IOT ENABLED SMART FARMING APPLICATION.

Sprint Delivery – 1

TEAMID : PNT2022TMID41836

Introduction

The main aim of this project is to help farmers automate their farms by providing them with a Web site through which they can monitor the parameters of the field like Temperature, soil moisture, humidity and etc and control the equipment like water motor and other devices remotely via internet without their actual presence in the field.

Problem Statement

A well-articulated customer problem statement allows you and your team to find the ideal solution for the challenges your customers face. Throughout the process, you'll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.

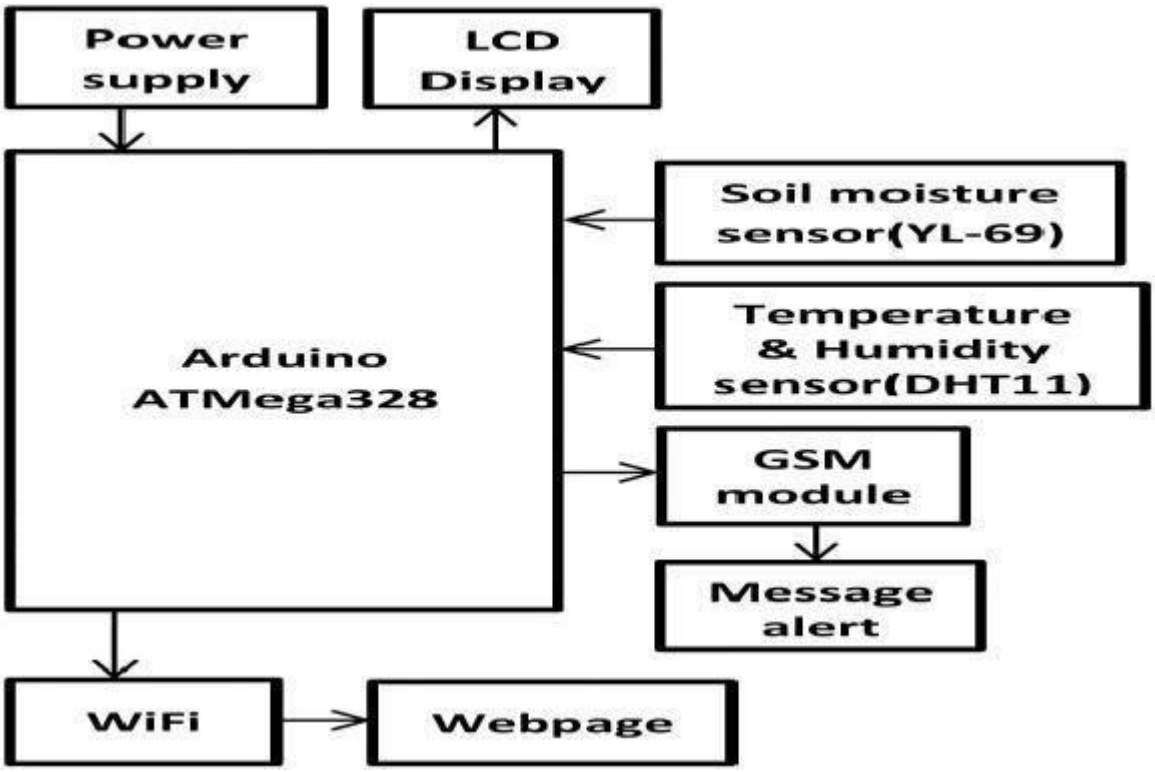
Proposed Solution

In order to improve the farmer's working conditions and make them easier, we introduce IoT services to him in which we use cloud services and internet to enable farmer to continue his work remotely via internet. He can monitor the field parameters and control the devices in farm.

Theoretical Analysis

Block Diagram

In order to implement the solution , the following approach as shown in the block diagram is used

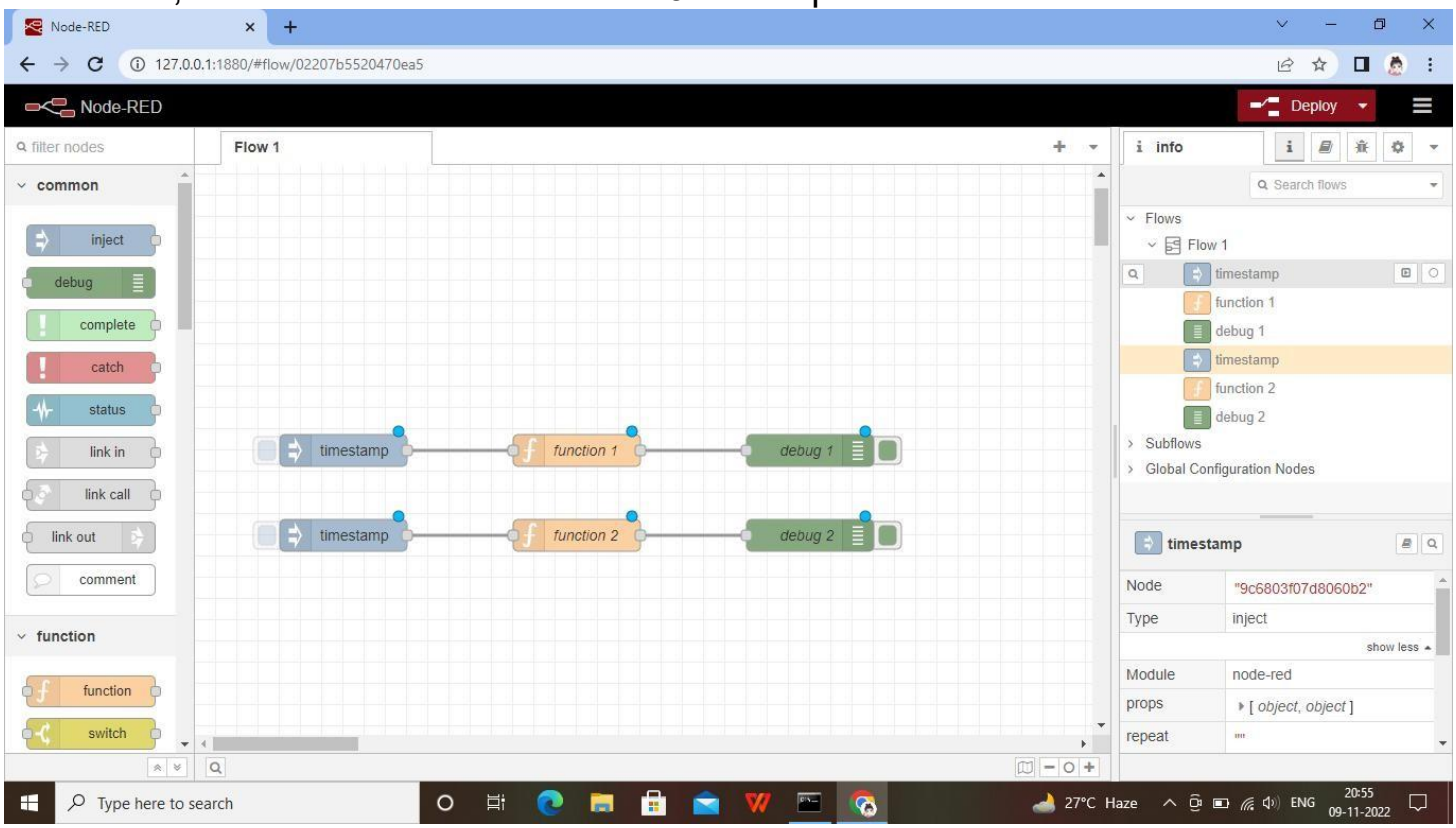


Required Software Installation

Node-Red

Node-RED is a flow-based development tool for visual programming developed originally by IBM for wiring together hardware devices, APIs and online services as

part of the Internet of Things. Node-RED provides a web browser-based flow editor, which can be used to create JavaScript functions.



Installation :

- First install npm/node.js
- Open cmd prompt
- Type => npm install node-red

To run the application :

- Open cmd prompt
- Type=>node-red
- Then open http://localhost:1880/ in browser

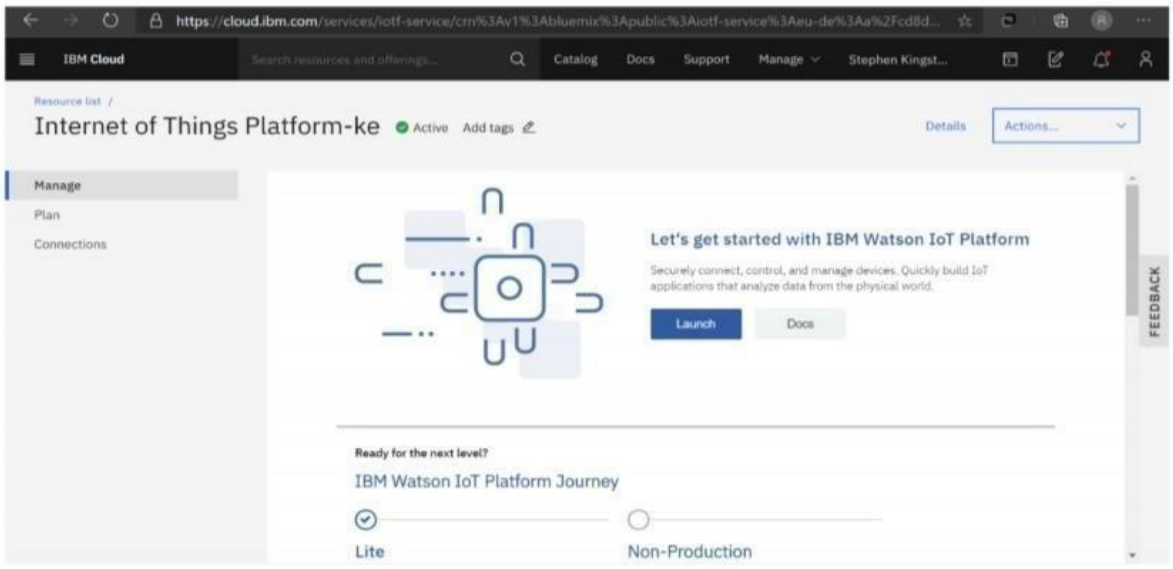
Installation of IBM IoT and Dashboard nodes for Node-Red

In order to connect to IBM Watson IoT platform and create the Web App UI these nodes are required 1. IBM IoT node

2. Dashboard node

IBM Watson IoT Platform

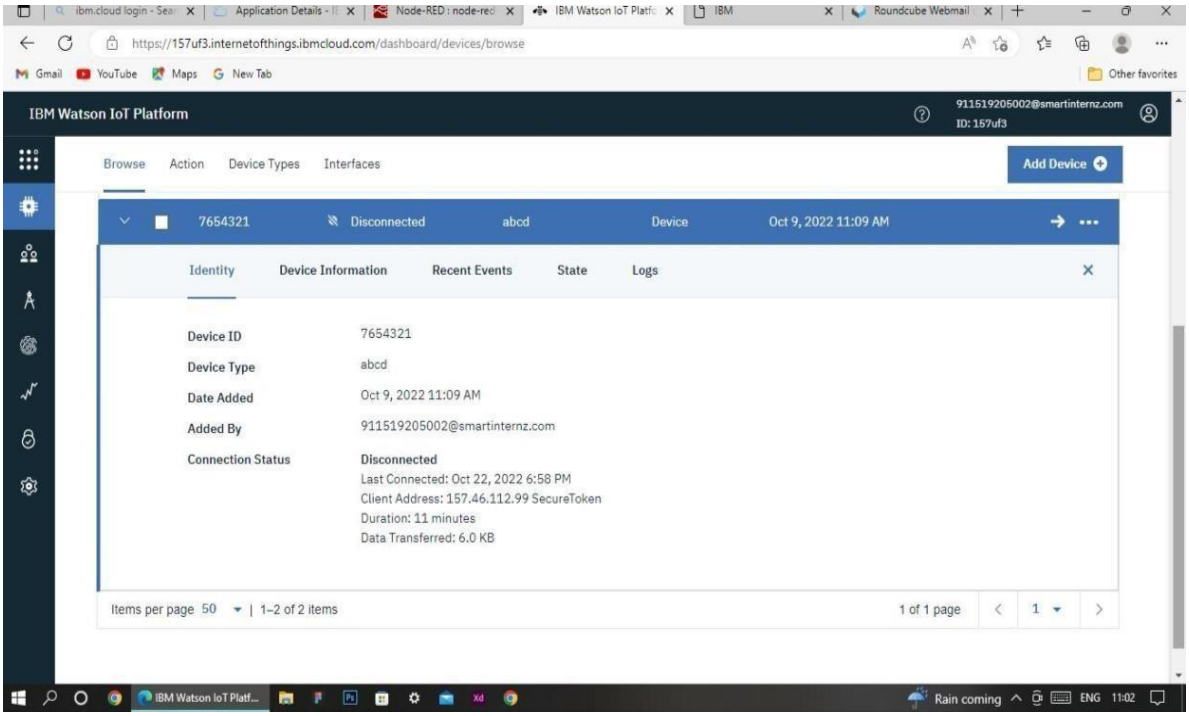
A fully managed, cloud-hosted service with capabilities for device registration, connectivity, control, rapid visualization and data storage. IBM Watson IoT Platform is a managed, cloud-hosted service designed to make it simple to derive value from your IoT devices.



Steps to configure:

- Create an account in IBM cloud using your email ID
- Create IBM Watson Platform in services in your IBM cloud account
- Launch the IBM Watson IoT Platform
- Create a new device
- Give credentials like device type, device ID, Auth. Token

- Create API key and store API key and token elsewhere.



Python IDE

Install Python3 compiler

Install any python IDE to execute python scripts, in my case I used Spyder to execute the code.

IoT Simulator

In our project in the place of sensors we are going to use IoT sensor simulator which give random readings to the connected cloud.

The link to simulator:

https://watson-iot-sensor-simulator.mybluemix.net/

We need to give the credentials of the created device in IBM Watson IoT Platform to connect cloud to simulator.

CODING:

```
{
    margin:0
    padding: 0;
    box-sizing: border-box;
}
.full-page
{
    height: 100%;
    width: 100%;
    background-image: linear-gradient(rgba(0,0,0,0.4),rgba(0,0,0,0.4)),url(images/bg-2. jpg);
    background-position: center;
    background-size: cover;
    position: absolute;
    /*   this block of code is for the setting the background image and position of image
etc...*/
}
.navbar
{
    display: flex;
    align-items: center;
    padding:20 px;
    padding-left: 50 px;
    padding-right: 30 px;
    padding-top: 50 px;
    /*   this if for setting the dimensions of the navigation bar*/
}
nav
{
    flex: 1;
    text-align: right;
}
nav ul
{
    display: inline-block;
```

```
    list-style: none;
}
nav ul li
{
    display: inline-block;
    margin-right: 70 px;
    /*this block are for styling the list element*/
}
nav ul li a
{
    text-decoration: none;
    font-size: 20 px;
    color: white;
    font-family: sans-serif;
}
nav ul li button
{
    font-size: 20 px;
    color: white;
    outline: none;
    border: none;
    background: transparent;
    cursor: pointer;
    font-family: sans-serif;
}
nav ul li button:hover
{
    color: aqua;
}
nav ul li a:hover
{
    color: aqua;
}
a
{
    text-decoration: none;
    color: palevioletred;
    font-size: 28 px;
}
#login-form
{
    display: none;
    /*         this block if for hiding the form     */
}
.form-box
{
    width:380 px;
    height:480 px;
    position:relative;
```

```
    margin:2% auto;
    background:rgba(0,0,0,0.3);
    padding:10 px;
    overflow: hidden;
    /*     creating the form with width height and etc   */
}
.button-box
{
    width:220 px;
    margin:35 px auto;
    position:relative;
    box-shadow: 0 0 20 px 9 px #ff61241f;
    border-radius: 30 px;
}
.toggle-btn
{
    padding:10 px 30 px;
    cursor:pointer;
    background:transparent;
    border:0;
    outline: none;
    position: relative;
}
#btn
{
    top: 0;
    left:0;
    position: absolute;
    width: 110 px;
    height: 100%;
    background: #F3C693;
    border-radius: 30 px;
    transition: .5 s;
}
.input-group-login
{
    top: 150 px;
    position:absolute;
    width:280 px;
    transition:.5 s;
}
.input-group-register
{
    top: 120 px;
    position:absolute;
    width:280 px;
    transition:.5 s;
}
.input-field
```

```
{
    width: 100%;
    padding:10 px 0;
    margin:5 px 0;
    border-left:0;
    border-top:0;
    border-right:0;
    border-bottom: 1 px solid #999;
    outline:none;
    background: transparent;
}
.submit-btn
{
    width: 85%;
    padding: 10 px 30 px;
    cursor: pointer;
    display: block;
    margin: auto;
    background: #F3C693;
    border: 0;
    outline: none;
    border-radius: 30 px;
}
.check-box
{*/
    margin: 30 px 10 px 34 px 0;
}
span
{
    color:#777;
    font-size:12 px;
    bottom:68 px;
    position:absolute;
}
#login
{
    left:50 px;
}
#login input
{
    color:white;
    font-size:15;
}
#register
{
    left:450 px;
}
#register input
{
```

```
    color:white;
    font-size: 15;
}
```