Software:

Python Code:

```
Import time

Import sys

Import ibmiotf.application

Import ibmiotf.device

Import random

#Provide your IBM Watson Device Credentials

Organization = “kv09p4”

deviceType = “Groot”

deviceId = “13”

authMethod = “token”

authToken = “12345678”

global flag

flag=0

n=int(input(“Enter no of Field Divisions”))

# Initialize GPIO

Def myCommandCallback(cmd):

   Print(“Command received: %s” % cmd.data[‘command’])

   Status=cmd.data[‘command’]

   If status==”motoron”:

      Print (“motor is on”)

   If status==”motoroff” :

      Print (“motor is off”)

   #print(cmd)

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€)
```

```
        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:
    Sug=”Suggestion For Irrigation”
    #Get Sensor Data from DHT11
    Avgt=0
    Avgh=0
    Avgs=0
    Avgp=0
    For I in range(0,n):
      Temp=random.randint(0,100)
      Humid=random.randint(0,100)
      Soilmoisture=random.randint(0,1023)
      Phlevel=random.randint(0,14)
      Print(“T:”,temp)
      Print(“H:”,Humid)
      Print(“S:”,soilmoisture)
      Print(“P:”,Phlevel,”\n”)
      Avgt += temp
      Avgh += Humid
      Avgs += soilmoisture
      Avgp += Phlevel
    Temp = avgt/n
    Humid = avgh/n
    Soilmoisture = avgs/n
    Phlevel = avgp/n
    Data = { ‘temp’ : temp, ‘Humid’: Humid,’soilmoisture’ : soilmoisture ,’Phlevel’ : Phlevel}
    #print data
    Def myOnPublishCallback():
      Print (“Published Temperature = %s C” % temp, “Humidity = %s %%” % Humid,”Soil Moisture is %s %%” %
soilmoisture,”PH level is %s” %Phlevel ,”to IBM Watson”)
```

```
        Success = deviceCli.publishEvent(“IoTSensor”, “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()
```