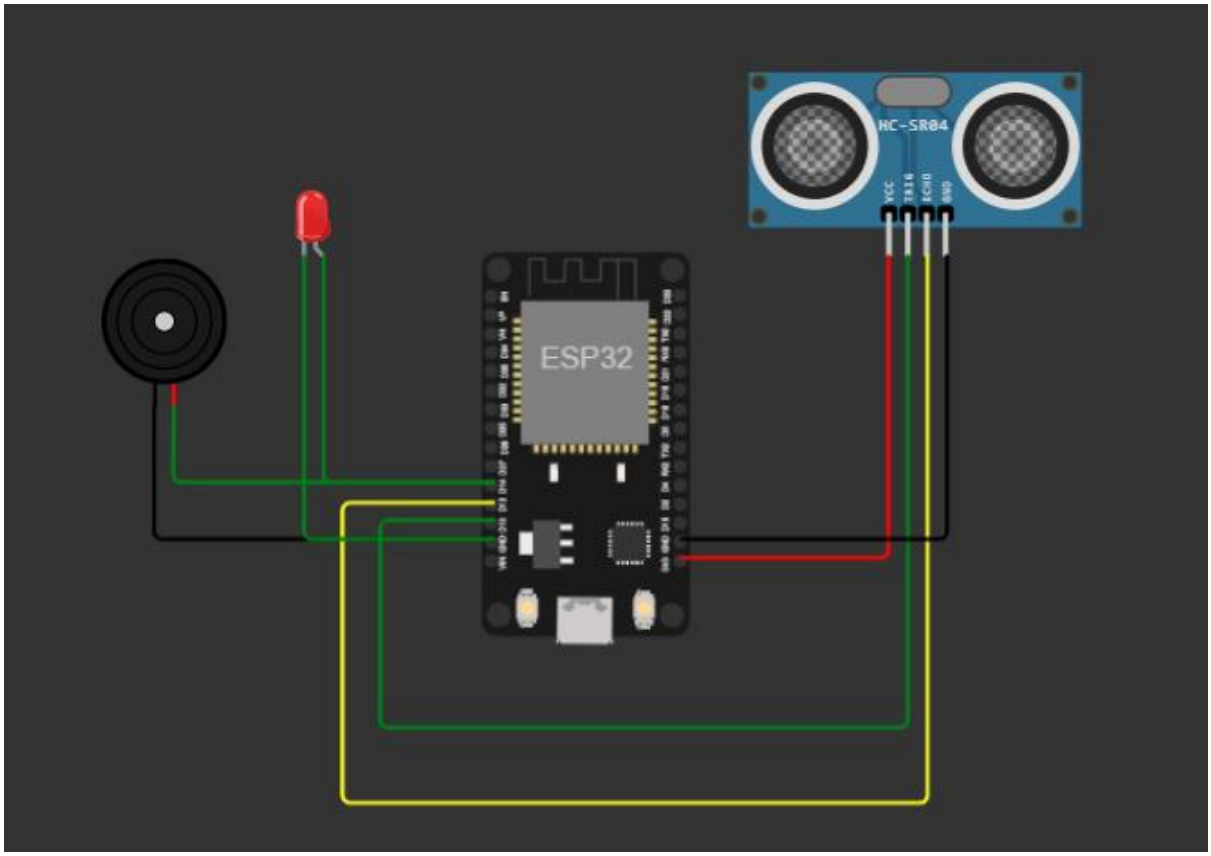
Phase 4 Project: Flood Monitoring and EWS

INTRODUCTION:

In this phase of the project, we are going to access the data on a web platform. The platform that we are going to use today is called Blynk. Blynk is an IOT platform that can be accessed from smartphones and computers to control Arduino, Raspberry Pi, and NodeMCU modules through the internet. This application is used to create a graphical interface by compiling and providing addresses on the appropriate widgets(These widgets can be created and customized by us. Example: Gauges, Led display,4 segment Led Display, Text ..etc)

The sensor and module we used for this project are the ultrasonic sensor and ESP32 module respectively. We use this ultrasonic sensor to calculate the level of water in a dam and if it's above the desired level (changes for different sizes of dams) and alert the residents using a Speaker and warning lights. (in the module we used a buzzer to simulate a real-life speaker and led to simulate warning lights.)



We are going to implement this in the Blynk application. The code to run the module and send the data to the web platform(Blynk) is given below.

CODE:

```
#define BLYNK_TEMPLATE_ID "TMPL3w-dCObVZ"
#define BLYNK_TEMPLATE_NAME "IOTPhase4"
#define BLYNK_AUTH_TOKEN "i4xhUCb3epG8weF80tEfBppGaOts0SMv"
#include  <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#define trigger_wave 13
#define distance_echo 12
#define buzzer 14
unsigned int condition=0;
BlynkTimer timer;
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "Wokwi-GUEST";
char pass[] = "";
#define BLYNK_PRINT Serial
void setup()
 {
  Serial.begin(115200);
  pinMode(trigger_wave, OUTPUT);
  pinMode(distance_echo, INPUT);
  pinMode(buzzer,OUTPUT);
```

```
  Blynk.begin(auth,ssid,pass,"blynk.cloud",8080);
}
void loop()
{
  //send the wave to measure the length and the capacity of the water
  digitalWrite(trigger_wave, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigger_wave, LOW);
  // receiving wave to judge the water level and hit the buzzer
  int duration = pulseIn(distance_echo, HIGH);
  Serial.print("Distance in feet: ");
  Serial.println((duration / 58)*0.3);
//now I have a value of 400 and the average dam will be  120 feet
//I need to convert my values to 120 so
condition = (duration / 58)*0.3;
Blynk.virtualWrite(V0,condition);
if(condition >= 90)
{
  digitalWrite(buzzer,HIGH);
  Serial.print("DAM WILL BE OPENED FLOW WARNING!!!!!!!!! ");
  Blynk.virtualWrite(V0,condition);
}
else if (condition<90)
```

```
{
  digitalWrite(buzzer,LOW);
}
  delay(1000);
  Blynk.run();
}
```

CODE EXPLANATION:

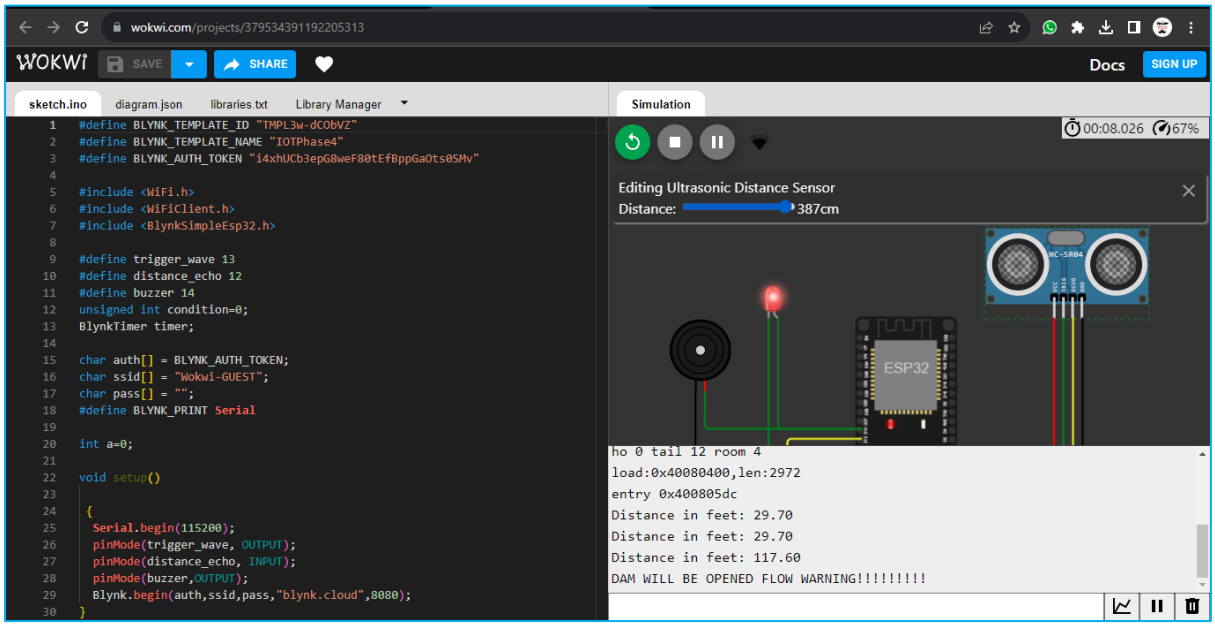
- THE BLYNK TEMPLATE ID AND ITS NAME IS USED AS PROVIDED BY THE SOURCE .
- TO WORK MY PROGRAM THE NECESSARY LIBRARIES ARE ADDED IN THE WOKWI SOFTWARE.
    - EG:WIFICLIENT , WIFI,BLYNK SOFTWARES ETC.
- WE ARE DEFINING THE TEMPLATE ID , TEMPLATE NAME , BLYNK AUTHENTICATION TOKEN ETC AS REQUIRED ANS FOR AUTHENTICATION TOKEN .
- WE USE #INCLUDE KEYWORD TO INCLUDE THE LIBRARIY FILES SO THAT OUR CODE CAN ACTIVAE AND RUN TO PRODUCE OUR REQUIRED FLOOD MONITORING OUTPUT.
- WE DEFINE THE PIN NNUMBER FROM ESP32 BOARD WITH THE VARIABLE NAME TRIGGER_WAVE AS PIN 13 AND DISTANCE_ECHO AS PIN 12.
- WE ALSO DEFINE A VARIABLE BUZZER FOR PIN 14
- WE INITIALIZE INTEGER CODITION AS 0 .

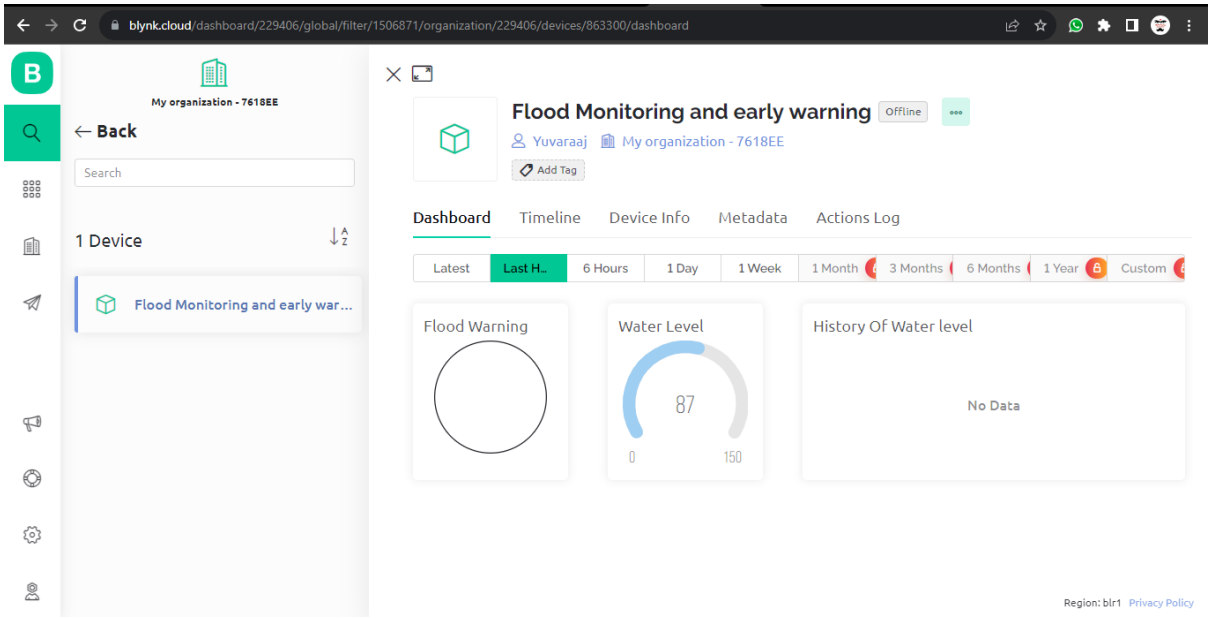
- **NOW WITH THE CHARATER TYPE WE SET THE AUTH,SSID, PASSWORD FORM OUR PROGRAM TO CONNNCT WITH .**

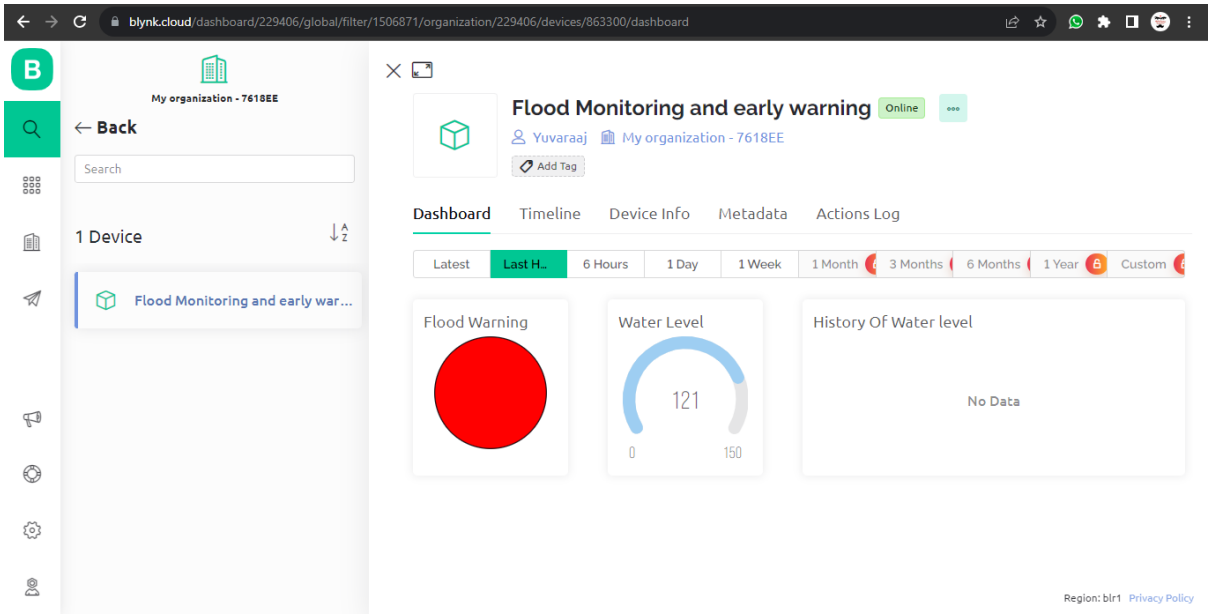
- **NOW WITHIN THE VOID SETUP SEGMENT.WE PROVIDE THE BAUD RATE FOR OUR PROGRAM WITH THE ESP32 VIRTUAL BOARD.**

- **NOW WE SET THE PIN MODE FOR OUR DECLARED VARIABLES EITHER THE PIN MODE WILL BE INPUT OR OUTPUT. FOR OUR PROGRAM , TRIGGER_WAVE IS SET AS OUTPUT , DISTANCE_ECHO IS SET AS INPUT,BUZZER IS SET TO OUTPUT SINCE WE NEED TO HEAR THE SOUND AS ITS OUTPUT.**
- **LETS START OUR LOOPING PART WHICH RUNS THE PROGRAM CONTINUOUSLY.**
- **DIGITALWRITE IN THE CODE SAYS THAT OUR TRIGGER_WAVE IS SET TO HIGH TO SWITCH THE PART ON IN ULTRASONIC SENSOR.**
- **DIGITALWRITE IN THE CODE FOR TRIGGER_WAVE IS SET TO LOWTO SWITH THE PART OFF IN ULTRASONIC SENSOR.**
- **NOW THE DISTANCE _ECHO PART IS SET TO HIGH SO THAT IT CAN RECEIVE THE TRIGGER_WAVE SIGNAL WITH THE MEASUREMENT.**
- **NOW WE SET OUR CALCULATION (DURATION/58)*0.3 IN THE VARIABLE CONDITION TO DISPLAY THE OUTPUT AS A RESULT IN PROGRAM AS WELL AS BLYNK PLATFORM.**
- **NOW TO CHECK THE CONDIION THAT IF THE LEVEL IS GREATER THAN 90 THEN WARNING MUST BE GIVEN BY MAKING THE BUZZER ON AND INDICATING THROUGH TEXT.**
- **OR ELSE SIMPLY THE LEVEL OF WATER MUST BE DISPLAYED WITHOUT ANY SIDE OPERATION.**
- **ID THE LEVEL OF WATER IS LESS THEN 90 THEN THE BUZZER WILL BE LOW.**
- **BLYNK.VIRTUALWRITE KEYWORD IS USED TO ASSIGN THE PIN IN BLYNK SO THAT IT MAY CALL THE BLYNK PIN TO DISPLAY THE OUTPUT.**

OUTPUT IN BLYNK:







Wokwi Code:

https://wokwi.com/projects/379534391192205313

CONCLUSION:

In conclusion, the project on flood monitoring and early warning systems has been a significant endeavor focused on improving the efficiency, effectiveness, and safeness of the public during floods and heavy rains .Throughout this project, we have analyzed various aspects of the existing data and proposed innovative solutions to improve the prevention of life during floods.