

A PROJECT REPORT

ON

**WIRELESS SENSOR NETWORK BASED SMART IRRIGATION SYSTEM**

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS

OF THE DEGREE OF

B. Tech

BY

1. V.V.DURGAPRASAD (177Y1A05D0 )
2. ANVITHA (187Y1A0468)
3. SWAPNA (187Y1A04B4)

UNDER THE ESTIMED GUIDANCE OF



ACKNOWLEDGEMENT:

A dissertation work, which integrated skill full and hard work need a great deal of guidance and helping hand. I would like to add a few heartfelt words for the people who have directly or indirectly in preparing in this dissertation work in numerous ways.

My heartfelt thanks to **Marri Laxman Reddy**, chairman of our college for providing me an opportunity to care out this project work form this college.

My sincere thanks to **Dr. K Venkateswara Reddy**, the principal of our college for providing me the required facilities to complete my project.With profound sense of gratitude and regards, I acknowledge with great pleasure the Guidance and support extended by, **Mr. K Abdul Basith & Mr. K Nagabhushanam**, heads of the Computer Science & Electronics and Communication Engineering departments and internal guide, **Mr. B Prasad** for his timely suggestion, logical thought that have greatly contributed in bringing out the project in a systematic way. He has been inculcating innovative skills and technicalities at all stages at performing this project.

My sincere thanks to all the teaching and non- teaching staff of our department for their constant co-operation extended to me to complete the project successfully. My thanks to all my classmates and friends for their constant help and support during my project work.My heartfelt gratitude to my parents and my family for their constant encouragement throughout the work carried out to prepare this project report.

*INDEX:*

Abstract - 4

Introduction - 5

Block Diagram - 6

Hardware and Software - 7

requirements

Node Mcu - 8

Soil moisture - 9

DHT11 sensor - 10

DC motor - 11

L293D Motor Driver - 12

Arduino ide - 13

Ibm cloud

Iot platform - 14

Node red - 15

Mit app inventor - 16

PROGRAM - 17-25

DIAGRAM - 26

ADVANTAGES - 27

TEMPERATURE,HUMIDITYAND

SOIL MOISTURE READING IN SERIAL MONITOR - 28

TEMPERATURE,HUMIDITY AND SOIL MOISTURE

READING IN NODE RED - 29

SMART IRRIGATION SYSTEM APP - 30

CONCLUSION - 31

**ABSTRACT:**

This paper on "Smart Irrigation System" is develop to create an automated irrigation mechanism which turns the pumping motor ON and OFF on detecting the moisture content of the earth using the soil moisture sensor without the intervention of human. The benefit of employing these techniques is to decrease human interference and it is quite feasible and affordable. This Smart irrigation system project is using a node mcu micro-controller, that is programmed to collect the input signal according to moisture content of the soil and its output is given to the that will operate the pump. The key objective of the paper is to monitor the soil’s moisture content during its dry and wet conditions with the aid of a moisture sensor circuit, an automatic water inlet setup which can also monitor and record temperature, humidity and sunlight, which is constantly modified and can be controlled in future to optimize these resources so that the plant growth and yield is maximized.

Introduction:

In India, agriculture is the need of most of the Indians livelihood and it is one of the main sources of livelihood. Agriculture also has a major impact on economy of the country. The consumption of water increases day by day that may leads to the problem of water scarcity. Now a days, farmer are struggling hard in the agriculture field and the task of irrigating field is becoming quite difficult for the farmers due to lack of regularity in their work and negligence because sometimes they switch on the motor and then forget to switch off which may leads to wastage of water. Similarly, they even forget to switch ON the irrigation system, which again leads to damage to the crops. To overcome this problem, we have implemented a new technique by using node mcu micro controller. In this project we are using soil moisture sensor which is used to sensing moisture level whether the soil is dry or wet. The moisture sensor is interface with node mcu micro controller that will work by the process of simulating on ARDUINO IDE software and based on that it activate the DC motor through op-amp which compare the level of moisture content of the soil with the reference value that will operate the pump through relay.. Water is a very precious resource and must be properly utilized. Agriculture is one of those areas which consume a lot of water. Irrigation is a time consuming process and must be done on a timely basis.The aim of the article is to develop an smart irrigation system

BLOCK DIAGRAM:

SOIL MOISTURE

DHT SENSOR

DC MOTOR

NODE MCU

IBM CLOUD

MOBILE APP

HARDWARE AND SOFTWARE REQUIREMENTS:

HARDWARE COMPONENTS:

1.NODE MCU

2.SOIL MOISTURE

3.DHT SENSOR

4.DC MOTOR

5.L293D MOTOR DRIVER

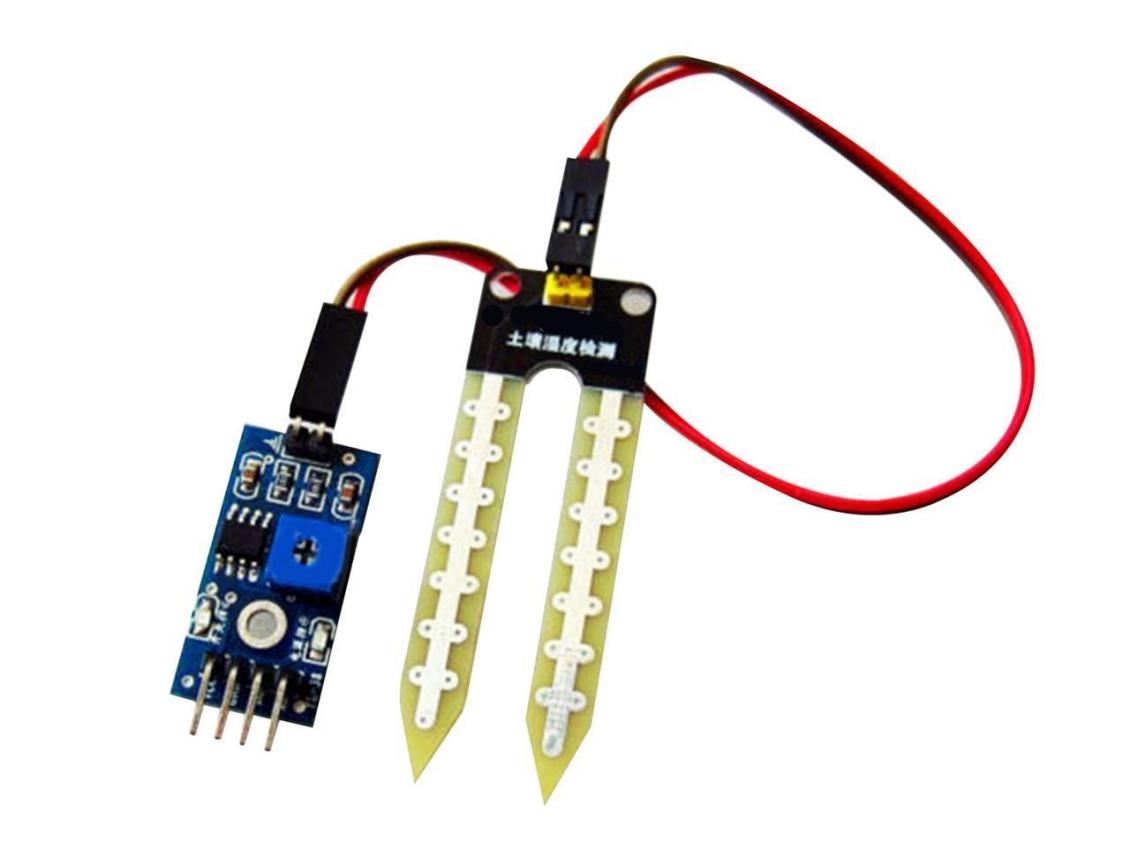
HARDWARE REQUIREMENTS:

1.NODE MCU:



**NodeMCU** is an open source [IoT](https://en.wikipedia.org/wiki/Internet_of_Things" \t "Internet of Things) platform. It includes [firmware](https://en.wikipedia.org/wiki/Firmware" \t "Firmware) which runs on the [ESP8266](https://en.wikipedia.org/wiki/ESP8266" \t "ESP8266) [Wi-Fi](https://en.wikipedia.org/wiki/Wi-Fi" \t "Wi-Fi) [SoC](https://en.wikipedia.org/wiki/System_on_a_chip" \t "System on a chip) from [Espressif Systems](https://en.wikipedia.org/w/index.php?title=Espressif_Systems&action=edit&redlink=1" \t "Espressif Systems (page does not exist)), and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the development kits. The firmware uses the [Lua](https://en.wikipedia.org/wiki/Lua_(programming_language)" \t "Lua (programming language)) scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson and [SPIFFS](https://en.wikipedia.org/w/index.php?title=SPIFFS&action=edit&redlink=1" \t "SPIFFS (page does not exist)).

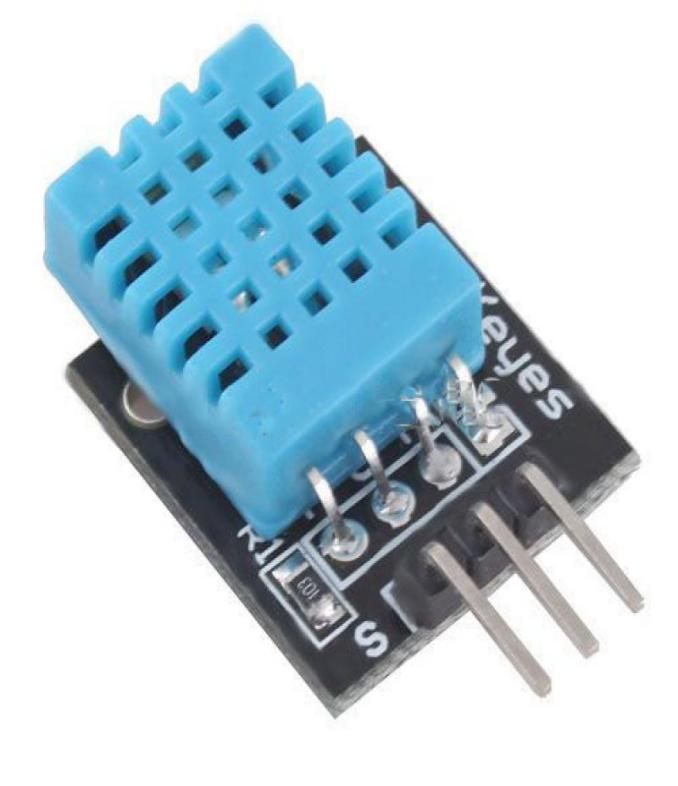
1. SOIL MOISTURE:



Soil moisture is basically the content of water present in soil. This can be measured using a soil moisture sensor which consists of two conducting probes that act as a probe. It can measure the moisture content in the soil based on the change in resistance between the two conducting plates.

The resistance between the two conducting plates varies in an inverse manner with the amount of moisture present in the soil.

3.DHT11 SENSOR:



DHT11 sensor measures and provides humidity and temperature values serially over a single wire.It can measure relative humidity in percentage (20 to 90% RH) and temperature in degree Celsius in the range of 0 to 50°C.It has 4 pins; one of which is used for data communication in serial form.Pulses of different TON and TOFF are decoded as logic 1 or logic 0 or start pulse or end of frame.The electric connection to the NodeMCU is very simple, as the DHT series can be powered direct with 3.3V. Only 3 wires are needed: VCC, GND and the data line.

4.DC MOTOR:

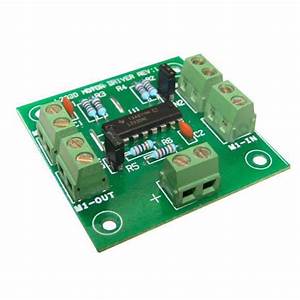


DC motor converts electrical energy in the form of Direct Current into mechanical energy in the form of rotational motion of the motor shaft.

The DC motor speed can be controlled by applying varying DC voltage; whereas the direction of rotation of motor can be changed by reversing the direction of current through it.

For applying varying voltage, we can make use of PWM technique.For reversing the current, we can make use of H-Bridge circuit or motor driver ICs that employ the H-Bridge technique.NodeMCU based ESP8266 can be used to control the speed and rotational direction of DC Motor. NodeMCU has PWM feature on its GPIO pins using which we can control DC motor.

1. L293D MOTOR DRIVER:



A motor driver is an integrated circuit chip which is usually used to control motors in autonomous robots. Motor driver act as an interface between Arduino and the motors . The most commonly used motor driver IC’s are from the L293 series such as L293D, L293NE, etc. These ICs are designed to control 2 DC motors simultaneously. L293D consist of two H-bridge. H-bridge is the simplest circuit for controlling a low current rated motor. We will be referring the motor driver IC as L293D only. L293D has 16 pins.

SOFTWARE REQUIREMENTS:

Arduino IDE

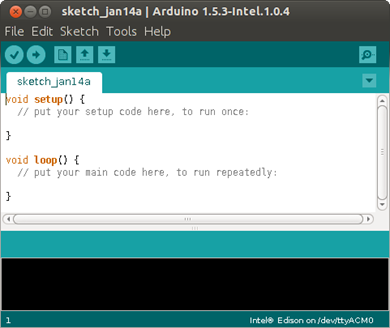
IBM Cloud

IOT Platform

Node-Red

**1.Arduino IDE:**

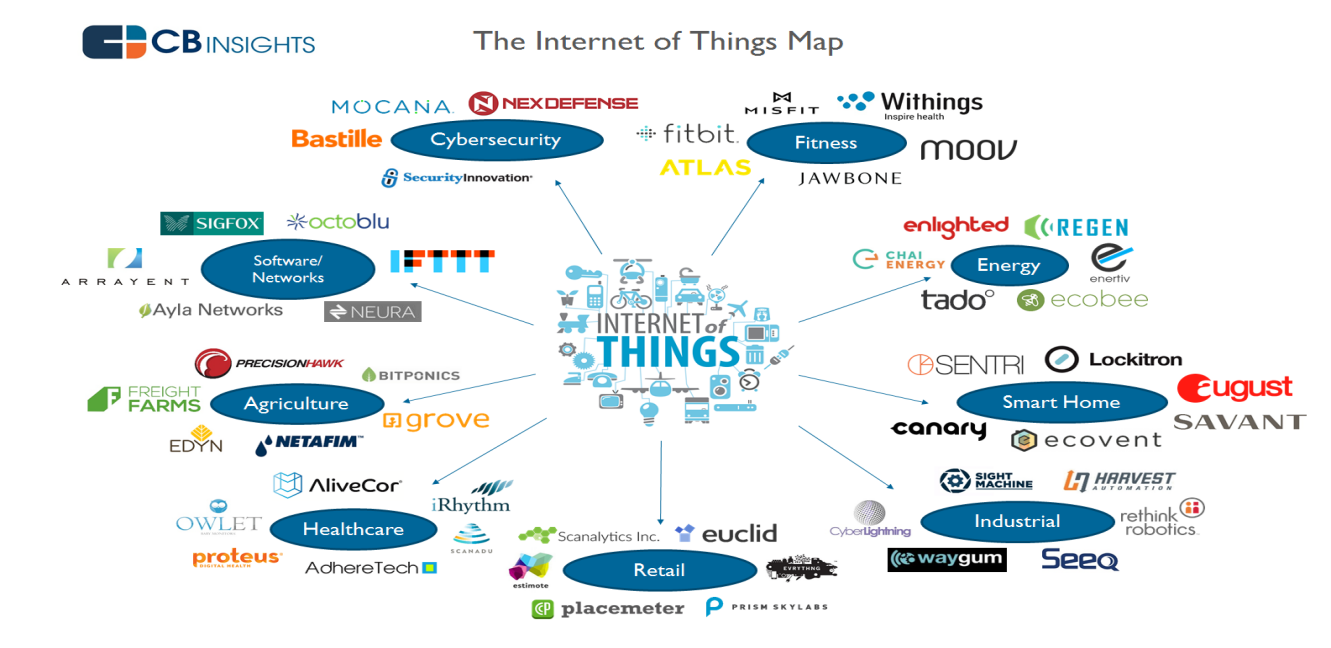
The [Arduino](https://en.wikipedia.org/wiki/Arduino" \t "Arduino) integrated development environment ([IDE](https://en.wikipedia.org/wiki/Integrated_development_environment" \t "Integrated development environment)) is a [cross-platform](https://en.wikipedia.org/wiki/Cross-platform" \t "Cross-platform) application (for [Windows](https://en.wikipedia.org/wiki/Windows" \t "Windows), [macOS](https://en.wikipedia.org/wiki/MacOS" \t "MacOS), [Linux](https://en.wikipedia.org/wiki/Linux" \t "Linux)) that is written in the programming language [Java](https://en.wikipedia.org/wiki/Java_(programming_language)" \t "Java (programming language)). It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.The source code for the IDE is released under the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License" \t "GNU General Public License), version 2. The Arduino IDE supports the languages [C](https://en.wikipedia.org/wiki/C_(programming_language)" \t "C (programming language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B" \t "C++) using special rules of code structuring.



**2.IBM Cloud:**

1. **IOT Platform:**

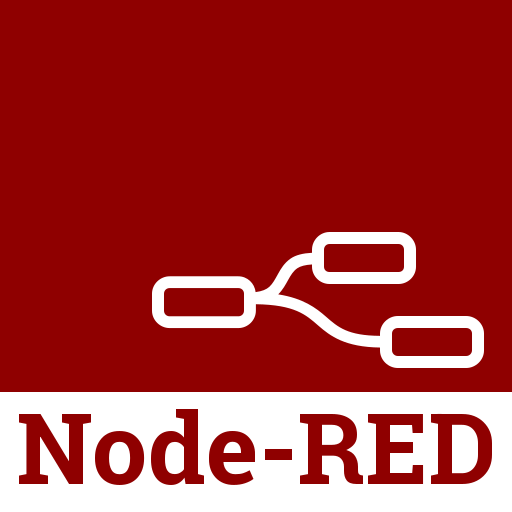
An IoT platform is a multi-layer technology that enables straightforward provisioning, management, and automation of connected devices within the Internet of Things universe. It basically connects your hardware, however diverse, to the cloud by using flexible connectivity options, enterprise-grade security mechanisms, and broad data processing powers. For developers, an IoT platform provides a set of ready-to-use features that greatly speed up development of applications for connected devices as well as take care of scalability and cross-device compatibility.



1. **Node-Red:**

Node-RED is a [flow-based](https://en.wikipedia.org/wiki/Flow-based_programming" \t "Flow-based programming) development tool for [visual programming](https://en.wikipedia.org/wiki/Visual_programming_language" \t "Visual programming language) developed originally by [IBM](https://en.wikipedia.org/wiki/IBM" \t "IBM) for wiring together hardware devices, [APIs](https://en.wikipedia.org/wiki/Application_programming_interface" \t "Application programming interface) and [online services](https://en.wikipedia.org/wiki/Online_services" \t "Online services) as part of the [Internet of Things](https://en.wikipedia.org/wiki/Internet_of_Things" \t "Internet of Things).

Node-RED provides a [web browser](https://en.wikipedia.org/wiki/Web_browser" \t "Web browser)-based flow editor, which can be used to create [JavaScript](https://en.wikipedia.org/wiki/JavaScript" \t "JavaScript) functions. Elements of applications can be saved or shared for re-use. The runtime is built on [Node.js](https://en.wikipedia.org/wiki/Node.js" \t "Node.js). The flows created in Node-RED are stored using [JSON](https://en.wikipedia.org/wiki/JSON" \t "JSON). Since version 0.14 [MQTT](https://en.wikipedia.org/wiki/MQTT" \t "MQTT) nodes can make properly configured [TLS](https://en.wikipedia.org/wiki/Transport_Layer_Security" \t "Transport Layer Security) connections.

****

1. MIT app inventor 2:



MIT App inventor 2 is most advanced cloud based app building tool. Using App inventor you can build apps from your web browser very quickly and easily. The MIT App tool is very useful for beginners to get their own apps. You can build multiple type apps in simple ways like drag and drop. You will find many tools to build your app in minutes. you can follow the link below to get your fist app ready.MIT App Inventor 2 was originally constructed as a programming language and is sponsored by Google as an open-source web application. Accredited by the Massachusetts Institute of Technology (MIT), it provides abundant resources for children to learn computer programming and create software applications for the Android operating system

PROGRAM FOR SMART IRRIGATION SYSTEM IN ARDUINO IDE:

const int sensor\_pin = A0;

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

void callback(char\* topic, byte\* payload, unsigned int payloadLength);

const char\* ssid = "VASU GUITAR";

const char\* password = "VASU2660";

#include "DHT.h"

#define DHTPIN D2

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

#define ORG "ti62vq"

#define DEVICE\_TYPE "nodemcu555"

#define DEVICE\_ID "5555"

#define TOKEN "55555555"

String data;

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";

char subtopic[] = "iot-2/cmd/home/fmt/String";

char pubtopic[] = "iot-2/evt/dht11/fmt/json";

char authMethod[] = "use-token-auth";

char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE\_TYPE ":" DEVICE\_ID;

WiFiClient wifiClient;

PubSubClient client(server, 1883, callback, wifiClient);

float moisture\_percentage;

void setup()

{

Serial.begin(9600);

Serial.println();

wifiConnect();

mqttConnect();

Serial.println("DHTxx test!");

dht.begin();

Serial.begin(9600);

pinMode(D6, OUTPUT);

pinMode(D7, OUTPUT);

Serial.print("Connecting to ");

Serial.print(ssid);

WiFi.begin(ssid, password);

while (WiFi.status()!= WL\_CONNECTED)

{

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.print("WiFi connected, IP address : ");

Serial.println(WiFi.localIP());

}

void loop()

{

if (!client.loop())

{

mqttConnect();

}

delay(100);

float moisture\_percentage;

moisture\_percentage = ( 100.00 - ( (analogRead(sensor\_pin)/1023.00) \* 100.00 ) );

Serial.print("Soil Moisture(in Percentage) = ");

Serial.print(moisture\_percentage);

Serial.println("%");

float h = dht.readHumidity();

float t = dht.readTemperature();

float f = dht.readTemperature(true);

if (isnan(h) || isnan(t) || isnan(f))

{

Serial.println("Failed to read from DHT sensor!");

return;

}

Serial.print("Humidity: ");

Serial.print(h);

Serial.print(" %\t");

Serial.print("Temperature: ");

Serial.print(t);

Serial.println(" \*C ");

digitalWrite(D6,LOW);

digitalWrite(D7,HIGH);

PublishData(t,h,moisture\_percentage);

delay(1000);

}

void wifiConnect()

{

Serial.print("Connecting to "); Serial.print(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED)

{

delay(500);

Serial.print(".");

}

Serial.print("nWiFi connected, IP address: "); Serial.println(WiFi.localIP());

}

void mqttConnect()

{

if (!client.connected())

{

Serial.print("Reconnecting MQTT client to ");

Serial.println(server);

while (!client.connect(clientId, authMethod, token))

{

Serial.print(".");

delay(500);

}

initManagedDevice();

Serial.println();

}

}

void initManagedDevice()

{

if (client.subscribe(subtopic))

{

Serial.println("subscribe to cmd OK");

}

else

{

Serial.println("subscribe to cmd FAILED");

}

}

void callback(char\* topic, byte\* payload, unsigned int payloadLength)

{

Serial.print("callback invoked for topic: ");

Serial.println(subtopic);

for (int i = 0; i < payloadLength; i++)

{

data += (char)payload[i];

}

Serial.println(data);

if(data == "light\_ON")

{

digitalWrite(D1,HIGH);

Serial.println("Light is Switched ON");

}

else if(data == "light\_OFF")

{

digitalWrite(D1,LOW);

Serial.println("Light is Switched OFF");

}

data ="";

}

void PublishData(float temp, float humid,float m)

{

if (!client.connected())

{

Serial.print("Reconnecting client to ");

Serial.println(server);

while (!client.connect(clientId, authMethod, token))

{

Serial.print(".");

delay(500);

}

Serial.println();

}

String payload = "{\"d\":{\"temperature\":";

payload += temp;

payload+="," "\"humidity\":";

payload += humid;

payload+="," "\"soil\_moisture\":";

payload += m;

payload += "}}";

Serial.print("Sending payload: ");

Serial.println(payload);

if (client.publish(pubtopic, (char\*) payload.c\_str()))

{

Serial.println("Publish ok");

}

else

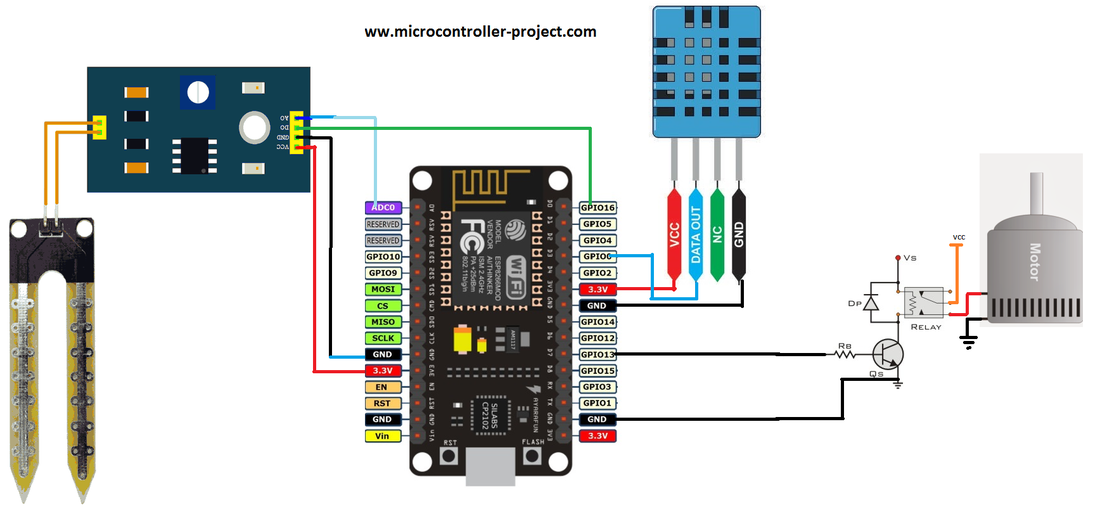
{

Serial.println("Publish failed");

}

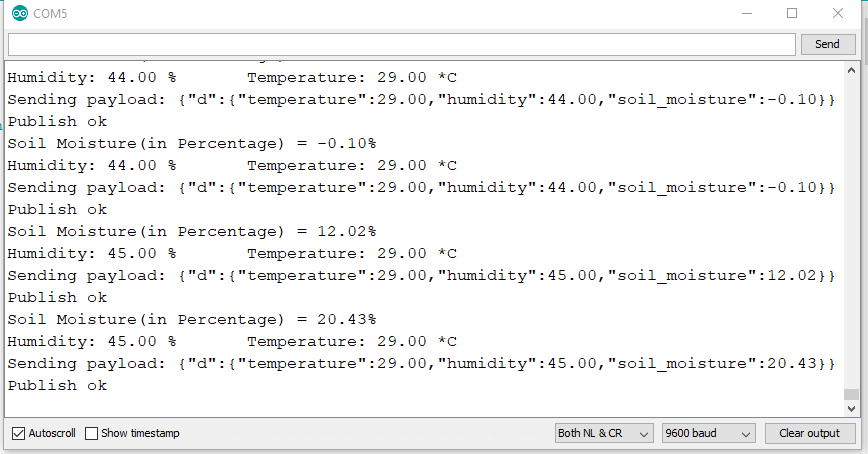
}

DIAGRAM ABOUT SMART IRRIGATION SYSTEM:

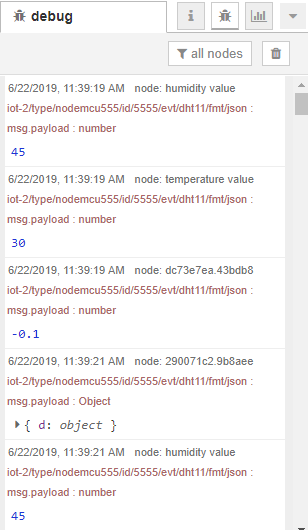
BASIC ADVANTAGES SMART IRRIGATION SYSTEM:

1. One of the greatest advantages of a smart irrigation system is its ability to save water.
2. The first thing that a smart irrigation system will save you is money. You’re paying for water and perhaps even sewer costs, and you want to make sure you’re getting the most from every penny you spend. Smart irrigation ensures that you have to use up to 70% less water. That means spending 70% less money to water your business’ lawn and plants.

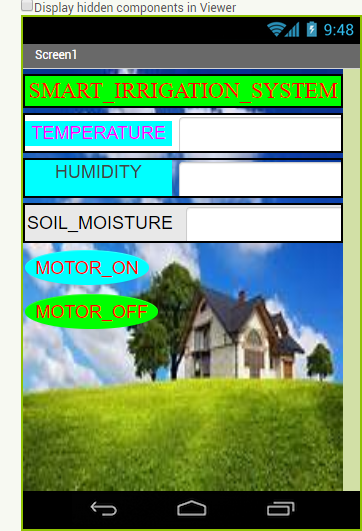
Temperature,humidity,soil moistures readings in serial monitor:



Temperature,humidity and soil moisture readings in the node red:



SMART IRRIGATION SYSTEM APP:



CONCLUSION:

1.THE smart irrigation system is feasible and cost effective for optimizing water resources for agricultural production

2.This irrigation system allows cultivation in places with water scarcity thereby improving

Sustainability

3.It proves that the use of water can be diminished

4.The use of solar power in this system is significantly important for organic crops