A PROJECT REPORT ON

“FLOOD ALARMING SYSTEM USING ARDUINO UNO AND MOISTURE SENSOR”

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE OF

BACHELOR OF ENGINEERING IN COMPUTER ENGINEERING

PREPARED BY

MEGHSHAM VINAYAK KAPURE.	{72007422D}
SOHAM SANTOSH SOLAT.	{71918366K}
DNYANAL KUMAR VEDPATHAK.	{71918374L}

UNDER THE GUIDANCE OF

Prof. K.S. KHAMBKAR

DEPARTMENT OF COMPUTER ENGINEERING

RDTC’S SCSCOE Dhangawadi, Bhor 412206

Batch of 2021-22.

CERTIFICATE

This is to certify that the project report entitles,

“FLOOD ALARMING SYSTEM USING ARDUINO UNO AND MOISTURE SENSOR”

Submitted By

MEGHSHAM VINAYAK KAPURE	BCO19D23
DNYANAL KUMAR VEDPATHAK	BCO18F67
SOHAM SANTOSH SOLAT	BCO18F63

are bonafide student of this institute and the work has been carried out by them under the supervision of Prof. K.S. Khambkar and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of Bachelor of Engineering (COMPUTER ENGINEERING).

Prof. K.S. Khambkar	Prof . B.D.Thorat	Dr. S. B. Patil.
Internal Guide	HOD	Principal
DEPT. COMPUTER ENGG.	DEPT. COMPUTER ENGG.	RDTC’S SCSCOE, Pune

Date:
Place:

ACKNOWLEDGEMENT

I am using this opportunity to express my gratitude to everyone who supported me throughout the course of this project. I am thankful for their aspiring guidance, invaluably constructive criticism and friendly advice during the project work. I am sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to the project.

I express my warm thanks to my fellow colleagues for their constant support and equal participation throughout the completion of this project and, for the valuable information provided by them in their respective field. I am grateful for their cooperation during the period of my assignment.

I would also like to thank my project guide Prof.K.S Khamkar and all the people who provided me with the facilities being required and conductive conditions for my project.

Thank you,
Best Regards,

Meghsham Vinayak Kapure.

Soham Santosh Solat.

Dnyanal Kumar Vedpathak.

ABSTRACT

In developing countries, flooding due to natural disasters such as hurricanes and earthquakes results in massive loss of life and property. Warning communities of the incoming flood provides an effective solution to this by giving people sufficient time to evacuate and protect their property.

However, the range of early warning system solutions introduces a tangle of conflicting requirements including cost and reliability, and creates several interesting problems from factors as diverse as technological, social, and political. The complexity of these systems and need for autonomy within the context of a developing country while remaining maintainable and accessible by non-technical personnel provides a challenge not often solved within developed countries, much less the developing. After describing this problem, the paper discusses a proposed solution for the problem, initial experiments in implementing the solution, and lessons learned through that work.

Keywords: Flooding alert, ARDUINO

CONTENTS

CHAPTER 1

INTRODUCTION

Problem Statement:

The objective of this project is to notify or alert the user of the rise in water level of source like river or dams to reduce hazard and disaster by flooding.

Project Theory:

Flood is one of the natural disasters that occurs worldwide. It destroys the infrastructure and causes fatalities. Flood alarming system can monitor the flood level and warn people upon the danger of the flood. Existing flood monitoring techniques include multi-satellite analysis, image classifications and wireless sensor networks.

Unlike the existing systems, this project intends to develop a more robust and durable system which can withstand the wet weather condition. It aims to monitor the water level and alert the authorities as well as notifying victims. In order to do this, the system needs to have the basic information such as water conditions, water level and precipitation level to detect the increase of water level during flood. Two major components consisting of the sensor network and the data transmission were designed in this project.

“FLOOD ALARMING SYSTEM” is an amazing and very useful project. The objective of this project is to notify the user the amount of water that is present in the water source using alert.

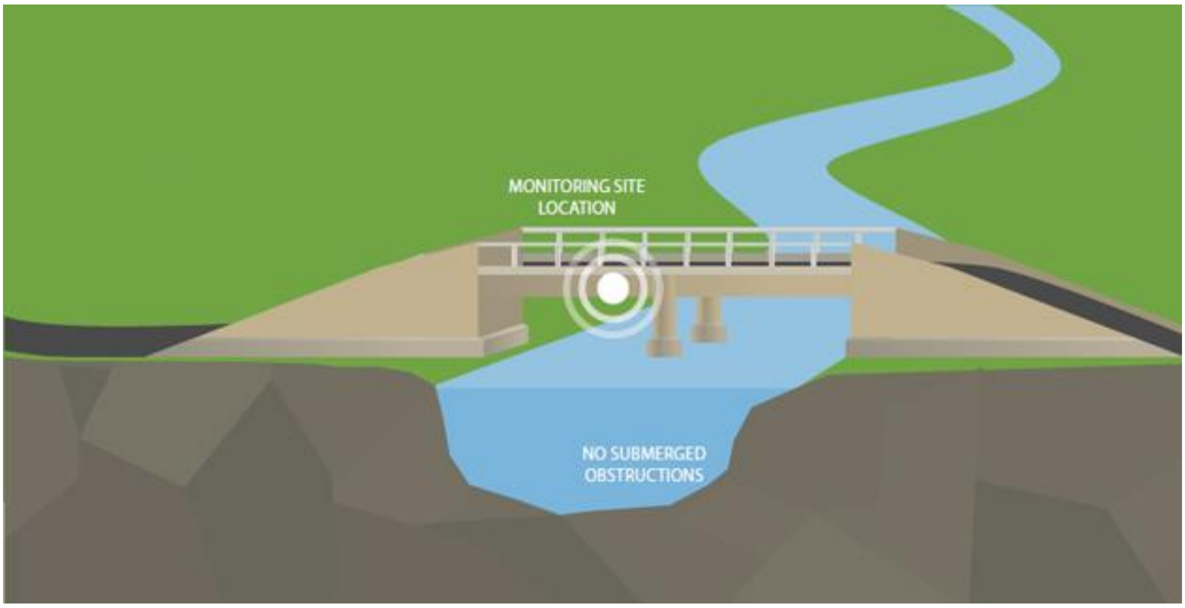
CHAPTER 2

APPLICATIONS

Monitoring System:

As mentioned before, the ideal placement for a flood warning gage will depend largely on the site considerations of the waterway where it is located. Careful planning is needed to select the location, determine substrate stability and water level fluctuation, and to design a housing solution that will effectively protect the gage from acts of nature or vandalism. Physical constraints of a site, the time required to reach the location, legal and physical access to a site, and safety issues must be considered when making site selections.

Site Construction:



- The monitoring location should have access to a bridge or overhead structure for securely mounting the radar sensor. There should be a clear path between the sensor and the water to avoid false reflections.
- Avoid submerged obstructions such as rocks or bridge piers that disturb or influence the water level. Check for such obstructions when the water is at the lowest anticipated level.
- The mounting location should also avoid horizontal structural surfaces such as beams, brackets, and side wall joints, these surfaces tend to reflect a strong false signal.
- Avoid man-made control/flow structures upstream or downstream of the site that may change flow profiles erratically, as this will making rating development difficult.

CHAPTER 3

SYSTEM REQUIREMENTS

Hardware used:

1. Arduino Uno
2. Soil Moisture Sensor HW-103
3. LED (Generic)
4. Jumper Wires (Generic)
5. Buzzer (Generic)

Software used:

1. Programming Language: C++
2. IDE: Arduino IDE

Other:

1. USB 2.0 A/B cable
2. Casing

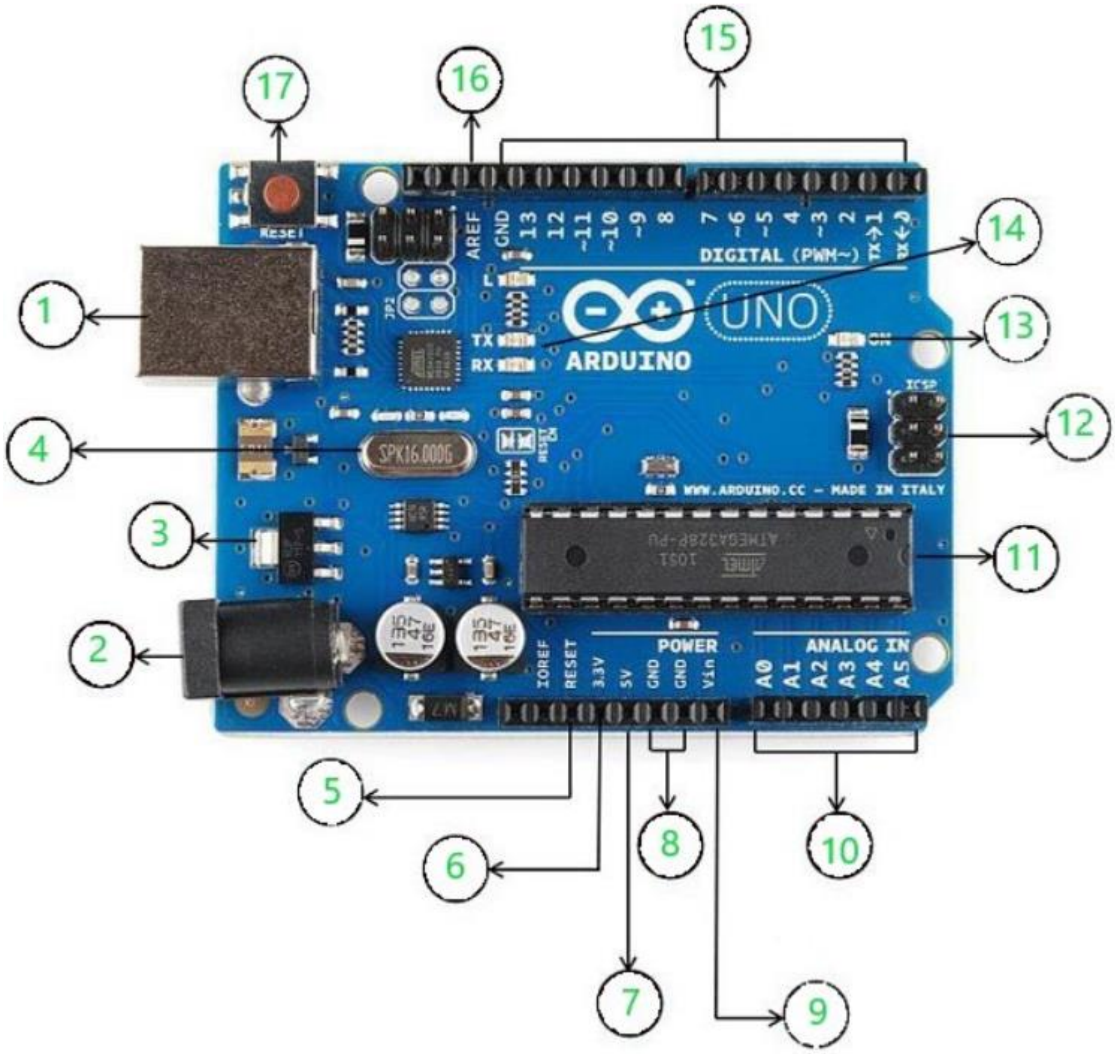
CHAPTER 4

WORKING OF COMPONENTS

1. Arduino Uno:

Arduino UNO is a low-cost, flexible, and easy-to-use programmable open-source microcontroller board that can be integrated into a variety of electronic projects. This board can be interfaced with other Arduino boards, Arduino shields, Raspberry Pi boards and can control relays, LEDs, servos, and motors as an output.

Following image shows the components of ARDUINO UNO :



Using the above image as a reference, the labeled components of the board respectively are-

1) USB: can be used for both power and communication with the IDE
2) Barrel Jack: used for power supply
3) Voltage Regulator: regulates and stabilizes the input and output voltages
4) Crystal Oscillator: keeps track of time and regulates processor frequency
5) Reset Pin: can be used to reset the Arduino Uno
6) 3.3V pin: can be used as a 3.3V output
7) 5V pin: can be used as a 5V output
8) GND pin: can be used to ground the circuit
9) Vin pin: can be used to supply power to the board
10) Analog pins(A0-A5): can be used to read analog signals to the board
11) Microcontroller (ATMega328): the processing and logical unit of the board
12) ICSP pin: a programming header on the board also called SPI
13) Power indicator LED: indicates the power status of the board
14) RX and TX LEDs: receive(RX) and transmit(TX) LEDs, blink when sending or receiving serial data respectively
15) Digital I/O pins: 14 pins capable of reading and outputting digital signals; 6 of these pins are also capable of PWM
16) AREF pins: can be used to set an external reference voltage as the upper limit for the analog pins
17) Reset button: can be used to reset the board

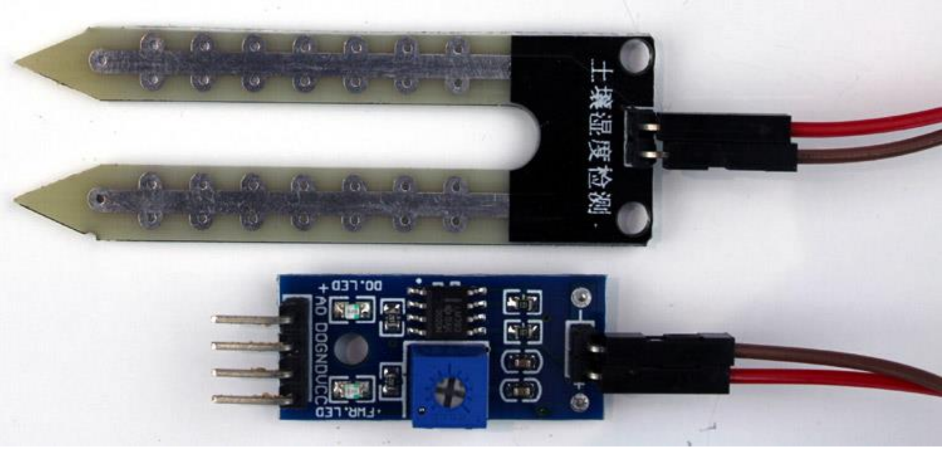
2. Soil Moisture Sensor HW-103

The soil moisture sensor module is there to convert the incoming analog signal to digital signal; this is designed in such a way that the sensor can be used without microcontroller support. The module consists of two signal input pins where the probe gets connected. It also has four other pins two of which are VCC and GND. The other two are Digital Output and Analog Output pins.

This module also consists of a High Precision Comparator, LM393 that is used to digitize the analog signal coming out of the sensor probe. The module has a built-in potentiometer that is used for sensitivity adjustment of the digital output. The main objective of the potentiometer is to set a threshold, so that when the moisture level exceeds the threshold value, the module will output LOW otherwise HIGH. This feature of the module can come in very handy because when a certain threshold is reached, you can trigger a relay that can start pumping water.

Sensors should be placed at several different depths and locations in the field. Typically, sensors are placed in pairs at one-third and two-thirds the depth of the crop root zone and at two or more locations in the field, preferably away from high points, depressions and slopes.

Following image shows the Soil Moisture Sensor HW-103



1) **S (signal pin)** is an analog output that will be connected to the analog pin of the Arduino.
2) **+VCC** is powering pin of the sensor. The approved input voltage is 3.3v to 5v.
3) **-GND** is simply a ground connection.

3. LED (Generic)

A light-emitting diode (LED) is a semiconductor light source that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons.

4. Jumper Wires (Generic)

A jump wire (also known as jumper, jumper wire, DuPont wire) is an 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 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 of a circuit board, or a piece of test equipment.

5. Buzzer (Generic)

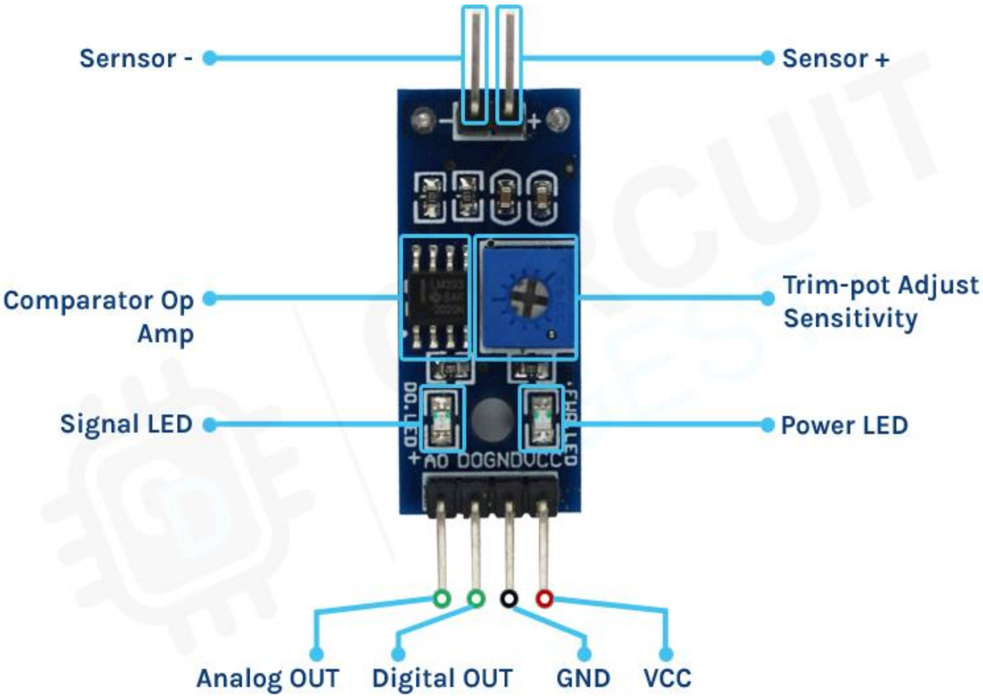
This is a small Buzzer, which can be mount on a PCB. It has two pins; the bigger pin should be connected to positive supply and shorter one should be connected to ground connection. Operating voltage range is 5-12V. If you provide square wave of 5V, it will create a good sound. For an optimum sound quality, provide 2-4KHz wave

Features:

- This is a small Buzzer, which can be mount on PCB
- Operating voltage range is 5-12V

6. The Soil Moisture Sensor Probe:

As we have said earlier, the sensor contains a fork-shaped probe with two big exposed conductive pads. The probe acts like a variable potentiometer, the value of which can be read by a microcontroller like Arduino.



7. USB 2.0 A/B cable (Generic)

The USB2HAB6 6ft USB A-to-B cable features one USB 'A' male connector and one USB 'B' male connector, providing a high-quality connection to USB 2.0 peripherals such as USB printers, scanners or external USB hard drives.

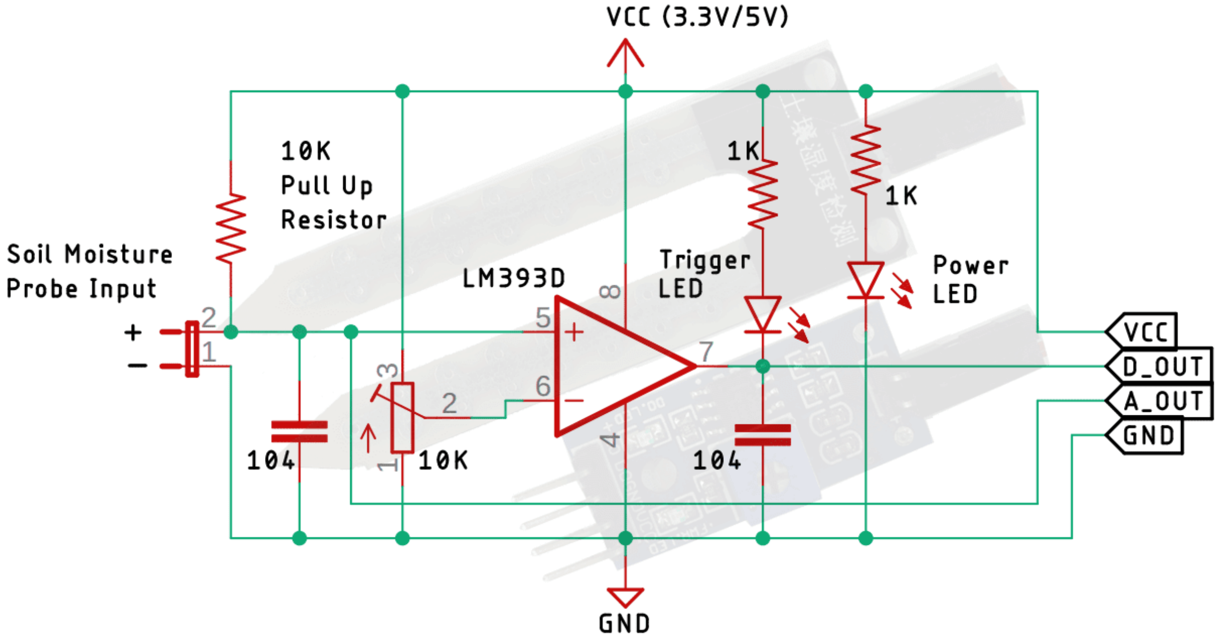
7. Casing

The Casing used to prevent the circuitry from harsh weather conditions.

CHAPTER 5

PROJECT DESIGN & IMPLANTAION OF CIRCUITRY

CIRCUIT DIAGRAM:



The schematic diagram for the soil moisture sensor module is shown below. The schematic itself is very simple and needs a handful of generic components to build. If you don't have a prebuilt module on hand but still want to test your project, the schematic below will come in handy.

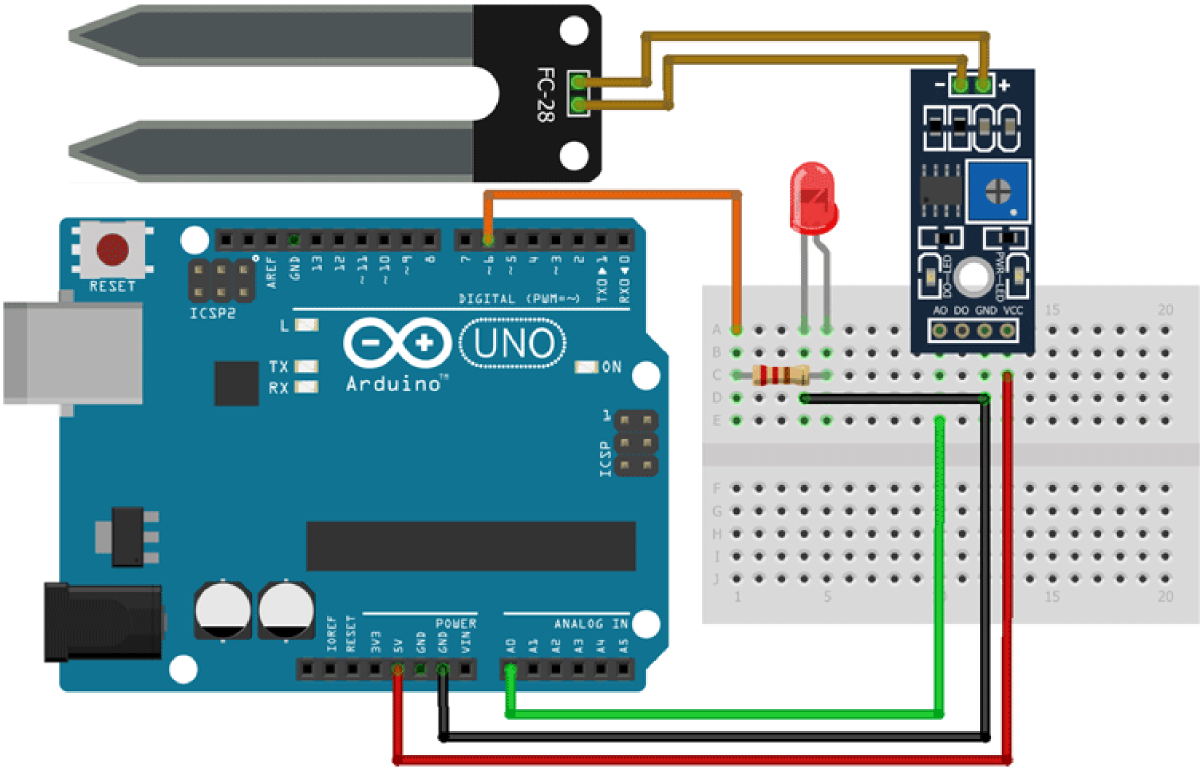
In the schematic, we have a LM393 op-amp which is a low-power low offset voltage op-amp that can be powered from a 3.3V or 5V supply. Please note that the analog output voltage of the device will depend on the input voltage. The main job for this op-amp is to convert the incoming analog signal from the sensor probe to digital signal. There is also this 10K potentiometer that is used to set a reference voltage for the op-amp, the input voltage of the sensor goes below the threshold voltage set by the potentiometer, the output of the op-map goes low. Other than that we have two LEDs. The first one is a power LED and the other one is the trigger LED. The power LED turns on when power is applied to the board and the trigger LED turns on when a certain set threshold is reached. This is how this basic circuit works.

CONNECTION DIAGRAM

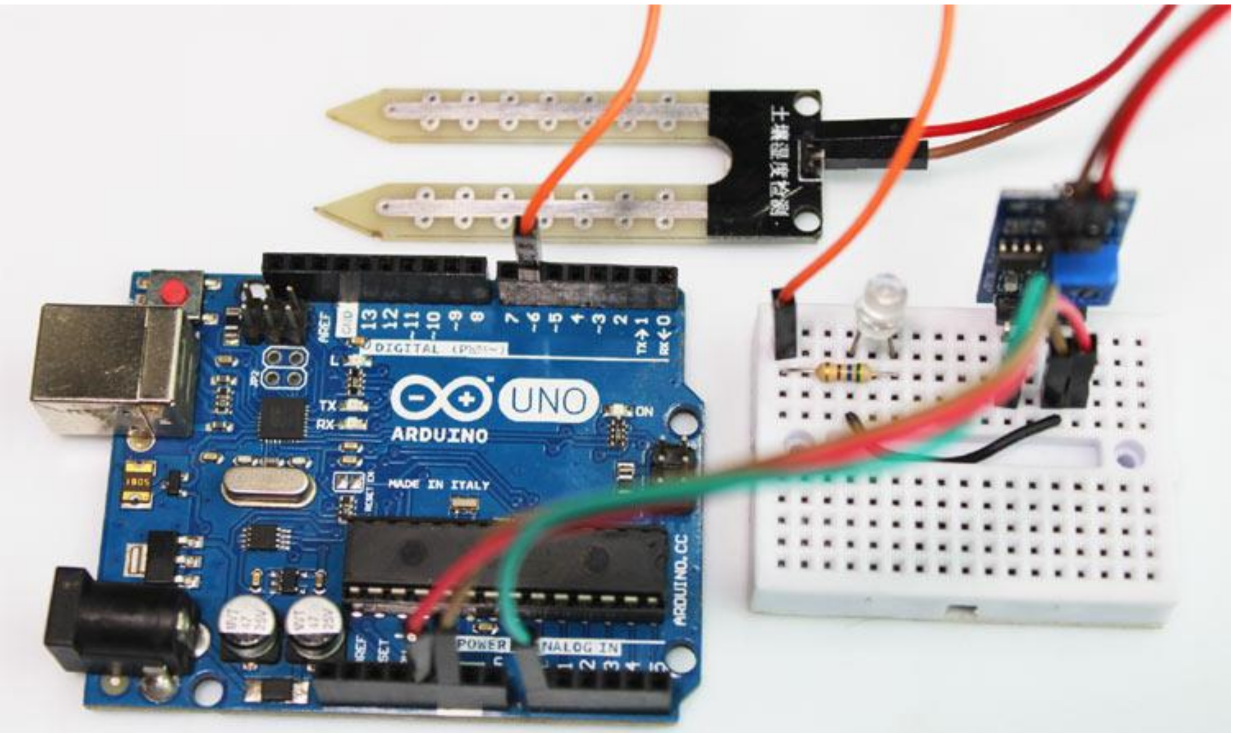
Now that we have a complete understanding of how a Soil Moisture sensor works, we can connect all the required wires to the Arduino UNO board. This section of the article will be divided into two parts, one shows analog output and another one shows the digital output. Let's begin with analog circuitry-

Soil Moisture Sensor - Analog Output:

To work with the sensor, we need to power the sensor first, for that we are using the 5V and GND pin of the Arduino UNO Board.

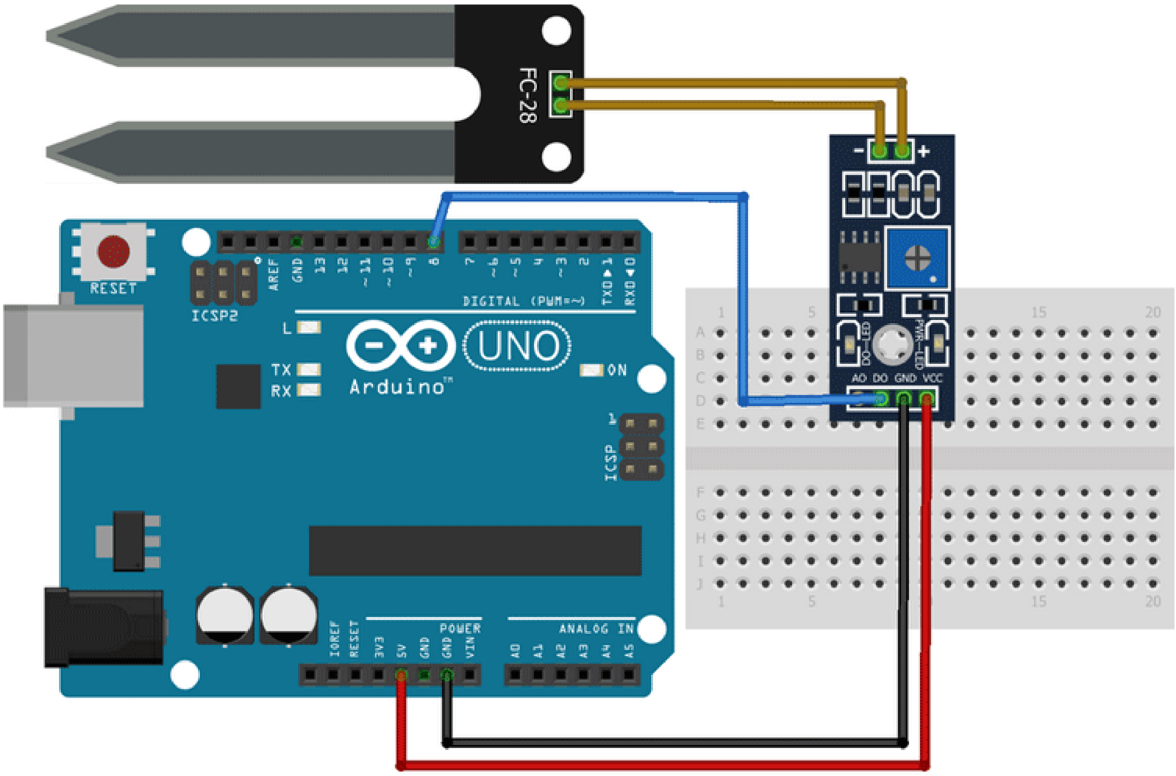


As shown in the above Arduino soil moisture sensor circuit diagram we have connected an LED to digital PIN 6 of the Arduino and the analog out pin of the sensor is connected to the A0 pin of the Arduino UNO board, finally, the ground is common between the LED and the sensor. We will program the Arduino so that the brightness of the LED will change depending on the soil moisture data sensed by the probe.

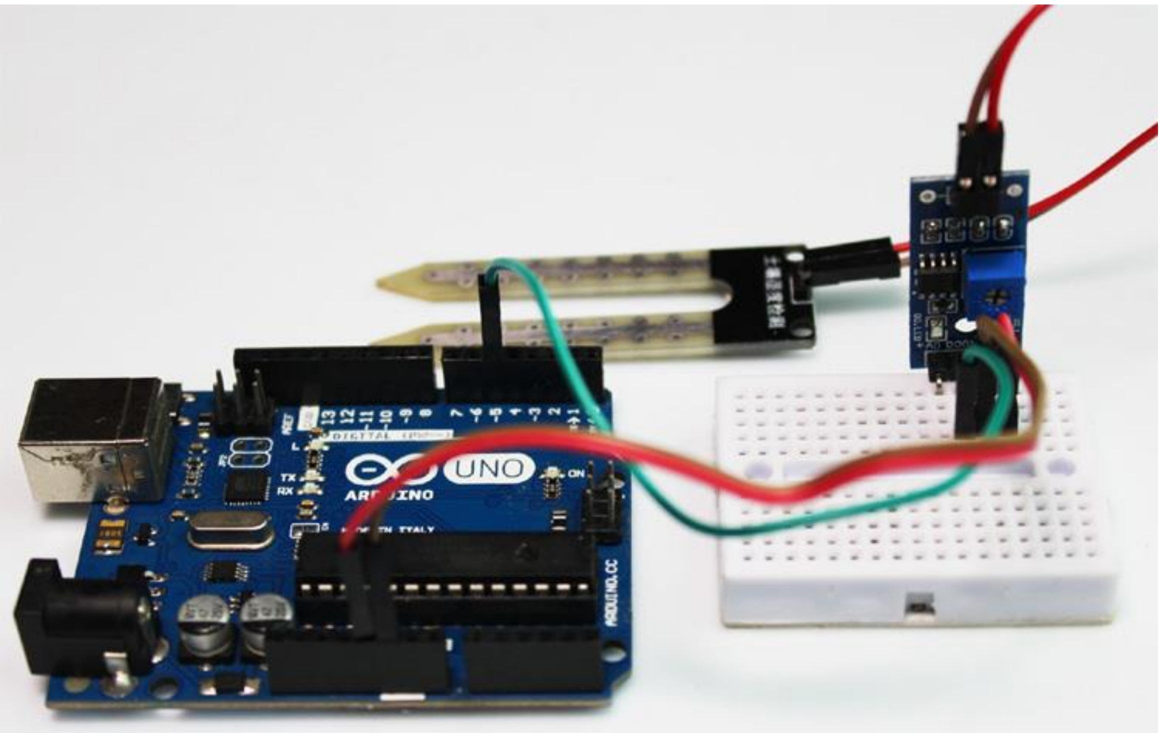


Soil Moisture Sensor - Digital Output:

For the digital interface part, we are also using the +5V and Ground from the Arduino to power the sensor module.



Connecting a soil moisture sensor to Arduino or any other microcontroller is pretty simple. As we all know the sensor outputs both analog and digital signals so processing this signal is very easy.



CHAPTER 7

SOFTWARE USED

1. **IDE**: -

a program for Arduino hardware may be written in any programming language with compilers that produce binary machine code for the target processor. Atmel provides a development environment for their 8-bit avr and 32-bit arm cortex-m based microcontrollers: avr studio (older) and atmel studio (newer) the Arduino integrated development environment (ide) is a cross-platform application (for macOS and Linux) that is written in the programming language java. it originated from the ide for the languages processing and wiring. it includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one- click mechanisms to compile and upload programs to an Arduino board. it also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus. the source code for the IDE is released under the gnu general public license, version 2.

the Arduino ide supports the languages c and C++ using special rules of code structuring. the Arduino ide supplies a software library from the wiring project, which provides many common input and output procedures. user-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the gnu toolchain, also included with the ide distribution. the Arduino ide employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

2. **Pro IDE: -**

on october 18th, 2019, Arduino 3333pro ide (alpha preview) was released. the system still uses Arduino cli (command line interface), but improvements include a more professional development environment, autocompletion support, and git integration. the application frontend is based on the eclipse their open-source ide. the main features available in the alpha release are:

- modern, fully featured development environment
- dual mode, classic mode (identical to the classic arduino ide) and pro mode (file system view)
- new board manager

- new library manager
- board list
- basic auto-completion (arm targets only)
- git integration
- serial monitor
- dark mode

3. **SKETCH**:

a *sketch* is a program written with the Arduino IDE sketches are saved on the development computer as text files with the file extension. ino. Arduino software (ide) pre-1.0 saved sketches with the extension. pde. a minimal Arduino C /C++ program consists of only two functions

- **setup():** this function is called once when a sketch starts after power-up or reset. it is used to initialize variables, input and output pin modes, and other libraries needed in the sketch. it is analogous to the function main()
- **loop():** after setup() function exits (ends), the loop() function is executed repeatedly in the main program. it controls the board until the board is powered off or is reset. it is analogous to the function while(1) blink example.

power led (red) and user led (green) attached to pin 13 on an arduino compatible board most arduino boards contain a light-emitting diode (led) and a current limiting resistor connected between pin 13 and ground, which is a convenient feature for many tests and program functions. a typical program used by beginners, akin to hello, world!, is "blink". which repeatedly blinks the on-board led integrated into the arduino board. this program uses the functions pinmodel). digitalwrite(), and delay(), which are provided by the internal libraries included in the ide environment, this program is usually loaded into a new Arduino board by the manufacturer.

4. LIBRARIES

The open-source nature of the Arduino project has facilitated the publication of many free software libraries that other developers use to augment their projects.

The uno is one of the more popular boards in the Arduino family and a great choice for beginners. we'll talk about what's on it and what it can do later in the tutorial.

CHAPTER 8

IMPLANTATION OF ARDUINO SOIL MOISTURE SENSOR CODE

The **code for Arduino Based Soil Moisture Sensor** is very simple and easy to understand. We are just reading the analog data out of the sensor and changing the brightness of the LED according to the received data. Please do remember that we are only processing the analog data coming out of the sensor for the digital data you can see the onboard LED in the module lights up.

We initialize our code by declaring two macros, the first one is for the led where we will connect an LED and the second one is the sensorPin through which we are reading the data coming out of the sensor.

```
// Sensor pins pin D6 LED output, pin A0 analog Input

#define ledPin 6

#define sensorPin A0
```

Next, we have our **setup()** function. In the setup function, we initialize the serial with 9600 baud. We also set the ledPin as output, and make the pin LOW. This way the pin will not float and turn the LED on.

```
void setup() {

  Serial.begin(9600);

  pinMode(ledPin, OUTPUT);

  digitalWrite(ledPin, LOW);

}
```

Next, we have our **loop()** function, in the loop function we print "Analog output:" as text on the serial monitor window and then we call the **readSensor()** function inside a **Serial.println()** function so that once the **readSensor()** function is executed, it returns the data and it also gets printed on the serial monitor window,

```
void loop() {
  Serial.print("Analog output: ");
  Serial.println(readSensor());
  delay(500);
}
```

Finally, we have our custom **readSensor()** function which returns the analog value that is read through the A0 pin of the Arduino. In the first line of this function, we have declared and defined a variable called **sensorValue** where we are putting the raw data, that is read through the A0 in the Arduino. This data is in 10-bit format and it goes from 0 -1023 so to convert that 10-bit data to 8-bit data. That is why we have used the map function, once the map function outputs the data we have initiated another variable **outputValue** and put the mapped data inside that variable. Finally, we have used the built-in **analogWrite(ledPin, outputValue)** function of the Arduino to generate a PWM signal that is proportional to the input data read by the ADC of the arduino. This marks the end of the code portion of the Arduino-based Soil Moisture sensor code. If you have any questions regarding the code do not hesitate to comment it down below.

```
intt readSensor() {
  int sensorValue = analogRead(sensorPin);  // Read the analog value
from sensor
  int outputValue = map(sensorValue, 0, 1023, 255, 0); // map the 10
-bit data to 8-bit data
  analogWrite(ledPin, outputValue); // generate PWM signal
  return outputValue;             // Return analog moisture value
}
```

CHAPTER 9

FUTURE SCOPE

This project can be improved by adding remote access though communication channel like SMS thus sensor information can reach faster to the authorities and also adding sustainable power source like solar power can give more power and stand alone feature to the system implemented.

CONCLUSION

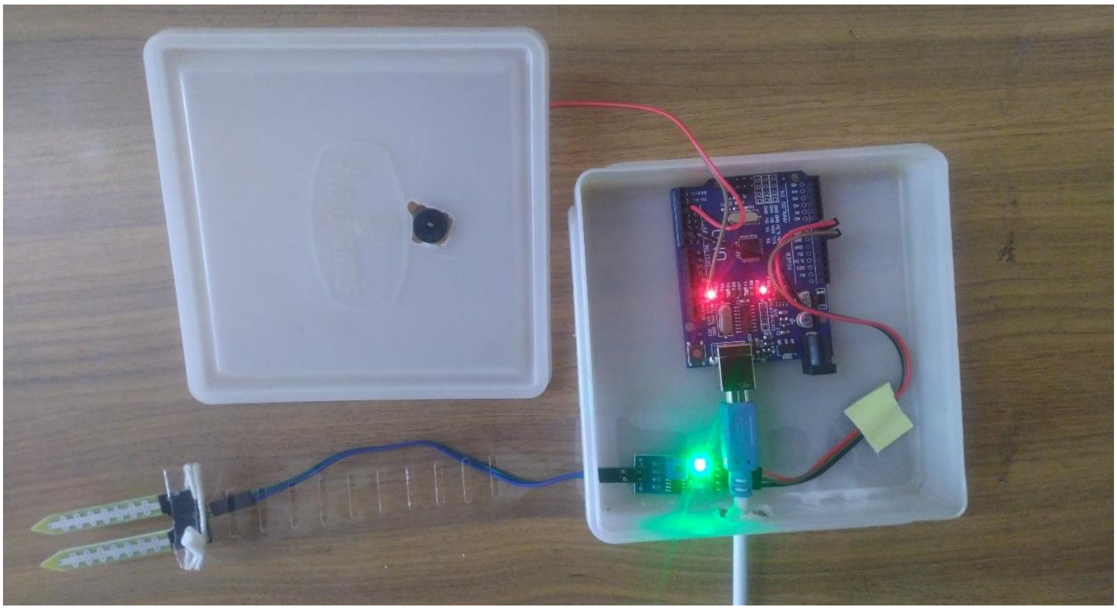
In this project we came to know the working of Arduino, its hardware / software features and its applications as to where it is currently being used. We have also learnt how to write sketches for Arduino in its own IDE (software). Developing new ideas with Arduino is endless. The possibilities of using an Arduino to learn and develop new ideas are infinite. Though it does have its own limitations, it is a great tool that can be used in learning.

Also implementing “Flood Alarming System Using” was great learning experience.

CHAPTER 10

OUTPUT

- Actual Project Hardware:



- Actual Project Working:

  https://drive.google.com/drive/folders/1BMjZApsy-PuafUm-suMnOu5YyGnBugXT

CHAPTER 11

References and Bibliography

Research Paper & Thesis:

1. Flood Disaster Indicator Of Water Level Monitoring System by Wan Haszerila Wan Hassan, Aiman Zakwan Jidin, Siti Asma Che Aziz, Norain Rahim

2. Flood Monitoring and Early Warning System Using Ultrasonic Sensor by J G Natividad and J M Mendez

3. Smart IoT Flood Monitoring System by Shahirah Binti Zahir, Phaklen Ehkan, Thennarasan Sabapathy, Muzammil Jusoh, Mohd Nasrun Osman, Mohd Najib Yasin, Yasmin Abdul Wahab, N.A.M Hambali, N. Ali, A.S. Bakhit

Websites :

- https://www.arduino.cc/
- https://en.wikipedia.org/wiki/Arduino
- https://www.researchgate.net/