

**IOT BASED SMART AGRICULTURE WITH IBM CLOUD**

**GROUP MEMBERS:**

## DEEPIKA NAGUDU (18481A05F1)

## KONAKALA VAMSI (18481A05C2)

**TABLE OF CONTENTS:**

**1 INTRODUCTION**

**1.1 Overview**

**1.2 Purpose**

**2** **LITERATURE SURVEY**

**2.1 Existing problem**

**2.2 Proposed solution**

**3**  **THEORITICAL ANALYSIS**

**3.1 block diagram**

**3.2 Hardware/software designing**

**4** **EXPERIMENTAL INVSTIGATION**

**5** **RESULT**

**6** **ADVANTAGES AND DISADVANTAGES**

**7** **CONCLUSION**

**8** **SOURCE CODE**







Agriculture is the backbone for a country’s development. Basically, agriculture depends on the monsoons which do not have enough water sources. To overcome this problem, the irrigation system is employed in the field of agriculture. In this system, based on the soil type, water will be provided to the field. In agriculture, there are two things, namely, the moisture content of the soil as well as the fertility of the soil. At the present time, there are several types of techniques available for irrigation to reduce the need for rain. This type of technique is driven by on/off schedule using electrical power. This project is about implementation of a Smart Irrigation System using IoT that can serve as a helping hand to farmer.



This project is used for the proper monitoring if water supply to the crops at times of altering weather conditions. So as to provide the required amount of water to the crop based on temperature, humidity and the moisture level of the soil.







The major operating parameters which influence the Smart Irrigation System performance are:

* 1. Inadequate trenching
  2. Lack of head-to-head design
  3. Incorrect pipe sizes used
  4. Poor quality cabling

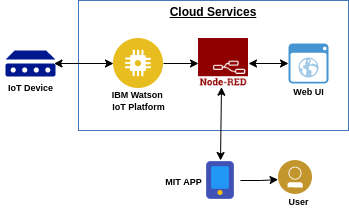


This Project examines and compares some IOT regression methods.

* + - Wireless sensor network makes the irrigation smarter and easier. The entire field is monitored continuously through the sensor network.
    - The soil moisture levels are transferred from routers to the main coordinator.
    - At the main controller the temperature is monitored, based on the temperature and soil moisture content the motor is made to ON/OFF.
    - Temperature, Humidity and Soil Moisture is visualized in the mobile app and also stored in the database.
    - Motor can be turned ON/OFF using the mobile app.







**Fig: Block diagram of smart agriculture system using IOT**





By using

* Python
* IBM iot
* Mit app
* Node-red

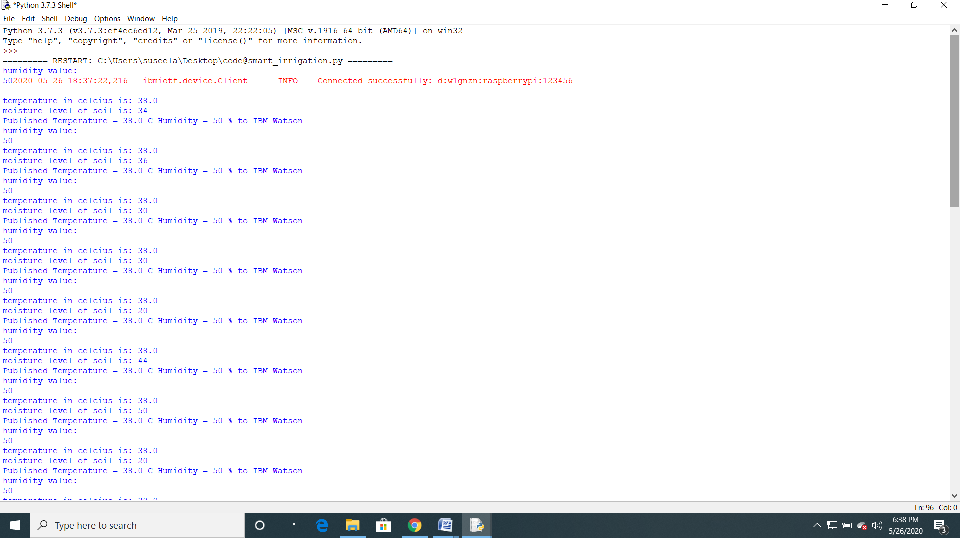




The term “IOT” stands for the Internet of Things and it can be defined as the interconnection between the individually identifiable embedded computing apparatus in the accessible internet infrastructure. ‘IOT’ connects various devices and transportations with the help of internet as well as electronic sensors. Smart Irrigation System is an IOT based device which is capable of automating the irrigation process by analyzing the moisture of soil and the climate condition (like High Temperature, Humidity).Also the data of sensors will be displayed on the developed Mobile application.

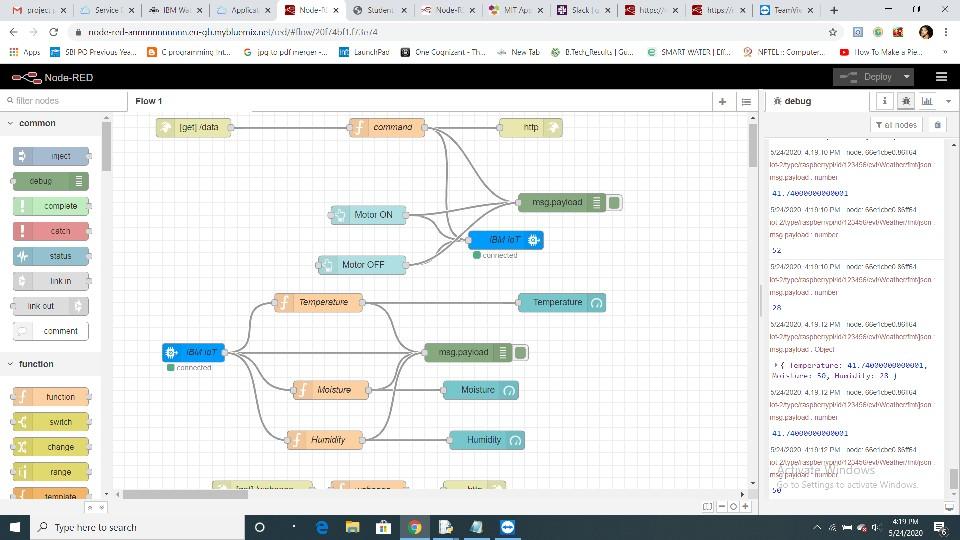
## RESULT:

**Python Code:**

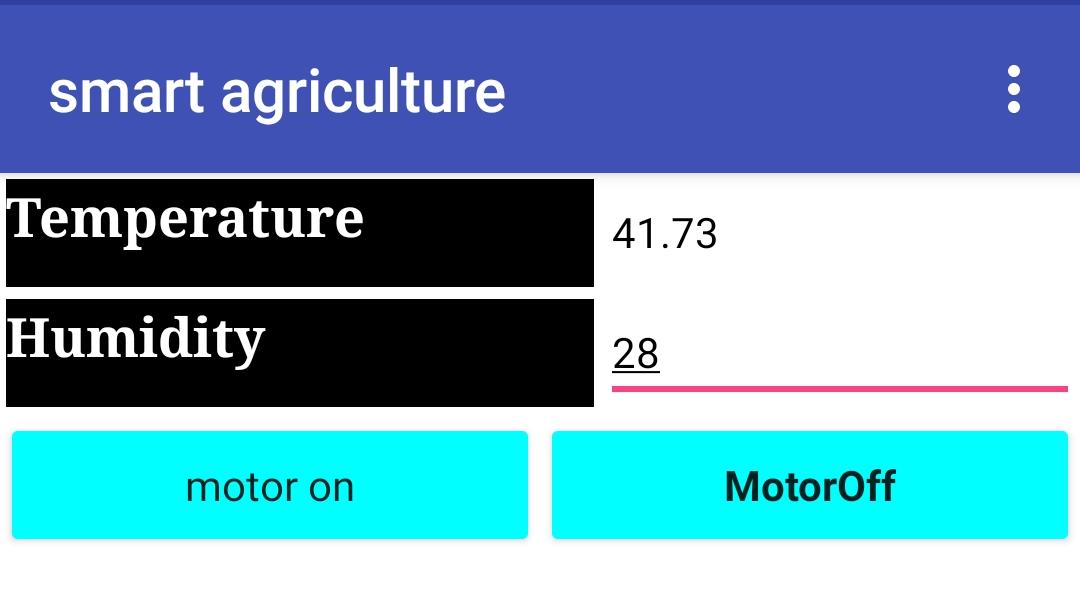








MITAPP:







* + Ability to save water
  + precision watering
  + Irrigating crops as per schedule
  + Decreasing water overflows
  + Ascertaining precise soil dampness levels



* + There could be wrong Analysis of Weather Conditions
  + Indian farmers are not used to these high-end technologies
  + Difficult in case of failure og GSM modem





* + It can be implemented in the modern irrigation system
  + It may also be used in rural areas
  + May be implemented by small agriculturists



A system to monitor moisture levels in the soil was designed and the project provided an opportunity to study the existing systems, along with their features and drawbacks. The proposed system can be used to switch on/off the water sprinkler according to soil moisture levels thereby automating the process of irrigation which is one of the most time consuming activities in farming.

Agriculture is one of the most water-consuming activities. The system uses information from soil moisture sensors to irrigate soil which helps to prevent over irrigation or under irrigation of soil thereby avoiding crop damage. The farm owner can monitor the process online through a Mobile application. Through this project it can be concluded that there can be considerable development in farming with the use of IOT and automation. Thus, the system is a potential solution to the problems faced in the existing manual and cumbersome process of irrigation by enabling efficient utilization of water resources.



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 i.e., he can switch on/off the pump in order to start/stop the process of irrigation without his physical presence at the farm. The farmer may choose to stop the flow of water 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 for irrigation can be extended further to other activities in farming such as cattle management, fire detection and climate control. This would minimize human intervention in farming activities. .





import time

import sys

import ibmiotf.application

import ibmiotf.device

import random

import json

#Provide your IBM Watson Device Credentials

organization = "6nzsgw"

deviceType = "iotdevice"

deviceId = "1001"

authMethod = "token"

authToken = "1234567890"

# Initialize the device client.

T=0

H=0

S=0

def myCommandCallback(cmd):

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

if cmd.data['command']=='motor on':

print("MOTOR ON IS RECEIVED")

elif cmd.data['command']=='motor off':

print("MOTOR OFF IS RECEIVED")

if cmd.command == "setInterval":

if 'interval' not in cmd.data:

print("Error - command is missing required information: 'interval'")

else:

interval = cmd.data['interval']

elif cmd.command == "print":

if 'message' not in cmd.data:

print("Error - command is missing required information: 'message'")

else:

print(cmd.data['message'])

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:

T=random.randint(22,40)

H=random.randint(32,65)

S=random.randint(20,95)

#Send Temperature & Humidity to IBM Watson

data = {"d":{ 'temperature' : T, 'humidity': H,'soilmoisture': S }}

#print data

def myOnPublishCallback():

print ("Published Temperature = %s C" % T, "Humidity = %s %%" % H,"Soilmoisture = %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(10)

deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud

deviceCli.disconnect()

