**Smart Irrigation System Using Relay And Motor**

A mini-project report submitted for

**Internet of Things (Semester V)**

by

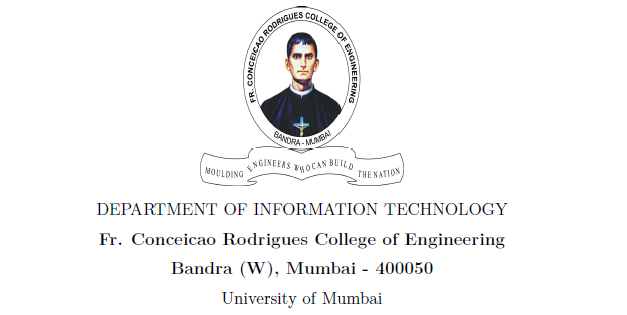
Smith Dabreo (8382)

Brian Dias (8384)

Under the guidance of

Prof. Kranti Wagle

(sign with date)



**Approval Sheet**

**Project Report Approval**

This project report entitled by **Smart Irrigation System Using Relay And Motor**

by **Smith Dabreo and Brian Dias** is approved as mini project in Third year Engineering, Information Technology.

Examiners

1.——————————

2.——————————

Date:

Place:

**Abstract**

With the advancement of automation technology, life is getting simpler and

easier in all aspects. In today’s world Automatic systems are being preferred over

manual system. Automatic system is a growing system of everyday object from

industrial machine to consumer goods that can complete tasks while you are busy

with other activities.

An automated irrigation system for efficient water management has been proposed. In the proposed Irrigation system IoT is implemented, in this system all the information that are received from the sensors and the various parameters are given to the Arduino mega microcontroller as an analog input. A preset value of soil moisture sensor is fixed in microcontroller and also for fencing. When it goes beyond the particular threshold value water is automatically irrigated to the crops and once the required amount of water is fulfilled it stops. The Microcontroller transmits that information on the internet through a network of IoT in the form of wifi module ESP8266-01 that is attached to it. This enhances automated irrigation as the water pump can be switched on or off through information given to the controller. Because this system is used there is less use of man power and also all processes are executed automatically thus improving irrigation.

**Table of Content**

|  |  |  |
| --- | --- | --- |
| **Sr.no** | **Topic** | **Page no** |
| 1. | Introduction | 13 |
| 2. | Problem Definition | 14 |
| 3. | Literature Survey | 15 |
| 4. | Hardware & Software Components of Project | 17 |
| 5. | Project Implementation | 23 |
| 6. | Testing & Debugging of Project | 27 |
| 7. | Results | 28 |
| 8. | Future Scope | 30 |
| 9. | References | 31 |
| 10. | Appendix | 32 |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Lab Outcomes**

LO1: Identify the requirements for the real world problems.

LO2: Conduct a survey of several available literatures in the preferred field of study.

LO3: Study and enhance software/hardware skills.

LO4: Demonstrate and build the project successfully by hardware requirements, coding, emulating and testing.

LO5: To report, present and demonstrate an ability to work in teams and manage the conduct of the research study.

Rubrics for the assessment (LO1):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Indicator | Very Poor | Poor | Average | Good | Excellent |
| Timeline  (2) | More than a 3session late (0) | More than a 2session late (0.5) | More than a 2session late (1) | More than a 1session late (1.5) | Early or on  time (2) |
| Identify real world problems(4) | NA  (0) | Very poor identification of real world problems(1) | Poor identification of real world problems(2) | Good identification of real world problems.(3) | Accurate identification of the real world problems. (4) |
| Design the problem solution  (4) | No requirement analysis is done(0) | Very poor requirement analysis is done(1) | Poor requirement analysis is done(2) | Good requirement analysis is done(3) | Requirement analysis done with best solution design(4) |

Marks:

Rubrics for the assessment (LO2):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Indicator | Very Poor | Poor | Average | Good | Excellent |
| Timeline  (2) | More than a 3session late (0) | More than a 2session late (0.5) | More than a 2session late (1) | More than a 1session late (1.5) | Early or on  time (2) |
| Selection of  Sources(8) | No information is gathered from a range of sources.(0) | Information is gathered from a limited number of sources.(2) | Sources rely heavily on  a small number of sources and are not  considered to be from  authoritative sources(4) | Information is gathered  from a range of sources but  do not entirely reflect the  breadth of the debate(6) | Information is gathered  from a wide range of  journals, books and related  authoritative research  materials.(8) |
| Formatting and  Presentation of  Assignment  (4) | Document contains  man  y errors in  formatting, punctuation  and writing was  incoherent.  (0) | Document contains many  errors in punctuation and  formatting. Referencing  is not consistent with  chosen style guide.  Writing style lacks clarity.(1) | Document contains  few  errors in formatting and  punctuati  on  . Style of  referencing is generally  consistent with chosen  style guide. Writing style is  coherent(2) | Document contains  few  errors in formatting and  punctuati  on  . Style of  referencing is generally  consistent with chosen  style guide. Writing style is  coherent(3) | Document  is professionally  presented  with virtually no  errors  in punctuation  and  is  in the correct format. The  style of referencing is  consistent with chosen  style guide. Writing style is  clear and engaging(4) |

Marks:

LO3: Study and enhance software/hardware skills.

Rubrics for the assessment:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Indicator | Very Poor | Poor | Average | Good | Excellent |
| Timeline  (2) | More than a 3session late (0) | More than a 2session late (0.5) | More than a 2session late (1) | More than a 1session late (1.5) | Early or on  time (2) |
| Installation of Arduino IDE/Raspbian OS(4) Programming | NA(0) | Installation not done(1) | Installation  With some drivers(2) | Installation without drivers(3) | Installation with drivers done(4) |
| Interfacing of sensors to Arduino/Raspberry board | NA(0) | Unable to do connection and but required output not obtained.(1) | Able to do connection and but required output not obtained(2) | Able to do connection and required output is obtained and no libraries are installed(3) | Able to do connection and required output is obtained and libraries are installed(4) |
| Sending data on ThingSpeak ,Analysis of Data | NA(0) | No data sent onthingspeak(1) | Data sent on thingspeak and no analysis done(2) | Data sent onthingspeak and some analysis notdone(3) | Data sent on thingspeak and analysis done(4) |

Marks:

LO4: Demonstrate and build the project successfully by hardware requirements, coding, emulating and testing.

Rubrics for the assessment (LO4):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Indicator | Very Poor | Poor | Average | Good | Excellent |
| Timeline  (2) | More than a 3session late (0) | More than a 2session late (0.5) | More than a 2session late (1) | More than a 1session late (1.5) | Early or on  time (2) |
| Code design(4) | NA  (0) | Very poor code design with no comments and indentation(1) | Poor code design with very comments and indentation  (2) | Design with good coding standards (3) | Accurate design with better coding standards (4) |
| Demo | No system set up was shown(0) | Incomplete System set up was shown.(1) | Partially Complete set up shown with working(2) | Almost Complete set up shown with working(3) | Complete set up shown with working(4) |

Marks :

LO5: To report, present and demonstrate an ability to work in teams and manage the conduct of the research study.

Rubrics for the assessment (L05):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Indicator | Very Poor | Poor | Average | Good | Excellent |
| Timeline  (2) | More than a 3session late (0) | More than a 2session late (0.5) | More than a 2session late (1) | More than a 1session late (1.5) | Early or on  time (2) |
| Teamwork and cooperation(4) | The project appears to have been carried out by only minimal (1-2) members for different tasks. (0) | ) The project appears to have been carried out by only by 2) members | . The project was carried out by most (3-4) members | The project was carried out by most (3-4) members | The project was carried out by all members. |
| Formatting andPresentation of Report  (4) | Document contains  man  y errors in  formatting, punctuation  and writing was  incoherent.  (0) | Document contains many  errors in punctuation and  formatting. Referencing  is not consistent with  chosen style guide.  Writing style lacks clarity.(1) | Document contains  few  errors in formatting and  punctuati  on  . Style of  referencing is generally  consistent with chosen  style guide. Writing style is  coherent(2) | Document contains  few  errors in formatting and  punctuati  on  . Style of  referencing is generally  consistent with chosen  style guide. Writing style is  coherent(3) | Document  is professionally  presented  with virtually no  errors  in punctuation  and  is  in the correct format. The  style of referencing is  consistent with chosen  style guide. Writing style is  clear and engaging(4) |

Marks**:**

**Chapter 1**

**Introduction**

Despite of emphasis on industrialization, agriculture remains primary segment of Indian economy both regarding commitment to the gross domestic product (GDP) and in addition a wellspring of job to millions the nation over. Increase in energy crisis, low availability of fresh water for irrigation and uneven environmental conditions are the major factors in agriculture. So, we need some innovation in Indian agriculture (i.e. modern tools and technologies) to improve the production efficiency, quality of yield India has to become smart and start learning some things from foreign countries and start implementing smart tools in agriculture like the one we have showcased here

In our project we take advantage of IOT applications and Arduino microcontroller, by programming the Arduino we are able to sense the soil moisture of the soil and the microcontroller acts accordingly and irrigates the crops if needed, particular conditions have been set in the Arduino so it can turn on or off the motor depending upon the soil moisture also because of IOT we are able to read the soil moisture from our phones/computers at the comfort of our homes and can control the motor wirelessly from our phones. We can even save the data for future research in a excel sheet or can store it in the form of graph.

**Chapter2**

**Problem Definition**

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 lead to the problem of water scarcity. In response to this problem we propose installation of a smart device which can solve this by automatically supplying the required amount of water to the crops and then stop and supply again when needed this can lead to less wastage of water. We propose a prototype of this smart device in this project.

**Chapter3**

**Literature Survey**

The evolution of information technology has opened door to many impossibilities. Over years, our cell phones, tablets, automobiles, the rise of "smart" technology have consumed the market and have become the new standard in the industries. Smart irrigation is one such technology which have attracted interest of many researchers and is evolving and improving from about a decade. This smart irrigation industry where water waste is minimised and is no longer sustainable socially, economically and conventionally as well [1]. Today, the farmers are using traditional approaches to monitor their agricultural environment and to irrigate their sugarcane crop. The following are some of the irrigation types that have been used by the farmers [2]. Irrigation can be defined as simulated use of water to the crop field. The watering system is critical on the grounds that, this can just guarantee the survival of the crops. In the event that the field is watered intensely with water, there are risks that the plant may die because of excessive watering. The water could likewise wash them away during strong force watering. Then again, if there is inadequate water, then additionally there are risks that the plant may die due to starvation.

For continuously increasing demand and decrease in supply of food necessities, it’s important to rapid improvement in production of food technology. Agriculture is only the source to provide this. This is the important factor in human societies to growing and dynamic demand in food production. Agriculture plays the important role in the economy and development, like India. Due to lack of water and scarcity of land water result the decreasing volume of water on earth, the farmer use irrigation. Irrigation may be defined as the science of artificial application of water to the land or soil that means depending on the soil type, plants are to be provided with water [3].

Various researches have been carried out on how soil irrigation can be made more efficient. The researchers have used different ideas depending on the condition of the soil and quantity of water Different technologies used and the design of the system was discussed by the researchers. This paper aims at reducing the wastage of water and the labor that is used to carry out irrigation manually. The proposed system aims at detecting the moisture content of the soil using sensors that are placed directly into the soil. These sensors sense the water level of the soil and if the water level is not adequate then the user will be notified through a message that will be sent to the application which would be installed on the user’s mobile phone. The Arduino board, a microcontroller, controls the digital connection and interaction between objects in the proposed system, enabling the objects to sense and act. Also, with its powerful on-board processing, various sensors and other application specific devices can be integrated to it. In the system, sensors detect the water and moisture level and send readings to a fixed access point, such as a personal computer, which in turn can access irrigation modules installed in the field or the physical module in the water tank, wirelessly over the internet. A wireless application of drip irrigation automation supported by soil moisture sensors Irrigation by help of freshwater resources in agricultural areas has a crucial importance. Traditional instrumentation based on discrete and wired solutions, presents many difficulties on measuring and control systems especially over the large geographical areas. If different kinds of sensors (i.e. humidity, and etc.) are involved in such irrigation in future works, it can be said that an internet based remote control of irrigation automation will be possible. An automated irrigation system was developed to optimize water use for agricultural crops. The system has a distributed wireless network of soil-moisture placed in the root zone of the plants. In addition, a gateway unit handles sensor information, triggers actuators, and transmits data to an android application.

**Chapter 4**

Hardware and Software Components used in Project: -

* Arduino Mega Microcontroller
* Soil Moisture Sensor
* ESP8266-01 WiFi Module
* 5V Relay Module
* 3-6V DC Motor
* Breadboard
* Jumper Wires
* Pipe

Arduino Mega Microcontroller: Arduino is an open-source physical computing platform based on a simple i/o board and a development environment that implements the processing/wiring language. Arduino can be used to develop stand-alone interactive objects or can be connected to software on your computer (e.g. Flash, Processing, MaxMSP). The open-source IDE can be downloaded for free (currently for Mac OS X, Windows, and Linux).

The Arduino Mega is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started

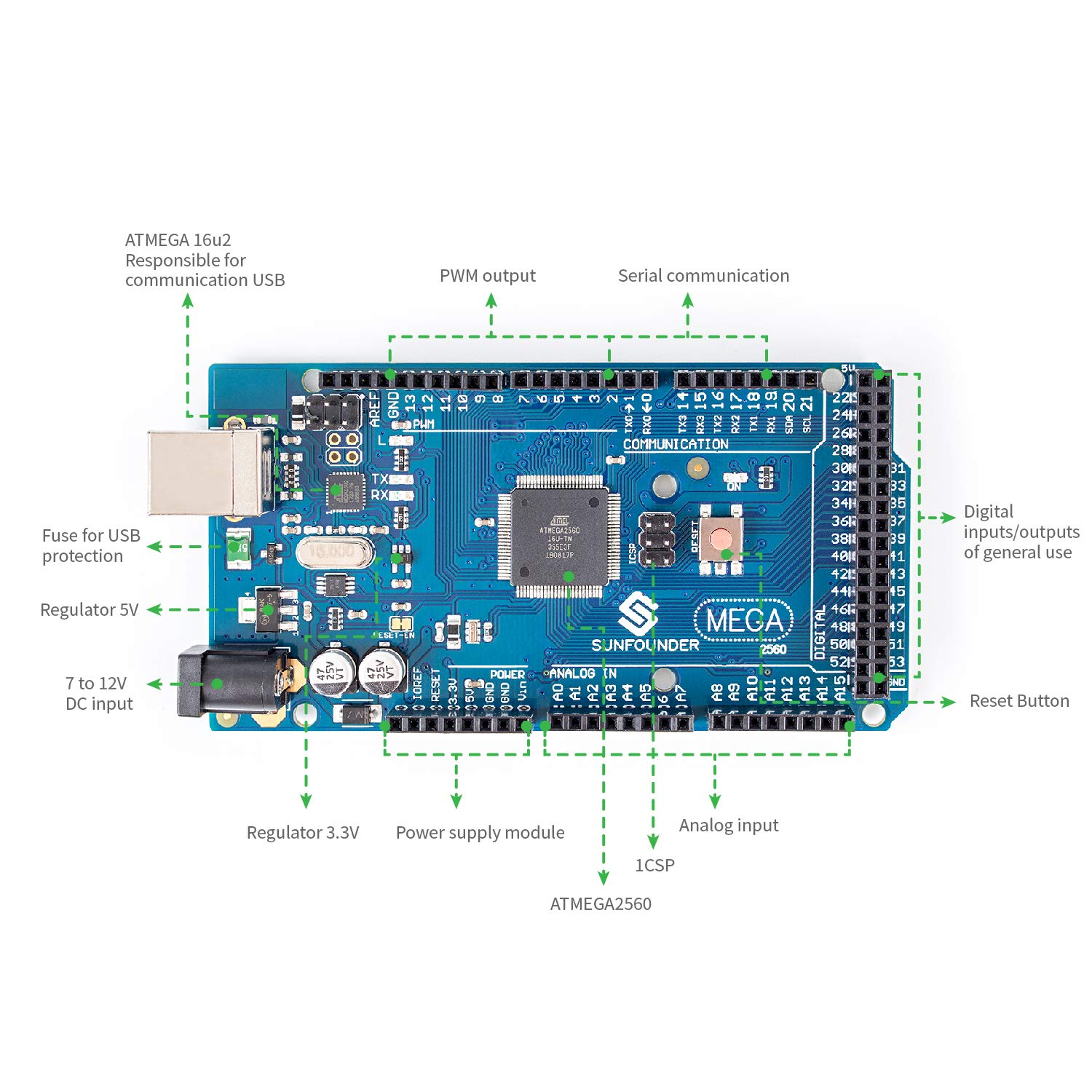


Fig 4.1

Soil Moisture Sensor: Soil moisture sensors measure the volumetric [water content](https://en.wikipedia.org/wiki/Water_content) in [soil](https://en.wikipedia.org/wiki/Soil). Since the direct [gravimetric measurement](https://en.wikipedia.org/wiki/Gravimetric_analysis) of free soil moisture requires removing, drying, and weighing of a sample, soil moisture sensors measure the volumetric water content indirectly by using some other property of the soil, such as electrical resistance, dielectric constant, or interaction with [neutrons](https://en.wikipedia.org/wiki/Neutron), as a proxy for the moisture content.

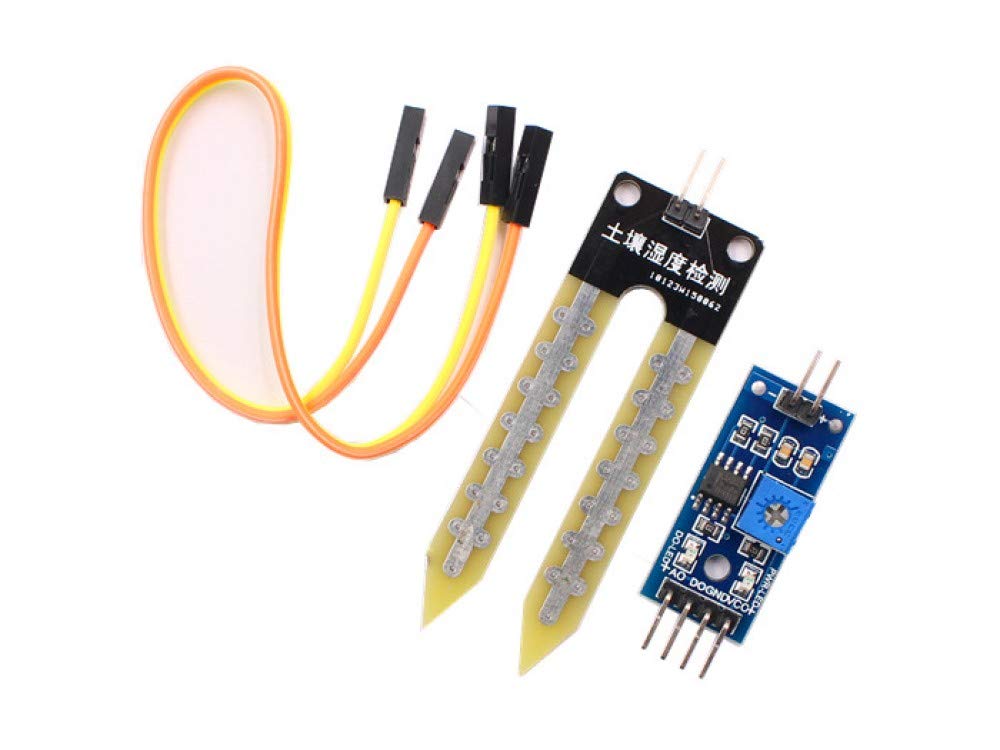


Fig 4.2

ESP8266-01 WiFi Module: The ESP8266 is a low-cost [Wi-Fi](https://en.wikipedia.org/wiki/Wi-Fi) microchip with full [TCP/IP stack](https://en.wikipedia.org/wiki/TCP/IP_stack) and [microcontroller](https://en.wikipedia.org/wiki/Microcontroller) capability produced by manufacturer [Espressif Systems](https://en.wikipedia.org/w/index.php?title=Espressif_Systems&action=edit&redlink=1) in Shanghai, China. The chip first came to the attention of Western [makers](https://en.wikipedia.org/wiki/Maker_culture) in August 2014 with the ESP-01 module, made by a third-party manufacturer Ai-Thinker. This small module allows microcontrollers to connect to a Wi-Fi network and make simple TCP/IP connections using [Hayes](https://en.wikipedia.org/wiki/Hayes_command_set)-style commands.

The ESP8285 is an ESP8266 with 1 MiB of built-in flash, allowing for single-chip devices capable of connecting to Wi-Fi.

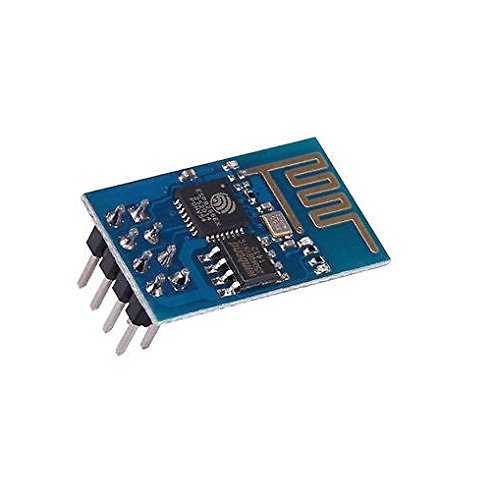


Fig 4.3

5V Relay Module: A relay is an [electrically](https://en.wikipedia.org/wiki/Electric) operated [switch](https://en.wikipedia.org/wiki/Switch). It consists of a set of input terminals for a single or multiple control signals, and a set of operating contact terminals. The switch may have any number of contacts in multiple [contact forms](https://en.wikipedia.org/wiki/Electrical_contact#Contact_form), such as make contacts, break contacts, or combinations thereof. Relays are used where it is necessary to control a circuit by an independent low-power signal, or where several circuits must be controlled by one signal

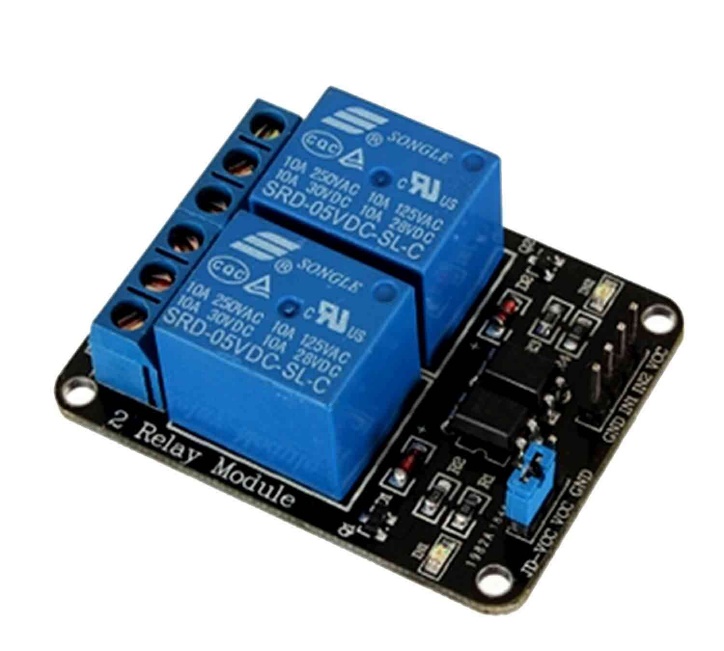


Fig 4.4

3-6V DC Motor: A DC motor is any of a class of rotary electrical machines that converts direct current electrical energy into mechanical energy. The most common types rely on the forces produced by magnetic fields. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic, to periodically change the direction of current flow in part of the motor.



Fig 4.5

Breadboard: A breadboard is a solderless device for temporary prototype with electronics and test circuit designs. Most electronic components in electronic circuits can be interconnected by inserting their leads or terminals into the holes and then making connections through wires where appropriate. The breadboard has strips of metal underneath the board and connect the holes on the top of the board. The metal strips are laid out as shown below. Note that the top and bottom rows of holes are connected horizontally and split in the middle while the remaining holes are connected vertically.



Fig 4.6

Jumper Wires: A jump wire (also known as jumper wire, or jumper) is an [electrical wire](https://en.wikipedia.org/wiki/Electrical_wire), or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a [breadboard](https://en.wikipedia.org/wiki/Breadboard) or other prototype or test circuit, internally or with other equipment or components, without soldering. Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the [header connector](https://en.wikipedia.org/wiki/Pin_header#Header_connector) of a circuit board, or a piece of test equipment.



Fig 4.7

Pipe: A pipe is a tubular section or hollow [cylinder](https://en.wikipedia.org/wiki/Cylinder_(geometry)), usually but not necessarily of [circular](https://en.wikipedia.org/wiki/Circle) [cross-section](https://en.wikipedia.org/wiki/Cross_section_(geometry)), used mainly to convey substances which can flow.

****

Fig 4.8

**Chapter 5**

**Project Implementation**

We have used a simple Arduino based connection and circuit. The Arduino and the Breadboard are the main connectors of this circuit. The breadboard is used for connecting multiple wires to a single source as the project demands it and the Arduino is the main microcontroller so everything should be connected to it for the whole project to function successfully.

**Arduino interfaced with soil sensor and wifi module**

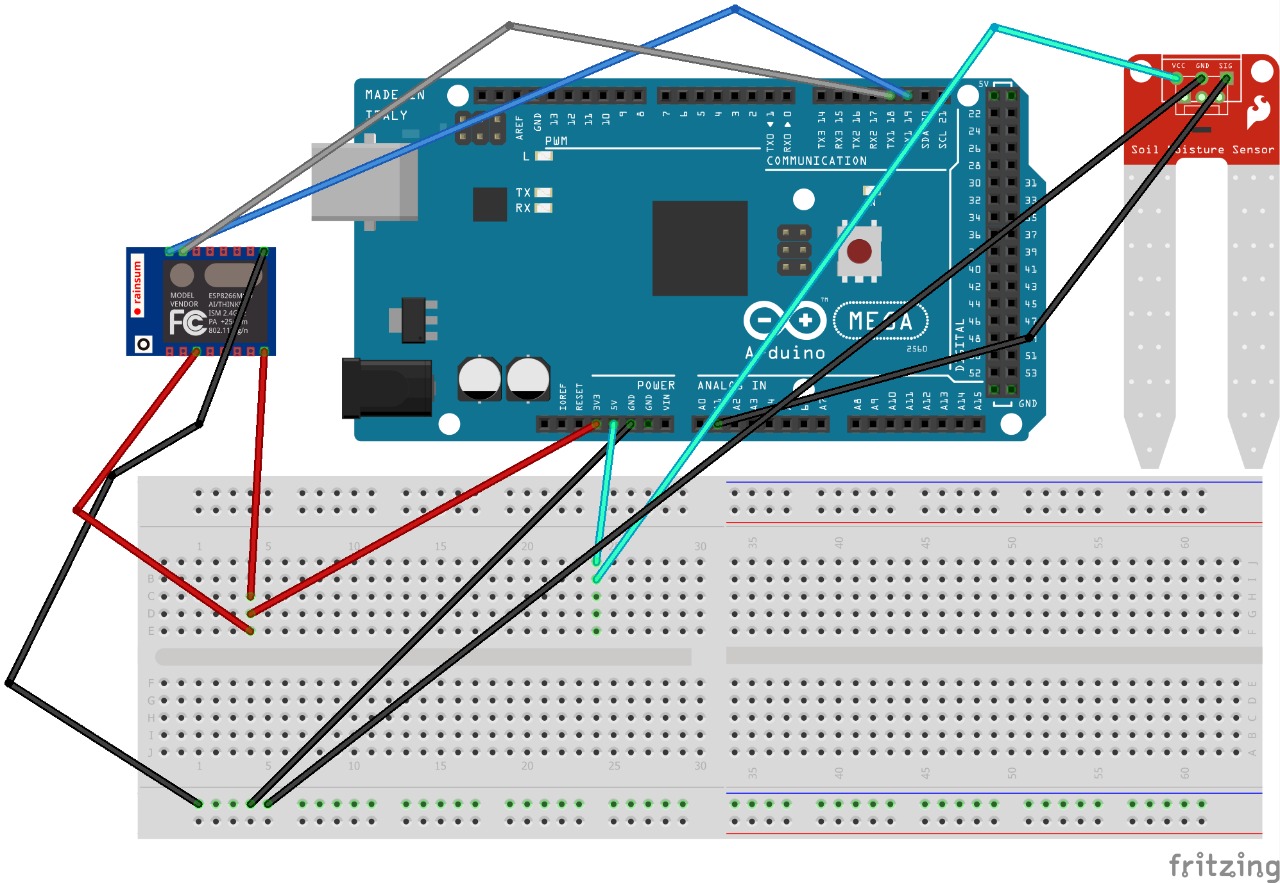


Fig 5.1

**Circuit Diagram**

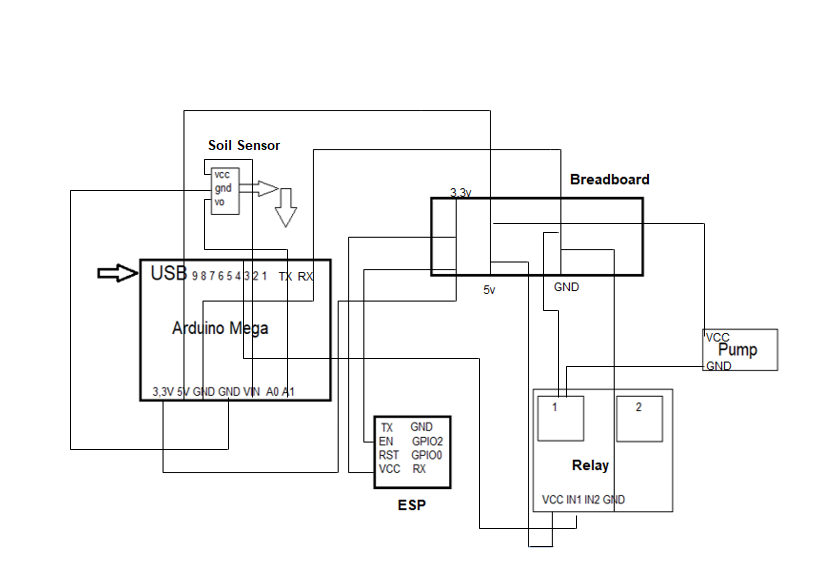
****

Fig 5.2

**Connections:**

1. Soil Moisture Sensor:

The sensor has 3 pins consisting of Ground, VCC and output pin (this pin send the readings to Arduino). The vcc pin is connected to Vin in Arduino which provides 5V Voltage, the Ground is connected to GND

And the output pin is connected to A1 pin of the Arduino.

1. Breadboard:

The Breadboard is connected with 5V and 3.3V power supply coming from Arduino and also connected with ground coming from Arduino

1. ESP8266 Module:

It consists of 8 pins where only 5 of them are required for our project namely TX, RX, GND, VCC and Enable pin. GND of the ESP is connected to ground at the breadboard using jumper wires. VCC and Enable is connected to 3.3V. TX and RX of ESP is connected to TX1 and RX1 of the Arduino Respectively.

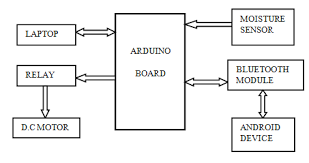
1. DC Motor:

One wire of this 5V motor is connected to NO1 of the Relay and the other wire connected to 5V at the breadboard using jumper wires

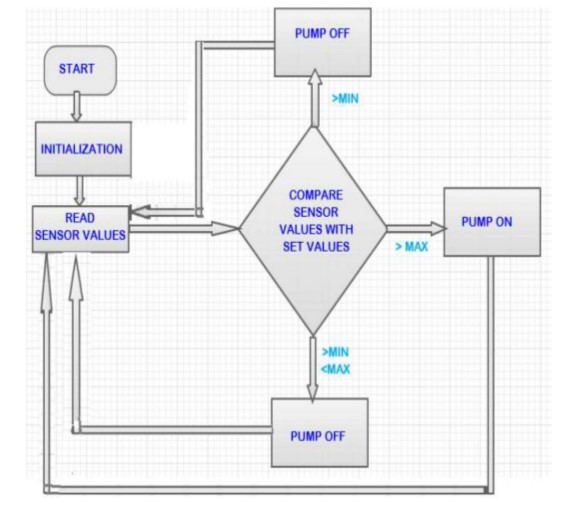
1. Relay:

The VCC of Relay is connected to the 5V at the breadboard and GND is connected to Ground at the breadboard using jumper wires. Also, the IN1(Input) pin is connected to pin 3 of the Arduino. Also, COM1 is connected to the Ground at the Breadboard

**Working:**



* **Flowchart:**

****

**Chapter 6**

**Testing&Debugging**

Here, explain how you have tested the different modules and problems faced while connection of different modules and how they were removed.

Testing a project is one of the most important part of any component-based project. The same goes with our project. Many things were tested and interfaced for better functionality of the project.

**Tested Components:**

* Soil Moisture Sensor:

For testing the sensor only, the sensor was interfaced with the Arduino board and a single program for displaying soil moisture on serial monitor was executed and it ran perfectly.

* Temperature and Humidity Sensor:

This sensor was also interfaced with the Arduino just like the soil moisture sensor and a particular code for temperature and humidity was executed but because of some external fault in the sensor the sensor was removed from the project

* WiFi Module:

The module was tested by trying to send data coming from the already interfaced soil moisture sensor to the Thingspeak server and after many trial and error in the wifi code the module ran perfectly and send the data to the Thinkspeak cloud. But because of slow data transfer Thingspeak server was dropped and the newly released Blynk android app was used and due to its easy UI and instructions provided in the app the module was connected and it was up and running in no time.

* Relay and Motor:

Relay and motor were tested by connecting them to the Arduino and breadboard as given in the diagram. And by adding a small condition code in the existing Arduino code the both components ran perfectly. Although the connections were a bit complicated and time consuming.

**Chapter 7**

**Results**

This project was tested multiple time between the duration of the project implementation. Various things were tested some were included some were removed and results were generated. From all that testing following are results:

* Thingspeak Output:

After the interfacing of the ESP WiFi Module the Thingspeak server was used to receive and display the readings on the cloud so the user can see it anywhere he/she wants by logging in to the server the following results were generated.

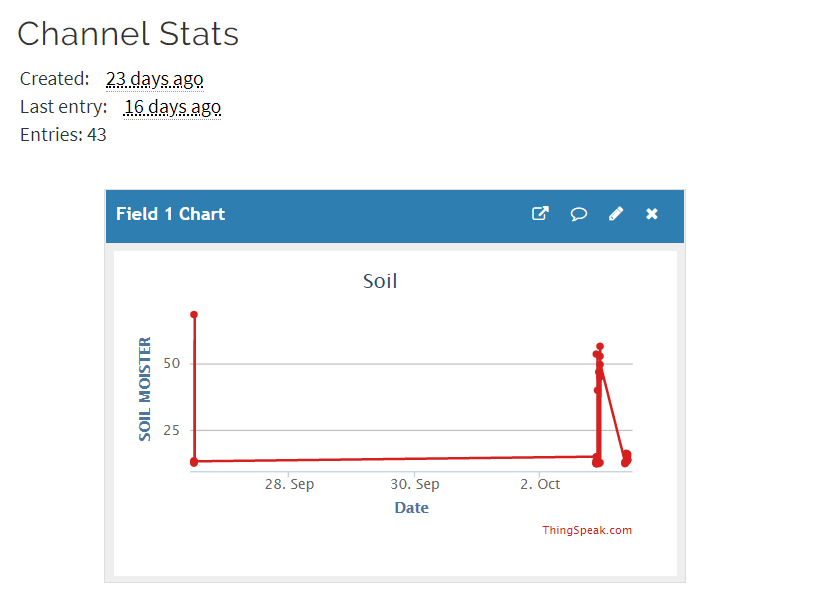


Fig 7.1

But because of the slow working of the Thingspeak server and there was no function to control the motor and relay the Thingspeak idea dropped.

* Blynk:

Blynk is capable of displaying the sensor reading converting the same readings into a graph and is also able to control the Relay and Motor it was used in the project as the cloud server

The results generated after using Blynk app are the following:

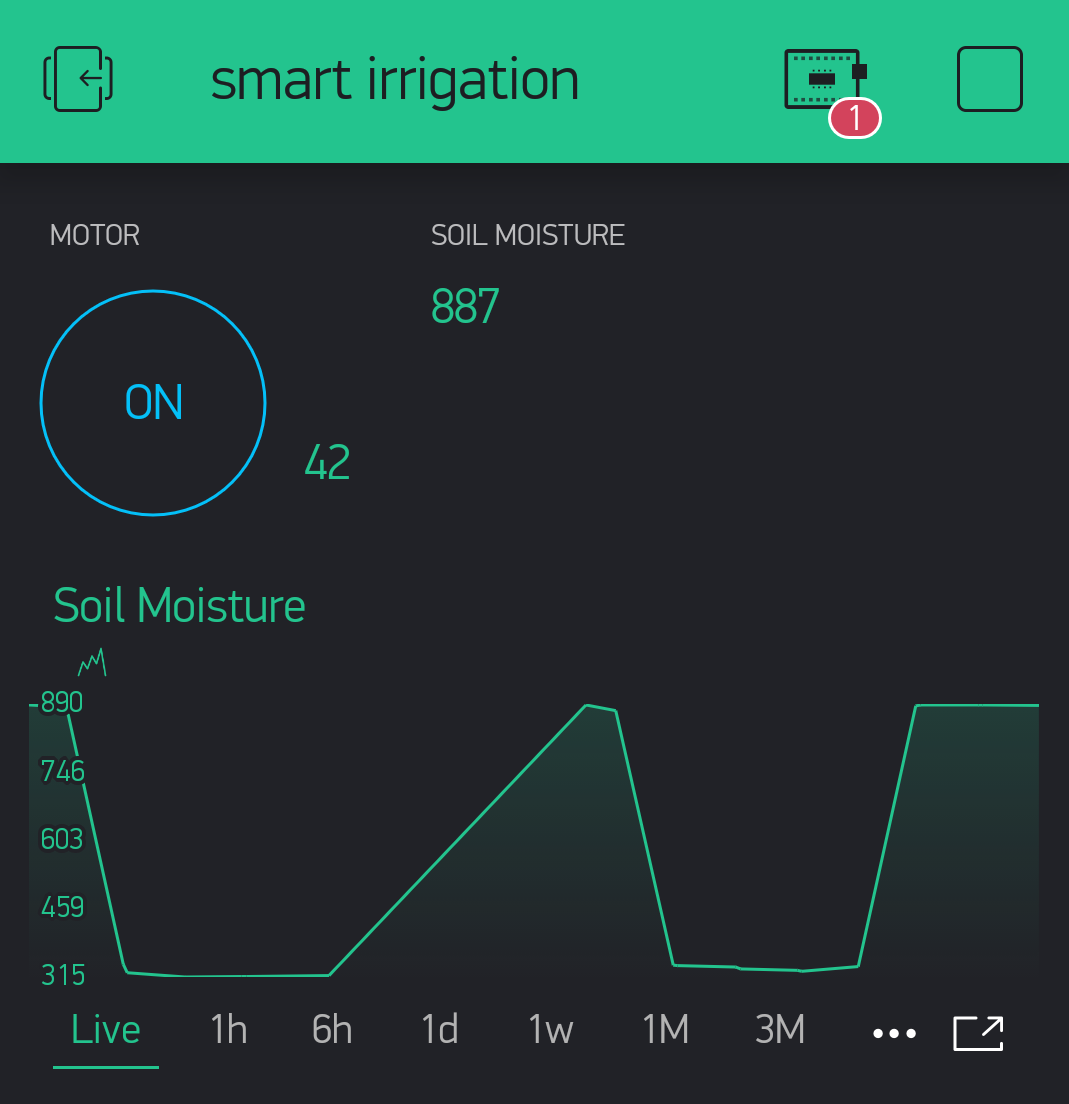


Fig 7.2

**Chapter 8**

**Futurescope**

In Future, we can advance to smart irrigation pump into a smart crop watering station. We can add more sensors like temperature sensor, humidity sensor, etc for more accurate results and watering system. Also, we can interface LCD screen in order to display the current status of the soil moisture content levels, percentage of water utilized to water the plant, duration of time for which the water pump is ON, etc. We could use the Blynk app in such way that the user will be notified with updates like moisture is low or the water pump is turned on, etc. this small-scale project could be converted into a big system for smart irrigating hundreds of crops, saving tons of water. The idea of using IOT for irrigation can be extended further to other activities in farming such as cattle management, fire detection and climate control. This would minimize human intervention in farming activities.

**Chapter 9**

**Reference**

1. **P. Singh and S. Saikia, “Arduino-based smart irrigation using water flow sensor, soil moisture sensor, temperature sensor and ESP8266 WiFi module,” 2016**
2. **S. Attar, S. Sudhakar “Real-Time Monitoring Of Agricultural Activities Using Wireless Sensor Network,” 2017**
3. **Souparno Sarkar, Saikat Das “automatic irrigation system,” 2018**

**Appendix**

**Program code:**

#define BLYNK\_PRINT Serial

#include <ESP8266\_Lib.h>

#include <BlynkSimpleShieldEsp8266.h>

const int sensor\_pin = A1; /\* Soil moisture sensor O/P pin \*/

int water; //random variable

char auth[] = "Nnkd3OYcZQPAGlqAMT20uDOnhNkTsocD";

char ssid[] = "iPhone";

char pass[] = "123456789";

#define EspSerial Serial1

#define ESP8266\_BAUD 115200

ESP8266 wifi(&EspSerial);

BlynkTimer timer;

// This function sends Arduino's up time every second to Virtual Pin (5).

// In the app, Widget's reading frequency should be set to PUSH. This means

// that you define how often to send data to Blynk App.

void myTimerEvent()

{

// You can send any value at any time.

// Please don't send more that 10 values per second.

Blynk.virtualWrite(V5, millis() / 1000);

}

void setup()

{

pinMode(3,OUTPUT); //output pin for relay board, this will sent signal to the relay

pinMode(A1,INPUT); //input pin coming from soil sensor

// Debug console

Serial.begin(9600);

// Set ESP8266 baud rate

EspSerial.begin(ESP8266\_BAUD);

delay(10);

Blynk.begin(auth, wifi, ssid, pass);

// You can also specify server:

//Blynk.begin(auth, wifi, ssid, pass, "blynk-cloud.com", 80);

//Blynk.begin(auth, wifi, ssid, pass, IPAddress(192,168,1,100), 8080);

// Setup a function to be called every second

timer.setInterval(1000L, myTimerEvent);

}

void loop()

{

float moisture\_percentage;

int sensor\_analog;

sensor\_analog = analogRead(sensor\_pin);

moisture\_percentage = ( 100 - ( (sensor\_analog/1023.00) \* 100 ) );

Serial.print("Moisture Percentage = ");

Serial.print(moisture\_percentage);

Serial.print("%\n\n");

delay(1000);

water = digitalRead(A1); // reading the coming signal from the soil sensor

if(water == HIGH) // if water level is full then cut the relay

{

digitalWrite(3,LOW); // low is to cut the relay

}

else

{

digitalWrite(3,HIGH); //high to continue proving signal and water supply

}

delay(400);

Blynk.run();

timer.run(); // Initiates BlynkTimer

}