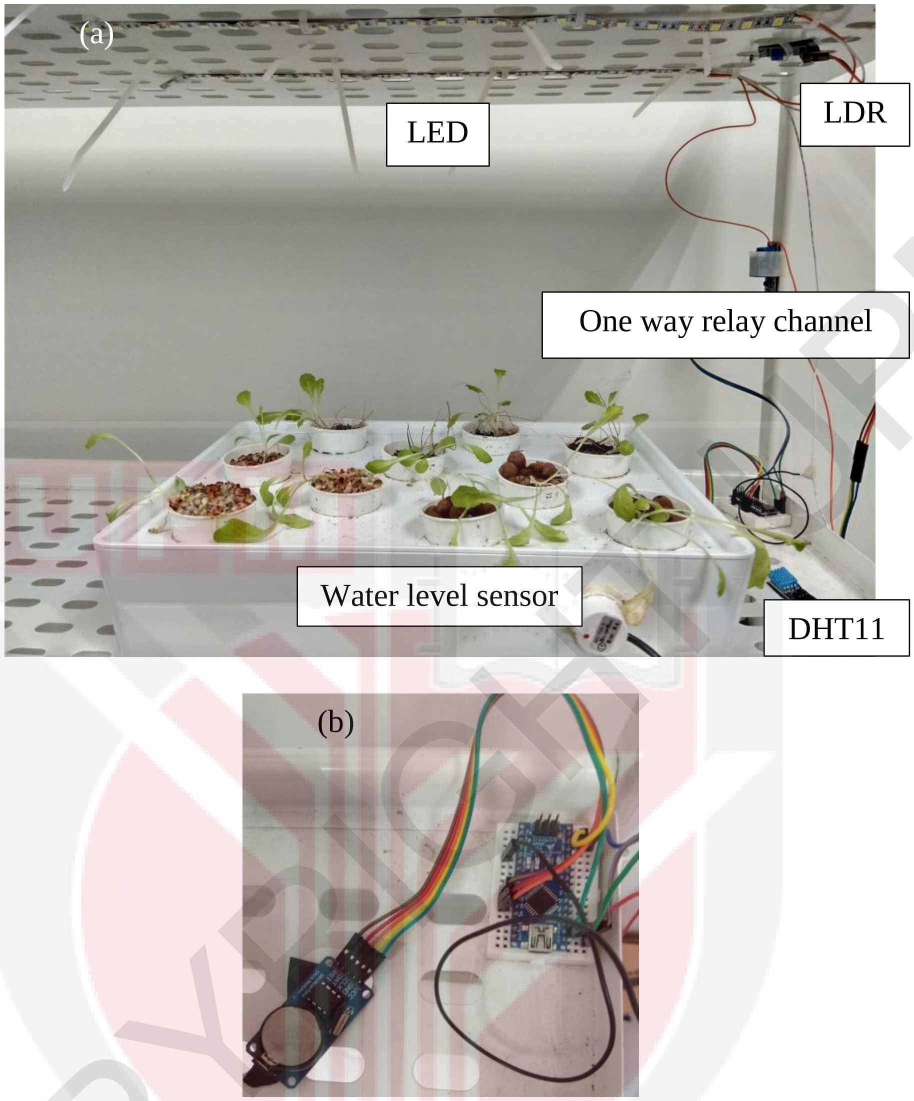


Figure 22 : (a) Timer system; (b) Arduino Nano and RTC

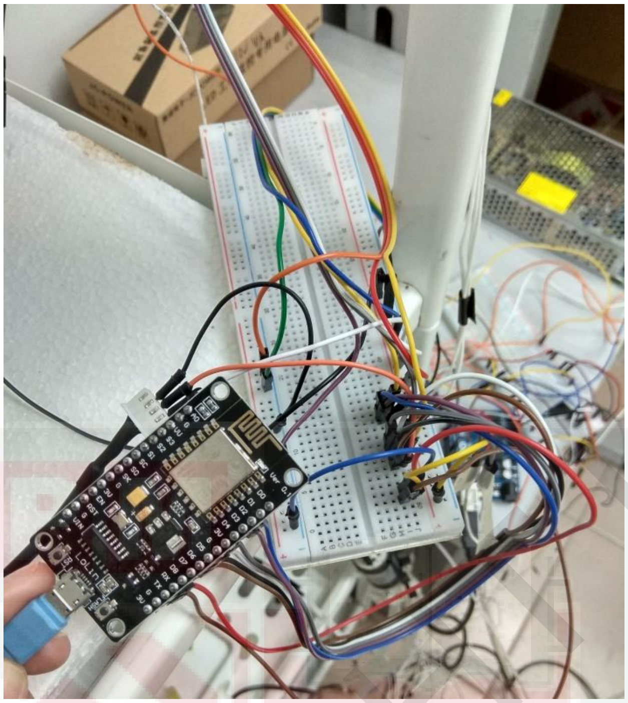


Figure 23 : The NodeMCU and the connections

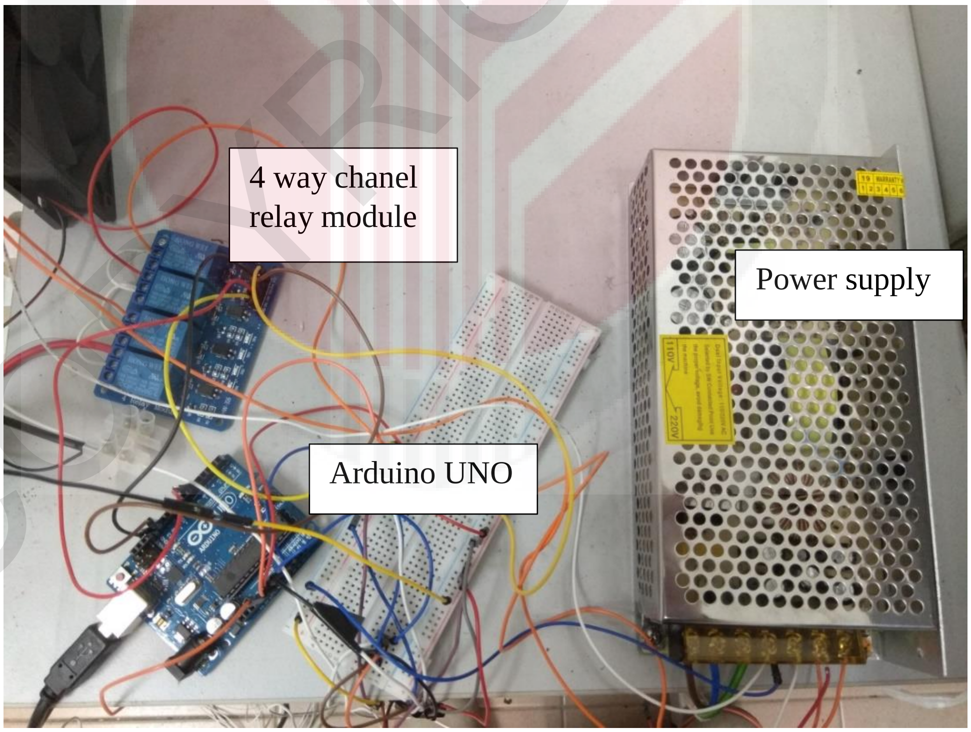


Figure 24 : The controlling system for smart system

```
#include <Wire.h>
#include "RTClib.h"
#define LAMP 13
int stat;
RTC_Millis rtc;

void setup () {
    Serial.begin(9600);
    // following line sets the RTC to the date & time this sketch was compiled
    rtc.begin(DateTime(__DATE__, __TIME__));

    pinMode(LAMP,OUTPUT);

}

void loop () {
    DateTime now = rtc.now();

    Serial.print(now.year(), DEC);
    Serial.print('/');
    Serial.print(now.month(), DEC);
    Serial.print('/');
    Serial.print(now.day(), DEC);
    Serial.print(' ');
    Serial.print(now.hour(), DEC);
    Serial.print(':');
    Serial.print(now.minute(), DEC);
    Serial.print(':');
    Serial.print(now.second(), DEC);
    Serial.println();

    Serial.print(" seconds since 1970: ");
    Serial.println(now.unixtime());

    // calculate a date which is 7 days and 30 seconds into the future
    DateTime future (now.unixtime() + 7 * 86400L + 30);

    Serial.print(" now + 7d + 30s: ");
```

```
    Serial.print(" now + 7d + 30s: ");
    Serial.print(future.year(), DEC);
    Serial.print('/');
    Serial.print(future.month(), DEC);
    Serial.print('/');
    Serial.print(future.day(), DEC);
    Serial.print(' ');
    Serial.print(future.hour(), DEC);
    Serial.print(':');
    Serial.print(future.minute(), DEC);
    Serial.print(':');
    Serial.print(future.second(), DEC);
    Serial.println();

    Serial.println();
    if (now.hour()>= 8 &&now.hour()<=16 )
    {
      stat=1;

    }
    else
    {
       stat=0;
    }
     Serial.println(stat);
     if (stat==0)
     {
      //on
      digitalWrite(LAMP,HIGH);
     }
     if (stat==1)
     {
      //off
      digitalWrite(LAMP,LOW);
     }
    delay(500);

}
```

Figure 25 : Coding for timer

```
#include <SPI.h>

#include "DFRobot_PH.h"
#include <EEPROM.h>
#define BLYNK_PRINT Serial
#include <BlynkSimpleEsp8266.h>
#include <ESP8266WiFi.h>
#include "dht.h"
dht DHT;

char auth[] = "d1b506fe974b4f1ba0d3e74729141dba";
char ssid[] = "iyaiyayoo";      // change it!
char pass[] = "12345678";  // change it!
const String myDevice = "deviceDefault@dynajamal"; // change it!
String apikey = "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6ImR5bmFqYW1hbCIsInJ1YWRfd3JpdGUiOnRydWUsImlhdCI6MTU0NTM2MTUwMn0.eS6fPdIoZRJM8hUAdo4oVDT09Q9-jBDWeq0S7uF4MP8";
char serverAdd[] = "api.favoriot.com";

int status = WL_IDLE_STATUS;
WiFiClient client;
//#define dht_apin D4     // What digital pin we're connected to
//#define LDRpin D3
//#define LEVELpin D5

WidgetTerminal terminal(V0);
#define dht_apin D4     // What digital pin we're connected to
#define LDRpin D3
#define LEVELpin D5
#define dht_apin1 D6     // What digital pin we're connected to
#define LDRpin1 D7
#define LEVELpin1 D8
#define PH_PIN D2
///////////////////////////////////////////////////////////////////////////
float voltage,phValue,temperature = 25;
DFRobot_PH ph;
//////////////////////////////////////////////////////////////////////////////////////////
BlynkTimer timer;
//////////////////////////////////////////////////////////////////////////////////////////
```

```
BlynkTimer timer;
//////////////////////////////////////////////////////////////////////////////////////////
class distSensor
{
  /* Class member variables */
  /* These are initialized at startup*/
  int sensorPin;
  long interval;

  /* These maintain the current state */
  volatile float sensorVal = 0;
  volatile unsigned long previousMillis;

  /* Constructor - create a distSensor and intializes the member
  variables and state*/
  public:
  distSensor(int pin, long on = 0)
  {
    sensorPin = pin;
    pinMode(sensorPin, INPUT);

    interval = on;
    previousMillis = 0;
  }

  float Read()
  {
    /* Check to see if it's time to change the state of the Sensor */
    unsigned long currentMillis = millis();

    if (currentMillis - previousMillis >= interval)
    {
      sensorVal = digitalRead(sensorPin);
      previousMillis = currentMillis;
      return sensorVal ;
    }
  }
```

Figure 26 : Part 1 - Main Coding

```
  }

  float Reset()
  {
    return sensorVal = 0;
  }
};
class LDR
{
  /* Class member variables */
  /* These are initialized at startup*/
  int sensorPin;
  long interval;

  /* These maintain the current state */
  volatile float sensorVal = 0;
  volatile unsigned long previousMillis;

  /* Constructor - create a distSensor and intializes the member
  variables and state*/
  public:
  LDR(float pin, long on = 0)
  {
    sensorPin = pin;
    pinMode(sensorPin, INPUT);

    interval = on;
    previousMillis = 0;
  }

  float Read()
  {
    /* Check to see if it's time to change the state of the Sensor */
    unsigned long currentMillis = millis();

    if (currentMillis - previousMillis >= interval)
    {
      int LDRread = analogRead(sensorPin);
```

```
    {
      int LDRread = analogRead(sensorPin);
      sensorVal = LDRread*0.00488;
      previousMillis = currentMillis;
      return sensorVal ;
    }
  }

  float Reset()
  {
    return sensorVal = 0;
  }
};
//////////////////////////////////////////////////////////////////////////////////////////////////////////
distSensor waterLevel(LEVELpin);
LDR ldrvalue(LDRpin);
distSensor waterLevel1(LEVELpin1);
LDR ldrvalue1(LDRpin1);
//////////////////////////////////////////////////////////////////////////////////////////////////////////
void sendSensor()///////////////////////////////////////////////////////////////////////////////////////blynk sned data
{
  DHT.read11(dht_apin);
  float h = DHT.humidity;
  float t = DHT.temperature; // or dht.readTemperature(true) for Fahrenheit

  if (isnan(h) || isnan(t)) {
    //SwSerial.println("Failed to read from DHT sensor!");
    return;
  }
  // You can send any value at any time.
  // Please don't send more that 10 values per second.
  Blynk.virtualWrite(V5, h);
  Blynk.virtualWrite(V6, t);

  int LDRread = analogRead(LDRpin);
  float Vout = LDRread*0.00488;
  Blynk.virtualWrite(V7, Vout);
  Blynk.virtualWrite(V8, waterLevel.Read());
```

Figure 27 : Part 2 - Main Coding

```
  Blynk.virtualWrite(V7, Vout);
  Blynk.virtualWrite(V8, waterLevel.Read());
  if (waterLevel.Read()== 0)
  {
     terminal.println("water level is Low") ;
  }
    if (waterLevel.Read()== 1)
  {
     terminal.println("water level is OK") ;
  }
  delay (1000);

}


void setup() {
  Serial.begin(115200);
  WiFi.disconnect();
  Serial.println("Connecting...");
  WiFi.begin(ssid,pass);
  Blynk.begin(auth, ssid, pass);
  ph.begin();
  timer.setInterval(1000L, sendSensor);
  terminal.println(F("Blynk v" BLYNK_VERSION ": Device started"));
  terminal.println(F("-------------"));
  terminal.flush();
  while((!(WiFi.status() == WL_CONNECTED))){
    delay(300);
    Serial.print(".");
  }

  Serial.println(WiFi.status());
  Serial.println("Connected!");
  Serial.println("");
}

void loop() {
//  ////////////////////////////////////////////////////////////////////////////////read value sensor
void loop() {
//  ////////////////////////////////////////////////////////////////////////////////read value sensor
    static unsigned long timepoint = millis();
    if(millis()-timepoint>1000U){                  //time interval: 1s
        timepoint = millis();
        //temperature = readTemperature();         // read your temperature sensor to execute temperature compensation
        voltage = analogRead(PH_PIN)/1024.0*5000;  // read the voltage

        phValue = ph.readPH(voltage,temperature)*-1.5+13.8;  // convert voltage to pH with temperature compensation
        Serial.print("temperature:");
        Serial.print(temperature,1);
        Serial.print("^C  pH:");
        Serial.println(phValue,2);
    }
    ph.calibration(voltage,temperature);
  ///////////////////////////////////////////////////////////////////////////////
  DHT.read11(dht_apin);
  float h = DHT.humidity;
  float t = DHT.temperature; // or dht.readTemperature(true) for Fahrenheit
  if (isnan(h) || isnan(t)) {
    //SwSerial.println("Failed to read from DHT sensor!");
    return;
  }
  ///////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////
    DHT.read11(dht_apin1);
  float h1 = DHT.humidity;
  float t1 = DHT.temperature; // or dht.readTemperature(true) for Fahrenheit
  if (isnan(h1) || isnan(t1)) {
    //SwSerial.println("Failed to read from DHT sensor!");
    return;
  }
  ///////////////////////////////////////////////////////////////////////////////////////////
  float celcius = t;
  float humidity = h;
  int waterlevel=waterLevel.Read();
  float ldrval = analogRead(LDRpin);
  float ldr = ldrval*0.0048;
```

Figure 28 : Part 3 – Main Coding

```
  }
  ///////////////////////////////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////////////////////////////
    DHT.read11(dht_apin1);
  float h1 = DHT.humidity;
  float t1 = DHT.temperature; // or dht.readTemperature(true) for Fahrenheit
  if (isnan(h1) || isnan(t1)) {
    //SwSerial.println("Failed to read from DHT sensor!");
    return;
  }
  //////////////////////////////////////////////////////////////////////////////////////////
  float celcius = t;
  float humidity = h;
  int waterlevel=waterLevel.Read();
  float ldrval = analogRead(LDRpin);
  float ldr = ldrval*0.0048;
  float celcius1 = t1;
  float humidity1 = h1;
  int waterlevel1=waterLevel1.Read();
  float ldr1 = ldrvalue1.Read();
  float PHvalue= phValue;
  ///////////////////////////////////////////////////////////////////////////////////////// send data to favoriot
  dataStream(celcius,humidity,waterlevel,ldr, celcius1,humidity1,waterlevel1,ldr1,PHvalue);
  /////////////////////////////////////////////////////////////////////////////////////////
  while (client.available()) {
    char c = client.read();
    Serial.write(c);
  }
  if (!client.connected()) {
    client.stop();
  }
  ///////////////////////////////////////////////////////////////////////////////////////////send data to blynk
  Blynk.run();
  timer.run();
  ///////////////////////////////////////////////////////////////////////////////////////////
  delay(1000);

}

void dataStream(float celcius, float humidity, int waterlevel, float ldr, float celcius1, float humidity1, int waterlevel1, float ldr1, float PHvalue)
{
  // Json Data to send to Platform
  String json = "{\"device_developer_id\":\"deviceDefault@dynajamal\",\"data\":{\"Temperature\":\""+String(celcius)+ "\",\"Humidity\":\""+String(humidity)+ "
\"Waterlevel\":\""+String(waterlevel)+ "\",\"LDR\":\""+String(ldr)+ "\",\"Temperature1\":\""+String(celcius1)+ "\",\"Humidity1\":\""+String(humidity1)+ "\",
\"Waterlevel1\":\""+String(waterlevel1)+ "\",\"LDR1\":\""+String(ldr1)+ "\",\"PHvalue\":\""+String(PHvalue)+ "\"}}";
  // display temperature value
  Serial.println("\n        TEMPERATURE : " + String(celcius)+ " Celcius");
  Serial.println("\n        HUMIDITY : " + String(humidity)+ " %");
  Serial.println("\n        WATERLEVEL : " + String(waterlevel)+ " .");
  Serial.println("\n        LDR : " + String(ldr)+ " V");
  Serial.println("\n        TEMPERATURE1 : " + String(celcius1)+ " Celcius");
  Serial.println("\n        HUMIDITY1 : " + String(humidity1)+ " %");
  Serial.println("\n        WATERLEVEL1 : " + String(waterlevel1)+ " .");
  Serial.println("\n        LDR1 : " + String(ldr1)+ " V");
  Serial.println("\n        PHvalue : " + String(PHvalue)+ " .");

  if (client.connect(serverAdd, 80)) {
    // Make a HTTP request:
    Serial.println("        STATUS : Sending data...");  //Display sending status
    client.println("POST /v1/streams HTTP/1.1");
    client.println("Host: api.favoriot.com");
    client.print(String("apikey: "));
    client.println(apikey);
    client.println("Content-Type: application/json");
    client.println("cache-control: no-cache");
    client.print("Content-Length: ");
    int thisLength = json.length();
    client.println(thisLength);
    client.println("Connection: close");
    client.println();
    client.println(json);
    Serial.println("        STATUS : Data sent!");  //display sent status
    }
}
```

Figure 29 : Part 4 - Main coding

```
#include <DHT.h>
#include <DHT_U.h>

#include <Adafruit_Sensor.h>

int LDR_Pin = A0;
#define output 12

// Example testing sketch for various DHT humidity/temperature sensors
// Written by ladyada, public domain

// REQUIRES the following Arduino libraries:
// - DHT Sensor Library: https://github.com/adafruit/DHT-sensor-library
// - Adafruit Unified Sensor Lib: https://github.com/adafruit/Adafruit_Sensor

#include "DHT.h"

#define DHTPIN A1     // Digital pin connected to the DHT sensor
// Feather HUZZAH ESP8266 note: use pins 3, 4, 5, 12, 13 or 14 --
// Pin 15 can work but DHT must be disconnected during program upload.
#define KIPAS 13
// Uncomment whatever type you're using!
#define DHTTYPE DHT11   // DHT 11
//#define DHTTYPE DHT22   // DHT 22  (AM2302), AM2321
//#define DHTTYPE DHT21   // DHT 21 (AM2301)
#define PINWTR A2
#define PUMP 5

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  Serial.println(("DHTxx test!"));

  dht.begin();
  pinMode (KIPAS, OUTPUT);
  digitalWrite (KIPAS, HIGH);

  pinMode (output,OUTPUT);
pinMode (LDR_Pin,INPUT);
digitalWrite (output, HIGH);

pinMode (PUMP, OUTPUT);
pinMode (PINWTR, INPUT);
digitalWrite (PUMP, HIGH);
}

void loop() {
  // Wait a few seconds between measurements.
  delay(500);

  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
  float h = dht.readHumidity();
  // Read temperature as Celsius (the default)
  float t = dht.readTemperature();
  // Read temperature as Fahrenheit (isFahrenheit = true)
  float f = dht.readTemperature(true);

  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }
```

Figure 30 : Part 1 - Coding for Smart System

```
if (t >= 24.8)
 {
 digitalWrite (KIPAS, LOW);
 }
 if (t <= 20)
 {
  digitalWrite (KIPAS, HIGH);
 }
delay (500);

  // Compute heat index in Fahrenheit (the default)
  float hif = dht.computeHeatIndex(f, h);
  // Compute heat index in Celsius (isFahreheit = false)
  float hic = dht.computeHeatIndex(t, h, false);

  Serial.print(F("Humidity: "));
  Serial.print(h);
  Serial.print(F("%  Temperature: "));
  Serial.print(t);
  Serial.print(F("°C "));
  Serial.print(f);
  Serial.print(F("°F  Heat index: "));
  Serial.print(hic);
  Serial.print(F("°C "));
  Serial.print(hif);
  Serial.println(F("°F"));


int LDRReading = analogRead (LDR_Pin);
float Vout = LDRReading*0.00488;
Serial.print ("Vout");
Serial.print ("\t");
Serial.println (Vout);


if (Vout >= 1.99)
 {
 digitalWrite (output, LOW);
 }
 if (Vout <= 1.98)
 {
  digitalWrite (output, HIGH);
 }
delay (500);

if (digitalRead(PINWTR)==0)
{
  digitalWrite (PUMP,LOW );
}

if (digitalRead(PINWTR)==1)
{
  digitalWrite (PUMP,HIGH );
}
Serial.println(digitalRead (PINWTR));
}
```

Figure 31 : Part 2 - Coding for Smart System