COLLEGE CODE:9512

COLLEGE NAME:JP COLLEGE OF ENGINEERING

DEPARTMENT:ECE

PROJECT CODE:Proj_211933_Team_1

FLOOD MONITROING AND EARLY WARNING SYSTEM

PHASE 5:DOCUMENTATION

TEAM MEMBERS:

1.JAYA MAHA VARSHINI.S(au951221106014)

2.RAMYA.G(au951221106031)

3.KALPANA DEVI(au951221106017)

4.RISWANA FATHIMA(au951221106034)

5.THANGA DURACHI (au951221106050)

OBJECTIVE:

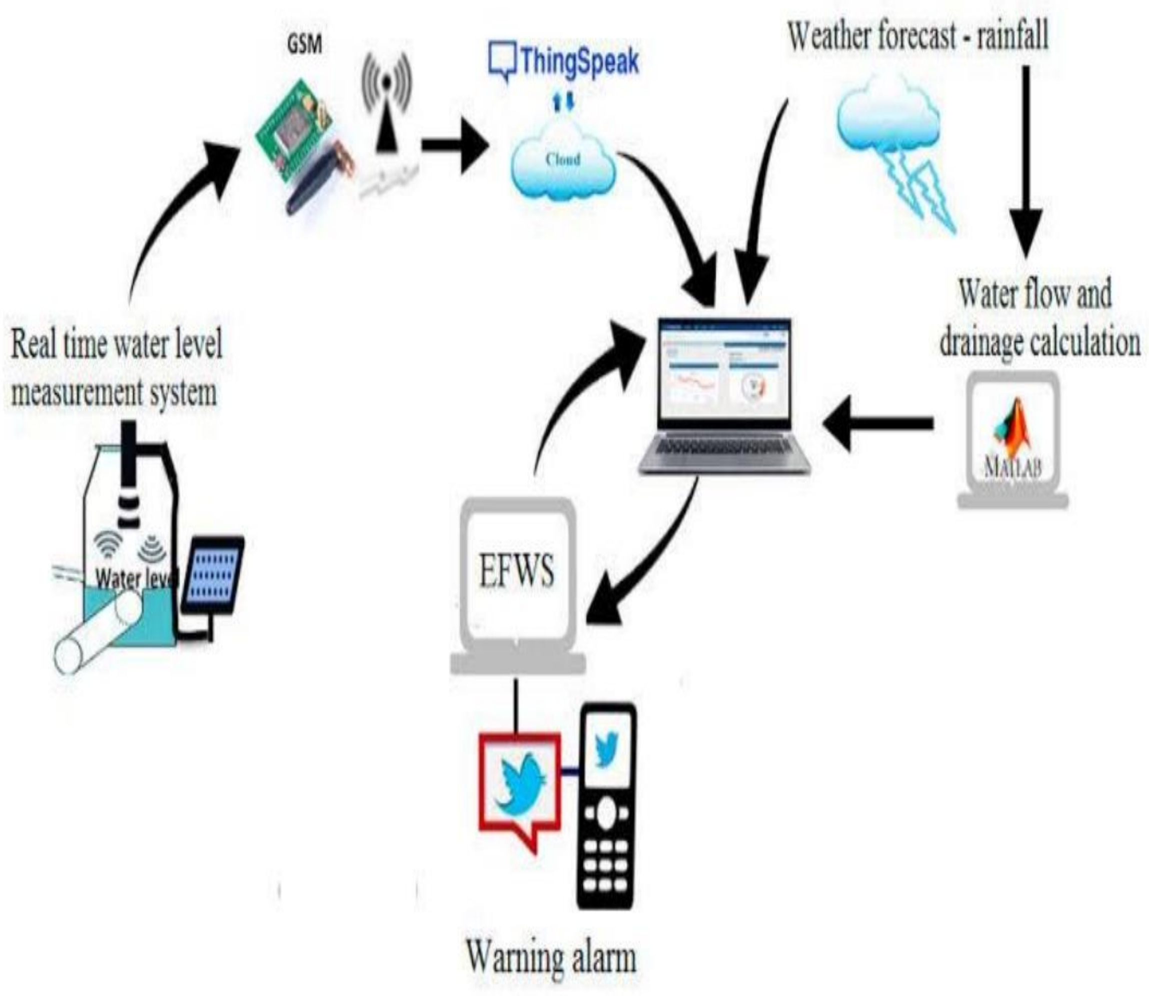
The objective of a flood monitoring and early warning system is to provide timely and accurate information about potential floods to the public and authorities, enabling them to take appropriate measures to minimize damage and loss of life. The system typically consists of four key components: **monitoring**, **forecasting**, **warning and response.**

WORKING PRINCIPLE :

It shows the diagramatic representation of the flood monitoring and early warning system

It clearly explains the functioning of the flood monitoring and early warning system.

WORKING DIAGRAM:


GSM
ThingSpeak
Cloud
Weather forecast - rainfall
Real time water level measurement system
Water level
Water flow and drainage calculation
MATLAB
EFWS
Warning alarm

SAMPLE CODE:

The sample python code for the implementation of flood monitoring and early warning system was given below.These are the sample codes by which the monitoring and warning programs will be done.

CODE:

```
import random
def acquire_data():
# Simulate data acquisition, replace with actual data retrieval code
return {
"rainfall": random.uniform(0, 50),
"river_level": random.uniform(0, 10)
}
def check_for_flood(data, thresholds):
if data["rainfall"] > thresholds["rainfall"] or data["river_level"] > thresholds["river_level"]:
return True
return False
def send_notification():
# Implement your notification logic here
print("Flood Warning: Take necessary precautions!")
```

```
if _name_ == "_main_":
thresholds = {
"rainfall": 30, # Example rainfall threshold in mm
"river_level": 7 # Example river level threshold in meters
}

while True:
data = acquire_data()
if check_for_flood(data, thresholds):
send_notification()
```

PHYTHON CODE FOR IDE IMPLEMENTATION:

```
# OpenCV packages for Python
import cv2
# Python plotting package
```

```
import matplotlib.pyplot as plt
# Fork of argparse to add features and simplify its
code
import argparse
# functions to make basic image processing functions
import imutils
# this for add math function
import math
import time
# package for array computing with Python
import pandas as pd
from numpy import asarray as pn
from sklearn.linear_model import LinearRegression
from imutils.perspective import four_point_transform
from imutils import paths
from sklearn.metrics import mean_squared_error
# capture frames from a camera
cap = cv2.VideoCapture(0)
cap.set(3, 640)
cap.set(4, 480)
count = 0
height = []
flag = 0
# reads frames from a camera
ret, frame = cap.read()
cv2.imwrite("testimage.jpg", frame)
im = cv2.imread("testimage.jpg")
r = cv2.selectROI(img=im, windowName="test")
```

```
t = time.localtime()
current_time = time.strftime("%H:%M:%S", t)
9
# loop runs if capturing has been initialized
while (1):
ret, frame = cap.read()
if frame is None:
break
# Crop image
frame = frame[int(r[1]):int(r[1] + r[3]),
int(r[0]):int(r[0] + r[2])]
# Convert the img to grayscale
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
# Apply edge detection method on the image
edges = cv2.Canny(gray, 100, 120)
# Run Hough on edge detected image
lines = cv2.HoughLinesP(edges, 1, math.pi/180, 20,
None, 20, 480)
dot1 = (lines[0][0][0], lines[0][0][1])
dot2 = (lines[0][0][2], lines[0][0][3])
```

```
slope = ((lines[0][0][3] -
lines[0][0][1])/(lines[0][0][2] - lines[0][0][0]))
#cv2.line draws a line in img from dot1 to dot2
# (255,0,0) denotes the colour of the line to be
drawn
if 0 <= slope <= 0.15:
cv2.line(frame, dot1, dot2, (255, 0, 0), 3)
length = 150 - lines[0][0][3]
print(length)
height.append(length)
cv2.imshow("Detected Line", frame)
# finds edges in the input video and
# marks them in the output map edges
edged_frame = cv2.Canny(frame, 1, 100)
```

```
cv2.imshow('Edged Frame', edged_frame)

if cv2.waitKey(1) & 0xFF == ord('q'):
break
x = []
y = []
```

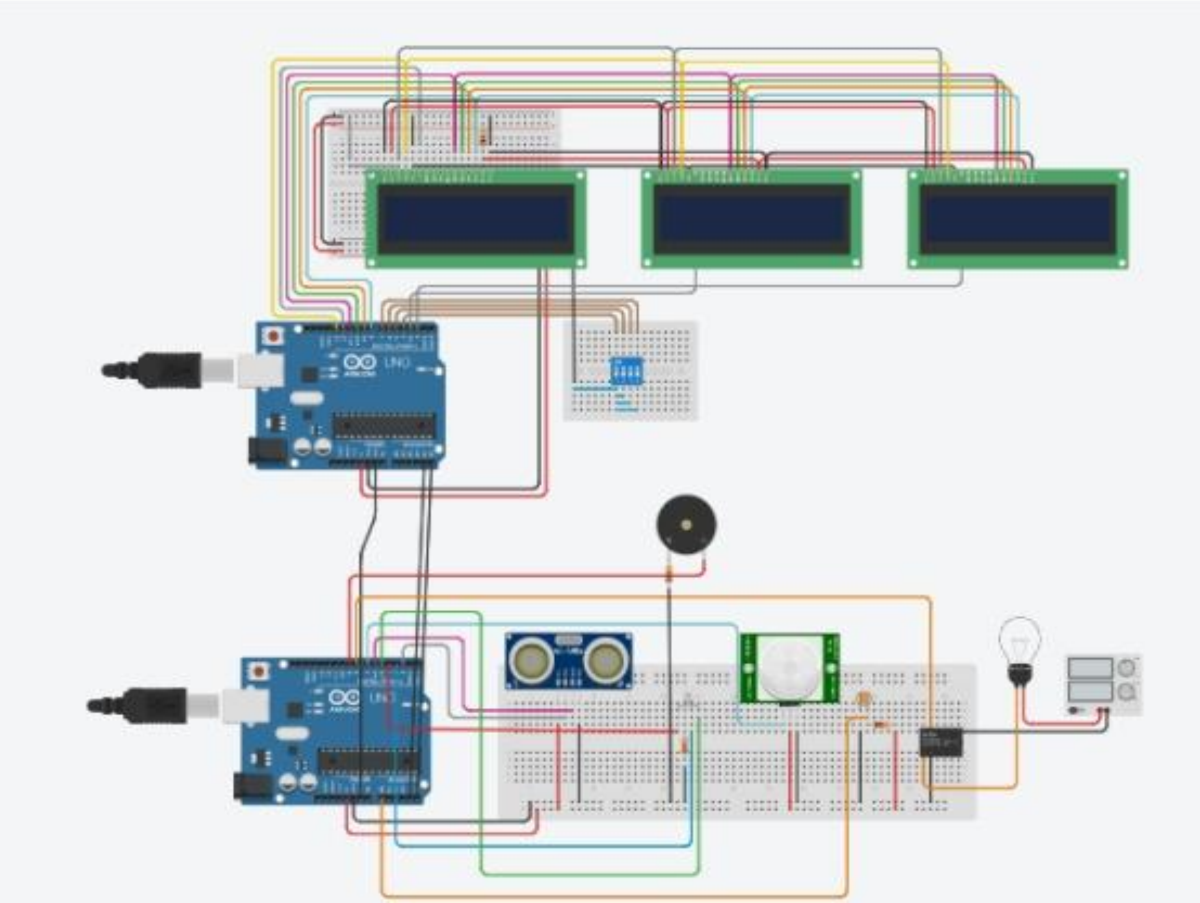
```
file = open("Saved.txt","a")
for i in range(len(height)):
x.append(i)
y.append(height[i])
file.write(str(x[i-1])+","+str(y[i-1])+"\n")
X=np(x)
Y=np(y)
X = X.reshape(len(X),1)
Y = Y.reshape(len(Y),1)
model = LinearRegression()
model.fit(X,Y)
model = LinearRegression().fit(X,Y)
r_sq = model.score(X,Y)
y_pred = model.predict(X)
y_pred = model.intercept_+ model.coef_*X
print('Predicted Response:', y_pred, sep='\n')
print('Start :', current_time)
print('Coefficient of Determination:', r_sq)
print('Intercept:', model.intercept_)
accuracy = mean_squared_error(y, y_pred)
```

```
print('Accuracy :', accuracy)
11
t = time.localtime()
current_time2 = time.strftime("%H:%M:%S", t)
print('Stop :', current_time2)
plt.plot(X,Y,'.',color='black')
cap.release()
cv2.destroyAllWindows()
plt.plot(X,y_pred)
plt.title('Test Data')
plt.xlabel('Time')
plt.ylabel('Height')
plt.show()
```

DEVELOPED MODEL:

The model for the purpose of monitoring the flood and warning was developed by using the IOT and and it was dumped into the devices like arduino for the the working of the devices.The model was shown below.

The model consists of electronic device which are functioned by internet of things.



WEB DEVELOPMENT CODE:

The model was provided with the real time output by using the web development code .The development code which was implemented is shown below:

```
//slave
 #include
 #define PIR 7
#define LDR A0
 #define bulb 8
#define buzzer 9
#define SLAVE_ADDR 9
 #define ANSWERSIZE 5
 #define led1 4 // led1 red
 #define led2 3 // led2 blue
 #define led3 5 // led3 green int trigPin = 2;
 int echoPin = 6;
int val = 0;
 int duration;
 float distance;
 float meter;
union { float x; byte myData[4]; }data;
 void setup() { Wire.begin(SLAVE_ADDR);
// join i2c bus with address
 #8 Wire.onRequest(sendEvent);
 // register event Serial.begin (9600);
 pinMode(trigPin, OUTPUT);
 digitalWrite(trigPin, LOW);
 delayMicroseconds(2);
 pinMode(echoPin, INPUT);
 pinMode(PIR,INPUT);
 pinMode(LDR, INPUT);
 pinMode(bulb, OUTPUT);
 pinMode(led1, OUTPUT);
 pinMode(led2, OUTPUT);
 pinMode(led3, OUTPUT);
 pinMode(buzzer, OUTPUT);
 delay(6000);
 Serial.println("Distance:");
 Serial.println("Water detect:"); }
void loop() { delay(50); int i = 0;
 // This can be set to any value to change the frequency.
digitalWrite(trigPin, HIGH);
 delayMicroseconds(10);
digitalWrite(trigPin, LOW);
```

```
 duration = pulseIn(echoPin, HIGH);
 distance = (duration/2) / 29.1;
 data.x = distance;
Serial.print(distance);
 Serial.print("cm");
 Serial.print("\n");
 // This is required to calculate the distance in centimeters.
val = digitalRead(PIR);
if (val == HIGH && distance <= 30) { RGB_color(255, 0, 0);
// Red digitalWrite(buzzer, HIGH); }
else { digitalWrite(buzzer, LOW); }
 if (distance <= 30) { RGB_color(255, 0, 0); // Red }
else { } if (distance <= 100 && distance > 30)
 { RGB_color(255, 255, 0); // Yellow }
 else { } if (distance > 100) { RGB_color(0, 255, 0);
 // Green }
 else { }
 if (analogRead(A0) > 500 && val == HIGH && distance <= 30)
 { digitalWrite(8, LOW); } else { digitalWrite(8, HIGH); } }
```

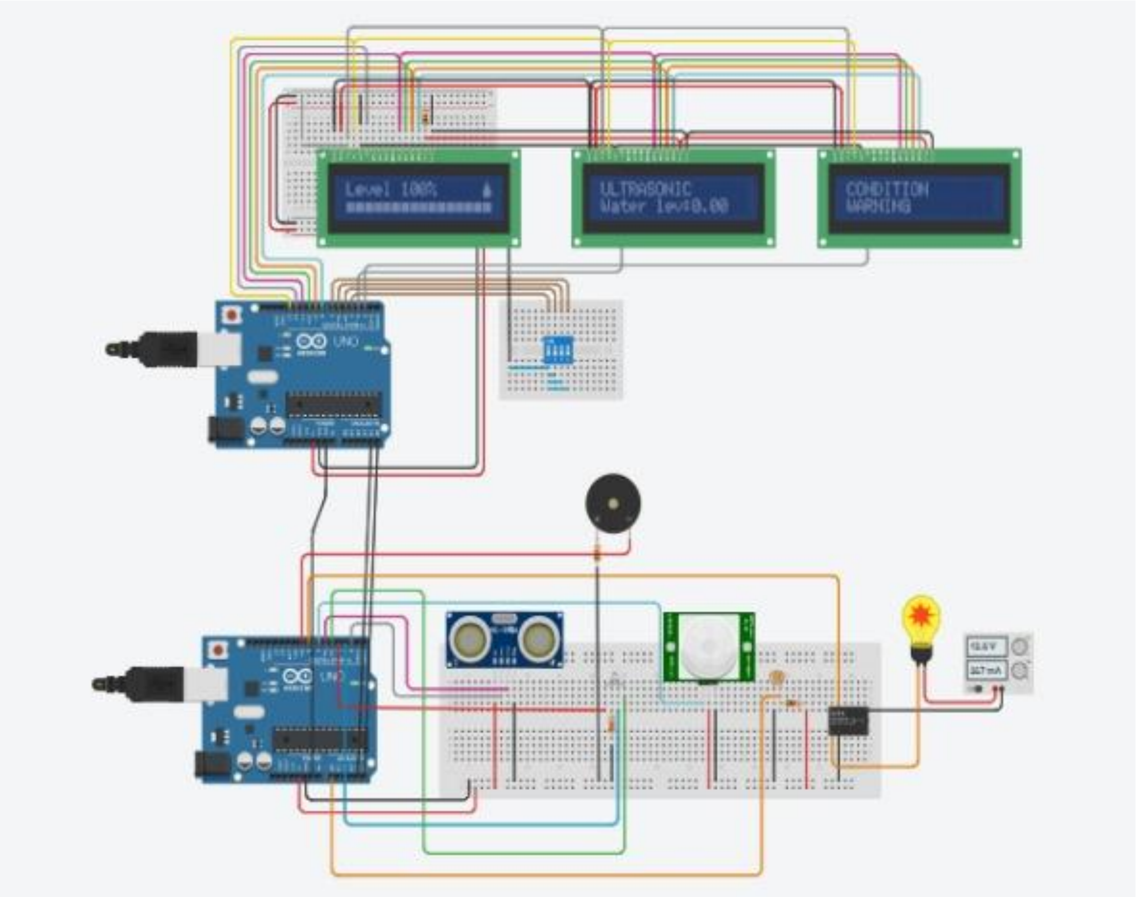
```
void RGB_color(int red_light_value, int green_light_value, int blue_light_value)
{ analogWrite(led1, red_light_value);
analogWrite(led3, green_light_value);
analogWrite(led2, blue_light_value); }
// function that executes whenever data is requested by master // this function is registered as an event,
see setup() void sendEvent()
{ Wire.write(data.myData, sizeof(data.myData));
//float distance goes to Master as 32-bit binary32 data /* while (1 < Wire.available())
// Loop through all but the last { float distance = Wire.read();
// Receive byte as a character Serial.print(distance);
// Print the distance values } int x = Wire.read();
// Receive byte as an integer Serial.println(x);
delay(10); */ }
```

ENHANCEMENT OF PUBLIC SAFETY:

The public safety is protected against the flood by monitoring the water level by the developed model.

It gives warning to the people by the signal that is delivered by the model developed

The below model was tested and it gives the output in such a way was modeled and shown below in the after simulation process.



EMERGENCY SITUATION:

On emergency citivation the warning was delivered by the developed model.in order to indicate the flood.

CONCLUSION:

The flood monitoring and early warning system was made completely and is ready to serve the public.