DEPARTMENT OF COMPUTER AND COMMUNICATION SYSTEM ENGINEERING
FACULTY OF ENGINEERING
UNIVERSITI PUTRA MALAYSIA
43400 UPM SERDANG
SELANGOR

IGARD (Intelligent (AIoT) Smart Garden) Robot

FINAL REPORT PROJECT

PREPARED FOR:

DR. FAISUL ARIF AHMAD

SUBMISSION DATE: - 10th JULY 2021	
PREPARED BY: GROUP 3	
NAME	MATRIC NO.
MUHAMMAD ZAKI BIN LUHUR BAMBANG SUBIANTORO	198772
SHABIL IZDIHAR BIN AHMAD AZMI	197812
AMIR HAKIM BIN AMIR SHARIFUDDIN	198686
NURUL AIN BINTI ABDUL AZIZ	198285

TABLE OF CONTENTS

ABSTRACT

This project was created to provide gardeners the ability to compute and comprehend environmental indicators. Hardware includes an Android phone, an Arduino Uno microcontroller, two 3.7-volt 13000mA lead Li-Ion batteries linked in series, an ESP32 camera module, an extension board V5, a pH sensor, a soil moisture sensor, a Bluetooth module, a robot tyre, a servo motor, a DC motor, and a driver motor. The systems can be used with an Android phone that has the IGARD app downloaded. MIT App Inventor was used to create the IGARD Application. Bluetooth was set as a controller to operate the IGARDRobot. This paper demonstrates how, using an Arduino Uno and wireless technology, we can observe circuit diagrams and display the results using the IGARD App. IGARDRobot is equipped with a camera and sends the data to mobile phones through Bluetooth network. The robot has portable sprinkler and servo motor for watering the farm. It works with humidity and pH sensors. The highlighting features of this project include real-time image sharing and sensors for measuring moisture and pH of the soil. A message is sent between Arduino Uno and IGARD App when it detects low moisture and warm temperature, or when the soil is too acidic or alkaline, and it manually begins the motor operated by the gardener/farmer in home gardening, farm, etc.

1. INTRODUCTION

Agriculture is one of the important sectors that accounts for growth in the developing world. Due to the increase demand of food, additional efforts and special techniques are being developed to multiply food production. Modern agriculture requires a larger production of food to meet the needs of the great global population. To achieve this goal, new technologies and solutions are being applied in agriculture to provide an optimal alternative for collecting and processing information to improve productivity. In addition, inefficient energy consumption, high cost, time consuming problems demand new and improved methods for modern agricultural exploitations. The need for automation then appears, and intelligent decision making is becoming more important to achieve this goal. In this sense, technologies such as quantum computing, Wireless Network Sensors (WNS), radio frequency modules (RFID), cloud computing, Internet of Things (IoT), satellite monitoring, and remote sensing are becoming increasing popular.

The rise of wireless sensor networks has stimulated a new direction in agriculture. Recently, WSN have been widely applied in various agricultural applications and were decide to embedded its features into the project IGARDRobot along with the AIoT. The purpose of IGARDRobot is to help urban gardeners monitor plants growth and its necessities such as soil humidity and pH value of the soil for optimum plant growth. Furthermore, IGARDRobot can reduce the cost of remote gardening system with an affordable price for all range of users. Moreover, the most important aspect of IGARDRobot is to reduce human force in agriculture monitoring.

2. LITERATURE REVIEW

General sensor based close loop control is a system where one or more behaviours or states are sensed, and a controller uses those sensor readings to adjust/maintain a system to a desired state. It involves a few components including a measuring element, controller, and process. A diagram of a closed loop system is shown below in Figure 1.

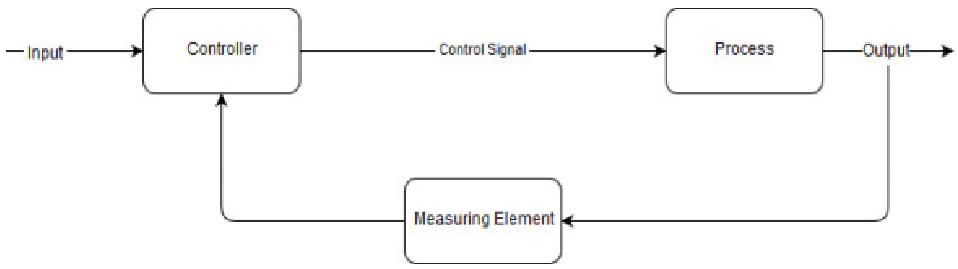


Figure 1 Closed loop system

In Figure 1 above, a controller uses data from a measuring element to control a repetitive process where a constant state is attempted to be achieved/maintained via a process controlled by the controller with a signal. The measuring element/sensor is used to provide an input to the controller and to measure the output of the process so the controller can perhaps fine tune the process to help achieve the desired state of the system. The controls/sensors can be combined together with actuators to be used in a variety of applications in the area of feedback control systems (Gauger, Minder, Marron, Wacker, & Lachenmann, 2008). The feedback control systems can be then used to perform actions resulting in being able to minimize resource or energy usage, instead of earlier home and garden automation systems where sensor inputs were available from the system, but nothing could be done with that information automatically since output/actuator options were minimal [1].

Present IOT based smart garden monitoring system which sense the requirement of the plant and provide it with water as the soil loses its moisture. Different soils have different fertility and moisture level so we have soil and moisture sensor used in this to detect this problem. In our country there are six different seasons and each day have different temperature and humidity level so to check the temperature and humidity for the better health and survival of plant temperature and humidity sensor are used which regularly sends data to the server. In this way it manages to perform its operations automatically [2].

A design for home automation system using ready-to-use, cost effective and energy efficient devices including raspberry pi, Arduino microcontrollers, xbee modules and relay boards was proposed. Arduino microcontrollers are used to receive the on/off commands from the raspberry pi using ZigBee protocol. The use of ultrasound sensors and solenoid valves make a smart drip irrigation system.

A control design of wireless sensor network system based on radio frequency (RF) transceiver for greenhouse, which consists of some sensor nodes placed in the greenhouse and a master node connected to upper PC in the monitoring centre. The sensor nodes collect signals of greenhouse temperature, humidity, light and CO_2, control the actuators, and transmit the data through the wireless RF transceiver; the master node receives the data through the RF transceiver and sends the data to the upper PC for real-time monitoring [3].

Irrigation system is a method of allowing water to drip slowly to the roots of plants, either onto the soil surface or directly onto the root zone, through solenoid valve. Thus, this paper proposes a design for smart home garden irrigation system that implements ready-to-use, energy-efficient, and cost-effective devices. Raspberry Pi, which is implemented in this system, is integrated with multi-sensors such as soil moisture sensors, ultrasonic sensors, and light sensors. This proposed system managed to reduce cost, minimize waste water, and reduce physical human interface. It is conducted with GUI using Android application to activate watering activity [4].

The sensors can be specialized in measuring temperature, humidity, pressure, light, noise, dust air, and so on. In this paper, a solution to transform a normal house in a smart house while reducing the energy consumption is proposed [5].

Android software is used to create mobile applications which are used to monitor the parameters of the garden and automate the watering process. ESP32 and Arduino is used to connect different sensors which collect the parameters of soil and transmits the information to through inbuilt Bluetooth [6].

Today's fast-paced society has made water scarcity one of the world's most significant problems, and farming consumes a significant amount of it. It is therefore necessary to develop an effective method of utilising the available water. Automated systems rely on the Internet of Things, which aims to connect the greatest number of devices possible over the internet. All around us are plants and trees that provide oxygen, nutrition, and other essentials to our survival [7].

Using the Arduino microcontroller, this moisture-based garden maintenance system can be easily set up and maintained. This system makes use of sensors such as the Humidity sensor and the Temperature sensor, which allows for the addition of intelligence to the system. Various places on the soil are used to measure the humidity and temperature of the soil at regular intervals by the system. When the application requests it, this information is delivered to it. Depending on the location, there may be multiple sensors. Additionally, this system offers automated watering based on the moisture content of the soil, in addition to manual and time-based management options. In accordance with the plant profile, a threshold value is established [8].

Internet of things (IOT) means interconnecting the various devices by the internet. Without human-to-human interaction data can be transferred through single identifier. Every object is connected. The cloud server stores the data for decision making. The green house systems use the Arduino technology to control the watering. The system performance and storage they use IOT communication technology and cloud server. Proposed system provides remote monitoring. It controls real-time operated sensing system of atmosphere and conditions of the soil like temperature of air, humidity level and moisture soil content of the environment [9].

The Arduino board and soil moisture sensor-based irrigation system proves to be a real time response control system which monitors and wheel all the activities of irrigation system. The present system is a model to modernize the agriculture industries at a mass scale with optimum expenditure. An automated irrigation model is designed and implemented this model considering low cost, reliability, and automatic control. As the proposed model is automatically controlled it will help the farmers to properly irrigate their fields. The model always ensures the sufficient level of water in different fields avoiding the under-irrigation and over-irrigation [10].

3. MATERIALS AND METHOD

3.1 Project Block Diagram

The software involved was Arduino Integrated Environment, TinkerCad web and MIT App Inventor. The hardware was programmed using the Arduino IDE. Tinkercad was used for designing the circuit of the project. To make an Android app, we used MIT App Inventor. The following hardware was included in block diagram as shown in Figure 1: android phone, the Arduino Uno microcontroller, two 3.7-volt 13000mA lead Li-Ion batteries connected in series, ESP32 camera module, expansion board V5, pH sensor, soil moisture sensor, Bluetooth module, robot tyre, servo motor, DC motor and driver motor. it shows the block diagram which consists of inputs, processor, and outputs that will be implement into the IGARD robot system.

This proposed design uses multiple sensors as inputs such as pH sensor, humidity sensor and timer. The processor use is Arduino Uno, and the outputs are LED as an indicator, water sprinkler as feedback to maintain plant's humidity, display to show the output of the camera through mobile phone. The mobile phone use to control the IGARD robot for monitoring purpose. Bluetooth module is used for the interconnection between the mobile phone and the IGARD robot.

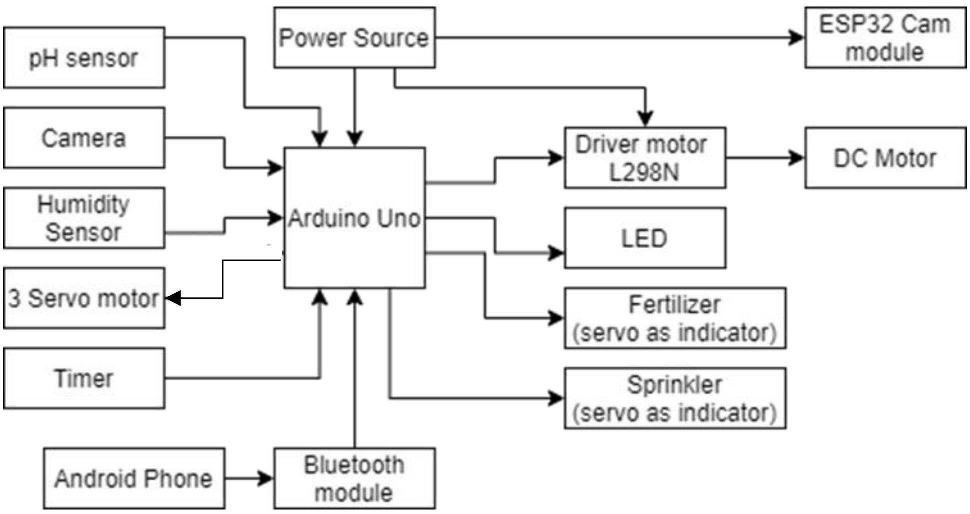
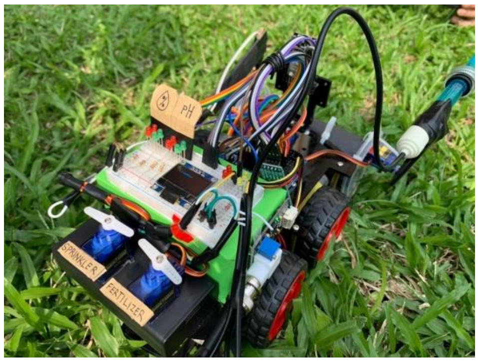


Figure 2 Block Diagram

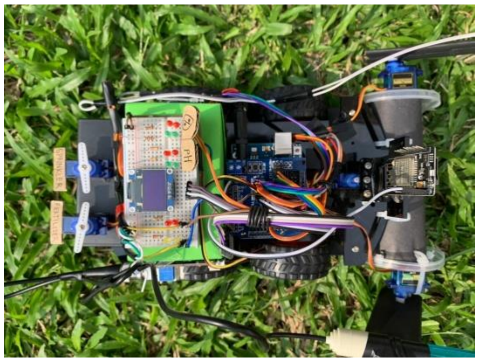
The Arduino Uno is a programmable microcontroller that controls the hardware. The total voltage of the two 3.7volts batteries connected in series is 7.4volts. It was in charge of the blade motor. The Android phone was connected to the Arduino Uno using the HC-05 Bluetooth module. It has a unique Bluetooth address that must be associated with an Android phone to use the IGARD app. Intelligent Smart Garden System was the inspiration for the term IGARD. The IGARD Bluetooth Module is placed to the top of the robot so that it may be readily modified or checked for functionality. Researchers can simply see whether Bluetooth is blinking or working when the android phone is detached/disconnected from the robot [1]. The motor driver is connected between the 1 acrylic Chassis as Figure 2. Meanwhile, battery as power supply was installed in the back and top acrylic Chassis of the robot by the researchers because it is a simple way to unmount the battery to charge.

A) Side view of IGARDRobot prototype.



C) Back view of IGARDRobot prototype.

B) Front view of IGARDRobot prototype.



D) Top view of IGARDRobot pro

Figure 3 IGARDRobot Prototype

In this project as shows in prototype view above, we have used 5 servo motors, where 3 from it were implemented to control movement pH sensor, soil moisture sensor, and ESP32 Camera Module at front of robot. The rest were set as indicator at the back side of the robot to show and as proof the output fertilizer and water sprinkler activated. On the other hand, green LED was used to indicate the soil is alkaline and red LED would indicate the acidic soil. For optimum soil moisture also indicated by another green LED that have placed separately from pH indicator.

3.2 Related Work and Project Flow in IGARD System

The following tests and tasks were performed after all of the hardware had been installed:

3.2.1 Soil pH Test

IGARDRobot has used the liquid pH sensor as in Figure 4 that represents the soil pH sensor to measure the soil pH value, either it is a suitable soil condition for plants or not. The liquid pH sensor was chosen for this prototype due to its lower price, whereas the actual soil pH sensor was costly. Even though the liquid sensor was used, it still has the same function as the soil pH sensor, which can measure the pH value of substances. The only difference is the material or substances that need to be tested besides the same measurement concept. The pH sensor measured the soil's pH value then IGARDRobot will react to the pH measurement value either it is acidic, neutral, or alkaline condition. The fertilizer pump will release its ingredients to the soil when IGARDRobot detects acid and alkaline pH values.

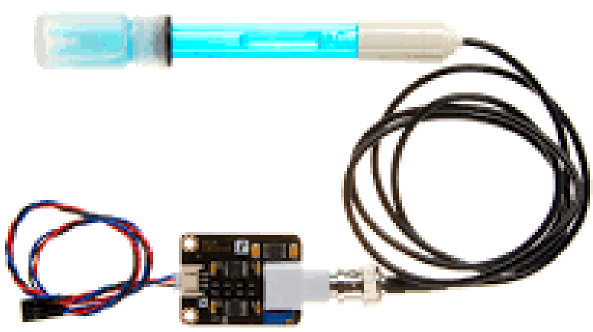


Figure 4 Arduino liquid pH sensor module.

3.2.2 Soil Moisture Test

Soil Moisture sensor that shows in Figure 5 have been used in this project where it is the cheapest module and simple to use. This is used to keep track of soil moisture levels. The dielectric permittivity of the surrounding medium is measured using a capacitance. The capacity of the dielectric permittivity in the soil is the water content. The sensor's voltage is proportional to the dielectric permittivity and the water content of the soil [2].

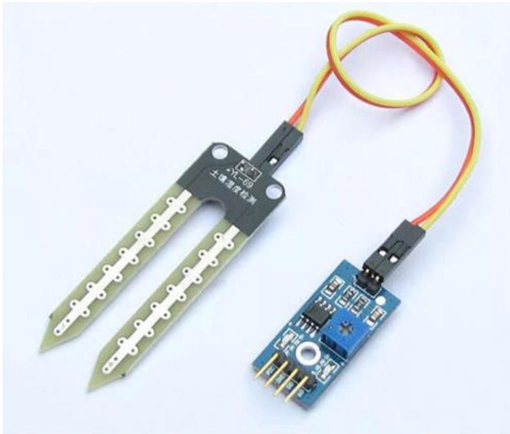


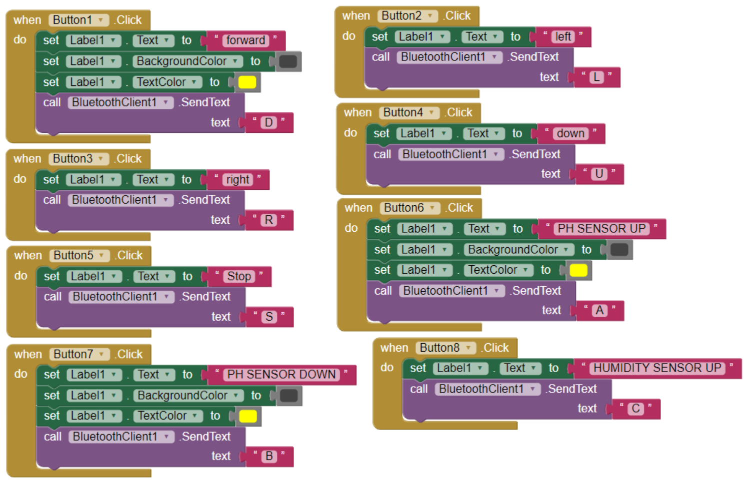
Figure 5 Arduino soil humidity sensor module.

3.2.3 IGARD Bluetooth Application Development

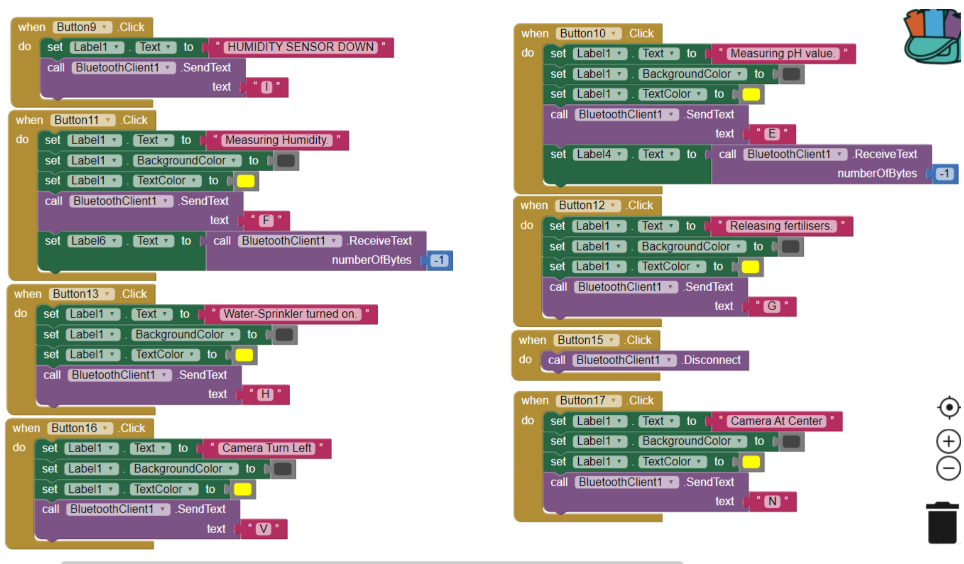
In this project, IGARDRobot app made in MIT APP Inventor was used to control the robot with a Bluetooth connection. The backend and frontend of the app were made using the blocks below through the MIT APP Inventor. Figure 6 A) shows the ListPicker blocks which functions as the Bluetooth connector between the app and the robot. The clock block will detect if the Bluetooth is connected and display a green “connected” on the screen while it will display a red “not connected” if the Bluetooth fail to connect. It also allows the connection to the web viewer. Figure 6 B) and C) shows the blocks for buttons that functions vary such as the movement of the robot, camera angles, movement of the sensors and Bluetooth connection. Lastly, figure 6 D) shows a button block and the Global URL initializer that makes the live camera display posible.


when ListPicker1 .BeforePicking
do set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames
when ListPicker1 .AfterPicking
do set ListPicker1 . Selection to call BluetoothClient1 .Connect
address ListPicker1 . Selection
when Clock1 .Timer
do if BluetoothClient1 . IsConnected
then set Label2 . Text to " Connected! "
set Label2 . TextColor to
set WebViewer1 . HomeUrl to join get global URL
" :81/stream "
if not BluetoothClient1 . IsConnected
then set Label2 . Text to " Not Connected! "
set Label2 . TextColor to

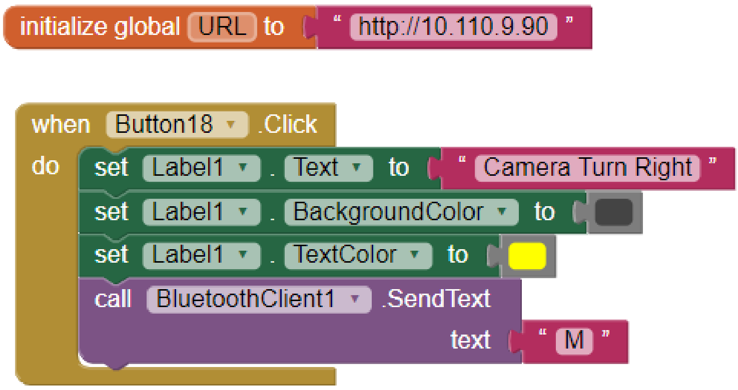
A)


when Button1 .Click
do set Label1 . Text to " forward "
set Label1 . BackgroundColor to
set Label1 . TextColor to
call BluetoothClient1 .SendText
text " D "
when Button2 .Click
do set Label1 . Text to " left "
call BluetoothClient1 .SendText
text " L "
when Button3 .Click
do set Label1 . Text to " right "
call BluetoothClient1 .SendText
text " R "
when Button4 .Click
do set Label1 . Text to " down "
call BluetoothClient1 .SendText
text " U "
when Button5 .Click
do set Label1 . Text to " Stop "
call BluetoothClient1 .SendText
text " S "
when Button6 .Click
do set Label1 . Text to " PH SENSOR UP "
set Label1 . BackgroundColor to
set Label1 . TextColor to
call BluetoothClient1 .SendText
text " A "
when Button7 .Click
do set Label1 . Text to " PH SENSOR DOWN "
set Label1 . BackgroundColor to
set Label1 . TextColor to
call BluetoothClient1 .SendText
text " B "
when Button8 .Click
do set Label1 . Text to " HUMIDITY SENSOR UP "
call BluetoothClient1 .SendText
text " C "

B)


when Button9 .Click
do set Label1 . Text to " HUMIDITY SENSOR DOWN "
call BluetoothClient1 .SendText
text " I "
when Button10 .Click
do set Label1 . Text to " Measuring pH value "
set Label1 . BackgroundColor to
set Label1 . TextColor to
call BluetoothClient1 .SendText
text " E "
set Label4 . Text to call BluetoothClient1 .ReceiveText
numberOfBytes -1
when Button11 .Click
do set Label1 . Text to " Measuring Humidity "
set Label1 . BackgroundColor to
set Label1 . TextColor to
call BluetoothClient1 .SendText
text " F "
set Label6 . Text to call BluetoothClient1 .ReceiveText
numberOfBytes -1
when Button12 .Click
do set Label1 . Text to " Releasing fertilisers "
set Label1 . BackgroundColor to
set Label1 . TextColor to
call BluetoothClient1 .SendText
text " G "
when Button13 .Click
do set Label1 . Text to " Water-Sprinkler turned on "
set Label1 . BackgroundColor to
set Label1 . TextColor to
call BluetoothClient1 .SendText
text " H "
when Button15 .Click
do call BluetoothClient1 .Disconnect
when Button16 .Click
do set Label1 . Text to " Camera Turn Left "
set Label1 . BackgroundColor to
set Label1 . TextColor to
call BluetoothClient1 .SendText
text " V "
when Button17 .Click
do set Label1 . Text to " Camera At Center "
set Label1 . BackgroundColor to
set Label1 . TextColor to
call BluetoothClient1 .SendText
text " N "

C)



D)

Figure 6: A) ListPicker (bluetooth connection) and Clock blocks. B) Button 1 – Button 8 block. C) Button 9 – Button 17 blocks. D) Button 18 and Global URL

3.2.4. Flow Chart IGARDRobot System

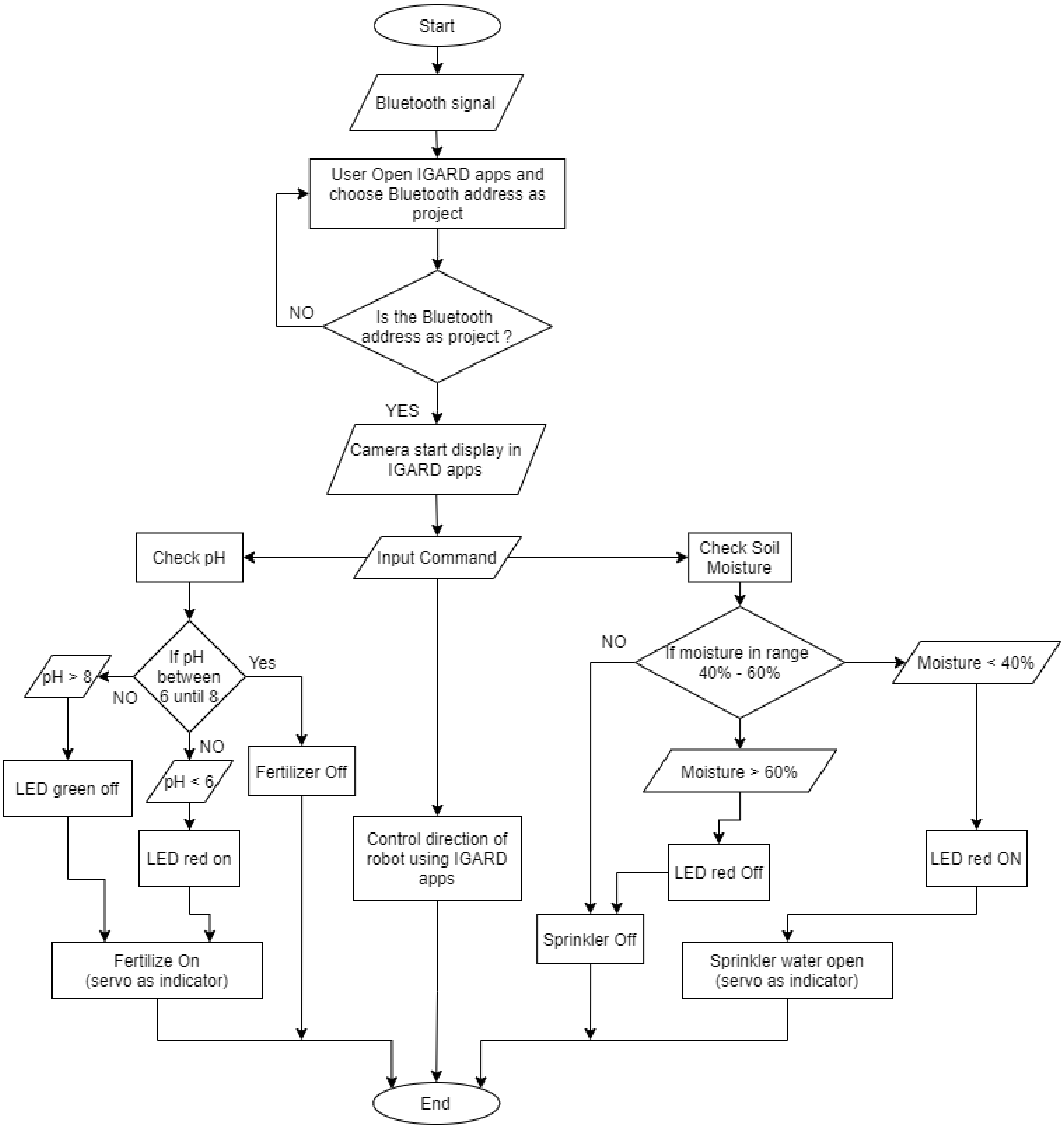


Figure 7: Flowchart in IGARDRobot system

Figure 7 shows the IGARDRobot system flowchart. The system starts with the user open the application and Bluetooth via mobile phone. If the system is connected, the camera will be open and display the IGARD robot's view. Then the system will wait for the input command from the user to execute specific tasks. There are three task that available for the user to control.

The first one is the movement of the robot in the multiple directions. Second, the user can check the percentage of the soil humidity. If the sensor read that the percentage is out of range 40 to 60 percent, the water sprinkler will be activated. Last but not least, the user can measure the pH of soil by using pH sensor. If the pH is not in between 6 until 8 which is neutral level for soil, LED colour will turn on depends on range of pH. Then, the system will neutralize it by using fertilizer.

4. RESULT AND DISCUSSION

4.1 IGARD Robot Specifications

4.1.1 Bluetooth Application Control with Mobile Display (IGARD App)

Figure 8: IGARDRobot controlled by using IGARD Bluetooth Application.

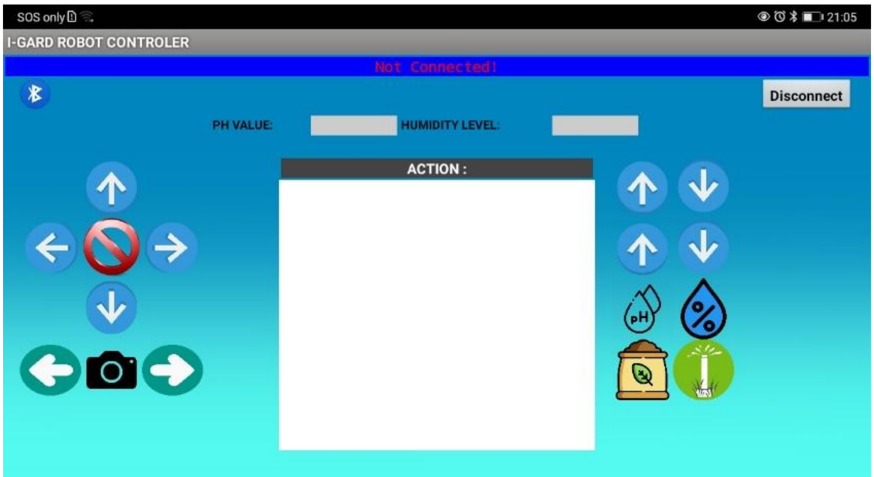
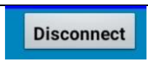
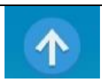
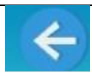
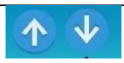


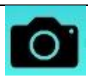
Figure 9: IGARD Bluetooth Application interface.

Figure 10: The HC06 Bluetooth Module.

IGARDRobot is a robot that can be controlled from a distance using a remote control device (smartphone application in Figure 9) and connected via Bluetooth (shown in Figure 10 above). The Bluetooth module used by IGARDRobot is HC-06. The HC-06 Bluetooth module connects two microcontrollers or systems through wireless over a distance, allowing them to interact with each other. In this case, the two devices are remote controllers, the smartphone and the prototype itself via IGARD Bluetooth application. The module communicates through Bluetooth 2.0 and is only capable of acting as a slave device. This module is the least expensive and most flexible method of wireless data transfer, capable of transmitting files at up to 2.1Mb/s. The data is sent to the module or received from the module through the UART interface. Therefore, the user can connect the module to any microcontroller which has an RS232 port.

Table 1: Button Configuration for IGARDRobot Application.

Button/Display	Char Symbol	Function/Action
	-	Bluetooth Connection
	-	Bluetooth Disconnect
	D	Forward
	R	Turn Right
	L	Turn Left
	U	Reverse
	S	Stop
	A & B	Turn Up and Down Right Servo Motor
	C & I	Turn Up and Down Left Servo Motor
	E	Measure Soil pH Value

	F	Measure Soil Humidity Percentage
	H	Manually Turn on Water Sprinkler Pump
	G	Manually Turn on Fertilizer Pump
	V	Turn Left Camera
	N	Set Camera at Center
	M	Turn Right Camera
	-	Camera Display
	-	PH Value Display
	-	Humidity Level Display

As shown in Table 1 above, the Bluetooth transmit the data sent by the user from the IGARD application, and the HC06 Bluetooth module received the data and sent it to the Arduino board to run the assigned function. IGARDRobot used specific char data types for each button to indicate the particular signal for running each assigned task. For example, when a button symbol of soil humidity is pressed, it sent the char F data type signal to the HC06 Bluetooth module on IGARDRobot then run/start its assigned char function to begin measuring soil humidity.

4.1.2 Robot View with ESP32 Wi-fi Camera Module

Figure 11: ESP32 Wi-fi Camera Module implemented on the IGARDRobot.

IGARDRobot is a remote-control robot which mean the user will not be in the same place with it. This is the reason why IGARDRobot is equipped with a Wi-Fi camera that will use by the user to be the robot's eye and control it. Camera module used by IGARDRobot is ESP32-CAM as shown in Figure 11. The ESP32-CAM is a feature-rich microcontroller that has an integrated video camera and a microSD card reader. It is low-cost and simple to use, making it ideal for IoT devices such as IGARDRobot. In this case, the camera will send the output data to the web browser URL. Then, the IGARDRobot application will connect to the web browser via Wi-Fi and display the output on the phone screen through this apps as in Figure 12.

The ESP32-CAM module was set and by upload the source code built with specific Wi-fi IP address and host name, thus it can automatically connect to the desired Wi-fi. For the IGARDRobot prototype, it was set with IP address of “Hotspot@PUTRA-2.4GHz” that provided by Universiti Putra Malaysia. The URL was set to http://10.110.8.126/ that use for streaming the camera display.

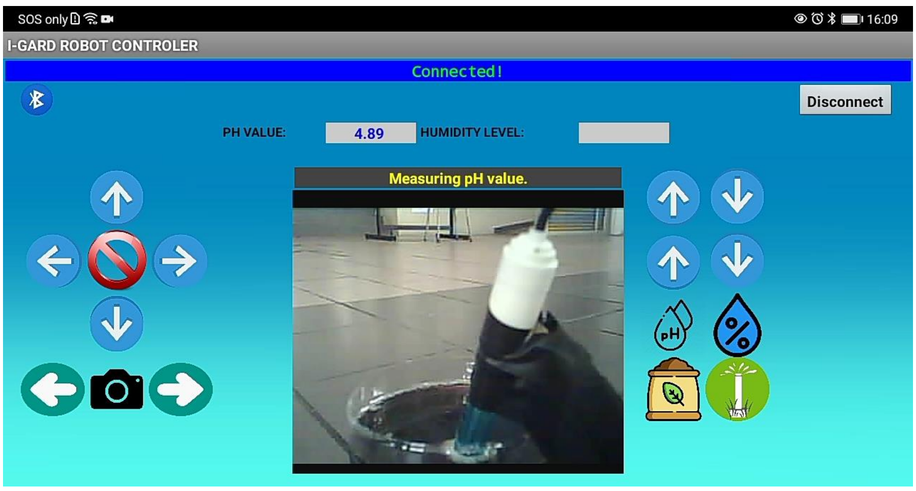


Figure 12: Example of camera view in IGARD App.

4.1.3 Robot Movement with DC Motor and L298N Motor Driver Module

The L298N Motor Driver was interfaced with Arduino UNO to control 4 DC motors that used for IGARDRobot's movement. It controlled both speed with Pulse Width Modulation (PWM) and spinning direction of the DC motors with H-Bridge. A DC motor's speed is regulated by changing its input voltage. PWM is a method for adjusting the average value of the input voltage through a sequence of ON-OFF (HIGH/LOW) pulses. The average voltage is proportional to the duty cycle of the pulses. The duty cycle setting determines the average voltage applied to the DC motor (High Speed) and the duty cycle setting determines the average voltage applied to the DC motor (Low Speed) [11].

Besides that, the H-Bridge was used to regulate the direction of rotation of DC motors. The polarity of the DC motor's input voltage is used to regulate its direction of rotation. An H-Bridge circuit consists of four switches arranged in an H-shape with the motor in the middle. By simultaneously closing two switches, the polarity of the voltage supplied to the motor is reversed. This results in a reversal of the motor's spin direction [11]. It determined whether the motor spun forward or backward by using the direction control pins. These pins are used to regulate the H-Bridge circuitry included inside the L298N IC's H-Bridge switches.

Each channel has two direction control pins. The IN1 and IN2 pins control the direction in which the motor A spins (LB and LF), whereas the IN3 and IN4 pins control the direction in which the motor B spins (RB and RF). The connection of motor A (which is LB and LF motors) were connected to pins OUT1 and OUT2 respectively. Whereas, the connection of motor B (which is RB and RF motors) were connected to pins OUT3 and OUT4 respectively. These inputs were used to regulate the speed of rotation of a motor by delivering either a logic HIGH (5 Volts) or a logic LOW (Ground) signal. It was done based on Table 2 and Table 3 below:

Table 2: Description of input condition for each DC motor.

<table>
<tr><th>Motor</th><th>Input</th><th>Input Condition</th><th>Indicator</th></tr>
<tr><td rowspan="4">Motor A</td><td rowspan="2">IN1 (LB)</td><td>HIGH</td><td>Rotate anticlockwise</td></tr>
<tr><td>LOW</td><td>Rotate clockwise</td></tr>
<tr><td rowspan="2">IN2 (LF)</td><td>HIGH</td><td>Rotate clockwise</td></tr>
<tr><td>LOW</td><td>Rotate anticlockwise</td></tr>
<tr><td rowspan="4">Motor B</td><td rowspan="2">IN3 (RB)</td><td>HIGH</td><td>Rotate clockwise</td></tr>
<tr><td>LOW</td><td>Rotate anticlockwise</td></tr>
<tr><td rowspan="2">IN4 (RF)</td><td>HIGH</td><td>Rotate anticlockwise</td></tr>
<tr><td>LOW</td><td>Rotate clockwise</td></tr>
</table>

Table 3: Truth Table for DC motors' input.

IN1 (LB)	**IN2 (LF)**	**IN3 (RB)**	**IN4 (RF)**	**Output**
HIGH	LOW	HIGH	LOW	Robot move forward
HIGH	LOW	LOW	HIGH	Robot move to right
LOW	HIGH	LOW	HIGH	Robot move backward (reverse)
LOW	HIGH	HIGH	LOW	Robot move to left
HIGH	HIGH	HIGH	HIGH	Robot is stopped

Speed control pins ENA (control motor A's speed) and ENB (control motor B's speed) were used to switch the motors ON, OFF, and to regulate their speed. Pulling these pins HIGH causes the motors to rotate, while pulling them LOW causes them to stop. However, using PWM, it is possible to regulate the speed of the DC motors. On these pins, the L298N DC Motor Driver module had a jumper. When this jumper is connected, the motor is activated and rotates at full speed. Thus, to programmatically control the speed of motors, the jumpers were removed and the motors were linked to PWM-enabled pins on the Arduino.

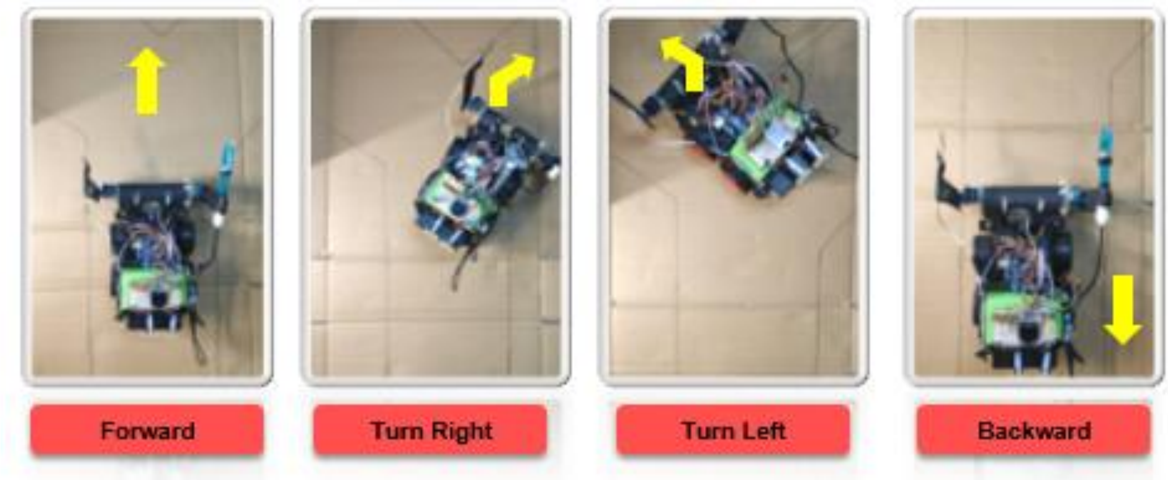
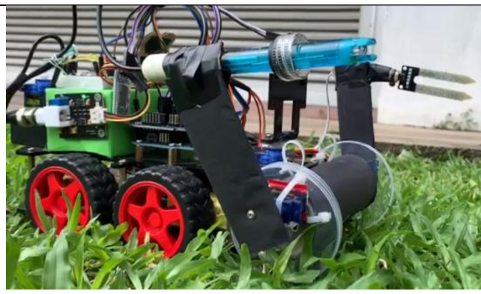
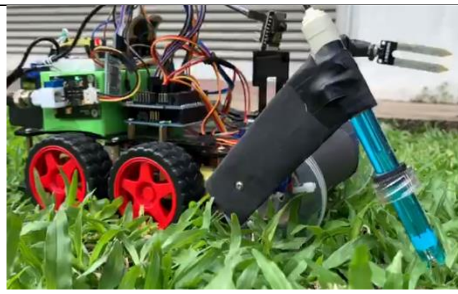


Figure 13: Movements of IGARDRobot.

4.1.4 Specific Angle Movement with Servo Motor

Table 4: Angle movement control with servo motor.

Servo Motor	Function	Position (Angle)	Output
Right Arm Servo	To control pH sensor	Up (90°)	
		Down (0°)	
Left Arm Servo	To control soil humidity sensor	Up (0°)	

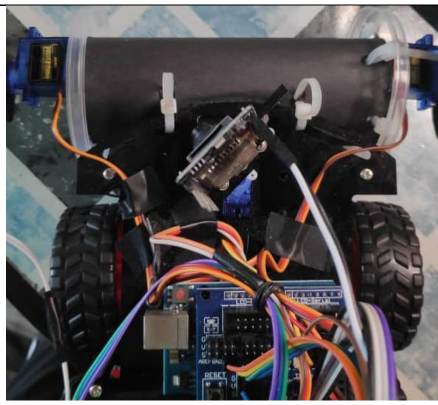
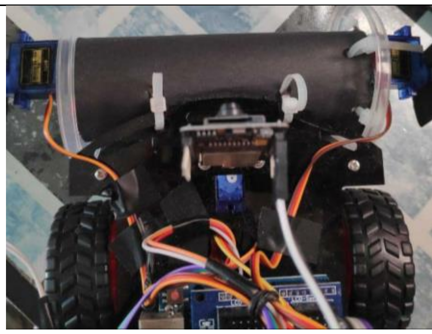
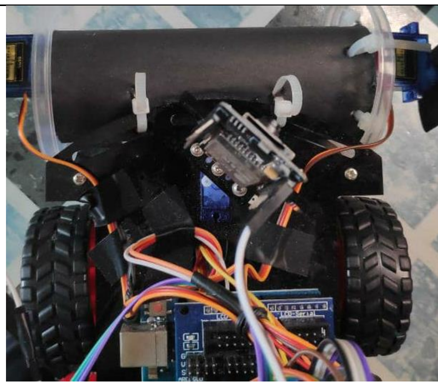
		Down (90°)	
Center Servo	To control ESP32 Wi-fi Camera Module view	Left (140°)	
		Center (90°)	
		Right (50°)	

Table 4 above shows the IGARDRobot's arms and camera module movement control by the servo motor. The servo motors were set into a specific angle to get the resulting function. First, the right arm was used to control the pH sensor during soil pH measurement. The right arm turned up when IGARDRobot does not do the measurement while turning down when the measurement needs to begin, which the user can control itself. The angle of this servo motor was set to 90 degrees for the up position, whereas 0 degrees for the down position. The speed of the servo motor was controlled to make sure the pH sensor did not too fast go down by putting the delay, hence prevent the substances from splash.

Second, when measuring soil humidity, the left arm was utilised to operate the humidity sensor. When the humidity sensor was not in use, the left arm was raised (set to 0 degrees), whereas it was lowered (set to 90 degrees) when the sensor was required to measure soil humidity. Additionally, it was configured to a considerable servo motor speed when the sensor is turned down to enable it to dig into the soil.

Lastly, the ESP32 Wi-Fi Camera module's view was held and controlled using the center servo motor. This provided the user with a wider perspective of their garden. The camera initially centred on the 90 degrees set by the servo motor, and it may swivel left (set at 140 degrees) to display the left side of IGARDRobot, in addition to the user monitoring the left arm control. Additionally, by adjusting the servo motor to 50 degrees, the camera may display the robot's right side. This enabled the user to exert control over their right arm, which was holding the pH sensor. The following illustrates the right and left arms' servo motor control coding function, which enables the rotational speed to be adjusted to the task's requirements:

```
void svR_down()//Right servo down
  {    int angleR = 90;
       for(angleR = 90; angleR > 0; angleR -= 1)    // move slow from 0 degrees to 90 degrees
            { myservoR.write(angleR);               // rotate the servo to the specified angle
              delay(100);
    }
}
void svL_down()// Left servo down
  {    int angleL = 0;
       for(angleL = 0; angleL < 90; angleL += 10)    // move fast from 0 to 180 degrees
    {
      myservoL.write(angleL);               // rotate the servo to the specified angle
      delay(100);
    }
```

4.2.5 Automatic Soil PH Sensor Measurement System

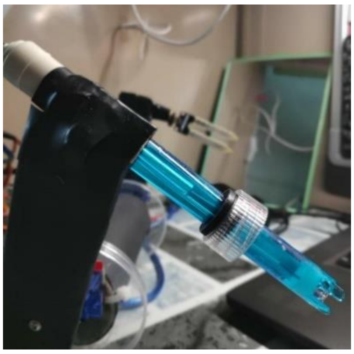
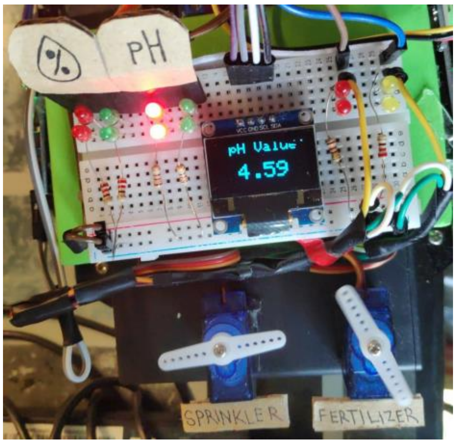
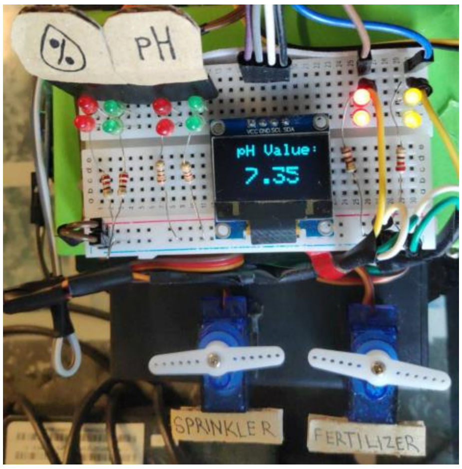
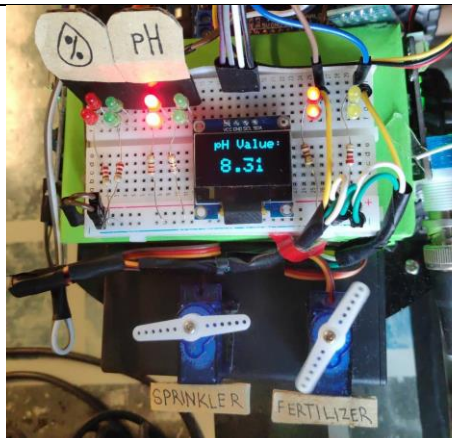


Figure 14: Liquid pH Sensor used on IGARDRobot prototype.

Table 5: Output of the pH sensor based on soil condition.

Soil Condition (pH Value)	Red LED	Fertilizer Pump	Example Output
Acidic (<6)	HIGH	HIGH	
Optimum/Neutral (6<pH<8)	LOW	LOW	

Alkaline (>8)	HIGH	HIGH	

Based on Table 5 above, the red LED is implemented as an indicator for the soil condition. Hence, when it is in optimum condition for the plant, nothing happens, but the LED will turn ON when the pH value is below 6 (represents acidic soil) and above 8 (represents alkaline soil). The OLED display was used to display the output of measurements on the IGARDRobot.

IGARDRobot prototype's pH sensor was tested with three different substances like Pepsi carbonated water, which acted as acid and concentrated detergent acts as alkaline, besides mineral water that means for neutral condition. As shown in Table 5, the carbonated water has a pH value of 4.59 while 7.35 and 8.31 pH values for mineral water and concentrated detergent, respectively. When the pH value measured below 6 (means an acidic condition for soil), the fertilizer pump will put the basic fertilizer into the plant; because it can cause pH to become neutral and optimum condition as plant needed. It works the same for the pH value above 8 who more alkaline, and then the fertilizer pump will turn ON and sprinkle the acid-forming fertilizer into the plant, thus making soil pH more balance.

Consequently, the majority of plants grow on soils with a pH of between 6 and 7.5 (slightly acidic to slightly alkaline). [12]. Thus, the fertilizer type is used differently for each soils condition. The fertilizer has three types, first, to make soil more acidic which acidic or acid-forming fertilizers that lower mix pH value. Second, the optimum fertilizer, a third class, neutral or non-acid forming fertilizer, does not affect the soil's pH value. Third, to make the soil more alkaline, basic fertilizers can cause the mix pH to increase [13].

Figure 15: Testing of pH sensor using differences substances.

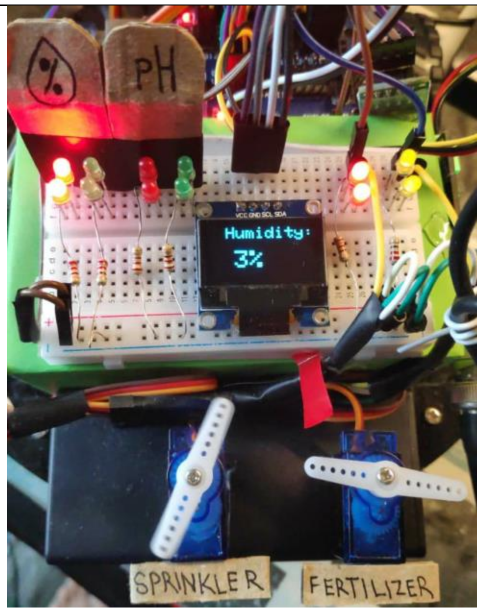
Below is the code used to tune the pH sensor to earn and display the best value of pH measurement:

```
void pH(){ //Function for measure soil pH
  for(int i=0;i<10;i++)       //Get 10 sample value from the sensor for smooth the value
  {
    buf[i]=analogRead(pH_pin);
    delay(10);
  }
  for(int i=0;i<9;i++)                          //sort the analog from small to large
  {
    for(int j=i+1;j<10;j++)
    {
      if(buf[i]>buf[j])
      {
        temp=buf[i];
        buf[i]=buf[j];
        buf[j]=temp;
      }
    }
  }
  avgValue=0;
  for(int i=2;i<8;i++)                          //take the average value of 6 center sample
    avgValue+=buf[i];
  phValue=(float)avgValue*5.0/1024/6;    //convert the analog into millivolt
  phValue=3.5*phValue;                   //convert the millivolt into pH value
  Serial.println(phValue);               //print the pH value
```

4.1.6 Automatic Soil Humidity Sensor Measurement System

Figure 16: Soil humidity sensor implemented on IGARDRobot protoype.

Table 6: Output of the humidity sensor based on soil condition.

Soil Condition (Humidity Percentage)	LED		Water Sprinkler Pump	Example Output
	Red LED	Green LED		
Too dry (<40)	HIGH	LOW	HIGH	

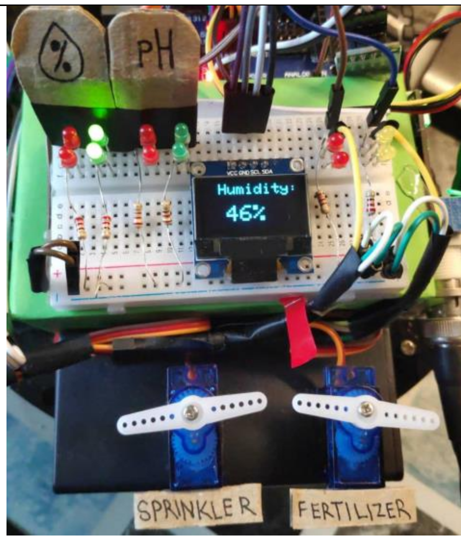
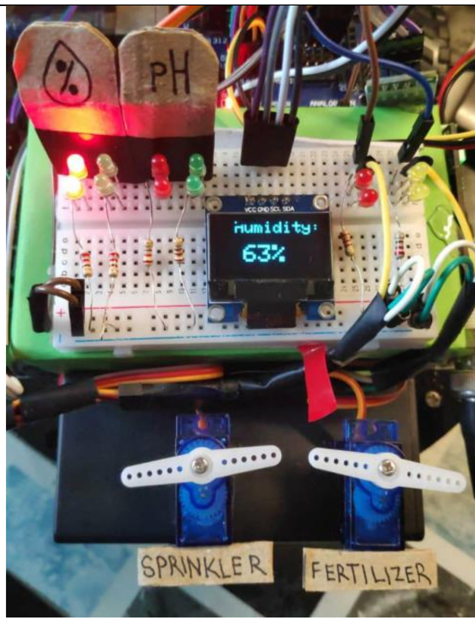
<table>
<tr><td>Optimum
(40<Humidity<60)</td><td>LOW</td><td>HIGH</td><td>LOW</td><td>
</td></tr>
<tr><td>Too wet (>60)</td><td>HIGH</td><td>LOW</td><td>LOW</td><td>
</td></tr>
</table>

Table 6 above shows the outputs of IGARDRobot's humidity sensor measurement based on different soil humidity levels. The soil humidity sensor shown in Figure 16 above was dip through the soil before starting the measurement. It has two probes that are used to determine the amount of water in the soil. The two probes conduct an electric current through the soil and, based on their resistance, determine the soil's moisture content [14]. When the soil has more water, it conducts electricity more efficiently, resulting in less resistance. As a result, the moisture content will be more significant. Conductivity is reduced in dry soil. As a consequence, when soil has less water, it conducts less electrical, resulting in an increase in resistance. As a consequence, the moisture content of the product will decrease.

When the measure begins, the sensor read the percentage of soil humidity then IGARDRobot indicated which condition and feedback should be triggered based on the sensor reading. If the humidity sensor reads the humidity below 30 percent means the soil was too dry, the OLED displayed the value, and the value also showed up in the IGARD Bluetooth Application, then the red LED was turned ON while the green LED was turned OFF. The red LED indicated the plant's soil is not in excellent or optimum condition for its growth. Then, the water sprinkler was triggered to be activated for sprinkling the water into the plant until the sensor read the humidity percentage above 40 percent.

By adding water to dry soil, shown in Table 6 above, the sensor read the value 46 percent of humidity level. When the humidity level was between 40 percent and 60 percent, the IGARDRobot determined that the soil was in optimal condition. The green LED will then illuminate to indicate this situation. Apart from that, if the sensor detects a humidity level of more than 60%, it suggests that the soil is excessively moist, causing the water sprinkler to be switched off and the red LED to illuminate, indicating that the plant's soil is not in good condition. Below is the code used to tune the humidity sensor to get the right value of humidity percentage:

```
void humidity(){ //Function for measure soil humidity
  //float moisture_percentage;
  int sensor_analog;
  sensor_analog = analogRead(humidity_pin);
  moisture_percentage = ( 100 - ( (sensor_analog/1023.00) * 100 ) );
  Serial.println(moisture_percentage);
```

An Analog to Digital Converter (ADC) is used to process the analog output of the soil moisture sensor. The serial monitor and IGARD App both show the moisture content in percentage form. The soil moisture sensor's output varies throughout the ADC value range of 0 to 1023. This may be expressed as a percentage moisture value using the formula below:

$$\text{AnalogOutput} = \text{ADCValue}/\ 1023$$

$$\text{Percentage moisture} = 100 - (\text{AnalogOutput} * 100).$$

It achieves a maximum value of 10-bit ADC, example, 1023, with zero moisture. This results in the moisture of 0%. Therefore, it increased the sensor accuracy.

5. PROJECT MANAGEMENT

5.1 Project personnel

Table 7: Roles and functions for every team member.

No.	Name	Roles	Function
1.	Muhammad Zaki	Project manager	Responsible for the completion of the project as planned. Lead role in the overall planning, executing, monitoring, controlling and closing of project.
2.	Shabil Izdihar	Integration and product lead	Responsible for the prototype's design and structure to make sure it is in line with the target market. Need to ensure that each component is in the appropriate place. Testing all the components such as sensors and wiring.
3.	Nurul Ain	Hardware and software lead	Responsible to lead the team in hardware installation and codding. This also includes the remote software for the prototype. Need to make the flowchart planning.
4.	Amir Hakim	Finance manager and project researcher	Responsible for the research that related to the project which is urban farming. Managing the financial health of the team and also responsible for providing an overview of the project.

5.2 Gant Chart

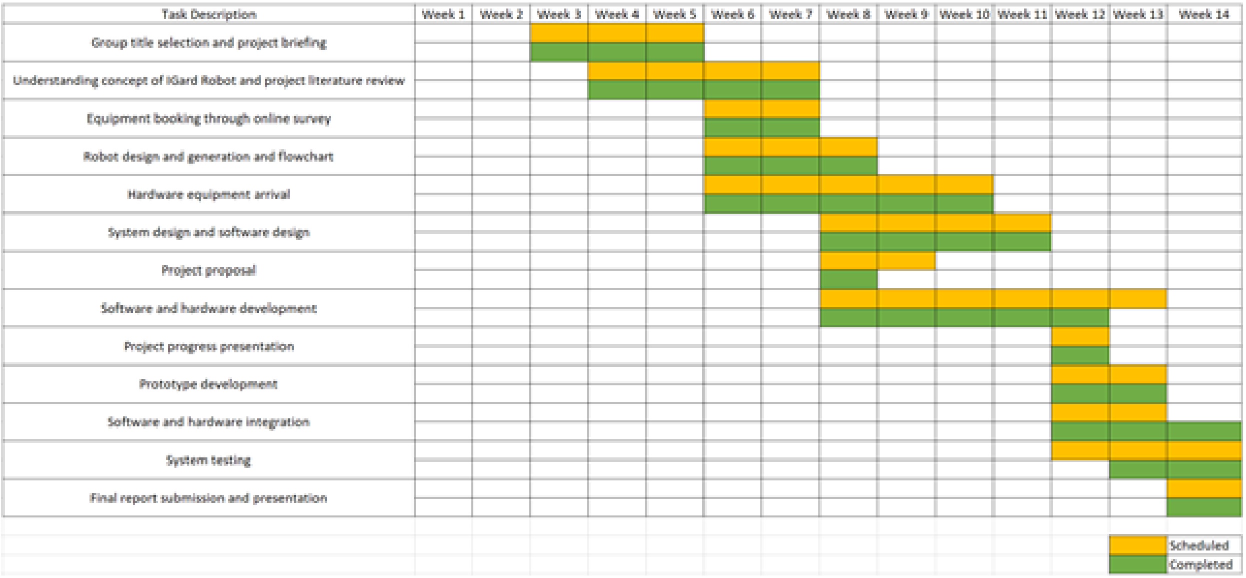


Figure 17: Gant chart for the project

This Gantt chart in Figure17 shows the time and duration for a task to be completed and it also shows the actual duration taken. the yellow bracket is the planned duration and the green color is the actual duration taken.

The first task was successfully completed as planned from the third to the fifth week. However, the second task has already run in the fourth week together with the first task, namely Understanding the concept of IGARDRobot and literature review project and it was also successfully completed as planned where it was in the seventh week. After that, three tasks were carried out simultaneously in the sixth week, namely Equipment booking through online survey, Robot design and generation and flowchart, Hardware equipment arrival and all were completed exactly in the specified week which is the seventh week, eighth week and tenth week. In the eighth week, three more new tasks have been carried out simultaneously, namely System design and software design, Project proposal and Software and hardware development. The project proposal was complete a week earlier, in the eighth week. The same goes for software and hardware development in the twelfth week. After the completion of software and hardware development, project progress presentation needs to be made in the twelfth week and prototype production has also been completed in the next week, which is the thirteenth week. In the same week with the presentation of the process, software and hardware integration was also carried out and it was successfully completed a week late. As a result, the time to do system testing has also been disrupted and caused it to start a week late. Once every task has been completed, the final report submission and presentation has been prepared and successfully submitted on time.

5.3 Project Cost

Table 8: Total cost for the components

No.	Components	Unit Cost (RM)	Quantity	Cost (RM)
1	PH Sensor Module	78.92	1	78.92
2	Soil Moisture Sensor	2.50	1	2.50
3	ESP32 Wi-Fi Camera Module	27.50	1	27.50
4	SONY 18650 Rechargeable Battery	11.90	2	23.80
5	Arduino UNO R3 Board	38.90	1	38.90
6	V5 Expanding Board	11.90	1	11.90
7	Bluetooth Module	14.20	1	14.20
8	Battery Case	1.90	1	1.90
9	Robot Tire	1.90	4	7.60
10	SG90 Servo Motor	7.10	3	21.30
11	DC Motor	6.30	4	25.20
12	Jumper Wire	3.70	4	14.80
13	Acrylic Chassis (Body Base)	15.50	2	31.00
			TOTAL	**299.52**

NRE Cost = Shipping Cost + Pepsi drink + Detergent

= RM 4.66 + RM 3.00 + RM 4.00

= RM 11.66

Total Cost = NRE Cost + Components Cost

= RM 11.66 + RM 299.52

= **RM 311.18**

The table above shows the total cost used to make the IGARDRobot. Thirteen types of components were used, and the total of all components was twenty-six components. The total cost for the component is RM 299.52 and it has not yet been added with the NRE Cost. The total of NRE cost is RM 11.66 and it only involve shipping cost and testing material for sensors such as Pepsi drink and detergent. Finally, the components cost and NRE cost will be added to get the total costs and the amount is RM 311.18.

6. FURTHER IMPROVEMENT

Every project will definitely have its weaknesses and it can be repaired and improved. For IGARDRobot, there are a few things that can be improved in terms of hardware and software. First, IGARDRobot can use Wi-Fi technology as a Bluetooth replacement to allow users to control the prototype from a greater distance. Wi-Fi has a larger network than Bluetooth because Wi-Fi is a technology that can now be accessed anywhere. Unlike Bluetooth which has a limited and small network area.

Second, the weight and cost of the IGARDRobot can be reduced by making another separate prototype that works to sprinkle water and fertilizer. As known, IGARDRobot need to carry tanks and pumps of water and fertilizer. This causes the IGARDRobot to need more power to carry it. Therefore, a new prototype can be made to replace that function and it is connected to the IGARDRobot system via Wi-Fi. The prototype needs to be placed around the user's garden so that every splash of water and fertilizer hits their plants.

Third, the camera module used on the IGARDRobot can be replaced with a module that has a better-quality camera. IGARDRobot uses the ESP32-CAM as an eye to the prototype. The ESP32-CAM camera is a little less quality where the camera sensor is only 2 megapixels. As a result, users find it difficult to see and control IGARDRobot around their garden. So, ESP32-CAM can be replaced with a better-quality camera module such as Pcam 5C where it can capture images with better quality of 5 megapixels. This will make it easier for users to see their garden and also control the IGARDRobot.

If a better-quality camera is used, the IGARDRobot system can be upgraded by implementing AI technology on the camera where it can detect a plant. This is because the user also has little difficulty in controlling this prototype so that the position between the IGARDRobot and the plant is in a good condition to perform the task. With a quality camera equipped with an AI system, it can detect the position of the plant and then this AI will send a signal to the IGARDRobot movement system to move the prototype according to the location of the plant. This will make it easier for the user to perform tasks on the plant.

Lastly, the humidity and pH sensors can be upgraded to more accurate ones. Existing sensors are capable of measuring correct values but they are not 100 percent accurate. A handful of gardeners are very concerned about this because accuracy is everything. Thus, better quality sensors are able to give better results to their garden with the help of IGARDRobot.

7. CONCLUSION

In conclusion, IGARDRobot is worth to be done since all the features available turns out to have a very useful impact on the users. The first feature of IGARDRobot is it can be completely controlled by the mobile phone. The robot is equipped with a camera which the output will be displayed via mobile phone. Secondly, it is equipped with a pH control system where it allows the robot to measure the pH value of the specific area. This system has been programmed so that the secretion of fertilizer will be performed if the pH sensor indicates that the pH value of the soil is not in the range of 6 to 8 value. Thirdly, the water sprinkle according to the soil humidity level. If the sensor gives an input that the humidity of the soil is low (below 40 percent), the sprinkler will act to spray water to moisten the soil. Lastly, IGARDRobot comes with a built-in timer. By using this system, the user does not need to perform the process of pesticides and fertilizers toward the plant manually by hand.

Clearly, this project has achieved its objective which is to help the urban farmer community in taking care of their garden. Urban farmer is a very busy community that there is no time for them to take care of the garden at home. The presence of

IGARDRobot in the industry is able to help this type of communities to solve this problem.

8. REFERENCE

[1] Satyam Kumar Sinha, Bhupendra Singh, Aashish Kumar Gupta, IOT Based Smart Garden Monitoring System, International Journal of Scientific & Engineering Research Volume 8, Issue 10, October-2017.

[2] N. Agrawal and S. Singhal, “Smart Drip Irrigation System using Raspberry Pi and Arduino”, International Conference on Computing, Communication & Automation, Noida, 2015.

[3] Yao Chuanan and Yu Yongchang, “Implementation of Greenhouse Monitoring System Based on RF Transceiver”, 2nd International Conference on Industrial Mechatronics and Automation, Wuhan, China, 2010.

[4] M. Sagi, D. Mijic, D. Milinkov and B. Bogovac, “Smart Home Automation”, 20th Telecommunications Forum (TELFOR), Belgrade, 2012.

[5] S. Folea, D. Bordencea, C. Hotea and H. Valean, “Smart Home Automation System Using Wi-Fi Low Power Devices”, Proceedings of 2012 IEEE International Conference on Automation, Quality and Testing, Robotics, Cluj-Napoca, 2012.

[6] T. Thamaraimanalan, S. P. Vivekk, G. Satheeshkumar and P. Saravanan, Smart Garden Monitoring System Using IOT, Asian Journal of Applied Science and Technology (AJAST), 2018.

[7] M. Sheth and P. Rupani, "Smart Gardening Automation using IoT With BLYNK App," *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, 2019, pp. 266-270, doi: 10.1109/ICOEI.2019.8862591. [Accessed July 9, 2021]

[8] N. Sukhdev, N. Nahata, S. Sridhara and G. Swamy, "IoT Enabled Smart Gardening," *2018 Fourteenth International Conference on Information Processing (ICINPRO)*, 2018, pp. 1-2, doi: 10.1109/ICINPRO43533.2018.9096672. [Accessed July 9, 2021]

[9] Shweta B. Saraf “IOT Based Smart Irrigation Monitoring And Controlling System” NBN Sinhgad School of Engineering, gujrathishweta1@gmail.com, Dhanashree H Gawali, dhanashree.gawali@gmail.com 2017

[10] Sandeep Minz. “Arduino Based Automatic Irrigation System” ADBU Journal of Electrical and Electronics Engineering (AJEEE) Volume 3 , issue 1, 2019.

[11] Last Minute Engineers. (2018, Nov. 28). “Interface L298N DC Motor Driver Module with Arduino,” [Online]. Available: https://lastminuteengineers.com/l298n-dc-stepper-driver-arduino-tutorial/

[12] E. Everhart. (2021). “How to Change Your Soil’s pH | Horticulture and Home Pest News,” Iastate.edu, 2021 [Online]. Available: https://hortnews.extension.iastate.edu/1994/4-6-1994/ph.html

[13] R. Steinkamp. (2013, Dec. 30). “Fertilizer Changes Growing Mix pH,” Greenhouse Grower, [Online]. Available: https://www.greenhousegrower.com/production/fertilization/fertilizer-changes-growing-mix-ph/

[14] Arduino Project Hub. (2019). “Complete Guide to Use Soil Moisture Sensor w/ Examples”. [Online]. Available: https://create.arduino.cc/projecthub/electropeak/complete-guide-to-use-soil-moisture-sensor-w-examples-756b1f

[15] Gauger, M., Minder, D., Marron, P. J., Wacker, A., & Lachenmann, A. (2008, April). Prototyping sensor-actuator networks for home automation. In Proceedings of the workshop on Real-world wireless sensor networks (pp. 56- 60). ACM.

APPENDIX

1. Group Photo with Prototype

Figure 18: Group 3 Members with IGARDRobot Prototype (From left; Shabil, Zaki, Ain, Amir).

2. Circuit Schematic:

   I. Power Source Connection

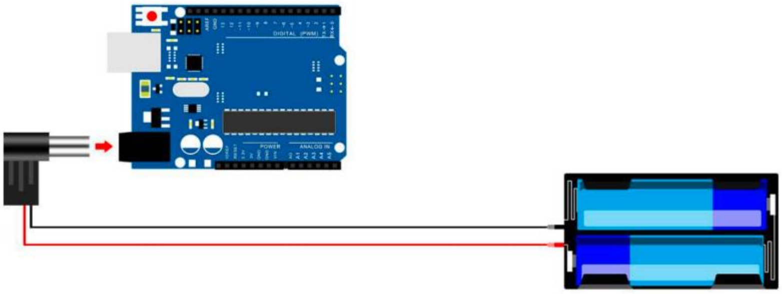


Figure 19 Power supply connection

   II. Bluetooth Connection

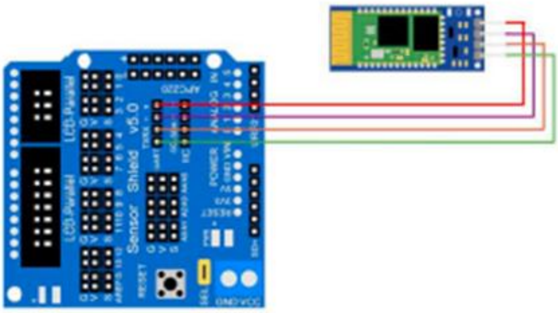


Figure 20 Bluetooth module connection

III. DC Motor with L298N Motor Driver module Connection.

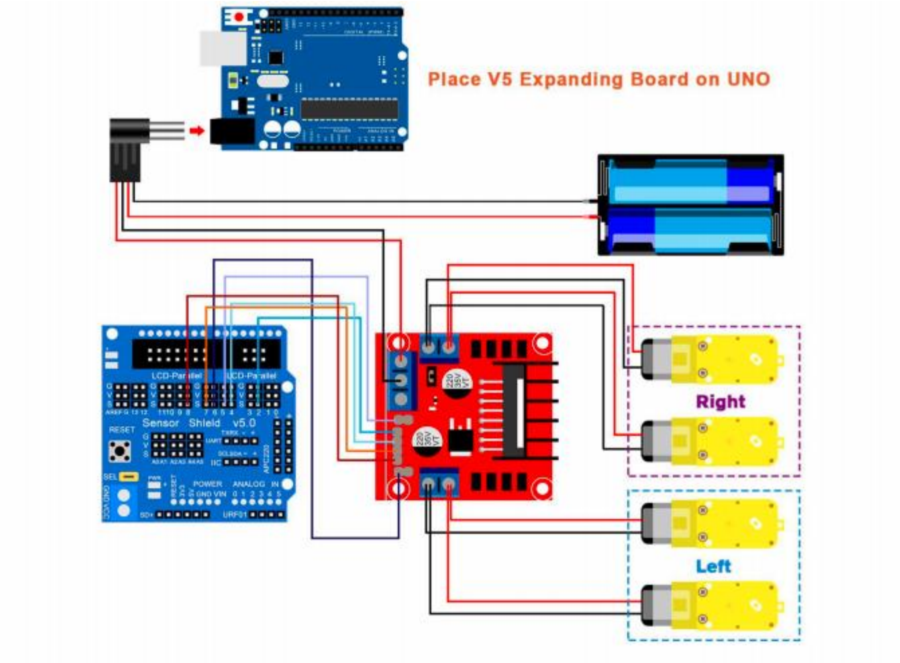


Figure 21 Motor connection to driver and power supply.

IV. Servo Motor Connection

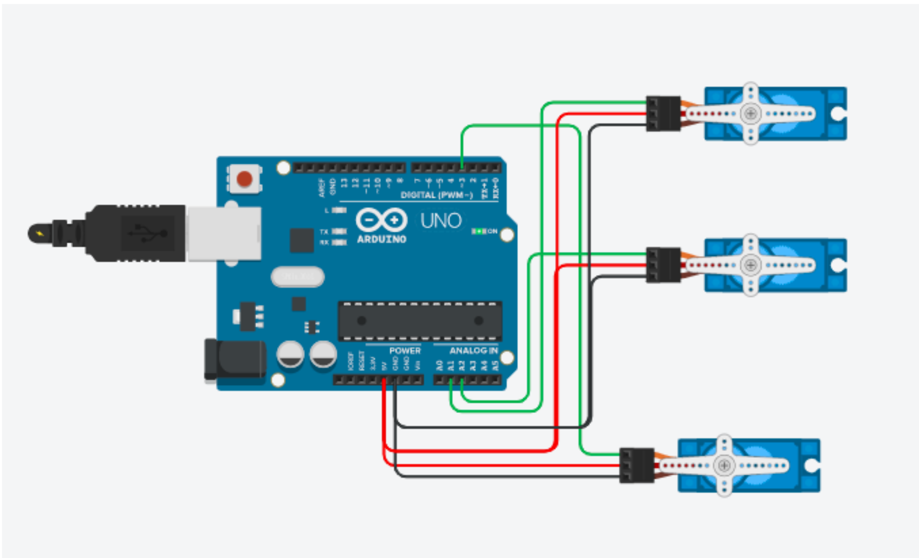


Figure 22 Connection of servo motor on Arduino Uno

V. PH Sensor Connection

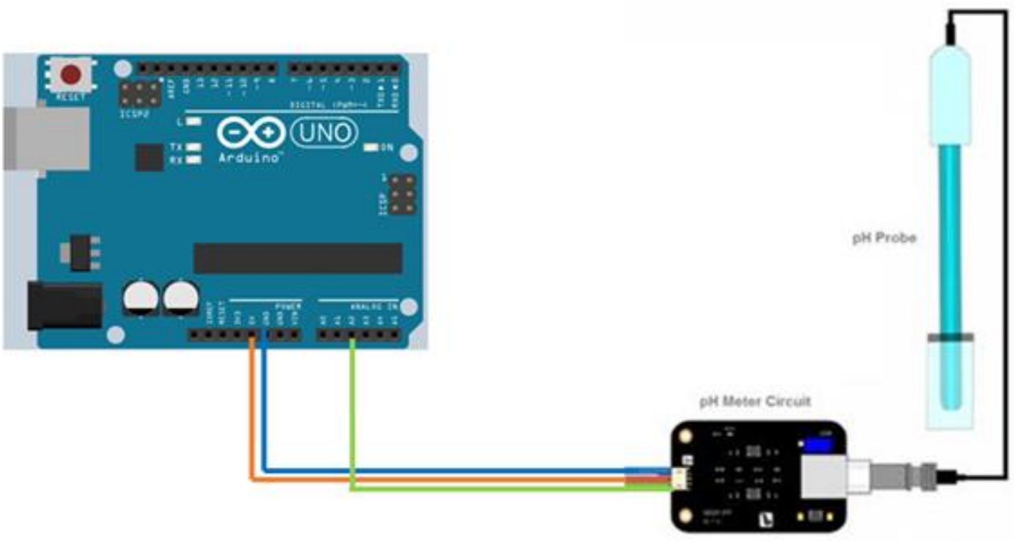


Figure 23 Ph sensor connection of pin

VI. Soil Humidity Sensor

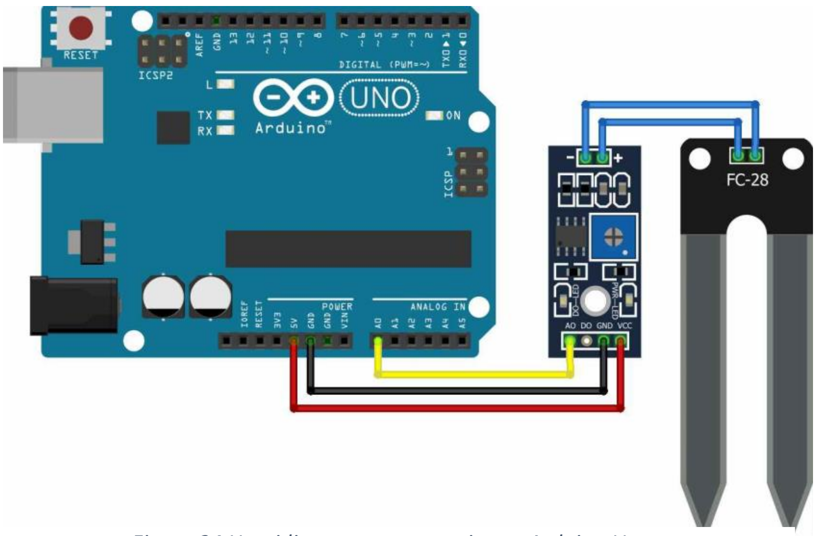


Figure 24 Humidity sensor connection to Arduino Uno

VII. LED Connection

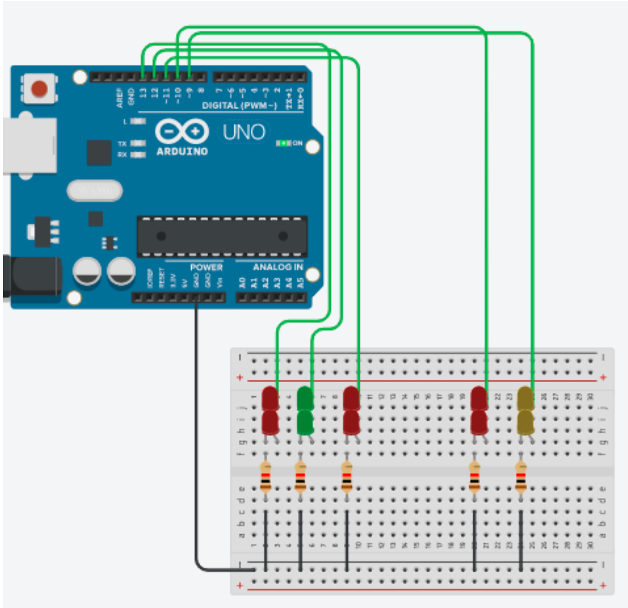


Figure 25 LED connection as indicator

VIII. ESP32 Wi-fi Camera Module Connection

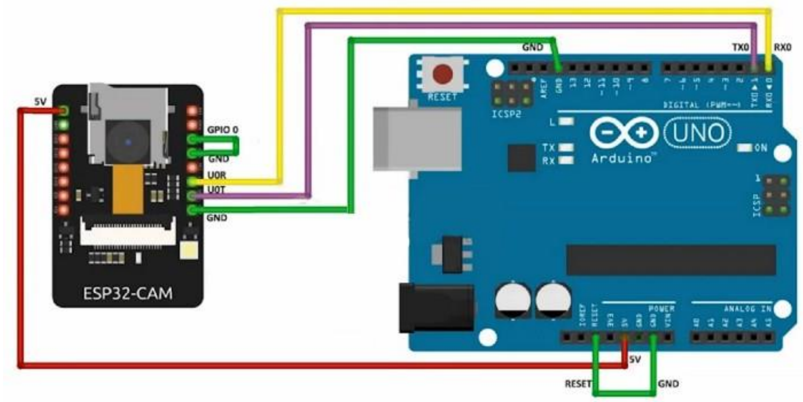


Figure 26 Connection ESP32 to loaded coding into it.

IX. OLED Display Connection

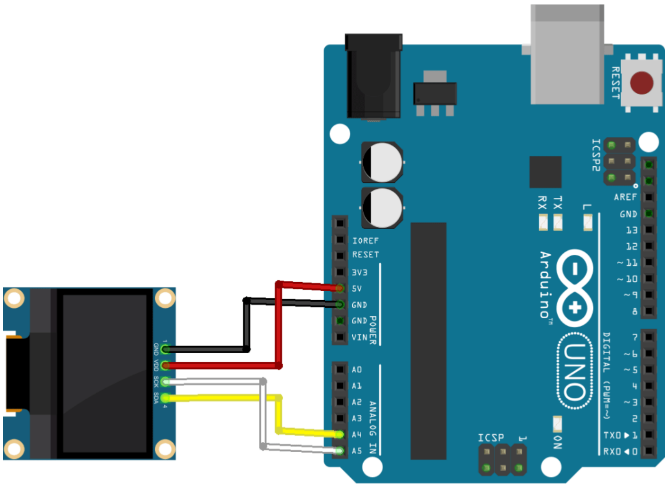


Figure 27 Connection for OLED

3. IGARDRobot Source Code

```
#include <Wire.h>           //including libraries of I2C
#include <Servo.h>          //including libraries of servo
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

Servo myservo;
Servo myservoR;
Servo myservoL;
Servo myfertil;
Servo mysprink;

unsigned char bluetooth_data;
unsigned long Key;
unsigned long Key1;

#define Lpwm_pin  5    //pin of controlling speed---- ENA of motor driver board
#define Rpwm_pin  6    //pin of controlling speed---- ENB of motor driver board

int pinLB=2;           //pin of controlling turning---- IN1 of motor driver board
int pinLF=4;           //pin of controlling turning---- IN2 of motor driver board
int pinRB=7;           //pin of controlling turning---- IN3 of motor driver board
int pinRF=8;           //pin of controlling turning---- IN4 of motor driver board
int flag=0;
int Car_state=0;           //the working state of car
int myangle;             //defining variable of angle
int pulsewidth;           //defining variable of pulse width
float phValue;
int moisture_percentage;
unsigned char DuoJiao=90;    //initialized angle of motor at 90°

//DEFINE SENSOR PIN
const int humidity_pin = A0;
#define pH_pin 3
unsigned long int avgValue;  //Store the average value of the pH sensor feedback
float b;
int buf[10],temp;

//DEFINE OLED SCREEN
#define SCREEN_WIDTH 128  // OLED display width, in pixels
#define SCREEN_HEIGHT 64  // OLED display height, in pixels
#define OLED_RESET    -1  // Reset pin # (or -1 if sharing reset pin)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);


//DEFINE LED PIN
int REDL = 13;
int GREENL = 12;
int REDR = 11;
int fertilizer = 10;
int sprinkler = 9;
```

```
void M_Control_IO_config(void) //function for setup output
{
 pinMode(pinLB,OUTPUT); // /pin 2
 pinMode(pinLF,OUTPUT); // pin 4
 pinMode(pinRB,OUTPUT); // pin 7
 pinMode(pinRF,OUTPUT);  // pin 8
 pinMode(Lpwm_pin,OUTPUT);  // pin 5 (PWM)
 pinMode(Rpwm_pin,OUTPUT);  // pin6(PWM)
 //LED
 pinMode(REDL, OUTPUT);
 pinMode(GREENL, OUTPUT);
 pinMode(REDR, OUTPUT);
 pinMode(fertilizer, OUTPUT);
 pinMode(sprinkler, OUTPUT);

}

void Set_Speed(unsigned char pwm) //function of setting speed
{
 analogWrite(Lpwm_pin,pwm);
 analogWrite(Rpwm_pin,pwm);
}

//////////////////MOVEMENT CONTROL///////////////////////
void stopp()        //stop
  {
   digitalWrite(pinRB,HIGH);
   digitalWrite(pinRF,HIGH);
   digitalWrite(pinLB,HIGH);
   digitalWrite(pinLF,HIGH);
   //Car_state = 5;
   oled(); //display IGARD ROBOT
   digitalWrite (REDL, LOW);
   digitalWrite (GREENL, LOW);
   digitalWrite (REDR, LOW);
   digitalWrite (fertilizer, LOW);
   digitalWrite (sprinkler, LOW);
   moisture_percentage=0;
   phValue=0;
   myfertil.write(0);
   mysprink.write(0);

  }

void advance()    //  going  //back
  {
   digitalWrite(pinRB,LOW);  // making motor move towards right rear
   digitalWrite(pinRF,HIGH);
   digitalWrite(pinLB,LOW);  // making motor move towards left rear
   digitalWrite(pinLF,HIGH);
   //Car_state = 1;

  }
```

```
void turnR()       //turning right(dual wheel)
   {
    digitalWrite(pinRB,LOW);  //making motor move towards right rear
    digitalWrite(pinRF,HIGH);
    digitalWrite(pinLB,HIGH);
    digitalWrite(pinLF,LOW);  //making motor move towards left front
    // Car_state = 4;
   }
void turnL()       //turning left(dual wheel)
   {
    digitalWrite(pinRB,HIGH);
    digitalWrite(pinRF,LOW );   //making motor move towards right front
    digitalWrite(pinLB,LOW);   //making motor move towards left rear
    digitalWrite(pinLF,HIGH);
    // Car_state = 3;
     }
void back()        //back up
   {
    digitalWrite(pinRB,HIGH);  //making motor move towards right rear
    digitalWrite(pinRF,LOW);
    digitalWrite(pinLB,HIGH);  //making motor move towards left rear
    digitalWrite(pinLF,LOW);
    //Car_state = 2;
   }
/////////////////////////CONTROL SERVO///////////////////////////////////
void svR_up()//up - initial
{  myservoR.write(90); //Right servo up
 }
void svL_up()//up - initial
{  myservoL.write(0); //Left servo up
    }
void svR_down()//Right servo down
{
 flag = 0;
  while (flag == 0)
  {
   int angleR = 90;
   for(angleR = 90; angleR > 0; angleR -= 1)    // command to move from 0 degrees to 90 degrees
   {
     myservoR.write(angleR);              //command to rotate the servo to the specified angle
     delay(100);
   }

  if (Serial.available())
    {
     bluetooth_data = Serial.read();
     if (bluetooth_data == 'S')
     {
       flag = 1;
     }
    }
  }
}
```

```
void svL_down()// Left servo down
{  flag = 0;
  while (flag == 0)
  {
    int angleL = 0;
    for(angleL = 0; angleL < 90; angleL += 10)    // command to move from 0 degrees to 180
degrees
    {
      myservoL.write(angleL);               //command to rotate the servo to the specified angle
      delay(100);
     }

    if (Serial.available())
     {
      bluetooth_data = Serial.read();
      if (bluetooth_data == 'S')
      {
        flag = 1;
      }
     }
  }
}
void cam_left(){
  myservo.write(140);
}
void cam_center(){
  myservo.write(90);
}
void cam_right(){
  myservo.write(50);
}
//////////////////////SENSOR///////////////////////////////
void humidity(){ //Function for measure soil humidity
  //float moisture_percentage;
  int sensor_analog;
  sensor_analog = analogRead(humidity_pin);
  moisture_percentage = ( 100 - ( (sensor_analog/1023.00) * 100 ) );
  Serial.println(moisture_percentage);
  display.clearDisplay();
  display.setTextSize(2);
  display.setCursor(20,0);
  display.println("Humidity:");

  display.setTextSize(3);
  display.setCursor(30,30);
  display.print(moisture_percentage);
  display.println("%");

  display.display();
```

```
 if (moisture_percentage>=40 && moisture_percentage<=60){
  digitalWrite(GREENL, HIGH);
  digitalWrite(REDL, LOW);
  digitalWrite(sprinkler, LOW);
  }
 else if (moisture_percentage<40){
  digitalWrite(GREENL, LOW);
  mysprink.write(180);
  digitalWrite(REDL, HIGH);
  digitalWrite(sprinkler, HIGH);
  delay(300);
  mysprink.write(90);
  delay(100);
  mysprink.write(0);
  delay(100);
 }
 else{
  digitalWrite(REDL, HIGH);
  digitalWrite(sprinkler, LOW);
 }

}

void sprinkler_manual(){ //Function for turn on water sprinkler pump manually
 digitalWrite(sprinkler, HIGH);
 mysprink.write(140);
}
void fertilizer_manual(){ //Function for turn on fertilizer pump manually
 digitalWrite(fertilizer, HIGH);
 myfertil.write(140);
}

void pH(){ //Function for measure soil pH
 for(int i=0;i<10;i++)       //Get 10 sample value from the sensor for smooth the value
 {
  buf[i]=analogRead(pH_pin);
  delay(10);
 }
 for(int i=0;i<9;i++)        //sort the analog from small to large
 {
  for(int j=i+1;j<10;j++)
  {
   if(buf[i]>buf[j])
   {
    temp=buf[i];
    buf[i]=buf[j];
    buf[j]=temp;
   }
  }
 }
 avgValue=0;
 for(int i=2;i<8;i++)                    //take the average value of 6 center sample
  avgValue+=buf[i];
 phValue=(float)avgValue*5.0/1024/6; //convert the analog into millivolt
 phValue=3.5*phValue;                    //convert the millivolt into pH value
 Serial.println(phValue);
```

```
display.clearDisplay();
 display.setTextSize(2);
 display.setCursor(20,0);
 display.println("pH Value:");
  display.setTextSize(3);
 display.setCursor(30,30);
 display.print(phValue);
  display.display();

  if (phValue>=6 && phValue <=8){
      digitalWrite(REDR, LOW);
     digitalWrite(fertilizer, LOW);
    }
 else {
   myfertil.write(180);
   digitalWrite(REDR, HIGH);
   digitalWrite(fertilizer, HIGH);
   delay(300);
   myfertil.write(90);
   delay(100);
   myfertil.write(0);
   delay(100);
 }
}
//////////OLED DISPLAY IGARD ROBOT//////////////
void oled(){
   if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C))
    {
       for(;;); // Don't proceed, loop forever
    }
     display.display();
     delay(2);
     display.clearDisplay();

    display.clearDisplay();
    display.setTextColor(WHITE);
    display.setTextSize(2);
    display.setCursor(0,5);
    display.print(" IGARD ");
    display.setCursor(30,30);
    display.print("   ROBOT");
    display.display();
    delay(100);
}
void setup() //setup function
{
  //pinMode(13,OUTPUT);
  int angle = 0;
  myservo.attach(A2);  // Camera servo
  myservoR.attach(3); //Right servo
  myservoL.attach(A1); //Left servo
  myfertil.attach(10);
  mysprink.attach(9);

  M_Control_IO_config();     //motor controlling the initialization of IO
  Set_Speed(80);  //setting initialized speed
```

```
 Serial.begin(9600);          //initialized serial port , using Bluetooth as serial port, setting baud
  myservo.write(DuoJiao);       //angle of motor at 90°
  stopp();                    //stop
  delay(1000);
  display.clearDisplay();
  display.display();
  oled();
}
void loop() //Loop function for repeating get the bluetooth data
{
 if (Serial.available())
 {
   bluetooth_data = Serial.read();
  }
 switch (bluetooth_data) {
   case 'U':
   advance(); //back
   Set_Speed(150);
   break;
  case 'D': //forward
   back();
   Set_Speed(150);
   break;
  case 'L':
   turnL();
  Set_Speed(120);
   break;
  case 'R':
   turnR();
   Set_Speed(120);
   break;
  case 'S':
   stopp();
   Set_Speed(0);
   break;
  case 'A':
   svR_up();  //Right servo up
   break;
  case 'B':
   svR_down(); //Right servo down
   break;
  case 'C':
   svL_up(); //Left servo up
   break;
  case 'I':
   svL_down(); //Left servo down
   break;
   case 'E':
   pH(); //check pH
   break;
  case 'F':
   humidity(); //check humidity
   break;
  case 'G':
   fertilizer_manual(); //fertilizer manually
   break;
```

```
case 'H':
   sprinkler_manual(); //water sprinkler manually
   break;
  case 'V':
   cam_left(); //camera servo turn left
   break;
  case 'N':
   cam_center(); //camera servo at center
   break;
  case 'M':
   cam_right(); //camera servo turn right
   break;
  default:
   break;
 }

}
```