

Farm Automation System With IoT Application



Inspiring Excellence

A Thesis Submitted in Partial Fulfillment of the Requirements for the
Degree of
Bachelor of Science in Electrical and Electronic Engineering

By

Mahmudul Hasan: 13121167

Syed Maksud Hossain: 13221006

Mohammad Saad Ur Rahman: 13121066

M. M. Sakib Ullah: 13221011

Supervised by

Dr. Md. Khalilur Rhaman

Associate Professor

Department of Computer Science and Engineering

Fall 2017

BRAC University, Dhaka

DECLARATION

We hereby declare that, the thesis titled “Farm Automation System with IoT Application” is our work. The work has not been presented elsewhere for assessment. Where material has been used from other sources it has been properly acknowledged.

Signature of Author

.....

Mahmudul Hasan

Student ID: 13121167

.....

Syed Maksud Hossain

Student ID: 13221006

.....

Mohammad Saad Ur Rahman

Student ID: 13121066

.....

M. M. Sakib Ullah

Student ID: 13221011

Signature of Supervisor

.....

Dr. Md. Khalilur Rhaman

Associate Professor

Department of Computer Science and Engineering

BRAC University Dhaka

ACKNOWLEDGEMENTS

Firstly, we would like to thank Dr. Md. Khalilur Rhaman, Associate Professor, Department of Computer Science And Engineering, BRAC University, Dhaka; for his supportive guidance and feedbacks for completion of the thesis.

ABSTRACT

Agribusiness is the broadest fiscal portion and accepts a basic part in the general monetary change of a country. The world's population is growing additionally; with that improvement we should make greater sustenance. For the immense number of populace it is extremely hard to guarantee the food. From one perspective to guarantee the sustenance of tremendous populace is troublesome then again it is likewise hard to create sustenance thing for food where less measure of individuals and youthful age is additionally losing their interest on cultivating. The extended age has, as it were, originate from incremental changes in development likewise, economies of scale, however that inclination is going to a level. Standard agribusiness procedures are unsustainable and an adjustment in context is required. The main purpose of our project is to make an advanced agricultural system with the help of IoT application, so that we can easily maintain a farm. We are going to create an automation system which can water the plants of a farm without the help of any human hand. Moreover it will have the options of planting seeds, measuring soil moisture etc. To implement these features, we will work with robotic hand which will be controlled by computer numerical control with the help of Arduino and Raspberry pi. The arm will move with the help of motor and wheels. There will be a metal rail through which the arm can be moved. Two motors will help the arm to move on both X and Y axis and a stepper motor will be used for moving the arm on Z axis. As the arm can move on three dimensional spaces, we can make the arm work with our target purposes. The following implementation procedure of Farm Automation with IoT application, Seeding with the help of seeder with a systematic way, Watering, Temperature measuring, Image processing to identify the soils condition, a big data base, 24 hour automatic monitoring, Collecting and sending data to the users, easy to operate. The main goal of this project is to create a new Agricultural revolution.

Keyword

IoT, image processing, Arduino, Raspberry pi, CNC, GSM

Table of Contents

1) DECLARATION.....	3
2) ACKNOWLEDGEMENTS	4
3) ABSTRACT	5
4) Keyword.....	5
5) Table of Figures	9
6) Introduction.....	10
a) Major crops of Bangladesh:	11
b) Agricultural Land:.....	12
c) Nitrous Oxide (NO ₂) Emissions:.....	13
d) Employment in Agriculture:.....	13
7) Technical Description	15
8) Hardware description.....	16
a) Electronics parts.....	19
i) Microcontroller	19
ii) Raspberry pi camera module	23
iii) The power window motor.....	24
iv) Circular wheel.....	25
v) Servomotor.....	26
vi) Water pump	26
vii) L298N Driver:.....	27
viii) BTS 7960 motor driver.....	27
ix) Cables	28
x) Batteries	28
9) Budget List:.....	29
10) Body.....	30
a) Working in X direction	30
b) Work in Y direction	31
c) Work in Z Axis	32
d) Seeding operation.....	33
e) Watering the plants	34
f) Fertilizing procedure.....	35

g) Work in GSM module.....	36
h) Working procedure of pi Camera	36
i) Working procedure in details	37
11) Circuit Setup:	39
a) Water Pump and Actuator Circuit Setup:	40
b) Seeder:	41
c) Temperature, humidity and Moisture Sensor:	42
12) Software Used:	45
a) Arduino:	45
b) Python:	45
c) PySerial:.....	46
d) Scikit-image:.....	46
e) Flask Framework:.....	47
f) Fritzing:	47
g) Building Web Server:	48
13) Operating Procedure:.....	50
a) Movement Control:	50
b) Soil Digging:.....	51
c) Seeding Process:	51
d) Watering Process:	51
e) Camera Module:	52
f) Image Processing:	53
14) Result Analysis:.....	54
a) Machine’s Movement Test:	54
i) Without Motor Driver (Directly Connected to Battery) along X-axis direction:	54
ii) Without Motor Driver (Directly Connected to Battery) along Y-axis direction:.....	54
iii) With Motor Driver Connected along X-axis direction:	55
iv) With Motor Driver Connected along Y-axis direction:	55
v) Result Analysis:.....	56
vi) Seeding machine test:	56
vii) Result analysis:	56
viii) Fertilizer testing.....	58

ix) Water pump test	58
15) Difficulties Faced:	58
16) Designing the system:	58
17) Designing the Seeder:.....	59
18) Congestion of the machine:	59
19) Running the entire module at a same time:	60
20) Limitations:.....	60
21) Designed for normal weather:	60
22) Seeder Efficiency:	61
23) System is not easily movable:	61
24) PH Level Testing:	61
25) Quite High Budget:.....	61
26) Future Work:	61
a) Rooftop Gardening:	61
b) Greenhouse:.....	62
c) Moving System:.....	62
d) Immune to harsh weather:.....	62
e) Developed Image Processing:.....	62
f) Self Harvesting:	62
g) Solar Power:.....	62
27) Conclusion:	62
28) References:.....	64
29) Appendix:	67

Table of Figures

Figure 1 Major crops of Bangladesh	11
Figure 2 Percentage of agricultural land	12
Figure 3 Nitrous Oxide emissions (%)	13
Figure 4 Employment in Agriculture	14
Figure 5 IoT Application	15
Figure 6 Design of the Project.....	16
Figure 7 Steel picture use in two side track.....	16
Figure 8 Wheels using in X axis	17
Figure 9 Steel wire for pulling	18
Figure 10 Actuator for Z axis	18
Figure 11 Wheels	19
Figure 12 Arduino Uno R3.....	20
Figure 13 Raspberry Pi 3 Model B.....	22
Figure 14 Camera Module.....	24
Figure 15 The power window motor	25
Figure 16 Circular Wheel.....	25
Figure 17 Servo Motor	26
Figure 18 Water Pump	26
Figure 19 L298N Driver	27
Figure 20 BTS7960 Driver.....	28
Figure 21 Cables	28
Figure 22 12V Battery	29
Figure 23 Work in GSM module	36
Figure 24 Working procedure of camera	37
Figure 25 Overall working procedure block diagram.....	38
Figure 26 Motor Circuit X-direction	40
Figure 27 Motor Circuit Y-direction	40
Figure 28 Connection for Actuator and Water pump.....	41
Figure 29 Seeder connection setup	42
Figure 30 Temperature Sensor Setup	43
Figure 31 Moisture Sensor setup	43
Figure 32 Circuit Setup in the field.....	44
Figure 33 Circuit Setup in the field for testing	44
Figure 34 Website without Style Sheet.....	48
Figure 35 Complete Website	49
Figure 36 live streaming and soil types result in the server.	53
Figure 37 Seeder Test Result.....	57

Introduction: Bangladesh is an agricultural country blessed with a lot of fertile land having three cropping seasons. It means we can use our land throughout the year for the purpose of cultivation which covers a major part of our GDP as well as one of our basic needs, food which is one of the most primary concerns for the people worldwide. Undoubtedly there is no substitution of food and better farming process is the only way to increase the production of food. In order to produce sufficient amount of food we must have to find some ways to make agriculture easier, time saving and digital. Once, more than sixty percent of our total occupied people were directly and indirectly involved in farming. As time passes by, we can now see a completely different scenario. Nowadays, in our country people are more centralized towards towns because farming is not an easy task and farmers don't get enough earnings through farming. Recent statistics shows us that the growth rate of farmers in Bangladesh has slowed down over last decade and causing the fall in rice production growth. Here we have come up with an idea where farming can be considered as less complicated task. People can easily monitor cultivation process from time to time even when they are not available in the field. We are living in the modern era where everything is getting digitized and farming is not exceptional among them but here we are proposing something very exceptional which can farm itself and can give all kinds of information which is very essential for a farmer. It is not just a machine or robot, it itself is a farmer.

First of all, we must mention that countries like U.S., China, and Japan are already spending a lot of money and time to make farming a lot easier though they don't have the fertile land like Bangladesh. Libin, Zhang, et al. [1] have mentioned in their article about the various kinds of robots like grafting robot, transplanting robot, spraying robot, mowing robot and various harvesting robots which are being used in China for the purpose of agriculture. Moreover, Yaghoubi, Sajjad, et al. [2] also showed in their research how fast the use of agricultural robot is growing every year and around 13% of the total industrial robot is belonged to the agricultural sector. As we can see, developed countries are using automation system for the increased efficiency of their crops whereas developing country like ours are way behind these automated system. Therefore, we have become inspired to do such project.

Major crops of Bangladesh:

More than 80 different kinds of crops are currently grown in Bangladesh. The main crop of our country is rice which has over 25 varieties and is grown all year round. Around 76 percents of total crop production is covered by rice. Other types of crops are mainly potato, jute, wheat, maize etc. The table below shows the crop production from the year of 2004 to 2007.

Area and production of the major crops (Average of 3 years: 2004/2007).

Crops	Area (000 m ton)	Production (000 ha)
Rice (total)	10,495	26,335
Wheat	479	816
Maize	105	593
Potato	324	4728
Jute	404	819 (bales)
Sugarcane	153	5910
Pulses (as group)	350	289
Oil seeds (as group)	344	643
Spices and condiments	324	1196
Tobacco	31	41

Source Yearbook of Agricultural Statistics of Bangladesh, 2007.

Figure 1 Major crops of Bangladesh

Crops are grown in three cropping seasons during a year which are known as Rabi (Dry season), Kharif-I and Kharif-II. Production of rice dominates the cropping pattern. Wheat, potato, sugarcane, pulses are the crops of Rabi season. In Kharif-I, jute and Boro rice is cultivated whereas Aman rice is cultivated in Kharif-II. Different crops are grown in different places of the country according to the types of soil, flooding condition and rainfall.

Agricultural Land:

According to the World Bank Collection of Development indicators, the agricultural land of Bangladesh was around 70% of the total land in the year of 2014. From the chart provided by The World Bank, we can see that the percentage of agricultural land has reduced drastically after the year 1991.

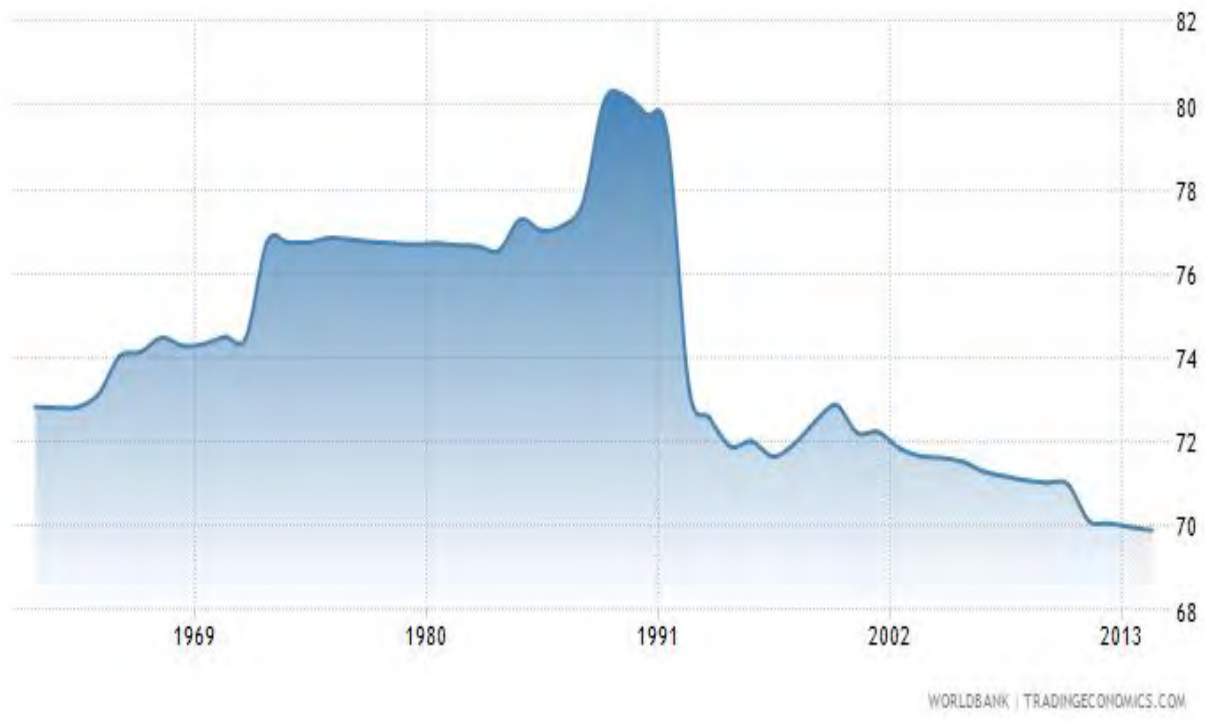


Figure 2 Percentage of agricultural land

Among these lands, only 6.38% of our lands are permanent cropland which is very low. The main reason behind decreasing agricultural lands is large number of population growth. As the population of our country increasing, the economy is developing too. Therefore, urbanization is taking places. Moreover, infrastructures like mills, educational and religious establishments, roads are taking the place of those agricultural lands. As a result, the shortage of food is taking place which is causing us to import more from the foreign country. Every year, around 2 to 3 million tons of foods are being imported by our country.

Nitrous Oxide (NO₂) Emissions:

When using fertilizers to the land, gas is emitted from it in several forms and nitrous oxide is the most common emitted gas among them which causes global warming. Excessive use of inorganic fertilizer can do real damages to our environment. According to The World Bank, the emission rate of nitrous oxide was 82.93% in the year of 2008 which was increased comparing to the previous records as shown in figure below.



Figure 3 Nitrous Oxide emissions (%)

Employment in Agriculture:

With the improvement of socio-economic condition, the people of our country have shifted their job to the town especially to the capital. Therefore, the employment rate in agricultural sector has decreased a lot. 47.48% of the total workable population was involved in agriculture as the record shows in the year of 2010 whereas; it was more than 65% in the year of 1991.

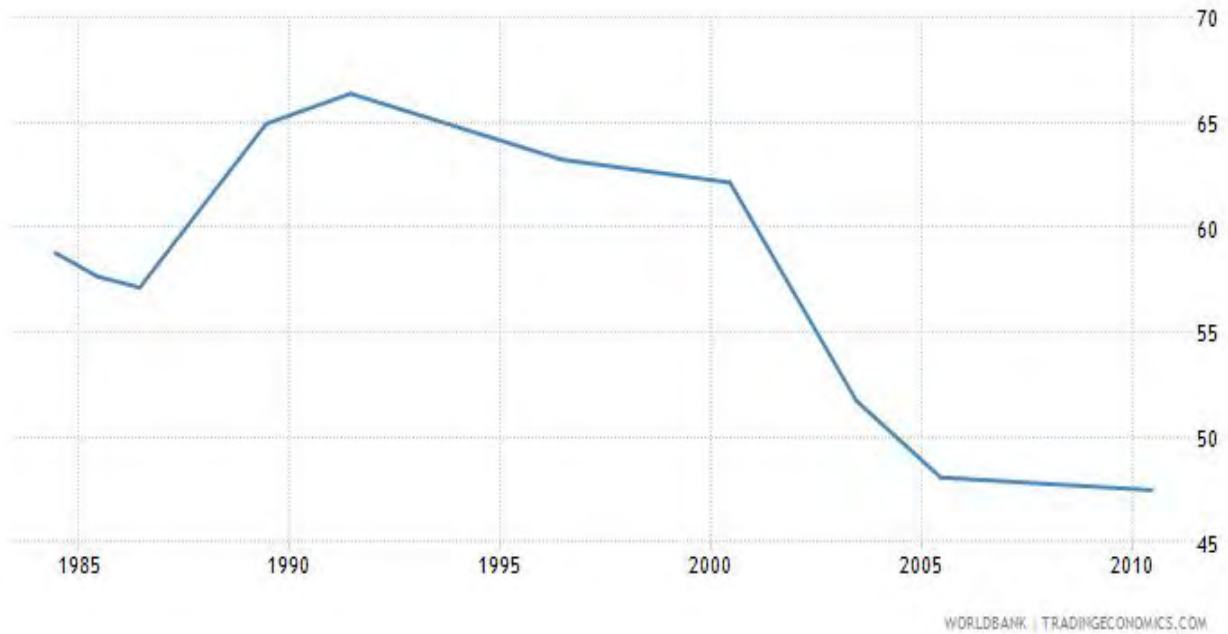


Figure 4 Employment in Agriculture

IoT:

Internet is all around us every day. Internet is basically a network of devices, and they are all part of the same network. So, IoT or internet of things means a network of internet-connected objects able to collect and exchange data using embedded sensors and devices. It basically means that a device with an ip address anywhere in the world can be connected to any other device with an ip address, anywhere in the world using the interconnected network of networks also called the internet.



Figure 5 IoT Application

IoT devices are a part of the larger concept of home automation, also known as domestic. Large smart home systems utilize a main hub or controller to provide users with a central control for all of their devices. These devices can include lighting, heating and air conditioning, media and security systems. And, that is where our automated farming machine with IoT comes in.

Anyone from anywhere with the ip address of the device in our case the raspberry can access the information in it. They can monitor the sensors, view live streaming through the camera connected to the device, give commands to the motors and change settings as they like.

Technical Description

Our machine is an automated precision farming machine which is controlled automatically and manually, software package designed from the ground up with today's technology. It is quite similar with today's 3D printers and CNC machine, our machines hardware employs liner guides in the X, Y, Z direction which allows the tools such as seeding, watering, fertilizing, sensing temperature, humidity, moisture and precisely positioned and gives information of plants and soils. Our system is automatically controlled for seeding, watering and fertilizing; the whole thing is numerically controlled by using Arduino and Raspberry-pi. Other features are software base like giving temperature, humidity and moisture information of soil. The total hardware system is very simple easy to operate and we can also operate it remotely by using web based software. We are using GSM module to send text to the client for informing the present condition of the harvest. The website contains full of data about various type of plant which is suitable for the land. The machine can also operate from the website manually and there is an option of live video steaming. For today's world the machine is very handy and easy to operate

for farmers to cultivate. This technology is very much standard level for today's world. The full things are explained in the other section in this paper.

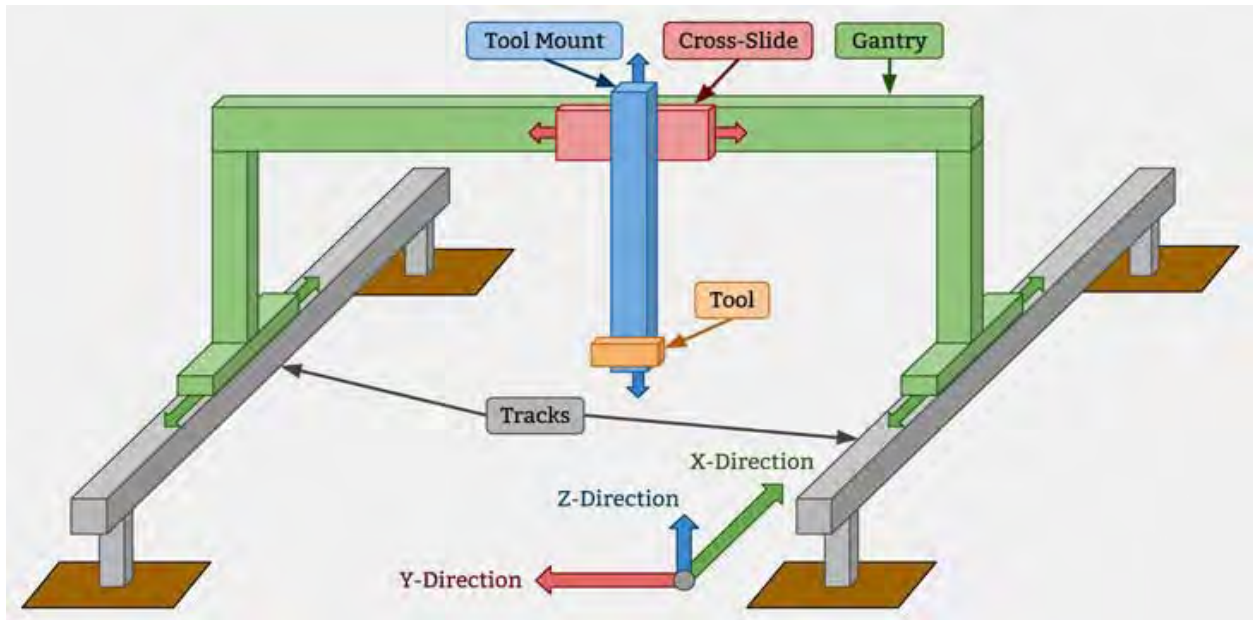


Figure 6 Design of the Project

Hardware description

Ours system hardware is very similar to today's 3D printing machine and CNC machine. The machine is working in three different directions like X, Y, Z direction. For X direction we used two tracks for move on. This tracks are made of steel the figure shows the steel.



Figure 7 Steel picture use in two side track

The tracks are fixed in the ground for allowing the system to work in different direction in a great manner. There are some reasons behind we use this steel to make the track. Like

- Low cost
- Smooth going
- Rust free
- Paths are in a line.

Inside the steel we use some wheels to move the tools smoothly. This wheels are attached the Y axis tools, so that the Y axis tools can easily moves in X direction. These wheels are actually used for curtain holding. We use this thing in the track for pulling the y axis tools. Figure 8 will



Figure 8 Wheels using in X axis

show the picture of the wheel. We welded the wheels with Y axis material and put them into the X axis track. We use a steel wire to pull the tools in X direction. Figure 9 will show the picture of the wire.



Figure 9 Steel wire for pulling

The primary structure of Y axis is an upside down square U shape. At each end of U shape the wheel is welded for smooth running in X axis track. The Y axis track also made of steel. And we use same materials to make the track for running the Z axis tools. Like steel track, wheels and steel wire which are made of aluminum for protecting from rust.

For the Z axis collaboration we use actuator which is properly welded with u axis. It's welded properly that there is no barrier in running in Y axis. An injector is properly tied with the lower part of the Actuator for digging the soil.



Figure 10 Actuator for Z axis

The Robot is attached with four wheels to move into the field. Our robot is mobile type. You can fix in any where you want or you can travel anywhere you want. The whole body lies in steel made frame which has four wheels in its four sides. Figure 11 will show the wheels



Figure 11 Wheels

The full frame is very much handy and it's geometrically very much significant.

The seeding machine is made of plastic bottle, which consist of servo motor and tube. We attach the seeding machine with the lower part of the actuator with a very significant way.

Electronics parts

Microcontroller

Arduino UNO

An Arduino UNO microcontroller, pictured in figure 12 is used in our project. Arduino is open source computer hardware and Software Company that has a huge project and user community that designs and utilizes microcontroller based development boards and Arduino Uno is one of

them which is based on ATmega328. It has 14 digital input/output pins where six of them can be used as Pulse Width Modulation (PWM) outputs. It also has 6 analog inputs, an In Circuit Serial Programming header, a 16 MHz crystal oscillator, USB port, a power jack and a reset button. The board can be powered via USB power cable or an external power supply. The operation voltage of Arduino is from 6 to 20 volts. The Atmega328 has 32 KB of memory. We use it to control the glass motors, water pump, servo and sensors. This thing is chosen for low cost, general ability, the expansive learning resources available, and the DIY community already using the platform. Arduino programs are written in C language, which is known to all of us. The main thing it is open source.

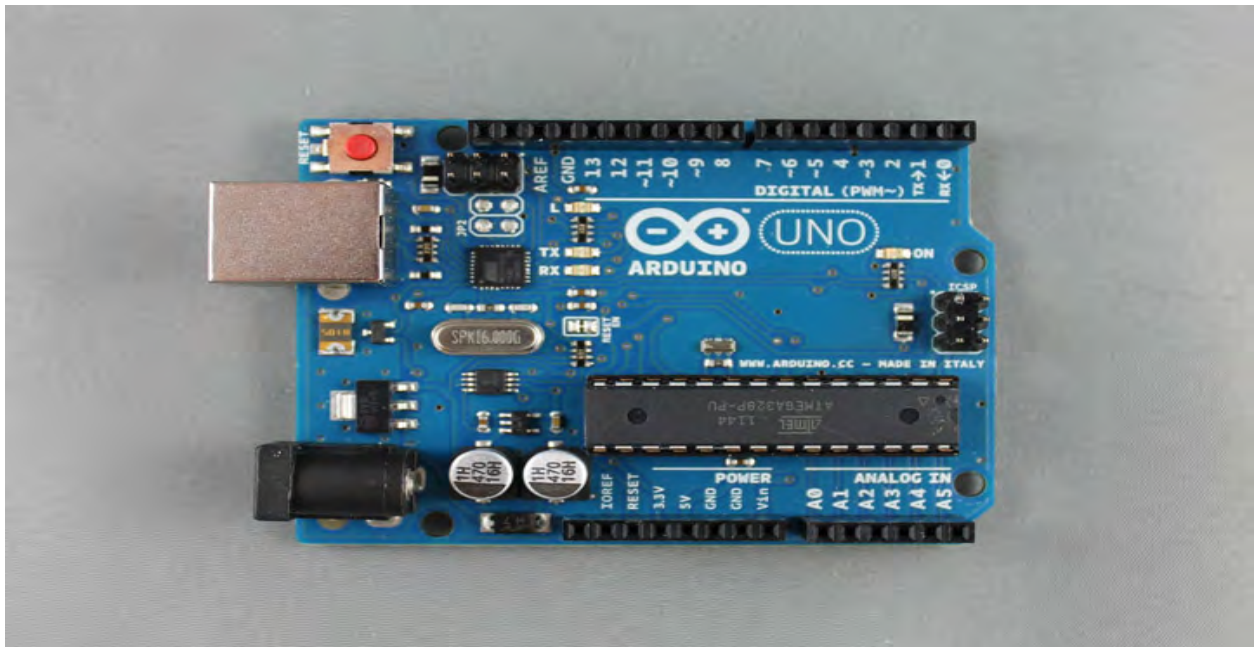


Figure 12 Arduino Uno R3

Specifications of Arduino Uno R3:

Microcontroller	ATmega328
Operating Voltage	5V
Recommended Input Voltage	7-12V
Input Voltage Limit	6-20V
Digital I/O Pins	14(6 of them provide PWM output)
Analog Output Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32KB (0.5 KB is used by boot loader)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Raspberry Pi 3:

Raspberry Pi is a tiny sized, programmable, efficient, high performer single board computer which carries a SOC, USB ports, processor chipset, RAM, Ethernet port etc. As a result, it is considered as a small computer which is very cheap to acquire. Moreover different kinds of programming languages are supported by Raspberry pi. One of the important programming languages Python can be used in it in a very easy and understandable way because it can function automatic memory management and so, it can compile codes in less line. It also supports open

source operating system and open source apps. In addition, Raspberry Pi supports external hardware like Camera Module, Gert-board, Component Modular Kit and HAT board. Through this external hardware Raspberry Pi can be connected with a lot of other hardware. It is not only cost efficient but also energy efficient. We use raspberry pi to build a website. Our whole system is IoT base. So we have to create a website for controlling the machine manually from remote place. So we use raspberry-pi to run this application. Figure 8 will show the Raspberry-pi.

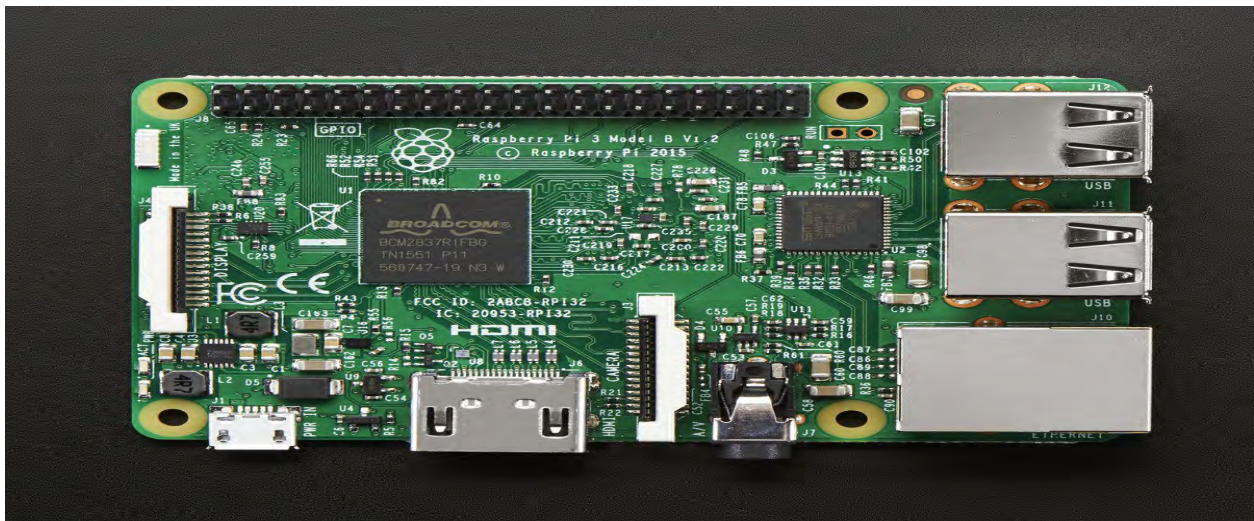


Figure 13 Raspberry Pi 3 Model B

Specification	
1.RAM	1GB SDRAM @400MHz
2.Storage	MicroSD
3.Processor Speed	Quad Core @1.2 GHz
4.Processor Chipset	Broadcom BCM2837, 64Bit Quad Core

	Processor powered Single Board Computer running at 1.2Ghz
5.USB 2.0	4* USB Ports
6.Max Power Draw/voltage	2.5A @5v
7.GPIO	40 pin
8.Ethernet Port	Yes
9.Wifi	Built in
10.Bluetooth LE	Built in

Raspberry pi camera module

The Raspberry Pi Camera Module v2 is a high quality 8 megapixel Sony IMX219 image sensor custom designed add-on board for Raspberry Pi, featuring a fixed focus lens. It's capable of 3280 x 2464 pixel static images, and also supports 1080p30, 720p60 and 640x480p60/90 video. It attaches to Pi by way of one of the small sockets on the board upper surface and uses the dedicated CSI interface, designed especially for interfacing to cameras.

- 8 megapixel native resolution sensor-capable of 3280 x 2464 pixel static images
- Supports 1080p30, 720p60 and 640x480p90 video
- Camera is supported in the latest version of Raspbian, Raspberry Pi's preferred operating system

The board itself is tiny, at around 25mm x 23mm x 9mm. It also weighs just over 3g, making it perfect for mobile or other applications where size and weight are important. It connects to Raspberry Pi by way of a short ribbon cable. The high quality Sony IMX219 image sensor itself has a native resolution of 8 megapixels, and has a fixed focus lens on-board. In terms of still

images, the camera is capable of 3280 x 2464 pixel static images, and also supports 1080p30, 720p60 and 640x480p90 video. We use this camera for our video streaming work. We first connect with the server the video streaming starts automatically. We can observe the whole system by using this camera.

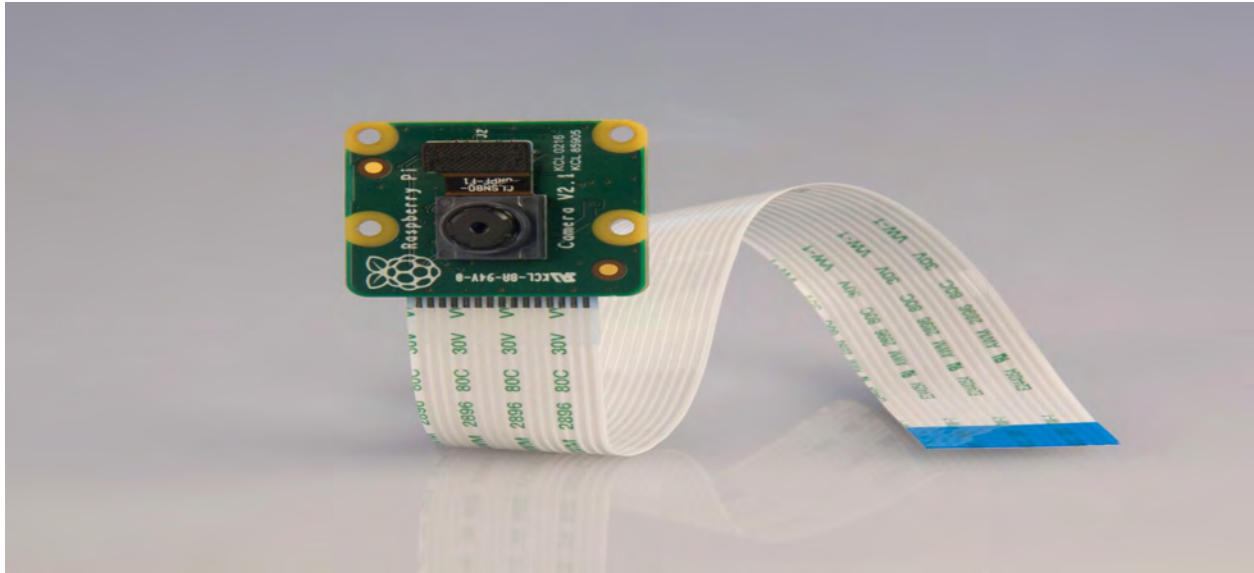


Figure 14 Camera Module

The power window motor

The power window motor has four mounting hole positions. There is a working voltage of 12 volts DC current. The unit is waterproof and ISO 9001 certified. The no-load speed or speed when no torque is applied to the motor shaft is 95 rotations per minute (rpm) and the no load current is less than 1.5 amperes. The stall torque or minimum torque needed to completely stop the motor shaft from rotating, or stall the motor, is less than 8 units or pound-feet (N.m) and the stall current is less than 20 amperes. We use three power window motor in our robot. Two motor is working in X direction and one is for working in Y direction. This motor is very power full motor. We can see this motor in this figure 15.



Figure 15 The power window motor

Circular wheel

We made one kind of circular wheel, which is attached with the head of the motor. So when the motor is rotating it can pull the steel wire. This circular wheel is made of hard plastic and it is attached with the motor. The figure will show the circular wheel.



Figure 16 Circular Wheel

Servomotor



Figure 17 Servo Motor

A servomotor is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. Servomotors are used in applications such as robotics, CNC machinery or automated manufacturing. We use this servo motor in our seeding machine. After digging the soil there is delay on that time the servo decrease its position from 120° to 55° . Then the seed drops from the hole.

Water pump

This water pump is used for watering the land. Usually it is 12V DC motor which actually use for small project. We control it by using motor driver. We use 1m pipe which is attached with the water pump. The figure will show the scenario.



Figure 18 Water Pump

L298N Driver:

Figure 19 L298N Driver

Operating a motor requires a driver because microcontrollers cannot send signal directly to the motor for working properly. Therefore, the signal first goes to the motor driver so that the motor driver can work as a switch and can control the motor according to the signal. L298N is one of the most popular motor drivers for controlling two motors at a time. It is developed to drive inductive loads like relays, DC motors, stepping motors. L298N has 2 amperes current capacity per channel and it can support up to 46 volts and it has maximum power dissipation of 25 Watts. We use this motor driver to control the servomotor and the actuator. Out A and out B is controlling the actuator, out C and out D is controlling the servomotor.

BTS 7960 motor driver

The drive uses an H-bridge driver module that composed by Infineon power drive chip BTS7960, with overheating and over current protection. Double BTS7960 H-bridge driver circuit, with strong drive and braking effect, use 74HC244 chip effectively isolate the microcontroller and the motor drive.

- Input voltage: 6V-27V
- Model: IBT-2
- Maximum current: 43A
- Input level: 3.3-5V

- Control mode: PWM or level
- Duty cycle: 0-100%

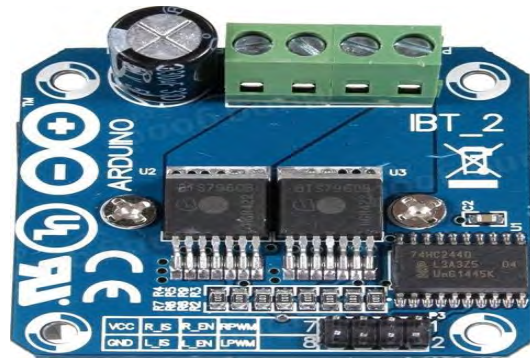


Figure 20 BTS7960 Driver

Here is the figure of motor driver. We use two this kind of motor driver to control the three windows motor. The motor driver is very much useful to control these high power motor.

Cables

We use various types of cables for making connection between the materials. We use jumper cable, male female wire, BRB cables and 1m cables.



Figure 21 Cables

Batteries

We use two 12v DIAMEC batteries as power supply. This is a very much efficient batteries. Its rated capacity 9AH, weight 2.5 kg. It is rechargeable.



Figure 22 12V Battery

Estimated Budget:

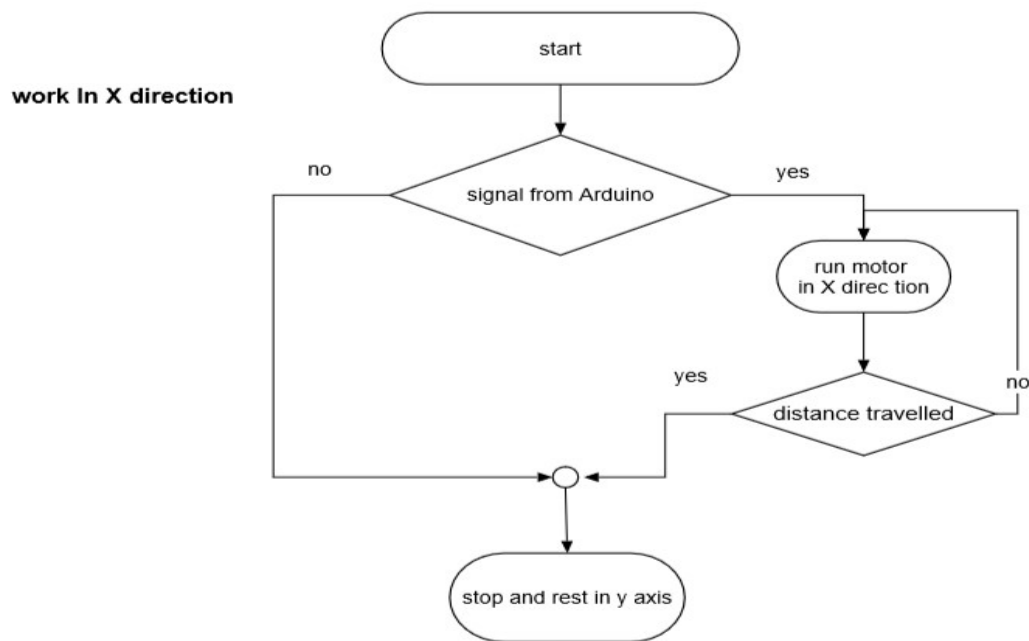
Budget List:

Item	Quantity	Cost
BTS7960	3	3,900
L298N	2	600
Buck Booster Module	1	150
Wheel	3	900
Water Pump	1	1,830
12V Battery	2	1,600
GSM Module	1	1,800
Moisture Sensor	1	600
Arduino Mega	1	1,000
Raspberry Pi	1	3,500
Raspberry Pi Camera Module	1	4,600
Wire, Pipe, PVC etc		400
LiPo Battery	1	1,400
Frame, Body & Base		5,300
Servo Motor	1	400
Breadboard	3	360
Seed, Soil, Fertilizer		400
	Total	28,740

Body

Let us first introduce about our project. Usually our machine is almost like a 3D printer or A CNC machine. Our machine is using for cultivation. Like CNC machine is working in X, Y, Z direction, our machine is also works in the same direction. Usually 3D printers and CNC machine work in X Y Z direction for 3D printing of a material. We use this machine for precision agriculture. The main objective of this machine is seeding, watering, fertilizing and using sensors for knowing about the soil condition. Below we are describing the working procedure step by step-

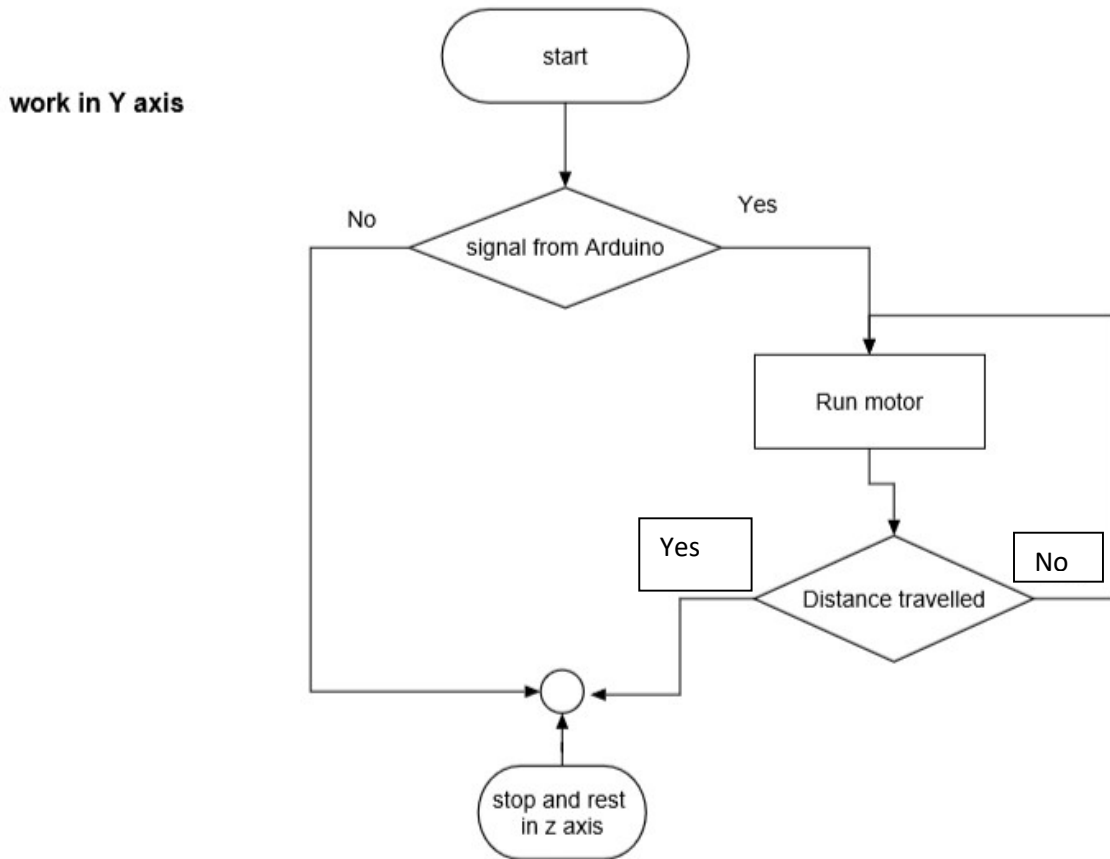
Working in X direction



For moving in X direction we use two power window motor. This is controlled by the Arduino UNO R3. A circular ring is attached with the head on the motor to pull the chain and moving in X direction. After initialized the port, pins and the power the Arduino sends signal to run the motor. If the Initialization is correct then the motor receive the single from Arduino. If the motor get proper signal then it will decide about the distance travelled. In X direction the machine will

work on. If the total distance is 8 feet then the machine is stop per 2 feet in its distance to work in Y and Z direction. By using motor driver the motor will run smoothly.

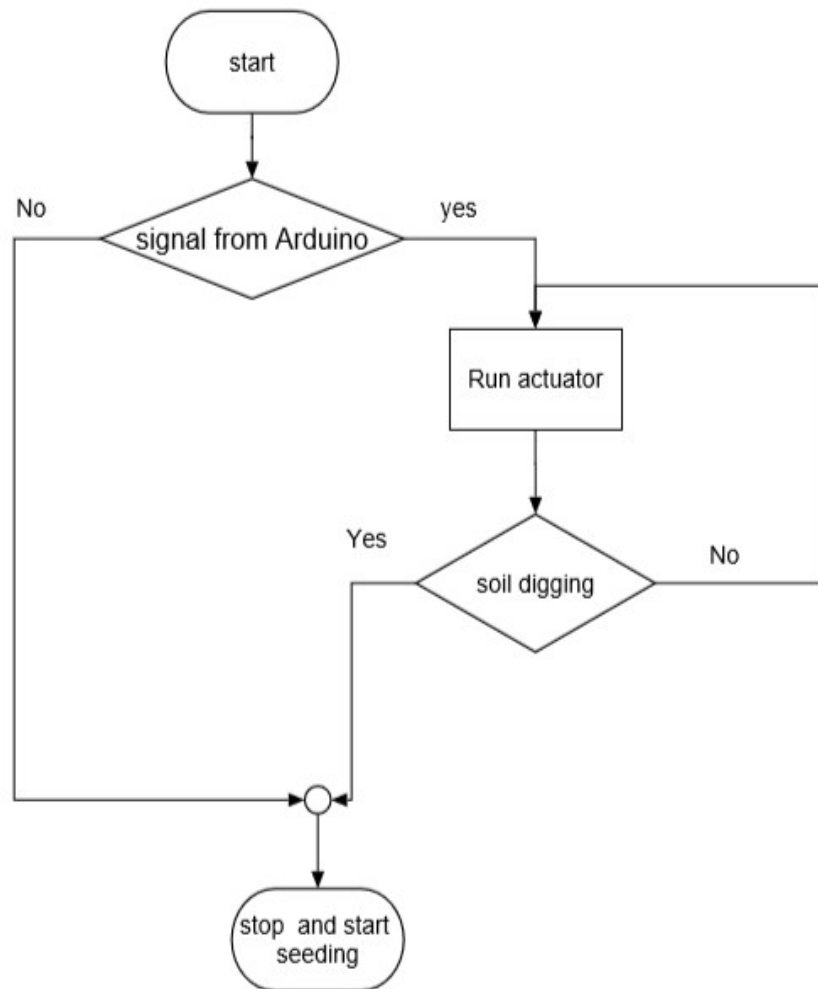
Work in Y direction



When there is a delay in X direction then the machine work in Y direction. For moving in Y direction there is window motor to pull the chain and move the Z axis materials. Same as before when the motor get the signal from the Arduino and if the signal is correct and the command is correct then the motor is working in Y direction. If the total length is 5 feet then it makes a delay each and every 1 foot in its way to work in Z axis. Thus the machine starts works in Z Axis. Here we also use a motor driver to control the speed of the motor.

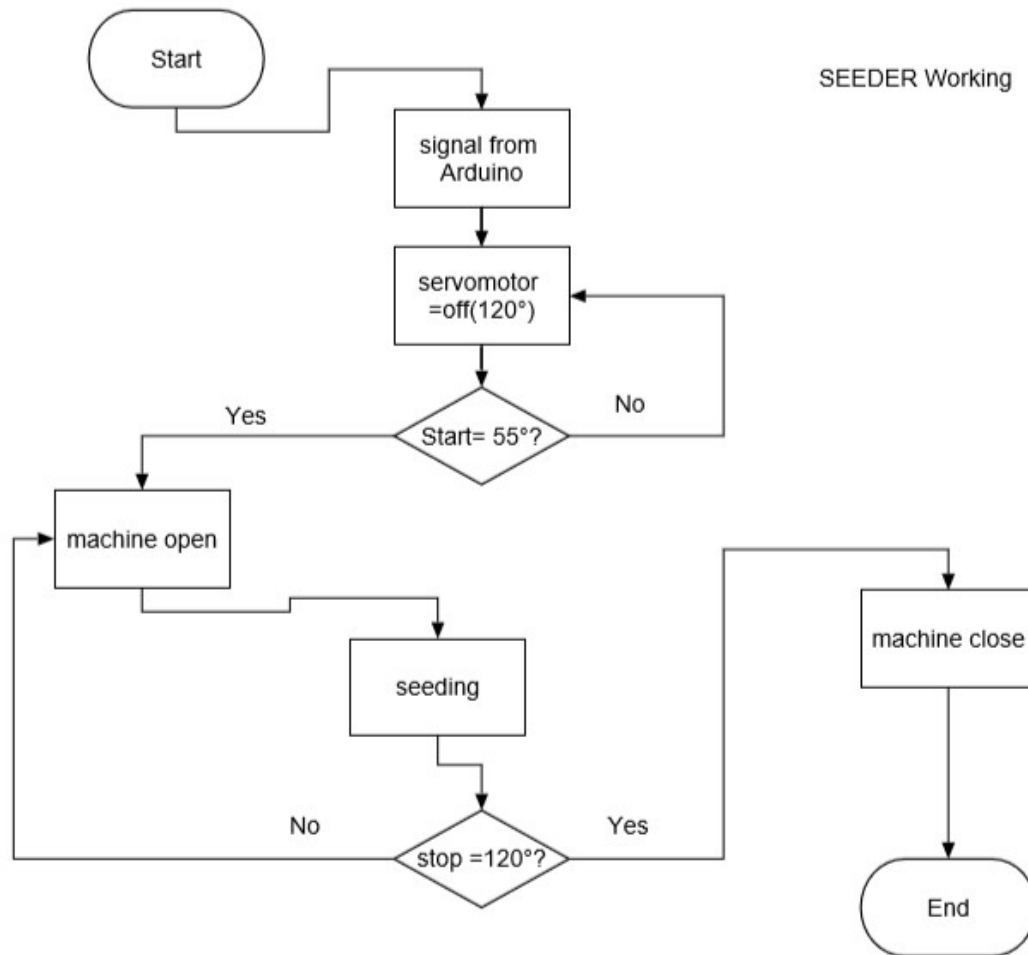
Work in Z Axis

work in Z direction



When there is delay in y axis the work of Z axis starts. To work in Z axis we use an actuator. Like before the Arduino sends the signal to the actuator. 1st it's initialized its pin and port then it sends command which is received from the client. When the signal is received by the actuator then if the signal is correct, it starts to dig the soil. After digging the soil there is delay initialized by the client for seeding. And the steps continue in its whole journey. Then the machine starts the seeding operation.

Seeding operation

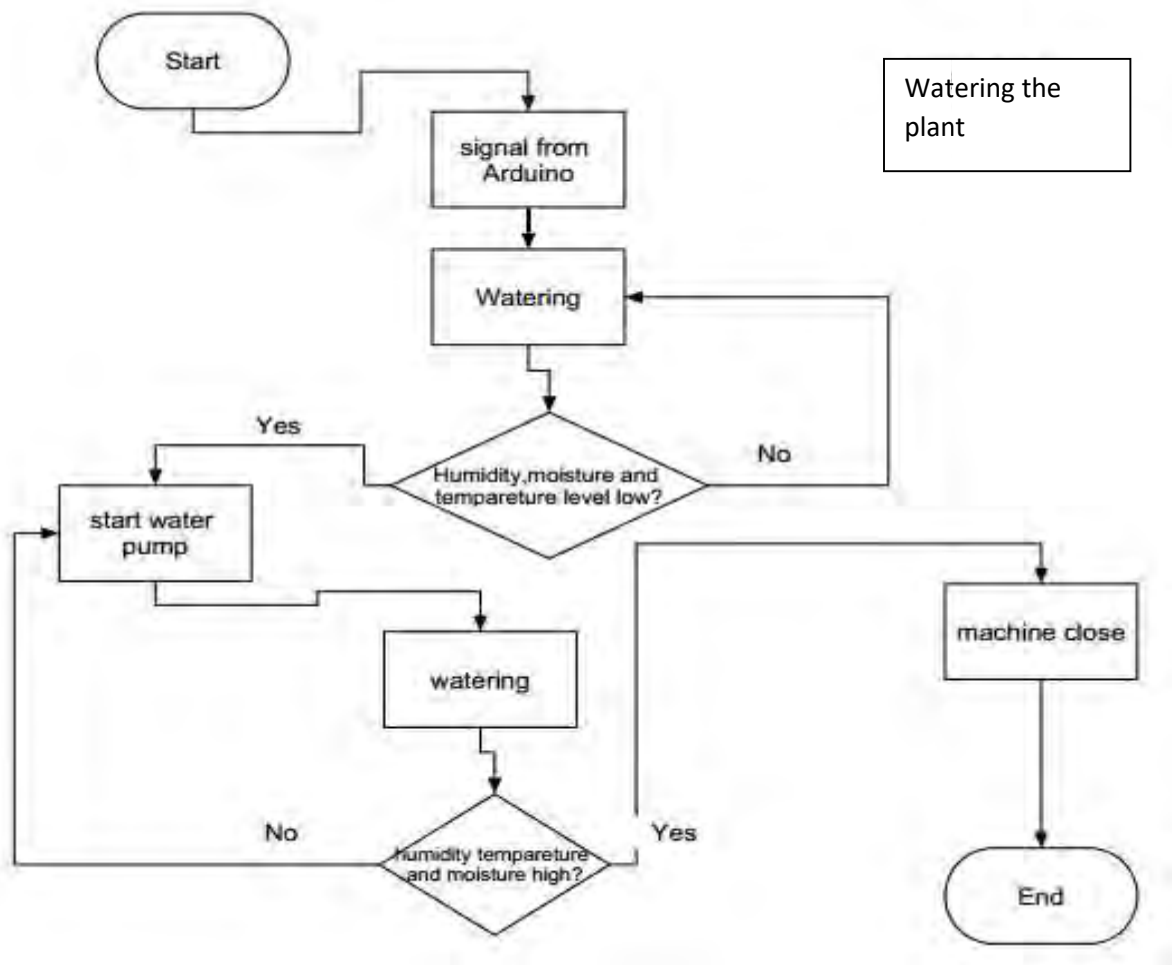


When there is a delay after digging the soil the seeder starts working. The seeder is made by a servomotor, a plastic bottle and a tube. There is gate open close system for seed drop. Like when the servomotor is in 120° position then it is in close position. And when it is in 55° position then it is in open position. It works in a loop so that seed can drop properly. At the early stage the door is closed means servo is in off position when it receives command from Arduino then decreases its position to 55° . Then the door open for some time like for a seed drop and it goes again into 120° position for close the door. This way the seeder is working in a loop. To control the servomotor we use a motor driver.

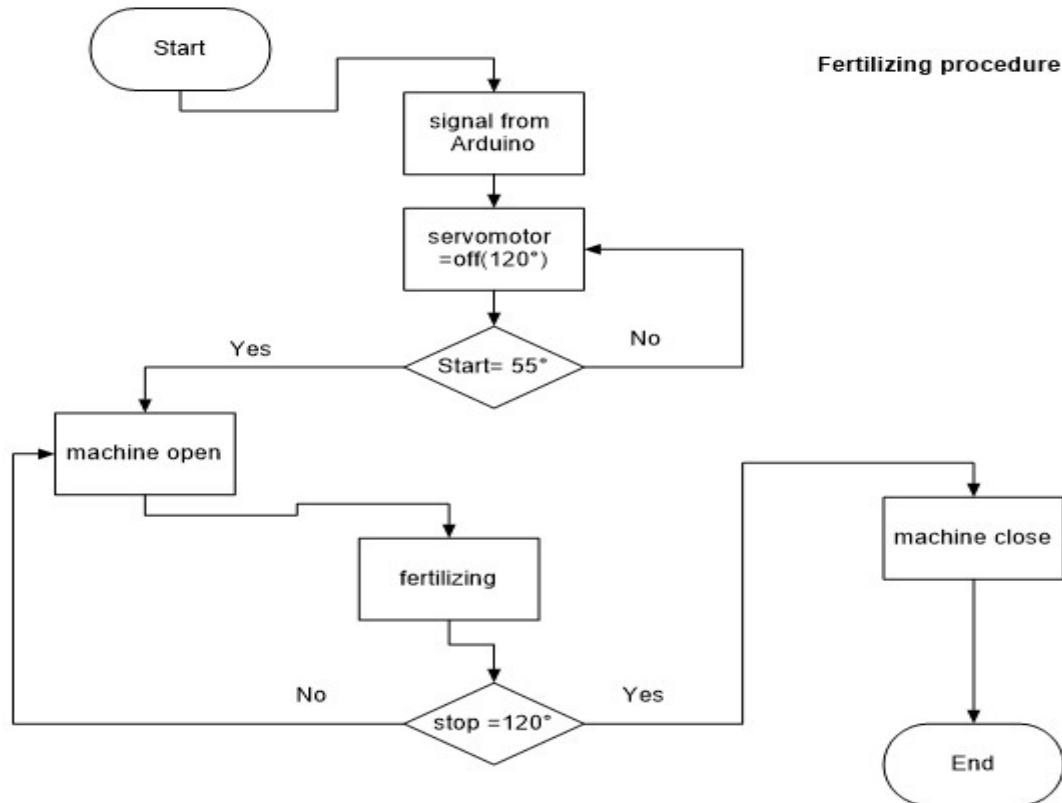
In this way the whole system is operating. Like 1st doing the X axis steps, 2nd Y axis steps, 3rd Z axis steps and finally seeding steps starts. Then the Arduino take the update of temperature, humidity and moisture of soil. If the humidity, moisture and temperature level is low then it sends the command of watering the plant.

Watering the plants

Arduino sends command to the temperature, humidity and moisture sensor to collect data about the soil. Then it verifies that the level of temperature, humidity and moisture is high or low. If high then it does nothing but if low then it sends command to the water pump to starts watering. Then the machine works in X, Y, Z direction and water the place where the seeds are dropped.



Fertilizing procedure



Here we use the seeder machine to give fertilizer. When the servomotor is in 120° position then it is in close position. And when it is in 55° position then it is in open position. It works in a loop so that fertilizer can drop properly. At the early stage the door is closed means servo is in off position when it receive command from Arduino then decrease its position to 55° . Then the door open for some time like for fertilizer to drop and it goes again into 120° position for close the door. This way the fertilizing procedure is working in a loop. To control the servomotor we use a motor driver.

Work in GSM module

The collecting data is shown to the built in website and after completing each and every loop there is a text message sent to the user. This text contents the information about the present condition of soil. This whole procedure is processing by using GSM module.

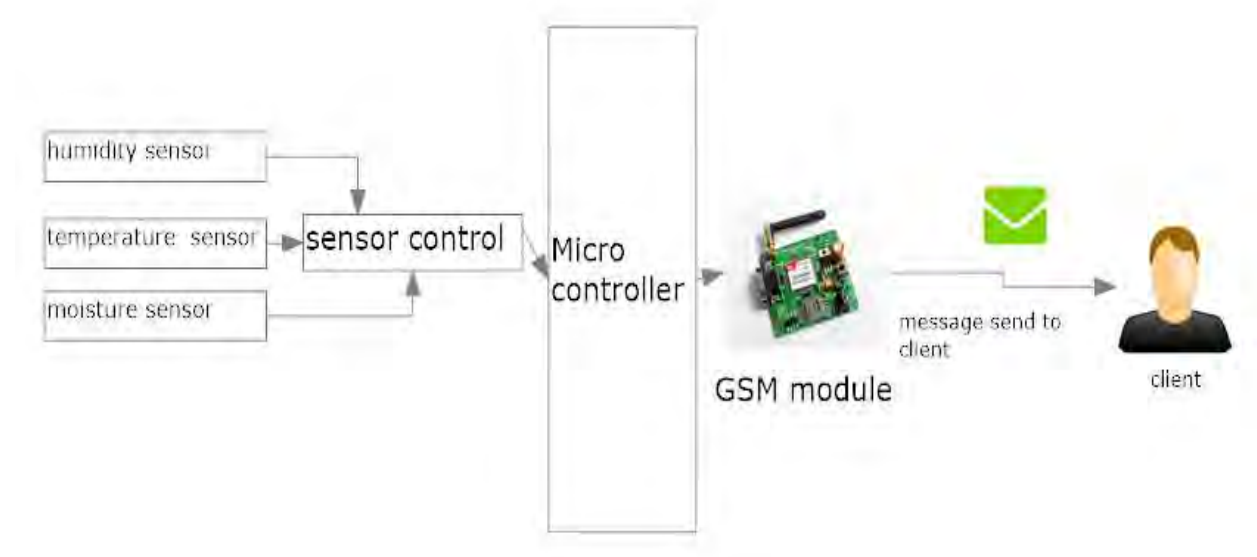


Figure 23 Work in GSM module

Firstly, all data like value of humidity, temperature and moisture is controlled by sensor controller and the data is going to the micro controller. The micro controller will collect the data after completing the cycle and using GSM module it sends the data to the client's mobile. The client will receive the data by his phone from any remote place.

Working procedure of pi Camera

We built a website for monitoring the machine. There is also a video monitoring option. Here we discuss the procedure of video steaming. The whole thing is controlled by Raspberry pi 3.



Figure 24 Working procedure of camera

Raspberry-pi has some built in website where we can add our feature to work on. Firstly we have to login into our web server by connecting raspberry pi using wifi or any reliable network system. Then after login to the website the video steaming is automatically starts. Client can easily monitoring his/her land and can have an idea about his/her cultivation. Condition is client and the device should use the same network. By using the camera user can easily manually control the robot direction. We can also manually watering, seeding and fertilizing the soil. We observe the plant condition by using this camera.

Working procedure in details

In figure all have clear idea about our overall system. We built robot which can seeding, fertilizing and watering the soil. It is CNC base machine means its overall work is in X, Y, Z

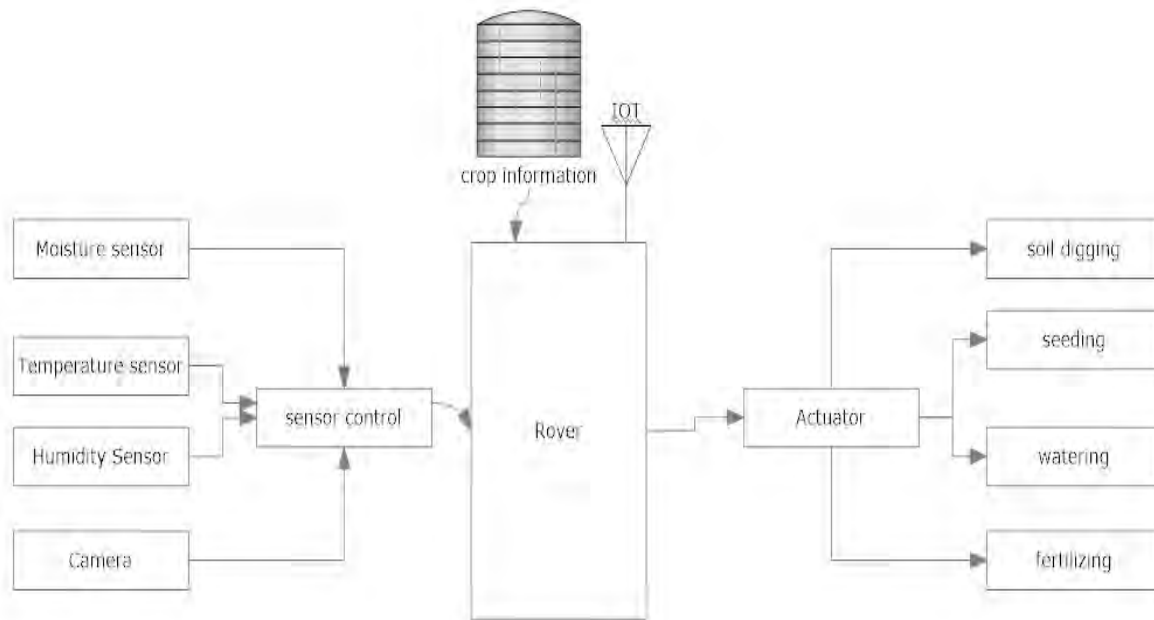


Figure 25 Overall working procedure block diagram

direction. We are using Arduino and Raspberry-pi as Micro controller. There are some sensing features. Like this machine can sense the value of temperature, humidity and moisture of soil by using the sensor. These sensors are controlled by a sensor controller. Our machine is IoT base so we have built a website which contains the sensors data in different tabs. This web site also contains the crop information for the help of the farmer. Various types of crop information are available in the website. Which one is good for which kind of soil user can easily identify these things. This website is built using raspberry pi. It also contains the two very important features like user can control the robot manually by using same network. It has control switch. Robot can be off and on from the website. One more important issue is that it can deliver live video steaming to monitoring the situation. Once the user login to the website automatically live steaming option is on. If the user want to manually operate the robot this operation is controlled both by Raspberry-pi and Arduino. The raspberry pi mainly uses to control the website. So when we want to operate it from the website the raspberry pi sends the command to the Arduino and then the Arduino will follow the command. Our machine is working in three different directions for different purpose of working. Three different directions are controlled by the Arduino. In X direction there are two tracks to move on. Y axis there in one tracks and in Z direction we use an

Actuator to work. Each and every axis has some specific work. The main work is in Z axis. Here we attached the seeding machine and pipe with the Actuator. First the machine will go into x direction take some delay work in Y direction again take some delay and finally work in Z direction and take some delay. In this delay it digs the soil and wait for a while, by this time the seeder starts working and seeds drop. In the same manner the machine is watering and fertilizing the plants. By this way it completes its journey. After completing each cycle it sends text about the present condition of the field to the user. For doing this operation we are using GSM module. This module is controlled by the Arduino. Each and every work is updated in to the web site. Thus our full system works and reduces human efforts.

Circuit Setup:

Motor Circuit of X-axis direction: Two motors are paralleled and connected to the battery through BTS7960 motor driver. Pin LPWM is connected to Arduino pin 9 and LPWM is connected to Arduino pin 7. ENA and ENB of the driver are connected to 5V. RIS and LIS pins are not used in the circuit.

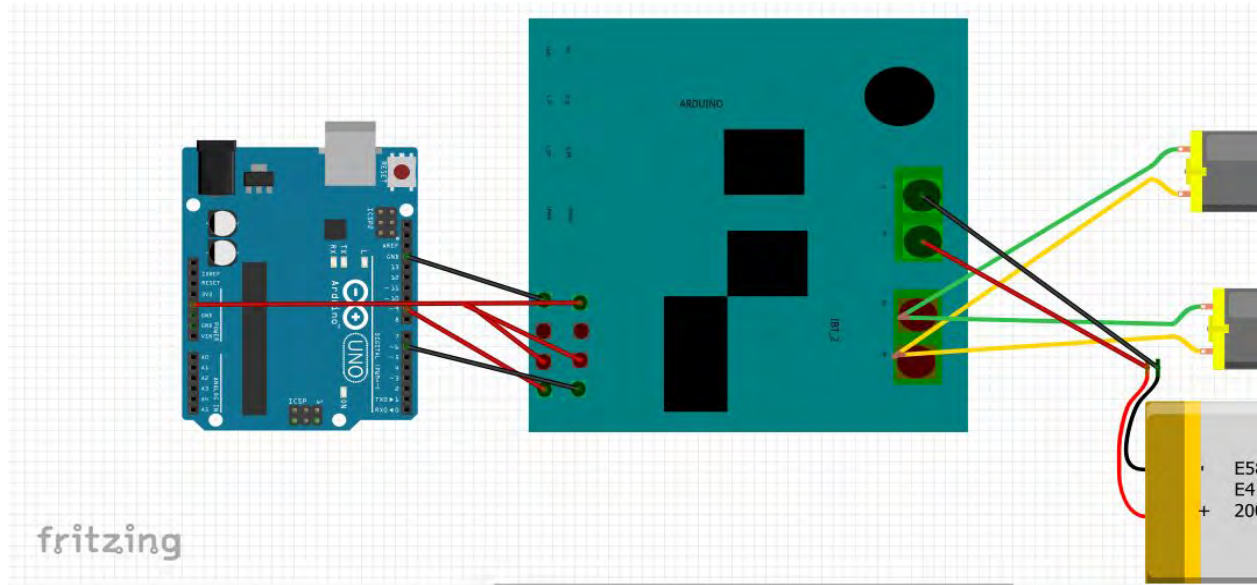


Figure 26 Motor Circuit X-direction

Motor Circuit of Y-axis direction: Single motor is used here. LPWM goes to Arduino pin 11 and RPWM to Arduino pin 10. Other connections are same as X-axis direction motor's connection.

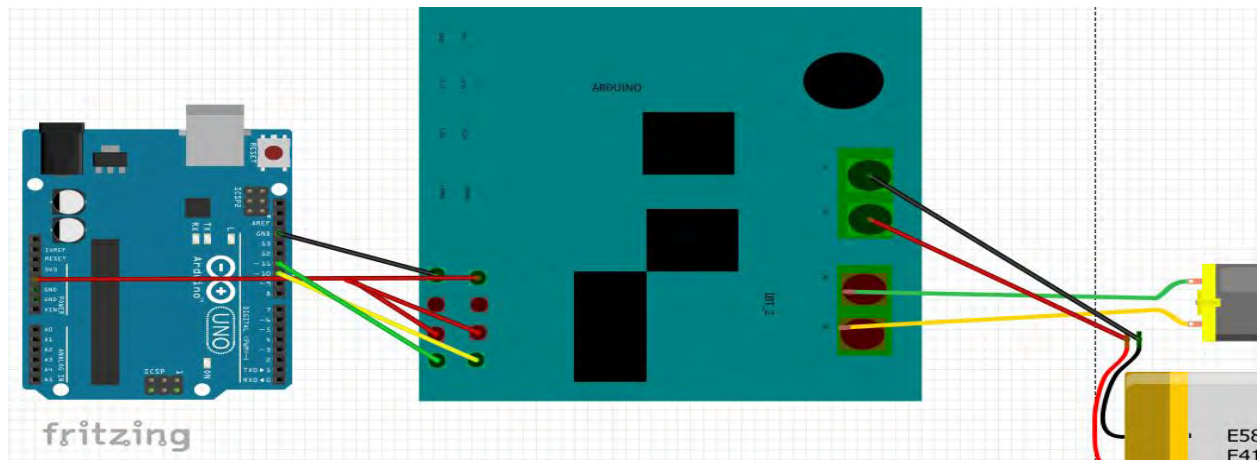


Figure 27 Motor Circuit Y-direction

Water Pump and Actuator Circuit Setup:

Both water pump and actuator is connected through L298N motor driver. For water pump, positive terminal goes from in3 to arduino pin 12 and negative terminal goes to in4 to arduino pin 2. For actuator, positive terminal goes from in1 to arduino pin 7 and negative terminal goes to in2 to arduino pin 8.

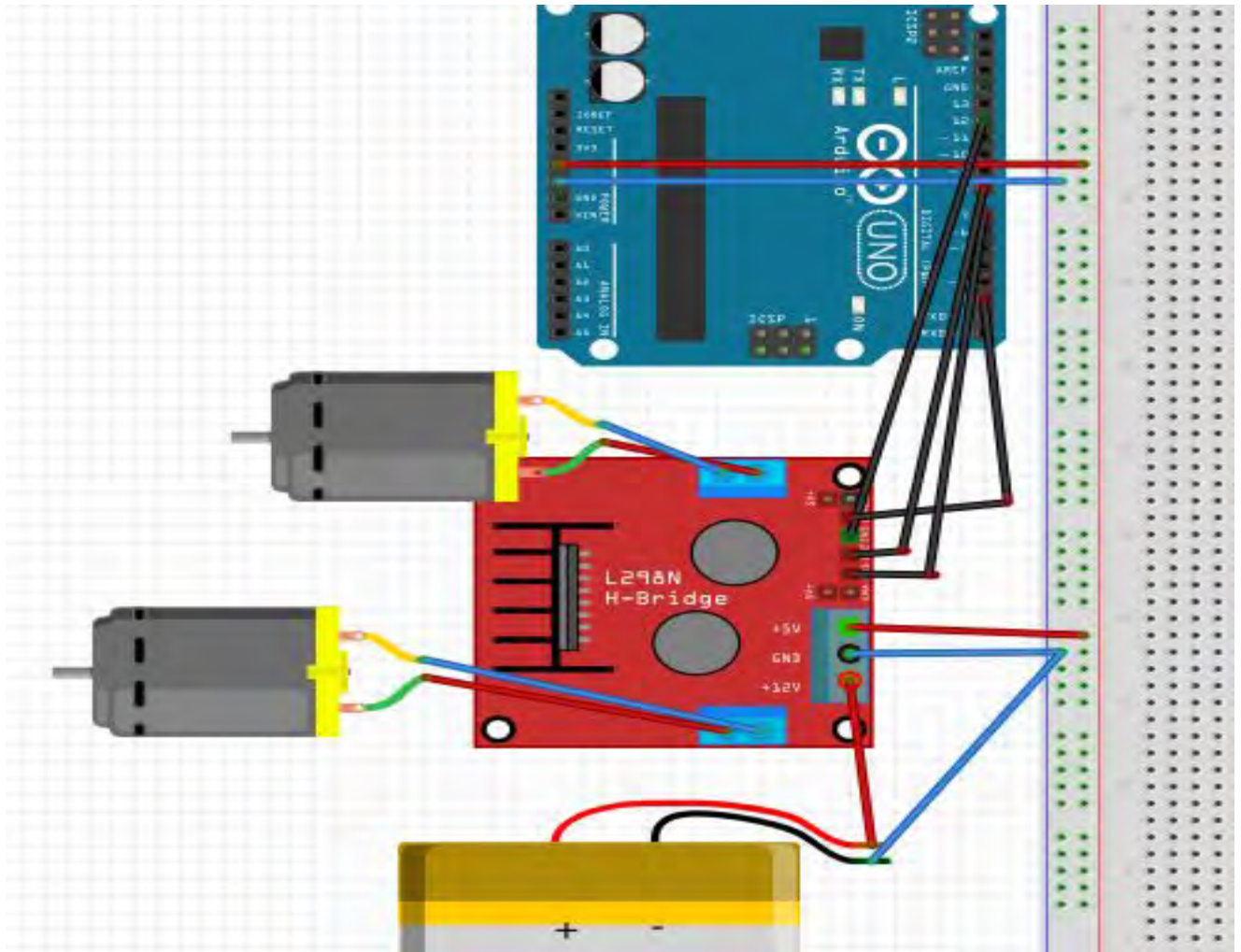


Figure 28 Connection for Actuator and Water pump

Seeder:

Signal pin of servo motor goes to the arduino pin number 5. Positive terminal goes to 5V and negative terminal goes to ground.

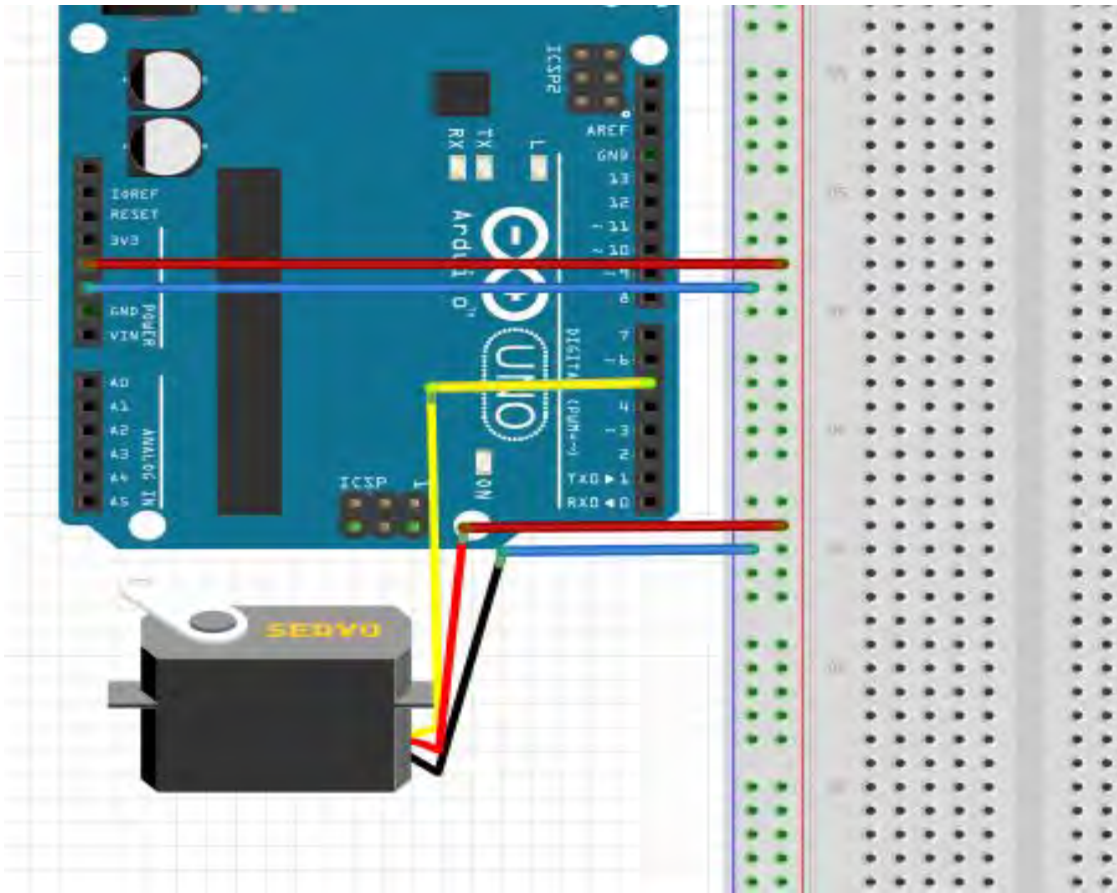


Figure 29 Seeder connection setup

Temperature, humidity and Moisture Sensor:

Signal pin goes to Arduino analog pin A1 for temperature and humidity sensor.

For moisture sensor, signal pin goes to the arduino analog pin A2.

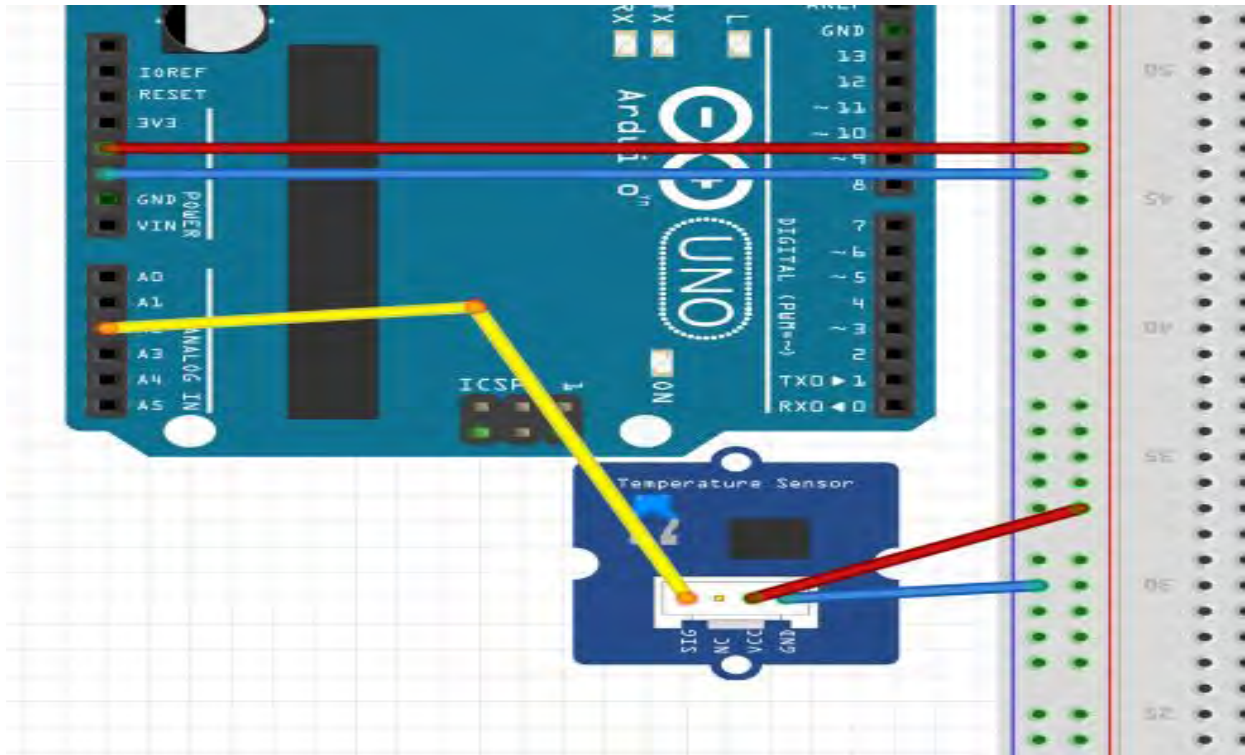


Figure 30 Temperature Sensor Setup

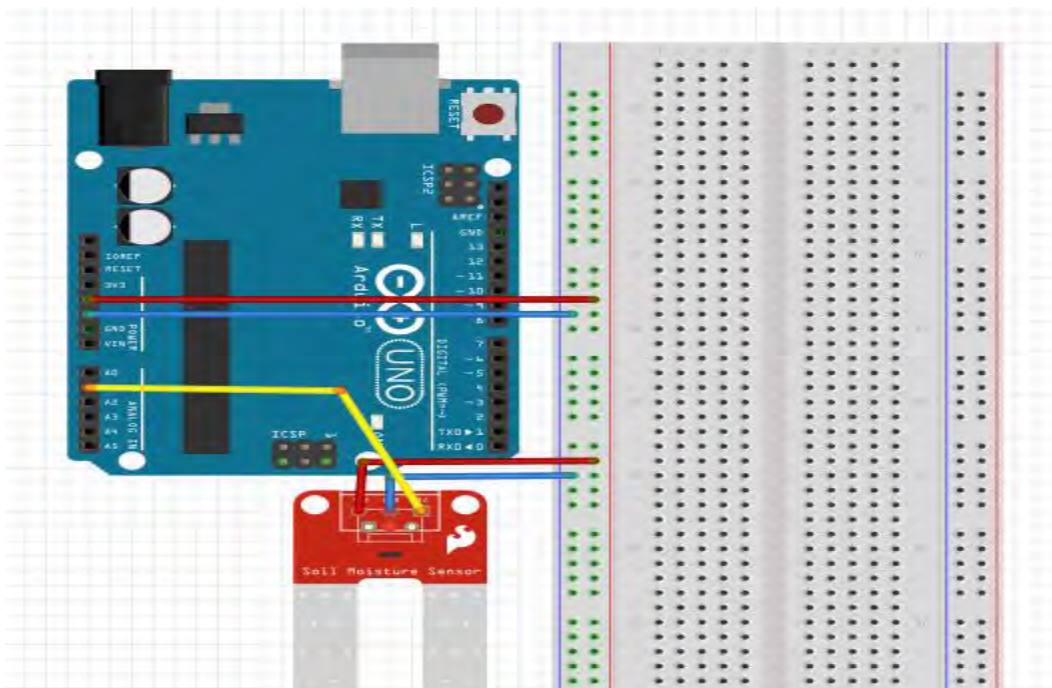


Figure 31 Moisture Sensor setup

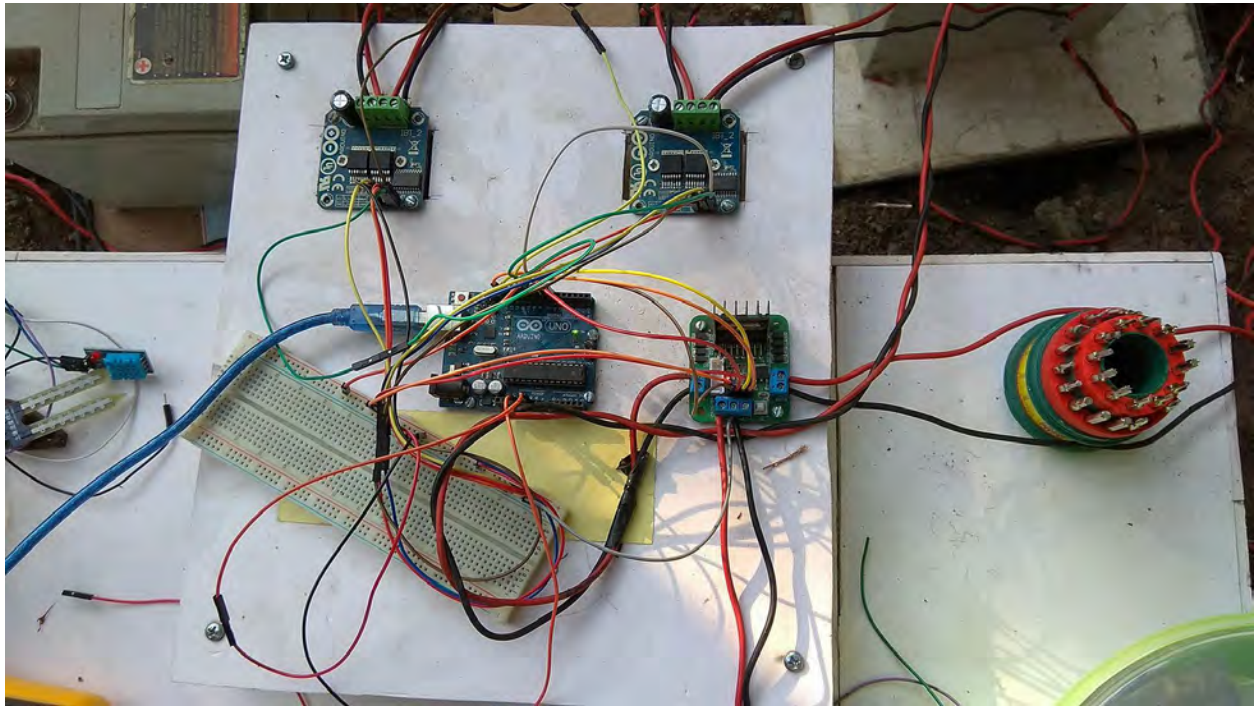


Figure 32 Circuit Setup in the field

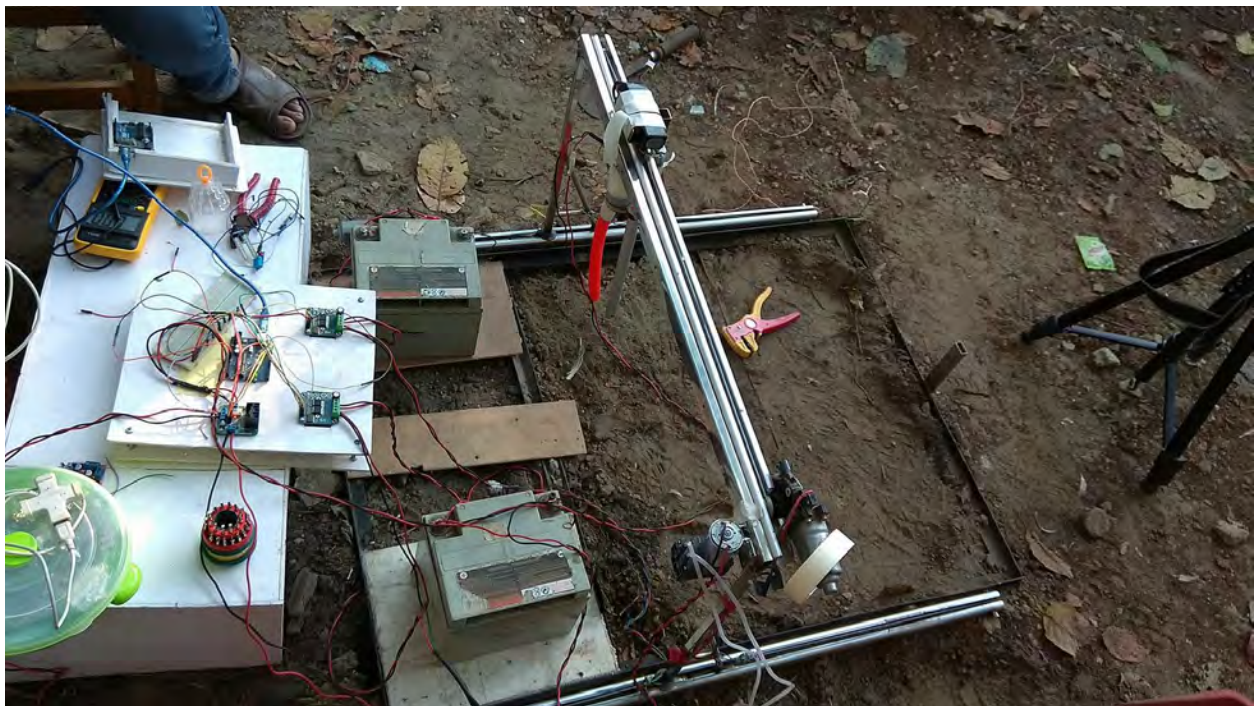


Figure 33 Circuit Setup in the field for testing

Software Used:

Arduino:

For constructing the code of our project we have used Arduino IDE version 1.8.4 which is open source Arduino software. The IDE is provided by Arduino community itself and it is a cross platform software which can runs on Windows, Mac OS and Linux. Compiling code with this software can easily be uploaded to any Arduino board.



Python:

Python is a very popular high level programming language developed by Python Software Foundation in 1991. Python code can be compiled and run by the IDE called Python.



We have used python software for the purpose of setting up the camera module which is connected to the raspberry pi. Moreover, operation of raspberry pi is based on python programming language therefore we have used this language and it's IDE.

PySerial:



Along with python, we have also used pySerial version 3.4 which is a python API module to access serial port. Therefore, pySerial is used in our project to build communication between python and arduino.

Scikit-image:



Scikit-image is another python package which is mainly used for the purpose of image processing. It uses NumPy arrays as image object. It means the whole image taken by the raspberry camera is divided into arrays. The array size depends on the quality of the picture taken. After that it assigns value from 0 to 255 per pixel of value where 0 represents white and 255 represents black. By this way, processing the image of soil gives us the idea of soil quality of the project field.

Flask Framework:



Flask is a micro framework which is based on two external libraries called Jinja2 template engine and Werkzeug WSGI toolkit. Flask is licensed under BSD. We have used this framework to design our website because it is based on python and very easy to set up. Moreover, Flask has built in development server and debugger. It is Unicode based framework and also supports for secure cookies of the client side sessions.

Fritzing:



Fritzing is open source software for designing electrical circuits. We have used this software for the purpose of simplification of our circuit which has been used in the project.

Building Web Server:

We have built the website using Raspberry pi and some of its software package. We used a SD card as the storage of Raspbian OS, a UNIX like operating system developed for the raspberry pi. Then we enabled the SSH for configuring the Pi over the network. After that, we installed pip, another python package manager which is required to install Flask. Then we installed Flask. With the help of Flask, we developed the coding for the website.

Automated Farming machine

with IOT!

Switch: {{ switch }}

MACHINE

MACHINE ON MACHINE OFF

Video Streaming Demonstration



Figure 34 Website without Style Sheet

As the website from plain html file looks ordinary, we have used Bootstrap for the purpose of using CSS and JavaScript. CSS means Cascading Style Sheet which is used to design and

organize a website. It can create borders, table, color including design and layout. After using bootstrap, our website looked like below while it was in operation.

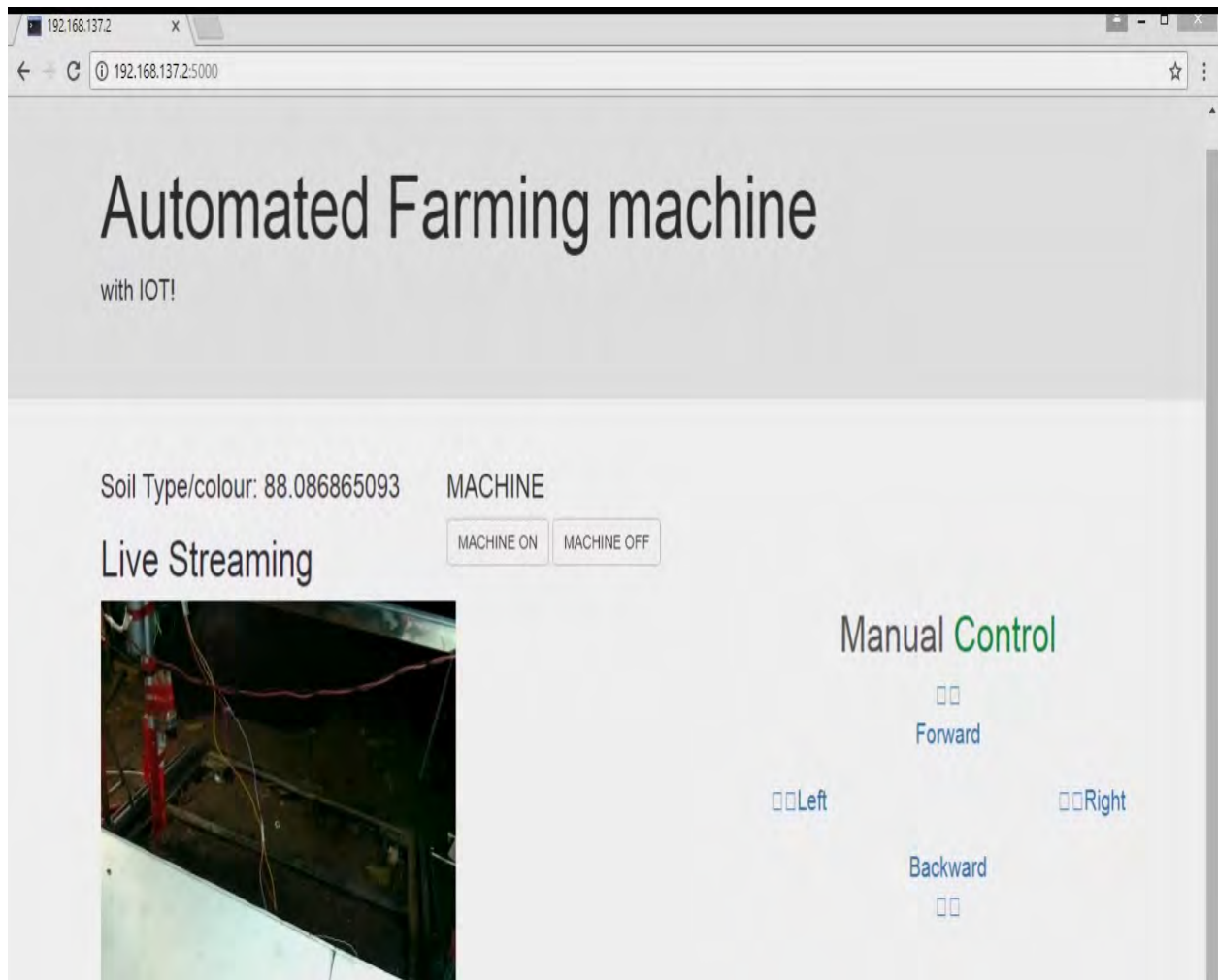


Figure 35 Complete Website

.When accessing the website through LAN cable or via raspberry pi directly, the address of the website is given as 192.168.137.2 which is the ip address provided to the raspberry pi. Flaskhasa dedicated port number of 5000.The web server is always running on the raspberry pi, so to access the webpage, we can open up any web browser and put the address of the ip address of the PI followed by port 5000 and we can access the website. So, the address is IP: 5000. Therefore, when we need to access the website from a remote location, we need to setup a router and configure port forwarding so that our Pi can be seen from anywhere in the world through the public IP of the router we have connected to it. Another way of accessing the webpage through the internet is to buy a domain name and assign that domain name to the raspberry pi By this

way, we can access the website remotely from anywhere in the world through the internet the same network. .

Operating Procedure:

The operation of our project is done in several steps like movement control, seeder operation, watering operation, sensing operation, operating the hand and camera module operations like streaming, taking picture and image processing of the soil. Each step is described briefly below with necessary diagrams.

Movement Control:

Movement control can be done in two ways in our project.

- Automatic control
- Manual control

While in automatic control, the machine runs itself by the help of the arduino code. But when in manual control mode, the process is done in several steps. First of all, when we press any button in the webpage such as forward, backward, left or right; it goes to main1.py file as a command number from the HTML file. This is done by assigning some values in the array. The array look like $\langle v_1, v_2, v_3 \rangle$ where v_1, v_2, v_3 etc. represents some six different numbers from 0 to 5. Where,

0 denotes => Machine Off

1 denotes => Machine On

2 denotes => Move Forward (X- axis direction)

3 denotes => Move Backward (X- axis direction)

4 denotes => Move Right (Y- axis direction)

5 denotes => Move Left (Y- axis direction)

After receiving these values, main1.py co-ordinates it with the code and send the processed command to thing.py file.

When it reaches to the file thing.py the code communicates with arduino by the help of pySerial. From there the signal goes to the arduino pin directly and works as the command is given by the user from the web server.

When the machine runs in automatic mode and the feedback of running motors goes to raspberry pi, the whole process is done in the opposite way. At that time, signal goes from arduino to thing.py and then from thing.py to main1.py file and then it updates on the website by rendering the HTML template.

Soil Digging:

Soil digging is done by the help of robotic hand which has an actuator installed on its end. When the machine runs for the first time, it does the duty of digging. The hand moves in the X-direction which is the column of the project field. After stopping at every column, the hand starts moving along the Y-axis which is the row of the project field. In every row, the hand digs the soil in an organized manner which is set previously with the delay time. Therefore, after a certain amount of time, each hole is being dug by the actuator.

Seeding Process:

The seeding process is done by the seeder which is made by a servo motor. Seeding process is also executed by the arduino code. After digging up the soil hole each time, the servo motor gets the command for opening the seeder port and the seed drops to the hole.

Watering Process:

Watering process is executed by the pump we have used and like before, this code is also written in arduino. After completing seeding process, the machine goes to its initial position and then it starts its action again for the purpose of watering. The same delay like soil digging is used so that

the machine can water the same area which has dug before. After watering process has done, the moisture of the soil will increase. Our system will take information from the database whether the humidity and moisture of the soil is in good range or not for that specific soil. When the amount of water will be decreased, the value of the moisture and humidity sensor will go down too. Then, the water pump will be started again to water the plants.

Camera Module:

By camera module the following processes is executed by our system.

- Live video streaming on the web page.
- Take still picture of the land.
- Image processing for finding soil quality.

Before any process is done, the camera is needed to set up with raspberry pi. For this purpose, camera module is set up with the help of camera.py file.

After the camera is being set, raspberry pi becomes a streaming server and the camera module becomes ready for further actions.

The camera module gets initiated by the file base_camera.py. After the user gets connected to the streaming server, the live streaming of the system gets started. The camera takes three frames per second and streams it to the server.

One of the main advantages of this process is, its power consumption is very low and it also can stop streaming after certain amount of time of user inactivity. For example, if the user starts the streaming and after some time of watching, shifts to a new tab of the web browser, the camera starts counting. When the streaming server runs without the client watching for ten seconds, it becomes stopped. The streaming can be resumed again after the user is connected to the server again.

Image Processing:

For the purpose of determining the soil quality, the camera takes a picture of the land and performs image processing by the help of image.py file. The image is divided into pixels and each pixel is assigned numbers from 0 to 255 according to the color of the soil. By the result of image processing, we can tell whether the soil is healthy or not, and what is the ph level of the soil, by analyzing the color of the soil. The following figure is showing the image processing code.

After the whole process, the streaming server provides us the live video and the soil type of our project land which is provided in the following figure. The raw color value needs to be tested more to map which color corresponds to which type of soil.



Figure 36 live streaming and soil types result in the server.

Result Analysis:**Machine's Movement Test:****Without Motor Driver (Directly Connected to Battery) along X-axis direction:**

Date	Avg. Voltage(V)	Avg. Current(A)	Power Consumption (W)	Time taken to travel (sec)
12 th July 2017	11.95	9.79	116.99	3.2
13 th July 2017	11.92	9.78	116.577	3.2
20 th July 2017	9.83	8.61	84.63	3.9
23 rd July 2017	9.27	8.13	75.36	4.2
27 th July 2017	11.98	9.88	118.36	3.1
30 th July 2017	11.92	9.79	116.69	3.2
1 st August 2017	11.83	9.75	115.34	3.3

Without Motor Driver (Directly Connected to Battery) along Y-axis direction:

Date	Avg. Voltage(V)	Avg. Current(A)	Power Consumption (W)	Time taken to travel (sec)
12 th July 2017	11.95	4.92	58.794	3.7
13 th July 2017	11.92	4.90	58.41	3.8
20 th July 2017	9.83	4.67	45.91	4.1
23 rd July 2017	9.27	4.46	41.34	4.3
27 th July 2017	11.98	4.95	59.30	3.4
30 th July 2017	11.92	4.91	58.53	3.6
1 st August 2017	11.83	4.87	57.61	3.7

With Motor Driver Connected along X-axis direction:

Date	Avg. Voltage(V)	Avg. Current(A)	Power Consumption (W)	Time taken to travel (sec)
10 th August 2017	11.77	9.53	112.17	7.2
13 th August 2017	11.74	9.48	111.29	7.4
30 th September 2017	9.53	8.74	83.29	9.5
12 th October 2017	9.48	8.69	82.38	9.8
19 th October 2017	9.45	8.65	81.74	10.1
7 th November 2017	11.96	9.71	116.13	7.1
10 th November 2017	11.94	9.88	117.97	7.0

With Motor Driver Connected along Y-axis direction:

Date	Avg. Voltage(V)	Avg. Current(A)	Power Consumption (W)	Time taken to travel (sec)
10 th August 2017	11.77	4.48	52.73	9.8
13 th August 2017	11.74	4.45	52.24	10.0
30 th September 2017	9.53	4.13	39.36	11.2
12 th October 2017	9.48	4.02	38.11	12.0
19 th October 2017	9.45	4.00	37.80	12.1
7 th November 2017	11.96	4.75	56.81	9.6
10 th November 2017	11.94	4.72	56.36	9.5

Result Analysis:

As two power window motors are connected in parallel in the X-axis direction, they are required a rated voltage of 12 Volts and 5 Amperes of current each. Our result shows that, their speed varies on different day as the DC power varies from day to day.

On the other hand, power window motor installed for Y-axis movement has only 1 motor. Therefore, it only needs 5 Amperes of rated current to operate and 12 Volts DC power supply.

We have also tried to run the motors by using the power of laptop charger which was rated as 19 Volts DC and 3.3 Amperes of current. We have seen that single motor can run very slowly in the Y-axis direction but in the X-axis direction, dual motors do not run at all. It is because the DC adapter cannot provide the required 10 Ampere rated current to the motors terminal.

Seeding machine test:

Item	Test(times)	Positive result	Negative result	Percentage of success	Percentage of error
Wet gram	37	21	16	56.75%	43.24%
Gram	41	33	8	80.48%	19.51%
Lentils	26	23	3	88.46%	11.53%
Mustard seed	44	43	1	97.72%	2.27%

Result analysis: we were using different type of seeds to check our seeder. The table and the bar graph show the scenario of its result. It gave different results for different items of seeds. Wet gram is quite big enough than any other seed. When we test with wet gram then we experimented 37 times, 21 times the seed dropped properly and it failed 16 times. The reason behind this was the size; some wet gram size is bigger than our seeder hole.

Then we did our experiment with gram which was smaller than the wet gram, it was quite easier for our seeder to pass. Gram tested 41 times and 33 of them passed through easily, 8 times it gives negative result. Some gram was bigger than the hole.

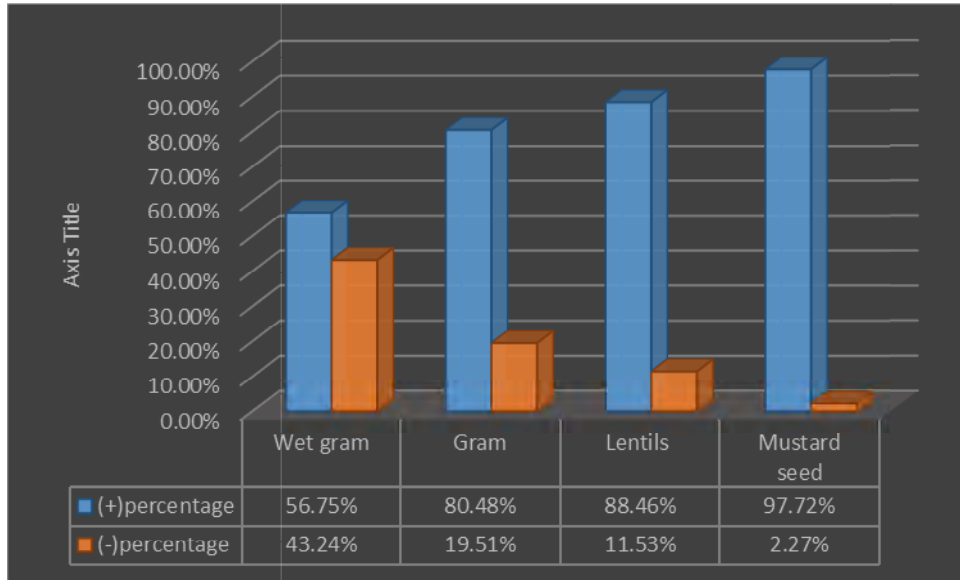


Figure 37 Seeder Test Result

Lentils are very smaller than the gram. When we experimented with it 23 times it gave the positive result out of 26 times. 3 times it failed because of lose connection. Our machine works like CNC machine so when we did our experiment many times the wires were disconnected because of pulling them every time.

Finally we did the experiment with mustard seed which was the smallest seed we tested for our experiment. It gave almost 100% positive results but due to lose connection one loop was missing that was the reason behind of that failure. So we can finally say that this seeding machine is working very nicely the problem is its size. If we fixed the size of the hole according with the seeds size and if there is no lose connection then it will always gives us positive result. The percentage of success is increasing with the decreasing number of size decreasing of seeds.

Fertilizer testing

We gave fertilizer with the same machine. The size of fertilizer is very smaller than the seeds. So did not face any size related problem here. We gave fertilizer 10 times every time it gives positive result, one time it takes long time to start but we have no idea why that delay occurred.

Water pump test

Watering the soil depends on the temperature, humidity and moisture condition of soil. Unfortunately we tested our machine in a rainy day, so the moisture level and temperature level was good enough but for the sake of our experiment. We first ran the motor controlling the water pumps at its rated power for one second, to splash a bit of water through the pipes, but no water came through the pipe, so then we ran the motor for 3 seconds, too much water came through the pipe, so we ran the motors from 1 second to 3 seconds, with an increment of 0.5 seconds and the best result was shown at 2 seconds and 2.5 seconds. So we chose around 2.3 seconds as the best time to run the water pump motor.

Difficulties Faced:

Throughout the implementation process of our project, we have faced some difficulties. Some of them are hardware based and some of them are software based. The main difficulties which we have faced are:

- Designing the system.
- Designing the seeder.
- Congestion of the machine.
- Running the entire module at a same time.

Designing the system:

Designing the machine frame was the first difficulty we have faced. We had to make a model in such a way so that the system becomes appropriate for the growing purpose of any plant that we wish to farm. First of all, we were worried about the shading because our machine can become an obstacle to the plant by blocking the direct sunlight. Moreover, we had to make sure that the hand, which will do all the farming works, can perfectly go to any place of the farming area. After taking the idea from CNC machine's mechanism, our next challenge was to make sure,

how the hand can move freely. We made some approaches to find out which motor will be suitable for the process and how the hand will be connected with the motor.

Secondly, designing the hand was another challenge. How the system can sow seeds how can it do the watering process was a brainstorming moment for all of us. At the beginning we decided to use vacuum pump to hold the seed with the hand. But after some researching, we found out that it might not be effective that much. Furthermore, it will consume more power than our current system.

Overall, modeling the system was our first and one of the great challenges we faced.

Designing the Seeder:

As mentioned before, seeder designing was another challenge we have faced because we were not sure whether the seed can drop in the right place at a right time. Moreover, we were also not sure where the seeds will be stored. At the very first moment we decided to keep the seeds in storage outside the farming area. The idea was to fetch one seed each time when it was needed to be sowed. We cancelled that idea after giving some thought. Then we were thinking to make a seeder with the help of stepper motor so that the motor could take the seeds to an open end of the storage and seed could be dropped. But again after some calculation, we found out that we can make the seeder using just one servo motor which would be more affordable and easier for the overall system.

Congestion of the machine:

After developing the basic frame of the system, a new challenge came to us. When we were trying to test the system with direct DC supply, it could not move freely, sometimes it was jammed totally. Therefore, we were thinking for using the chain for the purpose of pulling the load of the hand. But as the chain would increase the budget of the system heavily, we took a different approach.

For making the hand move freely, we installed three sets of small rail wheel, where, each set contains four wheels for the larger support. Two sets were installed in the X-axis direction and rest of the set was installed in the Y-axis direction. Also we have changed the attaching system of the wheels with the frame. We used one set of extra pulley at the end of every corner of the frame to make the hand more free.

By these ways after some days of struggling, we overcame our challenges and the machine finally got worked.

Running the entire module at a same time:

One of our software based difficulties was to run the whole system at a same time. It may seem to be less problematic in naked eye but when we developed the code, we found it very difficult to control the entire feature smoothly at a same time. The calibration needed to be perfect for the machine to work. Moreover, after digging and sowing, regular watering and monitoring the soil was not an easy task for us.

Limitations:

Though our project worked successfully, it has some limitations. If we can overcome these limitations, our project will be more efficient. Some of the limitations that our project has are:

- Designed for the normal weather only.
- Seeder efficiency rate is quite low.
- The system is not easily movable.
- Cannot test pH level of the soil.
- Budget is quite high.

Designed for normal weather:

Our machine does not have any storm proof solution. Though it has some cover for the safety of battery and other electrical instrument, heavy rain can cause damage to the system.

Seeder Efficiency:

The efficiency rate of the seeder is quite low. Sometimes the seeds got stuck in the seeder's dropping point because the seeder's shape is cylindrical. Therefore, even if the seeds are perfectly round shaped, it can be stuck. Moreover, if the seeds are too small like pulse seed, the seeds can be dropped in a bigger quantity at a single place.

System is not easily movable:

Our system is quite large and because of the frame size, it is not easily movable. Moreover, it is needed to be set up when it is placed in a new location.

pH Level Testing:

Our system still cannot measure the pH level of the soil. The available pH measurement tools in the market require soil to be mixed with water for the measurement of pH test. Therefore, we were unable to implement the pH level testing of the soil.

Quite High Budget:

The estimated budget of our project is 28 thousand taka which is equivalent to 350 USD approximately. Therefore, the cost is not affordable for all the people specially people of the rural areas of our country.

Future Work:

In future, we have the plan to overcome the limitations and add some new features to our project so that the general people of our country can get the benefit of happy farming. Some of the extensions that we would like to add in future are discussed briefly below.

Rooftop Gardening:

We can easily set the system in the rooftop of the city buildings to increase the greenery.

Greenhouse:

We have the plan to implement the system in greenhouse. If we can make it to the greenhouse, the agricultural system of the colder countries will be more improved.

Moving System:

By making the system able of moving, the efficiency can be increased greatly because moving automated system will have the scope to cover larger area without any boundary limitations.

Immune to harsh weather:

Improving the protection system of our project can make it resistive to harsh weather so that it can function normally in the rain, storm etc.

Developed Image Processing:

We have the plan to develop the image processing system so that it can also detect the diseases of the plant, can gather information about whether the plants are ready for harvest or not.

Self Harvesting:

Self harvesting system will be a great addition for the progress of our system.

Solar Power:

If we can use solar panel in our system, it will increase the budget but in the long run it will give us more outcome than the system which takes power from the DC source. Therefore, implementing solar system can be another great additional improvement for our system.

Conclusion:

Farmers are those people who work day and night under the bright sun during summer, under cloud and rain during rainy season. They don't have any vacation. Each and every season they have to cultivate something. Question comes why they do it though they are not paid enough money? They do it just to make sufficient food for us. They do it for the people of their country. They do it out of responsibility and love towards their country.

Now it is time for us to do something for those hard working people because of whom we can eat at least three times a day and fulfill the need of our stomach. In our project, we are using the science and technology to make farming easier and enjoyable. As we all know farming is boring and monotonous and so lot of people are not interested in this profession. As a result day by day

the ratio of farmers in respect to total population is decreasing. Somehow we have to attract people towards farming and we believe our machine can do that.

If we want to describe our device in short, we can say our device is mainly focused on three specific sides.

- To make farming easier
- To make farming faster
- To make farming enjoyable

Firstly, IoT Based Farm Automation System can perform watering, seeding, soil digging for plants. Moreover, it can sense the condition of soil through temperature and moisture sensor. So, it makes farming easier.

Secondly, we can observe the condition of our plant through raspberry pi camera module. As a result a person can know what is going on in his/her field and can take any kind of necessary step from remote area. As a result it makes farming faster.

Thirdly, one does not need to use his/her hand dirty and doesn't need to spend all the time in the field. Not just this, one can easily take rest and also can do multitasking because major duties of farming can be performed through this device. In fact any children can do farming by taking the help of this device. In this way, our device can make farming enjoyable.

References:

1. Libin, Zhang, et al. "Overview of research on agricultural robot in China." *International Journal of Agricultural and Biological Engineering* 1.1 (2008): 12-21.
2. Yaghoubi, Sajjad, et al. "Autonomous robots for agricultural tasks and farm assignment and future trends in agro robots." *International Journal of Mechanical and Mechatronics Engineering* 13.3 (2013): 1-6.
3. D'Ausilio, Alessandro. "Arduino: A low-cost multipurpose lab equipment." *Behavior research methods* 44.2 (2012): 305-313.
4. *Arduino UNO R3*, n.p., n.d., retrieved November 27, 2017, from www.hobbytronics.co.uk/arduino-uno-r3.
5. Pakdaman, Mehran, M. Mehdi Sanaatiyan, and Mahdi Rezaei Ghahroudi. "A line follower robot from design to implementation: Technical issues and problems." *Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on*. Vol. 1. IEEE, 2010.
6. Small, Raspberry Pi-A., and Cost Effective Powerful. "International Journal of Advanced Research in Computer Science and Software Engineering." *International Journal* 5.12 (2015).
7. "BTS 7960 High Current PN Half Bridge." *Robotpower.Com*, 2004, http://www.robotpower.com/downloads/BTS7960_v1.1_2004-12-07.pdf.
8. "Bangladesh - Agricultural Land (% of Land Area)." Agricultural Land (% of Land Area) in Bangladesh. TRADING ECONOMICS, n.d. Web. 22 Dec. 2017. <<https://tradingeconomics.com/bangladesh/agricultural-land-percent-of-land-area-wb-data.html>>.
9. "Agriculture." Agriculture - Banglapedia. Banglapedia- National Encyclopedia of Bangladesh, n.d. Web. 10 Dec. 2017. <<http://en.banglapedia.org/index.php?title=Agriculture>>.
10. TRADING ECONOMICS- 20 Million INDICATORS FROM 196 COUNTRIES. N.p., n.d. Web. 6 Dec. 2017. <<https://tradingeconomics.com/>>.
11. Krassenstein, Eddie. "FarmBot: An Open Source 3D Farming Printer That Aims to Create Food For Everyone." 3DPrint.com. N.p., 19 Aug. 2014. Web. <<https://3dprint.com/12325/farmbot-3d-farming-printer/>>.

12. G. Rajagopal, V. M. Lodd, A. Vignesh, R. Rajesh and V. Vijayaraghavan, "Low cost cloud based intelligent farm automation system using Bluetooth Low Energy," 2014 IEEE Region 10 Humanitarian Technology Conference (R10 HTC), Chennai, 2014, pp. 127-132.
13. D. J. Turnell, Q. V. de Fatima, M. Turnell, G. S. Deep and R. C. S. Freire, "FarmWeb-an integrated, modular farm automation system," Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on, San Diego, CA, 1998, pp. 1184-1189 vol.2.
14. Arouje, Sony. "Farm Automation System Based on Arduino and Raspberrypi." Farm Automation System Based on Arduino and Raspberrypi. N.p., 17 Feb. 2016. Web. 02 July 2017. <<https://sonyarouje.com/2016/02/17/farm-automation-system-based-on-arduino-and-raspberrypi/>>.
15. FARMBOT– SMALLSCALE, LOW COST FARMING AUTOMATION SYSTEM, n.p., n.d., retrieved November 27, 2017, from http://www.kscst.iisc.ernet.in/spp/39_series/SPP39S/02_Exhibition_Projects/135_39S_B_E_1871.pdf
16. Ahmad, Reaz. "Farm Growth Slows down." The Daily Star. The Daily Star, 23 Aug. 2017. Web. 02 Dec. 2017. <<http://www.thedailystar.net/frontpage/farm-growth-slows-down-1453204>>.
17. Pai, Akshay. "Manipulating Image Pixels with Python Scikit Image - Color Schemes." Source Dexter. N.p., 01 May 2017. Web. 22 Dec. 2017. <<https://sourcedexter.com/manipulating-image-pixels-with-python-scikit-image-color-schemes/>>.
18. "A Crash Course on NumPy for Images." A Crash Course on NumPy for Images — Skimage V0.14dev Docs. Scikit-image.org, n.d. Web. 06 Nov. 2017. <http://scikit-image.org/docs/dev/user_guide/numpy_images.html>.
19. Gouillart, Emmanuelle. "3.3. Scikit-image: Image Processing." Scikit-image: Image Processing. N.p., n.d. Web. 06 Nov. 2017. <<http://www.scipy-lectures.org/packages/scikit-image/>>.
20. Grinberg, Miguel. "Video Streaming with Flask." Video Streaming with Flask - Miguelgrinberg.com. N.p., n.d. Web. 03 Oct. 2017. <<https://blog.miguelgrinberg.com/post/video-streaming-with-flask>>.

21. Poole, James. "Raspberry Pi Web Controlled Robot With Video Stream." N.p., 11 May 2016. Web. 7 Oct. 2017. <<http://jamespoole.me/2016/04/29/web-controlled-robot-with-video-stream/>>.
22. Mjrovai. "IoT: Raspberry Pi Robot With Video Streamer and Pan/Tilt Camera Remote Control Over Internet." Instructables.com. Instructables, 27 Sept. 2017. Web. 22 Oct. 2017. <<http://www.instructables.com/id/IoT-Raspberry-Pi-Video-Streamer-and-PanTilt-Camera/>>.
23. Ali, Noor. "IoT: How to Control Appliances Using Python & Raspberry Pi." The Official Cloudways Blog. N.p., 14 July 2016. Web. 22 Aug. 2017. <<https://www.cloudways.com/blog/internet-of-things-remote-control-appliances-using-raspberry-pi/>>.
24. "Big Rob – Raspberry Pi Robot Motor Driver BTS7960B and Python Program." Raspberry Pi Roboter. N.p., 16 June 2017. Web. 03 Nov. 2017. <<https://custom-build-robots.com/raspberry-pi-robot-cars/big-rob-motor-driver-bts7960b/8155?lang=en>>.
25. Nelli, Fabio. "Tutorial - Sending Values from PC to Arduino by Serial Communication." Meccanismo Complesso. N.p., 30 July 2014. Web. 22 Dec. 2017. <<http://www.meccanismocomplesso.org/en/arduino-tutorial-serial-data-actuator/>>.
26. Chronc, Lukas. "DIY Arduino and Raspberry Pi Weather Station and Web Server." Instructables.com. Instructables, 01 Oct. 2017. Web. 22 Dec. 2017. <<http://www.instructables.com/id/DIY-Arduino-and-Raspberry-Pi-Weather-Station-and-W/>>.

Appendix:

```
@app.route('/left_side')
def left_side():
    data1 = "LEFT"
    GPIO.output(m11 , 0)
    GPIO.output(m12 , 0)
    GPIO.output(m21 , 1)
    GPIO.output(m22 , 0)
    return 'true'

@app.route('/right_side')
def right_side():
    data1 = "RIGHT"
    pi_thing.set_led(1)
    return 'true'

@app.route('/up_side')
def up_side():
    data1 = "FORWARD"
    pi_thing.set_led(1)

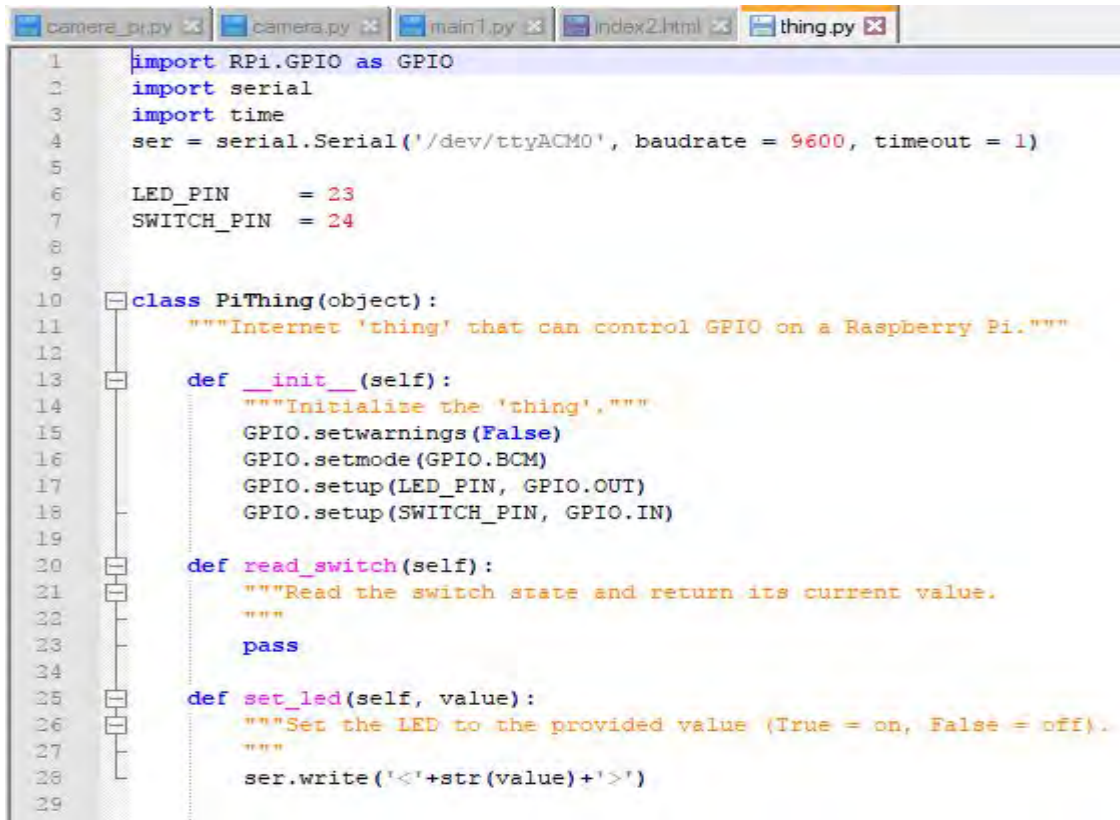
    return 'true'

@app.route('/down_side')
def down_side():
    data1 = "BACK"

    return 'true'

@app.route('/stop')
def stop():
    data1 = "STOP"
```

Figure 1a: thing.py code for running motors



```

1  import RPi.GPIO as GPIO
2  import serial
3  import time
4  ser = serial.Serial('/dev/ttyACM0', baudrate = 9600, timeout = 1)
5
6  LED_PIN      = 23
7  SWITCH_PIN   = 24
8
9
10 class PiThing(object):
11     """Internet 'thing' that can control GPIO on a Raspberry Pi."""
12
13     def __init__(self):
14         """Initialize the 'thing'."""
15         GPIO.setwarnings(False)
16         GPIO.setmode(GPIO.BCM)
17         GPIO.setup(LED_PIN, GPIO.OUT)
18         GPIO.setup(SWITCH_PIN, GPIO.IN)
19
20     def read_switch(self):
21         """Read the switch state and return its current value.
22         """
23         pass
24
25     def set_led(self, value):
26         """Set the LED to the provided value (True = on, False = off).
27         """
28         ser.write('<' + str(value) + '>')
29

```

Figure 1b: thing.py code for running motors

```

analogWrite(RPWM, pwm); // RUN X DIR MOTORS
analogWrite(LPWM, 0);

delay(persec);
analogWrite(RPWM, 0);
delay(500);

while(persecl<=6000){

analogWrite(RPWM1, pwm); // RUN Y DIRECTION MOTORS
analogWrite(LPWM1, 0);
delay(persecl);
analogWrite(RPWM1, 0);
persecl=persecl+1500;
delay(500);
}

```

Figure 1c: Arduino code for running motors

```

73 void recvWithStartEndMarkers() { //METHOD FOR RECIECING DATA FROM PI
74     static boolean recvInProgress = false;
75     static byte ndx = 0;
76     char startMarker = '<';
77     char endMarker = '>';
78     char rc;
79
80     while (Serial.available() > 0 && newData == false) {
81         rc = Serial.read();
82
83         if (recvInProgress == true) {
84             if (rc != endMarker) {
85                 receivedChars[ndx] = rc;
86                 ndx++;
87                 if (ndx >= numChars) {
88                     ndx = numChars - 1;
89                 }
90             }
91             else {
92                 receivedChars[ndx] = '\0';
93                 recvInProgress = false;
94                 ndx = 0;
95                 newData = true;
96             }
97         }
98
99         else if (rc == startMarker) {
100             recvInProgress = true;
101         }
102     }

```

Figure 2: Receiving data to Arduino

```

307 void parseData() { //PARSING THE DATA RECIEVED
308     char * strtokIndx;
309
310     strtokIndx = strtok(tempChars, ",");
311     integerFromPC = atoi(strtokIndx);
312
313     strtokIndx = strtok(NULL, ",");
314     integerFromPC1 = atoi(strtokIndx);
315
316     strtokIndx = strtok(NULL, ",");
317     integerFromPC2 = atoi(strtokIndx);
318

```

Figure 3: Parsing Data Received in Arduino

```

150     digitalWrite(in1, HIGH);           //RUN ACTUATORS
151     digitalWrite(in2, LOW);
152     delay(1000);
153     digitalWrite(in3, HIGH);
154     digitalWrite(in4, LOW);
155
156     delay(1000);
157
158     digitalWrite(in3, LOW);
159     digitalWrite(in1, LOW);
160     digitalWrite(in2, HIGH);
161     digitalWrite(in1, LOW);
162     digitalWrite(in2, LOW);
163     digitalWrite(in3, LOW);
164     digitalWrite(in4, LOW);
165

```

Figure 4: Code for the soil digging action

```

    myPointer.write(60);           //SEEDING ACTION
    delay(servoDelay);

myPointer.write(120);
myPointer.write(120);

for (pos=120; pos>=55; pos=pos-1) {

    myPointer.write(pos);
    delay(servoDelay);

}

myPointer.write(120);

```

Figure 5: Code of seeding action

```

22  #include <SimpleDHT.h>
23  int wait=1;
24  int pinDHT11 = A2;
25  SimpleDHT11 dht11;
26  int soilmoisture=A1;
27  //int soilmoisture=8;
28  int percentmoisture;
29
30

```

Figure 6a: Sensor code

```

92  myPointer.write(120);
93
94
95  byte temperature = 0; //SENSORS
96  byte humidity = 0;
97  if (dht11.read(pinDHT11, &temperature, &humidity, NULL)) {
98    Serial.print("Read DHT11 failed.");
99    return;
100 }//
101
102
103  Serial.print((int)temperature); Serial.print(" ^C, ");
104  Serial.print((int)humidity); Serial.println(" %");
105  int sensorvalue=analogRead(A1);//SOIL MOSITURE
106
107  Serial.print("mosture level: ");
108  Serial.println(sensorvalue);
109  percentmoisture=map(sensorvalue,245,1023,100,0);
110  Serial.println(percentmoisture);
111  delay(1000);
112  .....

```

Figure 6b: Sensors code for arduino


```

1  import time      //INITIATING THE CAMERA
2  from base_camera import BaseCamera
3
4
5  class Camera(BaseCamera):
6      """An emulated camera implementation that streams a repeated sequence of
7      files 1.jpg, 2.jpg and 3.jpg at a rate of one frame per second."""
8      imgs = [open(f + '.jpg', 'rb').read() for f in ['1', '2', '3']]
9
10     @staticmethod
11     def frames():
12         while True:
13             time.sleep(1)
14             yield Camera.imgs[int(time.time()) % 3]
15
16

```

Figure 7a: Setting up camera to raspberry pi

```

1  import time      //INITIATING THE CAMERA
2  from base_camera import BaseCamera
3
4
5  class Camera(BaseCamera):
6      """An emulated camera implementation that streams a repeated sequence of
7      files 1.jpg, 2.jpg and 3.jpg at a rate of one frame per second."""
8      imgs = [open(f + '.jpg', 'rb').read() for f in ['1', '2', '3']]
9
10     @staticmethod
11     def frames():
12         while True:
13             time.sleep(1)
14             yield Camera.imgs[int(time.time()) % 3]
15
16

```

Figure 7b: Setting up camera to raspberry pi


```
1  import time
2  import threading
3  try:
4      from greenlet import getcurrent as get_ident
5  except ImportError:
6      try:
7          from thread import get_ident
8      except ImportError:
9          from _thread import get_ident
10
11
12 class CameraEvent(object):
13     """An Event-like class that signals all active clients when a new frame is
14     available.
15     """
16     def __init__(self):
17         self.events = {}
18
19     def wait(self):
20         """Invoked from each client's thread to wait for the next frame."""
21         ident = get_ident()
22         if ident not in self.events:
23             # this is a new client
24             # add an entry for it in the self.events dict
25             # each entry has two elements, a threading.Event() and a timestamp
26             self.events[ident] = [threading.Event(), time.time()]
27         return self.events[ident][0].wait()
```

Figure 8a: Live Streaming Code

```

29 def set(self):
30     """Invoked by the camera thread when a new frame is available."""
31     now = time.time()
32     remove = None
33     for ident, event in self.events.items():
34         if not event[0].isSet():
35             # if this client's event is not set, then set it
36             # also update the last set timestamp to now
37             event[0].set()
38             event[1] = now
39         else:
40             # if the client's event is already set, it means the client
41             # did not process a previous frame
42             # if the event stays set for more than 5 seconds, then assume
43             # the client is gone and remove it
44             if now - event[1] > 5:
45                 remove = ident
46     if remove:
47         del self.events[remove]
48
49 def clear(self):
50     """Invoked from each client's thread after a frame was processed."""
51     self.events[get_ident()][0].clear()
52

```

Figure 8b: Live Streaming Code

```

54 class BaseCamera(object):
55     thread = None # background thread that reads frames from camera
56     frame = None # current frame is stored here by background thread
57     last_access = 0 # time of last client access to the camera
58     event = CameraEvent()
59
60     def __init__(self):
61         """Start the background camera thread if it isn't running yet."""
62         if BaseCamera.thread is None:
63             BaseCamera.last_access = time.time()
64
65             # start background frame thread
66             BaseCamera.thread = threading.Thread(target=self._thread)
67             BaseCamera.thread.start()
68
69             # wait until frames are available
70             while self.get_frame() is None:
71                 time.sleep(0)
72
73     def get_frame(self):
74         """Return the current camera frame."""
75         BaseCamera.last_access = time.time()
76
77         # wait for a signal from the camera thread
78         BaseCamera.event.wait()
79         BaseCamera.event.clear()
80
81         return BaseCamera.frame
82

```

Figure 8c: Live Streaming Code


```

83     @staticmethod
84     def frames():
85         """Generator that returns frames from the camera."""
86         raise RuntimeError('Must be implemented by subclasses.')
87
88     @classmethod
89     def _thread(cls):
90         """Camera background thread."""
91         print('Starting camera thread.')
92         frames_iterator = cls.frames()
93         for frame in frames_iterator:
94             BaseCamera.frame = frame
95             BaseCamera.event.set() # send signal to clients
96             time.sleep(0)
97
98             # if there hasn't been any clients asking for frames in
99             # the last 10 seconds then stop the thread
100            if time.time() - BaseCamera.last_access > 10:
101                frames_iterator.close()
102                print('Stopping camera thread due to inactivity.')
103                break
104            BaseCamera.thread = None
105

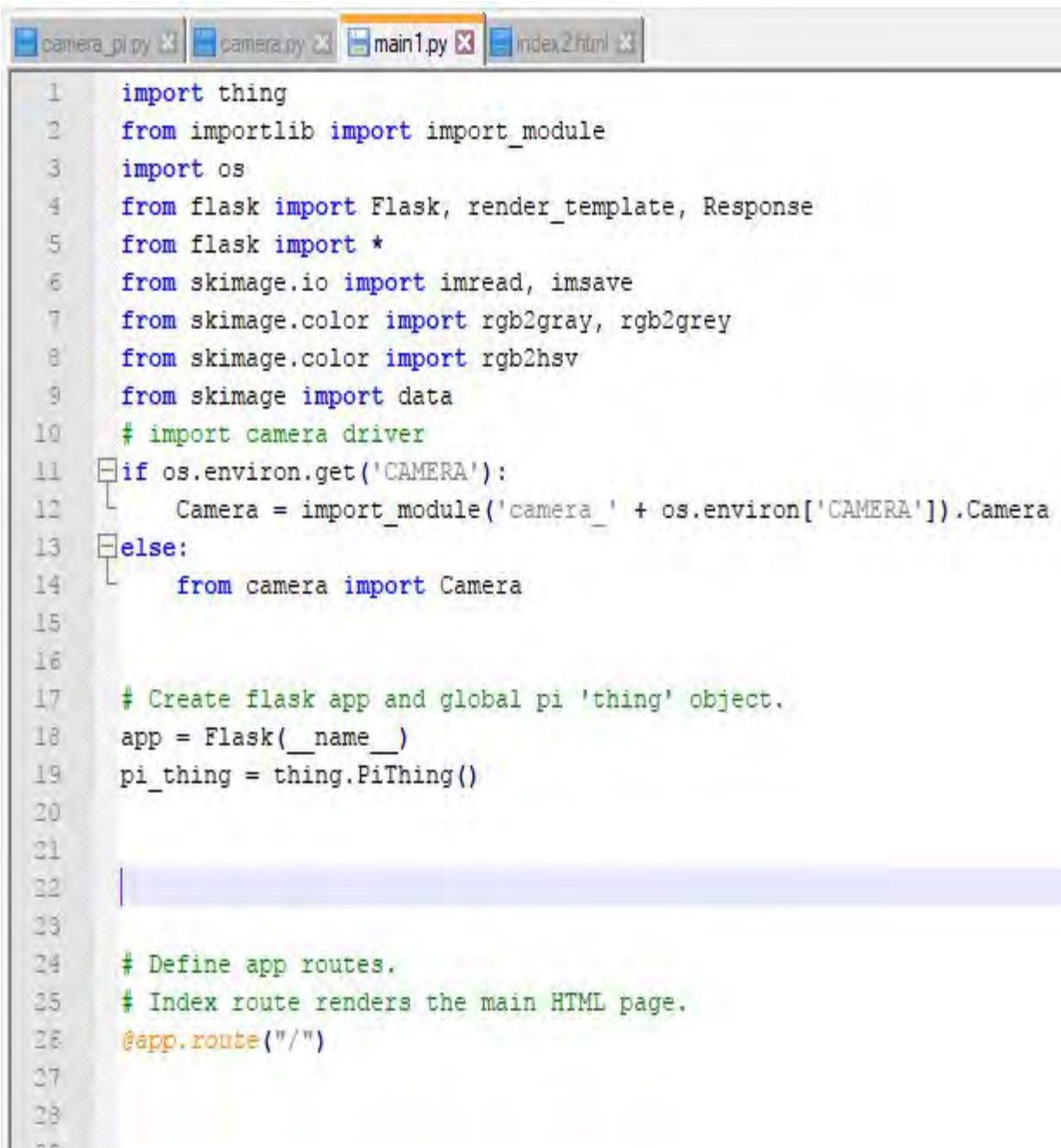
```

Figure 8d: Live Streaming Code

```
1  from skimage.io import imread, imsave
2  from skimage.color import rgb2gray, rgb2grey
3  from skimage.color import rgb2hsv
4  inp_image = imread('/home/pi/imagell1.png')
5
6  img_gray = rgb2gray(inp_image)
7
8
9  hsv_img = rgb2hsv(inp_image)
10 imsave('/home/pi/imagell3.png', hsv_img)
11
```

Figure 9: Code of image processing.

Web site building



```

1  import thing
2  from importlib import import_module
3  import os
4  from flask import Flask, render_template, Response
5  from flask import *
6  from skimage.io import imread, imsave
7  from skimage.color import rgb2gray, rgb2grey
8  from skimage.color import rgb2hsv
9  from skimage import data
10 # import camera driver
11 if os.environ.get('CAMERA'):
12     Camera = import_module('camera_' + os.environ['CAMERA']).Camera
13 else:
14     from camera import Camera
15
16
17 # Create flask app and global pi 'thing' object.
18 app = Flask(__name__)
19 pi_thing = thing.PiThing()
20
21
22
23
24 # Define app routes.
25 # Index route renders the main HTML page.
26 @app.route("/")
27
28
29

```

Figure 10a: Code for the Webpage

```

29
30 def index():
31     # Read the current switch state to pass to the template.
32     inp_image = imread('/home/pi/imagell1.png')
33     y = inp_image.mean()
34     switch = y
35     # Render index.html template.
36     return render_template('index2.html', switch=switch)
37 def gen(camera):
38     """Video streaming generator function."""
39     while True:
40         frame = camera.get_frame()
41         yield (b'--frame\r\n'
42              b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')
43
44     # LED route allows changing the LED state with a POST request.
45     @app.route('/video_feed')
46 def video_feed():
47     """Video streaming route. Put this in the src attribute of an img tag."""
48     return Response(gen(Camera()),
49                    mimetype='multipart/x-mixed-replace; boundary=frame')
50
51     @app.route('/left_side')
52 def left_side():
53     data1 = "LEFT"
54     GPIO.output(m11 , 0)
55     GPIO.output(m12 , 0)
56     GPIO.output(m21 , 1)
57     GPIO.output(m22 , 0)
58     return 'true'

```

Figure 10b: Code for the Webpage

```

59     @app.route('/right_side')
60     def right_side():
61         data1 = "RIGHT"
62         pi_thing.set_led(1)
63         return 'true'
64     @app.route('/up_side')
65     def up_side():
66         data1 = "FORWARD"
67         pi_thing.set_led(1)
68
69
70
71         return 'true'
72     @app.route('/down_side')
73     def down_side():
74         data1 = "BACK"
75
76
77         return 'true'
78     @app.route('/stop')
79     def stop():
80         data1 = "STOP"
81
82
83         return 'true'
84

```

Figure 10c: Code for the Webpage

```

85     @app.route("/led/<int:state>", methods=['POST'])
86     def led(state):
87         # Check if the led state is 0 (off) or 1 (on) and set the LED accordingly.
88         if state == 0:
89             pi_thing.set_led(0)
90         elif state == 1:
91             pi_thing.set_led(1)
92         else:
93             return ('Unknown LED state', 400)
94         return ('', 204)
95
96
97     # Start the flask debug server listening on the pi port 5000 by default.
98     if __name__ == "__main__":
99         app.run(host='0.0.0.0', debug=True, threaded=True)
100

```

Figure 10d: Code for the Webpage


```
1 <!doctype html>
2 <html lang="">
3 <head>
4     <meta charset="utf-8">
5     <meta http-equiv="x-ua-compatible" content="ie=edge">
6     <title></title>
7     <meta name="description" content="">
8     <meta name="viewport" content="width=device-width, initial-scale=1">
9     <link rel="stylesheet" href="{{ url_for('static', filename='css/bootstrap.min.css') }}">
10    <!-- Place favicon.ico in the root directory -->
11 </head>
12 <body>
13
14    <!-- Add your site or application content here -->
15    <div class="jumbotron">
16        <div class="container">
17            <h1>Automated Farming machine</h1>
18            <p>with IOT!</p>
19        </div>
20    </div>
21
```

Figure 11a: HTML Code for the Webpage


```

57 <script src="https://code.jquery.com/jquery-1.12.0.min.js"></script>
58 <script>window.jQuery || document.write('<script src="js/vendor/jquery-1.12.0.min.js"></script>')</script>
59 <script>
60     $(document).ready(function() {
61         $('#led_on').click(function() {
62             $.post('/led/1');
63         });
64         $('#led_off').click(function() {
65             $.post('/led/0');
66         });
67     });
68     $( document ).ready(function(){
69     $('#down').on("mousedown", function() {
70     $.get('/down_side');
71     }).on('mouseup', function() {
72     $.get('/stop');
73     });
74     $('#up').on("mousedown", function() {
75     $.get('/up_side');
76     }).on('mouseup', function() {
77     $.get('/stop');
78     });
79     $('#left').on("mousedown", function() {
80     $.get('/left_side');
81     }).on('mouseup', function() {
82     $.get('/stop');
83     });
84     $('#right').on("mousedown", function() {
85     $.get('/right_side');
86     }).on('mouseup', function() {
87     $.get('/stop');
88     });
89     });
90 </script>
91 </body>
92 </html>

```

Figure 11c: HTML Code for the Webpage