SMART FARMER-IOT ENABLED SMART FARMING APPLICATION

Team ID : PNT2022TMID04781

Team Leader : Aravinth S (737819ITL109)

Team Members: Maheshsriram T (737819ITR045)

Merlin M (737819ITR046)

Mothis B (737819ITR048)

1. INTRODUCTION:

Agriculture is one of the largest income sources in India. Agriculture plays an important role in sustaining human life. Population growth is proportional to agricultural production growth. Fundamentally, agricultural production is subject to seasonal conditions such as scarcity of water resources. IoT-based smart farming systems are deployed to achieve beneficial results and overcome problems in agriculture.

Smart agriculture powered by IoT is highly efficient compared to traditional approaches. The proposed IoT-based irrigation system uses ESP8266 NodeMCU module and DHT11 sensor. In addition to automatically irrigating water based on soil moisture levels, it also sends data to Thing Speak servers to track soil conditions.

a. Project Overview:

IoT-based agricultural systems help farmers monitor various parameters of their fields, such as soil moisture, temperature, and humidity, using several sensors. A farmer can monitor all sensor her parameters through his web or mobile application without being near his field. Crop irrigation is his one of the most important tasks for a farmer. By monitoring sensor parameters and controlling motor pumps from the mobile application itself, crop irrigation or movement decisions can be made.

b. Purpose:

Parameters such as temperature, humidity and soil moisture are updated in the Watson IoT platform. The device subscribes to the mobile application's commands and controls the motors accordingly. APIs are developed using Node-RED services to communicate with mobile applications. With the help of MIT app inventors, a mobile application is being developed to monitor sensor parameters and control motors.

2. LITERATURE SURVEY:

An automated system explains how the grain fields are monitored. The system was developed using sensors and decisions made by the server based on collected data, irrigation. The system is automated by wireless transmission. The recorded data is transferred to the web server database.

If Irrigation is then automated humidity and temperature Field is lower than potential range. Users can able to remotely monitor and control your system. An application that provides a web interface to users. The system focuses on device and tool development by managing, viewing, and notifying users, Wireless sensor network system.

intended to make Smart agriculture with automation and IoT technology. cloud computing devices the system in which the entire computer system occurs from sensors to tools that observe data from agriculture. this Proposing a new methodology for smart agriculture Including intelligent sensor system and intelligent irrigation system Through wireless communication technology.

this Low installation cost of the system. You can access it here Control your farming system with your laptop, mobile phone or computer.

a. **Existing problem:**

Agriculture is the most important for the whole country. Agriculture is decreasing day by day. Indian agriculture faces challenges related to adapting to disruptions from climate change, fragmentation of land tenure and low agricultural productivity. In some cases, farmers have difficulty accessing arable land and keeping track of their crops. IoT system has the solution to help you. It has many sensors, control devices and networking techniques to transmit arable land information and maintain moisture without human intervention. This approach is aimed at domestic agricultural growth.

b. **References:**

“Application of smart irrigation systems for water conservation in Italian farms” - Stefano Casadei, Francesco Peppoloni, Flaminia Ventura, Razvan Teodorescu, Daniel Dunea, Nicolae Petrescu – 2021.

The irrigation advisory service has been calibrated and validated for pomegranate trees, aubergine, and zucchini through 2 years of tests, in a farm in Southern Italy (Calabria). The results are very satisfactory, both in terms of water resources management for irrigation, and for the feedback provided by the farmers involved in the project. Future development regarding the application of wireless technology in smart irrigation is also evaluate.

“Internet of things for smart farming” - Md Alimul Haque, Deepa Sonal, Shameemul Haque, Kailash Kumar – 2022.

This chapter provides the importance of wireless and IoT sensors in agriculture as well as the obstacles facing the integration of this technology with conventional agricultural practices. Detailed analysis of crop condition, fertilizer, insect detection and pesticide is carried out on IoT products and wireless sensor connectivity methods in farm applications. This chapter proposes an IoT-based model to alert farmers for soil moisture conditions, potential damage in agricultural land from the fire and automatic water irrigation started at low moisture of soil or fire. Finally, this chapter covers contemporary and future IoT trends in smart farming and research difficulties.

“Internet-of-Things (IOT)-Based Smart Agriculture” - MUHAMMAD AYAZ, MOHAMMAD AMMADUDDIN, ZUBAIR SHARIF, ALI MANSOUR, EL-HADI M. AGGOUNE – 2019.

Furthermore, the use of unmanned aerial vehicles for crop surveillance and other favourable applications such as optimizing crop yield is considered in this article. State-of-the-art IoT-based architectures and platforms used in agriculture are also highlighted wherever suitable. Finally, based on this thorough review, we identify current and future trends of IoT in agriculture and highlight potential research challenges.

“An Automated Irrigation System Using Arduino Microcontroller” – Aslinda Hassan, Nazrulazhar Bahaman, Wahidah Md Shah, Siah Bing Sheng – 2018.

When the soil moisture sensor senses the dry soil, it will show the moisture percentage on the LCD display, and the relay module will switch on the water pump automatically to start the watering process, or vice versa. Hardware testing is conducted to ensure the proposed system is fully functional.

IoT-Based Intelligent Irrigation System for Paddy Crop Using an Internet-Controlled Water Pump - Brijbhushan Sharma, Nagesh Kumar – 2021.

The condition of soil is monitored based on the parameter of soil-like moisture and water flow amount using the IoT, which is capable to turn on/off water pumps. The used dashboard is developed using open-source free server, namely “000webhost.” This paper has considered the paddy crop that is rice because water is essential for growth and development of rice plants. The experimental results show this system is more proficient than the existing conventional and unadventurous irrigation approach.

c. **Problem Statement Definition:**

I am a farmer I am trying to monitor my agricultural land 24x7. But I cannot stay in my place any time because my home is far away from my land. so, I want some solution that will help to monitor my land soil moisture and control motor.

3. IDEATION & PROPOSED SOLUTION:

a. **Empathy Map Canvas:**

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.

Fig 3.1 Empathy Map

b. **Ideation & Brainstorming:**

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions. Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you are not sitting in the same room.

Step-1: Team Gathering, Collaboration and Select the Problem Statemen

Fig 3.2 Empathy Map

Step-2: Brainstorm, Idea Listing and Grouping

Fig 3.3 Empathy Map

Step-3: Idea Prioritization

Fig 3.3 Empathy Map

c. **Proposed Solution:**

Project team shall fill the following information in proposed solution template.

Problem Statement:

I am a farmer I am trying to monitor my agricultural land 24x7. but, I can't stay in my place any time because my home is far away from my land. so, I want some solution that will help to monitor my land soil moisture and control motor.

Idea / Solution description:

An IoT system that has many sensors to gather information about agricultural land and then send that information to the mobile app. The mobile app provides an option to ON/OFF the motor also.

Novelty / Uniqueness:

Simple UI to understand the information and use the application very well.

Customer Satisfaction:

Both monitor the land and control the motor with a fingertip available 24x7. It saves more time and effort.

Business Model:

The system consists of a low amount of sensors and simple cloud functions. so, it is financially better.

Scalability of the Solution:

It is a simple solution compared with other technologies.

d. **Problem Solution fit:**

Customer segment(s)

Farmers or Agriculture related workers.

Jobs-to-be-done / problems

They must know about moisture and temperature for the crops and correct way to handle the system.

Triggers

When they feel about their crops and agricultural land and when they use this system to monitor the soil's moisture.

Emotions: before / after

Before,

Insecure > confident

Think like "waste of money"

After,

very nice

easy for using

Available solutions

Formers wants to monitor their crops and control their irrigation as much as in easy way. In past they must to visit that place then gather the information about their agricultural land. It provides a 24x7 monitoring for agricultural land. Sometimes any error will occur it make high cost for solve.

CUSTOMER CONSTRAINTS

spending power, budget, network connection.

BEHAVIOUR

Users monitor their land's moisture and temperature.

CHANNELS of BEHAVIOUR

ONLINE,

Monitoring soil moisture and temperature

Control the motor (ON/OFF)

OFFLINE,

Without the internet, users cannot do anything here because wireless devices need the internet to do tasks like transmitting/receiving signals

PROBLEM ROOT CAUSE

Most of the users have their agricultural land too far away from their home. Sometimes they cannot travel to that place then they will suffer. Sometimes they forget to irrigate to their crops by some reasons.

OUR SOLUTION

This is the current solution to monitor moisture, temperature, and humidity when they want. Then it can control the motor. The interface and design are well and good. finally, it was a simple easy solution. This system includes a mobile application that will show information about soil like temperature and moisture and motor control options also.

4. REQUIREMENT ANALYSIS:

a. Functional requirement:

User Monitoring

i. Monitoring Temperature

ii. Monitoring Humidity

iii. Monitoring Moisture

iv. Monitoring Motor state

User select mode

1. Confirmation system mode
2. Auto/Manual

User Controls

1. Change the motor state ON/OFF

b. **Non- Functional requirement:**

Usability

Simple and easy to use, 24x7 access through Web/Mobile Application

Security

cloud integrations

Reliability

IBM Watson IoT platform has session timeout, Username & Password authentication

Performance

Lightweight Application, 200 requests/second

Availability

24x7 in IBM Cloud server

Scalability

2 – tier, Micro-services

5. PROJECT DESIGN:

a. Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

Fig 5.1 dataflow diagram

b. Solution & Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table-1.

Fig 5.1 Technical Architecture

Table-1: Components & Technologies

S. No	Component	Description	Technology
1.	User Interface	Web UI, Mobile App	Node-red, MIT app inventor
2.	Application Logic-1	Sensor data that is randomly generated	Python
3.	Application Logic-2	Sensor data is published and command signals are subscribed	IBM Watson IoT service
4.	Application Logic-3	Connect the cloud data to the web and Mobile Application	Node red
5.	External API-1	To generate the weather data if required	Open Weather API
6.	ExternalAPI-2	To send an SMS to mobile devices	Fast2SMS API
7.	Infrastructure (Server / Cloud)	Local Server Configuration (Mobile device)	MIT Mobile Application

c. **User Stories:**

2. As a user I can see the Temperature, Humidity and Moisture level.

3. As a user I can able to control the motor.

4. As a user I can able to change control mode auto/manual.

6. PROJECT PLANNING & SCHEDULING:

a. Sprint Planning & Estimation:

Below table-2 shows the Sprint planning and Estimation of this project.

Table-2: Sprint Planning & Estimation

Sprint No	Tasks	Priority
Sprint 1	Find the temperature, humidity, and moisture level from environment	Medium
Sprint 2	Provide sensor data to IBM cloud then process and perform corresponding actions	High
Sprint 3	Create an end user application (web, mobile) and configure it	Low
Sprint 4	Create an option for select the mode of system (auto/manual) for user convenient.	High

b. Sprint Delivery Schedule:

Sprint No	Priority	Estimation	Completion
Sprint 1	Medium	3 days	2 days
Sprint 2	High	4 days	5 days
Sprint 3	Low	2 days	2 days
Sprint 4	High	4 days	6 days

c. Reports from JIRA:

7. CODING & SOLUTIONING:

a. **Feature 1:**

Toget a Temperature, Humidity, soil moisture from various kind of sensors (DHT11, Soil moisture sensor). Then processing it and push those data into IBM cloud (IBM Watson IOT platform).

Getdata.py:

```
import wiotp.sdk.device
import time
import os
import datetime
import random
myconfig = {
   "identity": {
      "orgId": "z69c1z",
      "typeId": "w_dev",
      "deviceId": "w_dev1"
   },
   "auth": {
      "token": "SJAYik-EXtra_TXPDQ"
   }
}
client=wiotp.sdk.device.DeviceClient(config=myconfig,logHandlersNon
e)
client.connect()
ms="OFF"
def myCommandCallback(cmd):
      print("Msg received from IBM IOT platform:%s" %
cmd.data['command'])
   global ms
   m = cmd.data['command']
   if (m == "motoron"):
      print("Motor is switched ON")
      ms="ON"
   elif (m == "motoroff"):
      print("motor is switched OFF")
      ms="OFF"
   print(" ")
while True:
```

```
            soil = random.randint(0, 100)
            temp = random.randint(5, 150)
            hum = random.randint(0, 100)
            myData = {'sensor_moisture': soil,
                  'sensor_temperature': temp,
                  'sensor_humidity': hum,
                  'sensor_mstate': ms
                  }
            client.publishEvent(eventId="status", msgFormat="json", data=myData,
qos=0, onPublish=None)
            print("Published data successfully: %s", myData)
            time.sleep(2)
            client.commandCallback = myCommandCallback
        client.disconnect()
```

b. **Feature 2:**

Processing the real time sensor data and act respective actions like send that information to the user through web or mobile application with the help of Node-red service. (shown in Fig 7.1)

Fig 7.1 Get information with node-red service

8. TESTING:

a. **Test Cases:**

- To get the Temperature, Humidity and Moisture level

Fig 8.1 getting data from sensors

- Send that information to the IBM cloud

Fig 8.2 sending information to the IBM cloud

b. **User Acceptance Testing:**

- To check the real time Temperature, Humidity and Soil Moisture

Fig 8.3 web App **Fig 8.4 Mobile App**

- Check user able to switch between two modes

Fig 8.5 Auto/Manual mode

- Control the motor (ON/OFF)

Fig 8.6 motor (On/Off)

9. RESULTS:

a. **Performance Metrics:**

10. ADVANTAGES & DISADVANTAGES:

a. **Advantages:**

- The remote-control system helps to operate the valves in the irrigation system according to the schedule. Irrigation of remote farm properties can be particularly cumbersome and laborious. It is difficult to understand when the valves are started and whether the ideal amount of water has been delivered.

- Submersible gravity sensors or ultrasonic sensors can monitor the build-up of tanks, lakes, wells, and various liquids such as fuel and compost. The product calculates the volume based on the shape of the reservoir or lake after a certain time. It transmits alarms based on different conditions.

b. **Disadvantages:**

- A major downside of smart farming is that it requires an unlimited or constant internet connection to succeed. This means that in rural communities, especially in developing countries where we have mass agricultural production, it is not possible to exploit this farming method at all.
- In places where internet connection is extremely slow, smart farming will be impossible. As stated earlier, smart farming uses advanced technologies that require technical skill and precision to succeed.

11. CONCLUSION:

Farmers can greatly benefit from an IoT-based smart farming system. Agriculture is affected by lack of irrigation water. Climatic factors such as humidity, temperature and humidity can be adjusted for local environmental variables. The technology also detects animal encroachment, a major cause of crop failure. This technology makes it easy to schedule irrigation based on field data and records from climate sources. This helps the farmer decide whether to irrigate or not. A persistent internet connection is required for continuous monitoring of sensor data. This can also be overcome by using an alternative GSM device for the mobile app. Through GSM, text messages can be sent to farmers' phones.

12. FEATURE SCOPE:

- In the current project, we have implemented a project that can protect and maintain crops. In this project, the farmer monitors and controls the field remotely. In the future we may add or update some more things to this project
- We can create several models of the same project so that farmers have full information.
- We can update this project using the solar mechanism. So, the power supply of the electric poles can be replaced by solar panels. It reduces the cost of power lines. It will be a one-time investment. We can add solar fence technology to this project.
- We can use GSM technology for this project so farmers can receive information directly

to their homes by SMS. This helps farmers get information if there is a problem with the internet.

- We can add camera functionality so farmers can monitor their fields in real time. This helps prevent theft.

13. APPENDIX:

Source code:

```
# import required lib......
import wiotp.sdk.device
import time
import os
import datetime
import random

myconfig = {
    "identity": {
        "orgId": "z69c1z",
        "typeId": "w_dev",
        "deviceId": "w_dev1"
    },
    "auth": {
        "token": "SJAYik-EXtra_TXPDQ"
    }
}

#Connect IBM IOT cloud...
client = wiotp.sdk.device.DeviceClient(config=myconfig, logHandlers=None)
client.connect()

#Declaring values
ms="OFF"
appmode="AUTO"

#Get responding data from IBM cloud...
```

```
def myCommandCallback(cmd):
    print("Msg received from IBM IOT platform:%s" % cmd.data['command'])
    global ms
    global appmode

    #Get motor & Application state...
    m = cmd.data['command']

    if (m == "motoron"):
        print("Motor is switched ON")
        ms="ON"
    elif (m == "motoroff"):
        print("motor is switched OFF")
        ms="OFF"
    elif(m == "auto"):
        print("Application is changed to Automatic")
        appmode="AUTO"
    elif (m == "manual"):
        print("Application is changed to Manual")
        appmode="MANUAL"
    print(" ")

#Start the Process...
while True:
    soil = random.randint(0, 100)
    temp = random.randint(5, 150)
    hum = random.randint(0, 100)

    #Set all sensor and motor,Application State...
    myData = {'sensor_moisture': soil,
              'sensor_temperature': temp,
              'sensor_humidity': hum,
              'sensor_mstate': ms,
              'sensor_amode': appmode
              }

    #Push the Data into IBM Cloud...
    client.publishEvent(eventId="status",msgFormat="json",data=myData,qos=0,
onPublish=None)
    print("Published data successfully: %s", myData)
    #Check the any response from IBM Cloud...
```

```
    time.sleep(2)
    client.commandCallback = myCommandCallback
    #In Auto mode change motor state correspondingly...
    if(appmode == "AUTO"):
        if(soil < 40):
            print("Motor is switched ON")
            ms="ON"
        else:
            print("motor is switched OFF")
            ms="OFF"
        print(" ")
#close the connection...
client.disconnect()
```

GitHub:

https://github.com/IBM-EPBL/IBM-Project-289661660119508.git

Project demo:

https://drive.google.com/file/d/1gGMFKaM5jN_EBieeEvVShKdsOUPJLsAl/view?usp=sharing