

SPRINT 1

Date	29 October 2022
Team ID	PNT2022TMID31166
Project Name	Project – Smart Farmer-IoT Enabled smart Farming Application

Connecting Sensors with Arduino using C++ code

```
#include "Arduino.h"

#include "dht.h"

#include "SoilMoisture.h"

#define dht_apin A0

const int sensor_pin = A1; //soil moisture

int pin_out = 9;

dht DHT;

int c=0;

void setup()
{
  pinMode(2, INPUT); //Pin 2 as INPUT
  pinMode(3, OUTPUT); //PIN 3 as OUTPUT
  pinMode(9, OUTPUT); //output for pump
}

void loop()
{
  if (digitalRead(2) == HIGH)
  {
```

```

digitalWrite(3, HIGH); // turn the LED/Buzz ON
delay(10000); // wait for 100 msecond
digitalWrite(3, LOW); // turn the LED/Buzz OFF
delay(100);
}
Serial.begin(9600);
    delay(1000);
    DHT.read11(dht_apin); //temprature
float h=DHT.humidity;
float t=DHT.temperature;
    delay(5000);
    Serial.begin(9600);
    float moisture_percentage;
int sensor_analog;
sensor_analog = analogRead(sensor_pin);
moisture_percentage = ( 100 - ( (sensor_analog/1023.00) *
100 ) );
float m=moisture_percentage;
delay(1000);
if(m<40)//pump
{
while(m<40)
{

```

```

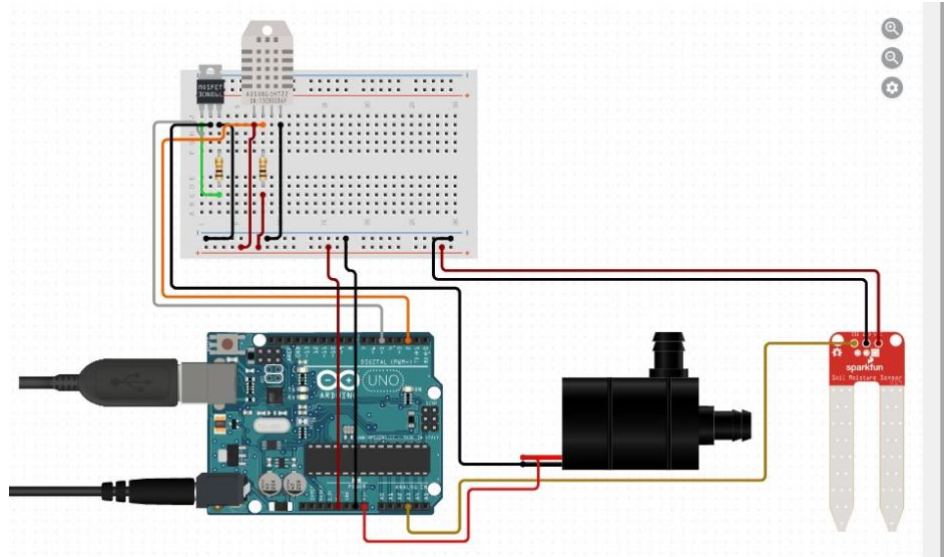
digitalWrite(pin_out,HIGH);    //open pump
sensor_analog = analogRead(sensor_pin);
moisture_percentage = ( 100 - ( (sensor_analog/1023.00) *
100 ) );
m=moisture_percentage;
delay(1000);
}
digitalWrite(pin_out,LOW);    //closepump
}
if(c>=0)
{
mySerial.begin(9600);
delay(15000);
Serial.begin(9600);
delay(1000);
Serial.print("\r");
delay(1000);

Serial.print((String)"update-
>" +(String)"Temprature="+t+(String)"Humidity="+h+(String)
)"Moisture="+m);

delay(1000);
}
}

```

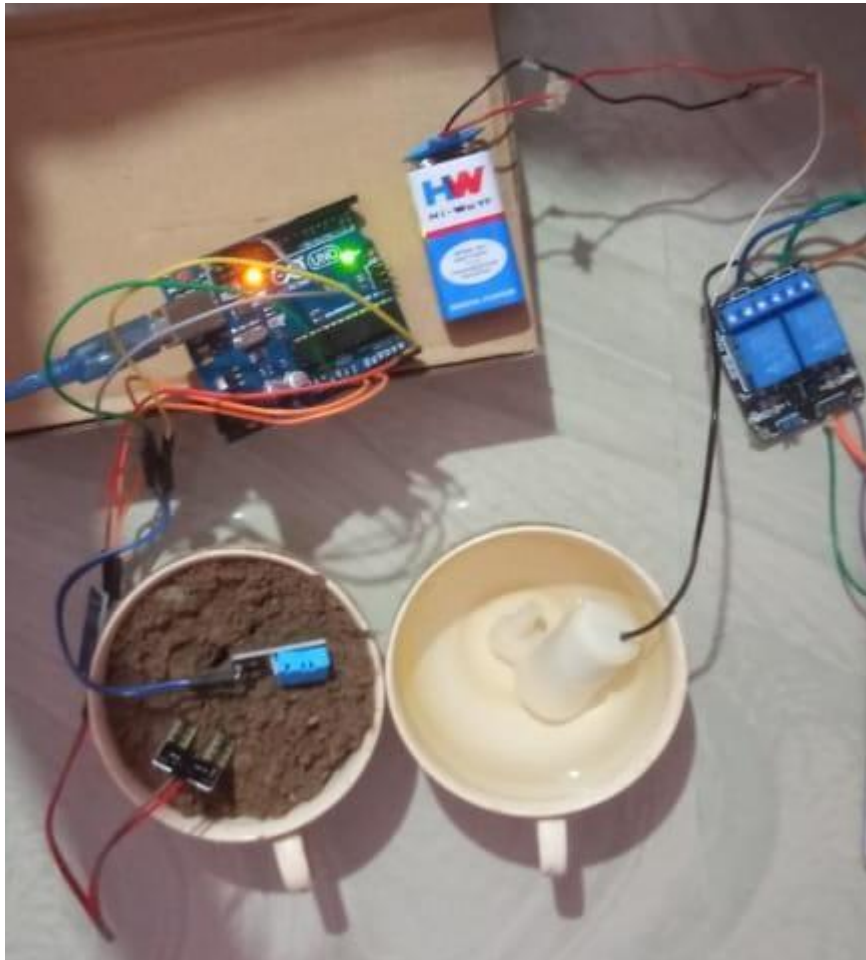
Circuit Diagram



SPRINT 2

Date	05 November 2022
Team ID	PNT2022TMID31166
Project Name	Project – Smart Farmer-IoT Enabled smart Farming Application

Sensor Connection



Device Details:

IBM Watson IoT Platform

Browse Action Device Types Interfaces

Add Device +

Browse Devices

All Devices Diagnose

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Search by Device ID

Device Simulator ☒

<input type="checkbox"/>	Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
> <input type="checkbox"/>	pga460_sensor	Disconnected	Ultrasonic	Device	Oct 25, 2022 1:19 PM	
> <input type="checkbox"/>	st_1018	Disconnected	smartfarmer	Device	Nov 5, 2022 10:39 PM	
> <input type="checkbox"/>	smartfarmer_1	Connected	smartfarmer	Device	Nov 5, 2022 10:49 PM	

Items per page 50 | 1-3 of 3 items

1 of 1 page

1 Simulation running

IBM Watson IoT Platform

Back

Device Drilldown - smartfarmer_1

Connection Information

Recent Events

State

Device Information

Metadata

Diagnostics

Connection Logs

Device Actions

Date Added: Nov 5, 2022 10:49 PM

Added By: 820419205501@smartinternz.com

Connection Status: Connected

Connection Time: Nov 5, 2022 10:50 PM

Client Address: 42.109.128.85 SecureToken

Recent Events

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
IoTSensor	["temp":15,"hum":78,"moist":66]	json	a few seconds ago
IoTSensor	["temp":41,"hum":90,"moist":15]	json	a few seconds ago
IoTSensor	["temp":54,"hum":29,"moist":81]	json	a few seconds ago
IoTSensor	["temp":46,"hum":38,"moist":77]	json	a few seconds ago

1 Simulation running

SPRINT 3

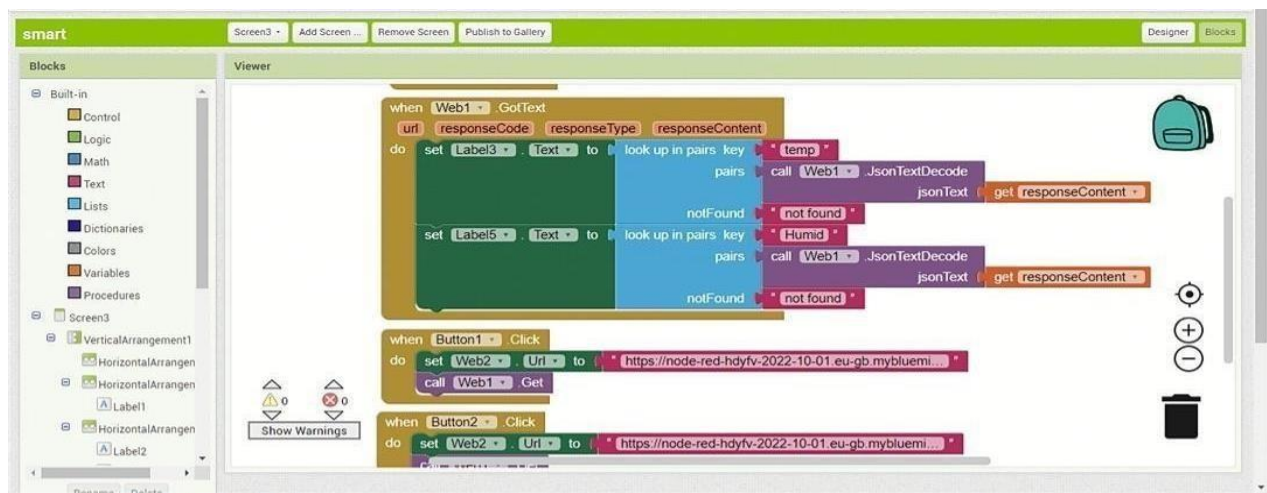
Date	12 November 2022
Team ID	PNT2022TMID31166
Project Name	Project – Smart Farmer-IoT Enabled smart Farming Application

Creating Mobile app using MIT app Inventor

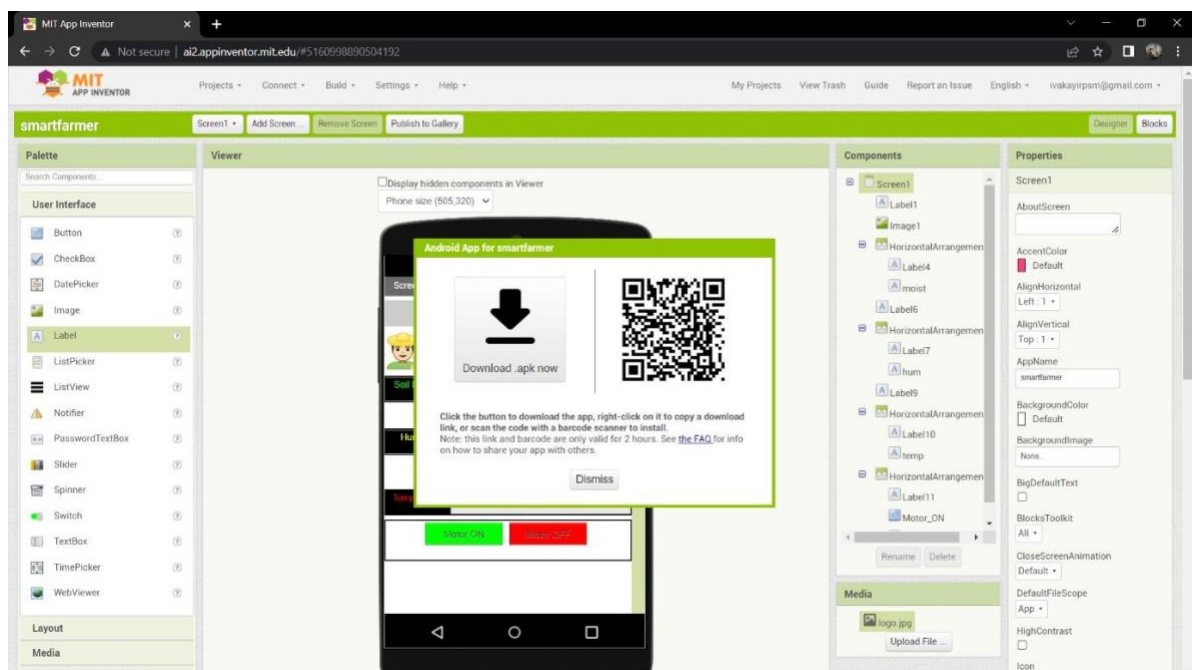
SMART FARMER APPLICATION

App Inventor + IoT can work together to help us understand, and change, our everyday lives.

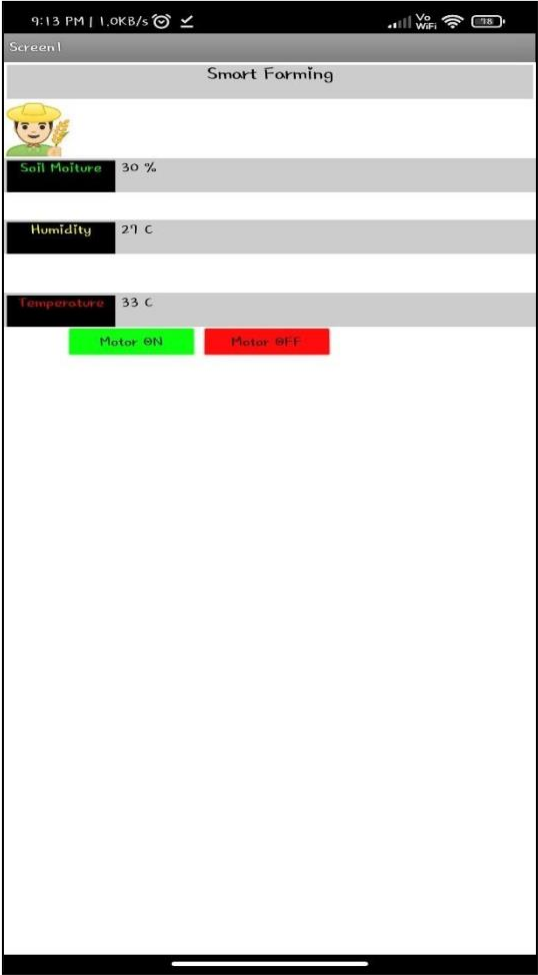
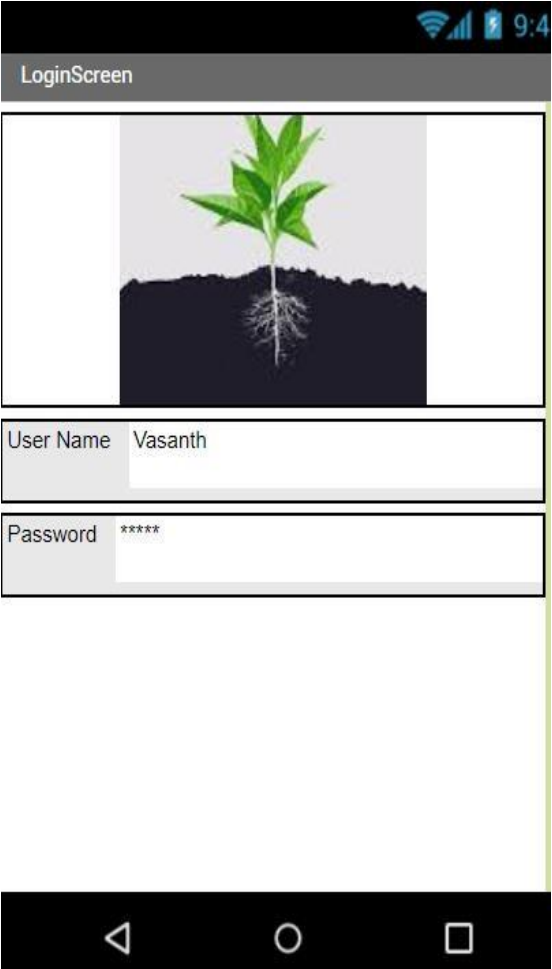
Blocks



QR Code for APK download:



Output Screen in Mobile :



SPRINT 4

Date	16 November 2022
Team ID	PNT2022TMID31166
Project Name	Project – Smart Farmer-IoT Enabled smart Farming Application

CONFIGURATION OF NODE-RED TO SEND COMMANDS TO IBM CLOUD

Here we add two buttons in UI

1 -> for motor on

2 -> for motor off

We used a function node to analyse the data received and assign command to each number.

The Java script code for the analysis is:

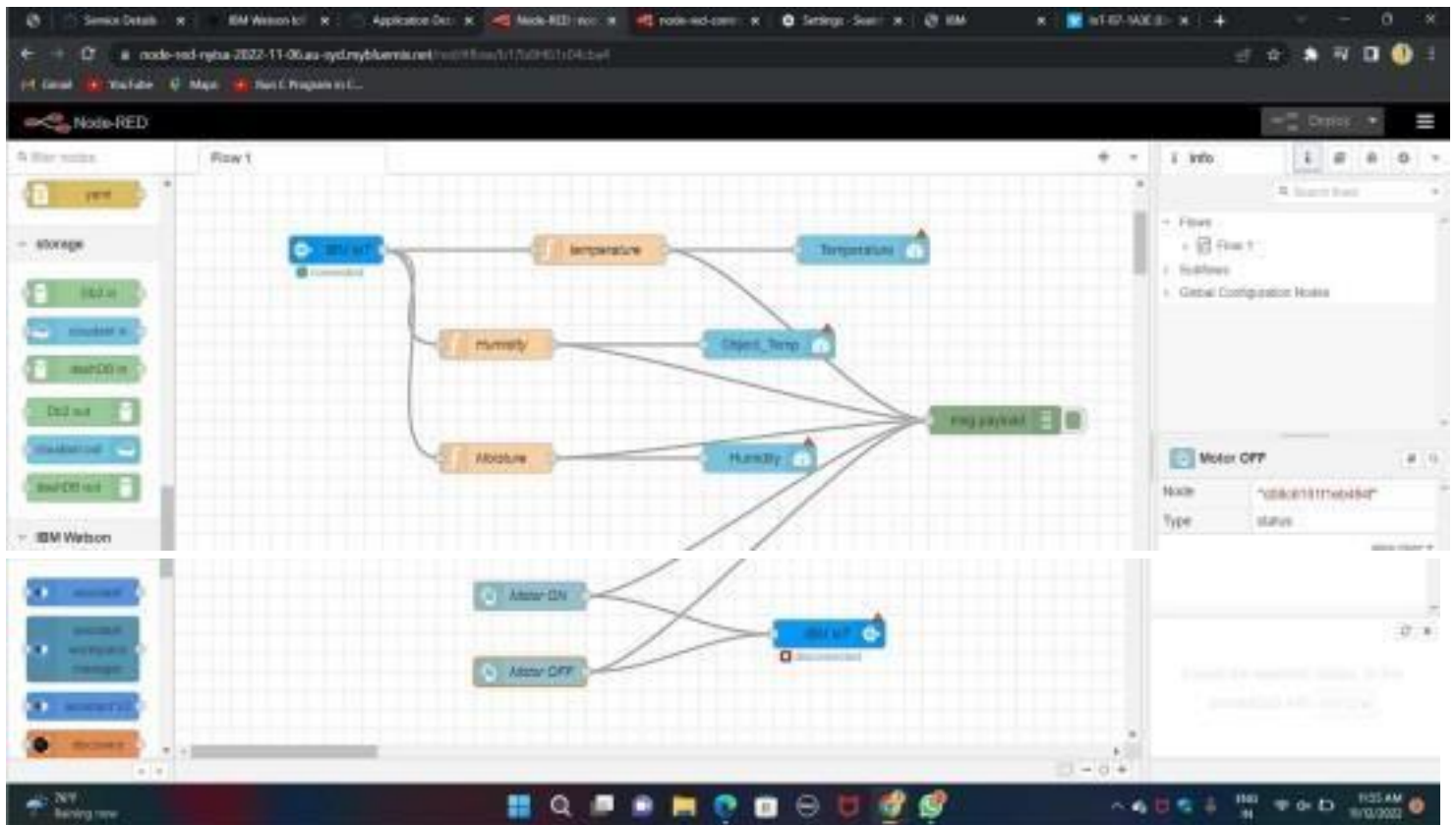
```
if(msg.payload===1)
msg.payload={"command": "ON"};
else if(msg.payload===0)
msg.payload={"command": "OFF"};
```

Adjusting User Interface

In order to display the parsed JSON data a Node-Red dashboard is created

Here we are using Gauges, text and button nodes to display in the UI and helps to monitor the parameters and control the farm equipment.

Below images we started to create the flow 1



COMPLETE PROGRAM FLOW :



SCREENSHOTS:

