SMART WATER MANAGEMENT

DESCRIPTION:

The project involves implemending IoT sensors to monitor water consumption in public places such as parks and gardens. The objective is to promote water conservation by making real time water consumption data publicly available. This project includes defining objectives, designing theIoT sensor system,developing the data sharing platform and integrating them using IoT technology and python.

DESIGN THINKING :

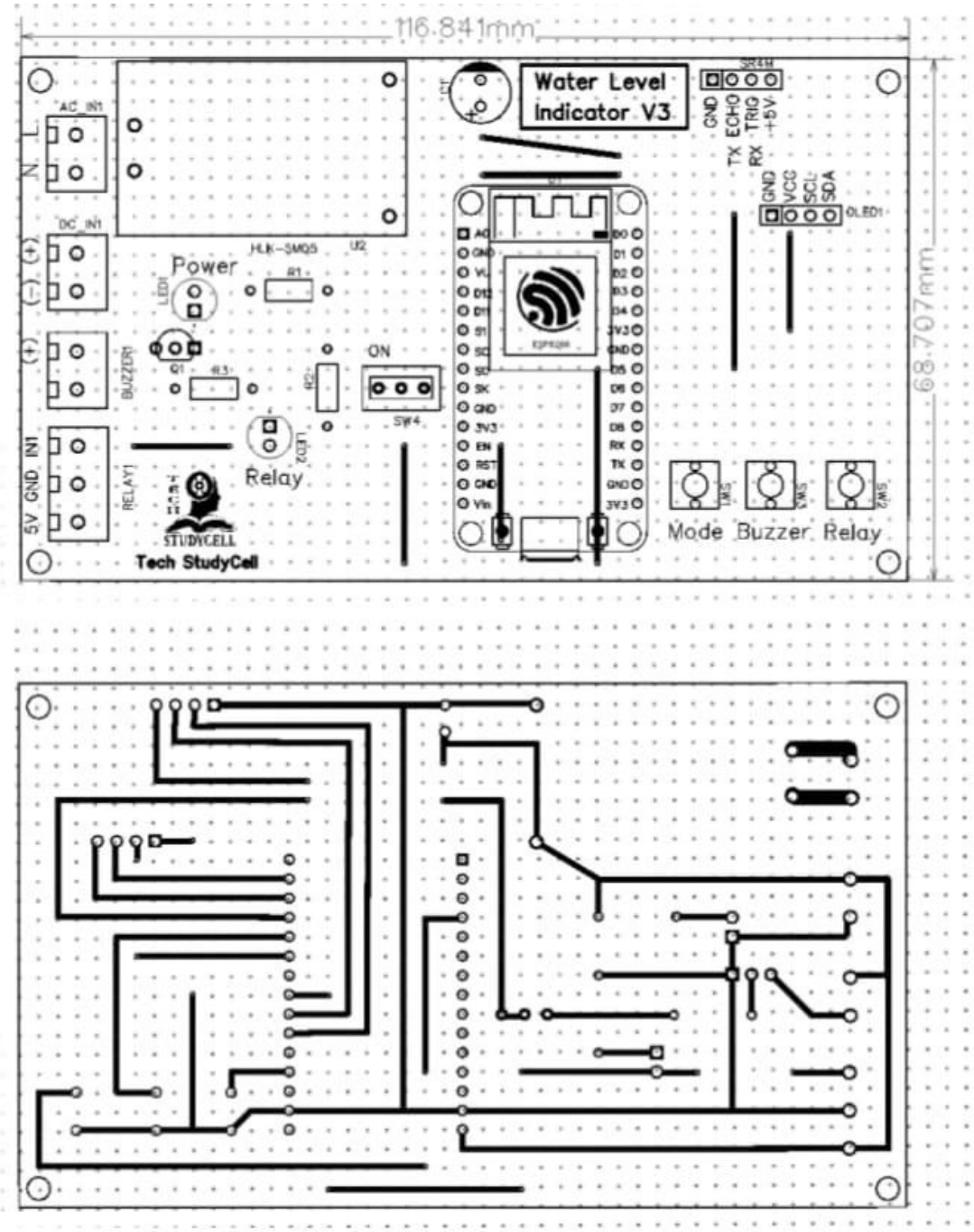
1.Project Objectives: Define Objectives such as real time waterconsumptionmonitorin,public awareness, water conservation and sustainable resource management.

2.IoT sensor Design: Plan the design and deployment of IoT sensors to monitor water consumption in public places.

3.Real timeTransit Information platform: Design a mobile app interface that displays real time parking availability to users.

4.Integration Approach : Determine how to IoT sensors will send data to the sharing platform.

ARCHITECTURE DESIGN FLOW:



DESCRIPTION OF COMPONENTS :

- NodeMCU ESP8266

● SR04M waterproof ultrasonic sensor

OR HC-SR04 sensor .0.96" OLED Display

• 1k 0.25watt Resistors - 3 no

●BC547 NPN Transistor

• LED 5mm - 2no

2-pin Push Button - 3no

● SPDT slider switch

• 220uF 25V DC Capacitor

2-pin Terminal connectors (3 no)

3-pin Terminal connectors (1 no)

• 5V DC Buzzer

• AC to DC converter HLK-5M05 5V (Optional)

HC-SR04 ULTRASONIC SENSOR FUNCTION :

The sensor has 4 pins. VCC and GND go to 5V and GND pins on the Arduino, and the Trig and Echo go to any digital Arduino pin. Using the Trig pin we send the ultrasound wave from the transmitter, and with the Echo pin we listen for the reflected signal.

INNOVATIVE IDEA USE:

1. Reducing waste of water -intensive industries.

2.Monitoring water quality to fight pollution and diseases.

3.Improving the efficiency of water systems.

4.Creating awareness of household water use thanks to smart meters.

5.providing running water through innovative solutions all around the world.

COMPONENTS PROGRAMMING:

HC-SRO4 ULTRASONIC SENSOR PROGRAM

```
// defines pins numbers
const int trigPin = 9;
const int echoPin = 10;
// defines variables
long duration;
int distance;
void setup() {
    pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
    pinMode(echoPin, INPUT); // Sets the echoPin as an Input
    Serial.begin(9600); // Starts the serial communication
}
void loop() {
    // 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
    distance = duration * 0.034 / 2;
    // Prints the distance on the Serial Monitor
    Serial.print("Distance: ");
    Serial.println(distance);
}
```

NODE MCU ESP8266 PROGRAM:

```
#include <ESP8266WiFi.h>
#include <SimpleEsp8266.h>
  // Go to the Project Settings (nut icon).
char auth[] = "pGE7Rb1n3v2SOd9CZvY6RlkeKItyalUR";
  // Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "*******";
char pass[] = "*******";
        int relay1 = 12;    // D6 pin
    int relay2 = 14;    // D2 pin
    int relay3 = 4;      // D1 pin
    int relay4 = 5;      // D5 pin
  void setup()
{
    // Debug console
```

```
    Serial.begin(115200);


    pinMode(relay1,OUTPUT);
    pinMode(relay2,OUTPUT);
    pinMode(relay3,OUTPUT);
    pinMode(relay4,OUTPUT);


    digitalWrite(relay1, HIGH);
    digitalWrite(relay2, HIGH);
    digitalWrite(relay3, HIGH);
    digitalWrite(relay4, HIGH);
  }
  void loop()
{
```