ABSTRACT

Smart Farming e-Monitoring System

IDEATION:

Our ingenious idea and pragmatic plan deals with the development of '**Smart Farming e-Monitoring System'**' which is an organized 'Electronic' monitoring scheme in the Agriculture sector that aims to bring a digitized sphere in monitoring the agricultural lands. The objective of this project is the farmers can monitor his agriculture lands by using his mobile phone. **Smart farming** brings out the concept of **Internet of Things** through which he monitors the data obtained from the sensors. IoT sensors capable of providing farmers with information about crop yields, rainfall, pest infestation, and soil nutrition are invaluable to production and offer precise data which can be used to improve farming techniques. This project is useful for the upcoming younger generations because they don't know the agricultural parameters how it is to be measured. Now with the help of this project they can monitor their lands through his mobile phones and they can take necessary steps before itself to solve the problems in the field.

PROBLEMS FACING:

The problems facing by farmers in agriculture is they were struggling hard in the agricultural fields round the clock. In the agricultural sector, water pumps are used for irrigating small tracts of land from tube wells or open wells. Agriculture receives power mostly during mid night as this reduces the cost

of electricity supply for the transmission and Distribution Company. The farmers have to be on their guard all the time due to the unpredictable nature of supply of electrical energy. And the farmers have to switch on their motor after electricity supply resumes.

Due to their negligence, sometimes they switch on the motor and then forget to switch off, which may lead to wastage of water. To overcome this problem, we are going for 'Smart Farming e-Monitoring System' to monitor the real time farming processes with critical historical data, such as weather events, climate changes, resources' availability, economics, product information.

DESCRIPTION OF INNOVATION

In this project the sensors (Moisture Sensor, Temperature Sensor, Humidity sensor, Light sensor, Flow sensor)which will be placed in the field and able to monitor amount of light falling on the plants, Temperature and Humidity level, water flow rate and the moisture content in the soil. When the moisture content in the soil is too low, the system will give command to switch ON the motor and water the soil. The flow meter monitors the water consumption.

The data obtained from the sensors will be send to the Cloud where the farmer can monitor the status of his land, availability of water and light intensity. Based on the moisture content, the Water Pump will be automatically Switched ON for watering the plant. Finally the Notification message will be send to the Farmer's Mobile regarding the amount of water flows to the land.

COMPONENTS USED IN THE PRODUCT:

Hardware part:

- Temperature Sensor
- Humidity Sensor
- Moisture Sensor
- Light Sensor

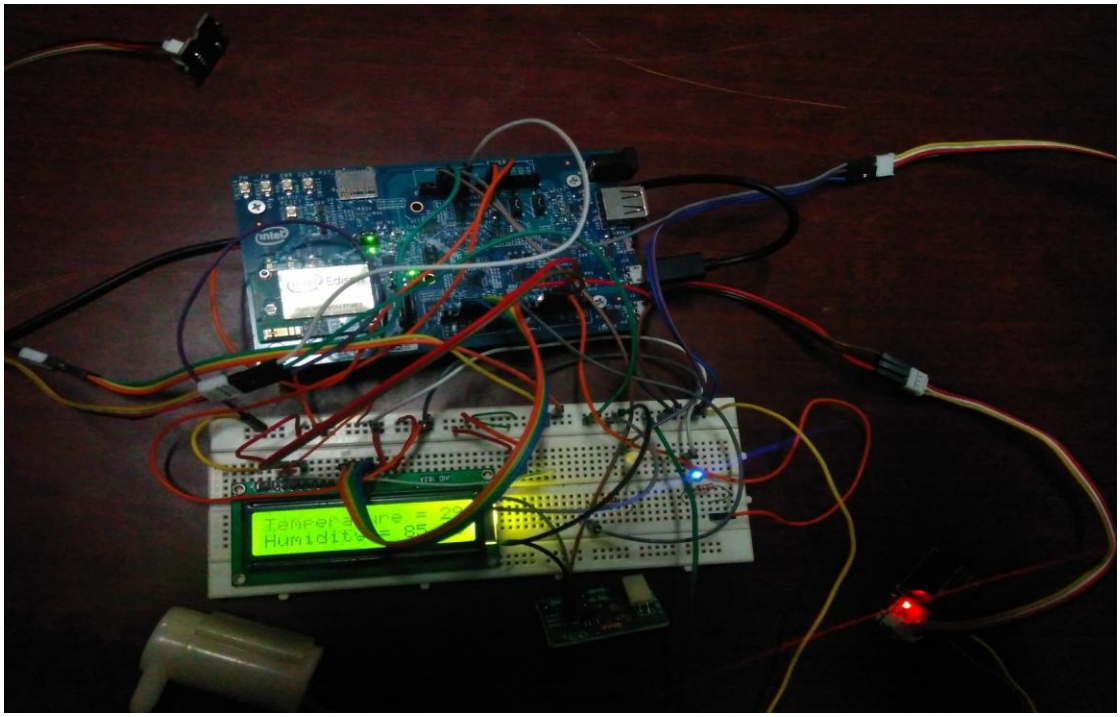
Software part:

- Arduino IDE
- Blynk App

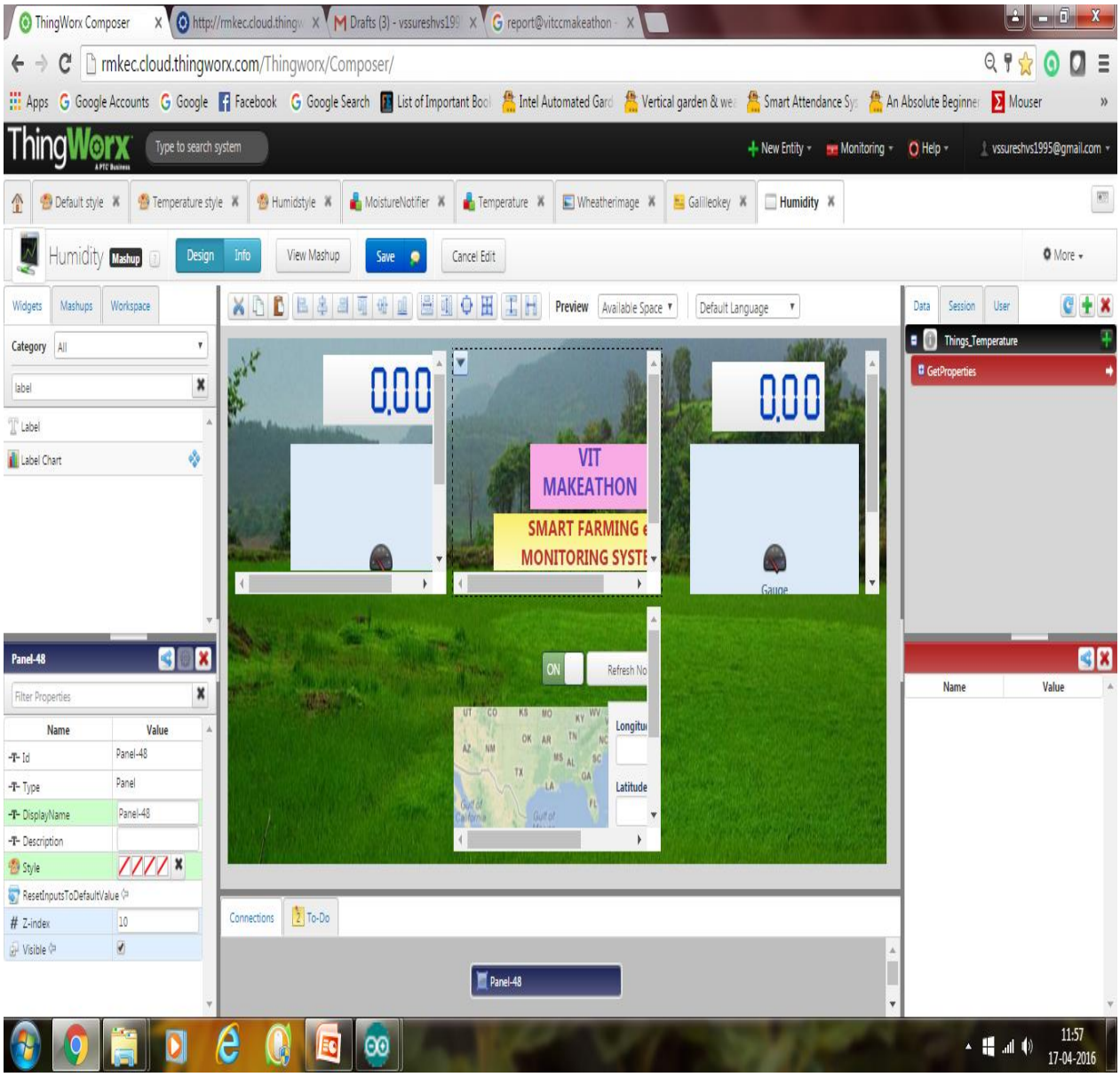
CLOUD services:

- THINGWORX PLATFORM

HARDWARE CONNECTIONS:

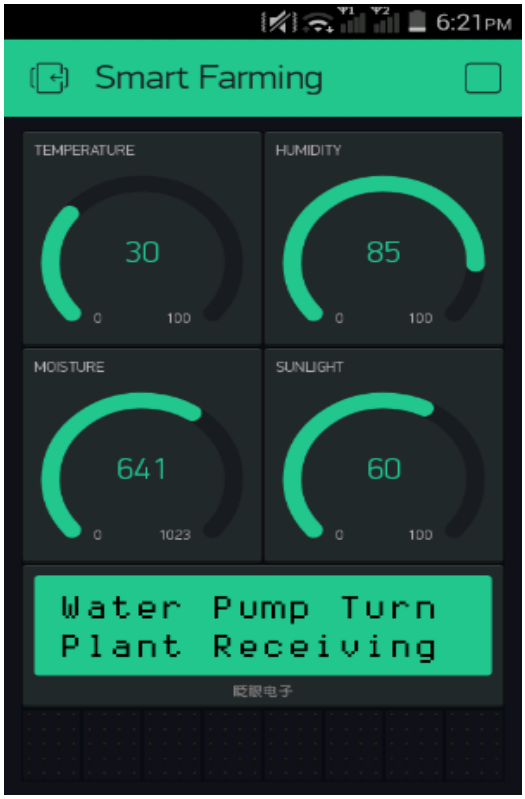


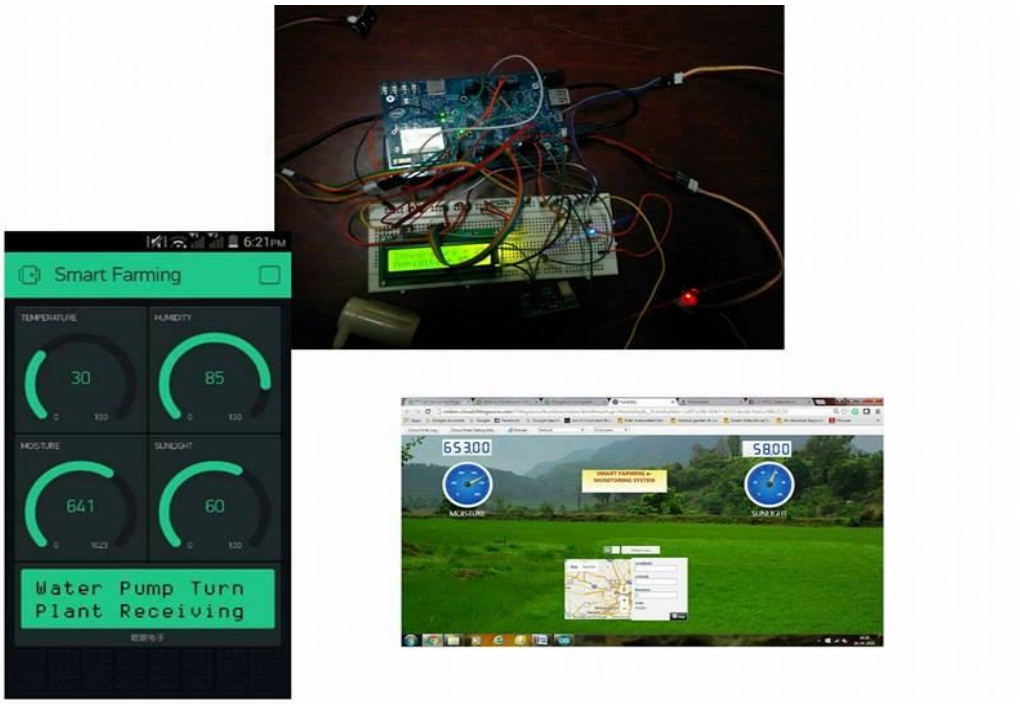
WORKING WITH THINGWORX PLATFORM:



DATA STREAMING INTO THINGWORX CLOUD PLATFORM:

FARMERS GETTING THE DATA FROM CLOUD TO HIS MOBILE PHONE:





CONCLUSION:

This Project enables precision agriculture in order to maximize food production, minimize environmental impact and reduce costs. This proposed idea of ‘Smart Farming e-Monitoring System' turning out to be an eye-opener to the whole world considering the realistic and innovative prospects offered in the domain.

SOURCE CODE:

```
//SMART FARMING#THINGWORX #BLYNK
//THINGWORX CLOUD
#include <twApi.h>
#include <twLogger.h>
#include <twOSPort.h>
#include <WiFi.h>
```

```
#include <stdio.h>
#include <string.h>
#include <Wire.h>
#define BLYNK_PRINT Serial
//#include <DHT.h>

#define BLYNK_PRINT Serial
// Comment this out to disable prints and save space
#include <BlynkSimpleIntelEdisonWiFi.h>
char auth[] = "0db731b46cd349caa6fd2097f0cbf6b8";
WidgetLCD lcd1(V4);

#include <LiquidCrystal.h>
char ssid[] = "Smart  Farming";  //Wifi username
char pass[] = "Farmer"; //Wifi  password

int status = WL_IDLE_STATUS;

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
//Temperature
const int pinTemp = A0;
const int B = 3975;

//Humidity
#define DHTIN 12
#define DHTOUT 9
#define DHTTYPE DHT11
DHT dht(DHTIN,DHTOUT,DHTTYPE);


//Moisture
int sensorPin = A1;    // select the input pin for the potentiometer
int sensorValue;
int relay = 6;
int pumpLed   = 13;

//Sunlight
```

```
const int pinLight = A2;
const int pinLed   = 7;
int thresholdvalue = 80;

/* Name of your thing */
char * thingName = "Temperature";
/* IP/hostname of your TWX server */
char * serverName = "rmkec.cloud.thingworx.com";
/* port */
int port = 80;
/* API key */
char * apiKey = "001e8013-72db-40d1-954e-a8916fa59f72";
/* refresh rate */
int timeBetweenRefresh = 1000;

/* Hold all the properties */
struct  {
  double Moisture;
  double Sunlight;
  double Temperature;
  double Humidity;
}
properties;



void sendPropertyUpdate() {
  /* Create the property list */
 propertyList * proplist =
twApi_CreatePropertyList("Moisture",twPrimitive_CreateFromNumber(propertie
s.Moisture), 0);
  if (!proplist) {
    TW_LOG(TW_ERROR,"sendPropertyUpdate: Error allocating property list");
    return;
  }
```

```
twApi_AddPropertyToList(proplist,"Sunlight",twPrimitive_CreateFromNumber(pr
operties.Sunlight), 0);
  twApi_PushProperties(TW_THING, thingName, proplist, -1, FALSE);
  twApi_DeletePropertyList(proplist);
}

void dataCollectionTask() {
  //Temperature Readings
  int val = analogRead(pinTemp);
  float resistance = (float)(1023-val)*10000/val;
  int temperature = 1/(log(resistance/10000)/B+1/298.15)-273.15;
    properties.Temperature=temperature;
      lcd.setCursor(0,0);
      lcd.print("Temperature = ");
      lcd.print(properties.Temperature);
    delay(500);

  //Humidity Readings
  float h = dht.readHumidity();
      properties.Humidity= h;
    lcd.setCursor(0,1);
    lcd.print("Humidity = ");
    lcd.print(properties.Humidity);
    delay(500);


  //Moisture Readings
  sensorValue = analogRead(sensorPin);
    delay(500);
  properties.Moisture=sensorValue;


  //Sunlight Readings
  int sensor = analogRead(pinLight);
  int sunlight = map(sensor,1023, 0,0,100);
  properties.Sunlight=sunlight;
```

```
Serial.print("Time:");
Serial.print(millis());
Serial.print(" \n >>>>>Temperature:");
Serial.print(properties.Temperature, 1);
Serial.print("C");
Blynk.virtualWrite(0, properties.Temperature);

Serial.print(" \n>>>>>> Humidity:");
Serial.print(properties.Humidity, 1);
Serial.print("%");
Blynk.virtualWrite(1, properties.Humidity);

Serial.print("\n >>>>>Moisture:");
Serial.print(properties.Moisture, 1);
Serial.print("%");
Blynk.virtualWrite(2, properties.Moisture);
  if (properties.Moisture <90)
{
  digitalWrite(relay, HIGH);
  digitalWrite(pumpLed, HIGH);
  Serial.print("\n Soil is Dry");
  Serial.print("\t Water Pump Turn ON");
  lcd1.print(0, 0, "Pump Turn ON");
  lcd1.print(0, 1, "Soil is Dry");
}
else if( properties.Moisture > 90 && properties.Moisture < 200) {
  Serial.print("\n Soil is Humid");
  lcd1.print(0, 0, "Soil is Humid");
  digitalWrite(relay,HIGH);
  digitalWrite(pumpLed,LOW);
}

  else{
  digitalWrite(relay, LOW);
  digitalWrite(pumpLed, LOW);
 Serial.print("\n Soil is Wet");
```

```
  Serial.print("\t Water Pump turn off");
  lcd1.print(0, 0, "Pump Turn OFF");
 }

 Serial.print(" \n >>>>Sunlight:");
 Serial.print(properties.Sunlight, 1);
 Blynk.virtualWrite(3, properties.Sunlight);
 if(properties.Sunlight > thresholdvalue)
  {
     digitalWrite(pinLed, HIGH);
     Serial.print("\n Plant Receiving Sun Light");
     Serial.print("\t  Chance of Rain Today is 10%");
  }
  else if(properties.Sunlight > 50 && properties.Sunlight<80){
       digitalWrite(pinLed, LOW);
       Serial.print("\nPlant Receiving Sun Light");
       Serial.print("\t Chance of Rain Today is 40%");
       //lcd1.print(0, 1, "Plant Receiving SunLight");
  }

  else
  {
     digitalWrite(pinLed, LOW);
     Serial.print("\n Climate is CLOUDY....");
     Serial.print("\t Chance of Rain today is 50%");
     //lcd1.print(0, 1, "Climate is CLOUDY");
  }
 Serial.println();
 /* Update the properties on the server */
 sendPropertyUpdate();
}



/*****************
 * Property Handler Callbacks
 *****************/
```

```
enum msgCodeEnum propertyHandler(const char * entityName, const char *
propertyName,  twInfoTable ** value, char isWrite, void * userdata) {
  char * asterisk = "*";
  if (!propertyName) propertyName = asterisk;
  TW_LOG(TW_TRACE,"propertyHandler - Function called for Entity %s, Property
%s", entityName, propertyName);
  if (value) {

     /* Property Reads */
     if (strcmp(propertyName, "Moisture") == 0) *value =
twInfoTable_CreateFromNumber(propertyName, properties.Moisture);
     else if (strcmp(propertyName, "Sunlight") == 0) *value =
twInfoTable_CreateFromNumber(propertyName, properties.Sunlight);
     else return TWX_NOT_FOUND;
    return TWX_SUCCESS;
  }
  else {
    TW_LOG(TW_ERROR,"propertyHandler - NULL pointer for value");
    return TWX_BAD_REQUEST;
  }
}

void setup() {
  int err=0;
  /* Open serial connection */
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass);
  pinMode(relay, OUTPUT);
  pinMode(pinLed, OUTPUT);
  lcd.init(1,12,255,11,5,4,3,2,0,0,0,0);
//  dht.begin();
  lcd.begin(16, 2);
    /* Wait for someone to launch the console */
  delay(1000);

  /* Setup the WiFi connection */
  if (WiFi.status() == WL_NO_SHIELD)
```

```
{ Serial.println("WiFi is not connected");
  // don't continue:
  while(true); }
// attempt to connect to Wifi network:
  if ( status != WL_CONNECTED)
{ Serial.print("Wifi is connecting to "); Serial.println(ssid);
// Connect to WPA/WPA2 network:
  status = WiFi.begin(ssid, pass);
// wait 10 seconds for connection: delay(10000);
} Serial.println("You're connected to the network");
  Serial.println("SMART FARMING!");
  delay(500);
  Serial.println("Intel Edison communicating with cloud");
  delay(1000);

  err = twApi_Initialize(serverName, port, TW_URI, apiKey, NULL,
MESSAGE_CHUNK_SIZE, MESSAGE_CHUNK_SIZE, TRUE);
  if (err) {
    Serial.println("Error initializing the API");
  }

  /* Allow self signed certs */
  twApi_SetSelfSignedOk();

  /* Regsiter our properties */
   twApi_RegisterProperty(TW_THING, thingName, "Moisture", TW_NUMBER,
NULL, "ALWAYS", 0, propertyHandler,NULL);
   twApi_RegisterProperty(TW_THING, thingName, "Sunlight", TW_NUMBER,
NULL, "ALWAYS", 0, propertyHandler,NULL);

  /* Bind our thing */
  twApi_BindThing(thingName);

  /* Connect to server */
  if (!twApi_Connect(CONNECT_TIMEOUT, CONNECT_RETRIES)) {
    Serial.println("sucessefully connected to Cloud!");
  }
```

```
}

void loop() {
  // put your main code here, to run repeatedly:
  delay(500);
  dataCollectionTask();
  delay(1000);
  Blynk.run();
  Serial.print(".");
}
```