INTRODUCTION :

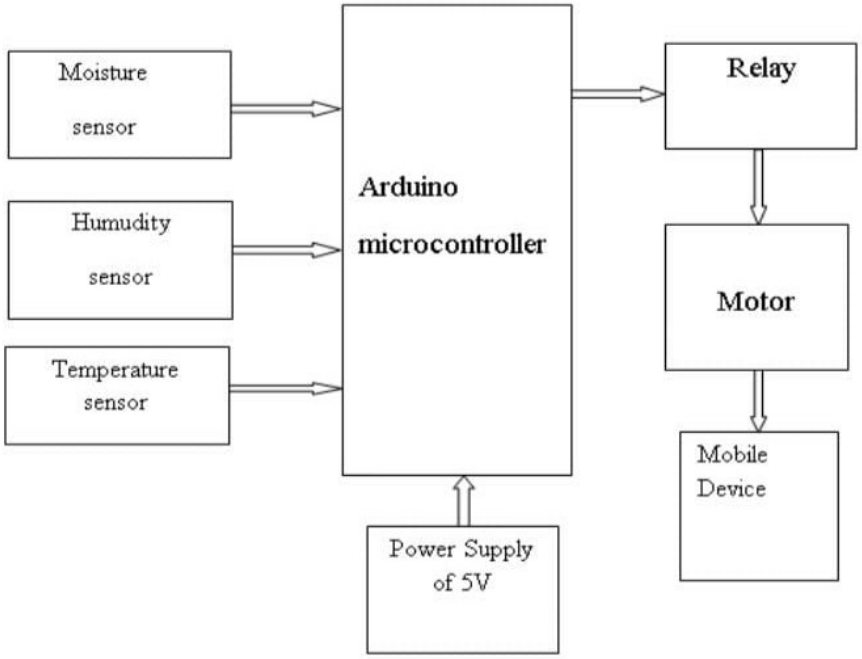
Smart sprinkler system is a Internet Of Things(IOT) based system which is used in the golf course for the purpose of sprinkling of the water based on the weather conditions, humidity, soil moisture etc . By developing a software using internet of things we can develop a real time smart water sprinkler system in the golf course , so it may help in lot of ways

The main purpose of this Real time weather based Smart Sprinkler System is to sprinkle the water in the golf course based on the weather conditions . So we can avoid the wastage of water and we had to do only less work. And different parts of the golf course gets amounts amount of water they required.

LITERATURE SURVEY :

The actual problem of watering the golf course is it's very difficult and some parts of the course used to get heavy water and some parts get less water. So while we playing golf it may become difficult for us due to the insufficient and over water at some of the places.

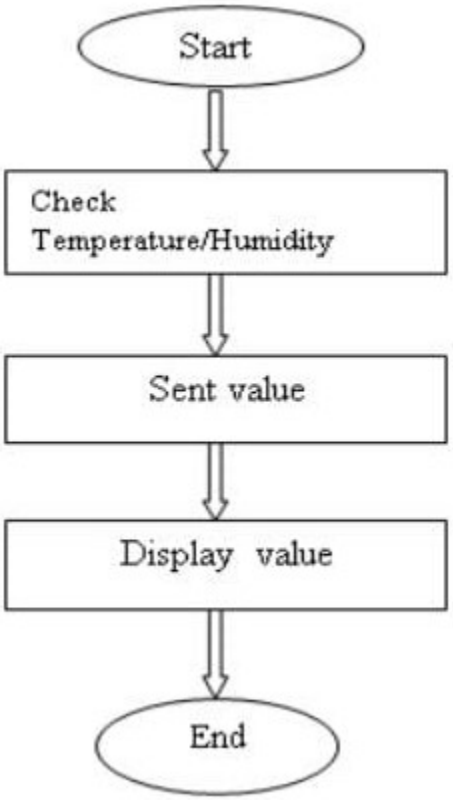
The proposed solution for this problem is the real time weather based Smart Water Sprinkler System , so that we can supply the water to the golf course based on the weather conditions , so that different parts of the golf course gets sufficient amount of water.

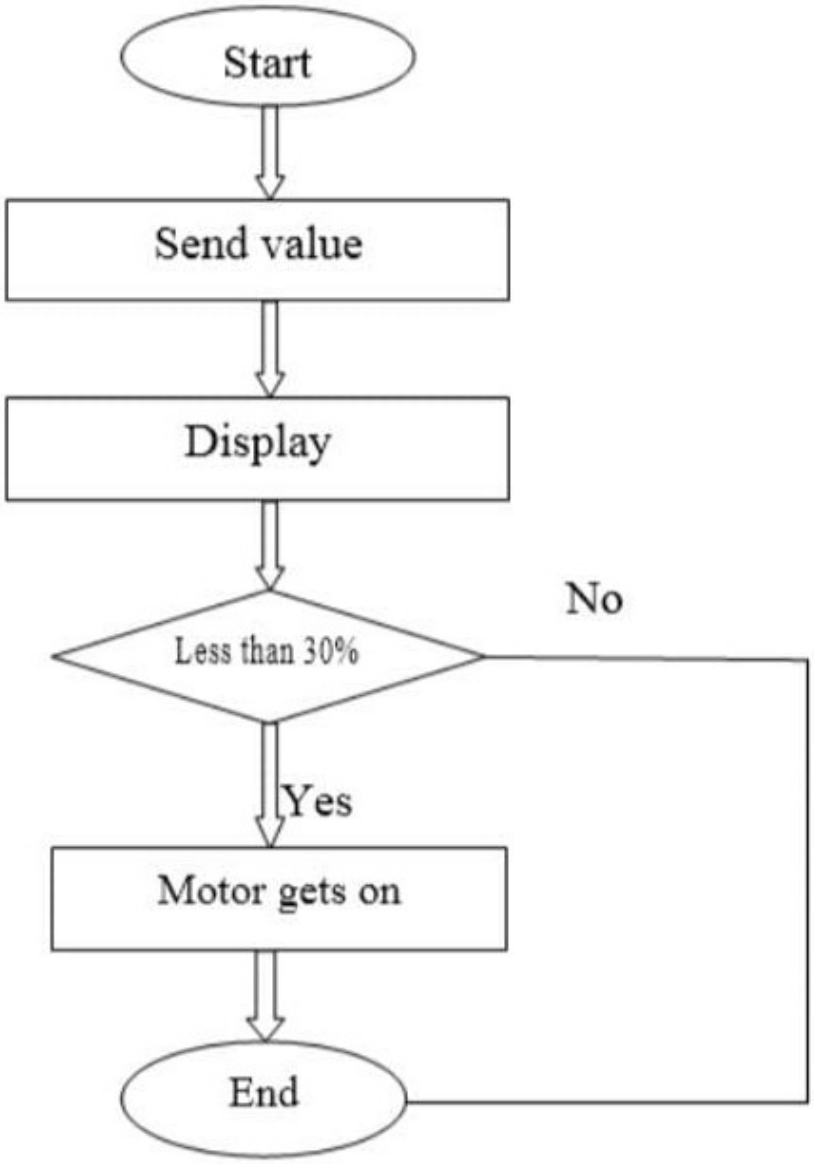


THEORITICAL ANALYSIS :

The sensor that is used in this smart sprinkler Dht11 sensor. Here we had taken the random value of the sensor and we had executed our sensor by using a Python code. Now we had to run the python code and it connected to out Node red services. After the completion of the process in the node red services we had to create a MIT app and connect to this Node red services. Now by using this mobile apk we can connect it to the Motor. So now based on the Motor ON/OFF condition the sprinkler will sprinkle the water in the golf course. Here by using the hardware like Raspberrypi or Aurdino we can detect the exact value of solid moisture using a Dht11 sensor.

FLOWCHART


Start
Check
Temperature/Humidity
Sent value
Display value
End


Start
Send value
Display
Less than 30%
No
Yes
Motor gets on
End

RESULT:

Hence we had created a Realtime Weather based Smart Sprinkler System for a Golf Course based on Internet Of Things (IOT).

ADVANTAGES :

One of the most obvious advantages is the time savings afforded by an automatic sprinkler or drip irrigation system. Once installed, many systems can be set to a timer to water at specific time intervals and on certain days of the week. This means there's no need to worry about forgetting to water the lawn and coming back from vacation to find crisp, yellow grass.

Another advantage is that irrigation systems, particularly the drip type, can be positioned so that water is more effectively targeted where it is needed. Nozzles can be adjusted, and underground drip tubes will deliver water right to the roots, rather than spraying walkways and driveways.

Another advantage is that automatic irrigation systems are generally hidden from view, which means there are no unsightly hoses stretched across the lawn and no more tripping hazards. Sprinkler heads pop up to spray and then retract when the job is done. Underground drip systems do their work out of view. For families with young children and

pets who share outdoor spaces, automatic systems may be a safer option.

DISADVANTAGES :

The primary disadvantage associated with a sprinkler system is the expense. These systems can be quite costly depending on the size of the property. Furthermore, portions of the lawn will have to be dug up to install pipework and attach it to the plumbing system of the home. This can equate to days or weeks without use of the yard. Afterward, the landscaping will have to be repaired.

It is best to install an irrigation system before installing sod or extensive landscaping because some of it will have to be torn up. Homeowners who already have pristine yards may be turned off by this reality.

Even the most efficient sprinkler systems can have their pitfalls. Wind can wreak havoc on sprinklers, directing water in the wrong direction. Underground pests may damage water-delivery systems, resulting in water pooling or broken parts. The repairs to fix an irrigation system can be much more costly than replacing a damaged garden hose.

Irrigation systems have their advantages and disadvantages, and homeowners should weigh their options before installing a new system.

Once installed, many systems can be set to a timer to water at specific time intervals and on certain days of the week.

APPLICATIONS :

Smart irrigation is **used** in residential **applications** such as
gardens
small plantations
lawns
agriculture
Residential users are employing advanced technology-oriented **irrigation systems** controllers and various sensors to minimize water wastage and water bills.

CONCLUSION :

This project involves establishing a contemporary design technique of monitoring and controlling the moisture level of soil using Mobile App. Providing comprehensive tools that need
to build any measurement or control application in dramatically less time.
By using the above project we can reduce the wastage of water and and also cam reduce the consumption of excess power
Thus the main moto of the project is to save water and reduce the wastage of power

FUTURE SCOPE :

We can interface LCD scree`in order to display the current status of the soil moisture content levels, percentage of water utilized to the water plant, duration of time for which the water pump is ON ,etc. We can also show the graphical representation of the moisture content levels in the soil. To improve the efficiency and effectiveness of the system , the following recommendations can be put into consideration. Option of controlling the water pump can be given to the farmer. The farmer may choose to stop the growth of crops or the crops may get damaged due to adverse weather conditions. In such cases farmer may need to stop the system remotely. The idea of using IOT in irrigation can be extend further to other activities in farming such as castle managamenet , fire detection and climate control. This may minimize human intervention in farming activities.

BIBILOGRAPHY :

For the project given...

1. We installed python software in our laptop/pc.
2. Then we created an ibm account to access the required platform.
3.Further we created a nodered application and then an iot watson platform and a mit app inventor to prepare an app for the project
4.Later code snippet required to run the program are created individually as assigned to us
5.Then we created a node red service to get data from device
6.Later we used the dash board nodes to create the UI i.e.,for the web app.

7.Then we created HTTP request to communicate with the mobile app that we designed
8.Now,we designed our mobile app with UI to display temperature, humidity,soil moisture.
9.Then, we configured the mobile application to receive the data from the cloud using HTTP request.
10.Then,we configured the mobile app to control the motor with the buttons we are represented inthe app using our UI.
11.After this,we once again checked everything is working or not and sent the information through the mobile UI and received the required output.
12.Therefore,The project assigned to us is completed successfully

SOURCE CODE :

```
import time
import sys
import ibmiotf.application
import ibmiotf.device

#Provide your IBM Watson Device Credentials
organization = "f2cody" # repalce it with organization ID
deviceType = "rsip" #replace it with device type
deviceId = "9010" #repalce with device id
authMethod = "token"
authToken = "1234567890"#repalce with token

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data)
    if cmd.data['command']=='motoron':
        print("MOTOR ON")
    elif cmd.data['command'] == 'motoroff':
       print("MOTOR OFF")

try:
                        deviceOptions = {"org": organization, "type": deviceType, "id":
deviceId, "auth-method": authMethod, "auth-token": authToken}
                        deviceCli = ibmiotf.device.Client(deviceOptions)
                        #.....................................................

except Exception as e:
```

```
print("Caught exception connecting device: %s" % str(e))
                                    sys.exit()

deviceCli.connect()

while True:
        T=50;
        H=32;
        M=45;
        #Send Temperature,moisture & Humidity to IBM Watson



data ={'d': { 'temperature' : T, 'humidity': H, 'moisture': M}}
        #print data
        def myOnPublishCallback():
           print ("Published Temperature = % s C" % T, "Humidity = % s" % H, "moisture= % s
C/M" % M,"to IBM Watson")

        success = deviceCli.publishEvent("event", "json", data, qos=0,
on_publish=myOnPublishCallback)
        if not success:
           print("Not connected to IoTF")
        time.sleep(1)

        deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud
deviceCli.disconnect()
```