PAPER NAME: - Project Stage - II PAPER CODE: - EC881

## TECHNO ENGINEERING COLLEGE BANIPUR

*Is Affiliated To*

## MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY

(Formerly known as West Bengal University of Technology)





*Is Affiliated To*

A PROJECT REPORT ON “IoT Early Flood Detection & Avoidance”

Submitted in partial fulfillment of the requirement for the award of the degree of

# BACHELOR OF TECHNOLOGY

**(Department of Electronics & Communication Engineering)**

PRESENTED BY

|  |  |  |
| --- | --- | --- |
| ***NAME*** | ***ROLL. NUMBER*** | ***REGISTRATION NUMBER*** |
| **SOURAV PAUL** | **24400319006** | **018974 OF 2019-20** |
| **ADITYA SHEE** | **24400319013** | **041923 OF 2019-20** |

Under the esteemed guidance of

MRS. TANUSHREE SAHA SARKAR

### Asst. Professor, Dept. of ECE

**CERTIFICATE OF APPROVAL**

This is to certify that the project entitled **“**IoT Early Flood Detection & Avoidance” is being submitted by **SOURAV PAUL (24400319006), ADITYA SHEE (24400319013)** **,** in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology in Electronics & Communication Engineering** from TECHNO ENGINEERING COLLEGE BANIPUR (Affiliated To **MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY)**.

### Signature of the Project Guide Signature of the Head of the Department

**Mrs. Tanushree Saha Sarkar**  **Mr. Sanjiv Kumar Dhara**

### Asst. Professor Asst. Professor

**Department of ECE Department of ECE**

**\_**

### Signature of the Examiner College Seal

**DECLARATION**

We **SOURAV PAUL (24400319006), ADITYA SHEE (24400319013),** hereby declare that the project report entitled **“IoT Early Flood Detection & Avoidance”** under the guidance of **Mrs. Tanushree Saha Sarkar,** TECHNO ENGINEERING COLLEGE BANIPUR (Affiliated to

**MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY**), is submitted in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Electronics & Communication Engineering.

Our main goal (by creating this project) is to embrace ourselves for caution, to minimize the damage caused by the flood. Natural disasters like a flood can be devastating leading to property damage and loss of lives. To eliminate or lessen the impacts of the flood, the system uses various natural factors to detect flood. The system has a Wi-Fi connectivity, thus it’s collected data can be accessed from anywhere quite easily using IoT. At first we have to gain popularity then we will monetize our system to gain some profit. Our target is to make this system affordable for everyone (this will be our future prospective).

### SOURAV PAUL ADITYA SHEE (24400319006) (24400319013)

**[SIGNATURE OF THE PROJECT DEVELOPERS]**

First and foremost, we wish to express our gratitude towards the institution TECHNO ENGINEERING COLLEGE BANIPUR (Affiliated to MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY) for fulfilling the most cherished goal of our life to do Bachelor of Technology (in Electronics & Communication Engineering).

It is great pleasure in expressing deep sense of gratitude to our **Project guide,**

Mrs. TANUSHREE SAHA SARKAR **(Asst. Professor, Department of ECE)** for her valuable guidance and freedom that she gave to us.

We also express our sincere thanks to MR. SANJIV KUMAR DHARA**, (Head of the Department of ECE)** for his encouragement and support throughout the project.

Our outmost thanks also goes to all of the **FACULTY MEMBERS** and **NON- TEACHING STAFF** of the Department of Electronics & Communication Engineering for their support throughout our project work.

**ABSTRACT**

Since we are now currently present in an era of Computing Technology, it is essential for everyone and everything to be connected to the internet. IOT is a technology that brings us more and more close to this goal.

Our project comprises of smart water monitoring system which is a small prototype for flood detection and avoidance system. This project explains the working and the workflow of all the components present inside our project. The sensors sense the environment and sends real-time data to the cloud (firebase cloud) and users can view and access this data via their mobile platform. The model gives a warning after the water level rises to a particular height.

Since it is a small scaled prototype for flood detection and avoidance system, the working of this model is good. The data are uploaded and changed in the cloud in precision to the sensor and real-time changes in the mobile application is achieved. This model can be used to greatly reduce the casualties in a devastating event of flood.

**CONTENTS**

*Chapter – 1*

1. INTRODUCTION
   1. Project Outline 10
   2. Software’s Used 10
   3. Hardware 11
   4. Block Diagram 12

*Chapter – 2*

1. Arduino Uno Board
   1. Overview 13
   2. Features 13
   3. Board Configuration 14
   4. Board Description 15-16

*Chapter – 3*

1. ESP8266 NodeMCU Wi-Fi Module
   1. Overview 17
   2. Technical Features 17
   3. Board Configuration 18
   4. Board Description 19-20

*Chapter – 4*

1. DHT11 Temperature & Humidity Sensor
   1. Overview 21
   2. Technical Features 21
   3. Working Principle 21
   4. Pin Configuration & Schematic Diagram 22-23

*Chapter – 5*

1. HC-SR04 Ultrasonic Sensor
   1. Overview 24
   2. Technical Features 24
   3. Working Principle 25
   4. Pin Configuration & Schematic Diagram 25-26

*Chapter – 6*

1. Rain Sensor Module
   1. Overview 27
   2. Technical Features 27
   3. Working Principle 28
   4. Pin Configuration & Schematic Diagram 29

*Chapter – 7*

1. Water Flow Sensor
   1. Overview 30
   2. Technical Features 30
   3. Working Principle 31
   4. Pin Configuration & Schematic Diagram 32

*Chapter – 8*

1. 20 x 4 LCD display
   1. Overview 33
   2. Technical Features 33
   3. Working Principle 34
   4. Pin Configuration & Schematic Diagram 35

*Chapter – 9*

1. I2C Serial Communication Module
   1. Overview 36
   2. Technical Features 36
   3. Working Principle 37
   4. Pin Configuration & Schematic Diagram 38

*Chapter – 10*

10.1 Others circuitry elements 39

*Chapter – 11*

1. Hardware Portion

11.1 Circuit Diagram 40

11.2 Circuit Description 41-42

11.3 Working Procedure 43

11.4 Technical Specification 44-45

*Chapter – 12*

1. Software Portion
   1. Introduction 46
   2. Programming 46
   3. Source code 47-52

*Chapter – 13*

1. Application of the Project 53

*Chapter – 14*

1. Future Work of the Project 54

*Chapter - 15*

1. Conclusion 55

*Chapter – 16*

1. References 56

***Chapter – 1* : INTRODUCTION**

The extreme climatic changes due to the effect from various human activities such as pollutions, cutting of innumerable trees and too much of gas emission are the some of the main reason for natural disasters that occur in worldwide. The most common factor that cause major damage to life, property and country’s economy is the flood.

Flooding is brought on by an increased quantity of water in lake or river when it is overflowing. When a dam fractures and abruptly releasing a massive quantity of water not only houses and property are damaged, sewage overflow and chemical spillage also leads to a variety of diseases afterwards. To manage these kind of situations and alert people understanding of increased water level and speed of water flow are valuable for discovering potential seriousness of the flood. This project presents the details of how the data - like flood level and rain intensity are collected from sensors and made available on cloud and sending alert messages by using Thingspeak-an IOT platform to relay data from sensors to computers or directly alert the People of that area through their mobile phone. The data from the IOT cloud can be accessed by android smart phones at anytime from anywhere in the world using the mobile app things view.

* 1. **Project Outline :**
* A brief introduction to internal architecture of Arduino Uno Board.
* An overview of programming of Arduino Uno IDE software.
* Arduino Uno board interfacing with different sensors.
* Interfacing ESP8266 NodeMCU with Arduino Uno R3.
* Wirelessly transmit data to Thingspeak server.

**1.2 Software Used :**

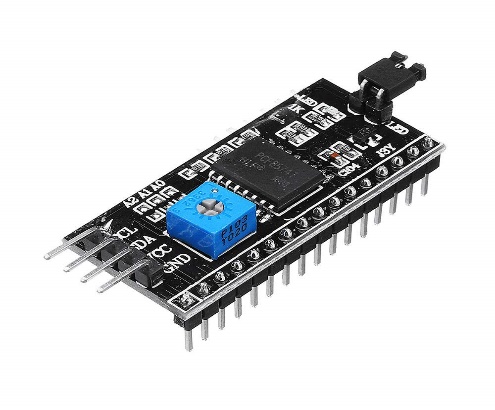
* The software for the different sensors and NodeMCU is written in Arduino Programming Language.
* The Arduino Uno is programmed using Arduino IDE software.

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.

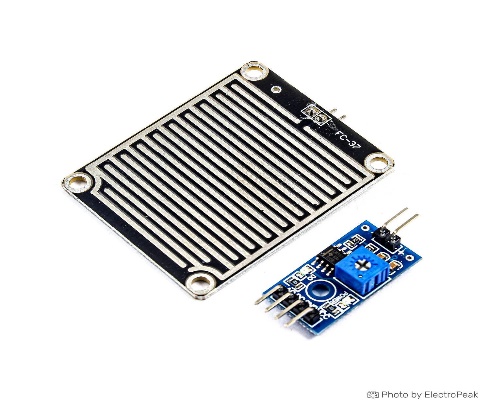
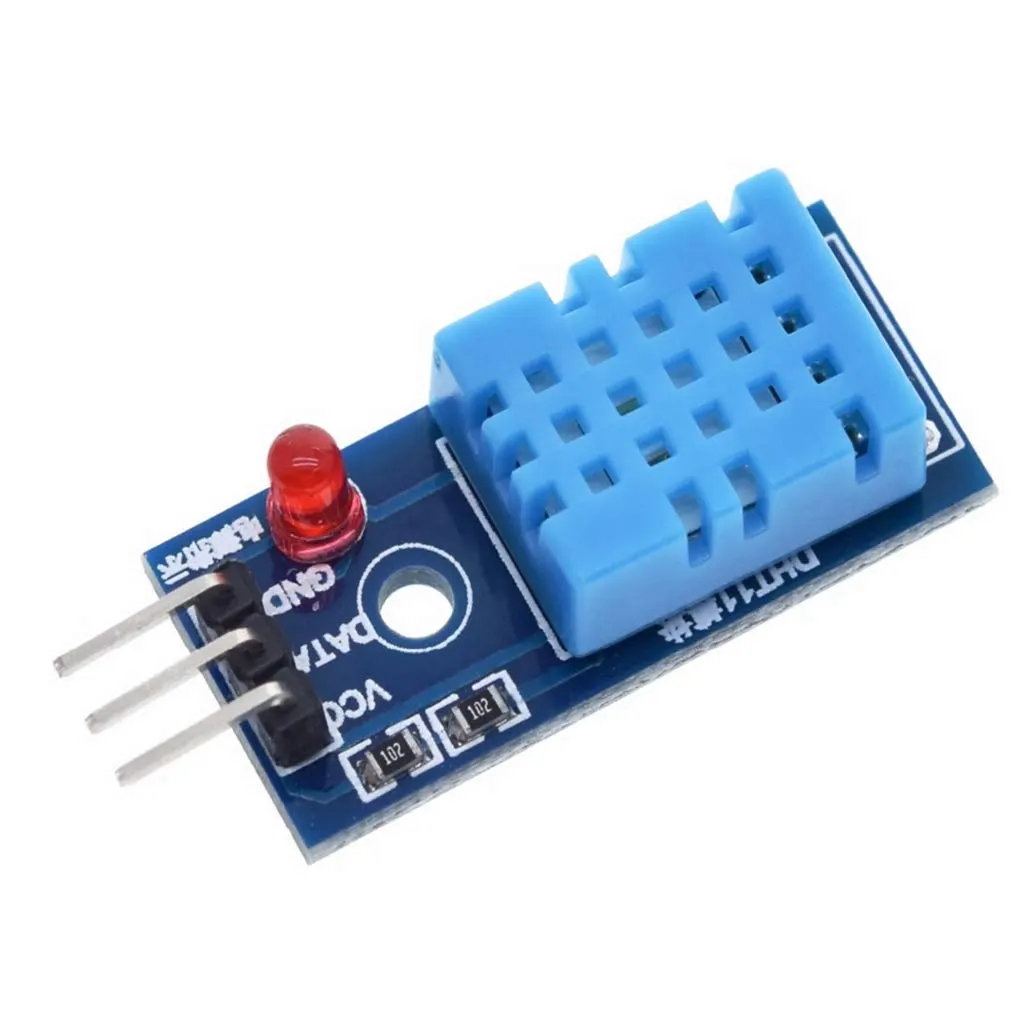


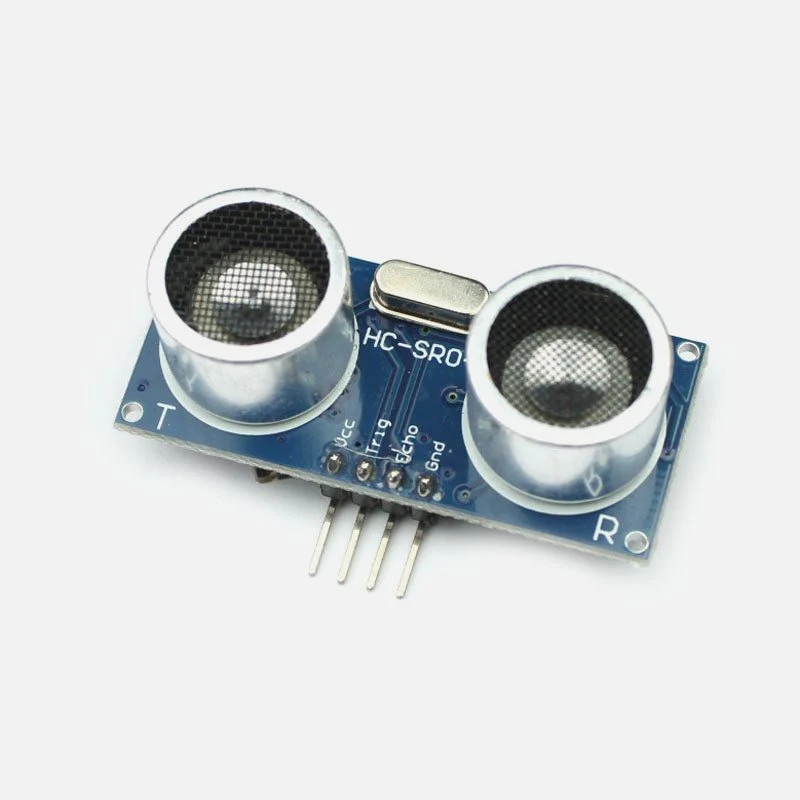
**1.3 Hardware Used :**

Arduino Uno Board ESP8266 NodeMCU I2C Serial Module

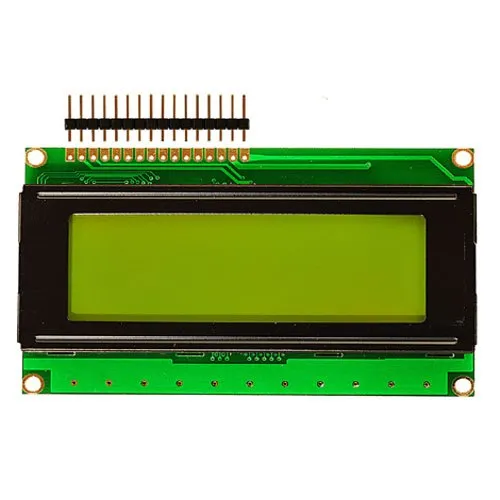


DHT11 sensor HC-SR04 Ultrasonic Rain module





Water Flow Sensor 20 x 4 Display Buzzer



**1.4 Block Diagram :**



This is the Block diagram of complete hardware of the project. In the block every sensor and module which are used in the project are mentioned.

***Chapter – 2* : Arduino Uno Board**

**2.1 Overview :**

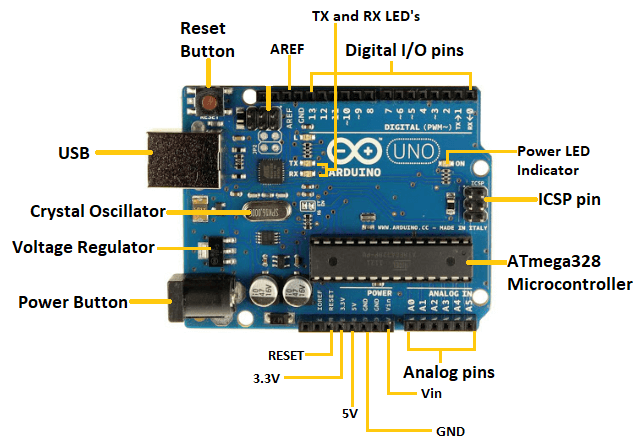
The Arduino UNO is a standard board of Arduino. Here UNO means 'one' in Italian. It was named as UNO to label the first release of Arduino Software. It was also the first USB board released by Arduino. It is considered as the powerful board used in various projects. Arduino.cc developed the Arduino UNO board.

Arduino UNO is based on an ATmega328P microcontroller. It is easy to use compared to other boards, such as the Arduino Mega board, etc. The board consists of digital and analog Input/output pins (I/O), shields, and other circuits.

The Arduino UNO includes 6 analog pin inputs, 14 digital pins, a USB connector, a power jack, and an ICSP (In-Circuit Serial Programming) header. It is programmed based on IDE, which stands for Integrated Development Environment. It can run on both online and offline platforms.

**2.2 Features :**

* 1.0 pinout : added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible with both the board that uses the AVR, which operates with 5V and with the Arduino that operates with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
* Stronger RESET circuit.
* Atmega 16U2 replace with 8U2.
* Microcontroller Atmega 328.
* Input voltage (recommended) 7V to 12V.
* Input voltage (limits) 7V to 20V.
* Digital i/o pin 14 (of which 6 provided PWM output).
* Analog input pins are 6.
* DC current per i/o pin is 40mA.
* DC current for 3.3V pin is 50mA.
* Flash memory is 32KB (Atmega328) of which 0.5KB used by Bootloader.
* SRAM is 2KB.
* EEPROM is 1KB (Atmega 328).
* Clock Speed 6 MHz.
* Length = 68.6 mm.
* Width = 53.4 mm.
* Weight = 25g.

**2.3 Board Configuration :**

**2.4 Board Description :**

* **Power Supply**

The Arduino Uno power supply can be done with the help of a USB cable or an external power supply. The external power supplies mainly include AC to DC adapter otherwise a battery. The adapter can be connected to the Arduino Uno by plugging into the power jack of the Arduino board. Similarly, the battery leads can be connected to the Vin pin and the GND pin of the POWER connector. The suggested voltage range will be 7 volts to 12 volts.

* **Input & Output**

The 14 digital pins on the Arduino Uno can be used as input & output with the help of the functions like pinMode(), digitalWrite(), & Digital Read().

Pin1 (TX) & Pin0 (RX) (Serial): This pin is used to transmit & receive TTL serial data, and these are connected to the ATmega8U2 USB to TTL Serial chip equivalent pins.

Pin 2 & Pin 3 (External Interrupts): External pins can be connected to activate an interrupt over a low value, change in value.

Pins 3, 5, 6, 9, 10, & 11 (PWM): This pin gives 8-bit PWM o/p by the function of analogWrite().

SPI Pins (Pin-10 (SS), Pin-11 (MOSI), Pin-12 (MISO), Pin-13 (SCK): These pins maintain SPI-communication, even though offered by the fundamental hardware, is not presently included within the Arduino language.

Pin-13(LED): The inbuilt LED can be connected to pin-13 (digital pin). As the HIGH-value pin, the light emitting diode is activated, whenever the pin is LOW.

Pin-4 (SDA) & Pin-5 (SCL) (I2C): It supports TWI-communication with the help of the Wire library.

AREF (Reference Voltage): The reference voltage is for the analog i/ps with analogReference().

Reset Pin: This pin is used for reset (RST) the microcontroller.

* **Memory**

The memory of this Atmega328 Arduino microcontroller includes flash memory-32 KB for storing code, SRAM-2 KB EEPROM-1 KB

* **Communication**

The Arduino Uno ATmega328 offers UART TTL-serial communication, and it is accessible on digital pins like TX (1) and RX (0). The software of an Arduino has a serial monitor that permits easy data. There are two LEDs on the board like RX & TX which will blink whenever data is being broadcasted through the USB.

A SoftwareSerial library permits for serial communication on Arduino Uno digital pins and the ATmega328P supports TWI (I2C) as well as SPI-communication. The Arduino software contains a wired library for simplifying the utilization of the I2C bus.

***Chapter – 3* : ESP8266 NodeMCU**

**3.1 Overview :**

ESP8266 can be used as an external Wi-Fi module, using the standard AT Command set Firmware by connecting it to any microcontroller using the serial UART, or directly serve as a Wi-Fi-enabled micro controller, by programming a new firmware using the provided SDK.

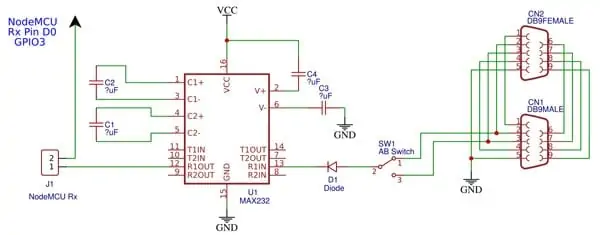
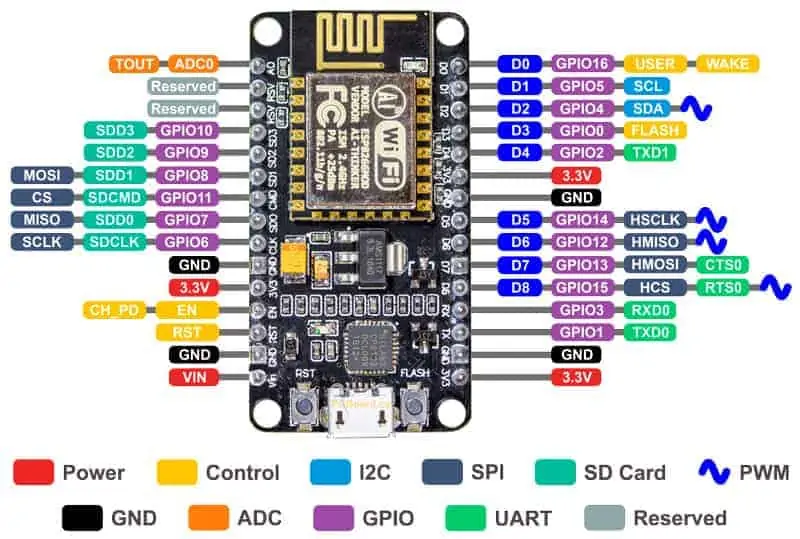
The GPIO pins allow Analog and Digital IO, plus PWM, SPI, I2C, etc.

This board has been around for almost a year now, and has been used mostly in IoT contexts, where we want to add connectivity for example to an Arduino project. A wide adoption has been facilitated by the very modest price, ranging from 2.50 to 10 USD depending on the features offered by the manufacturers.

**3.2 Technical Features :**

* 802.11 b / g / n
* Wi-Fi Direct (P2P), soft-AP
* Built-in TCP / IP protocol stack
* Built-in TR switch, balun, LNA, power amplifier and matching network
* Built-in PLL, voltage regulator and power management components
* 802.11b mode + 19.5dBm output power
* Built-in temperature sensor
* Support antenna diversity
* off leakage current is less than 10uA
* Built-in low-power 32-bit CPU: can double as an application processor
* SDIO 2.0, SPI, UART
* STBC, 1×1 MIMO, 2×1 MIMO
* A-MPDU, A-MSDU aggregation and the 0.4 Within wake
* 2ms, connect and transfer data packets
* standby power consumption of less than 1.0mW (DTIM3)

**3.3 Board Configuartion :**

****

**3.4 Board Description :**

Power Pins There are four power pins. **VIN** pin and three **3.3V** pins.

* **VIN** can be used to directly supply the NodeMCU/ESP8266 and its peripherals. Power delivered on **VIN** is regulated through the onboard regulator on the NodeMCU module – you can also supply 5V regulated to the **VIN** pin
* **3.3V** pins are the output of the onboard voltage regulator and can be used to supply power to external components.

GND are the ground pins of NodeMCU/ESP8266

I2C Pins are used to connect I2C sensors and peripherals. Both I2C Master and I2C Slave are supported. I2C interface functionality can be realized programmatically, and the clock frequency is 100 kHz at a maximum. It should be noted that I2C clock frequency should be higher than the slowest clock frequency of the slave device.

GPIO Pins NodeMCU/ESP8266 has 17 GPIO pins which can be assigned to functions such as I2C, I2S, UART, PWM, IR Remote Control, LED Light and Button programmatically. Each digital enabled GPIO can be configured to internal pull-up or pull-down, or set to high impedance. When configured as an input, it can also be set to edge-trigger or level-trigger to generate CPU interrupts.

ADC Channel The NodeMCU is embedded with a 10-bit precision SAR ADC. The two functions can be implemented using ADC. Testing power supply voltage of VDD3P3 pin and testing input voltage of TOUT pin. However, they cannot be implemented at the same time.

UART Pins NodeMCU/ESP8266 has 2 UART interfaces (UART0 and UART1) which provide asynchronous communication (RS232 and RS485), and can communicate at up to 4.5 Mbps. UART0 (TXD0, RXD0, RST0 & CTS0 pins) can be used for communication. However, UART1 (TXD1 pin) features only data transmit signal so, it is usually used for printing log.

SPI Pins NodeMCU/ESP8266 features two SPIs (SPI and HSPI) in slave and master modes. These SPIs also support the following general-purpose SPI features:

* + - 4 timing modes of the SPI format transfer
    - Up to 80 MHz and the divided clocks of 80 MHz

SDIO Pins NodeMCU/ESP8266 features Secure Digital Input/Output Interface (SDIO) which is used to directly interface SD cards. 4-bit 25 MHz SDIO v1.1 and 4-bit 50 MHz SDIO v2.0 are supported.

PWM Pins The board has 4 channels of Pulse Width Modulation (PWM). The PWM output can be implemented programmatically and used for driving digital motors and LEDs. PWM frequency range is adjustable from 1000 μs to 10000 μs (100 Hz and 1 kHz).

Control Pins are used to control the NodeMCU/ESP8266. These pins include Chip Enable pin (EN), Reset pin (RST) and WAKE pin.

* + - **EN:** The ESP8266 chip is enabled when EN pin is pulled HIGH. When pulled LOW the chip works at minimum power.
    - **RST:** RST pin is used to reset the ESP8266 chip.
* **WAKE:** Wake pin is used to wake the chip from deep-sleep.

Tiny Sine WaveControl Pins are used to control the NodeMCU/ESP8266. These pins include Chip Enable pin (EN), Reset pin (RST) and WAKE pin.

* + - **EN:** The ESP8266 chip is enabled when EN pin is pulled HIGH. When pulled LOW the chip works at minimum power.
    - **RST:** RST pin is used to reset the ESP8266 chip.
    - **WAKE:** Wake pin is used to wake the chip from deep-sleep.

***Chapter – 4* : DHT11 Temperature & Humidity Sensor**

**4.1 Overview :**

DHT11 is a low-cost digital sensor for sensing temperature and humidity. This sensor can be easily interfaced with any micro-controller such as Arduino, Raspberry Pi etc… to measure humidity and temperature instantaneously.

DHT11 humidity and temperature sensor is available as a sensor and as a module. The difference between this sensor and module is the pull-up resistor and a power-on LED. DHT11 is a relative humidity sensor. To measure the surrounding air this sensor uses a thermistor and a capacitive humidity sensor.

**4.2 Technical Features :**

* Low cost
* to 5V power and I/O
* 2.5mA max current use during conversion (while requesting data)
* Good for 20-80% humidity readings with 5% accuracy
* Good for 0-50°C temperature readings ±2°C accuracy
* No more than 1 Hz sampling rate (once every second)
* Body size 15.5mm x 12mm x 5.5mm
* pins with 0.1" spacing

**4.3 Working Principle :**

DHT11 sensor consists of a capacitive humidity sensing element and a thermistor for sensing temperature. The humidity sensing capacitor has two electrodes with a moisture holding substrate as a dielectric between them. Change in the capacitance value occurs with the change in humidity levels. The IC measure, process this changed resistance values and change them into digital form.

For measuring temperature this sensor uses a Negative Temperature coefficient thermistor, which causes a decrease in its resistance value with increase in temperature. To get larger resistance value even for the smallest change in temperature, this sensor is usually made up of semiconductor ceramics or polymers.

The temperature range of DHT11 is from 0 to 50 degree Celsius with a 2-degree accuracy. Humidity range of this sensor is from 20 to 80% with 5% accuracy. The sampling rate of this sensor is 1Hz .i.e. it gives one reading for every second. DHT11 is small in size with operating voltage from 3 to 5 volts. The maximum current used while measuring is 2.5mA.

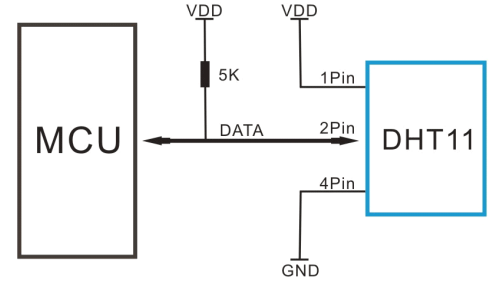
DHT11 sensor has four pins- VCC, GND, Data Pin and a not connected pin. A pull-up resistor of 5k to 10k ohms is provided for communication between sensor and micro-controller.

**4.4 Pinout Configuration & Schematic Diagram :**

### ****DHT11 Pinout Configuration****

|  |  |  |
| --- | --- | --- |
| **No:** | **Pin Name** | **Description** |
| **For DHT11 Sensor** | |
| 1 | Vcc | Power supply 3.5V to 5.5V |
| 2 | Data | Outputs both Temperature and Humidity through serial Data |
| 3 | NC | No Connection and hence not used |
| 4 | Ground | Connected to the ground of the circuit |

### 

 Pin Diagram of DHT11 Temperature & Humidity Sensor Module

Schematic Diagram of Temperature & Humidity Sensor Module

***Chapter – 5* : HC-SR04 Ultrasonic Sensor**

**5.1 Overview :**

The HC-SR04 is a type of ultrasonic sensor which uses sonar to find out the distance of the object from the sensor. It provides an outstanding range of non-contact detection with high accuracy & stable readings. It includes two modules like ultrasonic transmitter & receiver. This sensor is used in a variety of applications like measurement of direction and speed, burglar alarms, medical, sonar, humidifiers, wireless charging, non-destructive testing, and ultrasonography.

**5.2 Technical Features :**

* Input Voltage: 5V
* Current Draw: 20mA (Max)
* Digital Output: 5V
* Digital Output: 0V (Low)
* Working Temperature: -15°C to 70°C
* Sensing Angle: 30° Cone
* Angle of Effect: 15° Cone
* Ultrasonic Frequency: 40kHz
* Range: 2cm - 400cm
* Dimensions
* Length: 43mm
* Width: 20mm
* Height (with transmitters): 15mm
* Centre screw hole distance: 40mm x 15mm
* Screw hole diameter: 1mm (M1)
* Transmitter diameter: 8mm

**5.3 Working Principle :**

The HC-SR04 Ultrasonic sensor comes with four pins namely Vcc pin, Trigger pin, Echo pin, & Ground pin. This sensor is used to measure the accurate distance between the target and the sensor. This sensor mostly works on the sound waves.

When the power supply is given to this module, it generates the sound waves to travel throughout the air to hit the necessary object. These waves strike and come back from the object, then collects by the receiver module.

Here both the distance as well as time has taken is directly proportional because the time taken for more distance is high. If the trigger pin is kept high for 10 µs, then the ultrasonic waves will be generated which will travel at the sound speed. So it creates eight cycles of sonic burst that will be gathered within the Echo pin. This ultrasonic sensor is interfaced with Arduino to gauge the necessary distance between sensor & object. The distance can be calculated using the following formula.

S = (V x t)/2

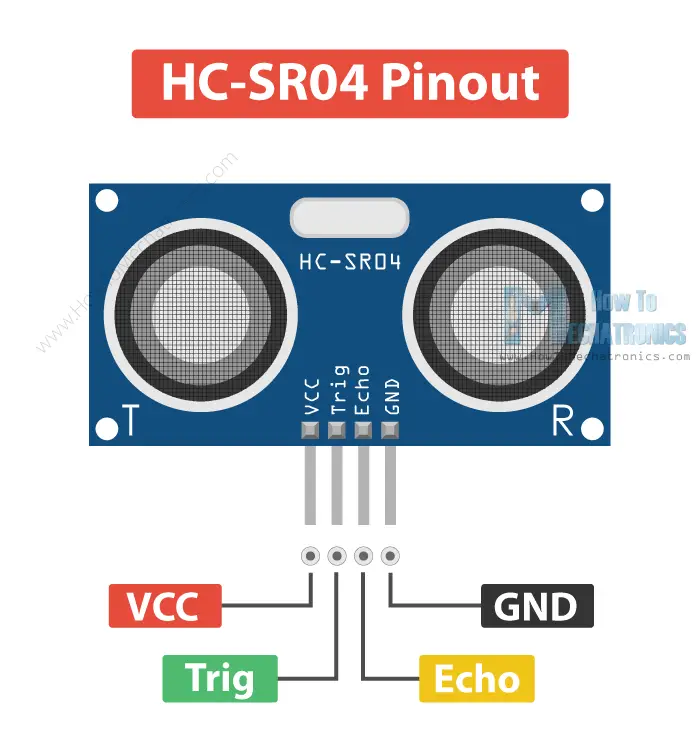
Where the ‘S’ is the required distance

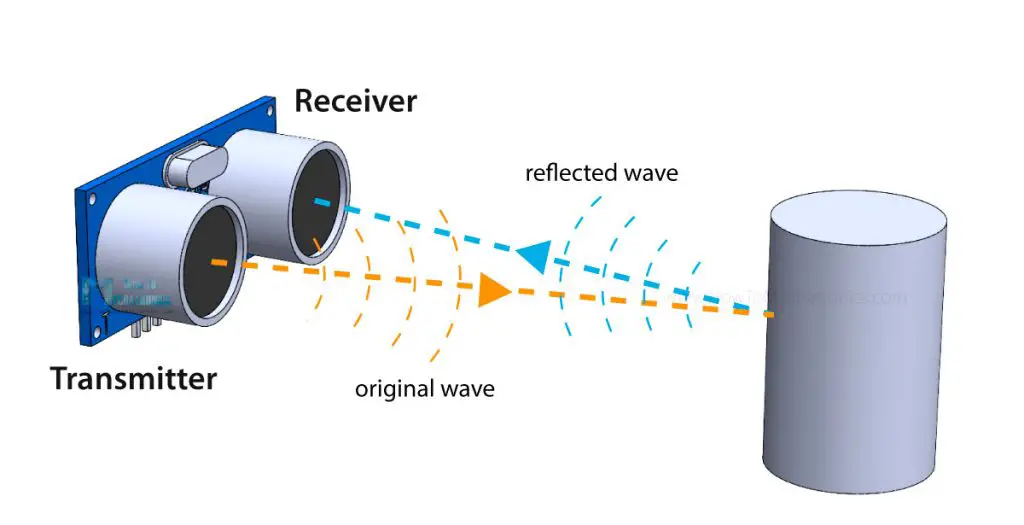
‘V’ is the sound’s speed

‘t’ is the time taken for sound waves to return back after striking the object.

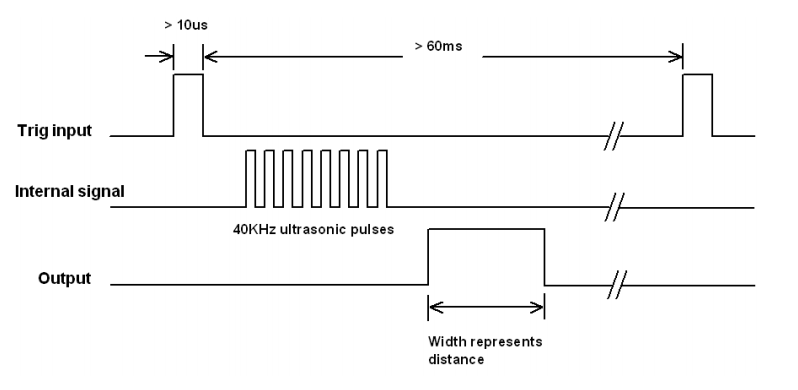
The actual distance can be calculated by dividing its value with 2 as the time will be twice once the waves travel and get back from the sensor.

**5.4 Pinout Configuration & Schematic Diagram :**





Ultrasonic wave that measures distance from the object



Ultrasonic signal (trigger, input and output)

***Chapter – 6* : Rain Sensor Module**

**6.1 Overview :**

The rain sensor module is an easy tool for rain detection. It can be used as a switch when raindrop falls through the raining board and also for measuring rainfall intensity. The module features, a rain board and the control board that is separate for more convenience, power indicator LED and an adjustable sensitivity though a potentiometer.

The analog output is used in detection of drops in the amount of rainfall. Connected to 5V power supply, the LED will turn on when induction board has no rain drop, and DO output is high. When dropping a little amount water, DO output is low, the

switch indicator will turn on. Brush off the water droplets, and when restored to the initial state, outputs high level.

**6.2 Technical Features :**

* Adopts high quality of RF-04 double sided material.
* Area: 5cm x 4cm nickel plate on side,
* Anti-oxidation, anti-conductivity, with long use time;
* Comparator output signal clean waveform is good, driving ability, over 15mA;
* Potentiometer adjust the sensitivity;
* Working voltage 5V;
* Output format: Digital switching output (0 and 1) and analog voltage output AO;
* With bolt holes for easy installation;
* Small board PCB size: 3.2cm x 1.4cm;
* Uses a wide voltage LM393 comparator

**6.3 Working Principle :**

* RAIN DROP SENSOR WORKING

A raindrop sensor has a board on which nickel is coated in the form of lines. It works on the principle of resistance. The principle is that when there is no rain drop on board, the resistance is high so we get high voltage according to V-IR. When raindrops are present the resistance is reduced as water is a conductor of electricity and its presence connects nickel lines in parallel so the reduced resistance and there is a voltage drop across it.

* COPPER PADS

The working of the raindrop sensor module is easy and straightforward. The sensor has a series of exposed copper paths that act as variable resistors whose resistance varies according to the amount of water on its surface. Generally, they are not connected but are bridged through the water. This resistance is inversely proportional to the amount of water. The more water on the surface of the rain pads the better the conductivity and this results in lower resistance. The sensor produces an output voltage which is used to determine whether it is raining or not.

* SENSOR MODULE

The raindrop sensor module consists of a control sensor and a rain-sensing pad which can be connected to any microcontroller. The module produces an output voltage according to the resistance of the sensing pad and is given at the analog output pin. The same signal is also passed over to the LM393 high-precision comparator to digitize and is made available at the TTL digital output pin.

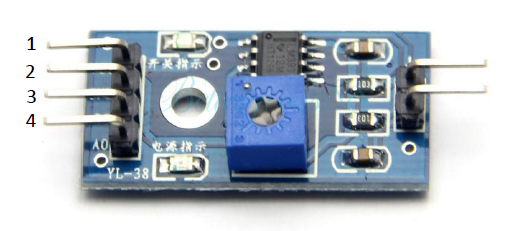
* CONTROL CIRCUIT

In addition, the raindrop module has a potentiometer that is responsible for adjusting the output of the digital pin. This potentiometer is calibrated to receive accurate readings. The potentiometer is connected to the inverting end of the LM393 comparator and sets the reference or threshold voltage meanwhile the input analog voltage is applied to the non-inverting side of the comparator. Then the respective comparator compares both the voltages. This leads to two conditions and gives the output accordingly. If the voltage given as an input is more than the reference voltage then the comparator shows a high state.

* INDICATION LED

Whereas if the threshold voltage is more than the applied voltage then the comparator output will be low. Apart from these, the module has two LEDs, a power LED and a status LED. The power LED will light up when the module is powered and the status LED glows and indicates the digital output pin status.

**6.4 Pinout Configuration & Schematic Diagram :**

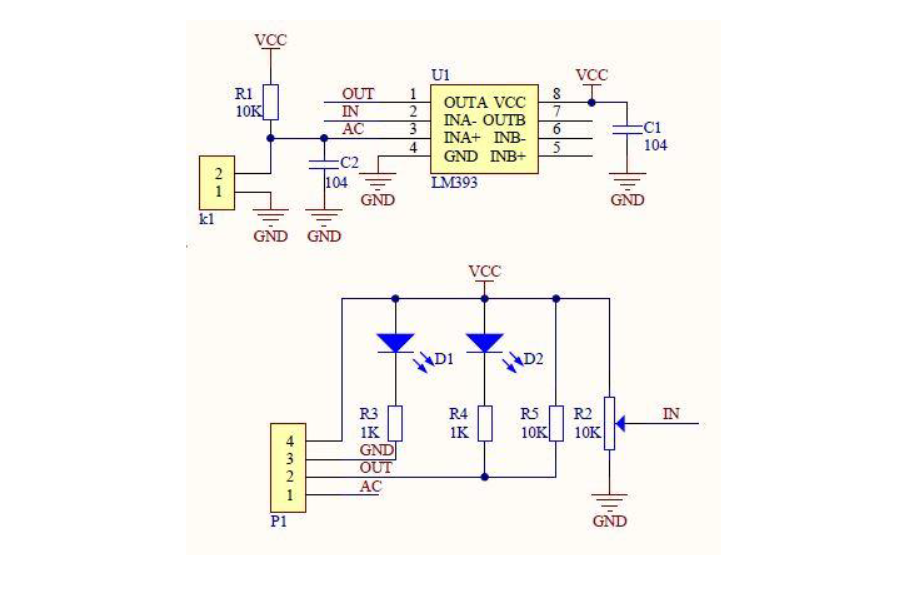


1. VCC: 5V DC

2. GND: ground

3. DO: high/low output

4. AO: analog output



Schematic diagram of rain sensor module

***Chapter – 7* : Water Flow Sensor**

**7.1 Overview :**

Huge industrial plants, commercial and residential buildings require a large amount of water supply. The public water supply system is used to meet this requirement. To monitor the amount of water being supplied and used, the rate of flow of water has to be measured. Water flow sensors are used for this purpose.

Water flow sensors are installed at the water source or pipes to measure the rate of flow of water and calculate the amount of water flowed through the pipe. Rate of flow of water is measured as liters per hour or cubic meters.

**7.2 Technical Features :**

|  |  |
| --- | --- |
| **Parameters** | **Value** |
| Dimensions | 0mm x0mm x0mm |
| Weight | G.W 79g |
| Battery | Exclude |
| Mini. Wokring Voltage | DC 4.5V |
| Max. Working Current | 15mA (DC 5V) |
| Working Voltage | DC 5V~15V |
| Interface Dimensions | G1/2Inch |
| Flow Rate Range | 1~25L/min |
| Frequency | F=(11\*Q)Q=L/MIN±3% |

**7.3 Working Principle :**

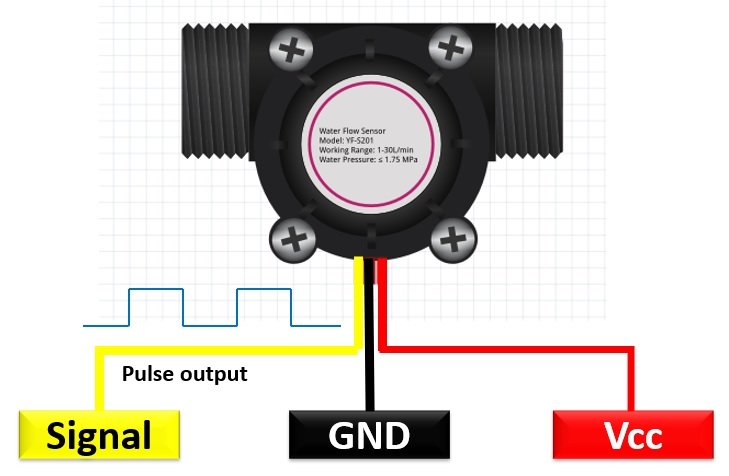
Water flow sensor consists of a plastic valve from which water can pass. A water rotor along with a hall effect sensor is present the sense and measure the water flow. When water flows through the valve it rotates the rotor. By this, the change can be observed in the speed of the motor. This change is calculated as output as a pulse signal by the hall effect sensor. Thus, the rate of flow of water can be measured.

The main working principle behind the working of this sensor is the Hall effect. According to this principle, in this sensor, a voltage difference is induced in the conductor due to the rotation of the rotor. This induced voltage difference is transverse to the electric current. When the moving fan is rotated due to the flow of water, it rotates the rotor which induces the voltage. This induced voltage is measured by the hall effect sensor and displayed on the LCD display. The water flow sensor can be used with hot waters, cold waters, warm waters, clean water, and dirty water also. These sensors are available in different diameters, with different flow rate ranges.

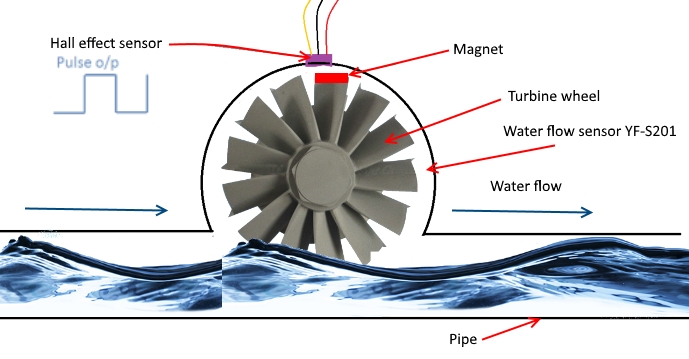
These sensors can be easily interfaced with microcontrollers like Arduino. For this, an Arduino microcontroller board for processing, a Hall effect water flow sensor, a 16×2 LCD display, and Breadboard connecting wires are required. The sensor is placed at the water source inlet or at the opening of the pipe.

The sensor contains three wires. Red wire to connect with supply voltage. Black wire to connect to ground and a yellow wire to collect output from Hall effect sensor. For supply voltage 5V to 18V of DC is required.

**7.4 Pinout Configuration & Schematic Diagram :**



Pin configuration of Water flow sensor



Water flow sensor schematic diagram

***Chapter – 8* : 20 x 4 LCD Display**

**8.1 Overview :**

A liquid crystal display or LCD draws its definition from its name itself. It is a combination of two states of matter, the solid and the liquid. LCD uses a liquid crystal to produce a visible image. Liquid crystal displays are super-thin technology display screens that are generally used in laptop computer screens, TVs, cell phones, and portable video games. LCD’s technologies allow displays to be much thinner when compared to a cathode ray tube (CRT) technology.

Liquid crystal display is composed of several layers which include two polarized panel filters and electrodes. LCD technology is used for displaying the image in a notebook or some other electronic devices like mini computers. Light is projected from a lens on a layer of liquid crystal. This combination of colored light with the grayscale image of the crystal (formed as electric current flows through the crystal) forms the colored image. This image is then displayed on the screen.

**8.2 Technical Features :**

|  |  |
| --- | --- |
| DISPLAY FORMAT | 20 Characters x 4 Lines |
| DOT MATRIX (W X H) | 5 x 8 Dots |
| CHARACTER SIZE (W X H) | 2.95 x 4.75 mm |
| CHARACTER PITCH (W X H) | 3.55 x 5.35 mm |
| LCD DRIVER IC | Sitronix ST7066U (or equivalent) |
| INTERFACE | 4-bit Parallel and 8-bit Parallel |
| LCD MODULE DIMENSIONS (W X H X D) | 98 x 60 x 13.6 (MAX) mm |
| VIEWING AREA (W X H) | 77 x 25.2 mm |
| DOT SIZE (W X H) | 0.55 x 0.55 mm |
| DOT PITCH (W X H) | 0.6 x 0.6 mm |
| DRIVING METHOD | 1/16 Duty |
| OPERATING TEMPERATURE | -20 ~ 70°C |

**8.3 Working Principle :**

The principle behind the LCDs is that when an electrical current is applied to the liquid crystal molecule, the molecule tends to untwist. This causes the angle of light which is passing through the molecule of the polarized glass and also causes a change in the angle of the top polarizing filter. As a result, a little light is allowed to pass the polarized glass through a particular area of the LCD.

Thus that particular area will become dark compared to others. The LCD works on the principle of blocking light. While constructing the LCDs, a reflected mirror is arranged at the back. An electrode plane is made of indium-tin-oxide which is kept on top and a polarized glass with a polarizing film is also added on the bottom of the device. The complete region of the LCD has to be enclosed by a common electrode and above it should be the liquid crystal matter.

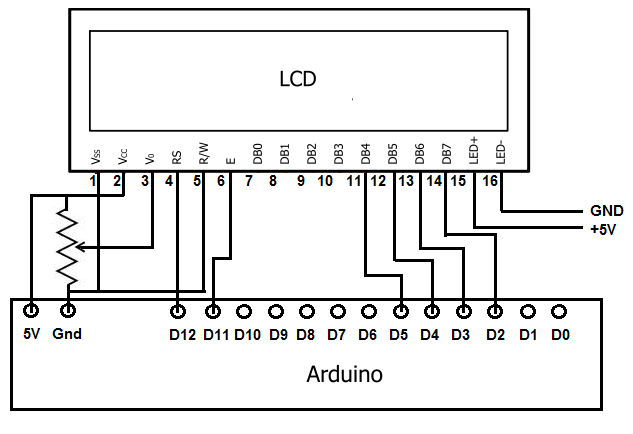
Next comes the second piece of glass with an electrode in the form of the rectangle on the bottom and, on top, another polarizing film. It must be considered that both the pieces are kept at the right angles. When there is no current, the light passes through the front of the LCD it will be reflected by the mirror and bounced back. As the electrode is connected to a battery the current from it will cause the liquid crystals between the common-plane electrode and the electrode shaped like a rectangle to untwist. Thus the light is blocked from passing through. That particular rectangular area appears blank.

**Advantages**

The advantages of liquid crystal display include the following.

* LCD’s consumes less amount of power compared to CRT and LED
* LCD’s are consist of some microwatts for display in comparison to some mill watts for LED’s
* LCDs are of low cost
* Provides excellent contrast
* LCD’s are thinner and lighter when compared to cathode-ray tube and LED

**8.4 Pinout Configuration & Schematic Diagram :**

20x4 LCD display schematic diagram

***Chapter – 9* : I2C Serial Communication Module**

**9.1 Overview :**

I2C Module has an inbuilt PCF8574 I2C chip that converts I2C serial data to parallel data for the LCD display. These modules are currently supplied with a default I2C address of either 0x27 or 0x3F. To determine which version you have check the black I2C adaptor board on the underside of the module. If there a 3 sets of pads labelled A0, A1, & A2 then the default address will be 0x3F. If there are no pads the default address will be 0x27. The module has a contrast adjustment pot on the underside of the display. This may require adjusting for the screen to display text correctly.

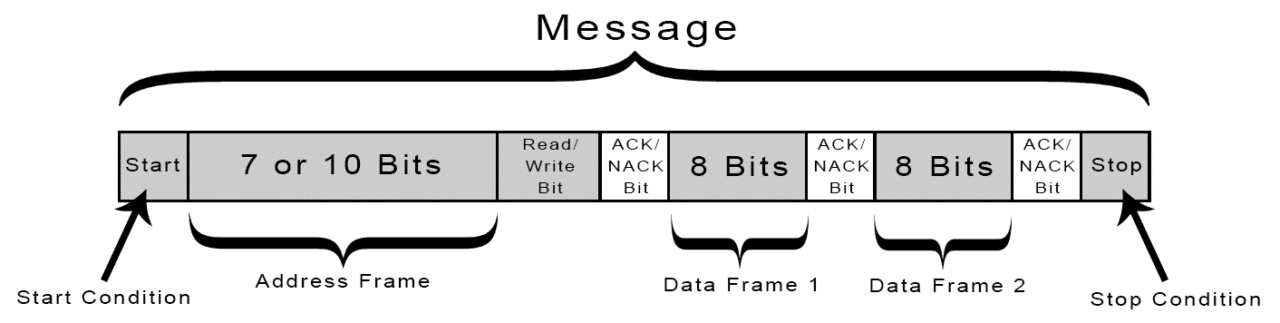
I2C modules are currently supplied with a default I2C address of either 0x27 or 0x3F, you can check which version by verifying underside of the module. If there a 3 sets of pads labelled A0, A1, & A2 then the default address will be 0x3F. If there are no pads the default address will be 0x27.

The module has a contrast adjustment pot on the underside of the display. This may require adjusting for the screen to display text correctly.

**9.2 Technical Features :**

* Display Mode: STN
* Display Format: 16 Character x 2 Line
* Viewing Direction: 6 O’Clock
* Input Data: 4-Bits or 8-Bits interface available
* Display Font : 5 x 8 Dots
* Power Supply : Single Power Supply (5V±10%)
* Driving Scheme : 1/16Duty,1/5Bias
* Backlight (Side) : LED (Yellow)
* I2C Address:0x20-0x27(0x20 default)
* Supply voltage: 5V
* Adjustable contrast

**9.3 Working Principle :**

With I2C, data is transferred in messages. Messages are broken up into frames of data. Each message has an address frame that contains the binary address of the slave, and one or more data frames that contain the data being transmitted. The message also includes start and stop conditions, read/write bits, and ACK/NACK bits between each data frame:

**Start Condition:** The SDA line switches from a high voltage level to a low voltage level before the SCL line switches from high to low.

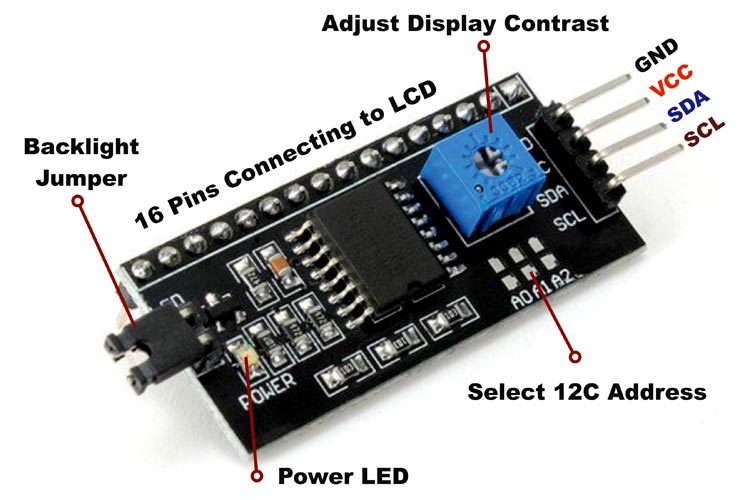
**Stop Condition:** The SDA line switches from a low voltage level to a high voltage level after the SCL line switches from low to high.

**Address Frame:** A 7 or 10 bit sequence unique to each slave that identifies the slave when the master wants to talk to it.

**Read/Write Bit:** A single bit specifying whether the master is sending data to the slave (low voltage level) or requesting data from it (high voltage level).

**ACK/NACK Bit:** Each frame in a message is followed by an acknowledge/no-acknowledge bit. If an address frame or data frame was successfully received, an ACK bit is returned to the sender from the receiving device.

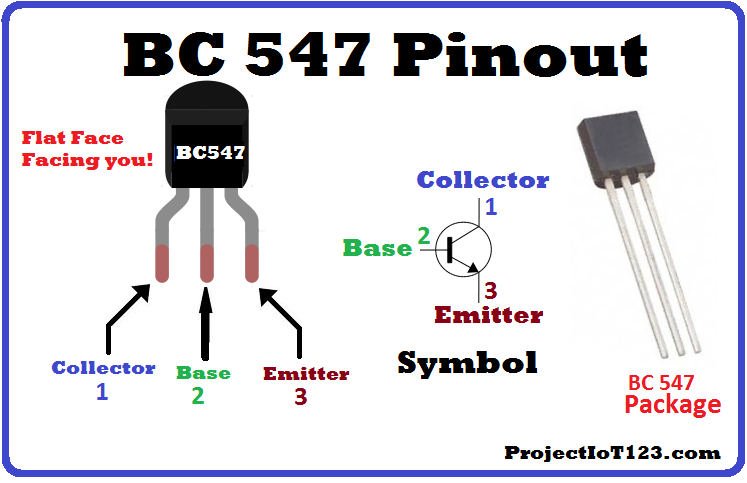
**9.4 Pinout Configuration & Schematic Diagram :**



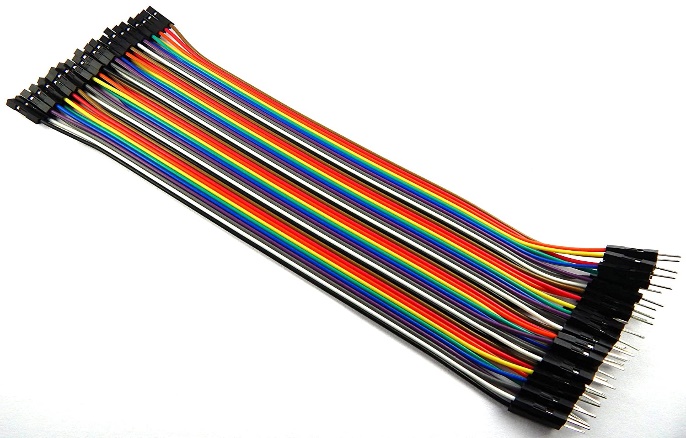
Pin configuration of I2C Serial communication module

***Chapter – 10* : Other Circuitry Elements**

BC547 npn transistor 9V DC battery



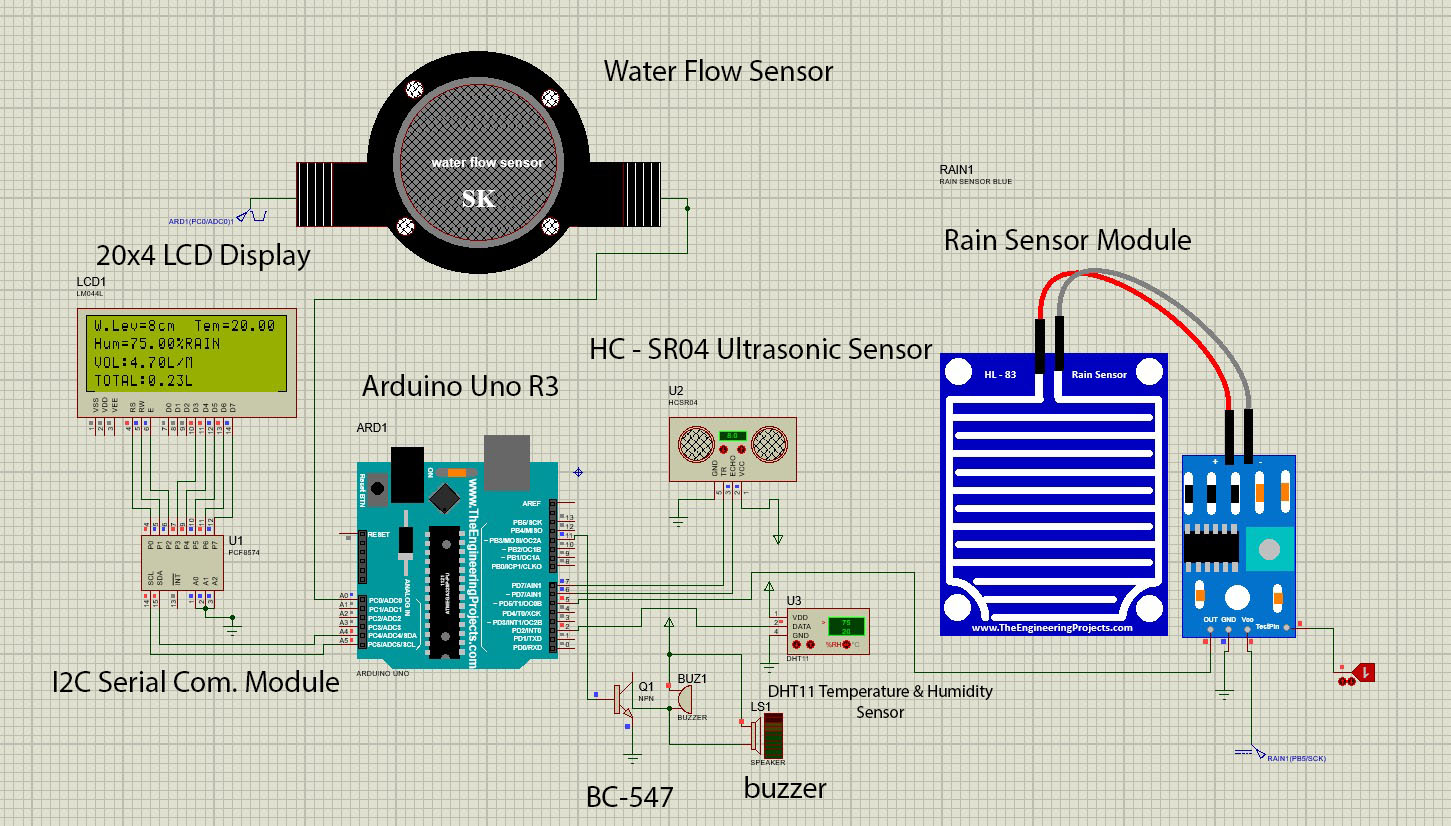
Jumper wire Small Bread Board

 Arduino Cable ESP8266 Cable



***Chapter – 11* : Hardware Portion**

**11.1 Circuit Diagram :**

****

* 1. **Circuit Description :**

The circuit is built around an Arduino Uno board, DHT11 sensor, HC-SR04 ultrasonic sensor, Rain Sensor Module, water flow sensor, ESP8266 NodeMCU and a few components.

The circuit uses one 9V battery and 2 9V AC to DC converter adapter to power all the sensors, Arduino Uno and ESP8266 NodeMCU. LED on the board indicates present of the power supply.

The temperature range of DHT11 is from 0 to 50 degree Celsius with a 2-degree accuracy. Humidity range of this sensor is from 20 to 80% with 5% accuracy. The sampling rate of this sensor is 1Hz .i.e. it gives one reading for every second. DHT11 is small in size with operating voltage from 3 to 5 volts. The maximum current used while measuring is 2.5mA.

DHT11 sensor has four pins- VCC, GND, Data Pin and a not connected pin. A pull-up resistor of 5k to 10k ohms is provided for communication between sensor and micro-controller.

Ultrasonic trans receiver module HC-SR04 uses sonar, like bats and dolphins, to determine distance to an object. it offers excellent noncontact range detection of 2cm to 400cm with high accuracy and stable readings in an easy to use package. It comes complete with an ultrasonic trans receiver module.

To start the measurement of distance, pin 2(TRIG) of the module should receive a high pulse for at least ten micro seconds. The pulse will initiate the module will transmit eight cycles of ultrasonic burst. When the sensor detect the reflected ultrasonic burst, it sets pin 3(ECHO) to high state. Duration of the reflected pulse depends on the distance of the obstacle, which can be easily calculated as: Distance (cm) T/58 where T=width pulse at ECHO pin in micro seconds.

The raindrop module has a potentiometer that is responsible for adjusting the output of the digital pin. This potentiometer is calibrated to receive accurate readings. The potentiometer is connected to the inverting end of the LM393 comparator and sets the reference or threshold voltage meanwhile the input analog voltage is applied to the non-inverting side of the comparator. Then the respective comparator compares both the voltages. This leads to two conditions and gives the output accordingly. If the voltage given as an input is more than the reference voltage then the comparator shows a high state.

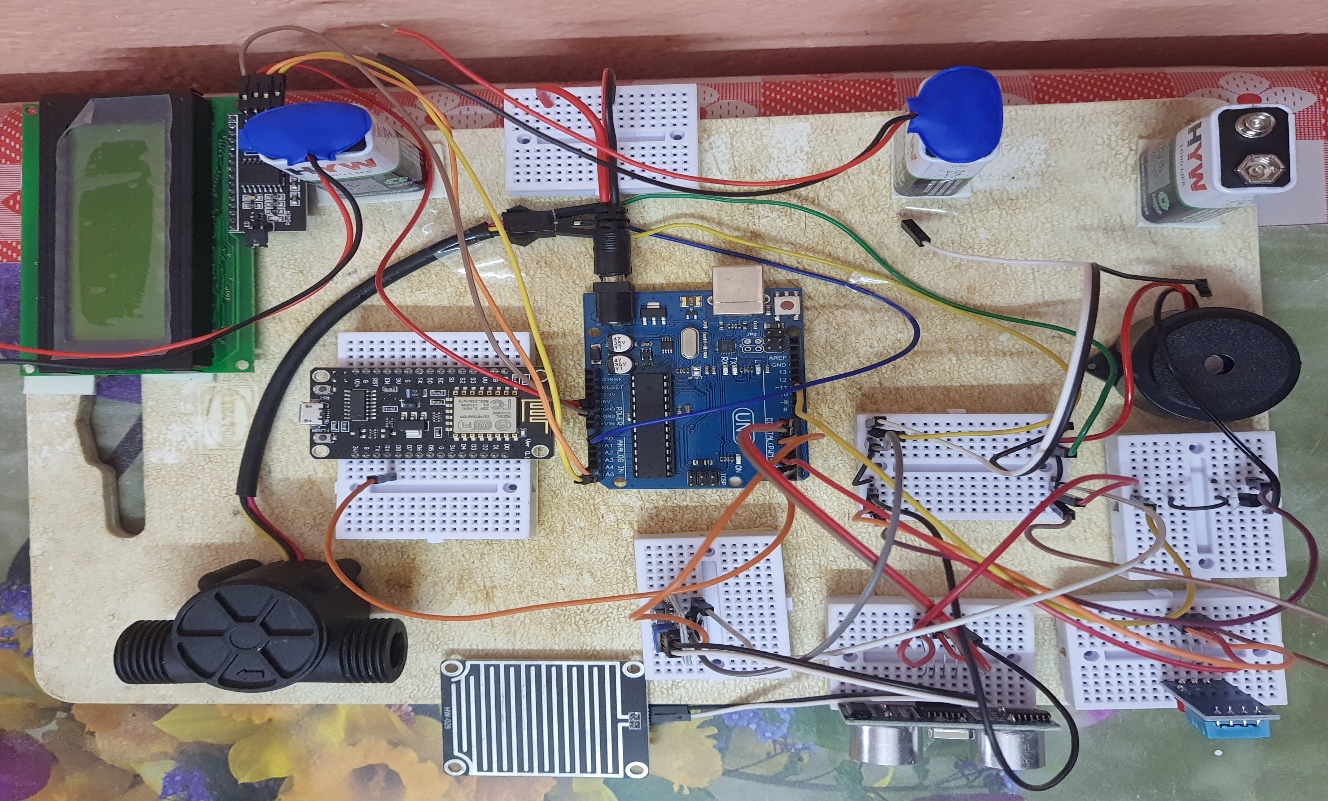
The rain sensor contains three wires. Red wire to connect with supply voltage. Black wire to connect to ground and a yellow wire to collect output from Hall Effect sensor. For supply voltage 5V to 18V of DC is required.

in this sensor, a voltage difference is induced in the conductor due to the rotation of the rotor. This induced voltage difference is transverse to the electric current. When the moving fan is rotated due to the flow of water, it rotates the rotor which induces the voltage. This induced voltage is measured by the Hall Effect sensor and displayed on the LCD display.

Arduino Uno is a board based on Atmega328 microcontroller. It has 14 digital input/output pins, six analogue inputs, a USB connection for programming the on-board microcontroller, power jack, and ICSP header and a reset button. It is operated with 16MHz crystal oscillator and contains everything needed to support the microcontroller.

In Arduino pin Tx is connected to the pin Rx on ESP8266 NodeMCU. Pin 2 is connected with DHT11 sensor, pin 7,6 is connected with the ECHO and TRIG pin of the HC-SR04 ultrasonic sensor. Pin 5 is connected with rain sensor and pin A0 is connected with the water flow sensor. I2C serial communication module has connected with pin 5V, Vcc, A3, A4 pins respectively.

Above mentioned every pin shows in the fig.



* 1. **Working Procedure :**

The five different sensors measure the various environmental and weather-related parameters and monitor them constantly. The data from these sensors is constantly fed to an Arduino controller. The Arduino program constantly checks for any irregularities in the sensor measurements and estimates the weather conditions based on the sensor data.

A Wi-Fi module is also connected to the Arduino controller. The Arduino sends the sensor data to the remote IOT platform using the IOT protocols over the Wi-Fi connection. The LCD is used to display the real-time values of all the sensors. A buzzer is also connected to the output of the Arduino. If the value of any sensor crosses over a certain threshold value, the buzzer is turned on. A GUI is constructed on the remote server IOT platform in order to display the sensor data in a visual format. Using this project, the flood-related parameters can be monitored from anywhere in the world remotely.

*Step -1 :*

Firstly on a rectangular plastic board setting up all the component with small bread boards using a double sided tape. DHT11 sensor is connected to Arduino at pin 2 along with Vcc and GND to the main power bread board. Ultrasonic sensor is connected to pin 7,6 to Arduino along with Vcc and GND to main power board. Water flow sensor and rain module is also connected with Arduino at pin A0 and 5 to Arduino with jumper wire and Vcc and GND are connected to main power board. 0ne 9V battery supply the power to the main power board.

*Step -2 :*

After power it up all the sensors, Arduino Uno board and ESP8266 NodeMCU, immediately sensors start to sense the data from the environment. DHT11 sensor sense the temperature and humidity with interval of every 1 second and send back the data to Arduino board at pin 2. HC-SR04 ultrasonic sensor sends data back to Arduino after calculating the water level distance in cm at pin 7. Rain sensor and water flow sensor sends back the data to Arduino at pin 5 and A0 respectively.

After getting all the data from the sensors Arduino sends the data to 20x4 LCD display which is incorporated with I2C serial communication module. Which shows all the relevant data and same data to ESP8266 NodeMCU which again upload these data to Thingspeak server.

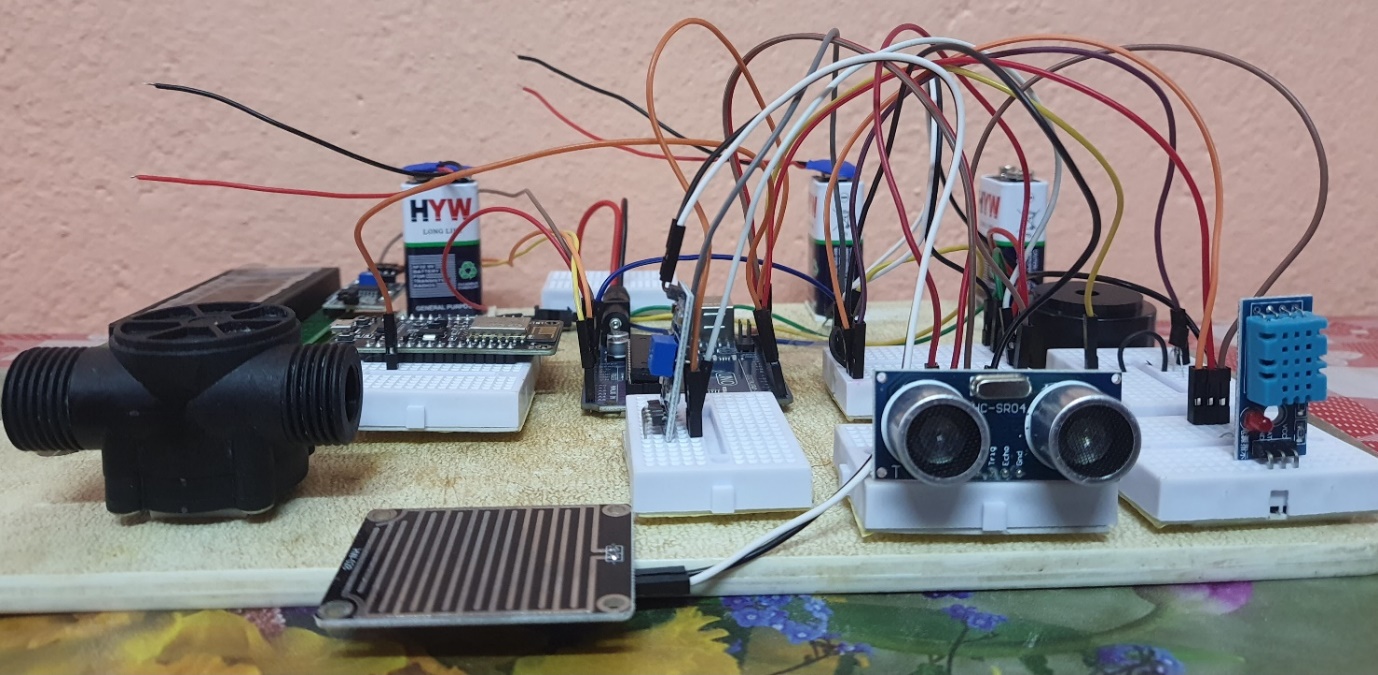
Thingspeak is a public server along the channel ID and different widgets to show how the data is generated and it will help user to calculate data using MATLAB code. As a public server we can access these data from anywhere using any kind of network. There is no need to connect same network to fetch all the data. Using PC, laptop or even in mobile phone also we can fetch and see the real time data sending by the sensors.

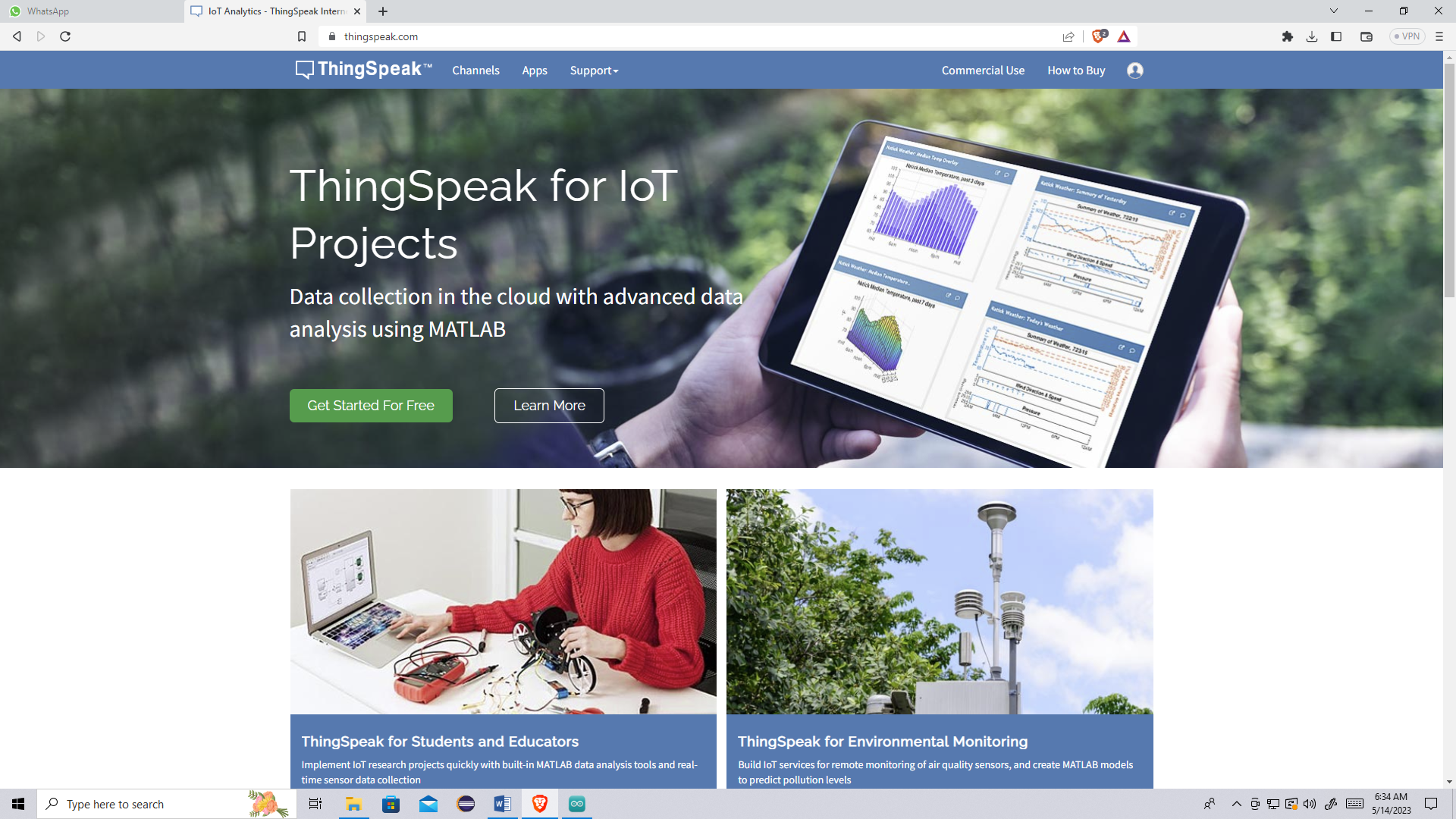
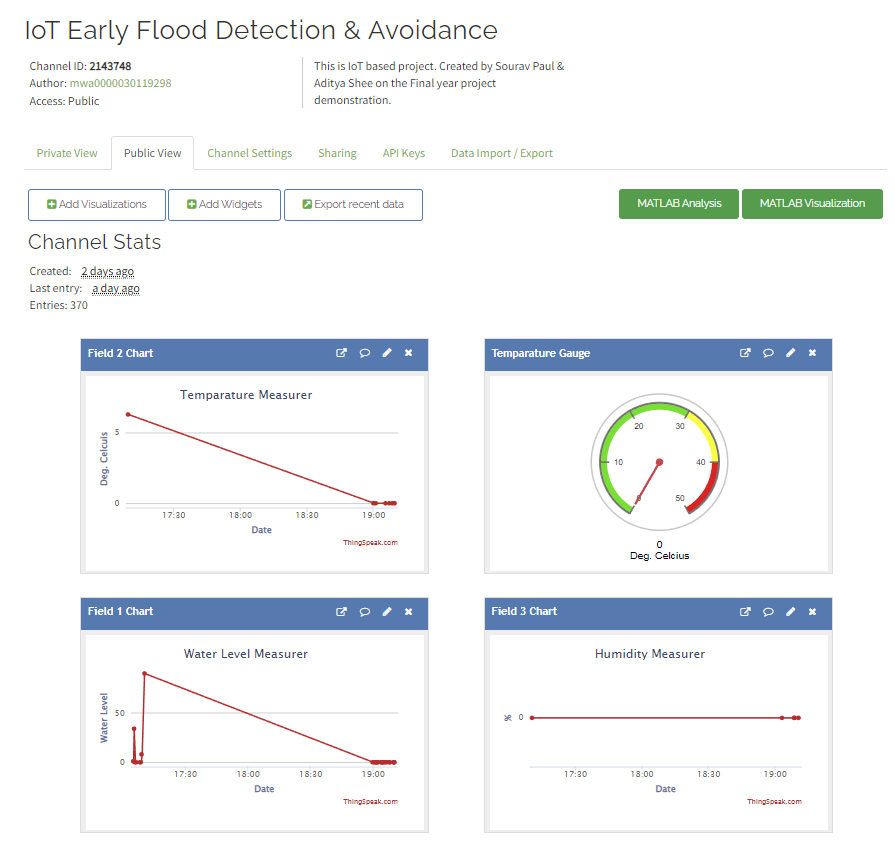
There is a buzzer on the system. If the water level is above, in case of our system, if the water level is below 20cm it will raise an alarm sound.

*Step -3 :*

Loading the software to Arduino Uno and ESP8266 NodeMCU to control all the sensors. Using the software it will accept the data from the sensors and also update these data to Thingspeak server.

* 1. **Technical Specification :**
* The temperature sensor measures the ambient temperature.
* The humidity sensor measures the level of humidity in the air.
* The level sensor measures the level of water.
* The flow sensor measures the flow rate of water.
* The ultrasonic sensor detects objects from a distance.
* The LCD screen is used to display the sensor data in real-time.
* The buzzer is turned on when the sensor value passes the threshold value.
* The Wi-Fi module is used to transmit the sensor data to the remote server-based IOT platform.
* The Arduino controller communicates with all the peripherals and sensors. It stores, processes, and communicates the sensor data.
* The IOT platform is installed on a remote server and is used to log the sensor data and display it in a visual format for remote monitoring.





***Chapter – 12* : Software Portion**

**12.1 Introduction :**

The software for the system is written in Arduino programming language. The Arduino Uno is programmed using Arduino IDE software Atmega328P on Arduino Uno with a boot loader that allows you to upload new code to it without using an external hardware programmer. It uses STK500 protocol to communicate. You can bypass the boot loader and program the microcontroller through ICSP (In circuit serial programming) header, but using boot loader programming is quick and easy to. Select the correct board from ‘Tools -> Board’ menu in Arduino IDE and burn the program (Sketch) through standard USB port in the computer.

**12.2 Programming :**

The Arduino Uno can be programmed with Arduino software (download). Select “Arduino Uno from the tools > Board menu (according to the microcontroller on your board).

The Atmega328P on the Arduino Uno comes pre-burned with a boot loader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol.

The Atmega 16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The Atmega 16U2 / 8U2 is loaded with a DFU boot loader, which can be activated by :

* On Rev1 boards : connecting the solder jumper on the back of the board (nearly the map of Italy) and then resetting the 8U2.
* On Rev2 or later boards : there is a resistor that pulling the 8U2 / 16U2 HWB line to ground, making it easier to put into DFU mode.

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the Atmega 8U2/ 16U2 is connected to the reset line.

**12.3 Source Code :**

*This Source code Arduino Uno Sensor Control*

/\*

IoT Early Flood Detection and Avoidance.

This is Arduino Sender Code To NodeMCU ESP8266.

This is also Arduino Uno code for accepting all the sensors data.

Created by : Sourav Paul & Aditya Shee

-----------------------------------------------------------------------------------

\*/

/\*Attached Library Files

-----------------------------------------------------------------------------------\*/

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#include "DHT.h"

#include <SoftwareSerial.h>

#include <ArduinoJson.h>

//Define Arduino Pins and Creating Objects

LiquidCrystal\_I2C lcd(0x27,20,4);    // set the LCD address to 0x27 for a 20 chars and 4 line display

int trig=7;

int echo=6;

int safedistance;

int TimeMicro;

int Distance;

#define DHT11\_PIN 2

#define DHTTYPE DHT11

#define buzzer 11

DHT dht(DHT11\_PIN, DHTTYPE);

float t;

float h;

int X;

int Y;

float TIME = 0;

float FREQUENCY = 0;

float WATER = 0;

float TOTAL = 0;

float LS = 0;

const int input = A0;

/\*Void Setup Function

-----------------------------------------------------------------------------------\*/

void setup()

{

  Serial.begin(9600);

  lcd.init();                    // initialize the lcd

  lcd.backlight();               // turn on LCD backlight

  pinMode(7,OUTPUT);             // trigger pin for ultrasonic sensor

  pinMode(6,INPUT);              // echo pin for ultrasonic sensor

  dht.begin();                   // function begin for DHT11 sensor

  pinMode(input,INPUT);          // pinmode for water flow sensor

  pinMode(5, INPUT);             // pinmode for rain sensor module

}

/\* void loop function

-----------------------------------------------------------------------------------\*/

void loop()

{

  /\*function body for ultrasonic sensor\*/

  digitalWrite(trig,LOW);

  delayMicroseconds(2);

  digitalWrite(trig, HIGH);

  delayMicroseconds(10);

  digitalWrite(trig,LOW);

  TimeMicro=pulseIn(echo,HIGH);

  Distance=TimeMicro/29/2;

  safedistance=Distance;

  /\*logic for buzzer\*/

  if(safedistance<=20){

    digitalWrite(buzzer,HIGH);

  }

  else{

    digitalWrite(buzzer,LOW);

  }

  /\*logic to print data to LCD display\*/

  lcd.setCursor(0, 0);

  lcd.print("W.Lev=");

  lcd.print(Distance);

  lcd.print("cm");

  delay(2000);

  /\*logic for DHT11 sensor\*/

  h = dht.readHumidity();

  t = dht.readTemperature();

  lcd.setCursor(11, 0);

  lcd.print("Tem=");

  lcd.print(t);

  lcd.setCursor(0, 1);

  lcd.print("Hum=");

  lcd.print(h);

  lcd.print("%");

  delay(2000);

  /\*logic for water flow sensor\*/

  X = pulseIn(input, HIGH);

  Y = pulseIn(input, LOW);

  TIME = X + Y;

  FREQUENCY = 1000000/TIME;

  WATER = FREQUENCY/7.5;

  LS = WATER/60;

  if(FREQUENCY >= 0)

    {

      if(isinf(FREQUENCY))

        {

          lcd.clear();

          lcd.setCursor(0,2);

          lcd.print("VOL:0.0");

          lcd.setCursor(0,3);

          lcd.print("TOTAL:");

          lcd.print(TOTAL);

          lcd.print("L");

        }

      else

        {

          TOTAL = TOTAL + LS;

          lcd.clear();

          lcd.setCursor(0,2);

          lcd.print("VOL:");

          lcd.print(WATER);

          lcd.print("L/M");

          lcd.setCursor(0,3);

          lcd.print("TOTAL:");

          lcd.print(TOTAL);

          lcd.print("L");

        }

    }

delay(2000);

  /\*logic for rain sensor module\*

int rain = digitalRead(5);

if(rain==1)

{

  lcd.setCursor(11, 1);

  lcd.print("RAIN");

   delay(1000);

}

else if(rain==0)

{

  lcd.setCursor(11, 1);

  lcd.print("NO RAIN");

   delay(1000);

}

/\*logic for serial monitor printing in Arduino IDE\*/

Serial.print(Distance);

Serial.print(",");

Serial.print(t);

Serial.print(",");

Serial.print(h);

Serial.print(",");

Serial.print(WATER);

Serial.print(",");

Serial.print(TOTAL);

Serial.print(",");

Serial.print(rain);

Serial.println();

delay(1000);

}

*This Source code ESP8266 NodeMCU to Transfer Data To Thingspeak Server.*

/\*

IoT Early Flood Detection and Avoidance.

This is Reeceiver NodeMCU ESP8266 code from Arduino Uno.

This is code for accept data from Arduino and send them to ThingSpeak Cloud

Created by : Sourav Paul & Aditya Shee

-----------------------------------------------------------------------------------

\*/

/\*Attached Library Files

-----------------------------------------------------------------------------------\*/

#include <ESP8266WiFi.h>

#include <WiFiClient.h>

#include <ThingSpeak.h>

const char\* ssid = "F62\_HotSpot";                           // Your Network SSID

const char\* password = "Sourav@120101";                    // Your Network Password

WiFiClient client;

unsigned long myChannelNumber = 2143748;                   //Your Channel Number (Without Brackets)

const char \* myWriteAPIKey = "XUFZ6AWYDADO3G2Q";          //Your Write API Key

int d,r;

float t,h,w,to;

void setup() {

  Serial.begin(9600);

  delay(10);

  // Connect to WiFi network

  WiFi.begin(ssid, password);

  ThingSpeak.begin(client);

}

void loop() {

    if(Serial.available()>0){

      String data=Serial.readStringUntil('\n');

      d=data.substring(0,2).toInt();

      t=data.substring(3,8).toFloat();

      h=data.substring(9,14).toFloat();

      w=data.substring(15,18).toFloat();

      to=data.substring(19,26).toFloat();

      r=data.substring(27).toInt();

      Serial.println(d);

      Serial.println(t);

      Serial.println(h);

      Serial.println(w);

      Serial.println(to);

      Serial.println(r);

      delay(1000);

    }

    ThingSpeak.writeField(myChannelNumber,1,d,myWriteAPIKey);

    ThingSpeak.writeField(myChannelNumber,2,t,myWriteAPIKey);

    ThingSpeak.writeField(myChannelNumber,3,h,myWriteAPIKey);

    ThingSpeak.writeField(myChannelNumber,4,w,myWriteAPIKey);

    ThingSpeak.writeField(myChannelNumber,5,to,myWriteAPIKey);

    ThingSpeak.writeField(myChannelNumber,6,r,myWriteAPIKey);

    delay(1000);

}

***Chapter – 13* : Application**

Dams bring water to livestock and irrigation and supply many industries. They also play a pivotal role in flood control and can assist river navigation, so it’s crucial that dams and reservoirs function properly and their water levels are safe. The trouble is that traditional monitoring methods are time-consuming and complex.

Water level monitoring and management of dams using IoT can improve this, using ultrasonic, vibration, and pressure sensors to help monitor dam function. With pressure sensors, in particular, you can detect leaks in pipes and receive instant alerts. Predictive technologies ensure dam operators get early warnings and are able to keep watch over water availability in each reservoir. This may be particularly helpful for irrigation.

A smart solution can also give you remote control over the movement of gates, so there’s no need to send staff to the site in severe weather conditions like floods or storms. If the water reaches a certain level, the system can decide to open or close the gate.

### ****Main benefits****

* Real-time water level monitoring
* Better dam functionality
* Enhanced dam reliability
* Faster decision-making
* Saved time and resources
* No human involvement

***Chapter – 14* : Future Work**

With further improvement to the system we will be able to predict even more accurately and reduce the amount of false alerts. Adding to that with further modification we will be able to predict how much area around the river bank will be affected due to the water level which will be crucial so that the resident of that area could be evacuated to a safe place before flood actually causes any damage.

We can give solar power to this system, which will help in the conservation of power, and it will also be useful in remote areas. On large scale, we can connect multiple sides with a common server. We can also perform field tests to observe the communication process between authorities and commoners. If implemented on a large scale, then we can add a feature that alerts the authorities directly through SMS.

***Chapter – 15* : Conclusion**

Flood Detection System tries to help all kinds of people to be aware of the damages that could harm them. Be it related to farming, industries, or even normal residents; this system is trying to alert everyone. Since this natural phenomenon is uncalled for, we have to take precautions for our betterment.

The system that we are proposing will be able to detect and monitor the flood for the flood prone areas. The system uses ultrasonic sensor in order to sense the level of the water at different areas. Our System consists of major components like Sensor Networks, transporting module and cloud server. The sensor network will sense the current level of water and then send it to the coordinating node of the Wireless Sensor Network which will send the data to the cloud which implements a Regression Algorithm which will help to predict the future course of in terms of level of water. Based on the final calculation warning will be issued to the nearby people we used WSN nodes due to their tiny size, low power consumption and cost effective. Our system will play a vital role in providing warning to the people which are living in the flood prone areas, Our system will be able to accurately predict the danger of flood however in order to enhance the efficiency of the system we can increase the number of sensors we deployed in the geographical areas .

***Chapter – 16* : References**

[1] Available : <https://nevonprojects.com/iot-early-flood-detection-avoidance/>

[2] Available : <https://www.electricaltechnology.org/2021/03/early-flood-detection-system.html>

[3] Available : <https://www.projectsof8051.com/iot-early-flood-detection-and-avoidance-using-arduino/>

[4] Available : <https://www.researchgate.net/publication/352368929_IOT_Early_Flood_Detection_Alerting_System_using_Arduino>

[5] Available : <https://www.ijeast.com/papers/155-158,Tesma601,IJEAST.pdf>

[6] asmin Maurya, Hemlata Pant, Shivani Dwivedi, Muskan Jaiswal Department of Computer Science & Engineering Babu Banarasi Das Institute of Technology & Management Lucknow, Uttar Pradesh, India.

[7] Available : <https://www.ijsdr.org/papers/IJSDR1912007.pdf>

[8] 1SIMRAN CHAUS, 2LALIT WAGH, 3PRATHMESH DUBEY, 4 DURGESH KARPE, 5Dr.K.C.NALAVADE 1,2,3,4BE COMPUTER ENGINEERING, 5PHD COMPUTER ENGINEERING, SANDIP INSTITUTE OF ENGINEERING & MANAGEMENT, NASHIK, INDIA