

CODING AND SOLUTIONING

DATE	18 NOVEMBER 2022
TEAM ID	PNT2022TMID44357
PROJECT NAME	IoT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE
PROJECT DOMAIN	INTERNET OF THINGS

7.1: FEATURE 1

A Web Application is built which consists of,

- Graphical representation of Humidity, Temperature and Soil Moisture
- Motor ON and Motor OFF

Step 1: Generate random values of Humidity, Temperature, Soil Moisture are generated from events in the Watson IOT platform. These sensor values are generated using random functions from the events that is used in the device which was created.

PAYLOADS	SENSORS
Temp	Temperature
Hum	Humidity
Moist	Moisture

The screenshot displays the IBM Watson IoT Platform interface. The main view is the 'Browse Devices' page, which shows a table of devices. A modal window is open for creating a new event type for a NodeMCU device. The modal includes fields for 'Event type name' (set to 'event_1'), 'Schedule' (set to 'Every Minute'), and 'Payload'. The payload is a JSON object with four keys: 'randomnumber', 'temp', 'hum', and 'mois', each assigned a random value function. The 'Send' button is visible in the modal.

Device ID	Status	Device Type	Class ID	Date Added
3527	Connected	NodeMCU	Device	Nov 18, 2022

```
{  "randomnumber": random(0, 100),  "temp": random(90, 100),  "hum": random(60, 100),  "mois": random(30, 100)}
```

Step 2: The values are generated for every minute as payload from events in the form of **json** format in the recent events of the device created in Watson Platform

The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. The main content area shows the 'Recent Events' tab for a device named '3527'. The table below lists the recent events:

Event	Value	Format	Last Received
event_1	{"randomNumber":68,"temp":97,"hum":75,"mol..."}	json	a few seconds ago
event_1	{"randomNumber":31,"temp":94,"hum":82,"mol..."}	json	a few seconds ago
event_1	{"randomNumber":19,"temp":99,"hum":80,"mol..."}	json	a few seconds ago
event_1	{"randomNumber":16,"temp":98,"hum":76,"mol..."}	json	a few seconds ago

At the bottom of the interface, a status message indicates '1 Simulation running'.

Step 3: Node-RED is an editor used to create the flow between the nodes and has to be deployed once the flow has been made. Once deployment is done sensor values can be viewed in detail

The screenshot displays the Node-RED web interface in a browser window. The browser's address bar shows the URL: `node-red-oghgr-2022-11-09.au-svc.mybluemix.net/red/#flow/152a0d5086d25c4a`. The interface is divided into several sections:

- Left Panel (Palette):** Contains two tabs: "common" and "function". The "common" tab is active, showing nodes like "inject", "debug", "complete", "catch", "status", "link in", "link call", "link out", and "comment". The "function" tab is also visible, showing nodes like "function", "switch", "change", and "range".
- Canvas:** The central workspace where the flow is built. It features a grid background. The flow titled "Smart Crop Protection" includes:
 - An "IBM IoT" node (blue) with a "connected" status.
 - A "timestamp" node (blue) connected to an "http request" node (yellow).
 - A "msg.payload" node (green) connected to the "http request" node.
 - Three "function" nodes (orange) labeled "hum", "mois", and "temp", each connected to the "IBM IoT" node.
 - Three "FIELD_" nodes (blue) labeled "FIELD_HUMID", "FIELD_SOIL MOIS", and "FIELD_TEMP", each connected to the "function" nodes.
 - A "Motor ON" node (blue) and a "Motor OFF" node (blue) connected to an "IBM IoT" node (blue) with a "connected" status.
- Right Panel (Debug Console):** Shows a list of log messages. The messages are:
 - 11/19/2022, 12:01:03 AM node: 111451c08972b058
iot-2-type/NodeMCUID/3527/event/event_1/fm/json :
msg.payload: number
 - 84
 - 11/19/2022, 12:01:04 AM node: 111451c08972b058
iot-2-type/NodeMCUID/3527/event/event_1/fm/json :
msg.payload: number
 - 97
 - 11/19/2022, 12:01:59 AM node: 111451c08972b058
iot-2-type/NodeMCUID/3527/event/event_1/fm/json :
msg.payload: Object
 - { randomNumber: 83, temp: 100, hum: 93, mois: 50 }
 - 11/19/2022, 12:02:00 AM node: 111451c08972b058
iot-2-type/NodeMCUID/3527/event/event_1/fm/json :
msg.payload: number
 - 100
 - 11/19/2022, 12:02:01 AM node: 111451c08972b058
iot-2-type/NodeMCUID/3527/event/event_1/fm/json :
msg.payload: number
 - 50
 - 11/19/2022, 12:02:02 AM node: 111451c08972b058
iot-2-type/NodeMCUID/3527/event/event_1/fm/json :
msg.payload: number
 - 93

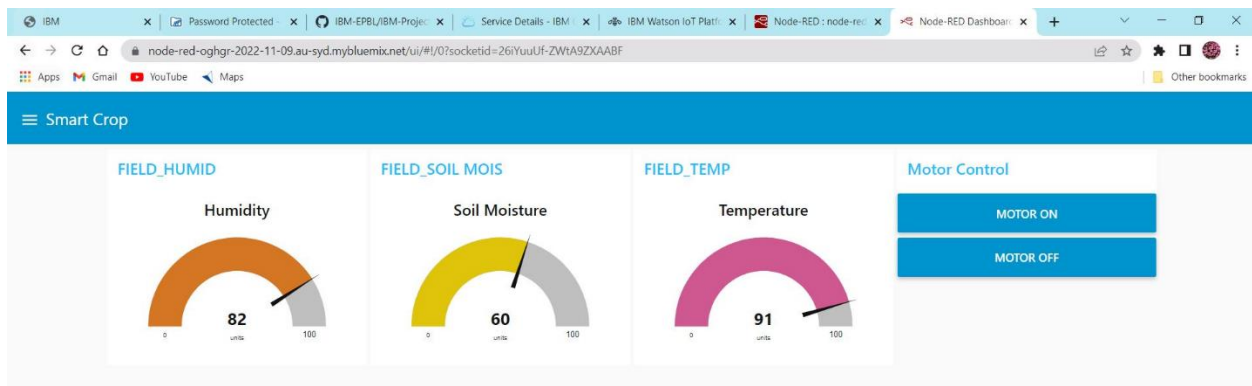
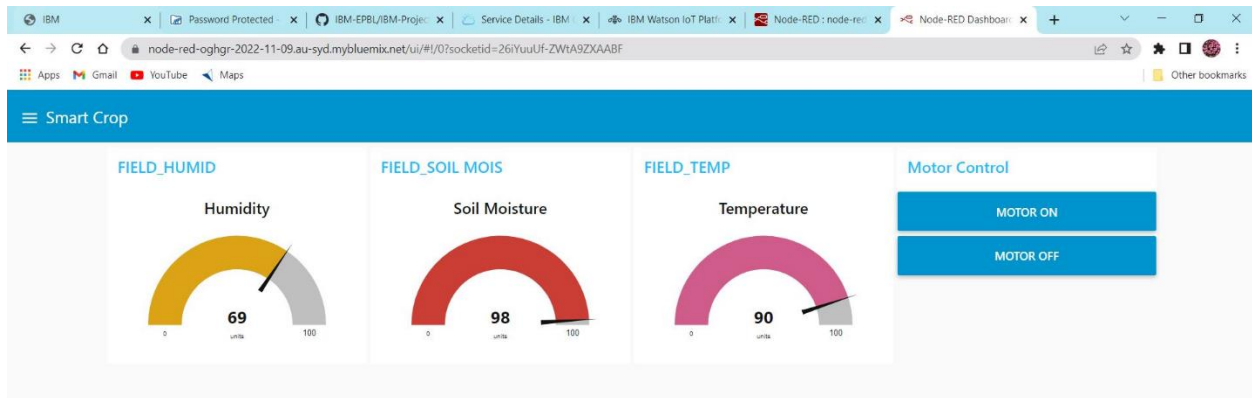
Step 4: The **Smart Crop** dashboard is viewed once the deployment is completed where we can able to view,

1. Moisture in the form of gauge
2. Temperature and Humidity in the form of gauge
3. MOTOR ON and MOTOR OFF buttons

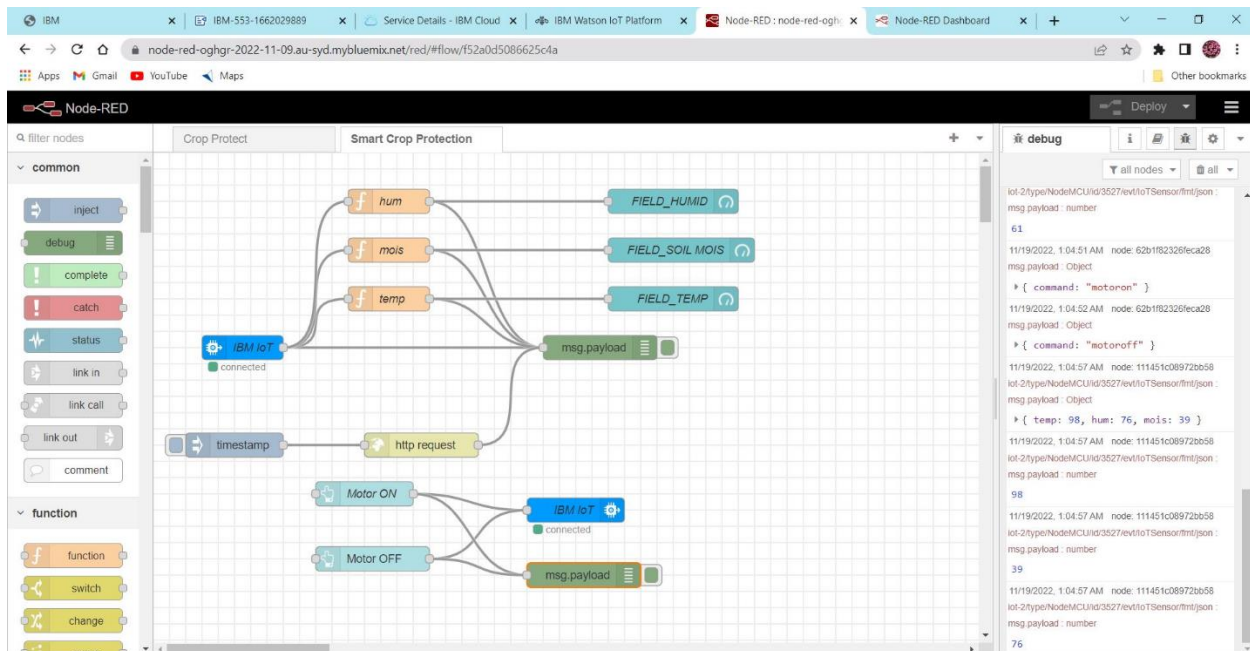
Details:

1. Dashboard is named as Smart Crop
2. Section is named as Field

In the section of Field, the sensors values are represented and motor control buttons are also given.



Step 5: When the **Motor ON** button is clicked the we receive the output as **“motoron”** and **Motor OFF** button is clicked we receive the output as **“motoroff”**. And these outputs are received in the debug section of the editor



Step 6: The output is also received in the **python code editor** when the buttons are clicked in the dashboard and random values are also generated. Device id is used to connect to IBM Watson.

```
Python 3.8.4 Shell*
File Edit Shell Debug Options Window Help
Python 3.8.4 (tags/v3.8.4:dfa645a, Jul 13 2020, 16:46:45) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\swast\AppData\Local\Programs\Python\Python38\iot.py =====
2022-11-18 19:29:49,790 ibmiotf.device.Client INFO Connected successfully: d:f41515:NodeMCU:3527
Published Temperature = 92 C Humidity = 66 % moisture= 99 % to IBM Watson
Published Temperature = 94 C Humidity = 97 % moisture= 32 % to IBM Watson
Command received: motoron
motor is on
Published Temperature = 95 C Humidity = 63 % moisture= 99 % to IBM Watson
Published Temperature = 93 C Humidity = 70 % moisture= 53 % to IBM Watson
Command received: motoroff
motor is off
Published Temperature = 97 C Humidity = 71 % moisture= 56 % to IBM Watson
Published Temperature = 96 C Humidity = 93 % moisture= 47 % to IBM Watson
```