```
#include <WiFi.h>         // For Wi-Fi connectivity
#include <HTTPClient.h>   // For HTTP requests to ThingSpeak
#include <NewPing.h>      // For ultrasonic sensor

// Wi-Fi credentials
const char* ssid = "work shop - 1";       // Replace with your Wi-Fi SSID
const char* password = "12345678";       // Replace with your Wi-Fi password

// ThingSpeak API
const char* server = "http://api.thingspeak.com/update"; // ThingSpeak API URL
const char* apiKey = "SNIM51Q47MRX3L6G";                // Replace with your actual
API key

// Flow Sensor Setup
#define FLOW_SENSOR_PIN 26  // GPIO pin connected to the flow sensor signal (yellow
wire)
volatile int pulseCount = 0;  // Counter for pulses
float totalVolume = 0.0;      // Total volume of water passed (in liters)

// Ultrasonic Sensor Setup
#define TRIG_PIN 33
#define ECHO_PIN 32
#define MAX_DISTANCE 200 // Maximum distance to measure (in centimeters)
NewPing sonar(TRIG_PIN, ECHO_PIN, MAX_DISTANCE);

// Interrupt function to count pulses
void IRAM_ATTR pulseCounter() {
  pulseCount++;
}

void setup() {
  Serial.begin(115200); // Start serial communication

  // Initialize Flow Sensor
  pinMode(FLOW_SENSOR_PIN, INPUT_PULLUP);  // Configure the signal pin with pull-up
  attachInterrupt(digitalPinToInterrupt(FLOW_SENSOR_PIN), pulseCounter,
FALLING);  // Interrupt on falling edge

  // Connect to Wi-Fi
  Serial.print("Connecting to Wi-Fi");
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\nWi-Fi Connected!");
  Serial.print("IP Address: ");
  Serial.println(WiFi.localIP());
  Serial.println("Sensors Initialized!");
}

void loop() {
  unsigned long startMillis = millis();
  unsigned long interval = 2000;  // 2 seconds interval
  int startCount = pulseCount;

  while (millis() - startMillis < interval) {}

  int endCount = pulseCount;
```

```
  int pulseFrequency = endCount - startCount;

  Serial.print("Pulse Count in 2 sec: ");
  Serial.println(pulseFrequency);

  float flowRate = 0.0;
  if (pulseFrequency > 0) {
    flowRate = (float)pulseFrequency * 60.0 / 450.0;  // Adjust 450 based on your
sensor's PPL rating
    totalVolume += (float)pulseFrequency / 450.0;    // Add volume in liters

    Serial.print("Flow Rate: ");
    Serial.print(flowRate, 2);
    Serial.println(" L/min");

    Serial.print("Total Volume: ");
    Serial.print(totalVolume, 2);
    Serial.println(" L");
  } else {
    Serial.println("No flow detected (Pulses = 0).");
  }
  unsigned int distance = sonar.ping_cm();
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
  sendToThingSpeak(flowRate, totalVolume, distance);  // -1 for flowRate and
totalVolume (not measured here)
  delay(15000);  // Delay before next reading (ThingSpeak limit: 15 sec)
}

void measureFlow() {
  unsigned long startMillis = millis();
  unsigned long interval = 2000;  // 2 seconds interval
  int startCount = pulseCount;

  while (millis() - startMillis < interval) {}

  int endCount = pulseCount;
  int pulseFrequency = endCount - startCount;

  Serial.print("Pulse Count in 2 sec: ");
  Serial.println(pulseFrequency);

  float flowRate = 0.0;
  if (pulseFrequency > 0) {
    flowRate = (float)pulseFrequency * 60.0 / 450.0;  // Adjust 450 based on your
sensor's PPL rating
    totalVolume += (float)pulseFrequency / 450.0;    // Add volume in liters

    Serial.print("Flow Rate: ");
    Serial.print(flowRate, 2);
    Serial.println(" L/min");

    Serial.print("Total Volume: ");
    Serial.print(totalVolume, 2);
    Serial.println(" L");
  } else {
    Serial.println("No flow detected (Pulses = 0).");
  }
```

```
  sendToThingSpeak(flowRate, totalVolume, -1);  // -1 for distance (not measured
here)
}

void measureDistance() {
  unsigned int distance = sonar.ping_cm();
  if (distance > 0) {
    Serial.print("Distance: ");
    Serial.print(distance);
    Serial.println(" cm");
    sendToThingSpeak(-1, -1, distance);  // -1 for flowRate and totalVolume (not
measured here)
  } else {
    Serial.println("Out of Range");
  }
}

void sendToThingSpeak(float flowRate, float totalVolume, int distance) {
  if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;
    String url = String(server) + "?api_key=" + apiKey;

    if (flowRate >= 0) url += "&field4=" + String(flowRate, 2);
    if (totalVolume >= 0) url += "&field5=" + String(totalVolume, 2);
    if (distance >= 0) url += "&field1=" + String(distance);

    http.begin(url); // Initialize HTTP client
    int httpResponseCode = http.GET();

    if (httpResponseCode > 0) {
      Serial.print("ThingSpeak Response: ");
      Serial.println(httpResponseCode);
    } else {
      Serial.print("Error Sending Data: ");
      Serial.println(http.errorToString(httpResponseCode));
    }
    http.end();
  } else {
    Serial.println("Wi-Fi Disconnected! Reconnecting...");
    WiFi.reconnect();
  }
}
```