DEMAND-OPERATED PLANT WATERING SYSTEM

YAP HENG YI

BACHELOR OF MECHATRONICS ENGINEERING WITH HONOURS
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2019

DEMAND-OPERATED PLANT WATERING SYSTEM

YAP HENG YI

A report submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Mechatronics Engineering with Honours

Faculty of Electrical Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2019

DECLARATION

I declare that this thesis entitled “DEMAND-OPERATED PLANT WATERING SYSTEM” is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature :
Name :
Date :

APPROVAL

I hereby declare that I have checked this report entitled “DEMAND-OPERATED PLANT WATERING SYSTEM” and in my opinion, this thesis it complies the partial fulfillment for awarding the award of the degree of Bachelor of Mechatronics Engineering with Honours

Signature : ....

Supervisor Name : ....

Date : ....

DEDICATIONS

To my beloved mother and father

ACKNOWLEDGEMENTS

While conducting my final year project, I was in contact with many people and they have directly and indirectly contributed towards my understanding and thought.

First of all, I would like to express my gratitude to my supervisor, Mr. Syed Mohamad Shazali Bin Syed Abdul Hamid for the encouragement, guidance and the comments he gave. Besides, I am very thankful for my panels Dr. Saifulza Bin Alwi @ Suhaimi and Profesor Madya Dr. Rozaimi Bin Ghazali for their guidance, advice, and motivation. Without their support and advice, this project would not have been the same as presented here.

I am also indebted to Universiti Teknikal Malaysia Melaka (UTeM) for the funding. I would like to extend my gratitude to my fellow friends who have lent a helping hand on various occasions. Their views and tips are useful indeed. Lastly, I am grateful to all my family members as well for their spiritual support.

ABSTRACT

In this new era, agriculture has improved a lot with advanced technology. As we all know, water is crucial as it helps plants to grow. Yet, freshwater resources are depleting and manual watering is exhausting, time-consuming and not efficient. Having to water the plants every day at the desired periods is a challenge. Besides, some timer-based watering systems have no means of controlling suitable soil moisture. Plus, the plants will constantly be watered once soil moisture drops below the threshold for some simple automated plant watering system and irrigation scheduling is usually available for the large-scale farm with a more complex system. Therefore, a demand-operated plant watering system using Arduino with time setting function and soil moisture feedback for the indoor small-scale vegetable garden has been designed and developed to solve the problems. The performance of the system has been accessed and analysed also in terms of time and soil moisture control. To design a demand-operated plant watering system, a microcontroller Arduino has been programmed to control all the inputs and outputs. An LCD display was used to show the date, time, soil moisture level and water level while some pushbuttons were used to set the watering time. Soil moisture sensor and water level sensor were used as the feedback to the system. A water pump was used to supply water to the plant from the water container and a buzzer was used to alert the user when the water level is low. From experiment 1, 70 % of soil moisture was determined as the threshold level to water the plants. From experiment 2, it was proven that the plant watering system able to water the plants at the set time by providing soil moisture slightly higher than the threshold.

ABSTRAK

Dalam era baru ini, pertanian telah bertambah baik dengan teknologi canggih. Seperti yang kita ketahui, air adalah penting kerana ia membantu pertumbuhan tanaman. Namun, sumber air tawar kini tengah berkurang dan penyiraman secara manual amat meletihi, buang masa dan tidak efisien. Menyiram pokok bunga atauan tanaman setiap hari pada masa yang dikehendaki adalah satu cabaran. Selain itu, beberapa sistem penyiram berasaskan pemasa tidak mempunyai cara untuk mengawal kelembapan tanah yang sesuai. Di samping itu, beberapa sistem penyiraman tanaman automatik akan sentiasa menyiram tumbuh-tumbuhan apabila kelembapan tanah jatuh ke tahap tertentu dan penjadualan penyiraman biasanya dipakai oleh ladang berskala besar dengan sistem yang lebih kompleks. Oleh itu, satu sistem penyiraman tumbuhan berdasarkan Arduino dengan fungsi penetapan masa dan maklum balas kelembapan tanah untuk kebun sayur-sayuran dalaman berskala kecil telah direka bentuk untuk menyelesaikan masalah-masalah tersebut. Prestasi sistem juga telah diakses dan dianalisis dari segi kawalan masa dan kelembapan tanah. Mikrokontroler Arduino telah diprogram untuk mengawal semua input dan output system penyiraman tumbuhan tersebut. LCD telah digunakan untuk menunjukkan tarikh, masa, tahap kelembapan tanah dan paras tangki air sementara beberapa butang telah digunakan untuk menetapkan masa untuk menyiram. Sensor kelembapan tanah dan sensor paras air telah digunakan sebagai maklum balas kepada sistem. Pam air telah digunakan untuk membekalkan air kepada tumbuh-tumbuhan dari bekas air dan buzzer telah digunakan untuk mengingatkan pengguna apabila paras air rendah. Dari eksperimen 1, 70% kelembapan tanah telah ditentukan sebagai paras ambang untuk menyiram tumbuhan. Dari eksperimen 2, sistem penyiraman tumbuhan ini dapat menyiram tumbuhan pada masa yang ditetapkan dengan kelembapan tanah yang lebih tinggi sedikit daripada paras ambang.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

ADC	-	Analog-to-digital converter
RTC	-	Real-time clock
I/O	-	Input/output
PWM	-	Pulse Width Modulation
LDR	-	Light Dependent Resistor
UV	-	Ultraviolet
USB	-	Universal Serial Bus
GUI	-	Graphical user interface
LCD	-	Liquid crystal display
IDE	-	Integrated development environment
CAD	-	Computer-aided design
NO	-	Normally open
NC	-	Normally closed
LED	-	Light-emitting diode
PID	-	Proportional-integral-derivative
EEPROM	-	Electrically Erasable Programmable Read-Only Memory
SDA	-	Serial data
SCL	-	Serial clock

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Motivation

In this day and age, the development of agriculture has improved tremendously. With advanced biotechnology, a variety of plants have been cultivated. One of the important elements for the plants to grow is water. Water seems abundant on our planet; however, less than 1% of the world's liquid freshwater is available for human use and about 70% of it is used for irrigation [1]. Figure 1.1 shows freshwater is used for irrigation of vegetable farm.

Figure 1.1: Vegetable farm

Apart from that, the manual irrigation method is exhausting, time-consuming and not efficient. Moreover, even the maintenance of a small garden can be a tedious job for most people [2]. Often, people do not have much time to perform watering since it is a daily routine which requires quite some time. Figure 1.2 shows the plants in the garden are watered manually by using a watering can.

Figure 1.2: Backyard garden

For indoor plant hobbyist, they might be lazy to water their potted plants sometimes or do not have time to do so due to a tight schedule. Plus, people tend to forget to water their plant and unable to do watering while they are away for vacation [3]. In fact, manual watering can lead to wastage as too much water might be poured for some area. In the age of advanced electronic, human life should be made easier and more comfortable with the aid of an automated system. Figure 1.3 shows some vegetables are grown indoor by using planter box or pots.

Figure 1.3: Indoor vegetable garden

1.2 Research background

Irrigation is an artificial application to water crop instead of depending on rainfall solely by different means such as bringing in water from pipes, canals, sprinklers and so on. Irrigation promotes crop growth and quality in areas which have uncertain rainfall and uniformity is the key to maximize irrigation effort. Irrigation was practiced in many parts of the world during ancient civilization.

The very first method may be farmers carrying buckets of water from rivers or wells to water their crops followed by the building of irrigation canals, dams, dikes, and water storage facilities [4]. Moreover, aqueducts which carry snowmelt from the Alps was constructed by Ancient Rome for washing, drinking and irrigation purpose [4].

Nowadays, tanks, wells, and reservoirs such as aquifers, basins which collect snowmelt, lakes, and basin formed by the dam are used to provide water for the crops. The water stored in the reservoir is delivered to the field by canals or pipelines with the aid of gravity or water pump [4].

Several methods are implemented for irrigation such as flooding the whole field, channeling water between rows of plants, spraying water via large sprinklers, or letting water drop onto plants through holes in pipes. In addition, irrigation scheduling is a crucial process where the timing and amount of water to be applied to the crops are decided by the irrigator.

However, the crop water requirement for different growth stage and climate is the main challenge as it might lead to over or under watering. To avoid that, methods include plant observation, feel and appearance of the soil, using soil moisture monitoring devices, and estimating available water from weather data can be used.

Soil moisture content is the quantity of water it contains [5]. In fact, soil moisture content plays a dominant role in ground recharge, agriculture and soil chemistry. For example, the optimum soil moisture content is beneficial for plant growth. Furthermore, soil water is an important medium in supplying nutrients to the plants as soil water dissolves salts and minerals and hence the solution can be absorbed by the roots of the plants. Besides, other advantages of soil moisture include soil temperature regulation, essential for photosynthesis, helps in chemical and biological activities of soil and so forth.

1.3 Problem statement

Generally, most people use a timer-controlled plant watering system which has no means of controlling the suitable soil moisture required by the plants. Overwater will occur frequently under such condition where too much water may rot the root while too less water may, on the other hand, restrict the transport of nutrients to the plant. Commonly, plants are required to be watered twice a day. Usually, the desired periods are in the morning and in the evening. Some current plant watering system is fully automated where the plants will be watered constantly when the soil moisture drops below the threshold and hence the user has no control over the period to water the plant. Besides, irrigation scheduling is available for some large-scale farm with a more complex system. It is hypothesized that a demand-operated plant watering system with timer setting and soil condition feedback for the indoor small-scale vegetable garden is able to cater the water requirements of the plants, allow the user to decide desired periods of watering as well as save water and time.

1.4 Objectives

The purposes of this project are:

- To design and develop a demand-operated plant watering system using Arduino with time setting function and soil moisture feedback for the indoor small-scale vegetable garden.

- To assess and analyse the performance of the demand-operated plant watering system in terms of time and soil moisture control.

1.5 Scope of work

The scope of work in this project focuses on the design and evaluation of a suitable plant watering system for the indoor vegetable garden. The coding and simulation of the system are done before building the hardware to know workability of the system. The functionality of the system is tested for the chili plant placed at an indoor vegetable garden. 2 experiments are designed to test the plant watering system as well.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

It is a well-known fact that agriculture supplies food source and human needs food to survive. Hence, agriculture has been developing with the advancement of technology nowadays. Watering is the key element when it comes to complementing water for plants when rainfall is insufficient. Besides, different plants have different water needs. For example, a cactus will require less water than a rose bush [6]. Plus, plants that are drought tolerant will need to be watered differently than plants requiring a lot of water [7]. A systematic demand-operated plant watering system is absent. To look into the matter, it is crucial to study recent research journals to obtain a better idea. The literature review is anticipated to provide greater insight and discover the potential of integrating various methods from different plant watering system.

Based on the journals, the three problems identified are wastage of scarce water resources, unequal distribution of water during irrigation which leads to damages for the plants, as well as a challenge to keep the plants alive and healthy as it is a time-consuming and tedious job to water the plants. Predictions are that by the year 2025, one-quarter of the world's population will face severe water shortage [2]. Thus, it is imperative to utilize current technology to minimize water loss through irrigation. Sometimes watering done was not compatible with the water necessity of plants [8]. For example, plants tend to die from excessive irrigation or the other way around. Watering for a small garden is already a tedious task. Farmers do not have much time to carry out irrigation on a daily basis since they have some other tasks to be done as well. For instance, pest control and harvest can consume quite some time.

Thus, the subsections as followed are meant to study and review the advantages and disadvantages of different modules.

2.2 Review of exteroceptive sensors

The sensor is an important element in designing a system as it acts as the input of the system with constant feedback. Sensors inform the system about the status of a certain parameter so that certain actions can be done upon it.

Soil moisture sensors measure the volumetric water content in soil [9]. A soil moisture probe is made up of multiple soil moisture sensor [10]. Generally, the technologies used in soil moisture sensors can be frequency domain sensors such as capacitance sensor, neutron moisture gauges which make use of the moderator properties of water for neutrons or electrical resistance of the soil. The volumetric water content in soil is measured by soil moisture sensor indirectly by using the property of soil including dielectric constant, electrical resistance or interaction with neutrons.

For the capacitance sensor, the soil is used as a dielectric which relies on the soil moisture content. However, variations in soil salinity, temperature, water, and bulk density can have an impact on the measurement. In spite of that, good accuracy and precision can still be achieved under certain condition and some sensors have been widely used in the scientific field as well.

For a resistive type sensor, less corrosive galvanized irons are used for the probes. The two metal conductors which spaced apart serve to measure the soil electrical conductivity. The current passes through the probes where the resistance varies in accordance with soil moisture content. Yet, the dissolved salts can alter the water conductivity and thus affect the measurements. Moreover, there is another type of sensor which measure the water potential and it is called soil water potential sensor which includes tensiometer and gypsum blocks. Figure 2.1 shows the resistive soil moisture sensor.

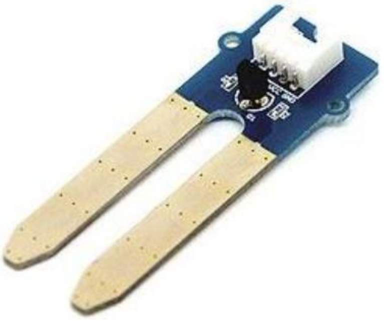


Figure 2.1: Soil moisture sensor [11]

Multiple sensors can be used to control irrigation of an individual zone or vice versa. For the case of one sensor controlling multiple zones, those zones are usually the driest where water is most needed. There are some common rules for the burial of the soil moisture sensors where it is better to bury the sensor near the root zone of the plants as roots absorb water.

For instance, the sensors for turfgrass are usually buried about three inches deep. Besides, sensors need to be in good contact with the soil after burial where there should be no air gaps surrounding the sensor and soil should be packed firmly but not excessively around the sensor [9]. Furthermore, it is advisable to bury the sensors in the zone that requires water the most for the case of one sensor controlling the whole irrigation system to make sure all zones acquire sufficient water. Normally, the zone is the area with the most sun exposure. To avoid over compaction of the soil around the sensor, it is not recommended to bury sensor in high traffic areas also.

Anil et al. set two thresholds for the soil moisture sensor which are 85 for lower threshold and 160 for the upper threshold while Lekjaroen et al. set the threshold of 20% and 80%. The threshold value is based on the ADC value of the sensor ranged from 0 to 255. The water pump turns on when the soil moisture goes below the lower threshold and turns off when the upper threshold has reached. In this way, a minimum soil moisture level can be maintained.

Soil temperature sensors obtain measurement through digital thermometer DS 1822 where analog temperature value is converted to 12-bit digital word and stored in 2-B temperature registers. Irrigation is needed when the temperature threshold is exceeded. For example, Gutiérrez et al. set a maximum soil temperature threshold of

30 °C to activate the irrigation pump for the desired period. Soil temperature is correlated with the soil moisture content but measuring the soil temperature does not fully represent water is needed by the plants.

On the other hand, humidity and temperature sensors are used to check the surrounding humidity and temperature. The magnitude of humidity and temperature can influence the health of the plants. Too high or too low temperature and humidity reading can be bad for certain plants [12]. User can always monitor the status of the plants through these sensors. Same goes to humidity and temperature sensor, these parameters affect the soil moisture level indirectly.

Some other sensors such as a light detection sensor or LDR sensor detects the presence of sunlight. UV diode complements such sensor to provide artificial sunlight as an energy source for plants when sunlight is absent or weak. Sandberg et al. installed a light detection sensor to detect sunlight and UV diode to provide artificial sunlight for the plant watering system. The rain sensor is another type of sensor which detects the presence of rainfall to prevent redundant irrigation. Regarding the usage of the rain sensor, it is more applicable for outdoor plantations as they are exposed to rain.

From the above review, functions of various sensors have been understood. It is found that soil moisture sensor has a direct connection with the watering status of the plants. In addition, measurements of both capacitive and resistive soil moisture sensor are affected by the soil conditions. On the contrary, other sensors such as humidity sensor and temperature sensor have an indirect link to the plant watering system as they serve to complement the plant watering system for better accuracy. For the LDR sensor and rain sensor, it is more suitable to apply these sensors for outdoor plantations.

2.3 Review of electronic control mechanism

In designing a system, a microcontroller acts as the brain of the system. The microcontroller decides the action of output based on the feedback from the input as programmed. All sensors and actuators will not work without the instructions from the microcontroller. Hence, it plays a vital role in controlling the whole system.

The Arduino UNO is a microcontroller board based on the ATmega328 [11]. There are 14 digital input/output pins which 6 can be used as PWM outputs, 6 analog inputs, a USB port, a 16 MHz ceramic resonator, a reset button, an ICSP header, and a power jack. ATmega is low power, high-performance RISC CPU, innovative Pico Power technology, and a rich feature set, so the AVR architecture ensures fast code execution combined with the lowest possible power consumption [3]. The Arduino UNO can be powered by connecting it with a computer via USB cable, AC-to-DC adapter or battery. The power jack can be connected with the adapter using a 2.1mm center-positive plug whereas Gnd and Vin pin headers of power connector can be connected with leads from the battery. The Arduino UNO can operate with external supply from 6 to 20 V. Yet, the board may be unstable when the supply is less than 7 V. Plus, the voltage regulator may overheat and subsequently damage the board if the supply is more than 12 V. Therefore, 7 to 12 V range is more commendable. The Arduino UNO is programmable through Arduino IDE where new code can be uploaded to it without the use of an external hardware programmer through the ATmega328 preburned with a boot loader. Devika et al. used Arduino UNO to control motor, water pump, and moisture sensors. The soil moisture level is measured by the moisture sensors and the signal is sent to the Arduino if watering is needed. When the desired moisture level is achieved, the motor and water pump stop supplying water to the plants. This is a simple and achievable method, but the user does not involve in the control of the water pump and the desired period of the watering cannot be decided as well. Figure 2.2 shows plant watering system consists of Arduino, moisture sensor, water pump, H-bridge and a rotating platform made from the servo motor.

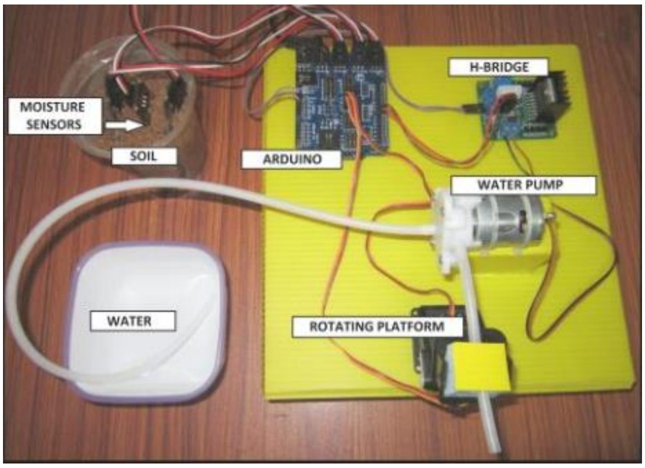


Figure 2.2: Setup of Arduino with other components [11]

PIC16F628A is another type of microcontroller which has 2K words of program memory and 224 bytes of data memory. It also has 16 input/output pins that are user configurable with pin-to-pin basis. PIC16F628A operates between 2 to 5.5 V which is lower than Arduino. To program the microcontroller, mikroC PRO software is used to generate hex file, error report and COD file. The hex file is then needed to be loaded through WINPIC800 software to the microcontroller chip. ISIS-Professional is another software which is used to do simulation for the written code. This is a little inconvenient compared with Arduino IDE which can upload the code directly to the Arduino board. Mobasher Ahmed et al. used 2 microcontrollers to control gate valve by monitoring sensor and give information to the third microcontroller. The microcontroller controls two pumps where the pump 1 is used to supply water to the land whereas the pump 2 is used to remove excess water from the land. The process of supplying and removing water is assisted by the gate valve, water sensor, and rain detector. The gate valve will open when the land is in a dry state to allow water to enter the respective land by using pump 1. Extra water will reach a certain level when rain is present and the signal will be sent to the microcontroller. At the moment, the gate valve is opened to allow excess water to be thrown up from land by pump 2. In addition, a password is required to enter the controlled room. The microcontroller will send a message through a wireless message system if the wrong password is detected. Pump condition and rain condition will also be sent by the microcontroller through message also. This system is strong in term of water control, communication and security. Yet, implementation of such design is quite difficult and security system such as password is not necessary for an indoor plant watering system. Despite, the water sensor used in the system is applicable in terms of measuring the water level of a water tank for an indoor plant.

Single-Chip PIC24FJ64GB004 is a 16-bit microcontroller with the most pins which are 44-pins and nanoWatt XLP technology that operates in a range of 2.0 to 3.6 V which is the lowest among all at 8 MHz with internal oscillator. There are 25 digital input/output ports, 10-bit analog-to-digital converters (ADC), two serial peripheral interface modules, two I2C, two UART, 5 16-bit timers, 64 KB of program memory, 8 KB of SRAM, and hardware real-time clock/calendar (RTCC). Due to the low-power operating current of 175 µA at 2.5 V, 8 MHz and 0.5 µA for standby current in sleep mode including the RTCC, the microcontroller is suitable for remote application. The microcontroller can be programmed in C compiler 4.12. Gutiérrez et al. programmed the microcontroller to monitor the soil-moisture probe through an analog-to-digital port and the soil temperature probe through another digital port. The microcontroller also interfaced with XBee radio modem to allow information from soil moisture sensor and soil temperature sensor to be transferred to the microcontroller. This system has the advantage of transferring data wirelessly within sensors and microcontroller which is more convenient. Yet, it is more suitable for a large farm with many sensors where wiring can be troublesome.

PIC16F877A is an 8-bit microcontroller where the operation is limited to 8 bits only with an operating voltage of 5V. There are 3 sizes which are short, standard and extended and they are all called PIC16F877A because they share some particular features and can be programmed under the same assembly language. Some of the features include 14 KB of program memory, 368 bytes of data memory, 8-bit data bus, 16-bit address bus, 16-bit timer, 4 register banks, 8-bit ports and so forth. The microcontroller can be programmed by using KEIL software. Gupta et al. programmed the microcontroller in a way that the soil moisture level can be detected and water will be supplied if needed. The microcontroller will always irrigate the field which first sends the signal in the case of more than one signal is sensed. This method is similar to the Arduino which controls soil moisture sensor and water pump.

The Teensy 2.0 is a 29-pin microcontroller with an AVR core [6]. Some features of Teensy 2.0 include 12 ADC pins, 25 digital I/O and 7 Pulsed-Width Modulation pins (PWM). Commonly, the Teensy 2.0 is powered by the 5 V supplied by a computer through a USB cable which is the same operating voltage as PIC16F877A. The code for Teensy 2.0 is written in the C language and programmed via the TeensyLoader application. Al-Bahadly et al. utilized the microcontroller to measure the soil moisture content in order to decide whether the plants require water. The microcontroller was connected to a Dual Output Tap Timer which consisted of two motorized water valves for watering purpose. This watering concept also resembles with working principles of Arduino and PIC16F877A previously except that water valve is used instead of the water pump.

Based on the review above, microcontrollers serve more or less the same features and they need to be programmed by certain programming language. However, it can be compared in term of operating voltage, I/O pins, and program method. PIC24FJ64GB004 has the highest number of I/O pins, but Arduino UNO has 14 digital I/O pins which are quite sufficient already. Although Arduino requires a higher operating voltage of 7 - 12 V, the method of programming is easier compared to PIC16F628A. This is because PIC16F628A needs to be programmed, simulated and loaded by 3 different software while Arduino only needs one. PIC16F877A and Teensy 2.0 are programmed by KEIL software and Teensy Loader respectively which are not as familiar as Arduino IDE. Apart from that, the microcontrollers can be programmed in different ways and interfaced with a variety of components. One of the simplest designs of a plant watering system is by linking a microcontroller with soil moisture sensor and water pump. Soil temperature sensor and rain sensor are optional as the soil water content is far more crucial than the other parameters but water level sensor is applicable in term of detecting the water level of the water tank for plant watering. Wireless communication between the microcontroller and other devices is an advanced feature, but the implementation of such a system is more complicated and the cost of the building may be higher as well. Figure 2.3 shows the flow chart of a simple plant watering system with a threshold ADC value of 300 ranging from 0 to 1023 for a soil moisture sensor.

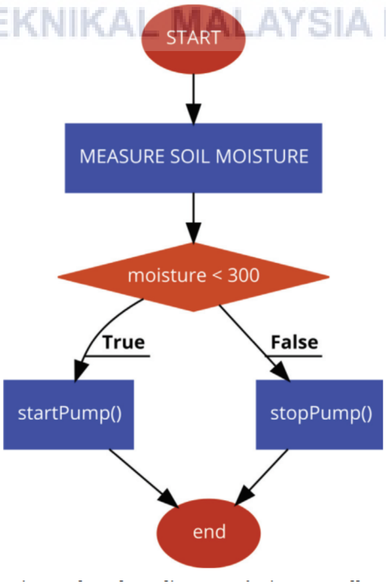


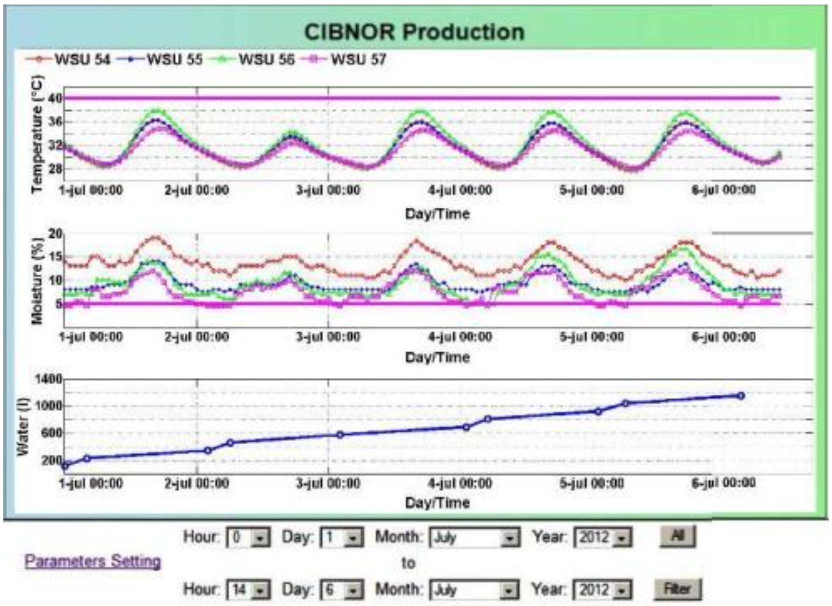
Figure 2.3: Flow chart of simple plant watering system [13]

2.4 Review of human-machine interface

A human machine interface links all the sensors and actuators by displaying information regarding the components where the user can interact with them. It allows the user to input or to extract desired information. So, interaction and communication between the user and the device are enhanced through the human-machine interface.

The web application is one of the graphical user interfaces where data from each sensor can be visualized graphically by the user. For example, real-time monitoring and programming of irrigation are available based on soil moisture and temperature data. The web application can be coded in C language by using Microsoft Visual Studio 2010 and the database can be implemented through SQL Server 2005. The database permits the storage of information where the user can read all the collected data later on via web interface. Furthermore, the user is allowed to configure the settings such as the Wi-Fi name, threshold value of soil moisture and so on. The web interface is powerful as more features are available.

Gutiérrez et al. developed a graphical user interface for real-time monitoring programming of irrigation based on soil moisture and temperature data. It can display the soil moisture and temperature graph as well as total water consumption. Plus, direct programming of scheduled irrigation and adjusting desired trigger values of the sensors are available also. On the other hand, Sandberg et al. created the web interface of iPlant which able to display all the collected data such as the sensor values. Similar to Gutiérrez's system, threshold values including minimum humidity, water and temperature can be changed according to one's need. Figure 2.4 shows the web application of the automated irrigation system while Figure 2.5 exhibits the web interface of iPlant. Both web interfaces display data such as time, temperature, water level, moisture level and so forth.

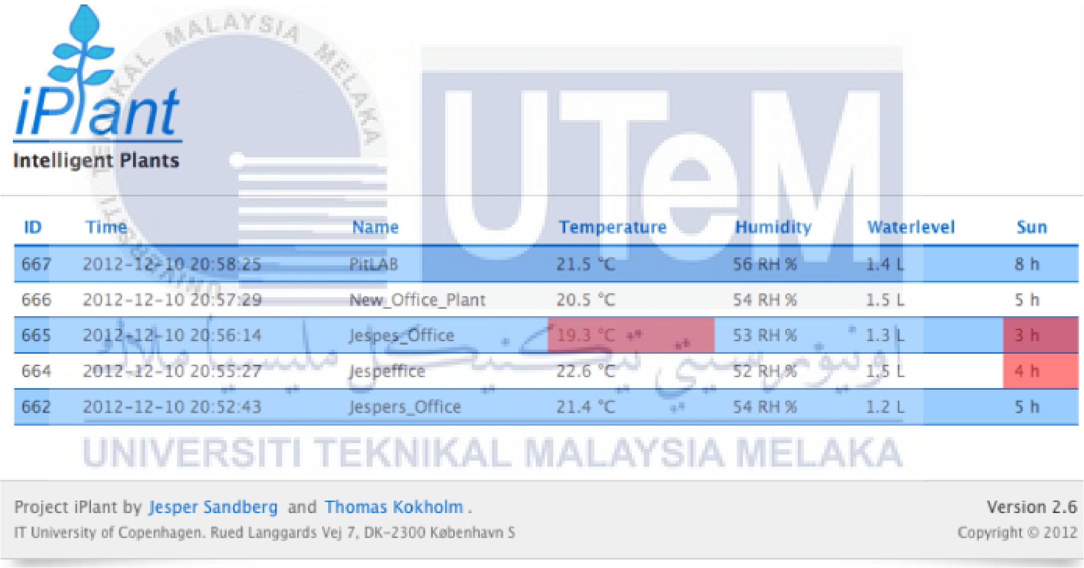


Hour: 0 Day: 1 Month: July Year: 2012

to

Hour: 14 Day: 6 Month: July Year: 2012

Figure 2.4: Web application of the automated irrigation system [14]

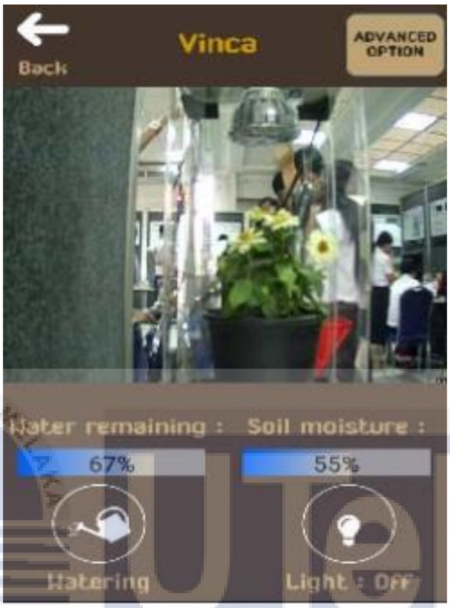


ID	Time	Name	Temperature	Humidity	Waterlevel	Sun
667	2012-12-10 20:58:25	PitLAB	21.5 °C	56 RH %	1.4 L	8 h
666	2012-12-10 20:57:29	New_Office_Plant	20.5 °C	54 RH %	1.5 L	5 h
665	2012-12-10 20:56:14	Jespes_Office	19.3 °C	53 RH %	1.3 L	3 h
664	2012-12-10 20:55:27	Jespeffice	22.6 °C	52 RH %	1.5 L	4 h
662	2012-12-10 20:52:43	Jespers_Office	21.4 °C	54 RH %	1.2 L	5 h

Project iPlant by Jesper Sandberg and Thomas Kokholm. Version 2.6
IT University of Copenhagen. Rued Langgaards Vej 7, DK-2300 København S Copyright © 2012

Figure 2.5: Web interface of iPlant [12]

Android is an open source mobile operating system which is widely used in smartphones nowadays [15]. Tools like Android Studio, Eclipse and Visual Studio can be used to design and develop android application. Android Studio is a tool for developing an Android application, its increase performance in design GUI and can preview developing application on a different platform [16]. An interface between the analog sensor, database, and Android application can be provided by Arduino. The mobile application is easy to download and user-friendly where the usability can further consolidate with internet connectivity. For instance, Lekjaroen et. al implemented android application interface for the elderly which is simple to use. The application has 4 pages which include the main page, tutorial page, plant selection menu, and monitoring page. Environmental values such as temperature, soil moisture, and water level are shown to the user and alert message will be shown when the values of sensors are out of range. Figure 2.6 shows the monitoring page of android application which displays the water level and soil moisture level of the plant watering system.



Water remaining : 67%

Soil moisture : 55%

Figure 2.6: Android application interface [16]

A liquid crystal display (LCD) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector [3]. A column of liquid crystal molecules suspended between two transparent electrodes builds up a pixel with two polarizing filters that are perpendicular to one another. The polarization of light entering one filter is twisted by the liquid crystal, so it can pass through the other. For the application, an LCD module can be applied in a system for monitoring purposes which include readings from sensors or the current status of valves. Moreover, keypad or toggle buttons can be used to complement an LCD to create a user interface for the system. The keypad or buttons allow the input to the system via different instances. Although this interface has limited features, the cost of implementation is the lowest among the rest and the basic feature is already enough to support a simple plant watering system. For example, Ojha et al. used an LCD to display the temperature and soil moisture while Gupta et al. used an LCD to provide the user with the information of time and date. Figure 2.7 exhibits an LCD which displays the current temperature and soil moisture.

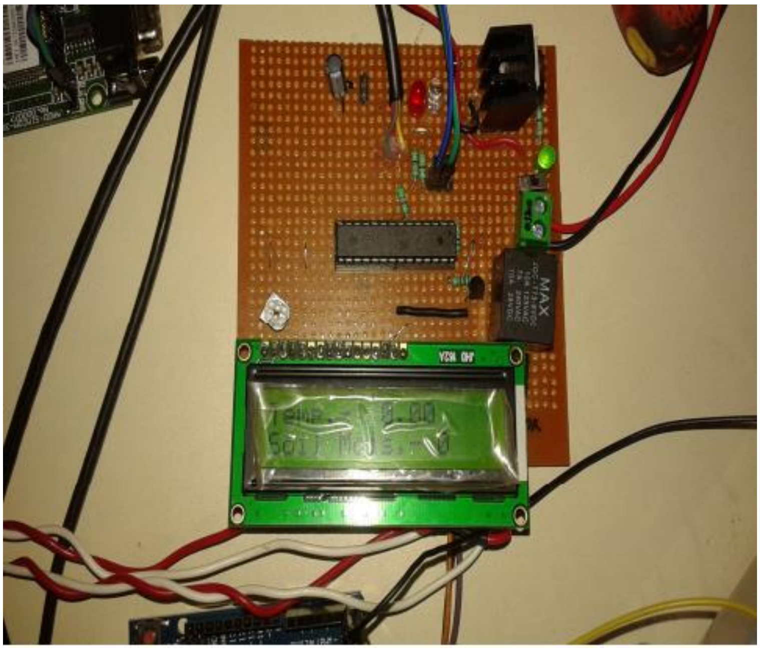


Figure 2.7: LCD display interface [9]

At the end of the review, it is seen that an LCD complemented with buttons is a user-friendly interface due to its simplicity. Android application is a convenient interface which can control the system wirelessly through Wi-Fi or Bluetooth but not necessary for a simple plant watering system. The android application might be a good interface as there are a lot of Android users nowadays. Yet, there are many IOS users as well and hence the implementation of applications for both system users can incur quite some time. In fact, the web interface is the best among all due to its comprehensive functionality. However, the creation of a web interface can be tricky and complex which requires higher programming skill. Plus, it is more suitable for a large farm with more sensors which requires supervision on the status of the plants and data logging. Hence, LCD interfaced with push buttons is a simple, low cost and low power consumption module. Plus, functions such as displaying soil moisture level and some simple settings are achievable through LCD interfaced with push buttons.

2.5 Review of water supply mechanism

In designing a system, actuators play a vital role as an instruction from the controller needs to be executed by the actuator. For the case of a plant watering system, actuators including the water pump, water valve, and servo motor are going to be reviewed.

The water pump supplies water artificially for a specific task and such a process is called pumping. It can be electronically controlled by interfacing it to a microcontroller [9]. ON/OFF signals serve to trigger the water pump. A relay is needed for the operation of the water pump as relay interfaces between electronic circuit and mechanical circuit. The water pump is connected to the relay module and it only works when the relay module gets a command from the microcontroller [13]. In other words, relay switches high powers such as the main supply with a weak power source. The pumping of water is a basic and practical technique, far more practical than scooping it up with one's hands or lifting it in a hand-held bucket. Đuzić et al. connected the water pump to the relay module and the water pump only works when the signal is received by the relay module from the microcontroller. Figure 2.8 shows a water pump for plant watering purpose.

Figure 2.8: Water pump [11]

Water valve controls the fluid flow by opening or closing it. For a solenoid valve, the fluid either being cut off or allowed when electrically energized or de-energized. To make the valves normally open or normally closed, a spring is needed. A magnetic field builds up when the valve is energized to pull a plunger or pivoted armature against the spring. The plunger or pivoted armature retracts to its original position when it is de-energized. The water valve requires control through a relay as well. The water valve can be used to control the flow of water through the outlet. For instance, the water supply from the hose will be cut off when the water valve is closed. The function of both the water valve and water pump may seem to be alike, but the water valve only allows free flow of water while the water pump draws out the water with external force driven by a motor. Al-Bahadly et al. controlled the flow of water through two outlets with the help of two motor-controlled valves. Figure 2.9 shows the structure of two motor-controlled valves with one input opening and two output openings.

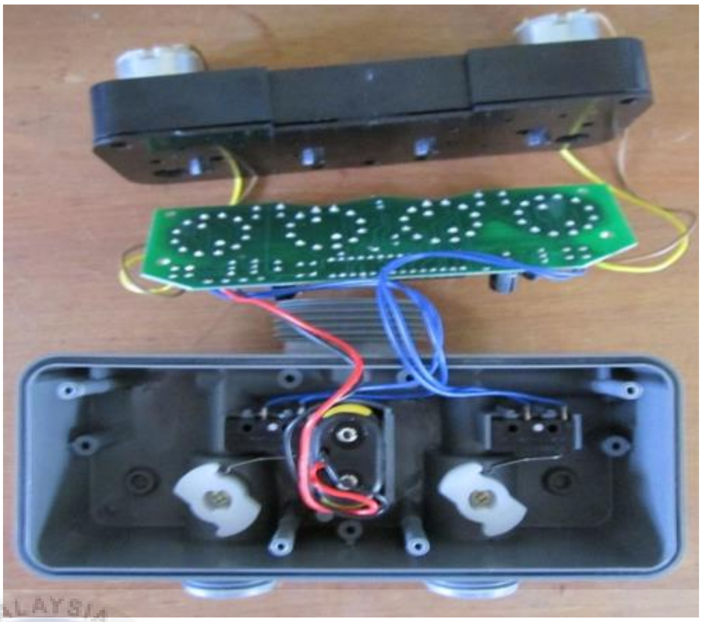


Figure 2.9: Water valve [6]

Servo motor controls angular position through integration with a sensor for position feedback. It is just an ordinary motor which uses servomechanism to achieve closed-loop control. A servomechanism is formed by a specific type of motor with rotary encoder. Position and speed feedback is usually provided by the encoder and more precise control of position can be fulfilled by the use of a PID controller. Devika et al. designed a plant watering system which makes use of servo motor as a rotating platform to direct water towards the plants.

According to the review, the functions of different actuators such as water pump, water valve, and servo motor are known. Both water pump and water valve are feasible to deliver water to the plants as they serve the same purpose only with different working principle. In addition, the servo motor is only used to assist the watering process which is less important. Yet, the water pump has an advantage of pumping dechlorinated water from rain barrel over water valve which controls water flows from a chlorinated water source such as tap water. Besides, the water pump which controlled by a motor can have different output depends on the power rating. For example, the required flow rate can be controlled by the water pump but not the water valve.

2.6 Review of assessment methods

Assessment or evaluation is a must when it comes to designing a system. There are numerous ways to conduct an assessment and a good assessment can reflect and impact a system in a good way. Hence, a few assessments are going to be reviewed in this section.

Sandberg et al. combined qualitative, quantitative and longitudinal research for human interaction test when evaluating the plant watering system. For qualitative research, interviews will be carried out while questionnaires are distributed to obtain data from many people for quantitative research. Furthermore, the longitudinal research is conducted for the prototype as it consists of plants which require a period of time to have changes.

A stress test is carried out as well to test the survivability of the plant with the prototype. In the stress test, two identical potted plants are placed in a fixed indoor environment to eliminate wind, fluctuating temperature and irregular sunlight given the same amount of water and mold in the first place. Both plants are left for 3 weeks where one of them is equipped with the iPlant prototype while the other is just a regular potted plant. By applying this test, the functionality of the system can be known because the plant will only remain healthy if sufficient water is supplied. Figure 2.10 displays the result of the stress test between regular plant and iPlant.

Figure 2.10: Comparison between regular plant (left) and iPlant (right) [12]

For human interaction test, it is to identify how the prototype affects the relationship between human and their household plants. A longitudinal test is conducted with a guy living in a small apartment. The guy takes care of the plant as if it was a regular potted plant for two weeks. Besides that, questionnaires are distributed to collect data and interview is carried out after the longitudinal test.

Anil et al. installed the HARITHA irrigation system in one of the gardens. Data from the soil moisture sensor was collected at an interval of one hour on a normal sunny day. The minimum threshold of the soil moisture sensor was set at 85 while the maximum threshold was set at 160 to ensure the plants obtain sufficient water. This is a good way to get to know whether the plants obtain enough water.

Kumar et al. tested the water retention of soil by using a soil moisture sensor. To carry out the test, three sets of different plant and soil combinations were used. The three sets of plant combinations were exposed to three different water level respectively which were low water content, normal water content, and excess water content. The same type of soil moisture sensors was inserted into the soil to check the performance and compatibility of the sensors under different conditions. By using the technique above, the functionality of a soil moisture sensor can be known where different soil moisture level should be determined by the sensor. Thus, optimum soil moisture level can be learned also by testing the sensor under different water condition. Figure 2.11 illustrates the three soil moisture conditions which are without water, average water and excess water.

Figure 2.11: Comparison of 3 different soil moisture condition [17]

From the review in this section, it can be seen that a combination of longitudinal research, questionnaire and interview is an interesting way to conduct the experiment. However, it requires a longer time and more resources to implement. The stress test is applicable but not a must as the functionality of the system can be tested with another method. For example, soil moisture data can be collected at an interval of one hour to see whether the plants are properly watered. One of the good ways is by testing the soil moisture sensor with three different combinations of soils and plants under three different water condition as the adaptability of the soil moisture sensor and the optimum soil moisture level can be known.

2.7 Summary

All in all, the reviews from different subsections have improved the understanding of a wide range of modules for plant watering system specifically. Exteroceptive sensors, electronic control mechanism, human-machine interface, water supply mechanism, and assessment methods have been covered.

It is known that soil moisture sensor is crucial in monitoring the water content in the soil to provide feedback to the system. Although a number of microcontrollers have more or less similar features, Arduino is one of a good choice due to its user-friendly IDE for programming. Plus, interfacing a microcontroller with a soil moisture sensor and the water pump is a fundamental design for plant watering system. Yet, more features can be added to the plant watering system such as timer-controlled, water level sensing and so on.

In choosing a human-machine interface, an easy-to-use LCD combined with input buttons is a better choice because of its simplicity, low cost of implementation and low power consumption. Regarding the water supply mechanism, the water pump is slightly better than water valve in designing a plant watering system as it can pump dechlorinated water source from a reservoir and water pump which relies on a motor can have various output depends on the power and hence flow rate can be controlled. Last but not least, testing the plant watering system under different water condition and obtaining soil moisture data are two good ways of assessing the system.

Actually, more literature reviews can still be done to boost the system with more sophisticated elements. However, the current review has provided a good understanding and knowledge on how to design a demand-operated plant watering system. Plant watering system will still be an essential technology to future agriculture. So, contribution to design, develop and evaluate a plant watering system plays a vital role in making human life easier.

CHAPTER 3

METHODOLOGY

3.1 Introduction

In this chapter, methodology in designing the system to achieve the objectives was discussed. Several methods have been studied and evaluated from the literature review. The methodology is comprised of flow charts of project and system, block diagram, hardware and software selection, simulation, cost of the material, experiment set-up as well as a Gantt chart. The methodology plays a vital role by ensuring a well-planned and systematic project development.

3.2 Project method development

Figure 3.1 shows the flow chart of the method development. It started with the literature review and evaluation followed by the planning of the flow of the system. Then, suitable hardware and software were selected based on the literature review. The simulation was done to ensure the feasibility of the program on hardware. Next, two experiments were planned in order to test the plant watering system when the assembly of the hardware was done. Lastly, the plant watering system was analysed and evaluated based on the designed experiments.

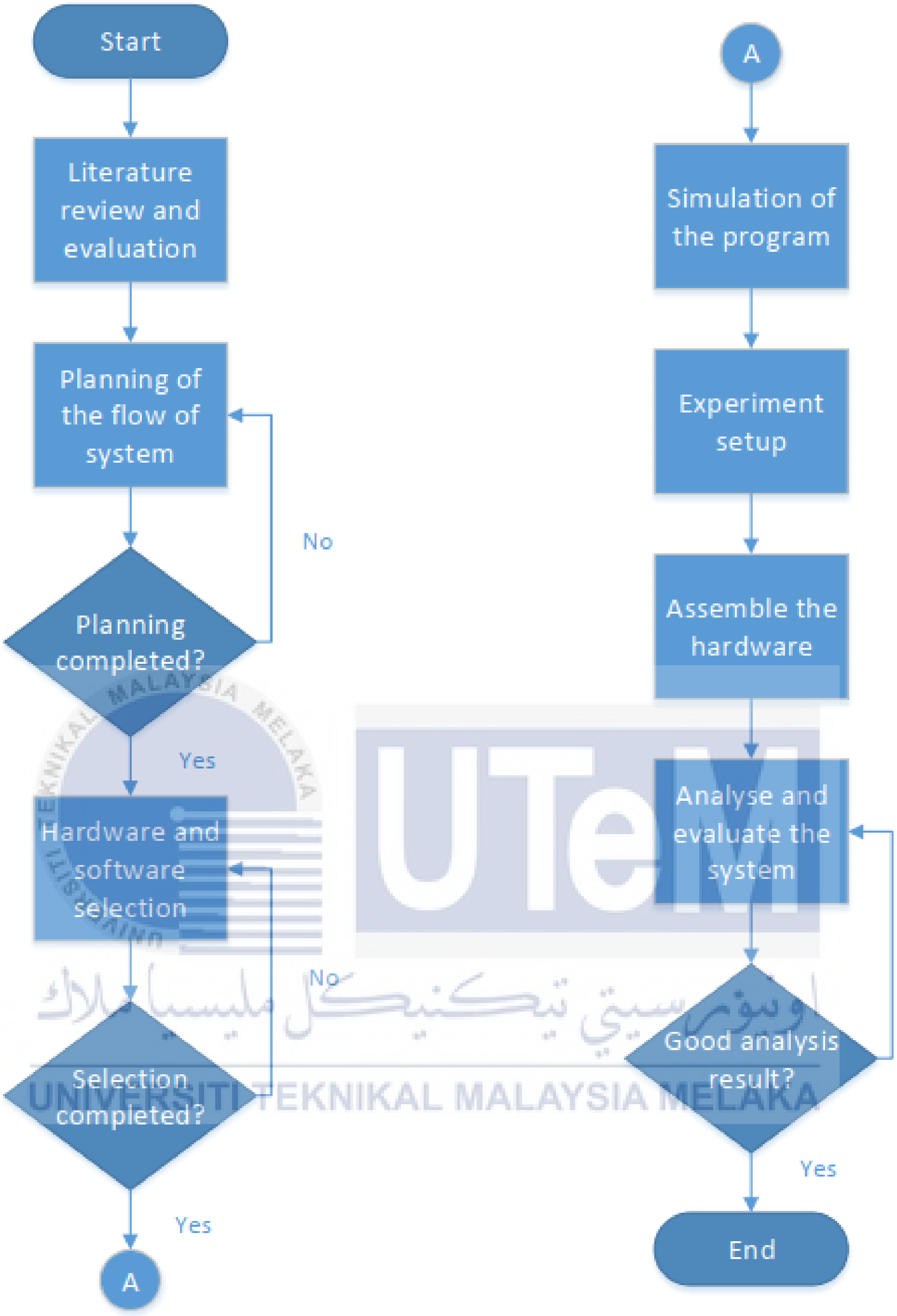


Figure 3.1: Flow chart of project method development

3.3 The flow of the program

According to Figure 3.2, the program starts by displaying the date, time, soil moisture level and water level. Then, the system will check whether the water level in the reservoir is above the threshold. The buzzer will go off if the water level is low to notify the user. The user is allowed to set the ON time and OFF time for watering. The system will check whether the ON time has reached. The system will keep checking until the ON time reaches and then check the soil moisture level followed by the OFF time. Next, the water pump will be turned on at the ON time if the soil moisture level and the OFF time are not yet achieved. The water pump will turn off immediately if the soil moisture is achieved before the OFF time. Same goes to the case of OFF time reached before the soil moisture level. The program will keep looping unless the power supply is cut off.

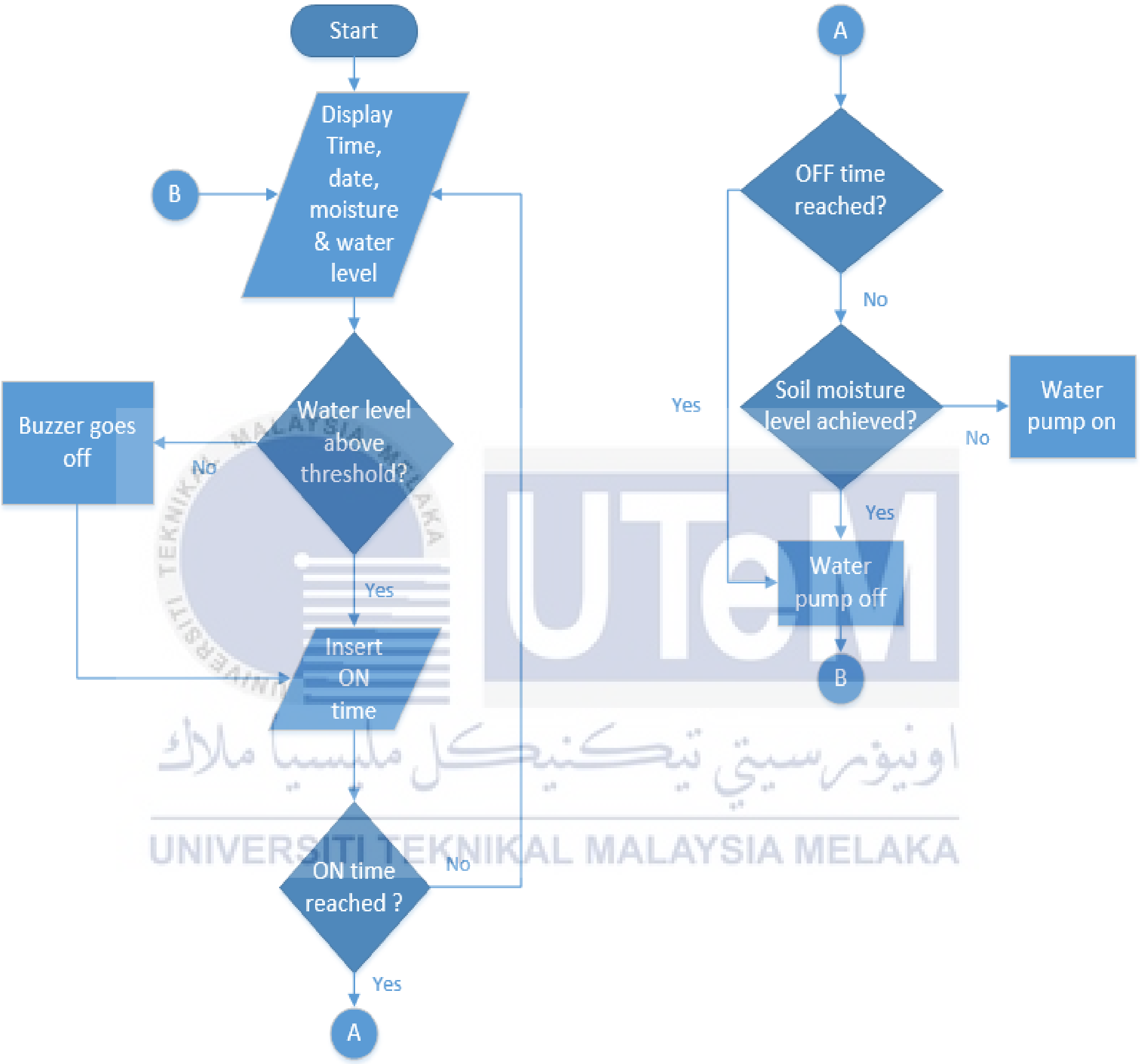


Figure 3.2: Flow chart of Demand-operated plant watering system

3.4 Design overview

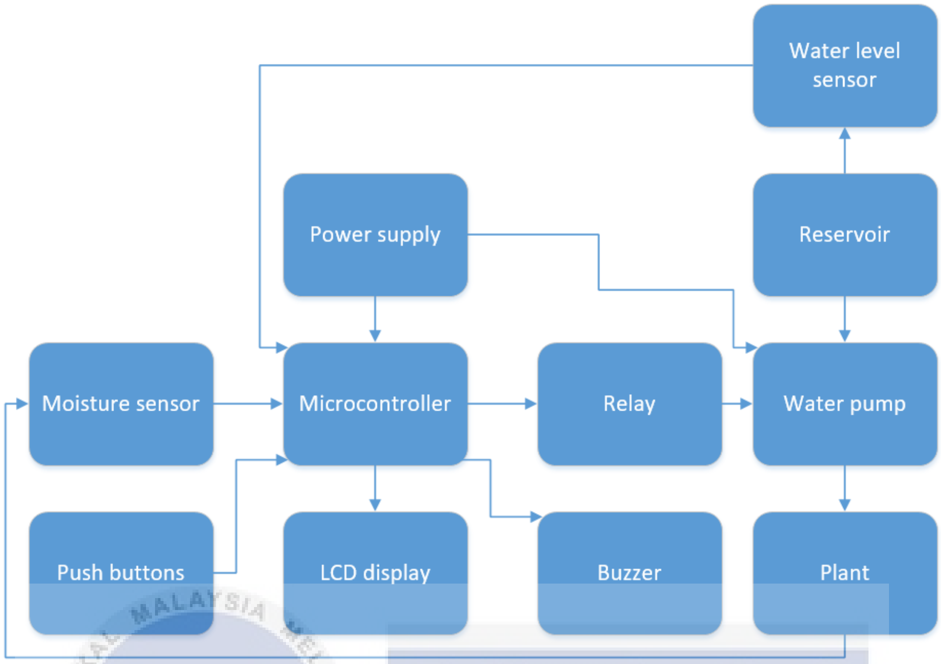


Figure 3.3: Block diagram of Demand-operated plant watering system

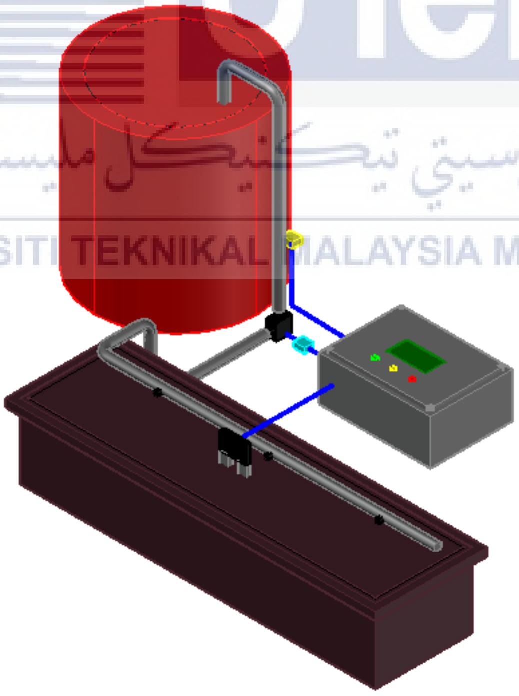


Figure 3.4: CAD drawing of demand-operated plant watering system

Based on Figure 3.3, there are input components such as soil moisture sensor, water level sensor, and pushbuttons as well as output components including LCD display, water pump, relay, and buzzer. All the inputs and outputs are controlled by the brain of the system which is a microcontroller. The pushbuttons are for the user to insert the run time for watering. The relay receives instruction from the microcontroller to trigger the water pump so that water can be supplied to the plants from the reservoir. In addition, soil moisture sensor acts as the feedback to update the soil moisture status to the system while the water level sensor updates the water level of the reservoir. The buzzer will go off when the water level is low to inform the user. Moreover, LCD displays the current date and time, water level and soil moisture level.

3.5 Design of the system

3.5.1 Electrical component

Figure 3.5: Arduino UNO

Figure 3.5 shows an Arduino UNO. It was used as the microcontroller of the plant watering system due to the ease of programming and enough number of input/output pins. The Arduino UNO acts as the brain of the system where it is interfaced with a soil moisture sensor, a water level sensor, an RTC module, a relay module, an LCD, a buzzer, push buttons and a water pump. The Arduino UNO with the uploaded Arduino code controls the outputs including the water pump, relay module, buzzer and LCD based on the input data from soil moisture sensor, push buttons, water level sensor and RTC module.

Figure 3.6: LCD module interfaced with I2C

Figure 3.6 shows an LCD module interfaced with I2C. The 20X4 I2C LCD acts as the user interface of the system. It was chosen because the 4 rows of information including current time, date, soil moisture level, the water level can be displayed. Besides, the 20X4 I2C LCD takes up only 2 pins for communication which are SDA and SCL. It saves a number of pins on the Arduino board and therefore it is better compared to normal LCD without I2C interface which requires 6 pins. Timers are displayed on the LCD also and it is adjustable by the user with the help of the push buttons.

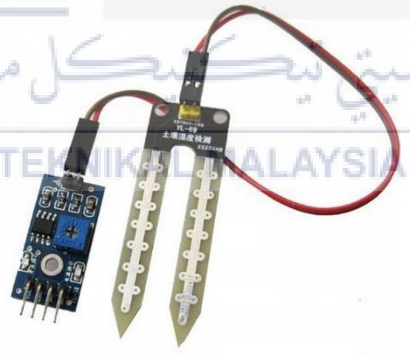


Figure 3.7: Soil moisture sensor

Figure 3.7 shows a soil moisture sensor. Soil moisture sensor plays a vital role in the plant watering system as it was used to monitor the soil moisture level where the health and survival of the plants depend on it. A threshold level was set for the sensor where a signal will be sent to the microcontroller if soil moisture value below the threshold is detected.

Figure 3.8: Water level sensor

Figure 3.8 shows a water level sensor. Water level sensor was used to monitor the water level in the water container. The water level sensor was placed at the bottom level of the water container. A signal will be sent to the microcontroller when the water level is below the float switch which indicates that the water level is low.

Figure 3.9: Buzzer

Figure 3.9 shows a buzzer. The buzzer serves as the alarm of the plant watering system. It will go off when a signal is received from the microcontroller. This was used to notify the user that the water level of the water container is low at that moment.

Figure 3.10: Relay module

Figure 3.10 shows a relay module. A relay is a switch with 2 contacts which are normally open (NO) contact and normally closed (NC) contact. Relay module which controls one device was chosen because only water pump requires a higher power supply. In the case of designing this plant watering system, the relay module was used to switch on the 240 V AC water pump since 5 V of supply from Arduino board is not enough to power the water pump. When the plants need to be watered, a signal will be sent by the microcontroller to the relay module. The relay module triggers the NO contact to complete the circuit and hence the water pump can be switched on.

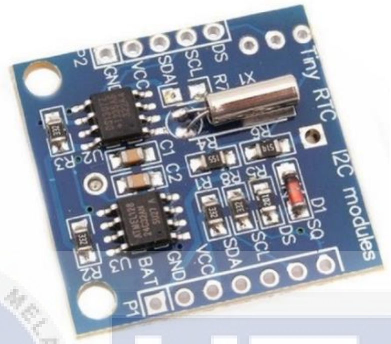


Figure 3.11: RTC module

Figure 3.11 shows an RTC module. The RTC module was used to update the plant watering system the current time. There is a backup battery mounted on the RTC module to ensure the time will not stop ticking. This way, the time will be retained even if the system is being turned off.

Figure 3.12: Pushbuttons

Figure 3.12 shows three pushbuttons. The pushbuttons were interfaced with the microcontroller to allow the user to input data to the system. For instance, the desired time to water the plants can be set through the pushbuttons.

Figure 3.13: Water pump

Figure 3.13 shows a submersible water pump. The water pump was used in the system to extract water from the container to the plant. A signal will be sent by the microcontroller to trigger the relay module and hence activate the water pump. This occurs when water is needed by the plant.

3.5.2 Mechanical component

Figure 3.14: Water container

Figure 3.14 shows a water container. The water container serves as the reservoir of the system. The water in the container can be pumped to supply the plants when needed. The water level in the container is monitored by the water level sensor.

Figure 3.15: Planter box

Figure 3.15 shows a planter box. The planter box was used to plant the desired vegetable or other plants. The water hose was connected to the planter box to allow the watering process.

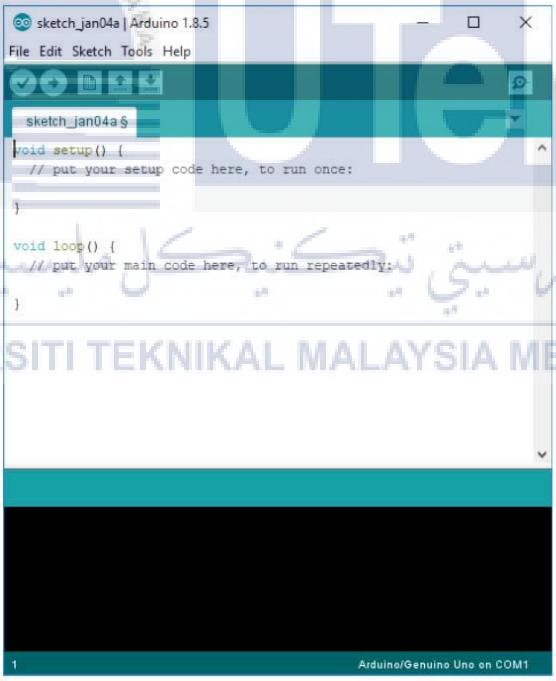
Figure 3.16: Water Hose

Figure 3.16 shows a black water hose. The water hose was connected from the water container to the edge of the planter box. In this way, water from the container can be delivered to the plants.

Figure 3.17: Electronic box

Figure 3.17 shows an electronic box. It was used to keep the Arduino board and other components inside to prevent damage to the components. However, the LCD and push buttons were placed on the surface on the electronic box to serve as the user interface.

3.5.3 Software



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Figure 3.18: Arduino IDE

Figure 3.18 shows the Arduino IDE. Arduino IDE is the Integrated Development Environment of Arduino where programming is done. Arduino language is similar to C language where a set of functions can be called from the code. To design the Demand-operated plant watering system, a program was assembled in the Arduino IDE based on the flow chart from Figure 3.2. RTC library was included in the code to allow the use of the real-time clock. The real-time clock keeps track of the current time. Apart from that, the EEPROM library was used as the memory of timers where the values which store in the bytes can be read or written. Besides, to allow the use of an LCD display, the LiquidCrystal_I2C library was included. Since LCD and RTC require I2C communication, Wire library was included as well. The program was written in a way that 4 timers are allowed for the user to set the on time and off time for watering. The LCD displays the current time and date based on RTC, the soil moisture level and water level based on the reading from the sensors. There are 7 pages for the display where the first page exhibits the current information while the consecutive 4 pages are the setting for the 4 timers and the last 2 shows the save setting. 3 input buttons were programmed to serve different functions where the first one is UP, the second one is SELECT and the third one is DOWN. The UP and DOWN buttons are used to move from one page to another and set the desired timer whereas the SELECT button is used to confirm the settings and save the set time. If else statement was also used to set the conditions for the operation of relay and buzzer.

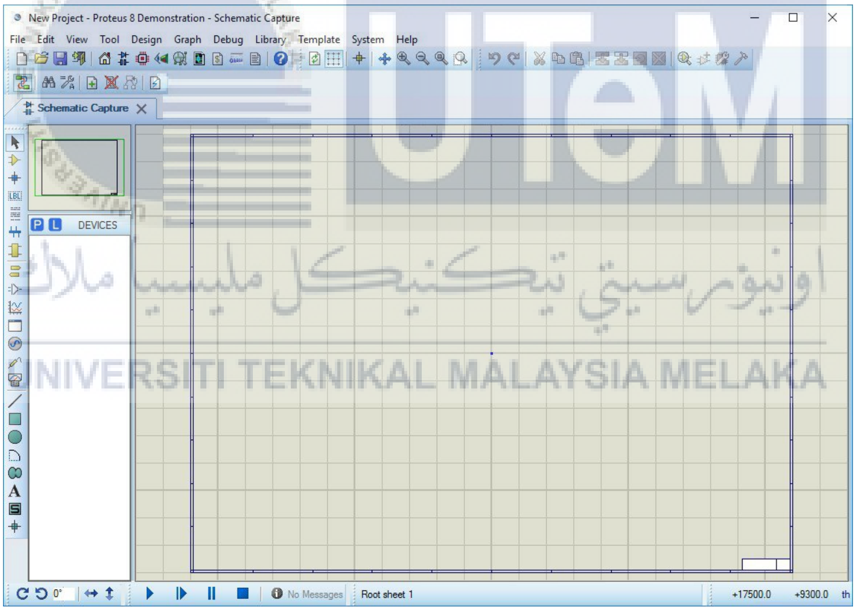


Figure 3.19: Proteus

Figure 3.19 shows the interface of the Proteus software. When the coding was finished and compiled successfully, the workability of the program was tested by Proteus software. Proteus is a software used for electronic design simulation. Components such as Arduino board, LCD, I2C bus interface, RTC, 3 pushbuttons, 2 LED, buzzer, DC motor, and 2 potentiometers were used. The motor was used to represent the output of the water pump while the potentiometers were used to represent the input of soil moisture sensor and water level sensor. All the components were linked by wires and the inputs and outputs of Arduino were connected according to the declared pins in the program. To run the simulation, the hex file of Arduino software was inserted to the Arduino board in Proteus.

3.5.4 Simulation

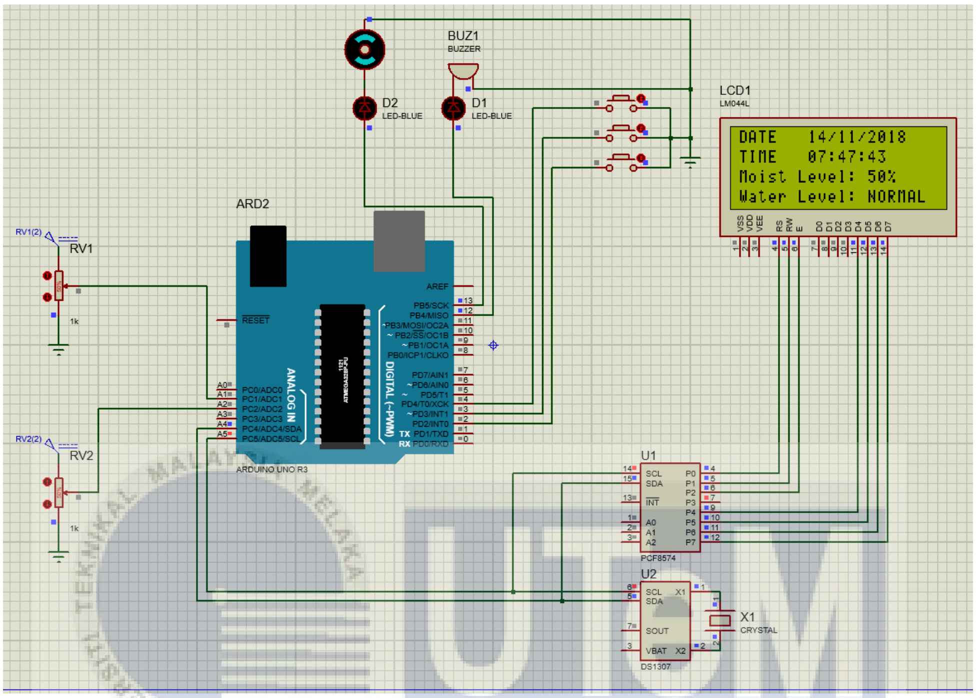


Figure 3.20: Simulation of Demand-operated plant watering system

Figure 3.20 shows the simulation was successfully done in Proteus. 3 pushbuttons were linked to pin 2, 3 and 4 respectively where pin 2 is declared as UP button, pin 3 is SELECT button and pin 4 is DOWN button. Pin A1 and A2 have connected to two different potentiometers which represent the inputs from the soil moisture sensor and water level sensor. Besides that, the RTC and I2C interfaced with the 20x4 LCD were linked to pin 4 (SDA) and pin 5 (SCL) which are the communication pins. Lastly, the buzzer was connected in series with an LED to pin 12 and a DC motor and another LED which connect in series were linked to pin 13. The output of pin 12 is based on the input from pin A2 which is the water level sensor while the output of pin 13 is based on the input from pin A1 which is the soil moisture level sensor. From the LCD, information including date, time, soil moisture level and water level were displayed. Since the program was workable in the simulation, it should be able to work in hardware as well.

3.6 Cost of material

Table 3.1: Total price of material

Component	Unit	Price per unit (RM)	Total price (RM)
Arduino UNO	**1**	**40.00**	**40.00**
LCD		**50.00**	**50.00**
Soil moisture sensor	**1**	**15.00**	**15.00**
Water level sensor	**1**	**20.00**	**20.00**
Buzzer	**1**	**5.00**	**5.00**
Relay module	**1**	**15.00**	**15.00**
RTC module	**1**	**20.00**	**20.00**
Water container	**1**	**30.00**	**30.00**
Water pump	**1**	**20.00**	**20.00**
Pushbutton	**3**	**2.00**	**6.00**
Electronic box	**1**	**20.00**	**20.00**
Planter box	**1**	**20.00**	**20.00**
Total amount			**261.00**

3.7 Experiment setup

To test the plant watering system, two experiments were designed including the calibration test and functionality test.

3.7.1 Experiment 1

In this experiment, the ADC values ranged from 0 to 1023 of the soil moisture sensor were observed when it was tested with 5 different conditions. Figure 3.21 shows the setup of experiment 1.

Figure 3.21: Experiment 1 setup

A laptop was connected to the Arduino board of the plant watering system to acquire the ADC value through the serial monitor of Arduino software. Firstly, the readings of ADC when the soil moisture sensor was suspended in the air were taken. Then, the soil moisture sensor was inserted to the dry soil, humid soil, and water as seen in Figure 3.22.

Figure 3.22: Dry soil, humid soil, water

The soil moisture sensor was also tested in soil with excess water as shown in Figure 3.23.

Figure 3.23: Overwatered soil

All ADC values were observed and recorded from the serial monitor of Arduino software. To obtain better readings, the soil moisture sensor was properly placed into the soil and water where the soil and water should cover the whole metal probes of the sensor. Since the ADC values of the soil moisture sensor fluctuated when it was placed into humid soil and water, a range of the ADC values for each condition was taken. All the data were tabulated into a table and presented in the graph for analysis.

3.7.2 Experiment 2

In experiment 2, the readings from the soil moisture sensor displayed on the LCD were observed and recorded with an interval of one hour for consecutive two days from 7.00 am of the first day until 10.00 pm of the second day.

Experiment 2 was set up as in Figure 3.24. Three chili plants were planted into the planter box for the experiment with the soil moisture sensor placed near the root zone of the chili plant in the middle.

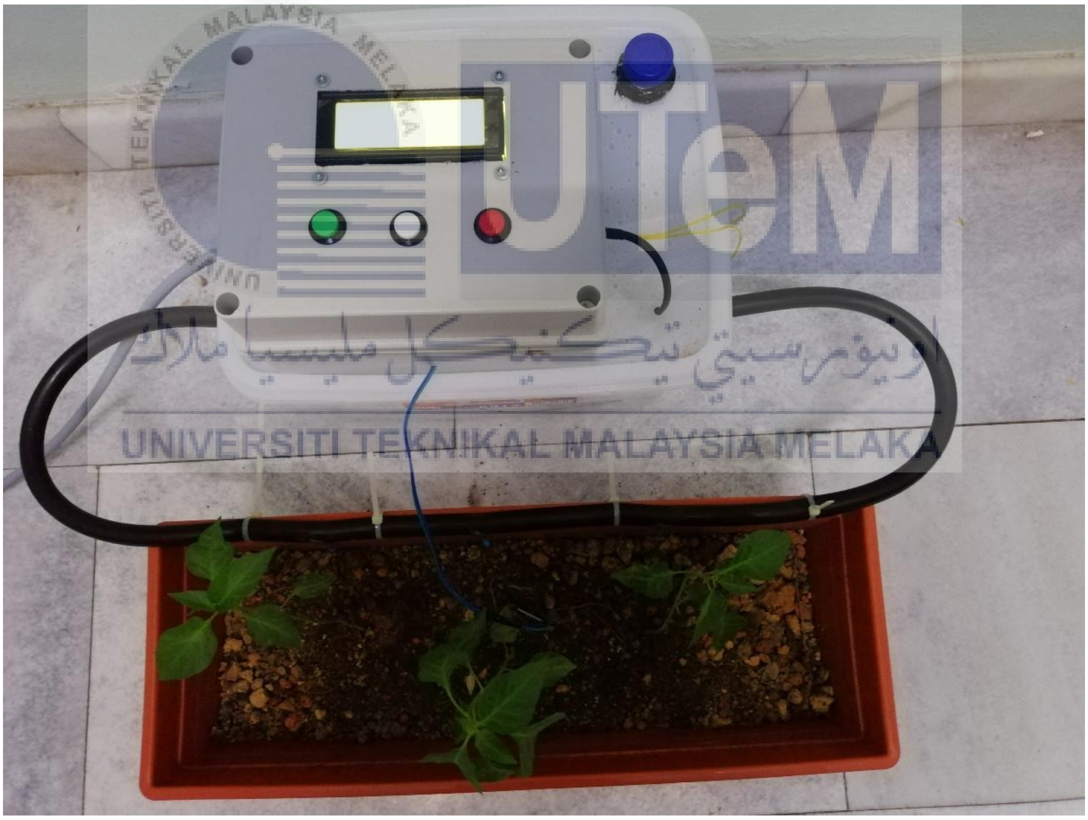
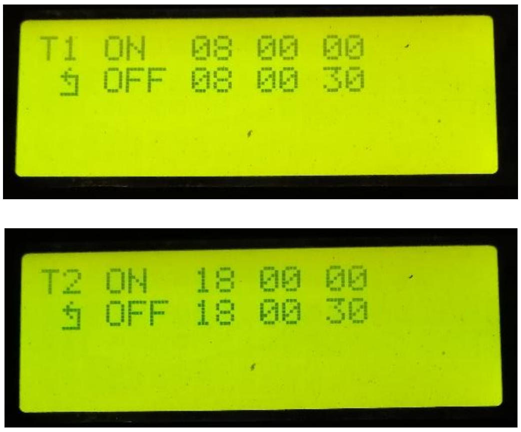


Figure 3.24: Experiment 2 setup

Plus, the desired time to water the plants was set at 8.00 am in the morning and 6.00 pm in the evening respectively for a duration of 30 seconds as seen in Figure 3.25.



```
T1 ON  08 00 00
 ↵ OFF 08 00 30
```

```
T2 ON  18 00 00
 ↵ OFF 18 00 30
```

Figure 3.25: Two desired water time

To obtain a more accurate result, the soil moisture sensor was properly placed with the metal probes fully inserted into the soil. In addition, the sensor was not moved until the end of the experiment. Besides, the experiment was conducted in the living room near the windows since it is a plant watering system for the indoor vegetable garden. The obtained data were tabulated into a table and presented by graph for analysis.

3.8 Gantt chart

Item	2018																2019																			
	September				October				November				December				January				February				March				April				May			
	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4
Introduction -Discuss with supervisor regarding the project -Objective -Scope	■	■	■	■																																
Literature review -Look for journals -Search additional information about the project				■	■	■	■																													
Methodology -Decide components -Way to design and develop the system -Design experiments							■	■	■	■																										
Expected result -Predict outcome of the system										■																										
FYP 1 presentation -Complete FYP 1 report											■	■	■	■																						
Design process -Coding -Design and develop the hardware																		■	■	■	■	■	■	■	■	■	■	■								
Testing process -Functionality test -Run experiment																									■	■	■	■								
Analysis and discussion -Data tabulation and Graph presentation -Analyze collected data -Discussion on result and error																													■	■	■	■	■			
FYP 2 presentation -Make conclusion -complete FYP 2 report																																		■	■	■

Figure 3.26: Gantt chart

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Overview

In this chapter, the results of the hardware, experiment 1 and experiment 2 were analysed and discussed. For the hardware part, the design and operation of the demand-operated plant watering system were shown and explained. For experiment 1, the responses of the soil moisture sensor under different conditions were determined. For experiment 2, the functionality of the plant watering system was tested where the soil moisture percentage was observed for consecutive two days. The two experiments are the calibration test and the functionality test.

4.2 Hardware result

The plant watering system consists of three main parts which are the electronic box, the water container, and the planter box. For the electronic box, there are LCD which displays the information such as date, time, soil moisture level, and water level as well as three buttons for the user to configure desired time to water the plants. Inside the electronic box, the components including the Arduino, RTC, and relay module were sealed in the electronic box. Figure 4.1 shows the top view of the prototype.

Figure 4.1: Top view

For the water container, a water level sensor and a water pump were placed within as seen in Figure 4.2. Apart from that, there is a blue cap which can be opened to allow the user to refill the water container when the water level is low.

Figure 4.2: Side view

On the other hand, a black hose was connected from the water pump to the edge of the planter box as shown in Figure 4.3. There are three outlets along the black hose while the end of the hose was sealed. A soil moisture sensor was also inserted in the soil to detect the soil moisture. In addition, a buzzer which can be seen from Figure 4.3 was placed at the side of the electronic box where the user will be alerted when the water level is low. To power the Arduino board and the water pump, a plug was used as well. Since the Arduino only required DC 9V, a down converter was used to regulate the AC voltage to DC 9V.

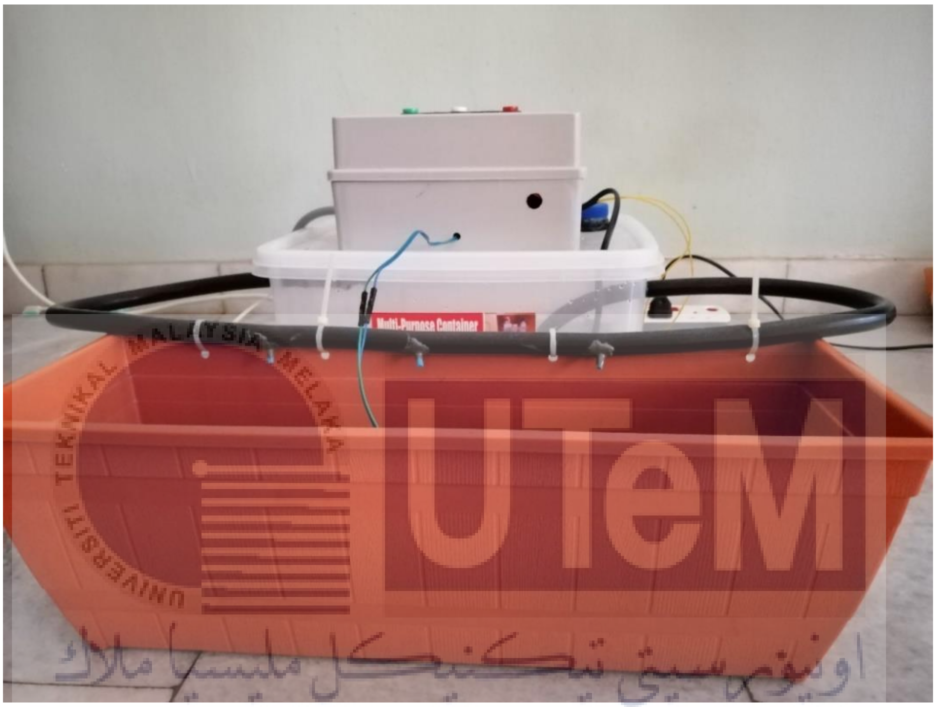


Figure 4.3: Front view

The LCD of the plant watering system can display 7 different pages including the main page, 4 timer setting pages, and 2 save setting pages. Figure 4.4 shows the main page of LCD where the first row of the 20x4 LCD displays the date while the next row exhibits the time. The date and time are acquired from the RTC and the RTC will always update the current time to the LCD. In addition, the third row and fourth row display the soil moisture level and water level of the container respectively.

```
DATE    26/05/2019
TIME    22:51:08
Moist Level: 72% WET
Water Level: NORMAL
```

Figure 4.4: Main page

For the soil moisture level, the ADC readings from the soil moisture sensor were converted into percentage before displaying on the LCD. The LCD displays “WET” when the soil moisture percentage is equal or more than 70 % and displays “LOW” when the soil moisture percentage is below 70 % but more than 10 %. In addition, the “DRY” will be displayed when the soil moisture is below or equal to 10 %. Figure 4.5 until Figure 4.7 show the three conditions displayed on the LCD.

```
DATE    26/05/2019
TIME    22:51:08
Moist Level: 72% WET
Water Level: NORMAL
```

Figure 4.5: Wet soil condition

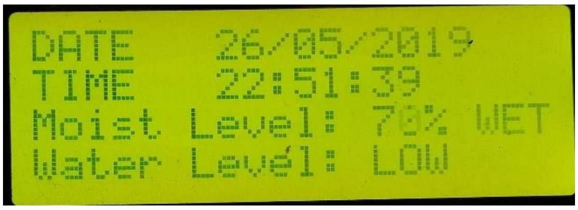
```
DATE    26/05/2019
TIME    22:40:30
Moist Level: 59% LOW
Water Level: NORMAL
```

Figure 4.6: Low soil moisture condition

```
DATE    26/05/2019
TIME    22:40:56
Moist Level: 1% DRY
Water Level: NORMAL
```

Figure 4.7: Dry soil condition

For the water level sensor, it is placed at the bottom level of the water container. Therefore, the LCD will display “LOW” as depicted in Figure 4.8 when the water level is below the sensor level and display “NORMAL” when the water level is above the sensor level. The readings from both the sensors will always be updated to the LCD, this way the user will get to know the current status of the soil condition and the water level of the container.



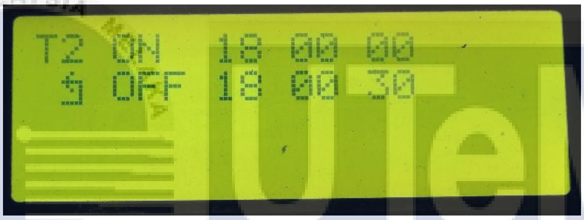
```
DATE    26/05/2019
TIME    22:51:39
Moist Level: 70% WET
Water Level: LOW
```

Figure 4.8: Low water level

Figure 4.9 until Figure 4.12 show the four setting pages of the plant watering system where four different times to water the plant can be set by the user.

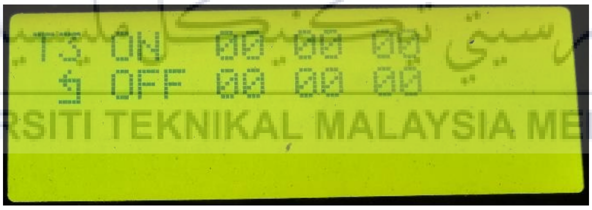
```
T1 ON  08 00 00
 ↰ OFF 08 00 30
```

Figure 4.9: Timer 1



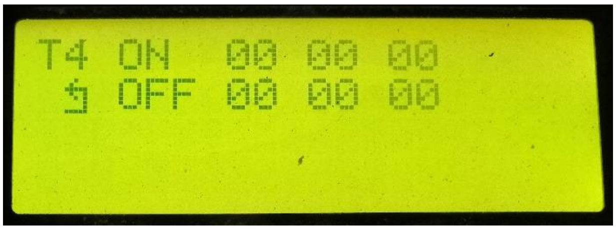
```
T2 ON  18 00 00
 ↰ OFF 18 00 30
```

Figure 4.10: Timer 2



```
T3 ON  00 00 00
 ↰ OFF 00 00 00
```

Figure 4.11: Timer 3



```
T4 ON  00 00 00
 ↰ OFF 00 00 00
```

Figure 4.12: Timer 4

Different on time and off time can be configured through the setting pages by using three pushbuttons. The green button is UP button and the red button is the DOWN button while the white one in the middle is the SELECT button. The main page which shows the current information and all the setting pages can be swapped by pressing the UP or DOWN button. By referring to the diagrams, timer number, return icon, ON time and OFF time are displayed on the LCD. Moreover, an arrow will appear for selection if the SELECT button is pressed. The desired ON time and OFF time can be set accordingly by using the three pushbuttons. When all the timers are well-set, subsequent page after Timer 4 will exhibit "PRESS SEL TO SAVE" as shown in Figure 4.13. When the SELECT button is pressed, the page which displays "SAVED" will appear as shown in figure 4.14 and then go back to the main page.

Figure 4.13: Save page

Figure 4.14: Save done

4.3 Experiments result

4.3.1 Experiment 1

In this experiment, the readings from the soil moisture sensor were taken under five different conditions. The results were tabulated into Table 4.1.

Table 4.1: ADC readings

Condition	ADC
Air	**1009 - 1010**
Dry soil	**1001 - 1008**
Humid soil	**326 - 391**
Water	**305 - 327**
Overwatered soil	**185 - 256**

Figure 4.15 shows the bar chart was plotted based on the ADC readings under different conditions.

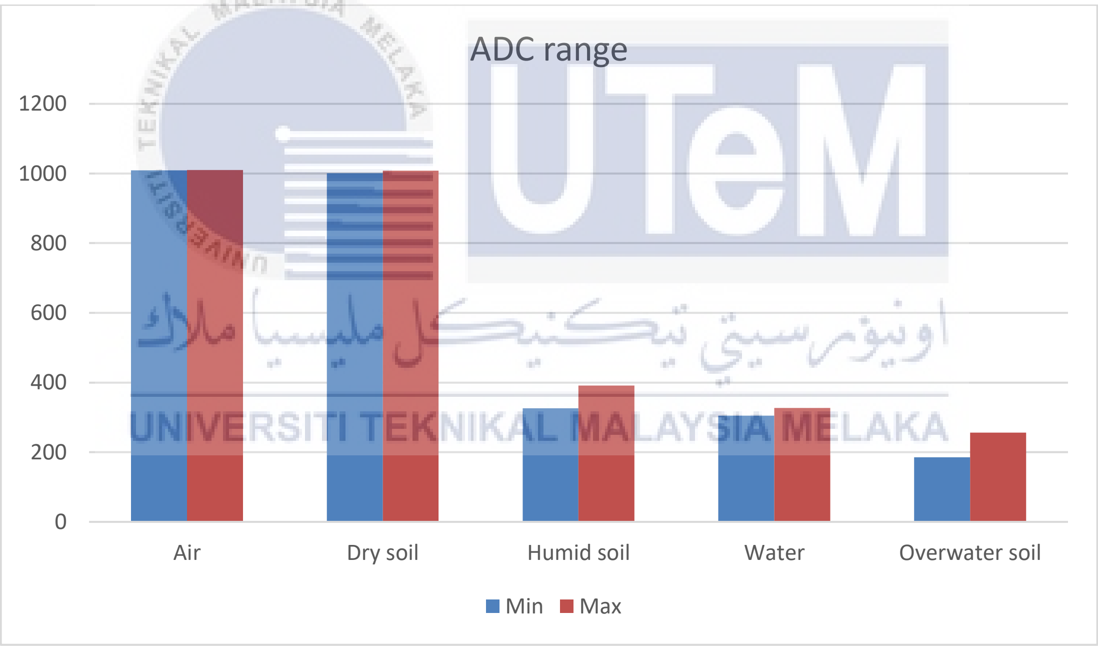


Figure 4.15: Graph of ADC readings

According to Figure 4.15, it shows a big difference in ADC when the water was present or not present. For air and dry soil, the ADC readings fall at roughly 1000 since there was no water. For humid soil, water and overwatered soil, the ADC readings fall within the range of 185 to 391 since the water was present.

The ADC value of the soil moisture sensor in the air is the highest while the ADC value in the overwatered soil is the lowest. Besides, the difference in ADC between the air and dry soil is insignificant. However, there is a slight difference in ADC value between humid soil, water and overwatered soil. The ADC value of soil moisture sensor is relatively high when it was placed in dry soil and in the air, but the ADC value drops drastically when it was in humid soil, water and overwatered soil. This proves that the change in ADC is big with the presence of water.

Although the exact amount of water is not known, it is still a good way to know the presence of water in the soil. In fact, the resistance between the two metal probes of the soil moisture sensor is represented by the ADC value ranged from 0 – 1023. The higher the resistance, the higher the value of ADC. Hence, the resistance between the metal probes decreases when water is present in the soil. As a result, the ADC value decreases as well.

Since it is not likely to reduce the resistance down to zero, so the lowest ADC which can be obtained is only 185. As a matter of fact, there is some resistance which can never be removed such as the air gap between the soil, the soil itself, the soil moisture sensor itself and so on. Same goes to the maximum ADC, the high ADC obtained is 1010 instead of 1023. In fact, the range of ADC obtained can still be slightly higher or lower than the range from the experiment depends on the conditions.

Hence, the range of ADC from 150 to 1010 was selected as the minimum and maximum value for the soil moisture sensor where 150 is slightly lower than the obtained reading of 185. Moreover, the ADC range was mapped into percentage also to make it easier for users to read this value.

Generally, the range of ADC obtained for the humid soil is 326 – 391. Therefore, the range of humid soil is preferred to set as the threshold to control the watering process.

When the ADC readings were mapped into percentage, the readings in percentage were tabulated into Table 4.2 as well.

Table 4.2: Soil moisture percentage

Condition	**Soil moisture percentage (%)**
Air	**0 - 0.11**
Dry soil	**0.23 – 1.05**
Humid soil	**71.98 – 79.53**
Water	**79.42 – 81.98**
Overwater soil	**87.67 - 95.93**

Figure 4.16 displays the bar chart of soil moisture percentage under five different conditions.

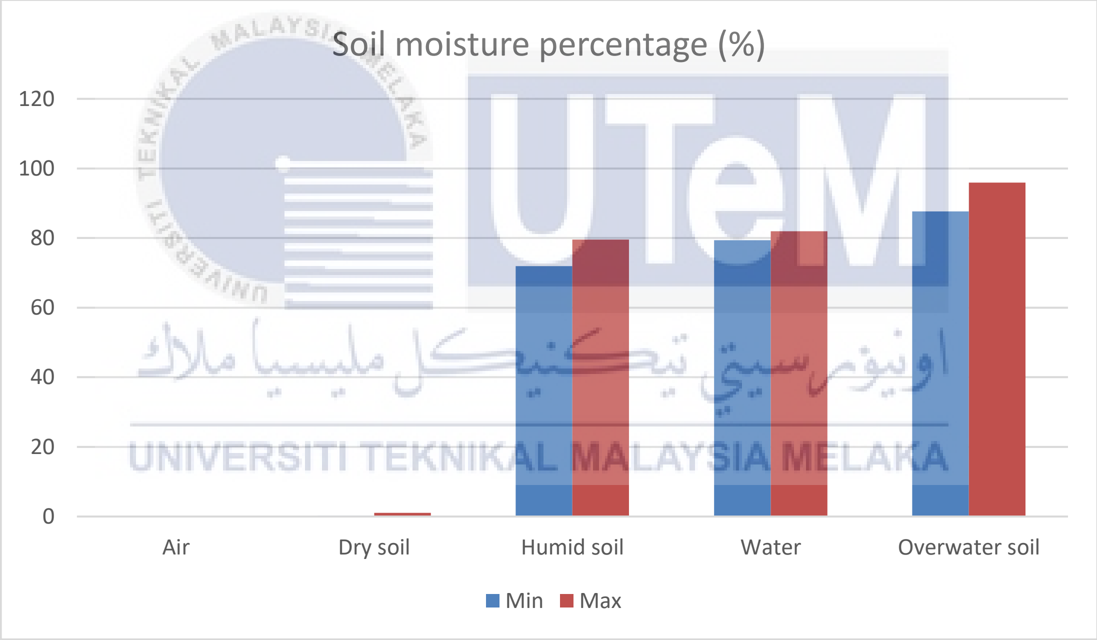


Figure 4.16: Graph of soil moisture percentage

Based on Figure 4.16, there is a big gap in soil moisture percentage when the water is present and not present. The soil moisture percentage in the air and dry soil are nearly zero while the soil moisture percentage in the humid soil, water and overwatered soil range from 71.98 % to 95.93 %.

The soil moisture percentage in the air is the lowest while the soil moisture percentage in overwatered soil is the highest. Besides that, the difference between the soil moisture percentage between air and dry soil is not significant. Yet, there are minor differences in soil moisture percentage between the humid soil, water and overwatered soil. This shows the inverse of the ADC readings. In fact, the soil moisture percentage is the inverse of the resistance represented by ADC. Thus, the lower the resistance, the higher the soil moisture percentage.

From Table 4.2, the range of soil moisture percentage is 71.98 – 79.53 % for humid soil which is preferable compared to overwatered soil of 87.67 - 95.93 %. Therefore, 70 % of the soil moisture was chosen as the threshold to trigger the water pump since it is slightly lower than the preferred range which indicates the presence of water in the soil. Any value below the threshold will trigger the water pump to engage the watering process.

4.3.2 Experiment 2

In experiment 2, the soil moisture percentage and status of the soil were observed with an interval of one hour for consecutive two days. The obtained readings were tabulated into Table 4.3.

Table 4.3: Soil moisture condition for consecutive 2 days

Day	Time	Soil moisture (%)	Status
1	0700	22	LOW
	0800	70	WET
	0900	67	LOW
	1000	68	LOW
	1100	68	LOW
	1200	68	LOW
	1300	69	LOW
	1400	70	WET
	1500	69	LOW
	1600	69	LOW
	1700	68	LOW
	1800	74	WET
	1900	72	WET
	2000	72	WET
	2100	72	WET
	2200	71	WET
	2300	71	WET

Day	**Time**	**Soil moisture (%)**	**Status**
2	**0000**	**70**	**WET**
	0100	**70**	**WET**
	0200	**70**	**WET**
	0300	**70**	**WET**
	0400	**69**	**LOW**
	0500	**69**	**LOW**
	0600	**68**	**LOW**
	0700	**68**	**LOW**
	0800	**74**	**WET**
	0900	**73**	**WET**
	1000	**72**	**WET**
	1100	**72**	**WET**
	1200	**71**	**WET**
	1300	**70**	**WET**
	1400	**70**	**WET**
	1500	**69**	**LOW**
	1600	**68**	**LOW**
	1700	**68**	**LOW**
	1800	**74**	**WET**
	1900	**72**	**WET**
	2000	**71**	**WET**
	2100	**70**	**WET**
	2200	**70**	**WET**
	2300	**69**	**LOW**

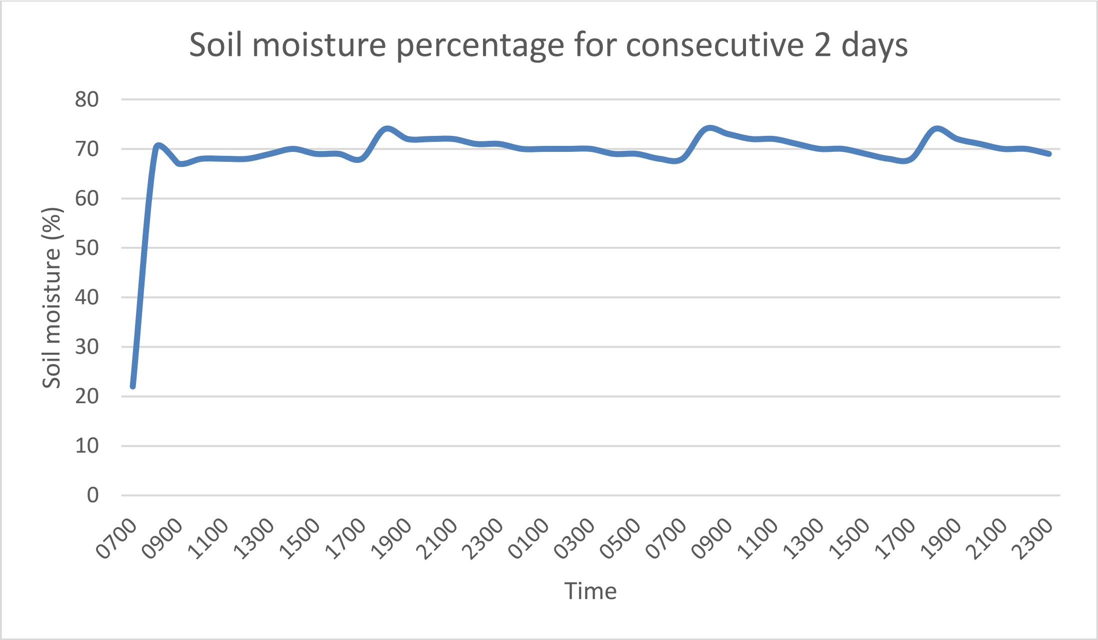


Figure 4.17: Graph of soil moisture percentage for consecutive 2 days

Based on Figure 4.17, there are four points where the soil moisture percentage rises. Plus, there is a spike of soil moisture percentage at the initial point. The soil moisture percentage falls gradually after each rise and then rises again. However, there are some fluctuations after the first rise where the soil moisture percentage drops and rises and then drops again before the second rise.

From the observation, the four points of soil moisture rise fall in the same period which is 8.00 am and 6.00 pm. It is proven that the plant watering system supplied water to the plants when the set time reached. It also shows that the plant watering system provides sufficient water slightly above the threshold after the watering process. A trend is discovered as depicted in the graph where the soil moisture rises at first and then falls before rising again. The same pattern repeated itself for the consecutive two days and therefore it is assumed that the same pattern would occur for the rest of the days.

In the experiment, the condition of the water pump was observed to see whether the pump is turned on. The results were tabulated into Table 4.4.

Table 4.4: Pump condition

	Set time	Pump condition
Day 1	**0800**	**ON**
	1800	**ON**
Day 2	**0800**	**ON**
	1800	**ON**

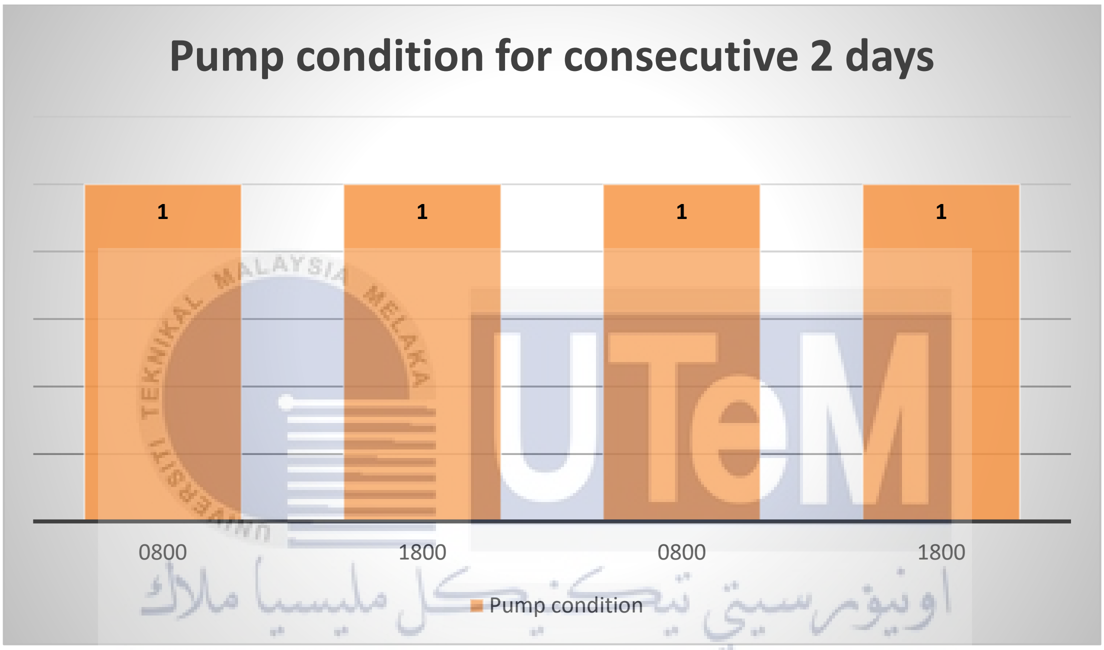


Figure 4.18: Pump condition for consecutive 2 days

From Figure 4.18, it is shown that the pump turned on at 8.00 am and 6.00 pm for the consecutive two days. The condition of the pump is represented by 1 and 0 where 1 indicates the pump is turned on while 0 indicates the pump is not activated. The activation of water pump indicates that the plants were watered.

4.4 Discussions

From the hardware result, it was proved that the demand-operated plant watering system has been assembled according to the design. Besides, the program has run successfully not only in simulation but in the hardware as well. The LCD can display necessary information such as the date, time, soil moisture percentage, water level and the desired time could be set through the three buttons. In addition, the buzzer can alert the user when the water level in the container is low meanwhile the LCD display low water level. The buzzer stops ringing as soon as the water in the container goes back to the normal level. In addition, the water pump can be controlled by the relay module through the signal from the Arduino when the soil moisture is below the threshold.

From experiment 1, the suitable soil moisture threshold was determined. The responses of the soil moisture sensor in different conditions including in the air, dry soil, humid soil, water and overwatered soil were tested. A range of ADC value from each condition was observed and recorded before it was converted to percentage. It is found that the range of soil moisture percentage in humid soil is 71.98 – 79.53 % and 0.23 – 1.05 % for dry soil. It is seen that there is a big gap in percentage or the ADC reading between these two conditions. In fact, the readings between the gap exist. However, the soil moisture percentage increases significantly with the slightest change in the amount of water. For example, 10 ml of water decreases the ADC value to 777 or 27.09 % of soil moisture while another 10 ml of water further decreases the ADC value to 354 or 76.28 % of soil moisture in one of the tests conducted. It makes setting and controlling the desired soil moisture difficult. Hence, a threshold value is needed to trigger the water pump and any value below the threshold is considered a low soil moisture level.

The threshold of soil moisture to trigger the water pump was set at 70 % which is slightly lower than the range for humid soil. Although 70 % is lower than the range of humid soil, there is still a big gap between the range of the dry soil and the wet soil which means the soil should still be moist enough for the plants. Moreover, programming the microcontroller to stop the water pump at exactly 70 % of soil moisture does not guarantee the same to occur on the hardware. This is due to the fact that there will a delay or interval when turning off the pump and the reading of soil moisture sensor increase drastically within a short period of time when watering which makes the readings hard to keep track. As a result, the reading obtained is usually slightly higher than the threshold value. Therefore, setting the threshold at 70 % of soil moisture will usually result in the soil moisture higher than the threshold which falls within the range of humid soil.

From experiment 2, the functionality of the plant watering system was tested for consecutive two days by observing the soil moisture condition for the plant watering system. It was found that the planting watering system able to water the plants at the desired period set by the user while maintaining the soil moisture percentage slightly above the threshold. The plants were watered twice each day for the consecutive 2 days according to the set time where the water pump was activated. Out of the four watering processes, 74 % of soil moisture was obtained after three watering processes and 70 % of soil moisture was obtained only after the first watering process. It proves that it is more likely the soil moisture percentage obtained after each watering process is higher than the threshold than getting the exact soil moisture percentage. Thus, setting the threshold at 70 % is a feasible way to prevent overwatering of plants while maintaining the soil under humid condition.

Since the plant watering system was placed indoor, it did not affect much by the weather change and therefore the water loss rate was slower if compared with outdoor plants. Plus, the plants are still small in size, so the water needed was lesser as well. In fact, the soil moisture sensor might fail sometimes where the soil moisture percentage will remain constant at 1 % no matter how much water is added due to the loss of connection. In this scenario, the timer function comes into play where the water pump will be stopped when the set period is over even though the soil moisture threshold is not reached. In this way, the timer acts as a redundant system to replace the soil moisture sensor to prevent overwatering of the plants from happening.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

All in all, a demand-operated plant watering system using Arduino with time setting function and soil moisture feedback for the indoor small-scale vegetable garden has been designed and developed. The performance of the demand-operated plant watering system in terms of time and soil moisture control has been assessed and analysed also. According to the full prototype, it is known that all the designed features have worked successfully. The soil moisture threshold was determined through experiment 1 also. For the performance and functionality of the demand-operated plant watering system, it is deemed successful also through experiment 2 since it could control the watering process in terms of time and soil moisture percentage.

5.2 Future Works

Although implementing the demand-operated plant watering system is viable, there are many other possibilities to improve the system. In order to design a reliable system, a robust soil moisture sensor is needed. A simple soil moisture sensor placed in a certain area of the soil cannot fully represent the soil moisture of the whole planter box. It will become less accurate when the covered area gets bigger. Therefore, it is crucial to install more sensors to obtain a more accurate reading. Besides, it is possible to program the system in a way that the user gets to decide the desired soil moisture level for the plants if the soil moisture sensors are accurate enough. With the correct adjustments and developments, this type of system has the potential to improve the health of plants by accurately and reliably providing the correct amount of water to each group of plants. In fact, the demand-operated plant watering system had been programmed where the user could set the desired soil moisture percentage. However, it was not reliable enough as it was difficult to control the exact soil moisture. Thus, improving the accuracy and robustness of the sensor is the key. There is still much room for further development, research, and test for the demand-operated plant watering system.

REFERENCES

[1] M. M. Ahmed, E. Ahmed and K. T. Ahmmed, "Automated Irrigation Control and Security System with Wireless Messaging," IEEE, 2013.

[2] A. Anil, A. R. Thampi, P. J. M and S. K. J, "Project HARITHA - An Automated Irrigation System for Home Gardens," IEEE, pp. 635-639, 2012.

[3] A. P and P. R, "Design and Implementation of Automatic Plant Watering System," International Journal of Advanced Engineering and Global Technology, vol. 4, no. 1, pp. 1567-1570, 2016.

[4] "Irrigation," National Geographic, [Online]. Available: https://www.nationalgeographic.org/encyclopedia/irrigation/.

[5] "Why Do We Need to Know the Soil Moisture Content of Soil?" Van Walt, 8 April 2015. [Online]. Available: https://www.vanwalt.com/news/2015/04/08/why-do-we-need-to-know-the-soil-moisture-content-of-soil/.

[6] I. Al-Bahadly and J. Thompson, "Garden Watering System Based on Moisture Sensing," in Ninth International Conference on Sensing Technology, 2015.

[7] K.Prathyusha and M. C. Suman, "DESIGN OF EMBEDDED SYSTEMS FOR THE AUTOMATION OF DRIP IRRIGATION," International Journal of Application or Innovation in Engineering & Management (IJAIEM), vol. 1, no. 2, pp. 255-258, 2012.

[8] I. Primisima, S. A. Sudiro and B. A. Wardijono, "Automatic Plant Watering Controller Component Using FPGA Device," IEEE, pp. 43-49, 2015.

[9] M. OJHA, S. MOHITE, S. KATHOLE and D. TARWARE, "MICROCONTROLLER BASED AUTOMATIC PLANT WATERING

SYSTEM," International Journal of Computer Science and Engineering (IJCSE), vol. 5, no. 3, pp. 25-36, 2016.

[10]A. Gupta, S. Kumawat and S. Garg, "Automatic Plant Watering System," Imperial Journal of Interdisciplinary (IJIR), vol. 2, no. 4, pp. 1123-1127, 2016.

[11]S. V. Devika, S. Khamuruddeen, S. Khamurunnisa, J. Thota and K. Shaik, "Arduino Based Automatic Plant Watering System," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 4, no. 10, pp. 449-456, 2014.

[12]J. Sandberg and T. Kokholm, "iPlant: Inteligent Plant System," IT-University of Copenhagen for the course Pervasive Project Fall 2012, 2012.

[13]N. Đuzić and D. Đumić, "Automatic Plant Watering System via Soil Moisture Sensing by means of Suitable Electronics and its Applications for Anthropological and Medical Purposes," Coll. Antropol, vol. 41, no. 2, 2017.

[14] J. Gutiérrez, J. F. Villa-Medina, A. Nieto-Garibay and M. Á. Porta-Gándara, "Automated Irrigation System Using a Wireless Sensor Network and GPRS Module," IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, vol. 63, no. 1, pp. 166-176, 2015.

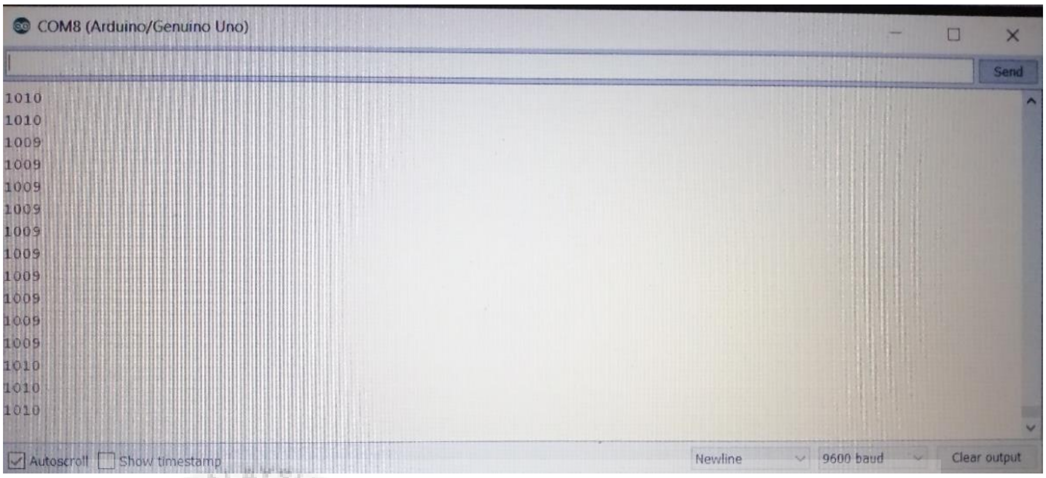
[15]A. Patil, M. Beldar, A. Naik and S. Deshpande, "Smart Farming using Arduino and Data Mining," in International Conference on Computing for Sustainable Global Development (INDIACom), India, 2016.

[16]K. Lekjaroen, R. Ponganantayotin, A. Charoenrat, U. S. Suree Funilkul and T. Triyason, "IoT Planting: Watering System Using Mobile Application for the Elderly," IEEE, 2016.

[17]N. Kumar and S. S. L, "Smart Farming System for Indian Farmers using Arduino based Technology," International Journal of Advance Research, Ideas and Innovations in Technology, vol. 3, no. 1, pp. 105-110, 2017.

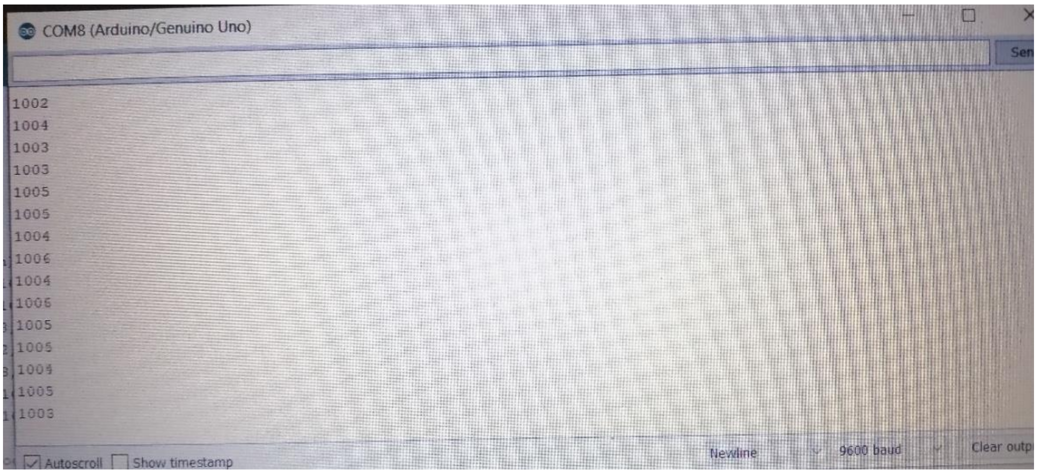
APPENDICES

APPENDIX A ADC READINGS FROM SERIAL MONITOR

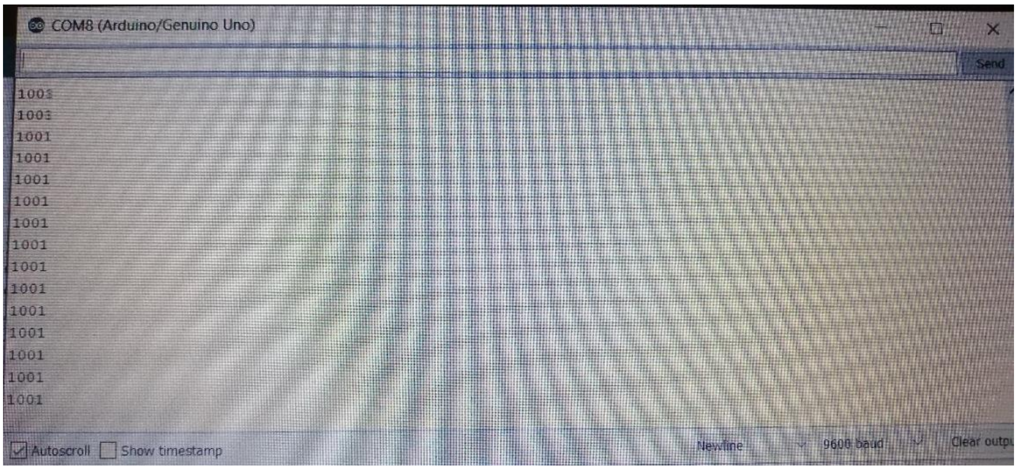


```
1010
1010
1009
1009
1009
1009
1009
1009
1009
1009
1009
1009
1010
1010
1010
```

```
1008
1008
1008
1008
1008
1008
1008
1008
1008
1008
1008
1008
1008
1008
1008
```

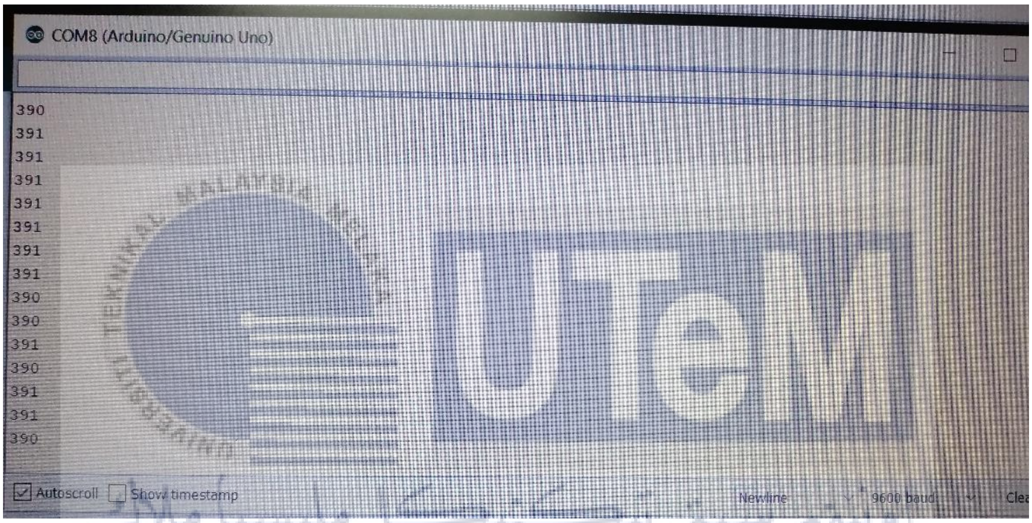


```
1002
1004
1003
1003
1005
1005
1004
1006
1004
1006
1005
1005
1004
1005
1003
```



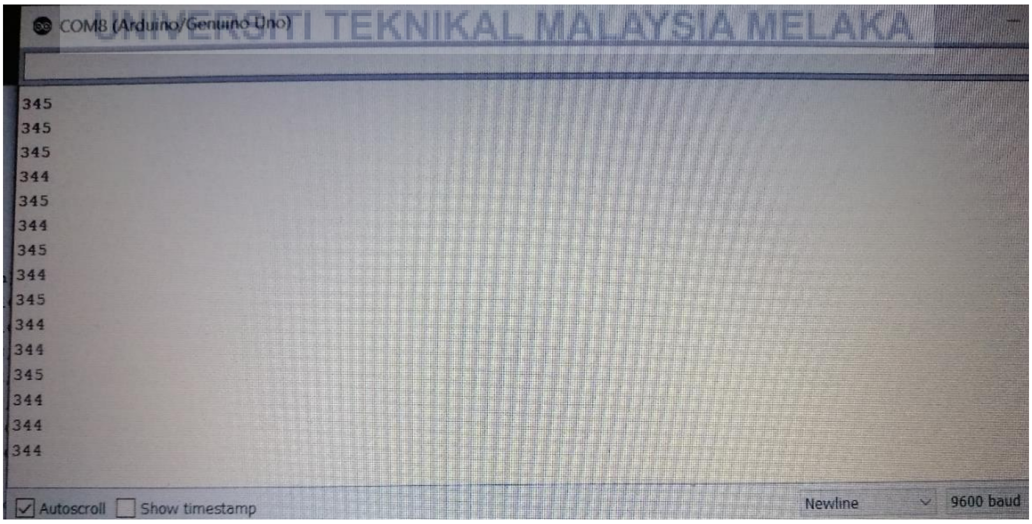
```
1001
1001
1001
1001
1001
1001
1001
1001
1001
1001
1001
1001
1001
1001
1001
```

☑ Autoscroll ☐ Show timestamp Newline 9600 baud



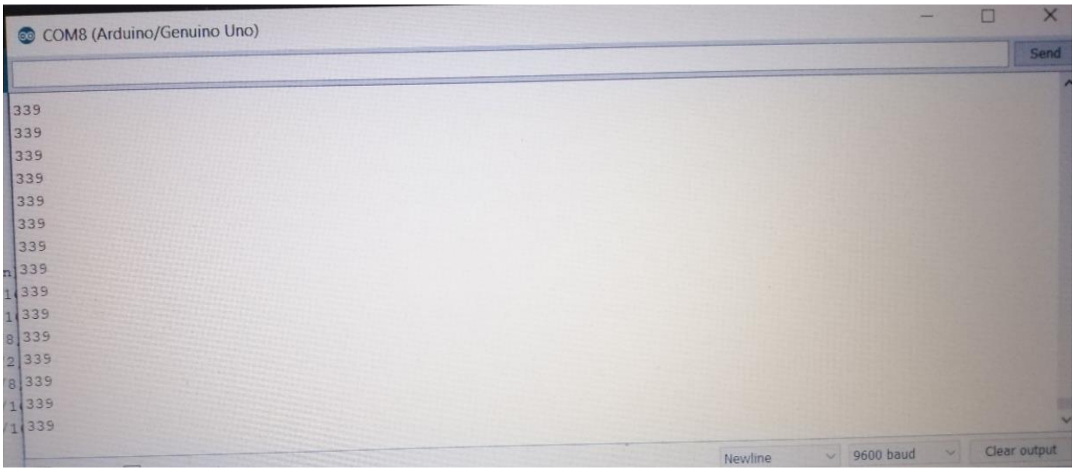
```
390
391
391
391
391
391
391
391
390
390
391
390
391
391
390
```

☑ Autoscroll ☐ Show timestamp Newline 9600 baud



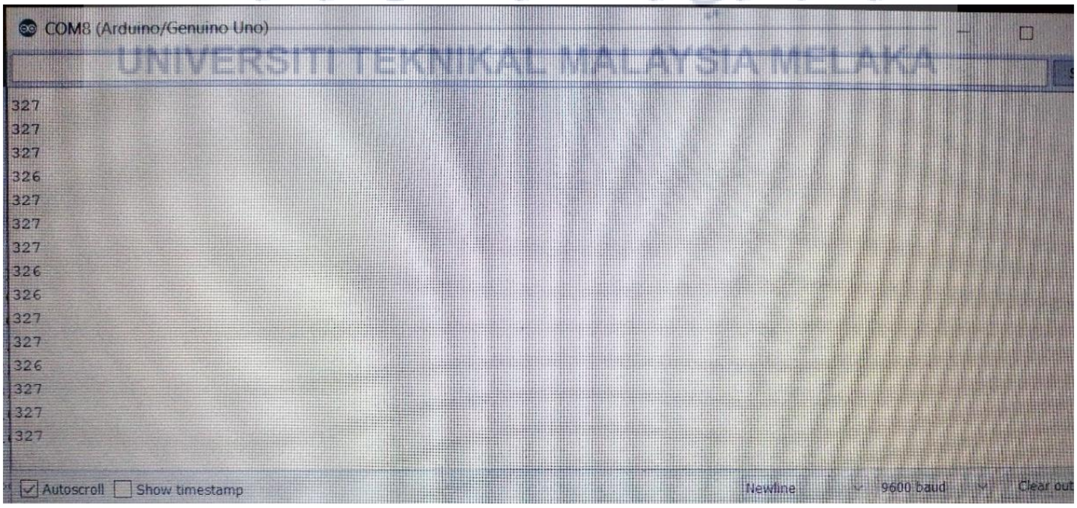
```
345
345
345
344
345
344
345
344
345
344
344
345
344
344
344
```

☑ Autoscroll ☐ Show timestamp Newline 9600 baud

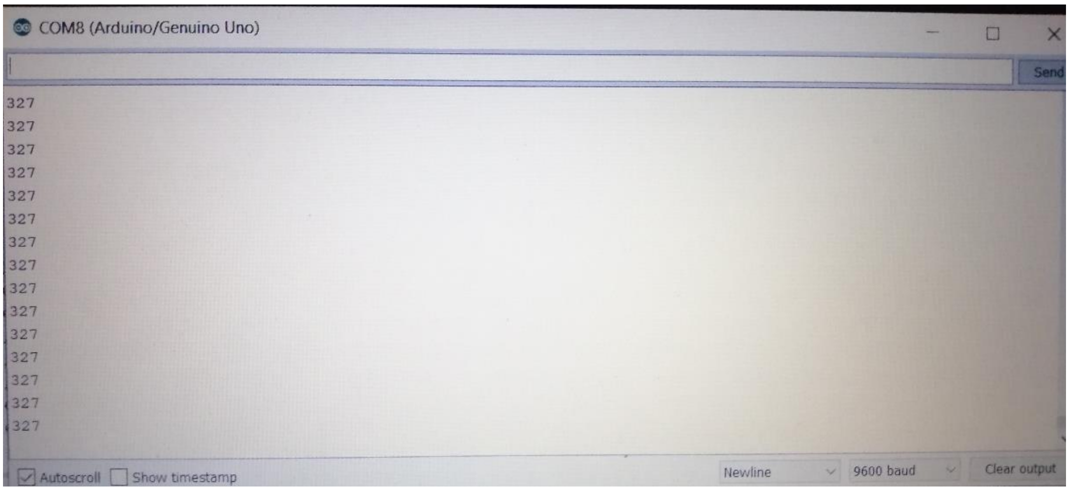


```
339
339
339
339
339
339
339
339
339
339
339
339
339
339
339
```

```
329
329
329
329
329
329
329
329
329
329
329
329
329
329
329
```

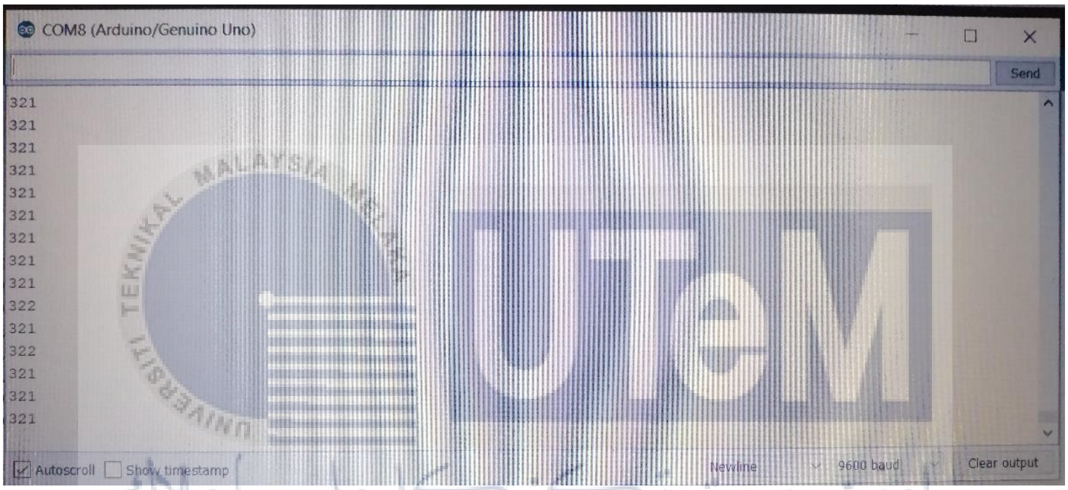


```
327
327
327
326
327
327
327
326
326
327
327
326
327
327
327
```



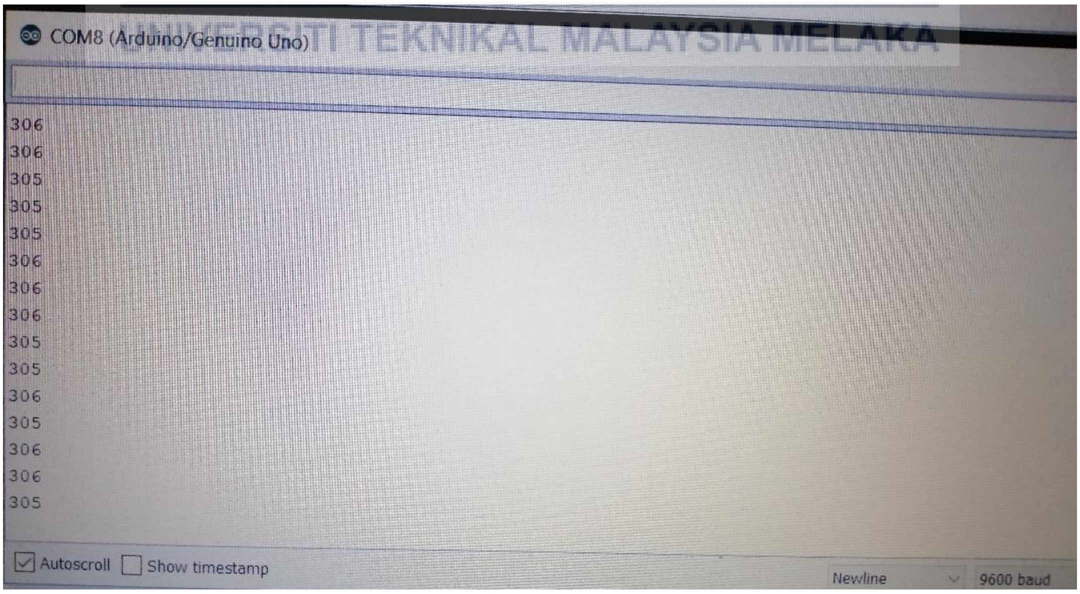
```
327
327
327
327
327
327
327
327
327
327
327
327
327
327
327
```

☑ Autoscroll ☐ Show timestamp Newline 9600 baud



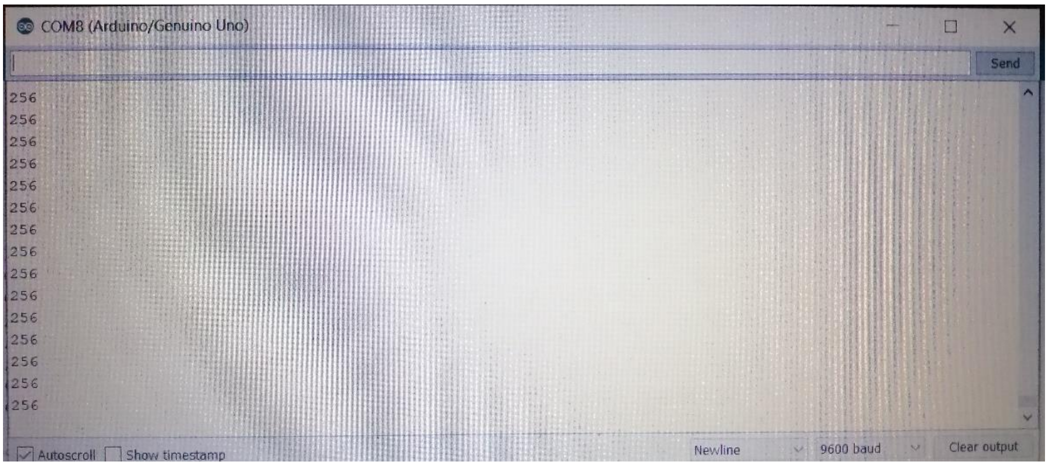
```
321
321
321
321
321
321
321
321
321
322
321
322
321
321
321
```

☑ Autoscroll ☐ Show timestamp Newline 9600 baud

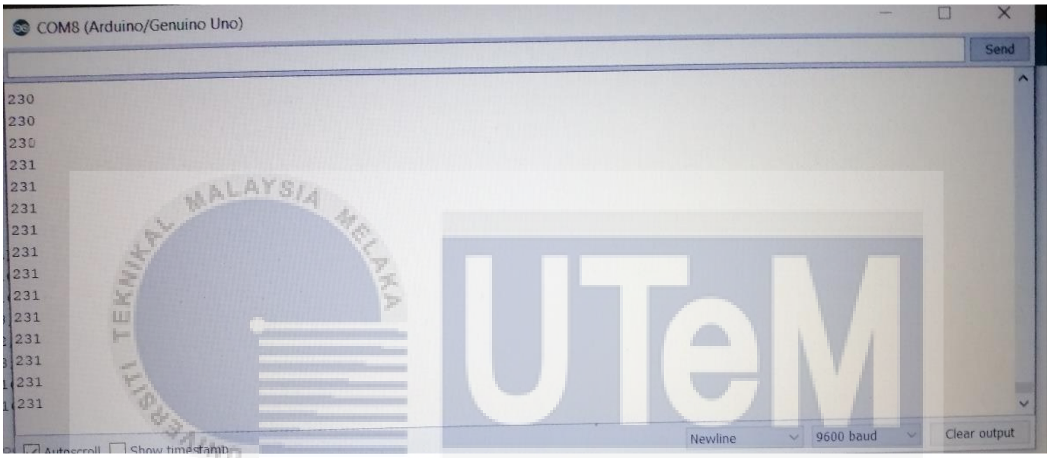


```
306
306
305
305
305
306
306
306
305
305
306
305
306
306
305
```

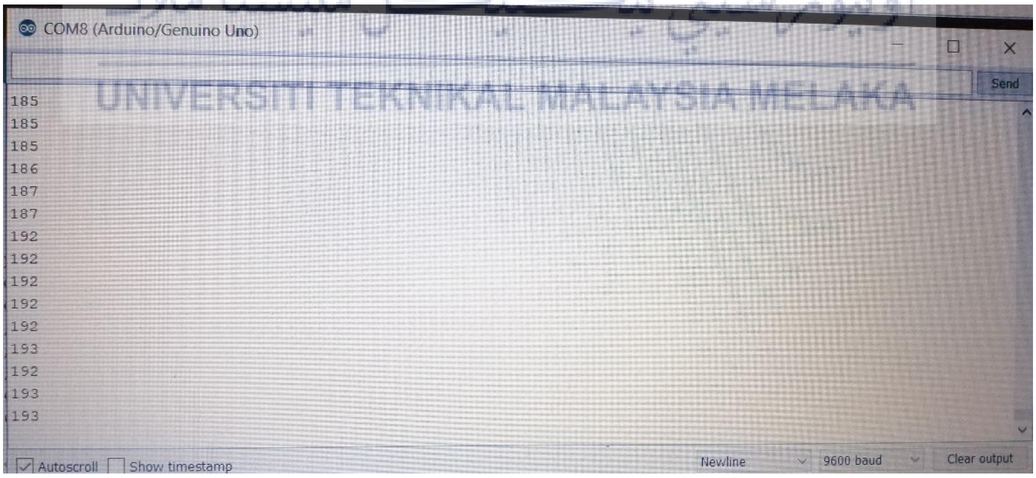
☑ Autoscroll ☐ Show timestamp Newline 9600 baud



```
256
256
256
256
256
256
256
256
256
256
256
256
256
256
256
```



```
230
230
230
231
231
231
231
231
231
231
231
231
231
231
231
```



```
185
185
185
186
187
187
192
192
192
192
192
193
192
193
193
```