King Saud University
College of Computer and Information Science
Department of Computer Engineering

Controlled Greenhouse

CEN 492 (Phase 1)

Ghanim Al-Ghanim 437104932
Othman Al-Beyabi 437100900

Supervisor: Dr. Abdulwadood Abdulwaheed

December 7, 2020

Contents

List of Figures

List of Algorithms

Abstract

The modern world is heavily digital since almost everything around us is managed by digitization and the role modern technologies must contribute to the farm and agriculture fields is more important now than any time before. Modern day research papers have shown great capabilities for Internet of Things (IoT) to have a positive impact on the world in general and specifically in farming. The agricultural industries must grow and evolve their application of IoT to sustain almost 10 billion people by the 2050s. Unforeseen challenges brought upon by climate change must be solved and overcome in order to sustain and feed the population. IoT in farming produce smart ways of growing, harvesting, and sustaining the farmer's crops, and will make work much more efficient in terms in numbers of fertilizers used and farming trucks gas mileage. Hardware and software of the IoT for smart farming have shown successful results. One of the hypotheses suggested is to use a mix of Arduino IoT and Wireless Sensor Network (WSN) to manage modern smart farming. The ecological conditions of agriculture play a big role in the yield of crops and must be observed and measured from time to time. The feature of this report includes the development of a system which can monitor temperature, humidity, and moisture, and in case of any inconsistency generate a notification.

Chapter 1

Introduction

1.1 Problem statement

With the rising problem of global climate change, our local environment could be harmed badly which calls for innovation in the way our farming system works.

This project aims to combat several challenges at once in that it will make farming and agriculture much more accessible to all people, the public would start making mini-farms inside houses and farmers lives will be made much easier with them gaining the ability to possibly remotely monitor and issue general commands regarding their farms [1].

Around the globe water is not being used optimally, but if the option of digitization is implemented water usage efficiency in agricultural systems could increase by up to 70 percent more efficient than traditional watering systems. Added relative adaptation as a possible result of future growth could allow certain fruits and vegetables to grow outside of their environmental comfort zones, like the native winter strawberries belonging to the Al-Qassim region being grown in the summer.

Wireless system networks will be used to provide statistics about the plants and will be coupled with a neural network (NN) to provide better options on what to do next for the farmer, like temperature control, irrigation level, and whether or not or not to harvest the plant [2].

1.2 Objective

What is presented in this project is a smart farming system that will help in increasing water usage proficiency, and water level being utilized in a rational manner and the greenhouse does not require the farmer to be around as much since it is automated.
Development of the crops depends on proper scheduling of irrigation and irrigation techniques. Humidity, temperature, and wind speed are some of the factors that need to be monitored, in addition to weather data which is a part of decision-making process [3].
The smart agriculture system proposed in this project will utilize and apply practically IoT and WSN in order to make farmers lives easier an example of which is to use an irrigation schedule for their farming needs, each according to their profile [3].
Water quantity control is easily modified by the end user by the web application which is interconnected with a microcontroller that uses an algorithm to dictate the proper threshold values for moisture of the soil and overall temperature .

1.3 Related work

The OpenAG™ personal food computer (PFC), is an open-source platform that is personal use which reduces the size, number of a system component, and it is a customizable platform. The PFC hardware uses a raspberry pi to saves pictures and neural network of the plants using two different cameras, and program the Arduino to control the customizable board called shields which are modular circuit boards that reads from all the sensors of the system are: water level sensor, water temperature sensor, PH sensor, EC sensor, CO2 sensor, and humidity temperature sensor. After gathering the reading. It provides commands to the shield, which is going to activate and deactivate the controlled subsystems as needed. The PFC software uses a graphical user interface (GUI) and a simple readable commands to achieve user-friendly software. It also gathers plant data using the sensors and saves it as a open database [14].

The FarmBot is an Open-Source Computer numerical control Farming Company that have provide a farming robot and provide all their research in the open source website.FarmBot have used aluminium extrusions to every axis. They also use silver extrusions to reduce the temperature. It uses 24V DC power supply which drives raspberry pi, special circuit board Farmduino, and the four custom step motors for the FarmBot movement. Farmduino is the microcontroller that controls and provides power to all the FarmBot electronic parts as shown in Figure 1.1. FarmBot system diagram shows that it communicates with the raspberry pi. This is done using G-code like language [9]. All the work is done by universal tool mount (UTM) which is the main head of all the tools that help the FarmBot to operate by attaching and detaching the tools automatically as needed. The tools are seed injector, weeder and soil sensor. The seed injector is operated by a vacuum pump and a needle. It uses suction to hold the seed at the tip of the needle and releases it into the soil. The weeder is a 3D printable component that pushes the weeds into the soil and disrupt their fragile root systems. The soil sensor accurately reads the soil properties by injecting the sensor into the soil [9]. FarmBot uses a sub-system for controlling light and another one to control the watering system.

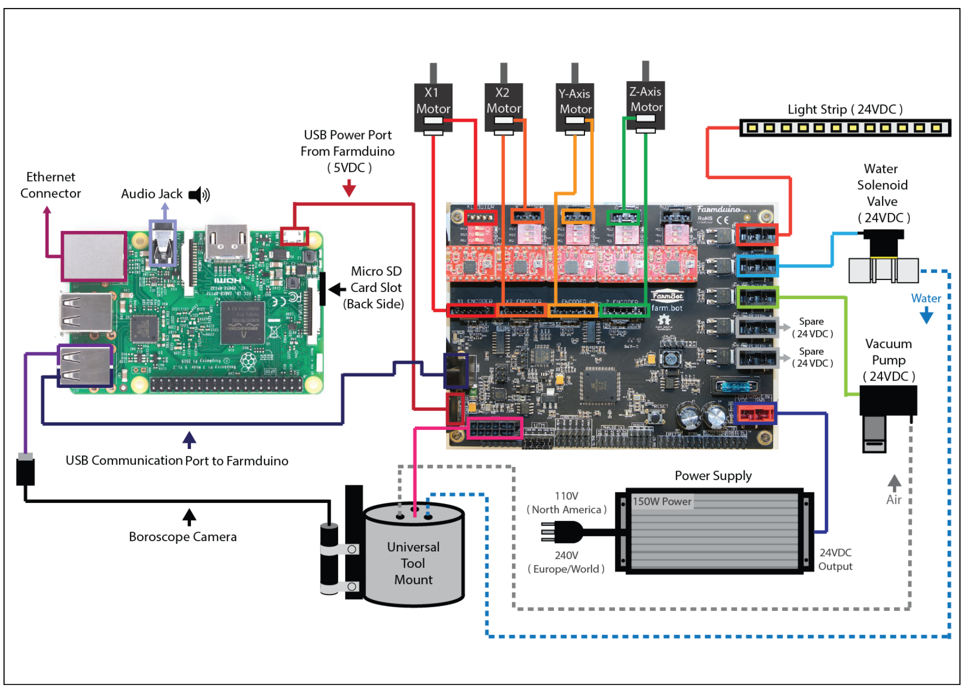

Figure 1.1: FarmBot system diagram [9].

Chapter 2

Greenhouse design

2.1 Introduction

In this section we are going through the three different designs we have seen. We will go through each one in details. Initially, we got inspired from MIT's Open Agriculture Initiative: Personal Food Computer [8]. Then we designed a 3D greenhouse to agree on the structure and have an idea about the shape and the layout of the plants and other parts in the greenhouse. We have used the first two designs to satisfy the needs of an effective automated greenhouse system.

1. First design:
   In this design we got inspired from MIT's Open Agriculture Initiative: Personal Food Computer [8]. We tried to scale up their design to meet our needs. We have a robotic arm in the greenhouse and the way they place the electrical and the water systems did not work for us. We have listed the advantages and the disadvantages of this design.

   - Advantages of using this design
     (a) Can be assembled and taken apart
     (b) The joints strength
   - Disadvantages of using this design
     (a) Percentage of the light transition
     (b) Electrical and water separation
     (c) Water circulation
     (d) Not enough space for arm movement

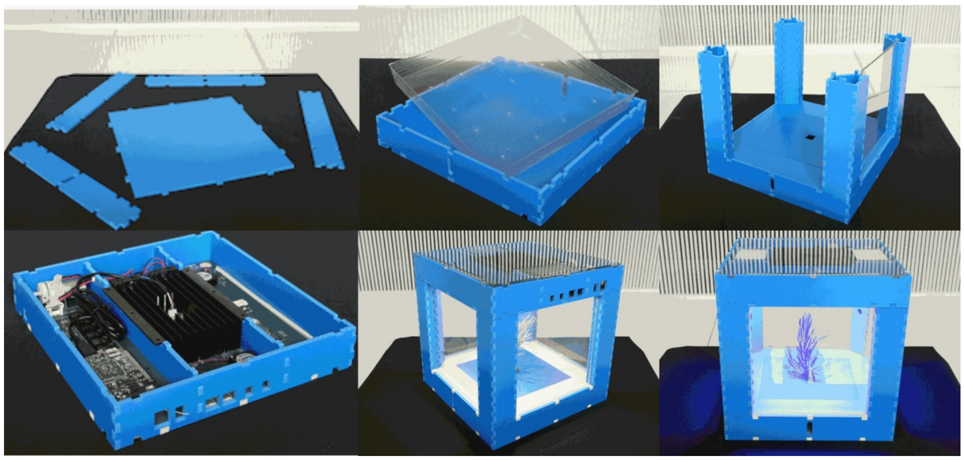

Figure 2.1: MIT's Personal Food Computer design [8].

2. Second design:
   In the second design we tried to use our 3D design skills to make a greenhouse with the robotic arm in the center. We decided to have four plants on every corner and the water tank and the water circulation in one side and the electrical circuits and controller in the other side, isolated from each other.

   - Advantages of using this design
     (a) Details of water circulation
     (b) Details of electrical and water separation
     (c) Percentage of the light transition
   - Disadvantages of using this design
     (a) No exact measurement of the greenhouse
     (b) Weak and cannot handle robotic arm movement

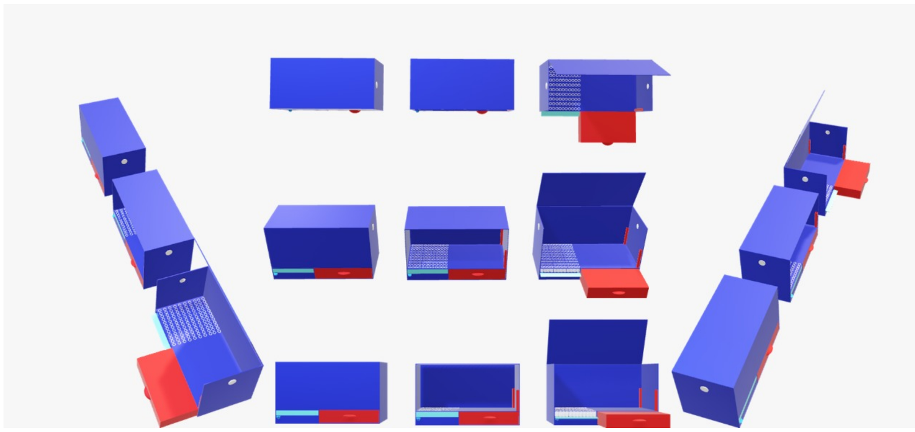

Figure 2.2: Greenhouse 3D design.

3. Third design:
   In this design we have combined what we wanted from the first and the second designs. We came with a much better design that has the water circulation and the separation from the first design. It has the strength, structure, and maintainable parts from the second design.

   - Advantages of using this design
     (a) Can be assembled and taken apart
     (b) The joints strength
     (c) Well-structured design
     (d) Can handle the arm movement
     (e) Water circulation
     (f) Electrical separation
   - Disadvantages of using this design
     (a) Cost
     (b) Manufacture cutting

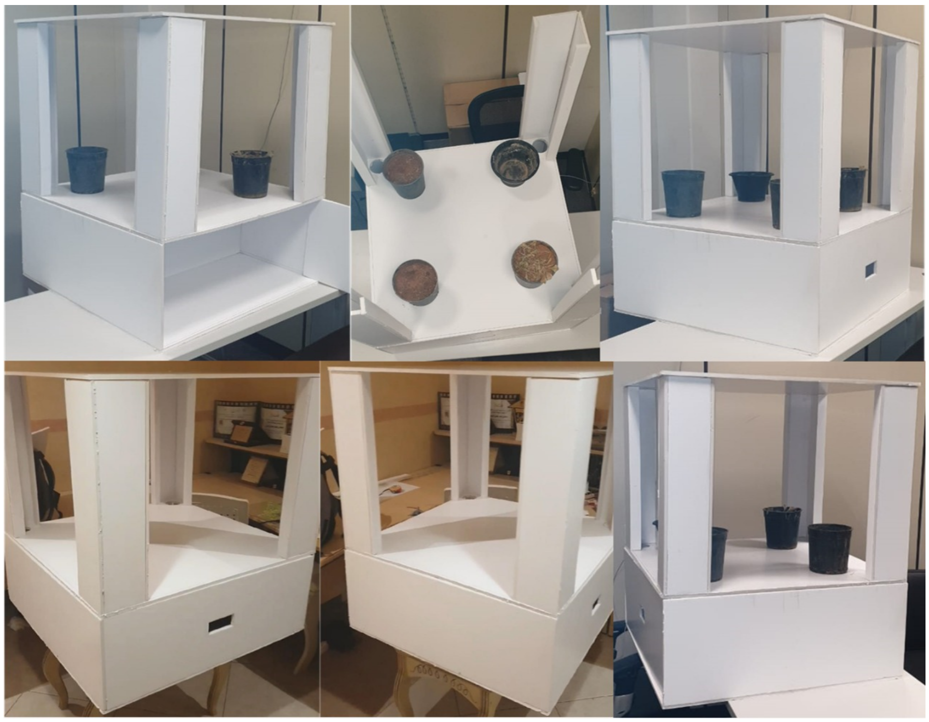

Figure 2.3: Proposed greenhouse design.

2.2 Overview of the controlled greenhouse system

The system consists of Arduino Megas, robotic arm, and sensors including but not limited to moisture, temperature, and humidity sensors.

Each subsystem has its own Arduino Mega so that it should not take turns and interrupt or miss data since it would read the sensors line by line.

Arduino Mega has useful features as shown in Figure 2.4, for example, the serial IO to send the data to the raspberry pi and the Bluetooth to control the arm.

The overall block diagram of the controlled greenhouse system includes multiple Arduino Megas. One is used to control the robotic arm, another one is connected to provide the raspberry pi with all the sensors reading, and the others are controlling the subsystems, as shown in Figure 2.5.

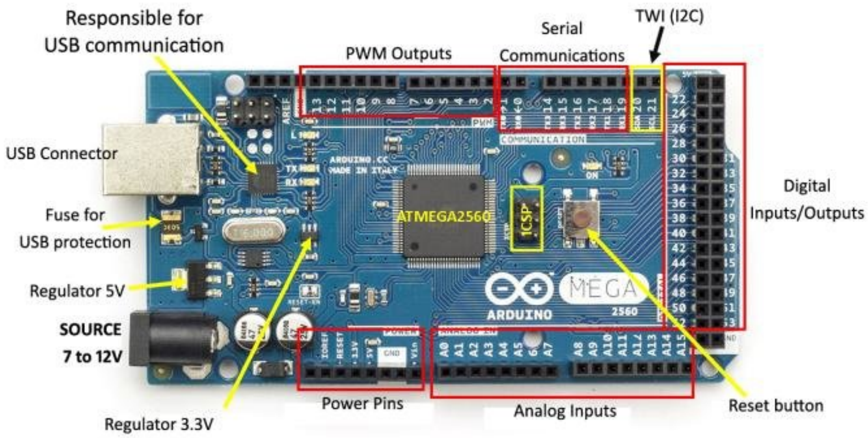

Figure 2.4: Block diagram - Arduino MEGA 2560[7].

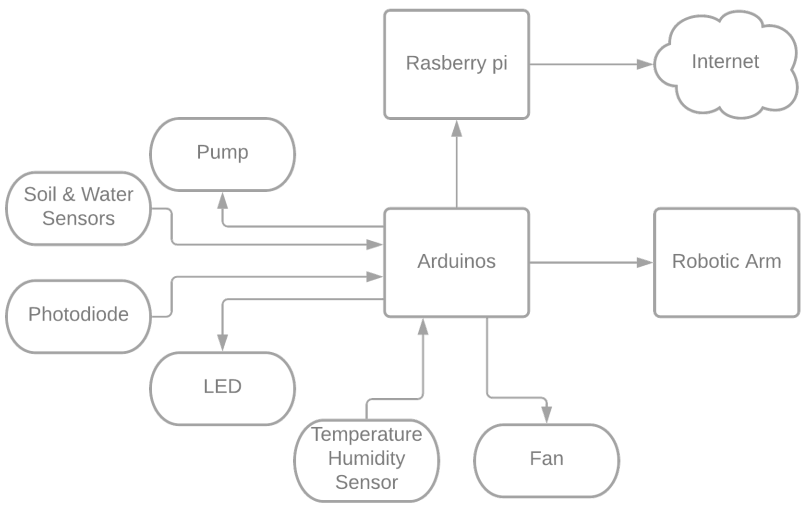

Figure 2.5: Overall block diagram of the controlled greenhouse.

Chapter 3

Controlled greenhouse subsystems

3.1 Introduction

The controlled greenhouse is divided into subsystems and each subsystem is controlled by an Arduino. The subsystems are 1) the water/pump subsystem to control the watering level and frequency of watering, 2) fan subsystem that is controlling the humidity and temperature level of the environment, and 3) the artificial lights system is controlling the amount of light in the environment.

3.2 Subsystems

3.2.1 Water/Pump

Water is the most important factor for the plant growth and being able to know when the plant requires to be watered is crucial. To automatically water the plants we use a soil moisture sensor and water level sensor which are attached to the soil that contains the plant. When the readings of the sensors are below the threshold value, the pump will start watering the plant using the built-in water tank.

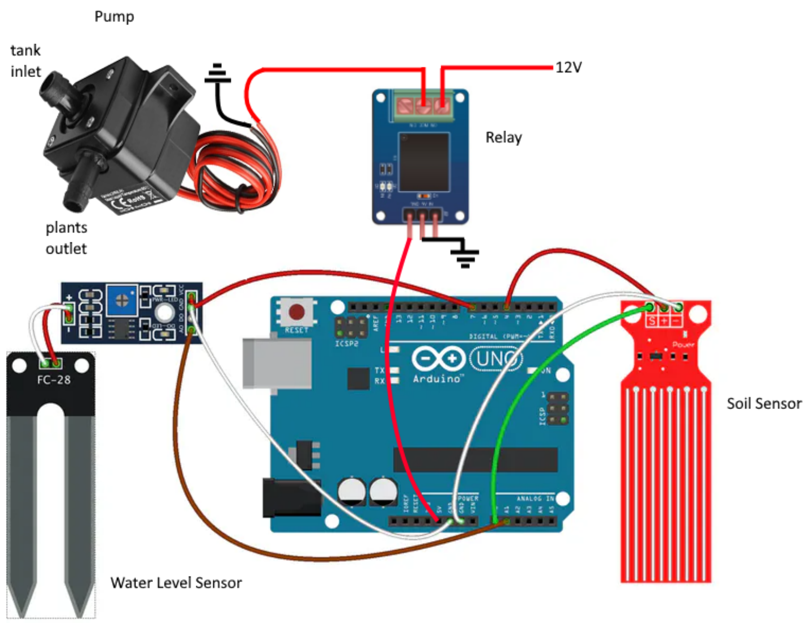

Figure 3.1: Water pump circuit diagram.

Pump activation conditions

To activate the pump we acquire data from the moisture and water level sensor. Based on the soil moisture sensor and the water level sensor values as shown in Table 3.1 the system will be determine when to activate the water pump or when to deactivate it. Soil moisture sensor determines the percentage moisture of the soil, while the water level sensor changes according to the amount of water in contact with the sensor. The pump algorithm controls the function of the water pump.

Soil Moisture Sensor	Water Level Sensor	Water Pump
Soil Sensor $<$ 10%	450 $>$ Water Sensor until 700	High
Soil Sensor $\geq$ 10%	700 $\geq$ Water Sensor $\geq$ 450	Low

Table 3.1: Pump activation conditions.

Pump activation algorithm

Algorithm 1 Pump

```
SoilSensorValue = analogRead(SoilSensorPin)
remap(SoilSensorValue from 550-0 to 0-100)
WaterLevelSensorValue = analogRead(WaterLevelSensorPin)
if (SoilSensorValue<10 AND WaterLevelSensorValue<450) then
    digitalWrite(Pump, HIGH);
    for WaterLevelSensorValue<700 until WaterLevelSensorValue>700 do
        SoilSensorValue = analogRead(SoilSensorPin)
        remap(SolSensorValue from 550-0 to 0-100)
        WaterLevelSensorValue = analogRead(WaterLevelSensorPin)
        Delay for 10 Seconds
    end for
else
    digitalWrite(Pump, LOW);
end if
Delay for 10 Seconds
```

3.2.2 Fan

Temperature is the second important factor, we can adjust the temperature of the plants since some plants require a different environment than others, not all plants have the same temperature requirements for optimal growth [4]. Furthermore, we can also adjust the humidity level of the environment based on the plant's natural environment since it affects the plant's health. When the temperature reaches a certain threshold, the system activates the fan to reduce the temperature of the environment to accommodate the plant's needs. To read the threshold we use a humidity temperature sensor (DHT11) [10] that provides information about the temperature and humidity of the greenhouse.

Fan activation algorithm

In this algorithm, the system reads the humidity and the temperature values in the environment using the temperature-humidity sensor (DHT11) [10] and activates the fan when the temperature reaches 30°C or the humidity reach 40% and deactivates it otherwise.

Algorithm 2 Fan

```
Temp Value = Temp-Humidity-Sensor.readTemperature()
Humidity Value = Temp-Humidity-Sensor.readHumidity()
if (Temp Value > 30 OR Humidity Value > 40) then
    digitalWrite(Fan, HIGH)
else
    digitalWrite(Fan, LOW)
end if
Delay For 10 Seconds
```

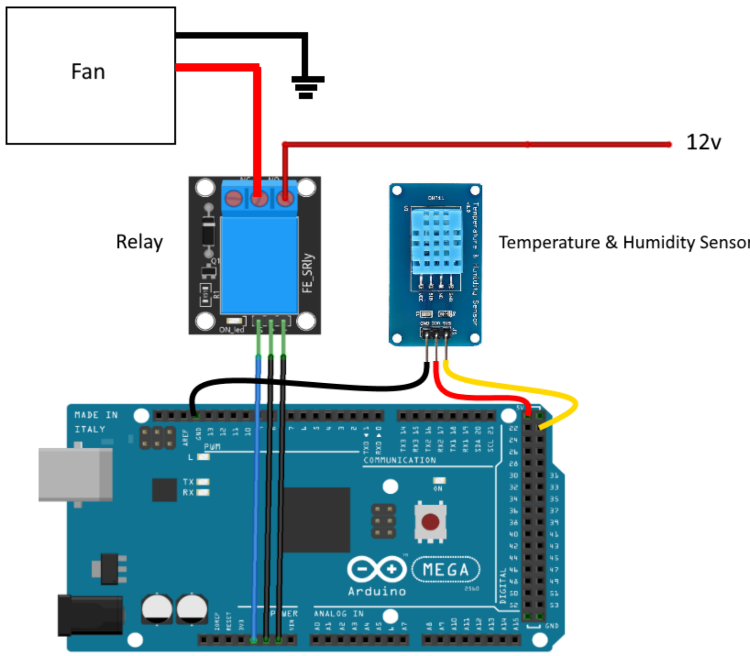

Figure 3.2: Fan circuit diagram.

3.2.3 Artificial lights

Plants must have enough light to survive and produce food and the general case is that the more light the more food is given by the plant. Since the plants are in a closed environment it cannot get light from the sun thus we use artificial light.

The general case for house plant lights is fluorescent lights since they are the cheapest. They come in compact bulbs or tubes that screw into regular lamp sockets. Their heat is not high enough to damage the plants and can safely be placed next to them. Generic fluorescent tubes and bulbs are higher in blue wavelengths, so it is better to seek a mix of warm and cool bulbs since the cool bulbs emit white light that contain the full light-spectrum. The fluorescent lights are are placed 1/3 meters away from the plants for the highest efficiency [5].

It is not recommended to place incandescent lights next to plants since they emit a red wavelength that produces heat that can damage the plant, however, if you wish to help the plant to bloom faster it is recommended to use a balance of incandescent and fluorescent lights to have an even spectrum. Specifically if 1/3 incandescent and 2/3 fluorescent light by watt is used [5].

LED lights are also a low heat, energy-efficient artificial light source. Because LED technology is customized, every diode is different, Diodes that produce blue and red lights is helpful for plants. Horticultural LED growlights are used to produce only the wavelengths most utilized by plants [5].

For our project we used LED because of its size, it emits light in a specific direction, it reduces the need for reflectors that can trap light and their very little heat. In comparison, incandescent bulbs release 90 % of their energy as heat and fluorescent lamps release about 80 % of their energy as heat.

LED activation algorithm

For the LED we read light values in the environment using the photoresistor (PGM5539) [11] and turn the LED on when the light value is less than 30% of what the plant needs and turn it off otherwise.

Algorithm 3 LED

PhotoresistorValue = analogRead(Photoresistorpin);
remap(PhotoresistorValue from maxRead-minRead to 0-100)
if (PhotoresistorValue < 30) **then**
    digitalWrite(LED, HIGH);
else
    digitalWrite(LED, LOW);
end if
Delay for 10 Seconds

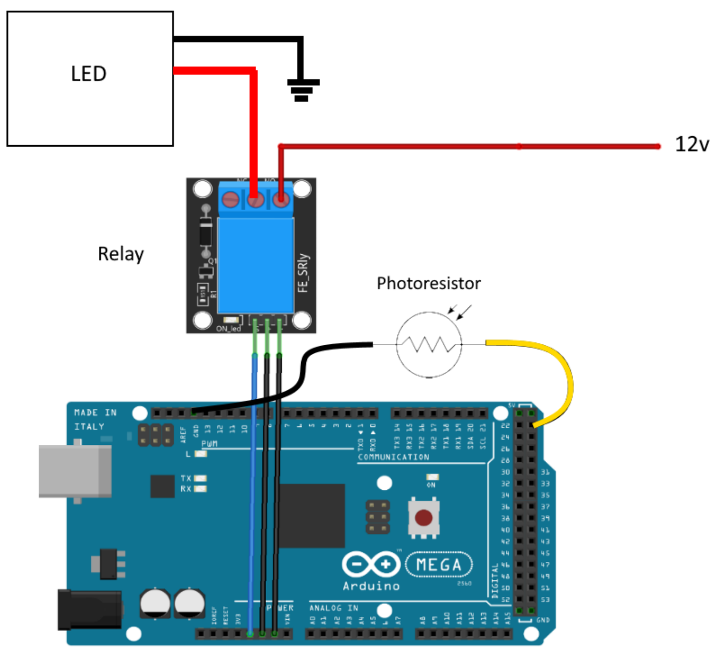

Figure 3.3: LED circuit diagram.

Chapter 4

Arm control system

4.1 Introduction

In this chapter, we will go through the assembly of the Arduino robotic arm from "How to Mechatronics" website [6]. We will also present how we have controlled the arm movement using Arduino and how we send commands wirelessly.

4.2 Assembling the robotic arm

Step 1: Place the first MG996R servo in the base of the arm and attach it using screws, then place the base top on top of the servo and attach it to the servo shaft output.

Step 2: Place the second MG996R servo into the base top and attach it, then attach the shoulder to the output shaft and it will be helpful to include springs to the shoulder to support the servo since it is the servo that hold all the rest parts weight.

Step 3: Repeating step 2 with the elbow in the same way using the third MG996R servo, then attach the rotational joint with the SG90 micro servo and similarly the second SG90 micro servo attached to the head.

Step 4: Use the rest of the parts to connect the SG90 micro servo to gripper as showed in the Figure 4.1. [12][13].

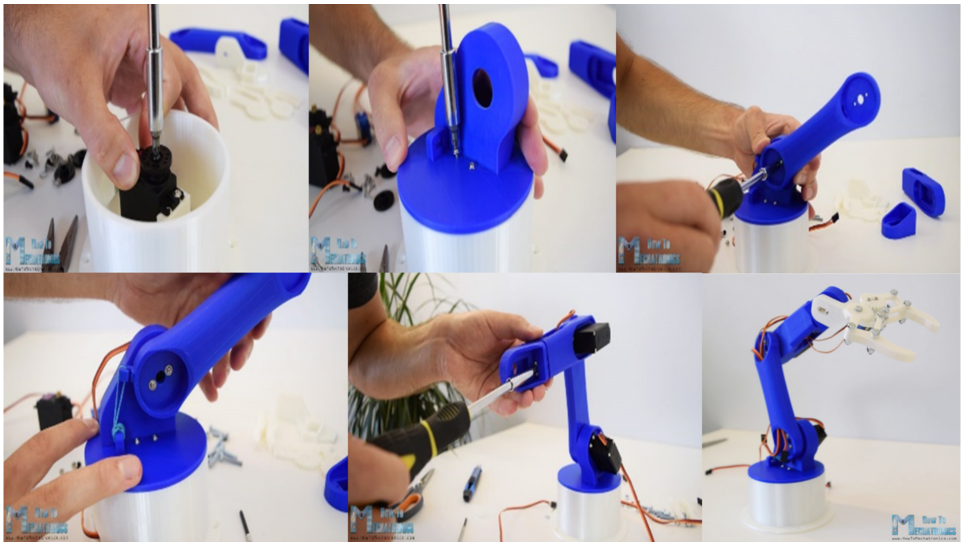

Figure 4.1: Assembling the robotic arm [6].

4.3 Arduino robotic arm code

We have written the Arduino code to control the arm using a Bluetooth module that receives the commands wirelessly to control the arm movement. First, we need to include the Software Serial library for the serial communication of the Bluetooth module as well as the servo library. Both libraries are provided by Arduino IDE. We have defined the six servos, the HC-05 Bluetooth module, the delay between the movement of the servo, and the number of characters the command can have along with the new data flag.

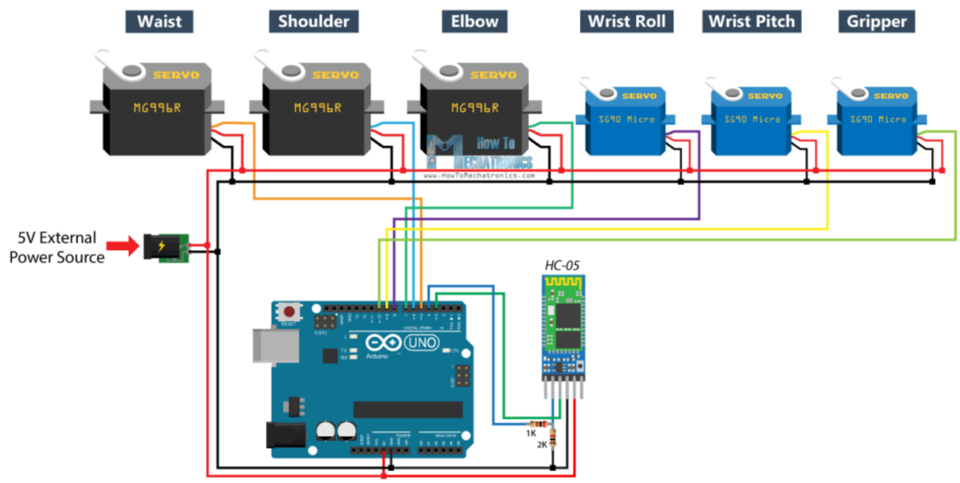

Figure 4.2: Arm control circuit diagram [6].

Arm control setup algorithm

In the setup algorithm we have attached the six servos with their pins in the Arduino. After that we have initiated the arm to the straight arm status by giving each servo with it is a specific angle, then we detach the servos and initiate the Bluetooth with 9600 baud rate.

Algorithm 4 Arm control setup

1: Servo Definition and Constant Declarations
2: Variable Declarations
3: Pin attachment to Servo
4: Servo initial positions

Arm control loop algorithm

The arm control loop algorithm keeps track of the serial port of the Bluetooth module to read the commands from the buffer using (ReadCurrentBuff) algorithm. It is reading the command and saves it. Then it sets the new data flag. Once the new data flag is true the (receivedChars) string is separated properly and sent to (servoSelector) algorithm witch is going to determine the targeted servo and the desired angle. Then the servoSelector sends the desired angle to the target servo which will apply the command on the servo and the cycle is repeated.

Algorithm 5 Arm control loop

```
if Serial.available>0 then
    ReadCurrentBuff()
    if (newData == true) then
        for (from i=0 to 6) do
            angle += receivedChars[i];
        end for
        servoSelector(receivedChars[0],angle.toInt());
        newData = false;
        angle="";
    end if
end if
```

servoSelector algorithm

servoSelector algorithm will receive (servoNum) and (servoPos) then determine the targeted servo based on the (servoNum) and the desired angle based on (servoPos), then send it to the (moveServoFast) algorithm.

Algorithm 6 ServoSelector

```
servoSelector(char servoNum,int servoPos)
Switch(servoNum)
if case 1 then
    moveServoFast(servo01,servoPos,5);
end if
if case 2 then
    moveServoFast(servo02,servoPos,6);
end if
if case 3 then
    moveServoFast(servo03,servoPos,7);
end if
if case 4 then
    moveServoFast(servo04,servoPos,8);
end if
if case 5 then
    moveServoFast(servo05,servoPos,9);
end if
if case 6 then
    moveServoFast(servo06,servoPos,10);
end if
```

MoveServoFast algorithm

MoveServoFast algorithm will receive the (servo), (newPos), and (pin). It will use the (pin) to attach the selected servo based on (servo) to the desired pin connected to the Arduino board. It will then apply the command based on (newPos) using (writeMicroseconds) algorithm from servo library. Then it will detach the servo after 0.5-second delay.

Algorithm 7 MoveServoFast

```
moveServoFast(Servo servo,byte newPos, byte pin)
servo.attach(pin);
servo.writeMicroseconds(newPos);
delay(500);
servo.detach();
```

MoveServoSlow algorithm

MoveServoSlow algorithm will receive the (servo), (newPos), and (pin). It will use the (pin) to attach the selected servo based on (servo) to the desired pin connected to the Arduino board. Then it will determined if the new position based on (newPos) is greater/smaller than the current position. Then it will increment/decrement the position of the servo using (writeMicroseconds) algorithm from servo library to apply the command. Then it will detach the servo after 0.5 second delay.

Algorithm 8 MoveServoSlow

```
moveServoSlow(Servo servo,byte newPos, byte pin)
servo.attach(pin);
int currentPos = servo.read();
If new position is smaller than current position
if (currentPos > newPos) then
    for (j = currentPos down to j ≥ newPos) do
        Run servo down to new position
        servo.write(j);
        delay(delaySpeed);
        defines the speed at which the servo rotates
    end for
end if
If current position is smaller than new position
if (currentPos < newPos) then
    for (j = currentPos up to j ≤ newPos) do
        Run servo up to new position
        servo.write(j);
        delay(delaySpeed);
    end for
end if
delay(500);
servo.detach();
```

ReadCurrentBuff algorithm

ReadCurrentBuff algorithm starts to read from the serial detecting the start marker of the command then it sets the (reciveInProgress) flag to true. Once (reciveInProgress) is true it is starts to read the command and saves it in (recvivedChars) string till it reaches the end marker. Then it makes the (reciveInProgress) false and terminates the (recvivedChars) string and clear the index and set the (newdata) flag to true.

Algorithm 9 ReadCurrentBuff

```
boolean recvInProgress = false;
byte ndx = 0;
char startMarker = '<';
char endMarker = '>';
char rc;
for Serial.available() ¿ 0 AND newData == false do
    rc = Serial.read();
    if (recvInProgress == true) then
        if (rc != endMarker) then
            receivedChars[ndx] = rc;
            ndx++;
            if (ndx ≥ numChars) then
                ndx = numChars-1;
            end if
        else
            receivedChars[ndx] = '\0'; // terminate the string
            recvInProgress = false;
            ndx = 0;
            newData = true;
        end if
    else
        if (rc == startMarker) then
            recvInProgress = true;
        end if
    end if
end for
```

Chapter 5

Non-Technical issues

5.1 Introduction

In this chapter, we will discuss non-technical issues such as budget, and ethical aspects.

5.2 Budget

1. Software
    - Open-source Arduino Software (IDE) to program the Arduino's
    - Paint 3D to design the first version of the Greenhouse design
2. Hardware
    - Arduino Mega 2560
    - 5v Relay Board
    - Water Level Sensor
    - Photoresistor (PGM5516P) [11]
    - Temperature Humidity Sensor (DHT11) [10]
    - Soil Moisture Sensor
    - 40W LED
    - Water Pump
    - Servo Motor (MG996R)[12]
    - Micro Servo Motor (SG90)[13]

- Bluetooth Module (HC-05)
- 5V 2A DC Power Supply

3. Documentation: We used free of cost papers and references from the internet or the Saudi Digital Library through King Saud University Network.

5.3 Ethical aspects

We follow the highest ethical requirements. Our project matches the ethical requirements, All the references that we used are cited and listed at the end of the report. We ensure that the results and methods are without fabrication.

5.4 Institute of Electrical and Electronics Engineering (IEEE) code of ethics

Since it is the largest professional association around the world that includes the highest educated engineering and technology experts, IEEE started to put guidelines and standards to organize and control the large amount of members around the world and their contributions. One of those rules is IEEE Code of Ethics which contains ten rules, and we will discuss how our project is harmonious with this code in all ten rules:

1. To avoid real or apparent conflicts of interest whenever possible, and to disclose them to involved parties when they do exist.

2. To hold utmost safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, to protect the privacy of others, and to disclose promptly factors that might endanger the society or the environment

3. To seek out, accept, and offer honest criticism of professional work, to acknowledge and correct errors, to be truthful and rational in stating claims or estimates based on accessible data, and to credit properly the contributions of others.

4. To avoid illegal conduct in professional activities, and to reject bribery in all its forms.

5. To improve the understanding by individuals and society of the abilities and community implications of traditional and emerging technologies, including intelligent systems.
6. We treat all people justly and never let their religion, race or gender affect our judgment about them.
7. To sustain and improve our technical ability and to embark on technological tasks for others only if experienced by training or experience, or after full disclosure of pertinent constraints.
8. To avoid injuring others, their property, reputation, or employment by misleading or mischievous actions, rumors or any other verbal or physical abuses.
9. To not participate in harassment of any kind, including sexual harassment or bullying behavior.
10. To support colleagues and co-workers in following this code of ethics, to make every effort to ensure the code is endorsed, and to not retaliate against individuals reporting a violation.

Chapter 6

Conclusion

6.1 Summary

The project is intended to provide a solution to automatically tend to plants by controlling water, lighting, humidity, and temperature without the user's input if necessary. We are successful in determining when the plants need watering when to switch the LED and the fan on or off. We developed algorithms to receive commands wirelessly and use it to control the robotic arm movement, and implementing a prototype of the greenhouse using high compression foam boards.

6.2 Future growth

1. Scale up the design for more plants.
2. Improve the environment of plants and tend other plants from out of it is preferred environment.
3. To detect if the plant is ready.
4. To be able to monitor the plants status over the internet.

Bibliography

[1] V. Puranik, Sharmila, A. Ranjan and A. Kumari, "Automation in Agriculture and IoT," 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), Ghaziabad, India, 2019, pp. 1-6, doi: 10.1109/IoT-SIU.2019.8777619.

[2] I. Mat, M. R. Mohd Kassim, A. N. Harun and I. M. Yusoff, "Smart Agriculture Using Internet of Things," 2018 IEEE Conference on Open Systems (ICOS), Langkawi Island, Malaysia, 2018, pp. 54-59, doi: 10.1109/ICOS.2018.8632817.

[3] G. Sushanth and S. Sujatha, "IOT Based Smart Agriculture System," 2018 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, 2018, pp. 1-4, doi: 10.1109/WiSPNET.2018.8538702.

[4] B. V. Pennisi, Growing Indoor Plants with Success, 01-Dec-2012. [Online]. Available: `https://extension.uga.edu/publications/detail.html?number=B1318&title=Growing%20Indoor%20Plants%20with%20Success`. [Accessed: 09-Nov-2020].

[5] D. Lipford, "How to Grow Houseplants in Artificial Light," Today's Homeowner, 14-Nov-2019. [Online]. Available: `https://todayshomeowner.com/how-to-grow-houseplants-in-artificial-light`. [Accessed: 15-Oct-2020].

[6] Dejan, "DIY Arduino Robot Arm with Smartphone Control," HowToMechatronics, 21-Apr-2020. [Online]. Available: `https://howtomechatronics.com/tutorials/arduino/diy-arduino-robot-arm-with-smartphone-control`. [Accessed: 19-Aug-2020].

[7] J. Hussain, ”Automated Color Recognition System for Visually Challenged and Achromatopsia People using Arduino and Mobile App,” International Journal of Advanced Research in Electronics and Communication Engineering, vol. 4, no. 8, pp. 2278-909, 2015.

[8] J. Jimmy , “Project Overview ’ Personal Food Computer,” MIT Media Lab, 01-Sep-2015. [Online]. Available: `https://www.media.mit.edu/projects/personal-food-computer/overview/`. [Accessed: 24-Aug-2020].

[9] G. Burnworth , “Open Source,” FarmBot, 28-Sep-2020. [Online]. Available: `https://farm.bot/pages/open-source`. [Accessed: 04-Oct-2020].

[10] Aosong Electronics Co., ”Digital-output relative humidity & temperature sensor/module”, DHT11 datasheet, [Revised Nov 2020].

[11] Token Electronics Industry Co., ”PGM CDS Photoresistors”, PGM5516P datasheet, [Revised Nov 2020].

[12] Tower Pro, ”High Torque Metal Gear Dual Ball Bearing Servo”, MG996R datasheet, [Revised Nov 2020].

[13] Tower Pro,”Micro Servo”, SG90 datasheet, [Revised Nov 2020].

[14] E. C. Ferrer, J. Rye, G. Brander, T. Savas, D. Chambers, H. England, and C. Harper, “Personal Food Computer: A New Device for Controlled-Environment Agriculture,” Proceedings of the Future Technologies Conference (FTC) 2018 Advances in Intelligent Systems and Computing, pp. 1077–1096, 2018.