**EXTERNSHIP**

**ON**

**INTERNET OF THINGS**



**A Project Report ON**

**REAL TIME WEATHER BASED SMART SPINKLER**

**FOR GOLF COURSE**

**CONTENTS**

1. Introduction
   * Overview
   * Purpose
2. Literature Survey

* Existing problem
* Proposed solution

1. Theoretical Analysis

* Hardware/Software designing

1. Experimental Investigations
2. Flowchart
3. Result
4. Advantages & Disadvantages
5. Applications
6. Conclusion
7. Future Scope
8. Bibliography
9. Appendix

* Source code
* UI output Screenshot

**INTRODUCTION**

* **OVERVIEW**

Sprinkler irrigation is a method of providing rainfall-like irrigation to the crops. Water is distributed through a system of pipes usually by pumping. Spray heads at the outlets distribute the water over the entire soil surface.

Region-wise spread of sprinkler irrigated area in 2006 was

* Americas 13.3 Mio. ha
* Europe 10.1 Mio. ha
* Asia 6.8 Mio ha
* Africa 1.9 Mio. ha
* Oceania 0.9 Mio. ha

The top ten sprinkler irrigated countries were USA, Russia, China, India, France, Brazil, Italy, Spain, Saudi Arabia and Ukraine. These countries together constituted 75% of total sprinkler-irrigated area.

A typical sprinkler irrigation system consists of the following components:

* Pump unit;
* Mainline (and sometimes sub-mainlines);
* Lateral lines;
* Sprinklers.

The pump unit is usually a centrifugal pump, which takes water from the source and provides adequate pressure for delivery into the pipe system.

Mainline and sub-mainline pipes deliver water from the pump to lateral pipes. In some cases these pipelines are permanent and are laid on the soil surface or buried below ground. In other cases they are temporary, and can be moved from field to field.

* **PURPOSE**

Smart sprinkler controllers let you turn your sprinklers on and off remotely, through an app. ... For example, if rain is in the forecast, the smart sprinkler controller will halt the watering schedule for that day or adjust the watering time depending on how much rain fell

**LITERATURE SURVEY**

* **EXISTING PROBLEM**

The existing system requires manual work. Every time the people should monitor the weather condition and sprinkler system and have to on it at a period of time.

* **PROPOSED SOLUTION**
* Adjustments. Make sure your sprinklers are spraying onto the grass and not on sidewalks, driveways, or the street.
* Runoff. Make sure you are not running your sprinklers too long on sloped or compacted soils.
* Obstructions. Make sure your sprinklers spray pattern is not blocked.

**THEORETICAL ANALYSIS**

* **HARDWARE/SOFTWARE DESIGNIN**

We developed this application with some software designations:

* IBM Cloud
  + iot watson platform
  + ibm cloudant
  + Node-red
* Python idle software
* MIT APP inventor

**EXPERIMENTAL INVESTIGATIONS**

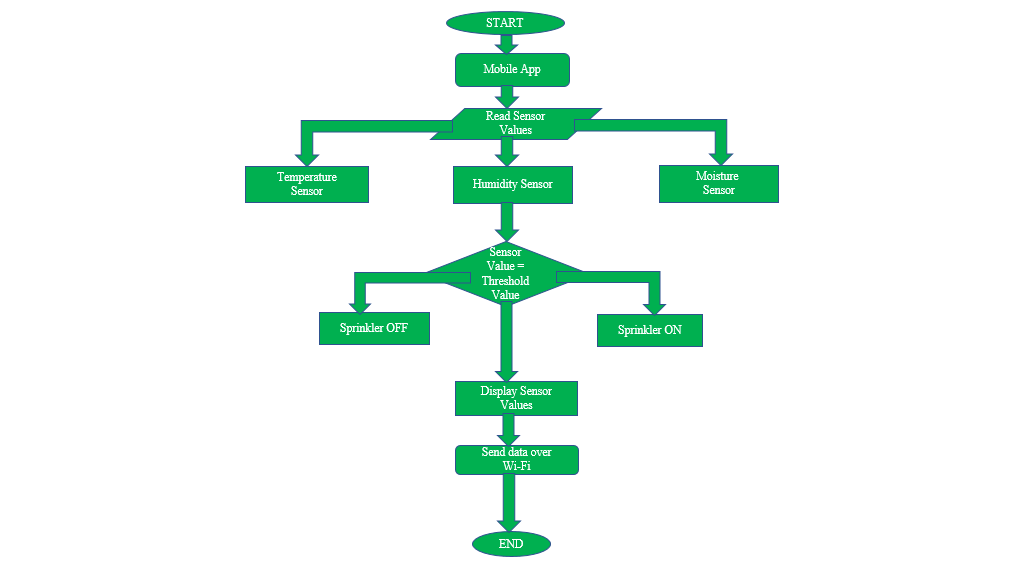
Whenever it rains, the smart sprinkler app’s going to make things so much easier for you. Instead of rushing out toward the shed to flip off the sprinkler, or just letting it slide because you’re lazy, hence wasting water, one can prompt shut it off through its app.

You won’t even have to get that done if you have the best smart sprinkler system in your yard. Most, if not all, smart sprinklers have the capability of monitoring the weather for you. By doing so, they automatically turn off when it rains, meaning, you won’t see the sprinklers running off.

Smart sprinklers are in general aware of rain in their weather forecast and will adapt the watering schedule accordingly. Moreover, apart from adapting your schedule to the weather, they also set a watering schedule from scratch.

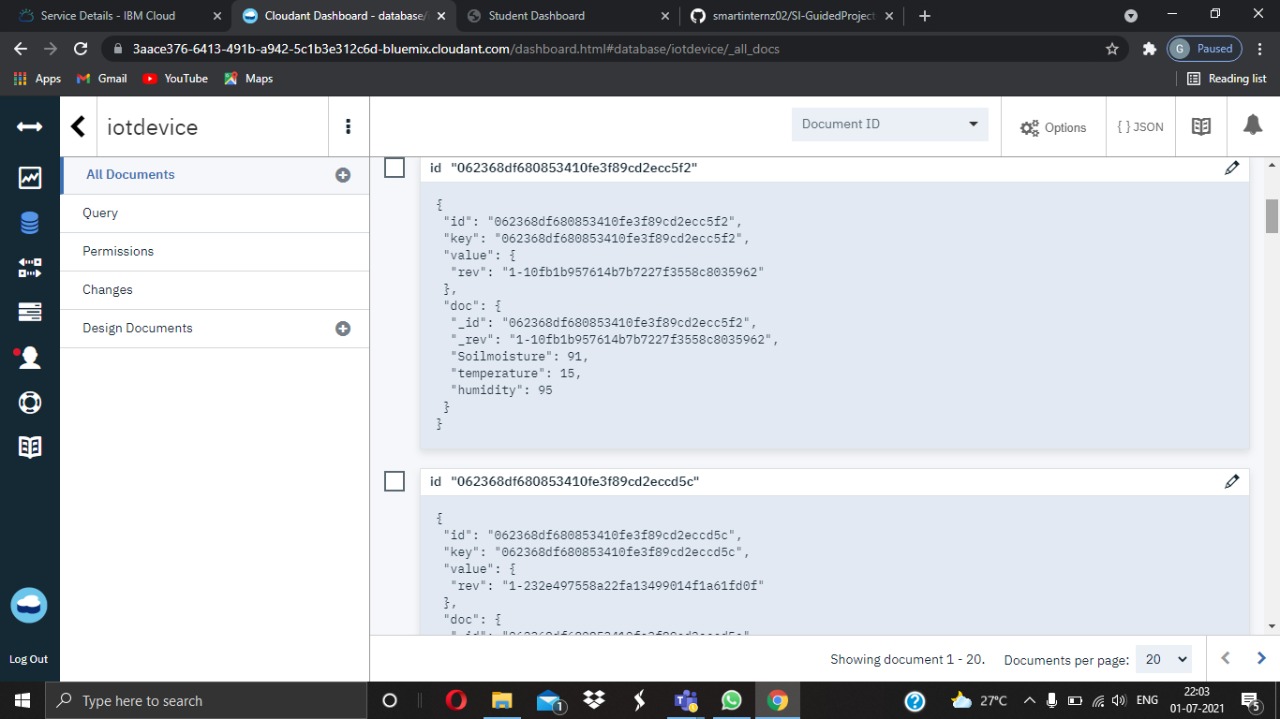
You no longer have to think about when the best times to water are and how often it should be done. Your smart sprinkler app will manage it all for you.

**FLOWCHART**

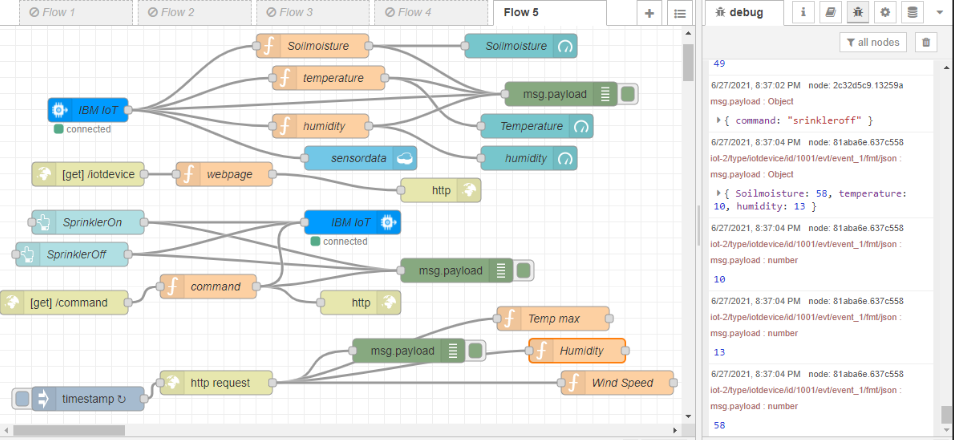


**RESULT**

* DATA STORED IN CLOUD database:

****

* NODE-RED:

****

**ADVANTAGES & DISADVANTAGES**

* **ADVANTAGES**
* Expansive land leveling or terracing is not required;
* Suitable for almost all soil types;
* Water saving intensity can be changed in accordance with the infiltration capacity of soil and crop water requirements;
* High efficiency due to uniform water distribution can be adapted to growth stage and conditions;
* Lower labor requirements as compared to traditional surface irrigation approaches
* **DIS-ADVANTAGES**
* High initial capital costs (investment in equipment - sprinklers and pipes) and high operation costs due to energy requirements for pumping and labor costs.
* Sensitivity to wind, causing evaporation losses (under high wind condition and high temperature distribution and application efficiency is poor);
* Unavoidable wetting of foliage in field crops results in increased sensitivity to diseases;
* Highly saline water (>7 millimhos/cm) causes leaf burning when temperature higher than 35 degrees (Celsius).

**APPLICATIONS**

* Non-Agricultural
  + Residential
  + Turf and Landscape
  + Golf Courses
  + Others (Others include non-agriculture applications include pastures land, parks, school or university campuses, industrial buildings, sports grounds, and forestry)
* Agricultural
  + Greenhouses
  + Open field

**CONCLUSION**

We hereby conclude that Smart sprinkler systems have the amazing ability to deliver just the right amount of water needed for your landscape every day. When you’re saving water, you’re saving money on the monthly water bills. A smart sprinkler can adjust the water levels, hence keeping our lawns beautiful. We recommend that you split your lawn areas in the smart sprinkler program as it have different watering needs. Also, smart sprinklers are able to calculate just how fast water evaporates and monitor when it’s going to pour. This way, less water is wasted and your lawn will look fresh as ever.

**FUTURE SCOPE**

The proposed system consists of less hardware as compared to the previous model hence it is compact as compared to the previous system. It is more cost efficient, this claim is made on the fact that the proposed system does not need the heavy and expensive hardware for implementation. This type of automated irrigation system consumes 40-50% less water as compared to the traditional system Ideal growth condition is been provided when small amount of water is been applied over large amount of time. This smart irrigation system extends watering time for plants, and provides ideal growth condition. It saves time and timer delay as per the environmental condition can be added for automatic watering. This smart irrigation system can be adjusted and modified according to the changing environment. It is simple to operate it starts by designing the map of your garden and marking the location of planting

**BIBILOGRAPHY**

* https://cloud.ibm.com/resources
* https://uh8nhd.internetofthings.ibmcloud.com/dashboard/devices
* https://node-red-gkdvc-2021-06-11.mybluemix.net/ui
* http://ai2.appinventor.mit.edu

**APPENDIX**

* **SOURCE CODE**

import time

import sys

import ibmiotf.application

import ibmiotf.device

import random

import json

#Provide your IBM Watson Device Credentials

organization = "x8sfnd"

deviceType = "iotdevice"

deviceId = "1001"

authMethod = "token"

authToken = "1234567890"

# Initialize the device client.

W=0

S=0

def myCommandCallback(cmd):

print("Command received: %s" % cmd.data['command'])

if cmd.data['command']=='sprinkleron':

print("Sprinkler ON IS RECEIVED")

elif cmd.data['command']=='sprinkleroff':

print("Sprinkler OFF IS RECEIVED")

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()

# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times

deviceCli.connect()

while True:

W=89

S=44

#Send Temperature & Humidity to IBM Watson

data = jsondata={"d":{ 'water level' : W, 'soil moisture':S }}

print (data)

def myOnPublishCallback():

print ("Published Water Level = %s units"% W, "Soil moisture = %s %%" % S, "to IBM Watson")

success = deviceCli.publishEvent("Data", "json", data, qos=0, on\_publish=myOnPublishCallback)

if not success:

print("Not connected to IoTF")

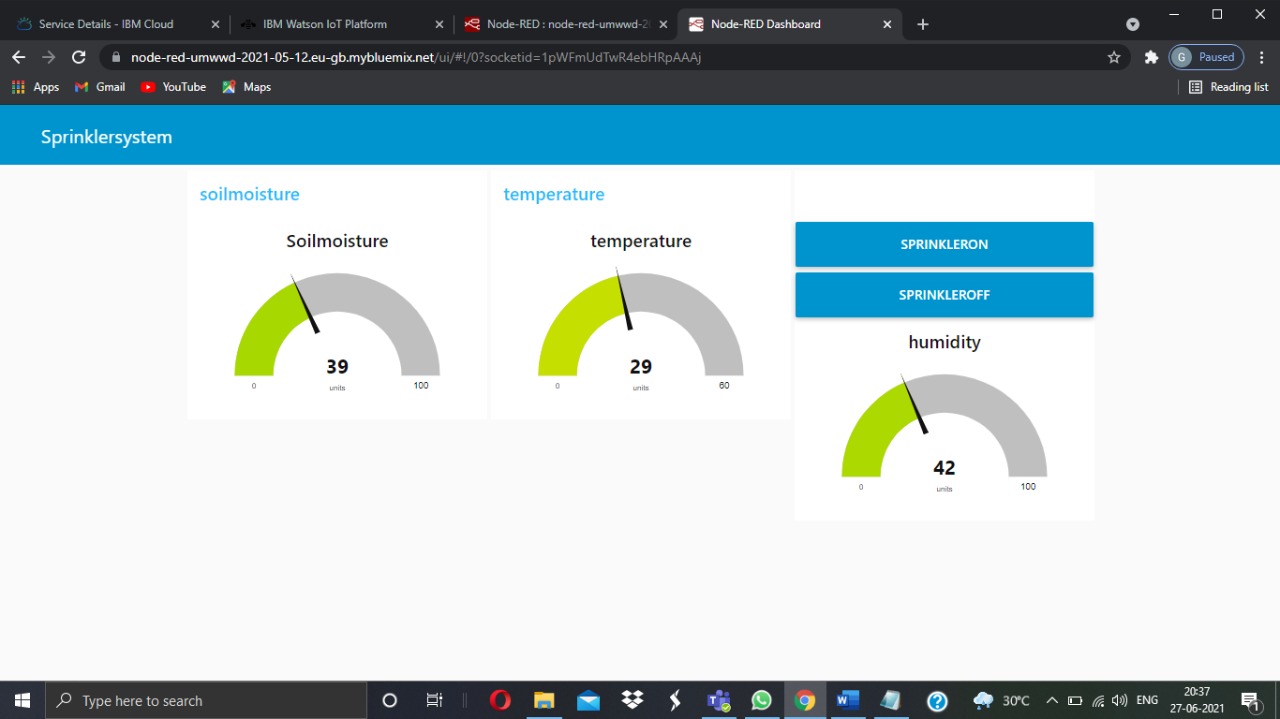
time.sleep(1)

deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud

deviceCli.disconnect()

* **UI OUTPUT SCREENSHOT**

****