**FLOOD MONITORING AND EARLY WARNING SYSTEM**

**INTRODUCTION TO FLOOD MONITORING AND EARLY WARNING SYSTEM:**

**The technology has been developed gigantically in order to prevent loss of life. This project is totally based on informing the civilians about the upcoming flood so that they can evacuate the danger area before the flood hits.**

**For detecting the rise in water level Ultrasonic Sensor and Water Level Sensor is used. For detecting the change in humidity and temperature Humidity and Temperature Sensor is used. The data from the DTH11 and HC-SR04 is read by the microprocessor and analyze the data in order to detect the level of water.**

**If the level of water is less than the defined threshold value then the microprocessor turns the LED and buzzer on.**

**Furthermore, the data obtained from the microprocessor is uploaded to the cloud. The values of the sensors updating in real time can be monitored in cloud platform.**

**The data of the cloud is now linked with the web API (Application Programming Interface) and trigger is set. And now when the level of water crosses the threshold value the trigger is triggered and the web API sends the SMS to the phone number registered to it**

PROJECT OBJECTIVES:

The main objective of this project is to develop and design a flood detection system that will detect flood automatically and send data to the Local Government Unit and to residents using an Arduino. Specific Objectives

* To design a circuit and create a programming code using the microcontroller.
* To apply the Serial Communication in transmitting the data from one place to another place.
* To detect the current level of the flood where the system sensor will be divided into four levels
* Flood warning systems have the objective of mitigating the detrimental effects of flooding and reducing the casualties and damages caused by flood disasters.
* These systems aim to provide early warnings to communities and authorities, allowing for timely evacuation procedures and the saving of lives and properties.
* They integrate various sensors and monitoring devices to forecast flow rate, water level, and flood status.
* The development of flood warning systems involves the use of prototyping models, data analysis, design, and implementation phases. These systems utilize technologies such as Global System for Mobile Communication (GSM), Short Message Service (SMS), and cloud storage for real-time data visualization.
* The ultimate goal of flood warning systems is to minimize flood damages and provide reliable information for flood risk reduction measures, policy making, and agricultural insurance.

IOT DEVICES

1. ESP32
2. ULTRASONIC SENSOR HRSC-04
3. DHT22
4. BUZZER
5. LED
6. JUMPER WIRES

DEVICE SETUP :

Setting up an IoT project using an ESP32 board, ultrasonic sensor, DHT22 Sensor, led, buzzers and jumpers involves several steps. I'll provide an overview of how you can set up this project, but please note that this is a complex project, and you may need to consult specific documentation and libraries for each component. Additionally, coding this project will require programming skills in platforms like Arduino IDE.

1. Gather the Required Components:

* ESP 32
* Ultrasonic sensor (e.g., HC-SR04).
* DHT22
* Buzzer
* Led
* Jumper wires
* Power supply for the servo motor if needed

1. Connect the Ultrasonic Sensor:

* Connect the VCC pin of the ultrasonic sensor to the 3.3V output of voltage pin
* Connect the GND pin of the ultrasonic sensor to the GND of microprocessor
* Connect the TRIG pin of the ultrasonic sensor to an analog input pin(eg..D13)
* Connect the ECHO pin of the ultrasonic sensor to an analog input pin(eg.D15)

3.CONNECT THE DHT22 SENSOR

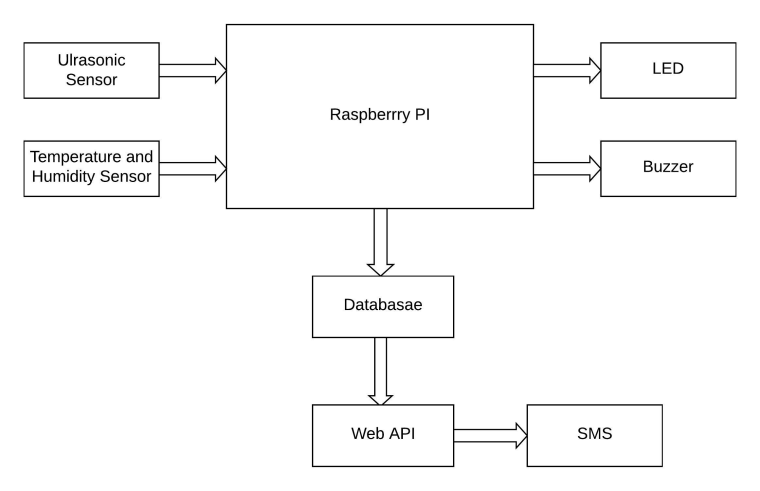
|  |  |
| --- | --- |
| * VCC | Positive voltage of DHT22 Connect to the 3v pin in esp32 |
| * SDA | Digital data pin (input/output) connect to the analog pin |
| * NC | Not connected |
| * GND | Ground in ground state |

4.CONNECT THE LED AND BUZZER

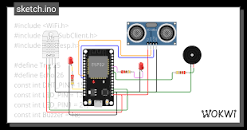
* Connect the LED digital input analog pin
* Connect the Buzzer digital input analog pin

**System Block Diagram**

The block diagram of the overall system is shown in the figure below. The sensors placed at the different places reads the data which are manipulated through the microcontroller and the values of the sensors are displayed. The values obtained repeatedly are send to database and through the web API, warning message is sent into the phone.



HARDWARE CONNECTION



#include <ThingSpeak.h>

#include <WiFi.h>

#include "DHT.h"

#define DHTPIN  15                      //here we are initialising a pin for DHT22

#define DHTTYPE DHT22

#define ledPin 14

#define CM\_TO\_INCH 0.393701

DHT dht(DHTPIN, DHTTYPE);

const int trigPin = 23;

const int echoPin = 22;

int statusCode;

// defines variables

long duration;

float distance;

float distanceInch;

float Humidity;

float Temperature;

// wifi

const char \*ssid =  "Wokwi-GUEST"; //your network SSID (name)

const char \*pass =  ""; //your network password

WiFiClient client;

//thingspeak settings

unsigned long mychannelNumber = 2308333; //your channel ID number\*\* dari channel thingspeak yg telah kita buat

const char \*myWriteAPIkey = "FMXC5QW3BFQ8TI6L"; //your channel write API Key

//int lum, i = 0;

void setup() {

    pinMode(ledPin, OUTPUT);

    pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output

    pinMode(echoPin, INPUT); // Sets the echoPin as an Input

**Serial**.begin(115200);

    delay(10);

    // Initialize the DHT sensor

  dht.begin();

    //connect to WiFi

**Serial**.print("Connecting to: "); **Serial**.println(ssid);

    WiFi.begin(ssid, pass);

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

      delay(500);

**Serial**.print(".");

    }

**Serial**.println("\nWiFi connected\n");

    ThingSpeak.begin(client); //initialize ThingSpeak

}

void loop() {

  float T = dht.readTemperature();

  float H = dht.readHumidity();

// Clears the trigPin

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

// Sets the trigPin on HIGH state for 10 micro seconds

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

// Reads the echoPin, returns the sound wave travel time in microseconds

duration = pulseIn(echoPin, HIGH);

// Calculating the distance cm

distance= duration\*0.034/2;

distanceInch = distance \* CM\_TO\_INCH;

 if(distance<20){

    digitalWrite(ledPin, HIGH);

    delay(1000);

    digitalWrite(ledPin, LOW);

    delay(100);

  }

  else{

    digitalWrite(ledPin, LOW);

  }

// Prints the distance on the Serial Monitor

**Serial**.print(("Humidity: "));

**Serial**.print(H);

**Serial**.print(("%  Temperature: "));

**Serial**.print(T);

**Serial**.print("Water level (Cm): ");

**Serial**.println(distance);

// i++;

 // lum = analogRead(34);

       if(distance<20){

**Serial**.println("WARNING!");

**Serial**.println("waterlevel(cm): " + String(distance));

  }

  else{

**Serial**.println("SAFE");

  }

  //tahan selama 1 detik, program tidak menjalankan yang lain

  delay(2000);

writeData();

}

void writeData()

{

  float T = dht.readTemperature();

  float H = dht.readHumidity();

  ThingSpeak.setField(1, distance);

  ThingSpeak.setField(2, Temperature);

  ThingSpeak.setField(3,Humidity);

statusCode = ThingSpeak.writeFields(mychannelNumber,myWriteAPIkey);

if(statusCode == 200) //successful writing code

**Serial**.println("Channel update successful.");

else

**Serial**.println("Problem Writing data. HTTP error code :" +

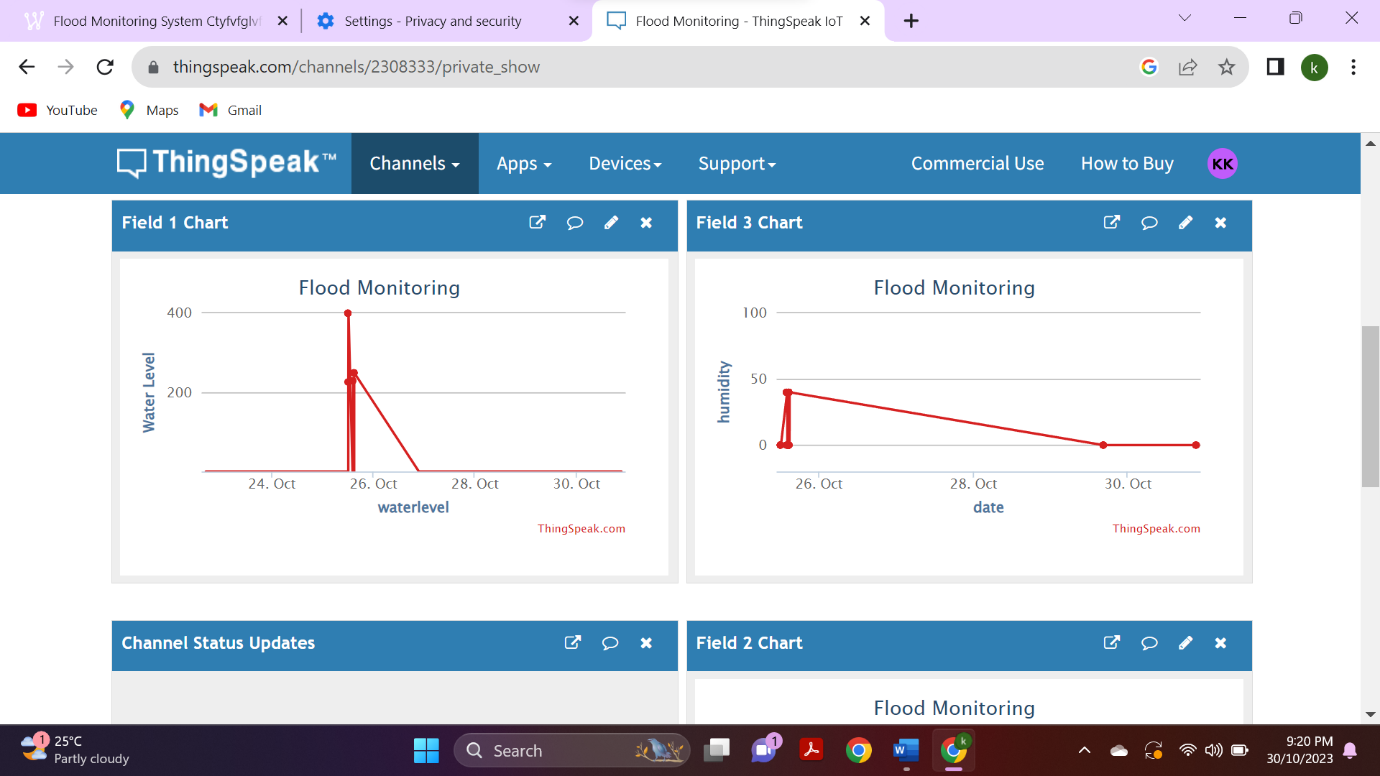
String(statusCode));

delay(5000); // data to be uploaded every 15secs

}

CONNECTING TO THE CLOUD

In wokwi platform simulation happen then it sends data into the ThingSpeak . It is an IoT analytics platform service that allows you to aggregate, visualize, and analyze live data streams in the cloud. You can send data to ThingSpeak from your devices, create instant visualization of live data, and send alerts.



DEVELOP APK USING MIT APP INVENTOR

For every simulation in the wokwi platform the data can be update into personal channel created in the Thingspeak. We can use the data to know the difference level daily update and also live stream the data into the API interface using MIT APP INVENTOR

Using MIT app inventor we have to create app that can be named as FAMS(Flood Monitoring ALERT SYSTEM)

FAMS can be used to monitor and regular update from the cloud system

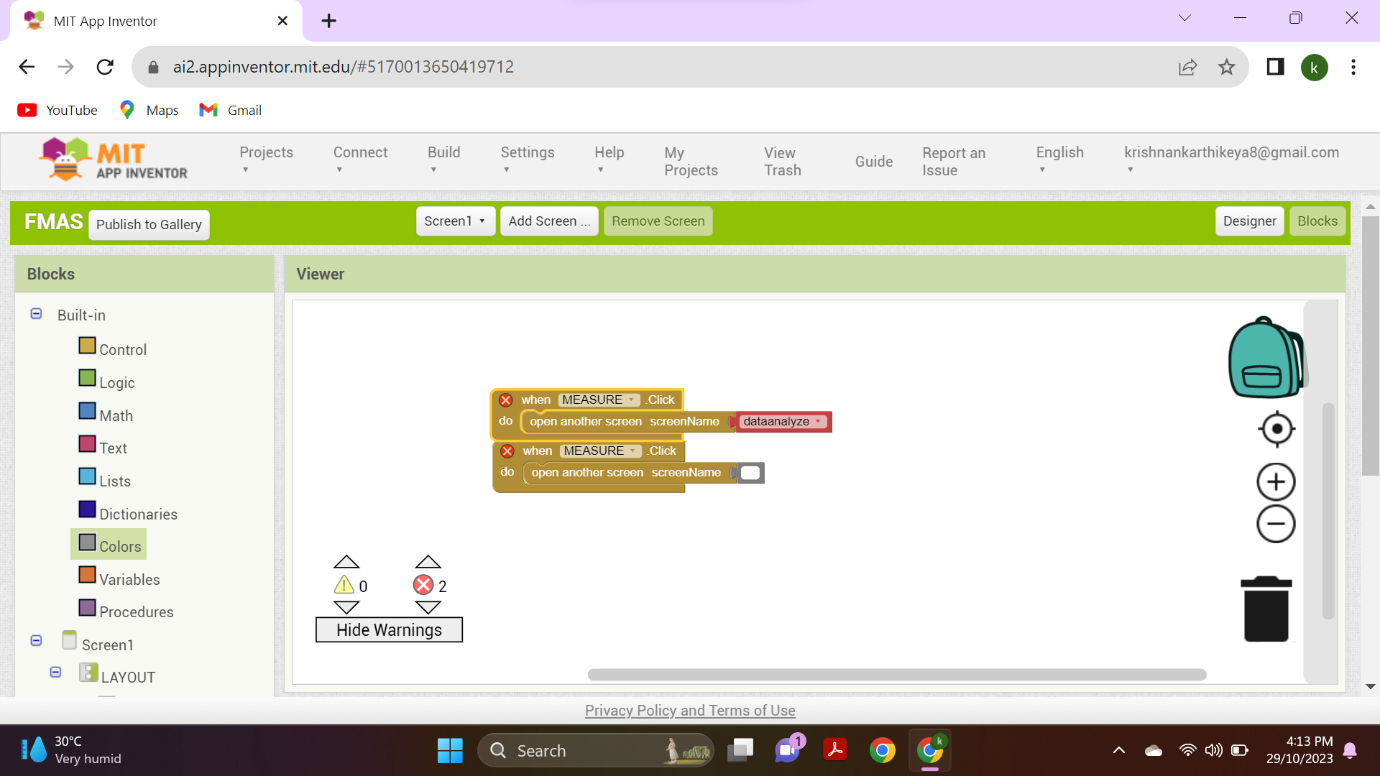
FAMS contains two screen

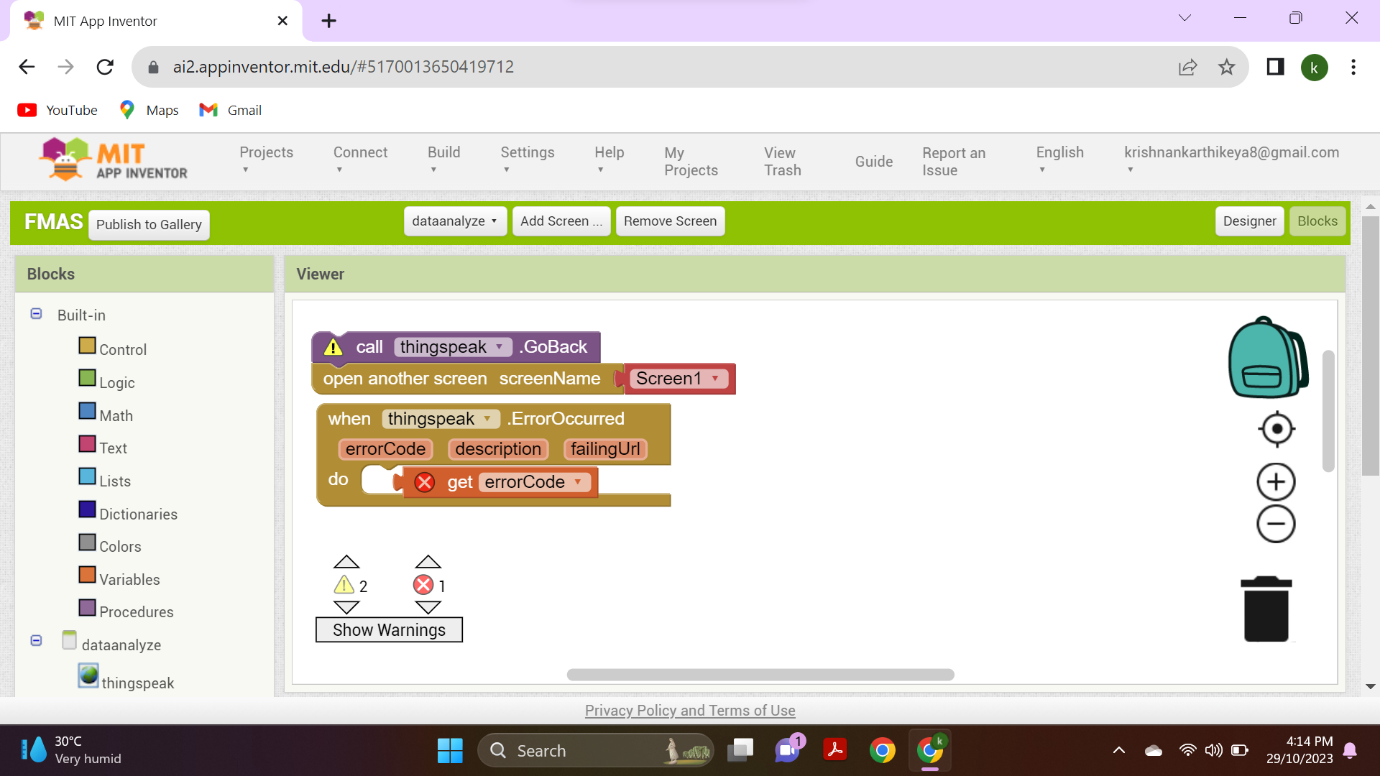
SCREEN 1

It is the open Desktop for the FAMS

DATA ANALYST

It is used to collect the data from the cloud and provide alert message to the device

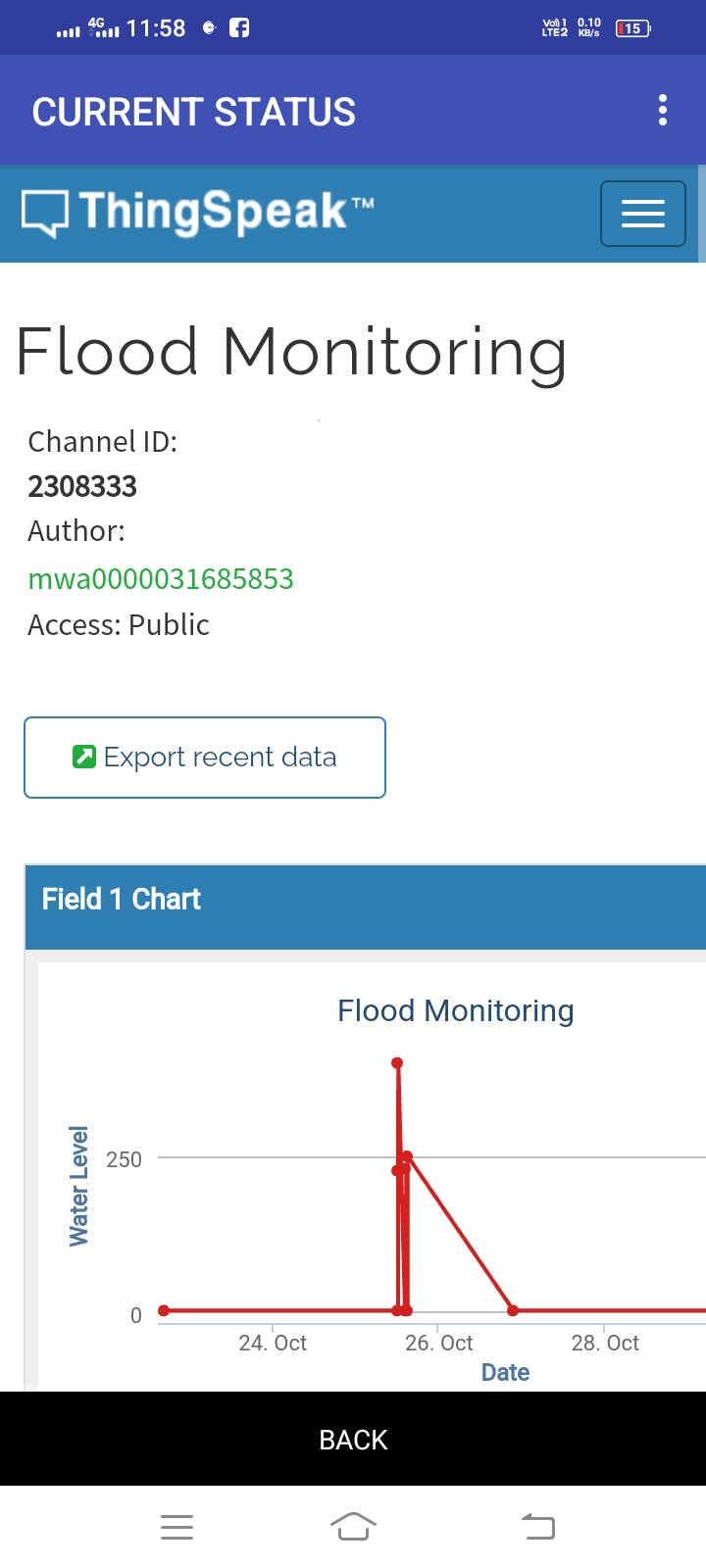




INSTALL THE APK MODE INTO THE MOBILE

1. USER INTERFACE





DATA ARRIVED FROM THINGSPEAK CONNECTED THROUGH THE MIT APP INVENTOR .

BENEFITS OF FLOOD MONITORING AND ALERT SYSTEM:

Because of the devastating effects that floods can have on people and their environments, flood monitoring systems have been developed . The systems can help prevent excessive damage and loss as a result of flooding and possibly save lives.

* High reliability as data is sent in real-time
* Timely detection of flood risks
* Tailored solutions that can easily be integrated with external development at any level such as connectivity, user application, and device.
* Data collected using a flood monitoring system can be used in future studies for weather patterns and climate change.
* Flood monitoring systems also have total integration and adaptation with emergency plans. They consume less energy and can be powered with solar energy. Flood monitoring system equipment is highly resilient and has a long working life span, making it very convenient and cost-effectively
* **challenges Faced in Flood Monitoring**
* Flood risk information can be acquired via past flood records that occurred in the area, surveying assets and people exposed to floods, and the use of predictive modeling.
* The use of historical flood records is often unavailable. To understand the vulnerable areas, expensive mapping and surveys need to be carried out and have to be constantly updated with dynamic urban growth. Predictive modeling also requires the collection of data sets such as land use, topography, exposure, and soil types.
* This task requires the availability of highly skilled staff who can run simulations and analyze the collected data. In rural areas, flood monitoring systems may not be available and rely on human observers as opposed to sensors.
* **New Developments and the Future of Flood Monitoring**
* Many western European countries were recently subjected to severe flooding which resulted in huge economic losses and multiple deaths. Floods have become a frequent threat as a result of climate change, but with the development of artificial intelligence (AI), a much more advanced flood detection system has been developed based on deep computer algorithms.
* The development of smart cameras and sensor technology helps detect the water levels and measure the probability of flooding even before the floods happen.
* With much more research being carried out to improve flood monitoring systems, many governments are investing in much quicker methods of detection. Aside from real-time analysis, visualization will also help emergency authorities to showcase current conditions, allowing citizens to take action and prepare adequately.
* **References and Further Reading**
* Linkwise Technology Group of Companies. n.d. *Flood Monitoring System | Applications*. [online] Available at: [https://linkwisetech.com/applications/flood-monitoring-system](https://linkwisetech.com/applications/flood-monitoring-system/)
* Crime Intervention Alarm. n.d. *The Importance of Flood Monitoring - Crime Intervention Alarm*. [online] Available at: <https://cialarm.com/2018/01/31/importance-flood-monitoring/>
* B-Air Blowers, Air Movers & Dehumidifiers | Commercial Equipment. 2018. *What Causes Floods? | The 8 Most Common Causes of Flooding*. [online] Available at: <https://b-air.com/2018/02/common-causes-flooding/>
* Nps.gov. n.d. *Flooding and Climate Change (U.S. National Park Service)*. [online] Available at: <https://www.nps.gov/articles/000/flooding-climateimpact.htm>
* NexSens. 2019. *Flood Warning Systems*. [online] Available at: <https://www.nexsens.com/blog/flood-warning-systems.htm>
* Denchak, M., 2019. *Flooding and Climate Change: Everything You Need to Know*. [online] NRDC. Available at: <https://www.nrdc.org/stories/flooding-and-climate-change-everything-you-need-know>

Conclusion

The study is all about detecting the level of the flood. Based from the existing way of reporting flooded roads in India have concluded that the Flood Detector System using Arduino can measure the height of the flood; and measurement data can be distributed to officer in charge and to the residents. The system also indicate passable and impassable road that will help commuters to avoid getting stuck in an impassable road. The system also provides camera to easily monitor the flood.