caution: might be some mistakes in documentation

LATE BHAUSAHEB HIRAY S.S. TRUST'S INSTITUTE OF COMPUTER APPLICATION

ISO 9001-2008 CERTIFIED

S.N. 341, Next to New English School, Govt. Colony, Bandra (East),
Mumbai – 400051, Tel: 91-22-26570892/3181

Date: 02-01-2023

A PROJECT REPORT ON

"Home Automation System"

BY

Mayuresh Redekar (202164)
Salil Bhagat (202199)
Hritik Gopale (202145)

(MCA SEM-III)
ACADEMIC YEAR 2022-2023

Under the guidance of

Prof. Minesh Ade
Prof. Rashmita Pradhan

---------------------------	------------------------------	---------------------------
Director	**Examiner**	**Internal**

TABLE OF CONTENTS

Abstract

The dearth of water resources is going to be one of the crucial predicaments encountering the majority of the cities in this world. The amount of wastage of drinkable water during channeling has been spotted as the prime cause, to establish some computing strategies in prompting the wastage to provide render financial build-ups and saving other resources as well. Thus, it ensures that this system contributes to deliver some water for our future too. This paper introduces an IoT enabled gadget which helps in administering and monitoring the consumption of water in multiple buildings simultaneously. This module can be effortlessly fixed onto desired water containers and carry on in the long run. The Ultrasonic sensor is placed on the top of the tank which continuously keeps track of the water level in real-time, which will inform the users about the level of liquid and automatically turn on/off the water pump as per the defined functions. This figure will be regularly revised on the website so that the user can analyze the amount of water usage and thus, control the wastage. According to the level of water in the tank, the motor functioning is automatically controlled. When the surface level drops below the threshold measure, the motor will be again switched on impulsively. This system expects to watch approximately 5-20 water storage based on just the sole site.

Acknowledgement

In completing this graduate project .We have been fortunate to have help, supportand encouragement from many people. We would like to acknowledge them for their cooperation.

First, we would like to thank **Prof.Rashmita Pradhan**, our project advisor, for guiding us through each and every step of the process with knowledge and support. Thank you for your advice, guidance and assistance.

Declaration

I hereby declare that the project entitled, Home Automation System (IOT Based) done at Hiray College, has not been in any case duplicated to submit to any other university. To the best of my knowledge other than me, no one has submitted to any other university. The project is done in partial fulfilment of the requirements for the award of the degree of MASTERS OF COMPUTER APPLICATIONS to be submitted as a second semester project as part of our curriculum.

Name of the students,

Mayuresh Redekar

Salil Bhagat

Hritik Gopale

Introduction

The facility requirements in many industries, farms, hostels, hotels, offices include an overhead tank for water, which is usually fed through an electric pump that is switched off when the tank is filledup and switched on when it is empty. so, the most common way of knowing when the tank is filled is by observing when it overflows the brim. depending on the type of liquid being handled, overfilling of such a tank could lead to a great liquid material losses ranging in the order of thousands of per week depending on the extent of such application. these losses can be prevented if the tank is monitored automatically by incorporating a feedback.

Water level indicator using ultrasonic sensor & esp8266 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 overhead watertank. this project can be further enhanced to control the water level in the tank by turning it on, when the water level is low, and turning it off when the water level is high. thus, the water level indicator helps in preventing wastage of water in overhead tank.

A transmitter circuit and a receiver circuit. the transmitter circuit makes use of an ultrasonic sensor to measure the water level in terms of distance. this data is sent to the receiver circuit using rf (Radio Frequency) communication.

Along with water monitoring system it also consist of home electronics system i.e. the electronics systems like lights, fans ,motors,etc can be controlled with an blink application .

Components used in this project include:

- BLYNK software
- ESP module
- TTGO TCall Module
- Ultrasonic sensor
- Fan Module
- LED’S
- Relay’s Module
- Berg Strip

Synopsis

One of the major problems faced by most of the countries is the issue of water scarcity in the world and wastage during transmission has been identified as a major culprit; this is one of the motivations for this research, to deploy computing techniques in creating a barrier to wastage in order to not only provide more financial gains and help the environment as well as the water cycle which in turn ensures that we save water for our future.

IOT based Water Level Monitoring system is an innovative system which will inform the users about the level of liquid and will prevent it from overflowing. To demonstrate this the system makes use of containers, where the ultrasonic sensors placed over the containers to detect the liquid level and compare it with the container’s depth. The liquid level is highlighted as colored to show the level of liquid present in the container with the help of a application to the user. The buzzer starts ringing when the set limit of the liquid is crossed. Thus this system helps to prevent the wastage of water by informing about the liquid levels of the containers.

Other problem also include innovation in home administration/management systems. This project also solves this issue. This project also focuses on home systems management.

i.e. the electronics systems like lights, fans ,motors,etc can be controlled with an blink application .

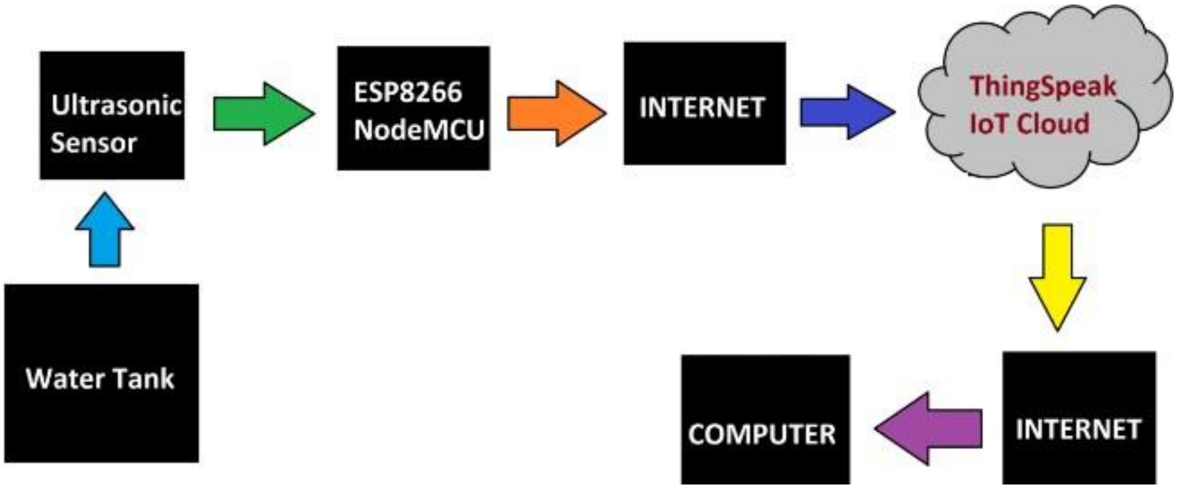
Benefits:

- Users can keep track of the water system and avoid over flowing of water by getting an automated alert and giving command to stop the water pump.
- Remote access: Control your home from mobile devices, including your laptop, tablet, or smartphone.
- Comfort: Use home automation to make your home a more comfortable, livable space. Pre-program you adjust amount of lights.

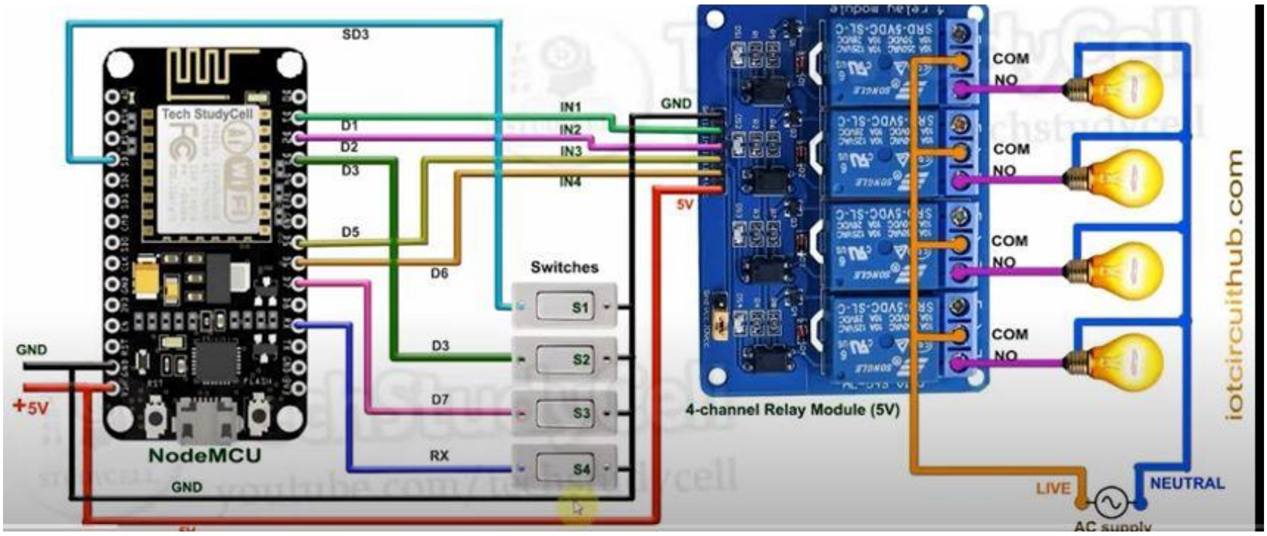
Scope:

- Maximum automated systems are both high-priced or calls for manpower,this system solves the both of these problems.
- Considering stressed out era is utilized in our proposed machine there is scope to further modify the proposed system.
- We can also add various different types of modules like power management, AI integration, Smart Fridge, AC, etc.

Flow Charts and Diagrams

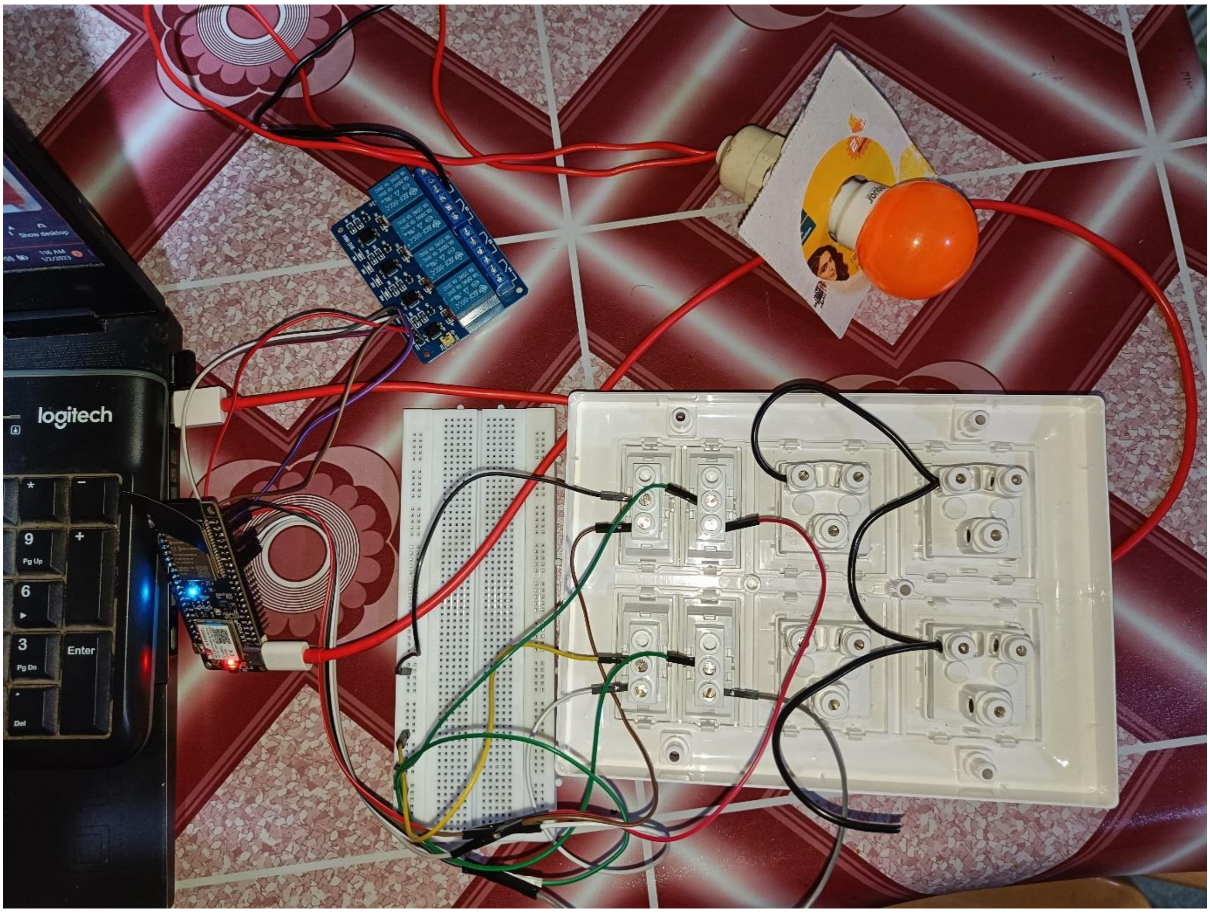


Flow chart for water level

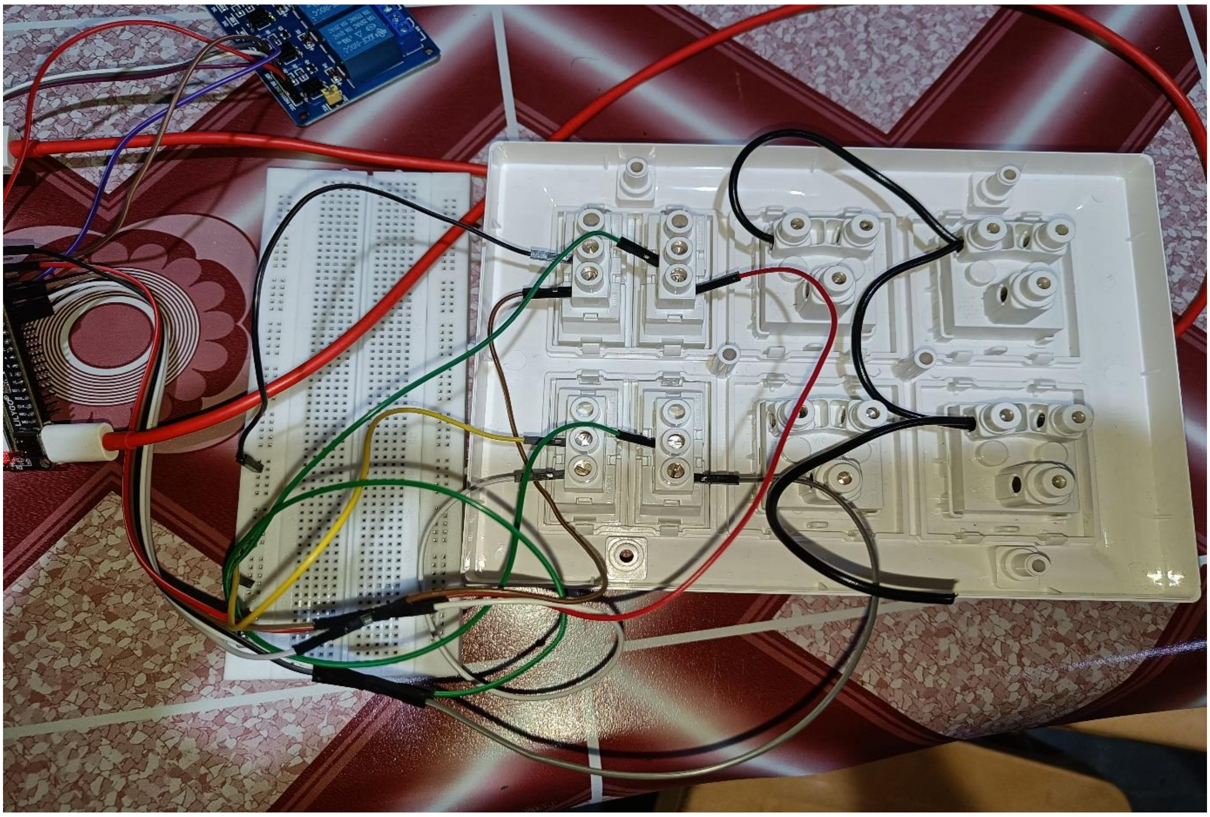


Circuit diagram

All connections

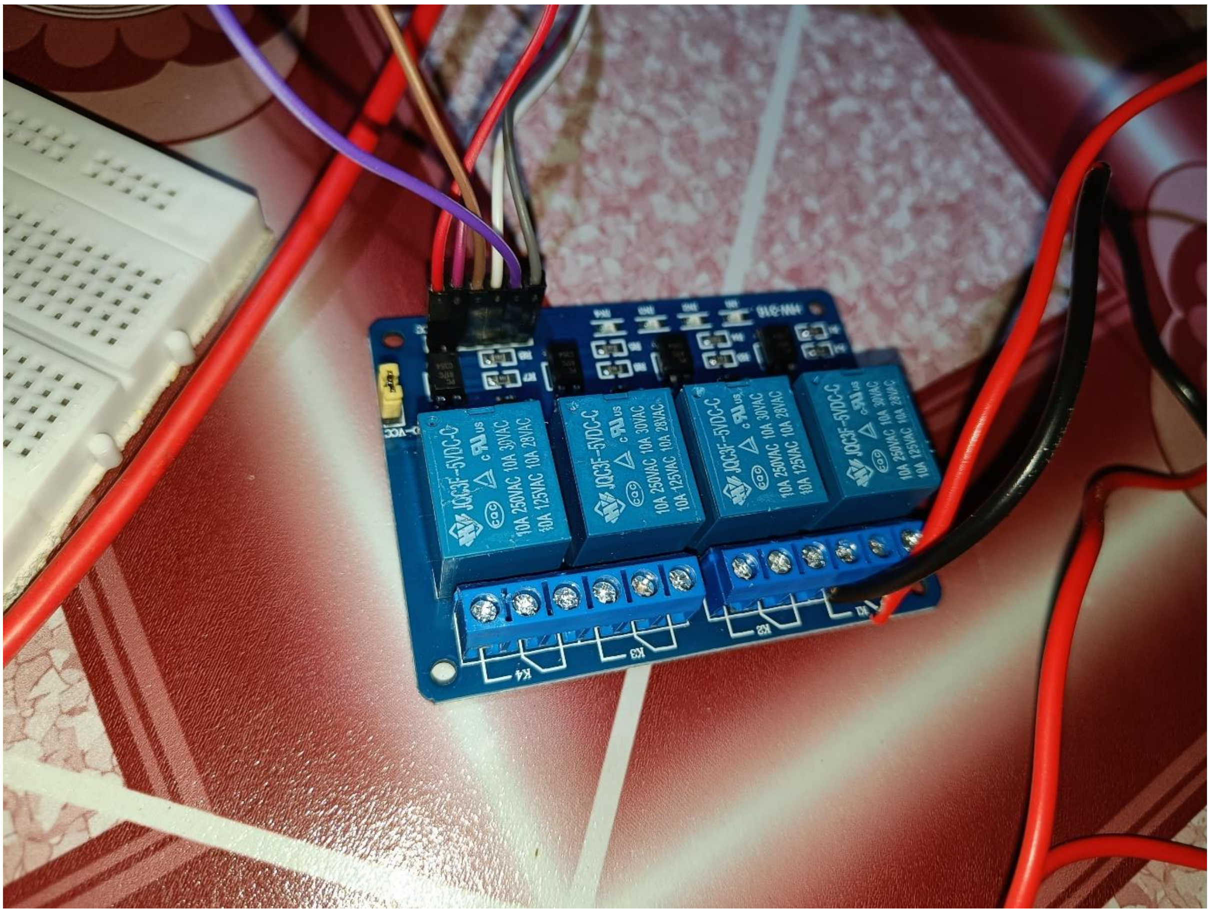


Board connection

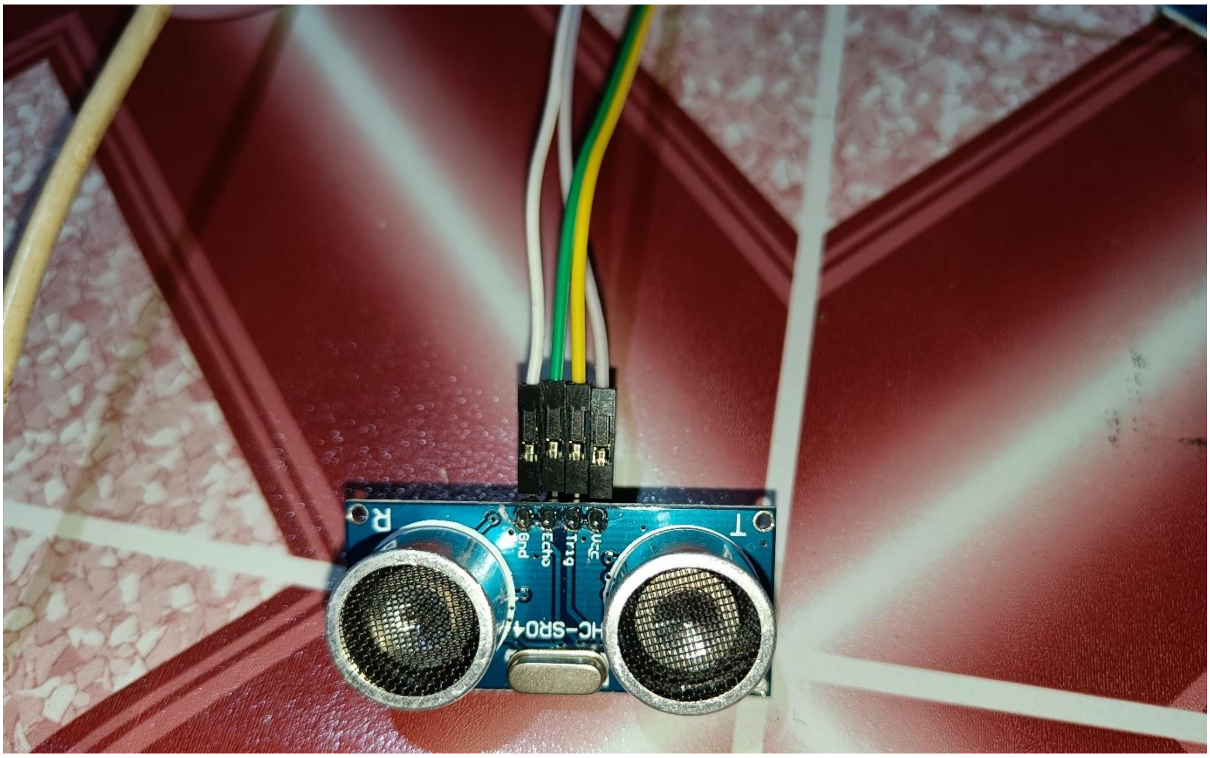


TTGO TCall Module

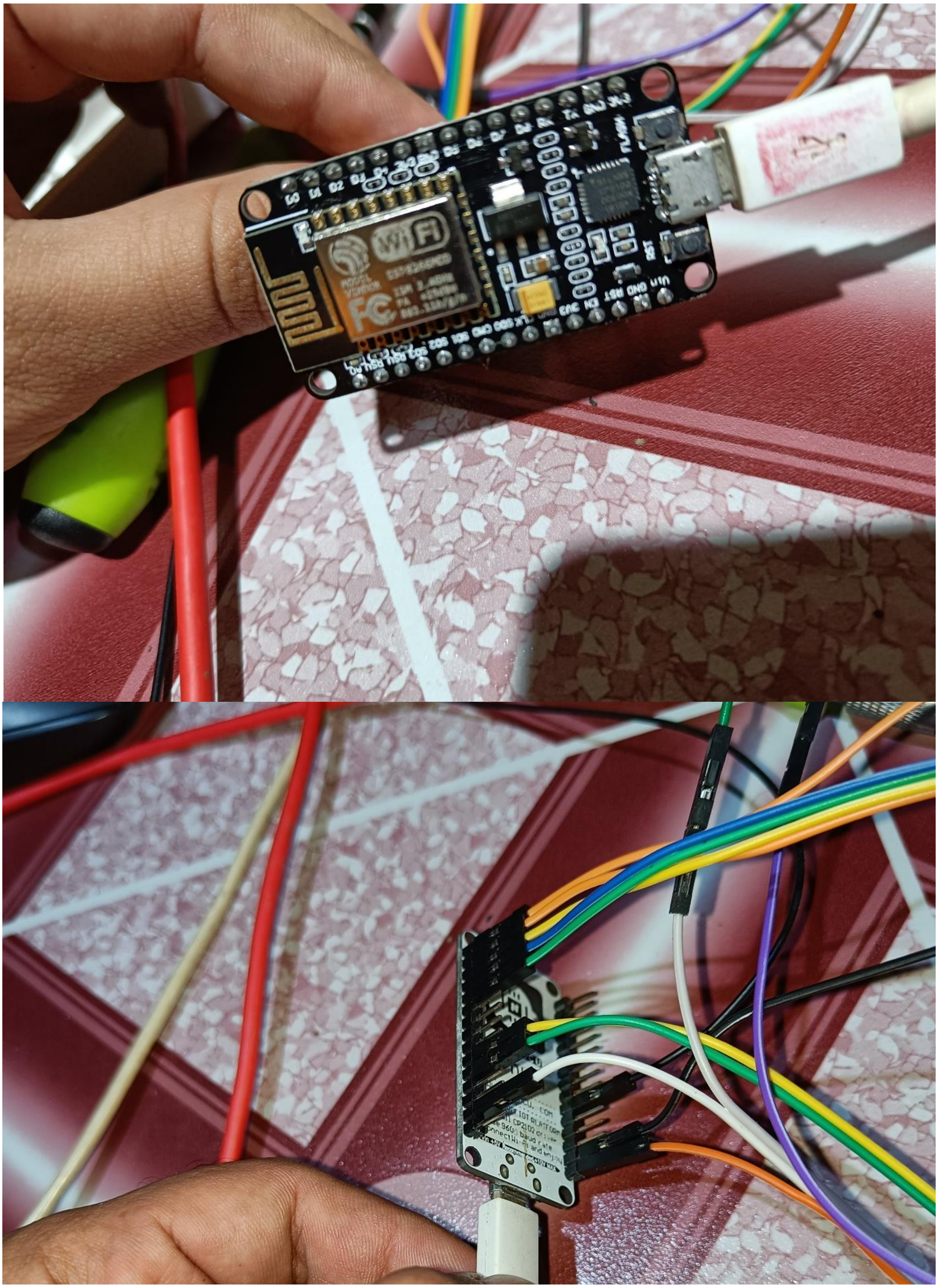
Relay connection



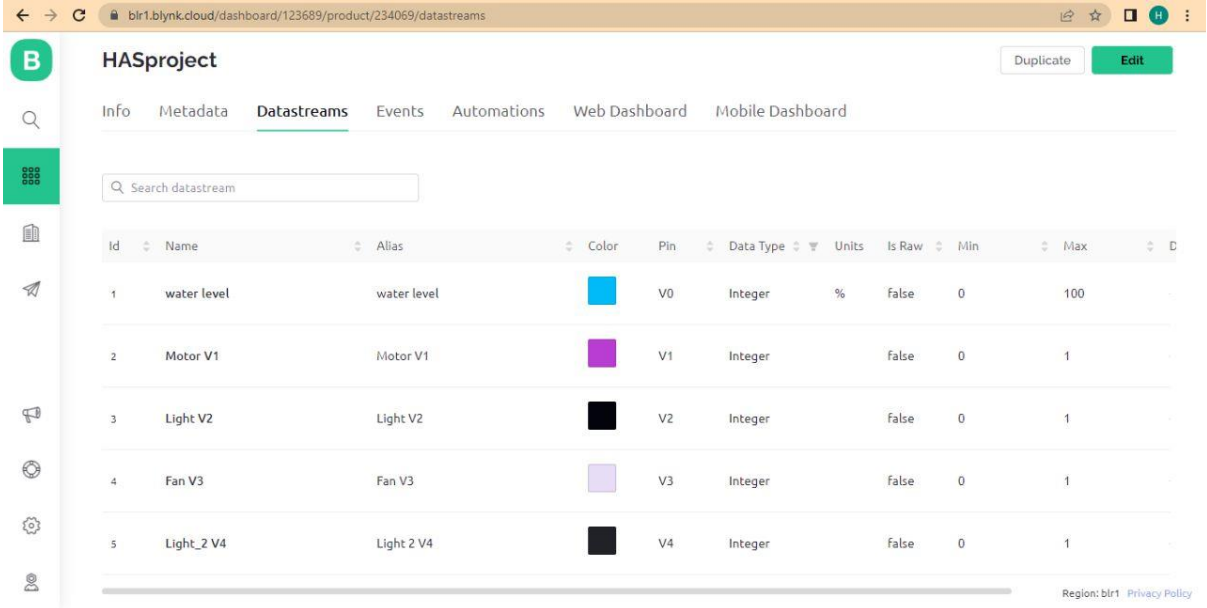
Ultrasonic sensors connection



ESP8266 MODULE CONNECTIONS



BLYNK DASHBOARD AND UI



blr1.blynk.cloud/dashboard/123689/product/234069/datastreams

HASproject

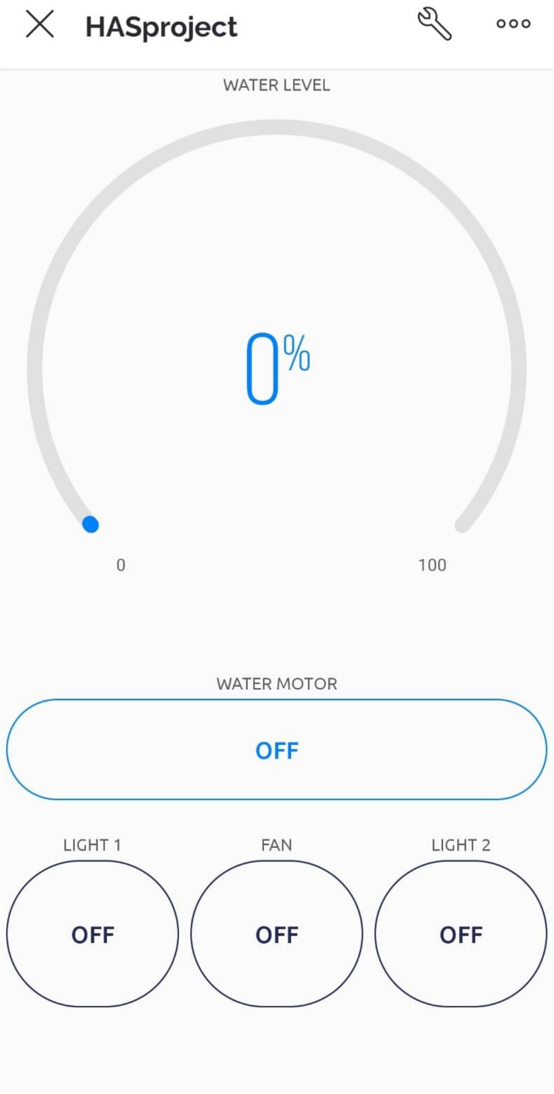
Duplicate Edit

Info Metadata Datastreams Events Automations Web Dashboard Mobile Dashboard

Search datastream

Id	Name	Alias	Color	Pin	Data Type	Units	Is Raw	Min	Max	D
1	water level	water level	🟦	V0	Integer	%	false	0	100	-
2	Motor V1	Motor V1	🟪	V1	Integer		false	0	1	-
3	Light V2	Light V2	⬛	V2	Integer		false	0	1	-
4	Fan V3	Fan V3	⬜	V3	Integer		false	0	1	-
5	Light_2 V4	Light 2 V4	⬛	V4	Integer		false	0	1	-

Region: blr1 Privacy Policy



Sample Screenshot & Coding

```
#define BLYNK_PRINT Serial    // Comment this out to disable prints and save space
#define BLYNK_TEMPLATE_ID "********"
#define BLYNK_DEVICE_NAME "HASproject"
#define BLYNK_AUTH_TOKEN "****************************"

// Default heartbeat interval for GSM is 60
// If you want override this value, uncomment and set this option:
// #define BLYNK_HEARTBEAT 30

// Please select the corresponding model

#define SIM800L_IP5306_VERSION_20190610
// #define SIM800L_AXP192_VERSION_20200327
// #define SIM800C_AXP192_VERSION_20200609
// #define SIM800L_IP5306_VERSION_20200811

#include "utilities.h"

#define S1 21
#define relay1 25

#define S2 19
#define relay2 14

#define S3 18
#define relay3 33
```

```
#define S4 22
#define relay4 13

// Select your modem:
#define TINY_GSM_MODEM_SIM800

// Set serial for debug console (to the Serial Monitor, default speed 115200)
#define SerialMon Serial

// Set serial for AT commands (to the module)
// Use Hardware Serial on Mega, Leonardo, Micro
#define SerialAT Serial1

// See all AT commands, if wanted
//#define DUMP_AT_COMMANDS

// Define the serial console for debug prints, if needed
#define TINY_GSM_DEBUG SerialMon

// set GSM PIN, if any
#define GSM_PIN ""

// Your GPRS credentials, if any
const char apn[] = "www";  // search for apn of your network provider on Google
const char gprsUser[] = "";
const char gprsPass[] = "";

#include <TinyGsmClient.h>
#include <BlynkSimpleTinyGSM.h>

#ifdef DUMP_AT_COMMANDS
```

```
#include <StreamDebugger.h>
StreamDebugger debugger(SerialAT, SerialMon);
TinyGsm modem(debugger);
#else
TinyGsm modem(SerialAT);
#endif
TinyGsmClient client(modem);

int switch_ON_Flag1_previous_I = 0;
int switch_ON_Flag1_previous_II = 0;
int switch_ON_Flag1_previous_III = 0;
int switch_ON_Flag1_previous_IV = 0;

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
const char auth[] = "jafynrefI2HUwm3fq46d1NUnZuqxQz7l";

BLYNK_WRITE(V1)
{
  int pinValue = param.asInt(); // assigning incoming value from pin V1 to a variable
  digitalWrite(relay1, !pinValue);
  delay(1000);
  // process received value
}

BLYNK_WRITE(V2)
{
  int pinValue = param.asInt(); // assigning incoming value from pin V1 to a variable
  digitalWrite(relay2, !pinValue);
  // process received value
}

BLYNK_WRITE(V3)
{
  int pinValue = param.asInt(); // assigning incoming value from pin V1 to a variable
  digitalWrite(relay3, !pinValue);
  // process received value
}

BLYNK_WRITE(V4)
{
  int pinValue = param.asInt(); // assigning incoming value from pin V1 to a variable
  digitalWrite(relay4, !pinValue);
  // process received value
}
void setup()
{
  // Set console baud rate
  SerialMon.begin(115200);

  pinMode(S1, INPUT_PULLUP);
  pinMode(relay1, OUTPUT);

  pinMode(S2, INPUT_PULLUP);
  pinMode(relay2, OUTPUT);

  pinMode(S3, INPUT_PULLUP);
  pinMode(relay3, OUTPUT);
  pinMode(S4, INPUT_PULLUP);
  pinMode(relay4, OUTPUT);

  digitalWrite(relay1, HIGH);
  digitalWrite(relay2, HIGH);
  digitalWrite(relay3, HIGH);
  digitalWrite(relay4, HIGH);

  delay(10);

  setupModem();

  SerialMon.println("Wait...");

  // Set GSM module baud rate and UART pins
  SerialAT.begin(115200, SERIAL_8N1, MODEM_RX, MODEM_TX);

  delay(6000);

  // Restart takes quite some time
  // To skip it, call init() instead of restart()
  SerialMon.println("Initializing modem...");
  modem.restart();
  // modem.init();

  String modemInfo = modem.getModemInfo();
  SerialMon.print("Modem Info: ");
  SerialMon.println(modemInfo);

  // Unlock your SIM card with a PIN
```

```
  //modem.simUnlock("1234");

  Blynk.begin(auth, modem, apn, gprsUser, gprsPass);

}

void loop()
{
  Blynk.run();
  Switch();
}

void Switch()
{
  if (digitalRead(S1) == LOW)
  {
    if (switch_ON_Flag1_previous_I == 0 )
    {
      digitalWrite(relay1, LOW);
      Serial.println("Relay1 ON");
      Blynk.virtualWrite(V1, 1);
      switch_ON_Flag1_previous_I = 1;
    }
    Serial.println("Switch1 ON");

  }
  if (digitalRead(S1) == HIGH )
  {
    if (switch_ON_Flag1_previous_I == 1)
    {
      digitalWrite(relay1, HIGH);
    Serial.println("Relay1 OFF");
    Blynk.virtualWrite(V1, 0);
    switch_ON_Flag1_previous_I = 0;
  }
  Serial.println("Switch1 OFF");
}


if (digitalRead(S2) == LOW)
{
  if (switch_ON_Flag1_previous_II == 0 )
  {
    digitalWrite(relay2, LOW);
    Serial.println("Relay2 ON");
    Blynk.virtualWrite(V2, 1);
    switch_ON_Flag1_previous_II = 1;
  }
  Serial.println("Switch2 ON");

}
if (digitalRead(S2) == HIGH )
{
  if (switch_ON_Flag1_previous_II == 1)
  {
    digitalWrite(relay2, HIGH);
    Serial.println("Relay2 OFF");
    Blynk.virtualWrite(V2, 0);
    switch_ON_Flag1_previous_II = 0;
  }
  Serial.println("Switch2 OFF");
}
```

```
if (digitalRead(S3) == LOW)
{
  if (switch_ON_Flag1_previous_III == 0 )
  {
    digitalWrite(relay3, LOW);
    Serial.println("Relay3 ON");
    Blynk.virtualWrite(V3, 1);
    switch_ON_Flag1_previous_III = 1;
  }
  Serial.println("Switch3 ON");

}
if (digitalRead(S3) == HIGH )
{
  if (switch_ON_Flag1_previous_III == 1)
  {
    digitalWrite(relay3, HIGH);
    Serial.println("Relay3 OFF");
    Blynk.virtualWrite(V3, 0);
    switch_ON_Flag1_previous_III = 0;
  }
  Serial.println("Switch3 OFF");
}



if (digitalRead(S4) == LOW)
{
  if (switch_ON_Flag1_previous_IV == 0 )
  {
      digitalWrite(relay4, LOW);
      Serial.println("Relay4 ON");
      Blynk.virtualWrite(V4, 1);
      switch_ON_Flag1_previous_IV = 1;
    }
    Serial.println("Switch4 ON");

  }
  if (digitalRead(S4) == HIGH )
  {
    if (switch_ON_Flag1_previous_IV == 1)
    {
      digitalWrite(relay4, HIGH);
      Serial.println("Relay4 OFF");
      Blynk.virtualWrite(V4, 0);
      switch_ON_Flag1_previous_IV = 0;
    }
    Serial.println("Switch4 OFF");
  }
}
```

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

#define BLYNK_TEMPLATE_ID "*********"
#define BLYNK_DEVICE_NAME "HASproject"
#define BLYNK_AUTH_TOKEN "***************************"

char auth[] = "jafynrefI2HUwm3fq46d1NUnZuqxQz7l";//Enter your Auth token
char ssid[] = "Mi 10i";//Enter your WIFI name
char pass[] = "salil12345";//Enter your WIFI password

BlynkTimer timer;
// Define the component pins
#define trig D7
#define echo D8

//Enter your tank max value(CM)
int MaxLevel = 100;

int Level1 = (MaxLevel * 75) / 100;
int Level2 = (MaxLevel * 65) / 100;
int Level3 = (MaxLevel * 55) / 100;
int Level4 = (MaxLevel * 45) / 100;
int Level5 = (MaxLevel * 35) / 100;

//Get the ultrasonic sensor values
void ultrasonic() {
  digitalWrite(trig, LOW);
  delayMicroseconds(4);
  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);
  long t = pulseIn(echo, HIGH);
  int distance = t/5/2 ;

  int blynkDistance = (distance - MaxLevel) * -1;
  if (distance <= MaxLevel) {
    Blynk.virtualWrite(V0, blynkDistance);
  }
  else {
    Blynk.virtualWrite(V0, 0);
  }
}

void setup() {
  Serial.begin(9600);
  pinMode(trig, OUTPUT);
  pinMode(echo, INPUT);



  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
  timer.setInterval(100L, ultrasonic);
}

void loop() {
  Blynk.run();//Run the Blynk library
  timer.run();//Run the Blynk timer
}
```

Conclusion

The project enables us to observe the level of water from a distant location and helps us to track it and protect it from overflowing and thereby enabling the user to ensure that no extra water gets used and there is no excess loss of water. We know that the major place where water gets wasted is industries and homes. So using this proposed system will help tominimize the water loss to a large extent. Bf the user knows about the water level in real-time he/she has the power to maintain the water loss to an extent by maintaining the water at a sustainable height, i.e., between 30-70% of the height of the original tank to ensure no overflowing.

This project also helps in managing electronics systems in house to be operated