JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

REAL-TIME STREET DRAINAGE MONITORING SYSTEM POWERED BY IOT

T.Parthiban,
Assistant Professor
Department of Electrical and Electronics Engineering
Sri Bharathi Engineering College for Women, Pudukkottai, Tamil Nadu, India.

Abstract: Urbanization and climate change are straining drainage infrastructure, leading to overflows and blockages that pose health and environmental risks. To address this, we're developing a real-time Drainage Monitoring System leveraging IoT technology. This innovative solution employs sensors to track water levels, detect hazardous gases, and monitor water quality. The data is transmitted to a cloud platform or mobile app via Wi-Fi, enabling swift response to potential issues. The system features automated alerts and alarms to prevent overflows, ensuring a proactive approach to drainage management. This cost-effective and maintainable solution is adaptable to urban and rural settings, promoting public health, environmental protection, and efficient drainage management.

Introduction:

Rapid urbanization has strained drainage infrastructure, leading to issues like waterlogging and waterborne diseases. Traditional inspection methods are inadequate, highlighting the need for a modern solution. The Drainage Monitoring System utilizes IoT technology to track water levels, detect hazardous gases, and assess water quality. Sensors transmit data to a centralized system via Wi-Fi, enabling real-time monitoring and alerts when thresholds are exceeded. This proactive approach allows authorities to take timely action, preventing serious incidents. The system also facilitates long-term analysis, helping identify recurring issues and inform maintenance planning. Its scalability and automation make it suitable for urban and rural environments, reducing the need for human supervision. By leveraging modern technology, this project improves drainage maintenance, protects the environment, and enhances public safety. It demonstrates the potential of IoT to address civic challenges effectively and affordably. The system's benefits include improved efficiency, reduced disease outbreaks, and enhanced public health. Its implementation can have a significant impact on communities, making it a valuable solution for modern drainage management. Several projects have explored IoT-based solutions for drainage system management, focusing on issues like blockages, contamination, and overflow.

Existing Projects:

1. Water Level Monitoring: Some projects use ultrasonic sensors to track water levels, sending alerts when thresholds are exceeded. However, these systems may lack scalability and integration with other critical parameters.
2. Gas Detection: Other projects focus on detecting hazardous gases like methane and hydrogen sulphide using gas sensors. These systems may not provide comprehensive monitoring of entire drainage networks.
3. Water Quality Analysis: Projects monitoring wastewater quality use sensors to track parameters like pH, TDS, and turbidity. However, these systems may not offer real-time alerts or monitor water levels. These projects often have limitations, including:
1. Limited scope: Many projects focus on a single aspect of drainage management.
2. Lack of scalability: Some systems may not be suitable for large-scale implementation.
3. Inadequate alert systems: Some projects rely on outdated notification methods or lack real-time alerts. There is a need for a comprehensive drainage management system that integrates multiple parameters, including water level, gas detection, and water quality monitoring, with real- time alerts and scalable architecture.

Proposed System:

This innovative system tracks drainage conditions in real-time, alerting authorities to potential hazards like flooding, gas leaks, and poor air and water quality. It uses sensors connected to a microcontroller to collect data and send it to a remote monitoring platform. The system's components include:

1. Water level sensors to detect blockages or flooding risks.
2. Gas sensors to identify harmful gases like methane and hydrogen sulphide.
3. Temperature and humidity sensors to track environmental conditions.
4. Fire sensors to detect flames or unusual heat.
5. GPS modules to track the exact location of the drainage system.

When abnormal conditions are detected, the system sends alerts and notifications to authorities, including GPS coordinates to facilitate targeted responses. The system also features an on-site alarm to warn nearby workers or residents. This smart monitoring system offers several benefits, including:

1. Reduced need for manual inspections
2. Improved response times
3. Prevention of health hazards
4. Support for cleaner, safer urban drainage systems

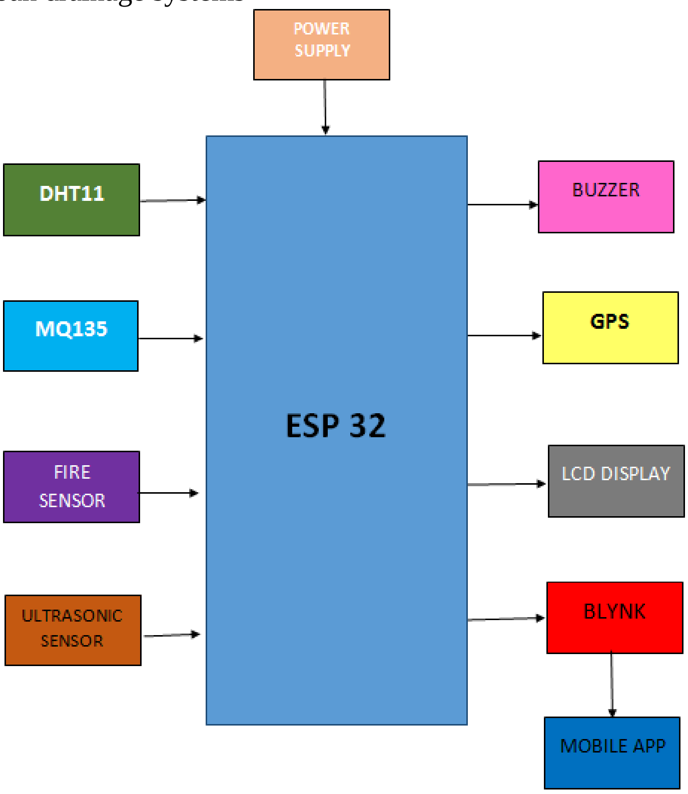

Fig: Block diagram of proposed system

Embedded System:

The system is energy-efficient and can be powered by batteries or solar panels, making it suitable for remote areas. A specialized computer system is integrated into the Street Drainage Monitoring System to perform a specific function. This system consists of hardware and software components, with the NodeMCU ESP32 serving as the central controller. The NodeMCU ESP32 collects data from sensors, processes it, and takes actions such as triggering alerts or sounding alarms. This enables the system to detect and respond to drainage issues promptly and automatically, reducing the need for manual intervention. By leveraging this technology, the system can efficiently monitor drainage conditions, identify potential problems, and notify authorities, ultimately contributing to improved public safety and infrastructure management. The development of an embedded system follows a structured process that takes it from concept to deployment and beyond. This process begins with identifying the system's purpose, requirements, and performance expectations. A detailed specification is then created, outlining the system's functional and non-functional requirements. The next steps involve deciding which functions will be implemented in hardware and software, selecting suitable hardware components, and designing the system's architecture. The software is then developed using programming languages like C or C++, and the hardware and software are integrated and tested.

The system is validated under real-world conditions to ensure reliability and accuracy before deployment. After deployment, the system requires regular monitoring, maintenance, and potential upgrades to ensure it continues to operate effectively and adapt to changing needs. This development process ensures the embedded system meets its design specifications, operates reliably, and can evolve over time.

Embedded systems have distinct characteristics that make them ideal for specific applications. They operate in real-time, responding to inputs and delivering outputs within strict time limits, which is critical for safety-critical tasks. These systems are designed to perform dedicated functions, unlike general-purpose computers. They are engineered to consume low power, making them suitable for battery-operated devices. Their compact and lightweight design allows for easy integration into larger machines or portable devices. Reliability and stability are crucial, as these systems often run continuously without failure. Embedded systems are also efficient, using minimal processing power and memory to perform tasks effectively. They are designed to be cost-effective, making them viable for mass production. The software in these systems is typically embedded permanently, ensuring consistent performance and protection from modifications. These characteristics enable embedded systems to provide dependable and focused performance, making them well-suited for automation and control in various industries.

Programm:

```
#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <DHT.h>
#include <TinyGPSPlus.h>
#include <HardwareSerial.h>
#define BLYNK_TEMPLATE_ID "TMPL3_cAcVUBE"
#define BLYNK_TEMPLATE_NAME "drainage monitoring"
#define BLYNK_AUTH_TOKEN "iTwotDjzWch1MIr2tv-EaMe8bC0QCP7U"
char ssid[] = "NANOTECH";
char pass[] = "123456789";
#define DHTPIN 4
#define DHTTYPE DHT11
#define MQ_PIN 34
#define FIRE_PIN 35
#define TRIG_PIN 12
#define ECHO_PIN 14
#define BUZZER_PIN 27
int gasThresholdPercent = 70;
int waterLevelThresholdCM = 4;
DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x27, 16, 2);
TinyGPSPlus gps;
HardwareSerial gpsSerial(1);
unsigned long lastUpdate = 0;
const unsigned long updateInterval = 1000;
void setup() {
  Serial.begin(115200);
  gpsSerial.begin(9600, SERIAL_8N1, 16, 17);
  dht.begin();
  lcd.init();
  lcd.backlight();
  pinMode(FIRE_PIN, INPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  lcd.setCursor(0, 0);
  lcd.print("Drainage Monitor");
  delay(2000);
  lcd.clear();
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
}
void loop() {
  // Run Blynk
  Blynk.run();
```

```
  while (gpsSerial.available()) {
    gps.encode(gpsSerial.read());
  }

 if (millis() - lastUpdate >= updateInterval) {
    lastUpdate = millis();
    float temp = dht.readTemperature();
    float humi = dht.readHumidity();
    int gasValue = analogRead(MQ_PIN);
    int gasPercent = map(gasValue, 0, 4095, 0, 100);
    digitalWrite(TRIG_PIN, LOW);
    delayMicroseconds(2);
    digitalWrite(TRIG_PIN, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIG_PIN, LOW);
    long duration = pulseIn(ECHO_PIN, HIGH);
    float distanceCM = duration * 0.034 / 2;
     bool fireDetected = digitalRead(FIRE_PIN) == LOW;
    Blynk.virtualWrite(V0, temp);
    Blynk.virtualWrite(V1, humi);
    Blynk.virtualWrite(V2, gasPercent);
    Blynk.virtualWrite(V3, distanceCM);
    Blynk.virtualWrite(V4, fireDetected);
    if (gps.location.isValid()) {
      float latitude = gps.location.lat();
      float longitude = gps.location.lng();
      Blynk.virtualWrite(V6, latitude);
      Blynk.virtualWrite(V7, longitude);
    } else {
      Blynk.virtualWrite(V6, 0);
      Blynk.virtualWrite(V7, 0);
    }
    lcd.setCursor(0, 0);
    lcd.print("T:");
    lcd.print(temp, 0);
    lcd.print(" H:");
    lcd.print(humi, 0);
    lcd.print("% ");
    lcd.setCursor(0, 1);
    lcd.print("G:");
    lcd.print(gasPercent);
    lcd.print("% W:");
    lcd.print((int)distanceCM);
    lcd.print("cm");
    bool alert = false;
    if (gasPercent > gasThresholdPercent || fireDetected || distanceCM <= waterLevelThresholdCM) {
      lcd.clear();
      lcd.setCursor(0, 0);
      if (gasPercent > gasThresholdPercent) {
        lcd.print("!! Gas Alert !!");
        alert = true;
```

Result and Conclusion:

This project showcases a reliable and efficient system for monitoring underground drainage conditions in real-time. Using a microcontroller and various sensors, the system collects data on water levels, gas concentrations, temperature, and humidity. This data is displayed locally and transmitted over the internet, enabling remote monitoring. The system effectively detects harmful gases and accurately measures water levels, helping to identify potential issues. It also tracks ambient temperature and humidity, providing valuable insights into sewer conditions. Location data is provided for detected issues, facilitating prompt maintenance responses. When thresholds are exceeded, the system triggers alarms and sends real-time alerts, enabling swift action to mitigate risks. This solution is low-cost, scalable, and suitable for smart city applications, ultimately contributing to improved sanitation management and public health and safety.

The Drainage Monitoring System effectively utilizes IoT technology to track drainage conditions in real-time. By combining various sensors, this system provides a comprehensive solution for managing drainage infrastructure. The system's key features include real-time data collection, enabling prompt issue detection. It also sends immediate alerts and notifications when predefined thresholds are breached, facilitating swift intervention. Users can easily monitor the system's status locally and remotely, enhancing accessibility. The system is cost-effective, scalable, and suitable for large drainage networks. It contributes to improving public health, safety, and the environment by detecting hazardous conditions and reducing contamination risks. The system's adaptability allows for future enhancements, such as adding sensors or integrating with other smart city solutions. This makes it a viable solution for sustainable drainage management.

References:

1. Iot-based underground drainage monitoring system using pic microcontroller Booma Devi C, Lakshanaa D, Jafrilsheran J, Logalakshmibala N, International Journal Of Engineering Research & Technology (IJERT), 2022.

2. Underground drainage monitoring system using IOTA, R. Sarath Kumar, P. Ananth, M. Mukunthan, N. Parthipan, Dr. B. Maruthi Shankar, International Journal of Advanced Science and Technology, 2020.

3. An iot-based water-logging detection system: a case study of dhaka, Authors: Md. Manirul Islam, Md. SadadMahamud, UmmeSalsabil, A. A. M. Mazharul Amin, SamiulHaque Suman, arXiv preprint, 2024

4. IOT-based detection of blockages in stormwater drains, MDPI Proceedings, 2023

5. An iot-based smart drain monitoring system with alert messages, Samiha Sultana, AnanyaRahaman, Anita Mahmud Jhara, Akash Chandra Paul, Jia Uddin, Lecture Notes in Computer Science, 2021

6. Urban underground drainage monitoring system using IOT, D. Deepak Kumar, N. Indrajith, R. Bhavya, Dr. V. Jayaprakasan, International Journal of Advance Research, Ideas and Innovations in Technology (IJARIIT), 2021

7. Smart real-time drainage monitoring system using internet of thing, Chetan Mahajan, GaurangSonawane, Yogita Dalvi, AnujaNikale, IRE Journals, 2018.

8. IoT-based Drainage System Monitoring." International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 5(2), 236-240.