Smart Irrigation: An IOT Based Approach

Pranav Dixit1

Prateek Agarwal1

**1Undergraduate Students, Department of Computer Science, Swami Keshvanand Institute of Technology.**

1dprnav@gmail.com

2prateekagarwal947@gmail.com

**Abstract: -** This Paper describes the purpose, use and method for the Smart Irrigation System, abbreviated SIS. SIS is an intelligent agriculture irrigation system that collects soil moisture, temperature, humidity, and geographical readings from in ground sensors and analyzes the data which determine the necessary amount of water needed for the field. The purpose of this project is to make Smart Irrigation Systems easier and intelligent by providing farmers with an automated watering solution that helps with water conservation and provides remote access through an easy to use mobile application and web application.India has 18% of world’s total population, having 4% of world's total fresh water, out of which 80% of it is used in agriculture purposes.

India receives an average of 4,000 billion cubic meters of precipitation every year. However, only 48% of it is used in India's surface and groundwater bodies. Therefore, there is a high need of a smart irrigation system which is better than the traditional method of irrigation.

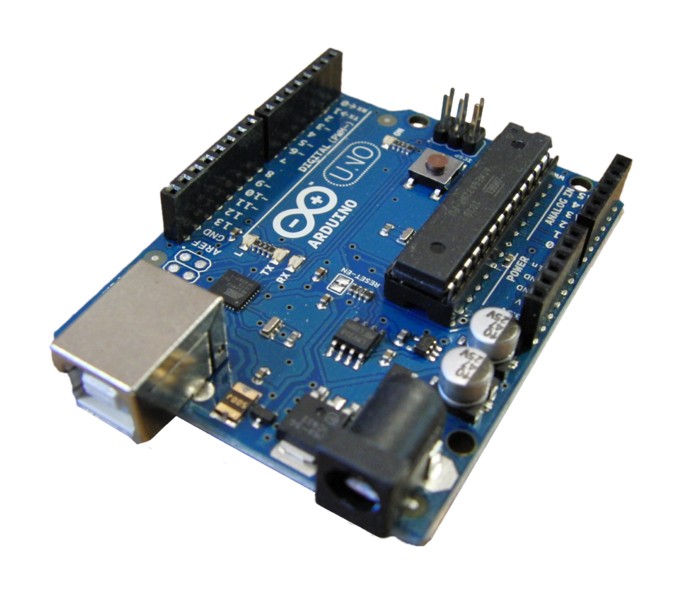
**Introduction: -**

The term “IoT” stands for the internet of things, can be defined as the interconnection between the individually identifiable embedded computing apparatus in the accessible internet infrastructure. The ‘IoT’ connects various devices and transportations with a help of internet as well as electronic sensors. Please refer to this link to know more about Experts Opinion on Application of Internet of Things (IoT) in Future.

The Arduino UNO is one of the most used microcontrollers in the industry. It is very easy to handle, convenient, and use. The coding of this microcontroller is very simple. The program of this microcontroller is considered as unstable due to the flash memory technology. The applications of this microcontroller involve a wide range of applications like security, home appliances, remote sensors, and industrial automation. This microcontroller has the ability to be joined on the internet and perform as a server too. This is capable of automating the irrigation process by analyzing the moisture of soil and the climate condition (like raining).

**Hardware Requirements:**

1. **Arduino UNO R3**



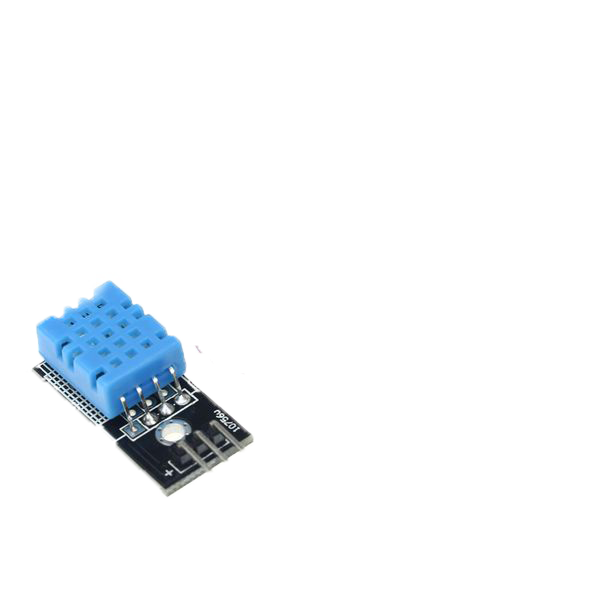
*Fig -01: Arduino Uno R3*

Arduino Uno is a board based microcontroller which contain 14 digital input/output pins out of which 6 can be used as PWM outputs, 6 analog inputs, a 16 MHz quartz crystal, 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 an AC-to-DC adapter or battery to get started.

The Arduino UNO is based on ATmega 328p microcontroller and it also has ATmega16U microcontroller.

* ***ATmega 328p:***It is the brain of the Arduino and it is a high performance Atmel pico power 8bit AVR RISC based microcontroller which is cable of executing powerful instruction in single clock cycle.
* ***ATmega 16U2:***This microcontroller takes care of the USB connection and ICSP bootloader.

1. **DHT 11**



*Fig -02: DHT11*

DHT11 is a digital sensor to measure temperature and humidity sensor. It gives a calibrated digital signal output of the temperature and humidity. Application of the DHT11 sensor is collectively used in temperature and humidity sensing technology, to ensure that the product has high reliability and excellent long-term stability. The sensor includes a resistive sense of wet components and an NTC temperature measurement device, and connected with a high-performance 8-bit microcontroller.

**Sample Code of DHT11**

#include "dht.h" //dht.h file should be included with this code

#define dht\_apin A1 // Analog Pin sensor is connected to A1

dht DHT;

void setup()

{

Serial.begin(9600);

}

void loop()

{

DHT.read11(dht\_apin);

Serial.print("Temp: ");

Serial.println(DHT.temperature);

Serial.print("Humidity: ");

Serial.println(DHT.humidity);

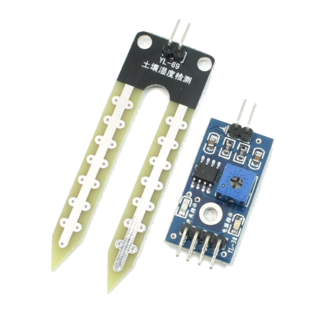
Serial.println("");

delay(1000);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Soil Sensor**



*Fig -03: Soil Sensor and Potentiometer*

Soil Moisture Sensor is a device which is used to measure the moisture in soil and similar materials. The soil moisture is the essential part of the SIS. The two large exposed pads function as probes for the sensor, they together act as a variable resistor.

Insert the sensor in the soil and we can measure Soil moisture or water content level of the soil. A calibration of analog output is needed as the 0 represent that there is no moisture in the soil and a specific maximum value is calibrated in this project we will calibrate to 100 in this paper.

**Sample Code for Soil Sensor**

float moistValue =0;

const int soil\_sensor = A0;

void setup()

{

Serial.begin(9600);

}

void loop()

{

/\* The water sensor will switch LOW when water is detected.

Get the Arduino to illuminate the LED and activate the buzzer

when water is detected, and switch both off when no water is present \*/

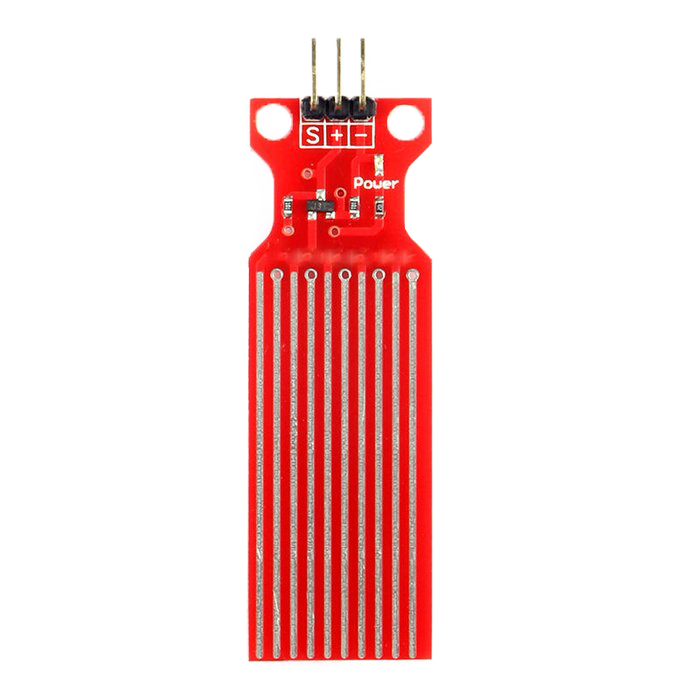
moistValue = 100 - (analogRead(soil\_sensor)/1023.00 \* 100);

Serial.println(moistValue);

delay(1000);

}

1. **Water sensor (funduino):**



*Fig -04: Water Sensor*

It can be used to detect the presence or absence of water, accurately gauge the water surface level, or you can even accurately gauge the volume of water present by using a volume measurement device such as a measuring cup in conjunction with the analog water sensor module. This makes it the Funduino analog water sensor a versatile component.

**Sample Code for Water Sensor**

#define Grove\_Water\_Sensor A2 // Attach Water sensor to Arduino Digital Pin 2

void setup()

{

Serial.begin(9600);

}

void loop()

{

Serial.println(analogRead(Grove\_Water\_Sensor));

delay (1000);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Relay (220V):**

*Fig -05: Relay*

Relay is an electrical switch using electrical signal to turn on/off the current. It is also used to control a circuit by a low-power signal.

On its body, you can see these characters and numbers:

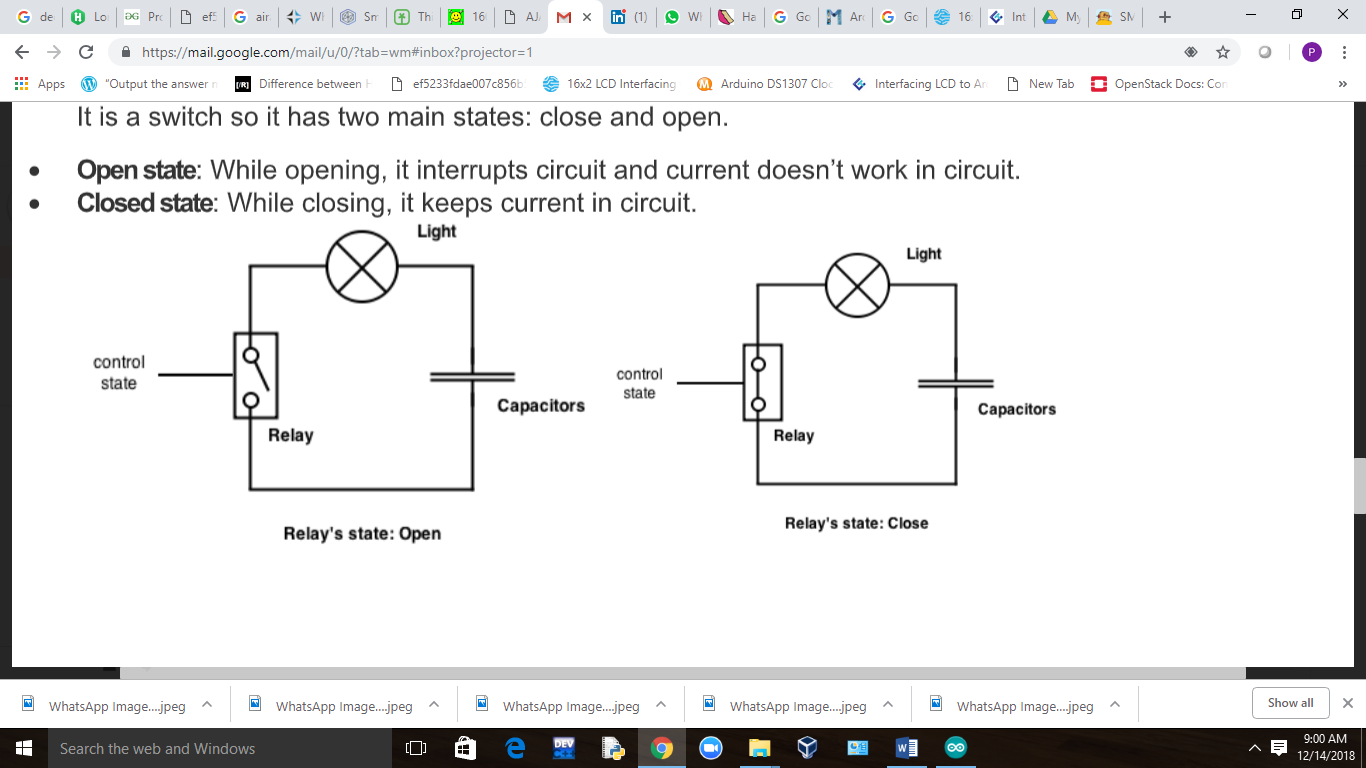
* 10A 250VAC.
* 10A 125VAC.
* 10A 30VDC.
* 10A 28VDC.
* SRD-05VDC-SL-C.

This is what they mean:

* 10A 250VAC: The max current intensity of connection, where voltage is equal or less than 250V (AC), is 10A.
* SRD-05VDC-SL-C: Voltage of signal to control circuit is 5V.
* State of relay

It is a switch so it has two main states: close and open.

* Open state: While opening, it interrupts circuit and current doesn’t work in circuit.
* Closed state: While closing, it keeps current in circuit.



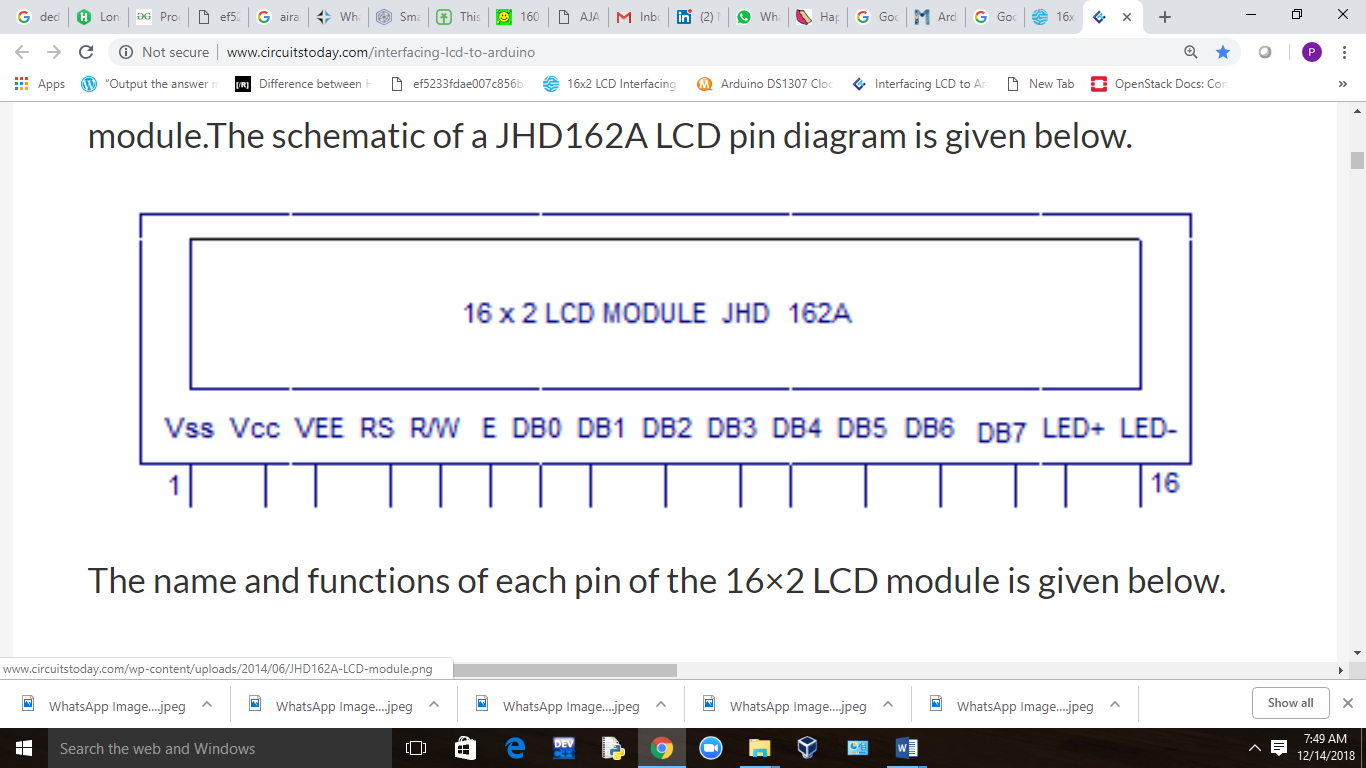
*Fig -06: Open and Closed state of Relay*

1. **LCD (16\*2):**



*Fig -07: A 16\*2 LCD*

To establish a good communication between human world and machine world, display units play an important role. And so they are an important part of embedded systems. Display units - big or small, work on the same basic principle. The 16x1 display unit will have 16 characters and are in one line. The 16x2 LCD will have 32 characters in total 16in 1st line and another 16 in 2nd line. Here one must understand that in each character there are 5x10=50 pixels so to display one character all 50 pixels must work together.



*Fig -08: Pin Diagram of 16\*2 LCD Module*

The connections which are done for LCD are given below:

PIN1 or VSS to ground

PIN2 or VDD or VCC to +5v power

PIN3 or VEE to 10K Potentiometer: Contrast adjustment pin. This is done by connecting the ends of a 10K potentiometer to +5V and ground and then connecting the slider pin to the VEE pin. The voltage at the VEE pin defines the contrast. The normal setting is between 0.4 and 0.9V.

PIN4 or RS (Register Selection) to PIN0 of ARDUINO UNO

PIN5 or RW (Read/Write) to ground (puts LCD in read mode eases the communication for user)

PIN6 or E (Enable) to PIN1 of ARDUINO UNO

PIN11 or D4 to PIN8 of ARDUINO UNO

PIN12 or D5 to PIN9 of ARDUINO UNO

PIN13 or D6 to PIN10 of ARDUINO UNO

PIN14 or D7 to PIN11 of ARDUINO UNO

PIN15: Anode of the back light LCD. When operated on 5V, a 560 ohm resistor should be connected in series to this pin. In Arduino based projects the back light LCD can be powered from the 3.3V source on the Arduino board.

PIN16 Cathode of the back light LCD.

**Sample Code for LCD Display**

// include the library code:

#include <LiquidCrystal.h>

// initialize the library by associating any needed LCD interface pin

// with the arduino pin number it is connected to

const int rs = 0, en = 1, d4 = 8, d5 = 9, d6 = 10, d7 = 11;

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup()

{

// set up the LCD's number of columns and rows:

lcd.begin(16, 2);

// Print a message to the LCD.

lcd.print("Hello World!");

}

void loop()

{

// set the cursor to column 0, line 1

// (note: line 1 is the second row, since counting begins with 0):

lcd.setCursor(0, 1);

// print the number of seconds since reset:

lcd.print(millis() / 1000);

}

1. **RTC IC – DS1307:**

The DS1307 real time clock (RTC) IC is an 8 pin device using an I2C interface. The DS1307 is a low-power clock/calendar with 56 bytes of battery backup SRAM. The clock/calendar provides seconds, minutes, hours, day, date, month and year qualified data. The end date of each month is automatically adjusted, especially for months with less than 31 days. They are available as integrated circuits (ICs) and supervise timing like a clock and also operate date like a calendar. The main advantage of RTC is that they have an arrangement of battery backup which keeps the clock/calendar running even if there is power failure. An exceptionally little current is required for keeping the RTC animated. We can find these RTCs in many applications like embedded systems and computer mother boards, etc.

**Sample Code for RTC IC – DS1307**

// Date and time functions using a DS1307 RTC connected via I2C and Wire lib

#include <Wire.h>

#include <LiquidCrystal.h>

#include "RTClib.h"

RTC\_DS1307 rtc;

LiquidCrystal lcd(0,1,8,9,10,11); // (rs, e, d4, d5, d6, d7)

char daysOfTheWeek[7][12] = {"Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"};

void setup ()

{

Serial.begin(9600);

lcd.begin(16, 2);

Serial.print("Hello");

if (! rtc.begin())

{

Serial.print("Couldn't find RTC");

while (1);

}

if (! rtc.isrunning())

{

Serial.print("RTC is NOT running!");

}

rtc.adjust(DateTime(F(\_\_DATE\_\_), F(\_\_TIME\_\_)));//auto update from computer time

//rtc.adjust(DateTime(2018, 1, 21, 3, 0, 0));// to set the time manualy

}

void loop ()

{

DateTime now = rtc.now();

lcd.setCursor(0, 1);

Serial.print(now.hour());

Serial.print(':');

Serial.print(now.minute());

Serial.print(':');

Serial.print(now.second());

Serial.print(" ");

lcd.setCursor(0, 0);

Serial.print(daysOfTheWeek[now.dayOfTheWeek()]);

Serial.print(" ,");

Serial.print(now.day());

Serial.print('/');

Serial.print(now.month());

Serial.print('/');

Serial.println(now.year());

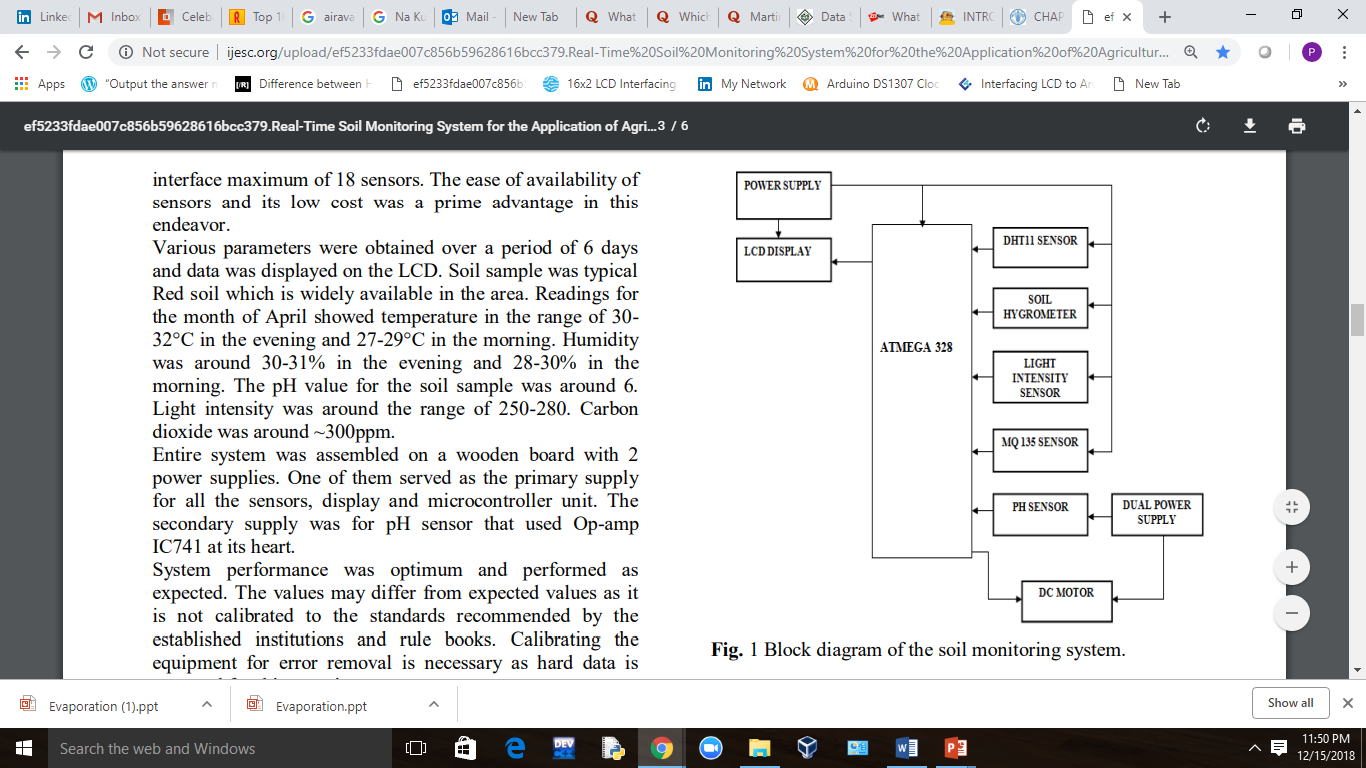
delay(2000);

}

1. **GSM Module:**

A **GSM Module** is basically a GSM Modem (like SIM 900) connected to a PCB with different types of output taken from the board – say TTL Output (for Arduino, 8051 and other microcontrollers) and RS232 Output to interface directly with a PC (personal computer). The board will also have pins or provisions to attach mic and speaker, to take out +5V or other values of power and ground connections. These type of provisions vary with different modules.

Lots of varieties of GSM modem and GSM Modules are available in the market to choose from. For our project of connecting a gsm modem or module to arduino and hence send and receive sms using arduino – it’s always good to choose an***arduino compatible GSM Module*** – that is a GSM module with TTL Output provisions.



*Fig -09: Block Diagram of Smart Irrigation System*

**Conclusion:**

Beside water amount can be reduced by the developed system due to preventing unnecessary irrigation, highest yield can be taken in agriculture preventing insufficient irrigation. In addition, that the irrigation system can be controlled by a smartphone makes it usable in these days smartphone usage rate are very high. Both the irrigation system can be automatically worked according to the values sent by the sensors and it can be manually worked or stopped on an Android smartphone. All the values are obtained by the sensors and then the amount of water by a particular crop is needed and then the motor is switched on until the need of water for the particular crop is not fulfilled. All the information is sent to the farmer by the SMS. Everything is automated in this project also this is very helpful for the farmers as this will reduce the man power in the field also this will conserve the water with a significant amount.

**References:**

1. The 8051 Microcontroller and Embedded Systems: Using Assembly and C by Mazidi.
2. PIC Microcontroller and Embedded Systems: Using assembly and C for PIC 18 by Mazidi.
3. AVR Microcontroller and Embedded Systems: Using Assembly and C by Muhmmad Ali Mazidi, Sarmad Naimi, Sepehr Naimi.
4. Elecia White : Making Embedded Systems by Elecia White, O’ Reilly Publication.
5. The Firmware Handbook (Embedded Technology) by Jack Ganssle.
6. Jack Purdum: Beginning C for Arduino, Second Edition: Learn C Programming for the Arduino.
7. Steve Gold: Aurdino: Mastering Basic Arduino: The Complete Beginner’s guide to Arduino.
8. Mark Torvalds: Arduino: Step-By-Step Guide to Master Arduino Hardware and Software.
9. Learn Electronics with Arduino: An Illustrated’s Begineer’s Guide to Physical Computing(Make) by Jody Culkin and Eric Hagan.
10. Arduino Sketches: Tools and Techniques for Programming Wizadrdy by James A. Langbridg.
11. Handbook of Modern Sensors: Physics, Designs, and Applications 5th ed. 2016 Edition by Jacob Fraden.
12. Circuit Design for Electronic Instrumentation: Analog and Digital Devices from Sensor to Display Hardcover – December 1, 1979 by Darold Wobschall.
13. Sensors and Signal Conditioning, 2nd Edition 2nd Edition by Ramón Pallás-Are.