A Project Report on

**Automated plant watering system**

Carried out as part of Internship during

**SUMMER 2019**

Submitted in partial fulfillment of the requirement of

Jawaharlal Nehru Technological University for the award of degree of

**BACHELOR OF TECHNOLOGY**

In

**INFORMATION TECHNOLOGY**

Submitted by

**Bhavya Vulchi**

Bearing Roll No. 17311A12E0

**Akhil Kanukula**

Bearing Roll no. 17311A12E1

Under the guidance of

**Mr. Venugopal**

General secretary

Malaysian Township North Phase

And

**Dr. K. Sreerama Murthy**

Associate Professor, Department of IT

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY**

**(Autonomous)**

Yamnampet, Ghatkesar Mandal, Medchal District, Hyderabad – 501301

**ACKNOWLEDGEMENTS**

We would like to express our immense gratitude and sincere thanks to Dr.K.Sreerama Murthy, Associate professor for his guidance, valuable suggestions and encouragement in completing the internship project within the stipulated time.

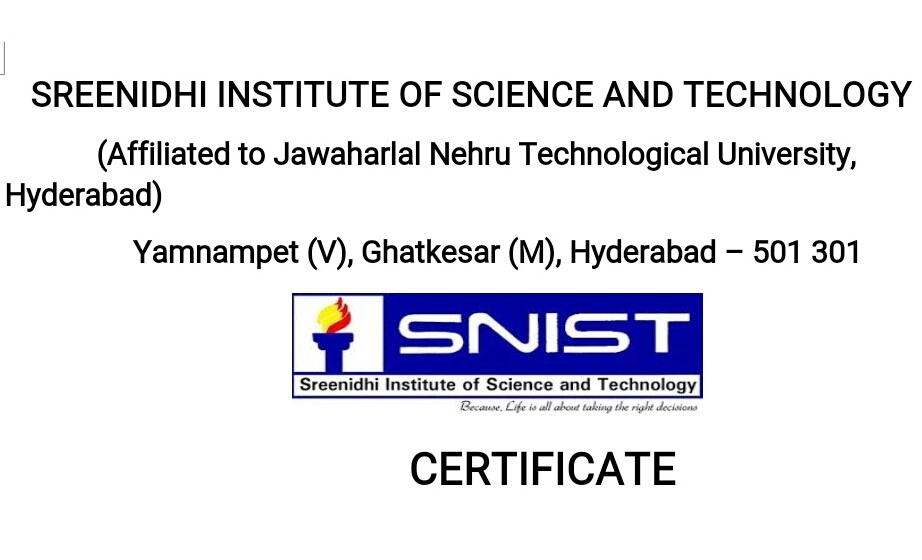
We would like to express our sincere thanks to Dr. P. Narasimha Reddy, Director, Dr. T. CH. Siva Reddy, Principal, Prof. V. V. S. S. S. Balaram, Head of the Department, Information Technology at Sreenidhi Institute of Science and Technology , Ghatkesar for allowing us to do internship

Finally, we would like to thank everyone who have directly or indirectly helped us including parents, relatives and friends for their cooperation in completing the internship.

Bhavya Vulchi – 17311A12E0

Akhil Kanukula – 17311A12E1

+



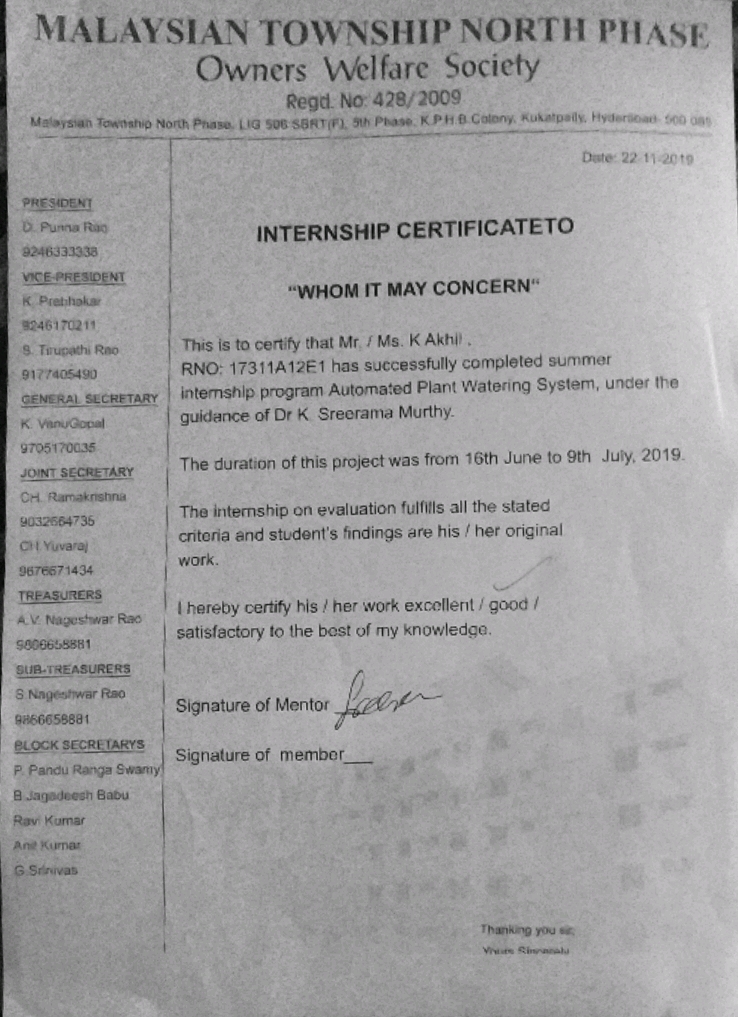
This is to certify that the project report entitled “**AUTOMATED PLANT WATERING SYSTEM**” is being submitted by  **AKHIL KANUKULA(17311A12E1)** in partial fulfilment of the requirements for the award of **B.Tech** degree in **Information Technology** to **Sreenidhi institute of science and technology** affiliated to Jawaharlal Nehru Technological University, Hyderabad(Telangana).This record is a bonafide work carried out by them under our guidance and supervision in the internship program carried out **MTNP**(MALAYSIAN TOWNSHIP NORTHPHASE).The results embodied in the report have not been submitted by any other University or Institution for the award of any degree or diploma.

**Project Coordinator Head of the Department**

Dr. K. Sreerama Murthy Dr.V.V.S.S.S.Balaram

Associate Professor Professor,

Department of IT. Department of IT.



**CONTENTS**

1.Abstract………………………………………………………………...6

2.Introduction…………………………………………………………..7

3.Tools and Technologies used………………………………….8

4.Resources required………………………………………………..9

5.Circuit diagram………………………………………………………12

6.Contributions towards project……………………………….14

7.Design and algorithms used……………………………….….17

8.Outcome……………………………………………………............22

9.Conclusions…………………………………………………………..23

10.Enhancements…………………………………………………….24

11.Bibiliography……………………………………………..………..25

**ABSTRACT**

The automated plant watering system is a project under the domain ‘**Internet Of Things**’ that can detect the water requirements of plants using a soil moisture detection sensor and can automatically turn on and off the supply of water accordingly.

The user has the convenience of setting up the system to either automated or manual mode and can check the status of soil and the details of the last time the plant has been watered.

This is accomplished by creating a webserver that contins certain buttons which provides the choice of the mode of supply and provides access to every other detail.

**INTRODUCTION**

In the presence day, everything is becoming automated. With the advancements in technology day we have to a stage where the communication takes place between any two objects and one such advancement is internet of things.

It is definitely a well known fact that plants are the reason for which human exist on this planet. There has been a lot of imbalance in the ecosystem for years now and for us to take any step further towards the contribution to plant saving, it is important to take care of plants initially and a group of plants or a farm or a crop always begins with one

Also the initial step for taking care of a plant is the monitoring of its requirements and fulfilling them. Sunlight, water and nutrients are the primary needs where sunlight and nutrients are more nature-driven and in this project we focus on the supply of water to a plant.

The water supply is controlled based on the requirements of the plant, neither too much nor too less supply of water should be given and this in turn reduces the wastage of water.

**TOOLS AND TECHNOLOGIES USED**

**IOT**: Internet of Things (IoT) is an ecosystem of connected physical objects that are accessible through the internet. The ‘thing’ in IoT could be a person with a heart monitor or an automobile with a built in sensor. The embedded technology in the objects helps them to interact with internal states or the external environment which in turn effects the decision taken.

IoT can connect devices embedded in various systems to the internet. The connectivity then helps us capture more data from more places ensuring more ways of improving efficiency and improving safety and IoT security.

Raspberry Pi: In this project a raspberry pi 3 B+ model is used to establish the hardware-software interface

It is a small single board credit card sized computer. It supports various operating systems such as Raspbian, ubuntu MATE, windows 10 IoT Core , ubuntu core, PiNet etc. we used Raspbian OS.



Raspberry pi

**1.Raspberry pi 3 b+**

The Raspbian is a Debian-based Linux distribution. It promotes python and scratch as the main programming languages.

The code of the project is written in python3 and it is also required that we install various python libraries for controlling GPIO, set up the webserver etc.

**RESOURCES REQUIRED**

1. Raspberry Pi 3 B+
2. Soil moisture sensor
3. 5V Relay module
4. Flexible water line
5. 3-6 V mini/micro submersible pump
6. 5V power supply
7. Connecting wires

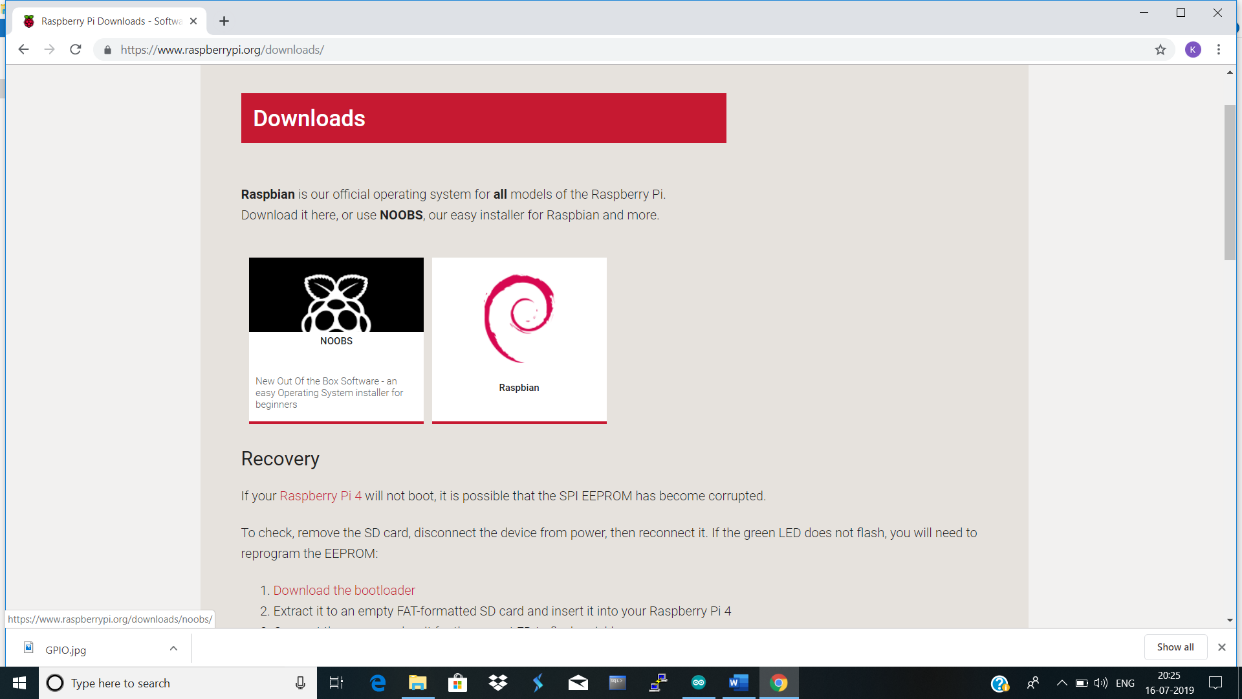
**Raspberry pi:**

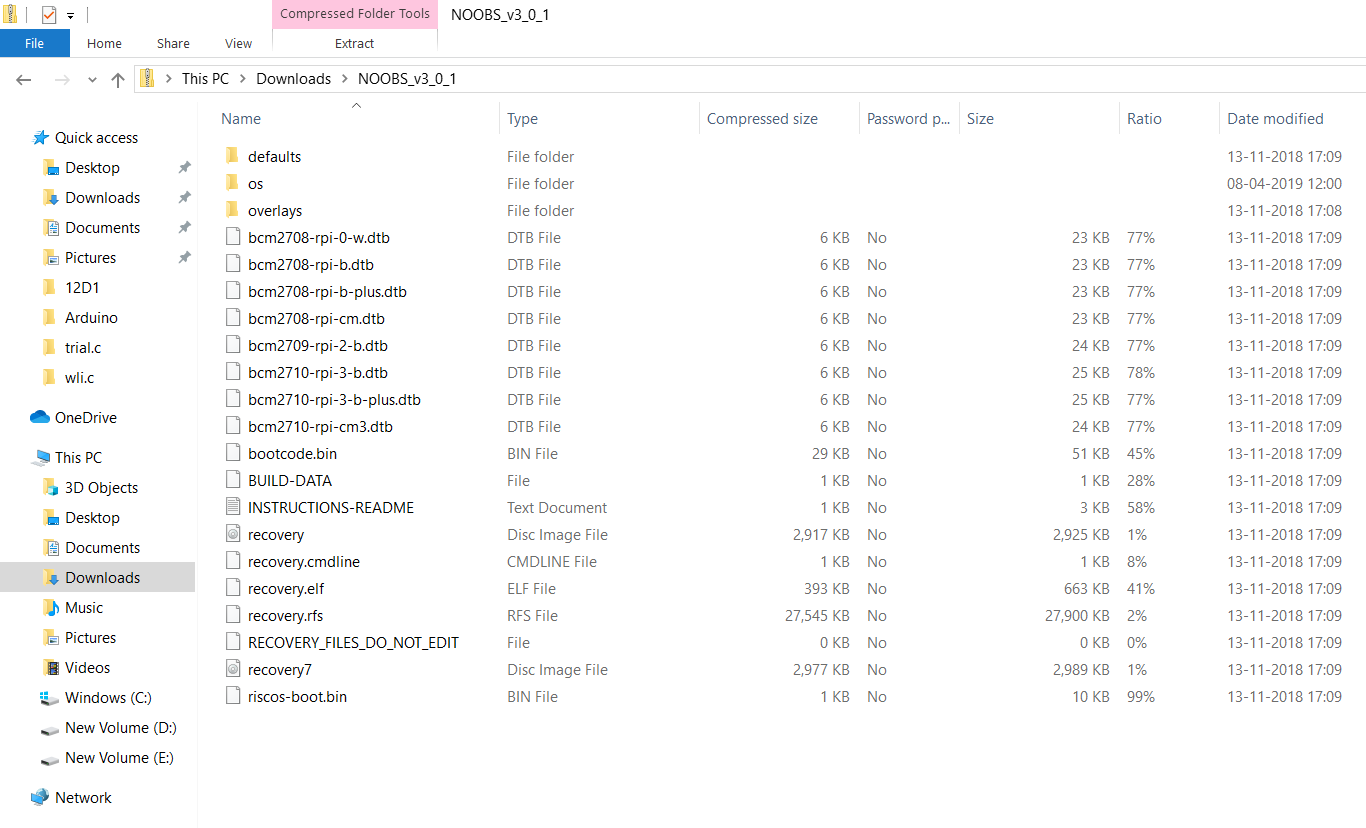
To use raspberry pi we have to initially installed the Raspbian OS (using NOOBS) which was available online at [www.raspberrypi.org](http://www.raspberrypi.org)

The zip files were downloaded into the system and then they were copied to an 16GB storage space class-10 memory card.

This card was inserted into the allotted slot in the raspberry pi and then it was connected to a monitor using required cables and a keyboard and mouse were connected.

Raspberry pi 3 B+ supports wifi and on switching on the set-up you are required to install the required OS and set-up language and time preferences. You are also required to set-up a user name and password.



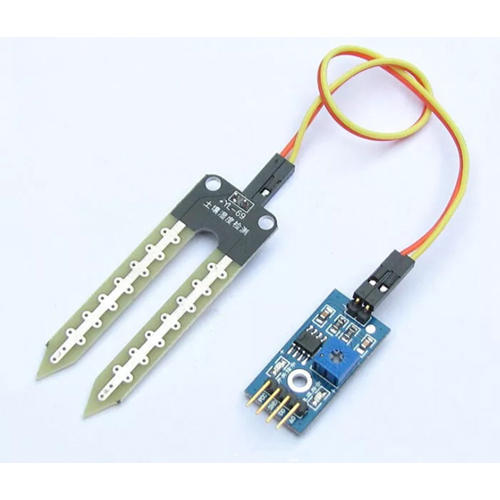


**Soil moisture sensor:**

The Soil moisture sensor is used to measure the water content of soil. It reminds the user to water their plants and also monitors the moisture content. 5V is the working voltage and it has an analog interface type.

This uses capacitance to measure dielectric permittivity of the surrounding medium. In soil dielectric permittivity is a function of the water content. The sensor creates a voltage proportional to the dielectric permittivity, and therefore the water content of the soil.

The sensor averages the water content over the entire length of the sensor. There is a 2cm zone of influence with respect to the flat surface of the sensor, but it has little or no sensitivity at the extreme edges.



soil moisture sensor

**5V Relay module:**

Relay is an electrically operated switch that can be turn on or off, letting the current go through or not and can be controlled with low voltages.

The six pins on the left side of the relay module and the pins on the right side connect the component that requires low voltage.

The high voltage side has two connectors, each with 3 sockets: common(COM),normally closed(NC) and normally open(NO)

**COM**: common pin

**NC**: the normally closed configuration when we want the relay to be closed by default i.e. current flows until a signal is sent.

**NO**: It is opposite to NC. The relay is always open and the circuit is open unless you send the signal to close the circuit.

The pin wiring:

**GND**: goes to ground

**IN1**: controls first relay

**IN2**: controls second relay only if you are using it

**VCC**: goes to 5v



5v Relay module

**Flexible water line:**

It is connected from the pump to the destination of water supply.

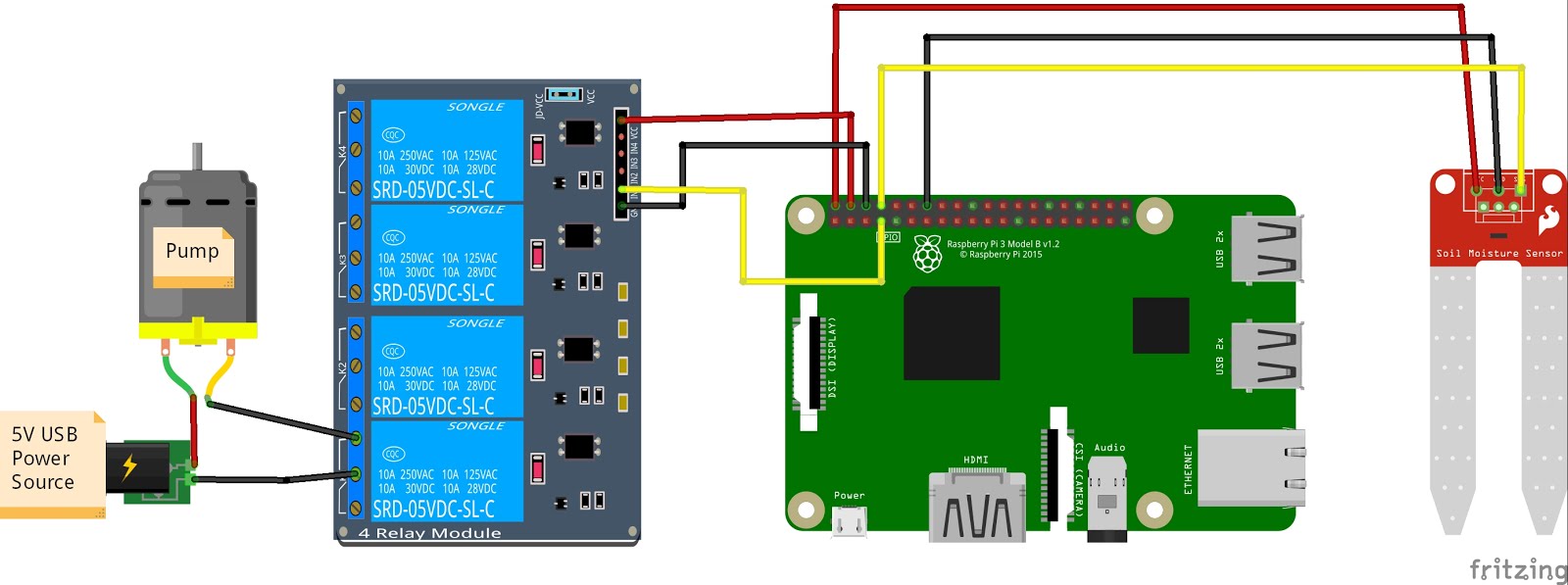
3-6v mini micro submersible pump:

This is immersed inside the water, helps in pumping the water to the surface or to the destination.it is very efficient and it never has to be primed.



5v submersible pump

**Circuit diagram:**



Connect the components according to the circuit diagram

Water Sensor - plug the positive lead from the water sensor to pin 2, and the negative lead to pin 6. Plug the signal wire to pin 8.

Relay - Plug the positive lead from pin 7 to IN1 on the Relay Board. Also connect Pin 2 to VCC, and Pin 5 to GND on the Relay board.

Pump - Connect your pump to a power source, run the black ground wire between slots B and C of relay module 1



**Contribution towards project**

This project was begun by the installation of the Raspbian operating system into a raspberry pi using a sufficiently spacious memory card to store the operating system files and then inserting it into the raspberry pi. the raspberry pi is then initialized and then the hardware setup was done.

Initially to test the working of the moisture sensor and to get ourselves acquainted with the usage of the raspberry pi, a simple experiment was performed in which the presence of moisture was tested. The output said if there was water or if there was no water in the sensor environment.

After the execution of this basic test we headed towards the installation of the required libraries for the execution of the actual pyhton code that was used for the automated watering of the plant

The code contained 4 sections, one of which includes the html code for a web page and the others for running, automated mode and interaction with the web page

After finishing all the pre-requisites for the set up of the circuit, we connected the entire hardware. The 5V power supply was given from the general source by using a mobile phone charger and also another cable was used for the power supply to the mini submersible pump that we used.

The soil moisture detection sensor was connected to the raspberry pi GPIO pins according to their voltage requirements. A 5V supply was given

The mini submersible pump was partially connected to the direct power source through a 5V adapter and then the other parts of the supply ant the pump to the relay module on the high voltage side

The other part of the relay module was connected to the raspberry pi GPIO pins accordingly and the user had the control over the execution of the process

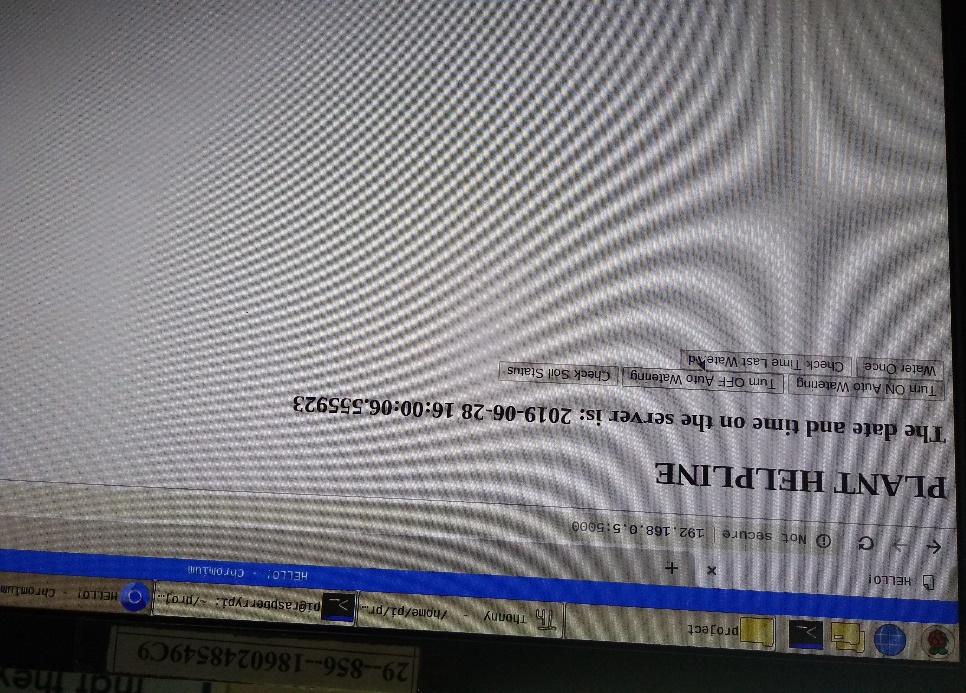
Jumper cables of different kinds were used to make all the types of connections from sensor and the relay module to the raspberry pi

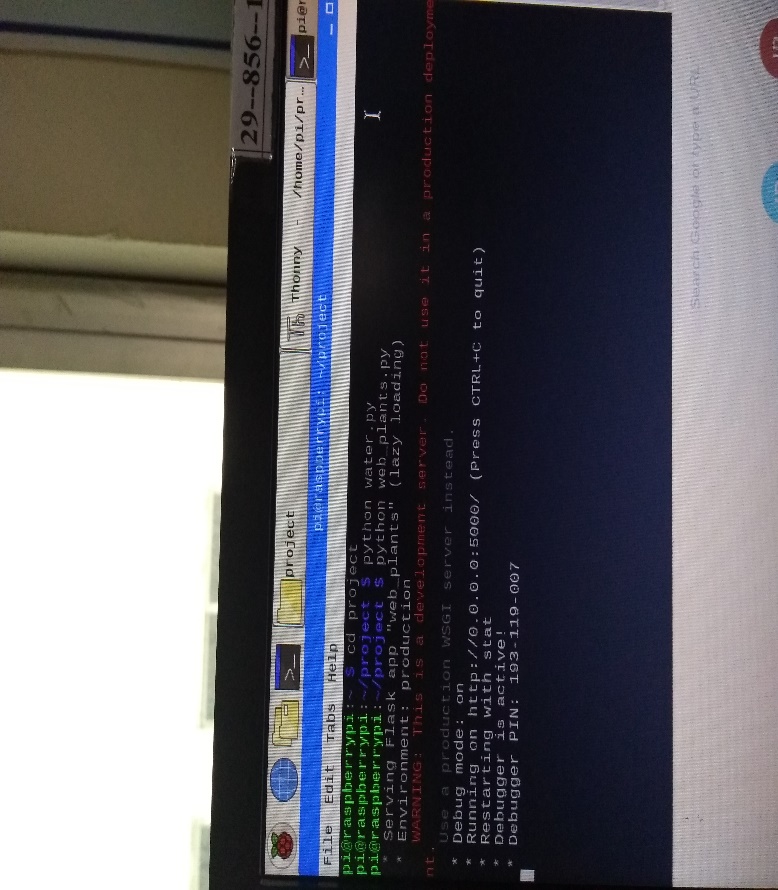
The power supply was then turned on and the programs were executed in the order of their code



After the programs are executed, the web page is opened using the address that is obtained after the execution of the program. This web-page provides buttons which when accessed gives the required results. The buttons include an option to check the state of the plant i.e , that tells if the plant needs water, a manual watering choice named “water once”, a button to check the last time the plant was watered, it gives us both date and time and another button is used to turn the auto watering mode on

In the auto watering mode, the sensor gives information about whether the plant requires water or not and in case of lack of water, the supply is turned on automatically else if there is water available, the pump will remain switched off until another signal indicating the need of water is sent by the sensor.





**Design and algorithms used**

The code for the project was written in the python language. Various libraries were installed to serve different purposes such as interaction with RPi GPIO, connecting the program to the web server etc.

$> python3 -m pip install RPi.GPIO

This code is used to control the GPIO.

|  |
| --- |
| # External module imp |
|  | import RPi.GPIO as GPIO |
|  | import datetime |
|  | import time |
|  |  |
|  | init = False |
|  |  |
|  | GPIO.setmode(GPIO.BOARD) # Broadcom pin-numbering scheme |
|  |  |
|  | def get\_last\_watered(): |
|  | try: |
|  | f = open("last\_watered.txt", "r") |
|  | return f.readline() |
|  | except: |
|  | return "NEVER!" |
|  |  |
|  | def get\_status(pin = 8): |
|  | GPIO.setup(pin, GPIO.IN) |
|  | return GPIO.input(pin) |
|  |  |
|  | def init\_output(pin): |
|  | GPIO.setup(pin, GPIO.OUT) |
|  | GPIO.output(pin, GPIO.LOW) |
|  | GPIO.output(pin, GPIO.HIGH) |
|  |  |
|  | def auto\_water(delay = 5, pump\_pin = 7, water\_sensor\_pin = 8): |
|  | consecutive\_water\_count = 0 |
|  | init\_output(pump\_pin) |
|  | print("Here we go! Press CTRL+C to exit") |
|  | try: |
|  | while 1 and consecutive\_water\_count < 10: |
|  | time.sleep(delay) |
|  | wet = get\_status(pin = water\_sensor\_pin) == 0 |
|  | if not wet: |
|  | if consecutive\_water\_count < 5: |
|  | pump\_on(pump\_pin, 1) |
|  | consecutive\_water\_count += 1 |
|  | else: |
|  | consecutive\_water\_count = 0 |
|  | except KeyboardInterrupt: # If CTRL+C is pressed, exit cleanly: |
|  | GPIO.cleanup() # cleanup all GPI |
|  |  |
|  | def pump\_on(pump\_pin = 7, delay = 1): |
|  | init\_output(pump\_pin) |
|  | f = open("last\_watered.txt", "w") |
|  | f.write("Last watered {}".format(datetime.datetime.now())) |
|  | f.close() |
|  | GPIO.output(pump\_pin, GPIO.LOW) |
|  | time.sleep(1) |
|  | GPIO.output(pump\_pin, GPIO.HIGH) |

This is the water.py code that is used to interact with the GPIO

The other code:

This python script enables various actions

The installations required to run this code are:

$> python3 -m pip install flask  
$> python3 -m pip install psutil

the program web\_plants.py

|  |
| --- |
| from flask import Flask, render\_template, redirect, url\_for |
|  | import psutil |
|  | import datetime |
|  | import water |
|  | import os |
|  |  |
|  | app = Flask(\_\_name\_\_) |
|  |  |
|  | def template(title = "HELLO!", text = ""): |
|  | now = datetime.datetime.now() |
|  | timeString = now |
|  | templateDate = { |
|  | 'title' : title, |
|  | 'time' : timeString, |
|  | 'text' : text |
|  | } |
|  | return templateDate |
|  |  |
|  | @app.route("/") |
|  | def hello(): |
|  | templateData = template() |
|  | return render\_template('main.html', \*\*templateData) |
|  |  |
|  | @app.route("/last\_watered") |
|  | def check\_last\_watered(): |
|  | templateData = template(text = water.get\_last\_watered()) |
|  | return render\_template('main.html', \*\*templateData) |
|  |  |
|  | @app.route("/sensor") |
|  | def action(): |
|  | status = water.get\_status() |
|  | message = "" |
|  | if (status == 1): |
|  | message = "Water me please!" |
|  | else: |
|  | message = "I'm a happy plant" |
|  |  |
|  | templateData = template(text = message) |
|  | return render\_template('main.html', \*\*templateData) |
|  |  |
|  | @app.route("/water") |
|  | def action2(): |
|  | water.pump\_on() |
|  | templateData = template(text = "Watered Once") |
|  | return render\_template('main.html', \*\*templateData) |
|  |  |
|  | @app.route("/auto/water/<toggle>") |
|  | def auto\_water(toggle): |
|  | running = False |
|  | if toggle == "ON": |
|  | templateData = template(text = "Auto Watering On") |
|  | for process in psutil.process\_iter(): |
|  | try: |
|  | if process.cmdline()[1] == 'auto\_water.py': |
|  | templateData = template(text = "Already running") |
|  | running = True |
|  | except: |
|  | Pass |
|  | if not running: |
|  | os.system("python3.4 auto\_water.py&") |
|  | else: |
|  | templateData = template(text = "Auto Watering Off") |
|  | os.system("pkill -f water.py") |
|  |  |
|  | return render\_template('main.html', \*\*templateData) |
|  |  |
|  | if \_\_name\_\_ == "\_\_main\_\_": |
|  | app.run(host='0.0.0.0', debug=True) |

Now for the automated watering :

This can be run by the command:

python3 auto\_water.py

|  |
| --- |
| import water |
|  |  |
|  | if \_\_name\_\_ == "\_\_main\_\_": |
|  | water.auto\_water() |

Now the html page should be stored in a sub-directory named “templates” that is a part of the directory in which the other parts of the code are saved.

The code is:

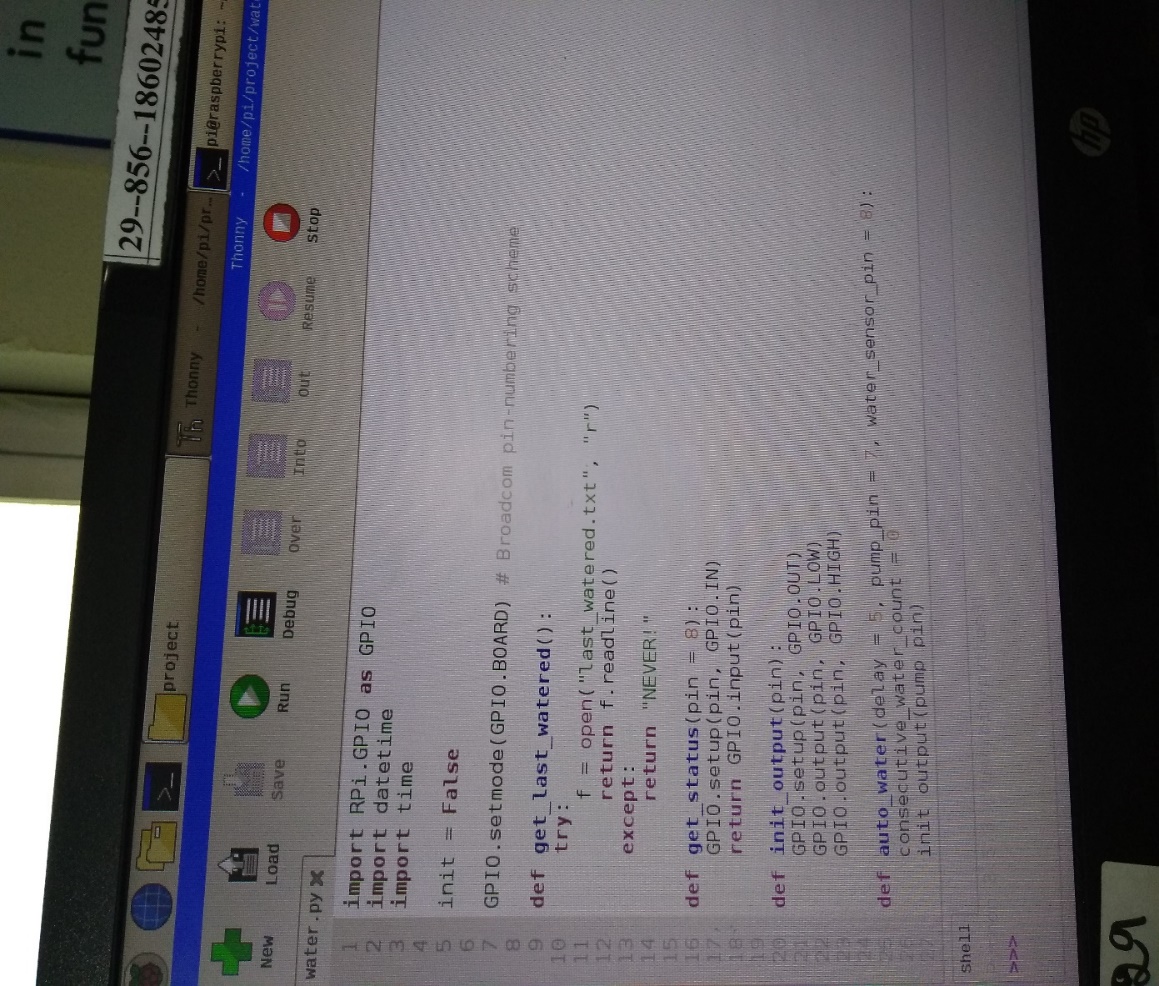
|  |
| --- |
| <!DOCTYPE html> |
|  | <head> |
|  | <title>{{ title }}</title> |
|  | </head> |
|  |  |
|  | <body> |
|  | <h1>PLANT HELPLINE</h1> |
|  | <h2>The date and time on the server is: {{ time }}</h2> |
|  | <h2> {{ text }} </h2> |
|  | <a href="/auto/water/ON"><button>Turn ON Auto Watering</button></a> |
|  | <a href="/auto/water/OFF"><button>Turn OFF Auto Watering</button></a> |
|  | <a href="/sensor"><button>Check Soil Status</button></a> <br> |
|  | <a href="/water"><button>Water Once</button></a> |
|  | <a href="/last\_watered"><button>Check Time Last Watered</button></a> |
|  |  |
|  | </body> |
|  | </html> |

To make sure that the website runs automatically once the program start running, we are required to run certain commands that include:

$> sudo crontab -e

@reboot cd <your path to web\_plants>; sudo python3 web\_plants.py

After the execution, when the system reboots and later on you can move forward with the execution of the program.



**Outcome**

After the program is executed , we will be able to monitor and control the supply of water to the plants, we will be able to check if the plant is in need of water or not, we will be able to check the last time we watered the plants, we can allow it to check and water by itself or we can manually turn on the water supply using a button

This method of watering the plants saves time and also the wastage of water. This helps in making the watering easier for humans

The led that is present on the moisture sensor used for indicating the presence of water can be observed with various intensities depending on the content of water and hence this feature of the sensor helps in watering in the manual mode



**Conclusion**

Automated plant watering system using a raspberry pi has made our gardening more efficient as the system can access the status of the soil and water then in time whenever it is required upon the approval or the type designed by the user.

The raspberry pi is a huge platform for developments in the field of object communication. Its advanced features enables us to expand the boundaries of possibilities to explore the technology further.

The python programming makes it simpler for the designer and the user to analyze and modify the code according to the requirements with frequent updates in the technology.

This hence encourages the growth of plants indoor too in an effective manner that gives valid contribution in the busy schedule of people’s lives in the present day.

**Enhancements**

There are a huge number of advancements that can be made to this system in various aspects of technology

The first one is the introduction of a third party which is the cloud. In the current process we can control the watering system using a plant using a local webserver but the water supply can also be done using cloud where you can use a mobile phone to take actions, give permissions and monitor the growth of the plant.

The growth can also be improved by taking care of the other requirement of a plant growth which is the nutrients content present in the soil. The nutrients required by a specific type of plant and also the level of water required by different plants can also be included in the part of the code for having them checked and then the actions be performed accordingly.

The same enhancement can further be improved to be taken to the machine learning level where the system itself can detect or predict the requirements of the soil by learning from few examples by its own.

To control the water supply with phone, it can be seen that the software is supported by a wide range of operating systems and their versions so that any category of a person, irrespective of his capability to afford a mobile with the latest software is be able to use it.

We can also add image processing to monitor the growth of the plant and predict its requirements in terms of nutrients and pesticides too.

It can also be designed to take everyday weather as input and control the amount of water supplied based on the climate.

**Bibliography**

<http://www.cyber-omelette.com/2017/09/automated-plant-watering.html>

[www.youtube.com/watch?v=mQNJpWkdmbc](http://www.youtube.com/watch?v=mQNJpWkdmbc)

[www.raspberrypi.org](http://www.raspberrypi.org)

<https://gist.github.com/benrules2/c4f3db455f4f2dfbe7d5b825b0b4ee36#file-web_plants-py>

<https://gist.github.com/benrules2/c4f3db455f4f2dfbe7d5b825b0b4ee36#file-auto_water-py%EF%BB%BF>

<https://gist.github.com/benrules2/e43c469b2c1263237dc67010fca18b53>

<https://www.youtube.com/watch?v=wV0RoQ6Zb9c>

<https://stackoverflow.com/>