SEWAGE LEVEL MAINTENANCE USING IOT



Made By: -

Uday Kiran Padhy 18B91A04L4

Shaik Syfuddeen 18B91A04J8

### ABSTRACT

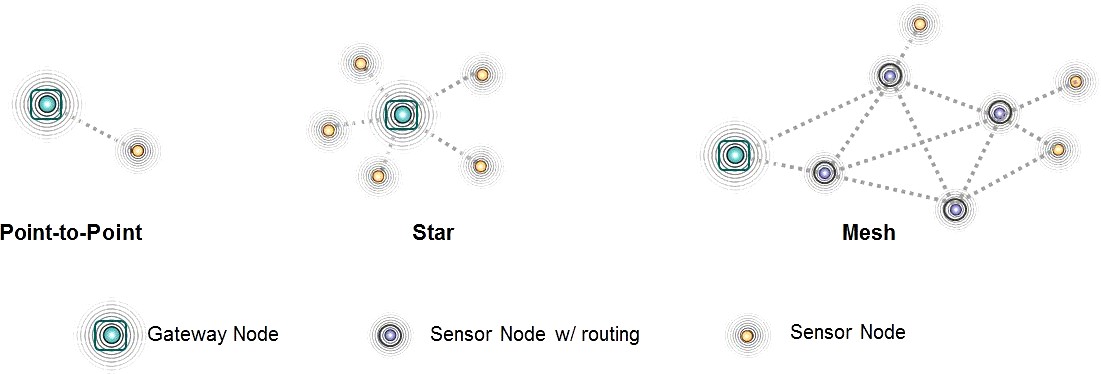
*Over flow of sewage on roads is been a major problem in many developed and under developed cities as well. Sewage is generally considered as waste water. The response to the complaints is not properly answered or taken into account. A precautionary system is developed where this issue of sewage overflow can be reduced by early sensing of increase in its level. The system design comprises of a sensor to sense the level, a controller to command, a communication network to register the complaints on blockage and continues increase in the level of sewage. A data base is to be maintained to record the data. The system rather simply monitoring the level, it generates alarm signals via complaints to the required departments through mail and SMS regarding prior to overflow.*

Components Required:

1. Arduino Board Uno (2no.)
2. NodeMCU Wifi Module (1no.)
3. NRF24 Transceiver Module (2no.)
4. Ultra-Sonic Sensor
5. Servo Motor
6. Solar Panels
7. Rechargeable Batteries

Working Mechanism:

In the method proposed, development of IoT based sewage level maintenance is designed, where the sewage level is monitored using an Ultra Sonic Sensor. The Ultra Sonic Sensor keeps on monitoring the level of the sewage. As the level changes, the Ultra Sonic sensor detects and send the signal to the Respective associated Node where the controller commands the IoT network to generate alert complaints to the municipal and corporation departments in prior to overflow. The complaints will be repeatedly registered until the level reaches the minimum set. The communication is done based on IoT where the complaints are auto registered and the entire system is controlled using an ARDUINO microcontroller.



## Communication Topology

The types of network topology that can be used are port to port topology, star, tree, mesh, ring bus and fully connected topology. The design caries port to port topology, where Switched point-to-point topologies are the basic model of conventional telephony.

14000

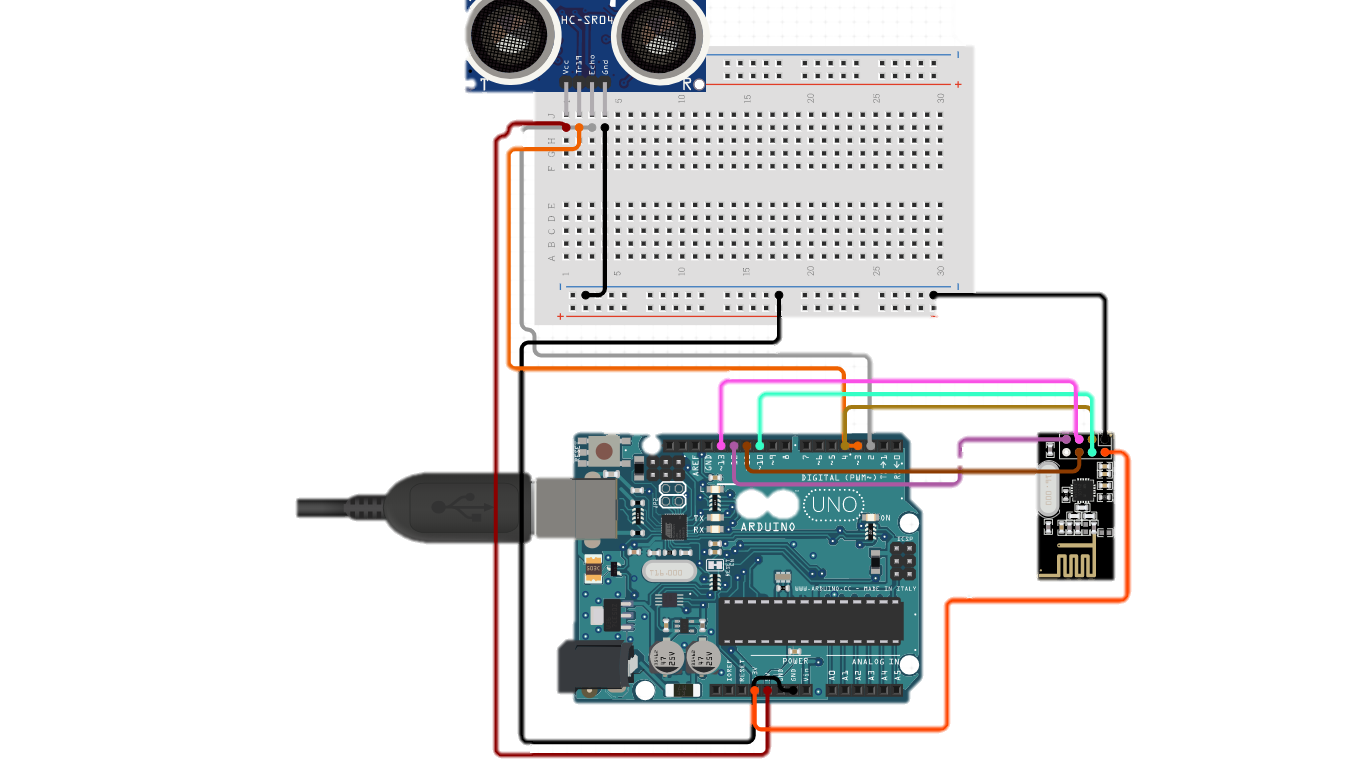


12000

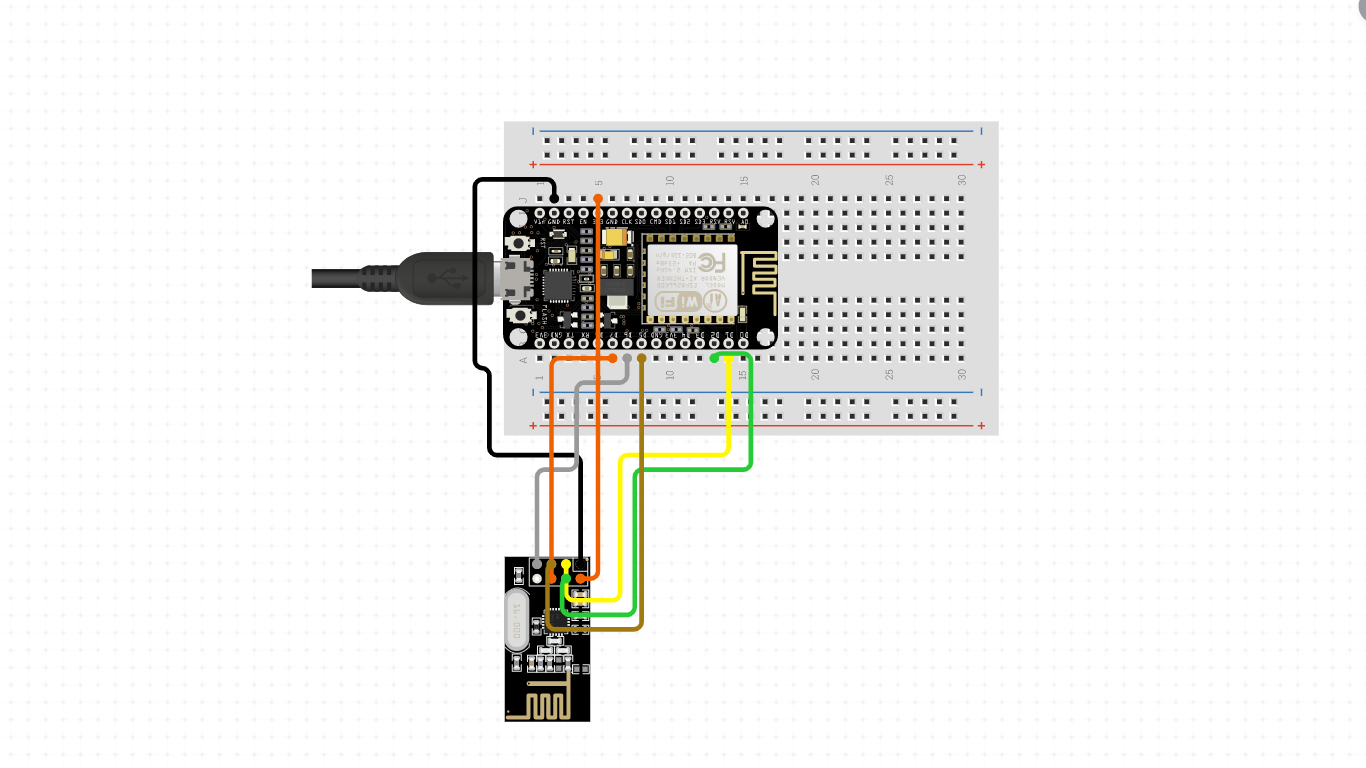
100

Circuit Diagram:

*Transmitter (Sensor Node):*



*Gateway Node:*

**

**Arduino Code:**

*Transmitter (Sensor Node):*

*#include <Arduino.h>*

*#include <SPI.h>*

*#include <RH\_NRF24.h>*

*#include <EEPROM.h>*

*RH\_NRF24 nrf24;*

*const int trigPin = 5;*

*const int echoPin = 6;*

*long duration;*

*int distance;*

*int deviceID = EEPROM.read(0);*

*void setup()*

*{*

*pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output*

*pinMode(echoPin, INPUT); // Sets the echoPin as an Input*

*Serial.begin(9600);*

*while (!Serial)*

*if (!nrf24.init()) {*

*Serial.println("init failed");*

*}*

*if (!nrf24.setChannel(3)) {*

*Serial.println("setChannel failed");*

*}*

*if (!nrf24.setRF(RH\_NRF24::DataRate2Mbps, RH\_NRF24::TransmitPower0dBm)) {*

*Serial.println("setRF failed");*

*}*

*Serial.println("Transmitter started");*

*}*

*void loop() {*

*Serial.println("Sending to gateway");*

*uint8\_t data[3];*

*data[0] = distance;*

*data[1] = 101;*

*digitalWrite(trigPin, LOW);*

*delay(1000);*

*digitalWrite(trigPin, HIGH);*

*delay(2000);*

*digitalWrite(trigPin, LOW);*

*duration = pulseIn(echoPin, HIGH);*

*distance= duration\*0.034/2;*

*Serial.print("Distance: ");*

*Serial.println(distance);*

*nrf24.send(data, sizeof(data));*

*nrf24.waitPacketSent();*

*uint8\_t buf[RH\_NRF24\_MAX\_MESSAGE\_LEN];*

*uint8\_t len = sizeof(buf);*

*if (nrf24.waitAvailableTimeout(1000))*

*{*

*if (nrf24.recv(buf, &len))*

*{*

*Serial.print("got reply: ");*

*Serial.println((char\*)buf);*

*}*

*else*

*{*

*Serial.println("recv failed");*

*}*

*}*

*else*

*{*

*Serial.println("No reply.");*

*}*

*delay(3000);*

*}*

*Gataway Node:*

*#include <SPI.h>*

*#include <RH\_NRF24.h>*

*#include <Wire.h>*

*#include <EEPROM.h>*

*#include <ESP8266WiFi.h>*

*#include <WiFiClient.h>*

*#include <ESP8266WebServer.h>*

*#include <ESP8266HTTPClient.h>*

*String apiKey = "XXXXXXXXXXXXXXX";*

*const char\* ssid = "Your-Wifi User Name";*

*const char\* password = "Your Wifi Password";*

*WiFiClient client;*

*HTTPClient http;*

*int gatewayID = EEPROM.read(0);*

*const char\* server = "api.thingspeak.com";*

*RH\_NRF24 nrf24(2, 4);*

*void setup()*

*{*

*Serial.begin(9600);*

*Serial.print("Receiver Started, ID: ");*

*Serial.print("Connecting to ");*

*Serial.println(ssid);*

*Serial.print(gatewayID);*

*Serial.println();*

*WiFi.begin(ssid, password);*

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

*delay(500);*

*Serial.print(".");*

*}*

*Serial.println("");*

*Serial.println("WiFi connected");*

*nrf24.init();*

*nrf24.setChannel(3);*

*nrf24.setRF(RH\_NRF24::DataRate2Mbps, RH\_NRF24::TransmitPower0dBm);*

*}*

*void loop()*

*{*

*if (nrf24.available())*

*{*

*uint8\_t buf[RH\_NRF24\_MAX\_MESSAGE\_LEN];*

*uint8\_t len = sizeof(buf);*

*if (nrf24.recv(buf, &len))*

*{*

*uint8\_t sdata[] = "Data Received.";*

*nrf24.send(sdata, sizeof(sdata));*

*nrf24.waitPacketSent();*

*int Level = buf[0];*

*int deviceID = buf[1];*

*Serial.println("--- Data retrieved from device ---");*

*if (client.connect(server, 80)) { // "184.106.153.149" or api.thingspeak.com*

*String postStr = apiKey;*

*postStr += "&field1=";*

*postStr += String(Level);*

*postStr += "\r\n\r\n";*

*client.print("POST /update HTTP/1.1\n");*

*client.print("Host: api.thingspeak.com\n");*

*client.print("Connection: close\n");*

*client.print("X-THINGSPEAKAPIKEY: " + apiKey + "\n");*

*client.print("Content-Type: application/x-www-form-urlencoded\n");*

*client.print("Content-Length: ");*

*client.print(postStr.length());*

*client.print("\n\n");*

*client.print(postStr);*

*Serial.print("Water Level:");*

*Serial.print(Level);*

*Serial.print(", Device ID: ");*

*Serial.println(deviceID);*

*Serial.println("---- Data sent to Thingspeak ----");*

*Serial.println();*

*Serial.println("...................................................");*

*Serial.println();*

*}*

*client.stop();*

*}*

*}*

*delay(15000);*

*}*

**Advantages:**

1. Reduces Human Intervention
2. Real Time Analysis Of sewage level
3. Making Possible to check the present Condition Of the sewage in any remote Location with access.
4. Due to the use of Solar Strips, it doesn’t require any other external power source as far as possible.
5. The Surveillance of Sewage and Drainage Levels has become a lot easier, with of the use of Radio Communication.
6. Because of the robotic arm the clearing of sewage is ensured at minimum level.

**Disadvantages:**

1. The System is still depended on internet connections.
2. It doesn’t locate the point where the blockade has occurred.
3. Due to the use of ultra-sonic sensor the reading might get changed because of Any large waste material.

**Conclusion:**

A precautionary system avoids or eliminates the issue of sewage overflow on roads which is being a major problem in many cities. The level sensors are efficiently used and system is designed in a social relevant idea thus to create an impact on hygiene and cleanliness by simply avoiding the problem of overflow on streets and also to ensure compulsory cleaning of blockage which causes the increase the sewage level by registering repeated complaints to random departments unless action is taken.