# **CHAPTER 1**

# **INTRODUCTION**

The Manhole Cover Monitoring System with NodeMCU presents an innovative solution designed to revolutionize urban infrastructure management by ensuring the efficient monitoring of manhole covers and water levels. With a primary focus on safety and operational efficiency, this project aims to address the shortcomings of traditional monitoring methods commonly associated with manhole covers.

Manual inspection of manhole covers is often labor-intensive and prone to human error, leading to delays in identifying potential hazards. In response to these challenges, the Manhole Cover Monitoring System integrates advanced sensor technology and automation to provide real-time feedback on the status of manhole covers and water levels within the manhole.

Central to the project's objectives is the implementation of continuous monitoring capabilities. By leveraging state-of-the-art sensors and monitoring devices, the system offers real-time insights into the condition of manhole covers and water levels. This real-time aspect is critical, particularly in urban environments where the timely detection of issues is essential for public safety and infrastructure maintenance.

The system's intelligent algorithms play a crucial role in early fault detection, swiftly identifying anomalies or faults in the manhole cover status and water levels. This proactive approach not only enhances operational efficiency but also ensures the timely response to potential hazards, minimizing the risk of accidents or disruptions.

Key features of the Manhole Cover Monitoring System include its commitment to remote accessibility. By enabling remote monitoring capabilities, operators and maintenance personnel can access real-time data and receive alerts from any location. This level of accessibility empowers quick decision-making and facilitates a proactive approach to infrastructure management.

Moreover, the system is designed to seamlessly integrate with existing urban infrastructure, facilitating compatibility with municipal systems and databases. By adhering to industry standards and regulations governing infrastructure management, the project prioritizes safety and reliability in urban environments.

In summary, the Manhole Cover Monitoring System with NodeMCU represents a significant advancement in urban infrastructure management.

Its integration of real-time monitoring, early fault detection, and remote accessibility underscores its commitment to safety and operational efficiency. By leveraging technology and automation, this project aims to enhance the resilience and reliability of urban infrastructure, ultimately contributing to safer and more sustainable cities.

## **1.1: Problem Statement:**

The urban infrastructure management sector faces significant challenges in effectively monitoring and managing manhole covers and water levels within manholes. Current methods rely heavily on manual inspections, which are time-consuming, labor-intensive, and prone to errors. This outdated approach poses risks to public safety and operational continuity, as delays in identifying faults or anomalies can lead to potential hazards and infrastructure disruptions.

Manual monitoring systems are particularly susceptible to human error, resulting in delays in detecting issues with manhole covers or water levels. In critical situations, such as during heavy rainfall or emergencies, these delays can have severe consequences, including flooding, sewer overflow, and accidents.

Furthermore, the lack of real-time monitoring exacerbates the challenges associated with the current infrastructure. Operators and maintenance personnel struggle to promptly detect faults or disruptions in manhole covers and water levels, hindering their ability to mitigate potential risks effectively.

The absence of remote accessibility further compounds these challenges. Without the ability to monitor manhole covers and water levels remotely, operators and maintenance personnel cannot respond swiftly to emerging issues, increasing the likelihood of infrastructure failures and safety incidents.

Additionally, the disjointed nature of existing systems impedes seamless integration with municipal infrastructure, including drainage systems and utility networks. This lack of integration not only limits data accessibility and interoperability but also adds complexity to overall infrastructure management.

Moreover, compliance with safety standards and regulations governing urban infrastructure is essential to ensure public safety. The absence of a dedicated monitoring system designed to meet or exceed these standards leaves municipalities vulnerable to safety breaches and regulatory non-compliance, jeopardizing the well-being of residents and infrastructure integrity.



Figure 1.1: Manhole cover

The current state of manhole cover and water level monitoring in urban infrastructure management is characterized by manual processes, delayed fault detection, lack of real-time monitoring, insufficient remote accessibility, and inadequate integration with existing infrastructure. These deficiencies underscore the urgent need for a comprehensive Manhole Cover Monitoring System with NodeMCU to address these challenges and ensure a safer, more efficient, and technologically advanced approach to managing urban infrastructure.

## **1.2: Problem Scope**:

The problem scope for implementing a Manhole Cover Monitoring System with NodeMCU encompasses several challenges and deficiencies within the current urban infrastructure management framework. These limitations hinder operational efficiency, compromise public safety, and impede the overall management of urban infrastructure.

**Manual Monitoring and Human Error:**

* Reliance on manual inspections for monitoring manhole covers introduces a significant risk of human error.
* Manual monitoring is time-consuming and may result in delays in identifying faults or anomalies, leaving critical areas vulnerable to potential hazards.

**Delayed Fault Detection:**

* The absence of real-time monitoring systems contributes to delayed detection of faults in manhole covers and water levels.
* Delays in identifying issues pose a substantial risk, particularly during adverse weather conditions or emergencies.

**Lack of Remote Accessibility:**

* The current systems lack remote monitoring capabilities, limiting access to real-time data from different locations.
* Remote inaccessibility hampers quick decision-making and proactive response to emerging issues, increasing the likelihood of infrastructure failures.

**Integration Challenges:**

* Existing systems often lack seamless integration with municipal infrastructure, complicating data flow and decision-making processes.
* The lack of integration with other urban infrastructure systems hinders interoperability and overall infrastructure management.

**Safety and Compliance Concerns:**

* Compliance with safety standards and regulations governing urban infrastructure management is paramount.
* The absence of a dedicated monitoring system designed to meet or exceed these standards leaves municipalities vulnerable to safety breaches and regulatory non-compliance.

**Operational Disruptions:**

* Inefficient monitoring and delayed fault detection contribute to the potential for operational disruptions in urban infrastructure.
* Operational disruptions pose a direct threat to public safety and can result in significant economic losses.

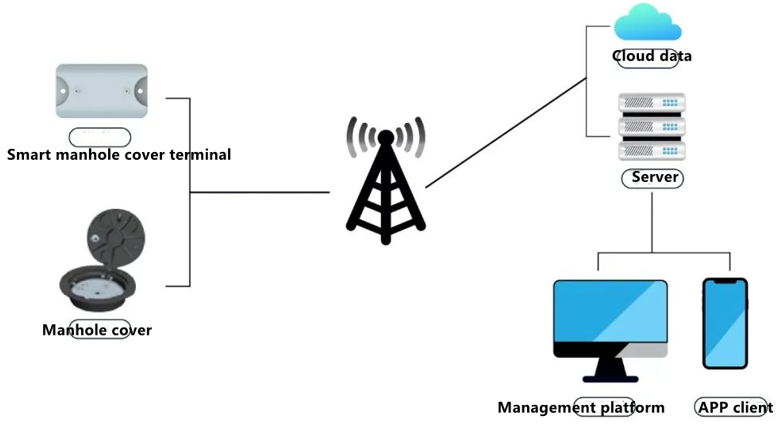
**Complex Facility Management:**

* The disjointed nature of existing systems adds complexity to overall facility management.
* Lack of integration with other municipal systems complicates data accessibility and decision-making processes.

**Risk to Public Safety:**

* The cumulative impact of manual monitoring, delayed fault detection, and operational disruptions poses a direct risk to public safety.
* Timely and reliable monitoring of manhole covers and water levels is essential for preventing accidents and ensuring the integrity of urban infrastructure.

Addressing these challenges within the defined problem scope requires the development and implementation of a comprehensive Manhole Cover Monitoring System with NodeMCU. This system aims to integrate advanced technologies, real-time monitoring capabilities, remote accessibility, and adherence to safety standards to enhance the efficiency and safety of urban infrastructure management. Ultimately, this project will contribute to improved public safety and the sustainable management of urban infrastructure.

Figure1.2:Technologies used for monitoring the manhole

## 

## **1.3: Advantages of Implementing a Manhole Cover Monitoring System with NodeMCU:**

## The implementation of a Manhole Cover Monitoring System with NodeMCU presents numerous advantages, significantly improving the efficiency, safety, and overall management of urban infrastructure. Key advantages include:

**Real-time Monitoring:**

* Enables continuous and real-time monitoring of manhole covers and water levels within the manhole.
* Promptly detects any anomalies, ensuring a timely response to potential hazards such as open manhole covers or rising water levels.

**Early Fault Detection:**

* Utilizes intelligent algorithms to detect faults and anomalies in manhole covers and water levels at an early stage.
* Minimizes the risk of accidents, injuries, and infrastructure damage by addressing issues before they escalate.

**Enhanced Public Safety:**

* Reduces the risk of accidents and injuries caused by open manhole covers or flooding in urban areas.
* Contributes to safer streets and sidewalks for pedestrians and motorists, preventing potential hazards.

**Remote Accessibility:**

* Facilitates remote monitoring of manhole covers and water levels from various locations.
* Enables operators and maintenance personnel to respond swiftly to emerging issues, even when not physically present on-site.

**Improved Operational Efficiency:**

* Streamlines the monitoring and management of urban infrastructure, reducing the need for manual inspections.
* Enhances operational efficiency by providing real-time data and enabling proactive decision-making in infrastructure maintenance.

**Seamless Integration:**

* Integrates seamlessly with existing municipal infrastructure, including drainage systems and utility networks.
* Ensures data accessibility, interoperability, and a cohesive approach to overall urban infrastructure management.

**Compliance with Safety Standards:**

* Adheres to safety standards and regulations governing urban infrastructure management.
* Mitigates the risk of safety breaches and ensures a secure and compliant urban environment for residents and pedestrians.

**Cost Efficiency:**

* Reduces operational costs associated with manual monitoring and potential infrastructure damage caused by delayed fault detection.
* Minimizes the financial impact of emergency interventions and downtime in urban infrastructure maintenance.

**Data Logging and Analysis:**

* Records and logs data related to manhole covers and water levels, facilitating historical analysis and trend identification.
* Supports data-driven decision-making and optimization of urban infrastructure maintenance schedules.

**Proactive Maintenance:**

* Facilitates predictive maintenance by identifying potential issues before they escalate into major infrastructure failures.
* Reduces the need for reactive and costly repairs, contributing to long-term cost savings in urban infrastructure management.

**Enhanced Emergency Preparedness:**

* Strengthens emergency response capabilities by providing real-time information during critical situations such as heavy rainfall or flooding.
* Enables municipalities to respond swiftly to unforeseen circumstances and emergencies, minimizing the impact on public safety and infrastructure integrity.

**Environmental Sustainability:**

* Contributes to environmental sustainability by optimizing urban infrastructure management practices and minimizing the risk of pollution or environmental damage.
* Supports eco-friendly initiatives aimed at promoting sustainable urban development and infrastructure maintenance.

The implementation of a Manhole Cover Monitoring System with NodeMCU offers a comprehensive solution to the challenges associated with managing urban infrastructure. By providing real-time monitoring, early fault detection, remote accessibility, and seamless integration, the system not only enhances public safety but also promotes operational efficiency and compliance with safety standards, ultimately contributing to the sustainable development of urban environments.

## **1.4 Proposed Solution:**

The proposed solution for optimizing the Manhole Cover Monitoring System with NodeMCU involves the strategic integration of Internet of Things (IoT) technologies. By harnessing the power of IoT, the system transitions into a smart, interconnected infrastructure, offering real-time monitoring, remote accessibility, and advanced analytics capabilities. Key components of the solution include:

**IoT-enabled Sensors:** A network of IoT-enabled sensors will be strategically deployed to monitor the status of manhole covers and water levels within the manholes. These sensors will wirelessly transmit data to a centralized monitoring system, providing real-time insights into urban infrastructure.

**Centralized Monitoring System:** Empowered by edge computing, the centralized monitoring system will process and analyze incoming sensor data in real-time. This system will offer immediate insights into the condition of manhole covers and water levels, enabling proactive decision-making and swift corrective actions.

**User-Friendly Dashboard:** A user-friendly dashboard accessible remotely via web interfaces or mobile applications will provide healthcare professionals and facility managers with comprehensive insights into the urban infrastructure. The dashboard will display real-time data, alerts, and predictive analytics, empowering stakeholders to monitor the system and take proactive measures.

**Machine Learning Algorithms:** Advanced machine learning algorithms will contribute to predictive analytics, forecasting potential issues and enabling proactive maintenance. These algorithms will analyze historical data patterns to identify trends and predict future infrastructure needs.

**Seamless Integration:** The solution will prioritize seamless integration with existing municipal infrastructure, including drainage systems and utility networks. This integration will ensure interoperability and data exchange, facilitating a cohesive approach to urban infrastructure management.

**Robust Security Measures:** The proposed solution will incorporate robust security measures to safeguard sensitive data and infrastructure assets. Encryption protocols, access controls, and intrusion detection systems will be implemented to mitigate cybersecurity risks.

**Scalability and Energy Efficiency:** Designed for scalability, the solution will accommodate future growth and expansion of urban infrastructure. Additionally, energy-efficient IoT devices and protocols will minimize power consumption and environmental impact.

By implementing this IoT-based solution, municipalities can enhance public safety, improve operational efficiency, and optimize urban infrastructure management. The proposed solution aims to address the challenges associated with manual monitoring, delayed fault detection, and inefficient infrastructure management, ultimately contributing to the sustainable development of urban environments.

**1.5 Aim and Objectives**

**Aim:**

The primary aim of implementing a Manhole Cover Monitoring System with NodeMCU is to revolutionize urban infrastructure management by ensuring the safety, reliability, and efficiency of manhole covers and water levels monitoring. This system aims to mitigate potential hazards, enhance public safety, and streamline operational processes through real-time monitoring, early fault detection, seamless integration with existing municipal infrastructure, and adherence to safety standards. By leveraging advanced IoT technologies, the aim is to create a holistic solution that not only addresses current challenges but also sets a benchmark for sustainable urban development and infrastructure management.

**Objectives:**

The objectives of implementing an Manhole Cover Monitoring System include:

* **Continuous Public Safety:**
  + Ensure the safety of pedestrians and motorists by maintaining the integrity of manhole covers and preventing potential hazards such as open covers or flooding.
* **Real-time Monitoring:**
  + Implement a continuous and real-time monitoring system for manhole covers and water levels to promptly detect anomalies and faults.
* **Swift Intervention through Early Fault Detection:**
  + Utilize intelligent algorithms to achieve early fault detection, enabling immediate corrective actions and minimizing the risk of accidents or infrastructure damage.
* **Operational Efficiency Enhancement:**
  + Streamline the monitoring and management of urban infrastructure to reduce reliance on manual inspections and improve overall operational efficiency.
* **Remote Accessibility:**
  + Enable operators and maintenance personnel to monitor the status of manhole covers and water levels remotely, facilitating quick responses and interventions regardless of physical location.
* **Integration with Municipal Infrastructure:**
  + Seamlessly integrate the monitoring system with existing municipal infrastructure, including drainage systems and utility networks, to enhance data accessibility and interoperability.
* **Adherence to Safety Standards:**
  + Ensure strict adherence to safety standards and regulations governing urban infrastructure management, mitigating the risk of safety breaches and maintaining a secure urban environment.
* **Cost Efficiency:**
  + Reduce operational costs associated with manual inspections, emergency interventions, and infrastructure damage, optimizing resource utilization and contributing to long-term financial sustainability.
* **Data-driven Decision Making:**
  + Facilitate data logging and analysis to support data-driven decision-making for the optimization of urban infrastructure maintenance schedules.
* **Proactive Maintenance:**
  + Implement predictive maintenance strategies by identifying potential issues before they escalate into major infrastructure failures, reducing the need for reactive and costly repairs.
* **Emergency Preparedness:**
  + Strengthen emergency response capabilities by providing real-time information during critical situations such as heavy rainfall or emergencies, ensuring swift and effective responses to unforeseen circumstances.
* **Environmental Sustainability:**
  + Contribute to environmental sustainability by optimizing urban infrastructure management practices, minimizing waste, and supporting eco-friendly practices in urban development.

By achieving these objectives, the Manhole Cover Monitoring System with NodeMCU aims to create a comprehensive, technologically advanced solution that enhances public safety, improves operational efficiency, and contributes to the sustainable development of urban environments.

# 

# 

# 

# 

# **CHAPTER 2**

# **Literature Survey**

The adoption of a Manhole Cover Monitoring System with NodeMCU, integrated with Internet of Things (IoT) technologies, has emerged as a focal point in urban infrastructure management literature. Researchers have extensively explored avenues to bolster the efficiency and safety of urban infrastructure through continuous monitoring facilitated by IoT-enabled sensors. These sensors are instrumental in acquiring real-time data on critical parameters such as manhole cover status and water levels, fostering proactive measures to ensure the integrity and functionality of urban infrastructure.

In the realm of sensor technology, literature underscores a myriad of sensors utilized in IoT-based systems for monitoring urban infrastructure. Research endeavors have centered on evaluating the accuracy, reliability, and real-time capabilities of these sensors, thereby influencing the overall efficacy of the monitoring system. Various sensor types, including proximity sensors, water level sensors, and accelerometer sensors, are investigated to discern their applicability for specific monitoring requirements.

Communication protocols serve as linchpins in establishing robust connections between IoT-enabled sensors and centralized monitoring systems. Literature extensively scrutinizes diverse communication protocols, with wireless technologies such as LoRaWAN, Zigbee, and NB-IoT emerging as prominent choices. Assessing the strengths and limitations of these protocols is pivotal in ensuring seamless and resilient data transmission, particularly in the context of urban infrastructure monitoring.

Mitigating security and privacy risks remains paramount in the development of IoT systems for urban infrastructure monitoring. The literature survey delves into the implementation of stringent security measures to safeguard sensitive urban data. Encryption methodologies, authentication mechanisms, and secure data transmission protocols are explored to fortify the framework against potential cyber threats and data breaches.

Integration with existing municipal infrastructure emerges as a critical consideration, as highlighted in the literature. Compatibility with Building Management Systems (BMS) and Geographic Information Systems (GIS) is emphasized to foster a cohesive and interconnected urban environment. Understanding the seamless integration of IoT-based solutions with these existing systems augments the feasibility and acceptance of monitoring technologies in urban settings.

Real-world case studies and implementations offer invaluable insights into the practical challenges and lessons gleaned from deploying IoT-based manhole cover monitoring systems in urban environments. Analyzing these cases furnishes a deeper comprehension of the system's performance, its impact on urban operations, and opportunities for optimization.

Recent trends in literature showcase a burgeoning interest in edge computing for expedited data processing. Furthermore, the integration of artificial intelligence (AI) for advanced analytics is gaining traction, enabling sophisticated data interpretation and decision-making. The exploration of emerging technologies such as 5G networks underscores the commitment to technological innovation in urban infrastructure management.

The comprehensive literature survey underscores a concerted effort to develop sophisticated and technologically advanced solutions for monitoring urban infrastructure. The exploration of IoT-enabled sensors, communication protocols, data security, integration with existing infrastructure, and real-world case studies provides a holistic perspective on the current state of research in this domain. The continuous evolution of these technologies epitomizes the ongoing commitment to revolutionize urban infrastructure monitoring systems and enhance public safety.

# **CHAPTER 3**

# **Methodology**

The methodology for implementing a Manhole Cover Monitoring System with NodeMCU, leveraging Internet of Things (IoT) technologies, adopts a systematic approach tailored to the unique demands of urban infrastructure management. It commences with a meticulous needs assessment, engaging municipal authorities and urban planners to identify critical requirements and operational challenges pertaining to manhole cover monitoring and water level detection in urban environments. Subsequently, both functional and technical prerequisites for the system are clearly defined, encompassing essential aspects such as real-time monitoring, remote accessibility, integration with existing infrastructure, and adherence to safety standards.

Selection of appropriate IoT-enabled sensors constitutes a pivotal step, where factors like accuracy, reliability, and power consumption are carefully evaluated to ensure compatibility with the monitoring requirements of manhole covers and water levels. Likewise, the choice of communication protocols, such as LoRaWAN or NB-IoT, is made strategically to facilitate seamless and secure data transmission between the sensors and the central monitoring system.

Development of the centralized monitoring system entails the incorporation of edge computing for local data processing, thereby minimizing latency and ensuring timely access to critical information. An intuitive user interface accessible through web applications or mobile devices is meticulously designed, featuring real-time monitoring, immediate alerts, and insightful visualization of historical data. Robust security measures, including encryption, authentication, and access controls, are implemented to safeguard sensitive urban data and prevent unauthorized access to the monitoring system.

Integration with existing municipal infrastructure, such as Geographic Information Systems (GIS) and utility networks, is carefully addressed to ensure interoperability and cohesive data exchange. Rigorous testing of the entire system, including sensor functionality, data transmission, and central monitoring system performance, is conducted in controlled environments before deployment. Training programs are then conducted for municipal personnel and maintenance crews to ensure effective utilization and management of the monitoring system.

Deployment is undertaken in a phased approach, starting with critical areas, followed by continuous monitoring and evaluation post-implementation. A proactive maintenance schedule is established to address regular updates, patches, and sensor calibration, ensuring the sustained performance of the system over time. Moreover, the incorporation of data analytics and machine learning algorithms enables predictive maintenance, fault prediction, and optimization of urban infrastructure management. Continuous evaluation and adaptation of the monitoring system ensure its effectiveness, efficiency, and alignment with evolving urban infrastructure needs.

IR Sensor

Buzzer

Flow Sensor

NodeMCU

(ESP8266)

Figure 3.1: Block Diagram of the Smart Manhole

## **3.1 NodeMCU (ESP8266 )**

The NodeMCU ESP8266 is a powerful and versatile platform designed for Internet of Things (IoT) development. It is a cost-effective Wi-Fi microchip known for its capability to enable wireless communication in IoT applications. NodeMCU, on the other hand, is an open-source firmware and development kit that simplifies the process of prototyping and programming the ESP8266, built-in Wi-Fi connectivity, the NodeMCU ESP8266 allows devices to connect to the internet wirelessly, making it suitable for a wide range of IoT projects. One notable feature is its support for the Lua scripting language, providing a high-level programming environment for developers. Additionally, it is compatible with the Arduino IDE, allowing those familiar with Arduino to use the NodeMCU platform. Equipped with General Purpose Input/Output (GPIO) pins, the ESP8266 facilitates interfacing with various electronic components, making it ideal for applications such as home automation and sensor networks. It has garnered significant community support, resulting in an extensive collection of libraries and documentation, making it a popular choice for rapid IoT prototyping and development.

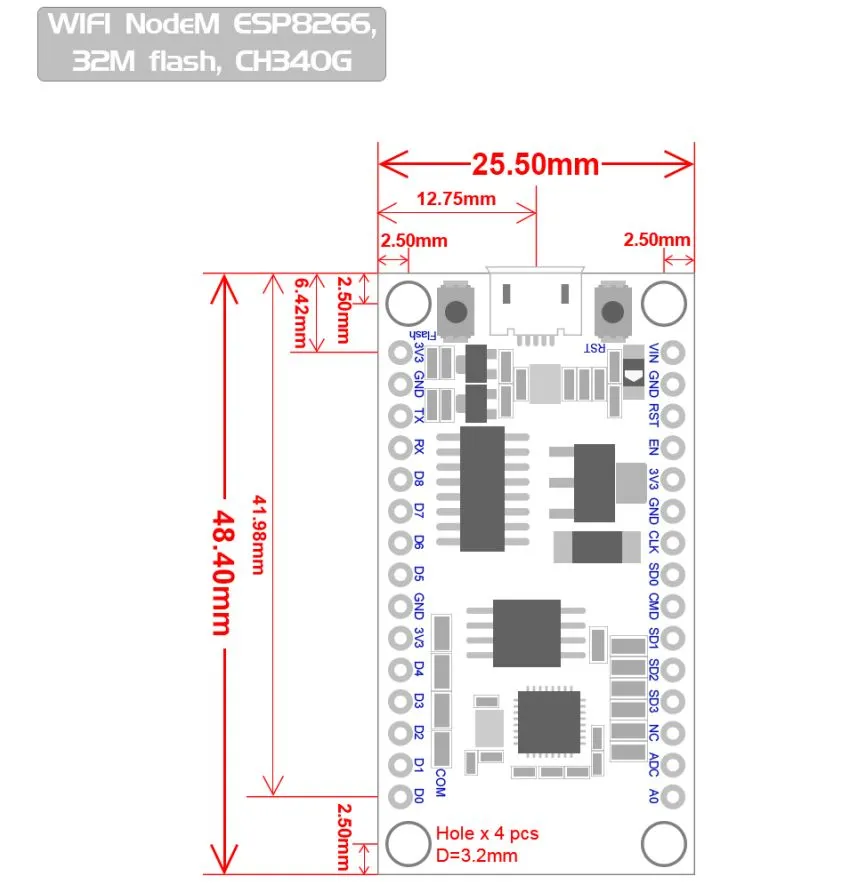


Figure 3.2 NodeMCU 2D View

**NodeMCU Specification:**

The NodeMCU development board is based on the ESP8266 microcontroller, and different versions of NodeMCU boards may have slight variations in specifications. As of my knowledge cutoff in January 2022, here are the general specifications for the NodeMCU ESP8266 development board:

**1. Microcontroller:** ESP8266 Wi-Fi microcontroller with 32-bit architecture.

**2. Processor:** Tensilica L106 32-bit microcontroller.

**3. Clock Frequency:** Typically operates at 80 MHz.

**4. Flash Memory:**

* Built-in Flash memory for program storage.
* Common configurations include 4MB or 16MB of Flash memory.

**5. RAM:** Typically equipped with 80 KB of RAM.

**6. Wireless Connectivity:**

* Integrated Wi-Fi (802.11 b/g/n) for wireless communication.
* Supports Station, SoftAP, and SoftAP + Station modes.

**7. GPIO Pins:** Multiple General Purpose Input/Output (GPIO) pins for interfacing with sensors, actuators, and other electronic components.

**8. Analog Pins:** Analog-to-digital converter (ADC) pins for reading analog sensor values.

**9. USB-to-Serial Converter:** Built-in USB-to-Serial converter for programming and debugging.

**10. Operating Voltage:** Typically operates at 3.3V (Note: It is crucial to connect external components accordingly to avoid damage).

**11. Programming Interface:** Programmable using the Arduino IDE, Lua scripting language, or other compatible frameworks.

**12. Voltage Regulator:** Onboard voltage regulator for stable operation.

**13. Reset Button:** Reset button for restarting the board.

**14. Dimensions:** Standard NodeMCU boards often have dimensions around 49mm x 24mm.

**15. Power Consumption:** Low power consumption, making it suitable for battery-operated applications.

**16. Community Support:** Active community support with extensive documentation and libraries.

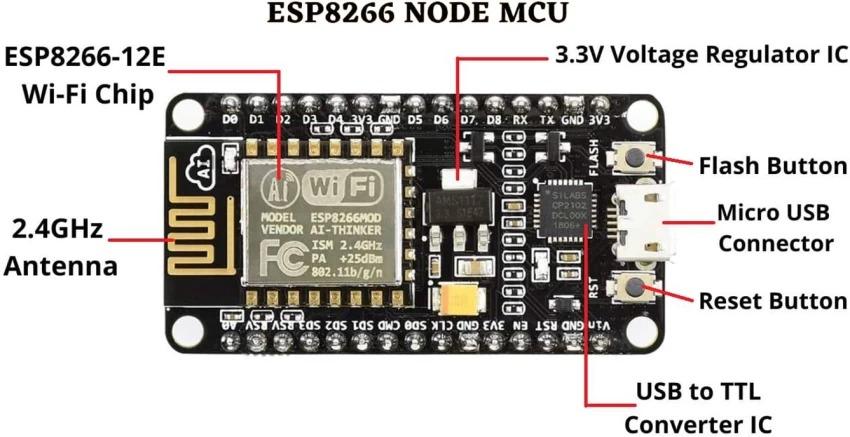


Figure 3.3: NodeMCU Parts

The NodeMCU ESP8266 development board typically has GPIO (General Purpose Input/Output) pins that can be used for various purposes, including interfacing with sensors, actuators, and other electronic components. Below is a common pinout configuration for the NodeMCU development board

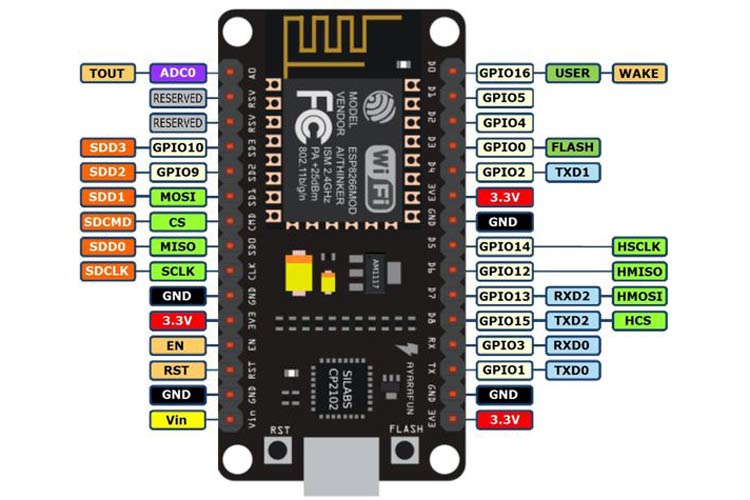


Figure 3.4: NodeMCU ESP8266 Pinout

ADC | A0 | GPIO16

EN | Enable | GPIO14

D0 | GPIO16 | GPIO12

D1 | GPIO5 | GPIO13

D2 | GPIO4 | GPIO15

D3 | GPIO0 | GPIO2

D4 | GPIO2 | GPIO9

D5 | GPIO14 | GPIO10

D6 | GPIO12 | GPIO3

D7 | GPIO13 | GPIO1

D8 | GPIO15 | TX (GPIO1)

D9 | GPIO3 (RX) | RX (GPIO3)

D10 | GPIO1 (TX) | D11 (MOSI)

D11 | MOSI | D12 (MISO)

D12 | MISO | D13 (SCK

**ADC**: Analog-to-Digital Converter pin for reading analog sensor values.

**EN** (Enable): Enable pin.

**D0-D8**: Digital GPIO pins.

**D9 (RX) and D10 (TX)**: Serial communication pins for programming and debugging.

**D11 (MOSI), D12 (MISO), D13 (SCK**): Pins used for SPI communication.

**D14 (SDA) and D15 (SCL)**: Pins used for I2C communication.

It's important to note that GPIO pins labeled as "D" (Digital) are typically used for general-purpose digital input/output. Additionally, GPIO pins labeled as "A" (Analog) can be used as analog inputs with the ADC. GPIO pins 6, 7, 8, 9, 10, and 11 have additional functions, so it's advised to refer to the specific NodeMCU documentation for detailed information on pin functionality and capabilities.

## **3.2 IR sensor:**

The sensor module light is adaptable to the environment, with a working voltage of 3.3 V to 5 V. It has a pair of infrared transmitting and receiving tubes when detecting direction meet with obstacles (reflecting surface), reflected infrared receiving by the tube.

After the comparator circuit processing, a green indicator will light up, at the same time the signal output interface output a digital signal (a low-level signal), can adjust the detection distance through the potentiometer knob, effective distance range 2 ~ 80 cm, easy to assemble and use. So it can be widely used in robot obstacle avoidance, obstacle avoidance car, line count, and so on many occasions. The IR sensor module operates at 5V and consumes around 150mA.

**Details of IR Sensor**

The IR sensor operates on the principle of detecting changes in the intensity of infrared radiation emitted or reflected by objects within its detection range. It typically consists of an IR emitter and a sensitive receiver. The IR emitter emits infrared radiation, and the receiver detects the reflected radiation. The distance between the sensor and the object is calculated based on the intensity of the reflected signal.

****

Figure 3.5: IR Sensor

**Technical Specifications of IR Sensor**

* Comparator chip: LM393
* Detection angle: 35 °
* Onboard detection indication
* The effective distance range of 2 cm to 80 cm
* A preset knob to fine-tune distance range
* There is an obstacle, the green indicator light on the circuit board
* TTL output is high whenever it senses an obstacle
* 3mm screw holes for easy mounting
* 3-5V DC power supply module can be used

**Connection with NodeMCU:**

The IR acts as a digital output so all you need to do is operate the pin to flip high (detected) or low (not detected).

Power is usually 3-5v DC input. Every IR module has a 3-pin connection.

The circuit connections are made as follows:

The Vcc pin of the IR module is connected to +3v of the NodeMCU.

Output pin of the IR module is connected to Digital pin as per the code of the NodeMCU.

The GND pin of the IR module is connected to the Ground pin (GND) of the NodeMCU.

## **3.3 Flow Sensor:**

Flow sensor measures liquid/water flow for your solar, computer cooling, or gardening project. This sensor sits in line with your water line, and uses a pinwheel sensor to measure how much liquid has moved through it. The pinwheel has a little magnet attached, and there's a hall effect magnetic sensor on the other side of the plastic tube that can measure how many spins the pinwheel has made through the plastic wall. This method allows the sensor to stay safe and dry.

****

Figure 3.7 Flow sensor

**Details of flow sensor**

Water flow sensor consists of a plastic valve body, a water rotor, and a hall-effect sensor. When water flows through the rotor, the rotor rolls. Its speed changes with different rates of flow. The hall-effect sensor outputs the corresponding pulse Signal. By counting the pulses from the output of the sensor, you can easily track fluid movement: each pulse is approximately 2.25 milliliters. Note this isn't a precision sensor, and the pulse rate does vary a bit depending on the flow rate, fluid pressure and sensor orientation. It will need careful calibration if better than 10% precision is required.

**Features of Temperature Sensors:**

1. **Flow Measurement Accuracy**: Flow sensors offer high accuracy in measuring fluid flow rates, ensuring precise monitoring and control of flow processes.
2. **Wide Flow Range:** They are designed to measure flow rates across a wide range, accommodating different fluid types and flow conditions.
3. **Compact Design:** Flow sensors are often compact and lightweight, making them suitable for integration into space-constrained systems and devices.
4. **Low Pressure Drop**: Many flow sensors have minimal pressure drop characteristics, ensuring that the fluid flow is not significantly impeded as it passes through the sensor.
5. **Compatibility with Various Fluids:** Flow sensors are compatible with a wide range of fluids, including liquids and gasses, making them versatile for different applications.
6. **Bi-Directional Flow Measurement:** Some flow sensors are capable of measuring flow rates in both directions, allowing for bidirectional flow monitoring.
7. **Digital Output:** Many modern flow sensors feature digital output signals, such as pulse or serial data, for easy interfacing with microcontrollers, PLCs, and other digital devices.
8. **Analog Output:** In addition to digital output, flow sensors may also provide analog output signals, such as voltage or current, for direct integration with analog control systems.
9. **Temperature and Pressure Compensation:** Advanced flow sensors incorporate temperature and pressure compensation mechanisms to ensure accurate flow measurement under varying environmental conditions.

**Connection:**

The VCC (Red Wire) and GND (Black Wire) pins of flow sensors are connected to 5V and GND pins of NodeMCU while the Signal pin (Yellow Wire) is connected to D4 pin of NodeMCU.

**Specifications:**

* Working Voltage: 5 to 24VDC
* Max current draw: 15mA @ 5V
* Working Flow Rate: 1 to 30 Liters/Minute
* Working Temperature range: -25 to 80°C
* Working Humidity Range: 35%-80% RH
* Maximum water pressure: 2.0 MPa
* Output duty cycle: 50% +-10%
* Output rise time: 0.04us
* Output fall time: 0.18us
* Liquid temperature <120 °C
* Flow rate pulse characteristics: Frequency (Hz) = 7.5 \* Flow rate (L/min)
* Pulses per Liter: 450
* Durability: minimum 300,000 cycles

**3.4 Buzzer**

It's a simple device that converts electrical signals into sound waves, producing a buzzing or beeping sound. Buzzers are widely used in various applications for providing audible notifications, alerts, alarms, and indications. Here's some information about buzzers.

****

Figure 3.8 Buzzer

**Working Principle:**

The working principle of a buzzer involves converting electrical energy into mechanical vibrations and then into sound waves. Here's a simplified explanation of how it works:

**Mechanical Activation**: In mechanical and magnetic buzzers, an electrical current causes a mechanical component (diaphragm or reed) to vibrate.

**Sound Production:** The vibrations of the mechanical component create pressure waves in the surrounding air, generating sound waves that we hear as a buzzing or beeping sound.

**Piezoelectric Activation**: In piezoelectric buzzers, an electrical signal is applied to a piezoelectric crystal. The crystal changes shape when subjected to the electric field, creating vibrations that produce sound waves.

Control and Sound Output: Buzzers can be controlled through voltage input, frequency modulation, or pulse width modulation (PWM). The pitch and volume of the sound generated can often be adjusted by varying the input parameters.

In summary, buzzers are versatile audio signaling devices used to provide audible alerts and notifications in various applications. They come in different types and configurations, allowing them to be tailored to specific needs and requirements

**Connection:**Buzzers are rated to 5V, and the GPIO only delivers 3.3V.

**Specifications:**Model Name/Number- AR083-5V-ACT-BUZZ

Size- 1 x 1 x 1 cm

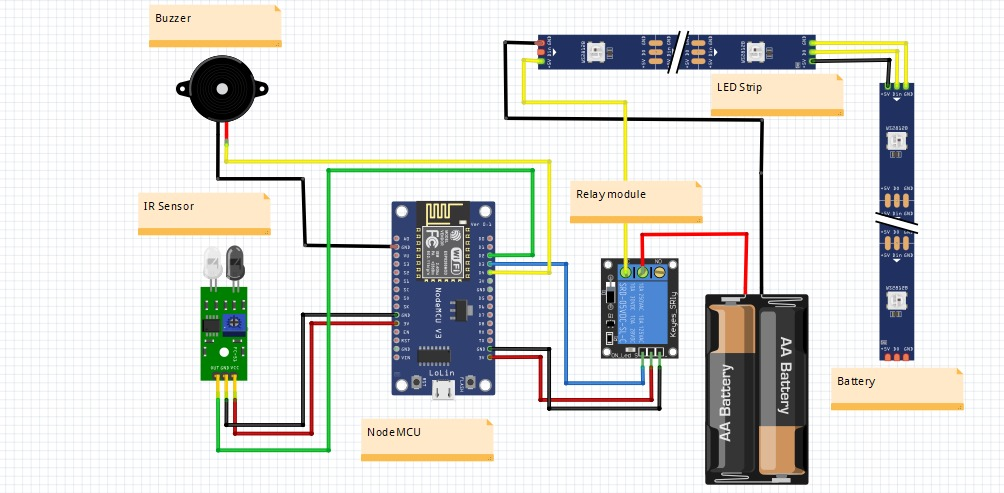
Voltage- 5 V

Power Source- DC

# **CHAPTER 4**

# **Design and Coding**

## **4.1 Circuit Diagram**



## **4.2 Code**

#include <ESP8266WiFi.h>

#include <Firebase\_ESP\_Client.h>

#include "addons/TokenHelper.h"

#include "addons/RTDBHelper.h"

int irSensorPin = D2; // Pin for the IR sensor

int floatSensorPin = D1; // Pin for the float sensor

int lightPin = D3; // Pin for the light

int buzzerPin = D4; // Pin for the buzzer

#define WIFI\_SSID "Kumar BOLLA"

#define WIFI\_PASSWORD "manasaaa"

#define API\_KEY "AIzaSyARjoYNzqtpw65r8k24OkH34fNtT2o1o9Y"

#define DATABASE\_URL "https://gnits-d5d15-default-rtdb.firebaseio.com/"

FirebaseData fbdo;

FirebaseAuth auth;

FirebaseConfig config;

unsigned long sendDataPrevMillis = 0;

bool signupOK = false;

void setup() {

Serial.begin(115200); // Start the serial monitor

pinMode(irSensorPin, INPUT); // Set the IR sensor pin as an input

pinMode(floatSensorPin, INPUT\_PULLUP); // Set the float sensor pin as an input

pinMode(lightPin, OUTPUT); // Set the light pin as an output

pinMode(buzzerPin, OUTPUT); // Set the buzzer pin as an output

WiFi.begin(WIFI\_SSID, WIFI\_PASSWORD);

Serial.print("Connecting to Wi-Fi");

while (WiFi.status() != WL\_CONNECTED){

Serial.print(".");

delay(300);

}

Serial.println();

Serial.print("Connected with IP: ");

Serial.println(WiFi.localIP());

Serial.println();

config.api\_key = API\_KEY;

config.database\_url = DATABASE\_URL;

if (Firebase.signUp(&config, &auth, "", "")){

Serial.println("ok");

signupOK = true;

}

else{

Serial.printf("%s\n", config.signer.signupError.message.c\_str());

}

config.token\_status\_callback = tokenStatusCallback; //see addons/TokenHelper.h

Firebase.begin(&config, &auth);

Firebase.reconnectWiFi(true);

}

void loop() {

int irValue = digitalRead(irSensorPin); // Read the IR sensor value

int floatValue = digitalRead(floatSensorPin); // Read the float sensor value

Serial.print("IR Value: ");

Serial.println(irValue);

Serial.print("Float Value: ");

Serial.println(floatValue);

if (irValue == HIGH) {

digitalWrite(lightPin, HIGH); // Turn on the light

digitalWrite(lightPin, LOW); // Turn off the light

}

if (floatValue == HIGH) {

digitalWrite(buzzerPin, HIGH); // Turn on the buzzer

} else {

digitalWrite(buzzerPin, LOW); // Turn off the buzzer

}

if (Firebase.ready() && signupOK && (millis() - sendDataPrevMillis > 1000 || sendDataPrevMillis == 0)){

sendDataPrevMillis = millis();

if (Firebase.RTDB.setInt(&fbdo, "mainbucket/manhole lid",irValue )){

Serial.println("PATH: " + fbdo.dataPath());

Serial.println("TYPE: " + fbdo.dataType());

}

else {

Serial.println("Failed REASON: " + fbdo.errorReason());

}

if (Firebase.RTDB.setInt(&fbdo, "mainbucket/water level",floatValue )){

Serial.println("PATH: " + fbdo.dataPath());

Serial.println("TYPE: " + fbdo.dataType());

}

else {

Serial.println("Failed REASON: " + fbdo.errorReason());

}

delay(1000); // Delay for stability

}

}

# **CHAPTER 5**

# **Results and Conclusion**

The smart manhole system that monitors real-time conditions and alerts the municipal office via an app developed using MIT App Inventor represents a significant advancement in urban infrastructure management. By providing timely information on potential hazards such as blockages, overflows, or damage, this system enhances public safety, reduces maintenance costs, and ensures a more efficient response from municipal services. The integration of IoT sensors and real-time alerts into a user-friendly app simplifies the monitoring process, contributing to smarter, safer, and more sustainable city operations.