HINDUSTAN
INSTITUTE OF TECHNOLOGY & SCIENCE
(DEEMED TO BE UNIVERSITY)
CHENNAI

A PROJECT REPORT

Smart Water Leakage Detection System

Submitted by

P. Sai Manoj - 19113101

K. Sumanth Kumar Reddy - 19113105

P. Umesh Chandra - 19113118

Under the supervision of

Dr. K. Ramesh, Professor, Computer Science Department

In partial fulfilment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

In

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING SCIENCES

HINDUSTAN INSTITUTE OF TECHNOLOGY AND SCIENCE

OCTOBER 2021

HINDUSTAN
INSTITUTE OF TECHNOLOGY & SCIENCE
(DEEMED TO BE UNIVERSITY)
CHENNAI

BONAFIDE CERTIFICATE

Certified that this project report **“SMART WATER LEAKAGE DETECTION SYSTEM”** the bonafide work of **“P. SAI MANOJ - 19113101, K. SUMANTH KUMAR REDDY-19113105, P. UMESH CHANDRA – 19113118.**” who carried out the project work undermy supervision.

Dr. K.Ramesh
SUPERVISIOR
PROFESSOR
DEPT OF CSE
HITS

ACKNOWLEDGEMENT

First and foremost we would like to thank **ALMIGHTY** who has provided us the strength to do justice to our work and contribute our best to it.

We wish to express our deep sense of gratitude from the bottom of our heart to our guide **Dr.K Ramesh,Professor, Computer Science and Engineering**, for his motivating discussions, overwhelming suggestions, ingenious encouragement, invaluable supervision, and exemplary guidance throughout this project work.

We have taken efforts in this project. However, it would not have been possible without the kind support of **Dr.ANGELINA GEETHA Professor CSE&DEAN(E&T)**for her guidance and constant supervision as well as for providing necessary information regarding the project.

We would like to extend our heartfelt gratitude to **Dr. THANGAKUMAR J, Associate Professor and Head, Department of Computer Science and Engineering** for his valuable suggestions and support in successfully completing the project.

We thank the management of **HINDUSTAN INSTITUTE OF TECHNOLOGY AND SCIENCE** for providing us the necessary facilities and support required for successfully.

ABSTRACT

Water supply management has become a serious issue due to rapid urbanisation as water is one of the most important natural resource required for our survival. The underground pipelines which carry water to our household are somehow damaged sometimes mainly due to ageing or sometimes due to the pressure above land. These are hard to detect and may lead to huge loss of water.

The proposed system detects if there is a leak in pipe and shows to the management through a web server. This helps to save a lot of water and make it easier for the maintenance people to easily maintain pipes.

The system uses an Arduino microprocessor to process data and the output is shown through a Thingspeak.com web server. The data is transferred from the Arduino to the web server using a Wireless Fidelity (Wi-Fi) module (which is also known as Node Microcontroller Unit (MCU)) through the internet.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER -1

1. INTRODUCTION

This project comes under Internet of things (IOT) genre. Internet of things describe the system of networks or hardware which are connected to various sensors and use internet as the communication medium to exchange data between systems, servers or devices over the internet. It also uses the services like cloud to store data and then access it using internet.

In the recent times urbanisation has become rapid over many countries and especially India has been much faster in this process. Due to the rapid urbanisation usage of natural resources like water, soil etc has been immense. Mainly water usage is much and required water is transported to household or industries through underground pipelines. These pipelines are mainly buried underground and are difficult to monitor.

In this scenario, water supply management has become a serious issue due to rapid urbanisation as water is one of the most important natural resource required for our survival. The underground pipelines which carry water to our household are somehow damaged sometimes mainly due to ageing or sometimes due to the pressure above land. These are hard to detect and may lead to huge loss of water. Maintenance teams find it difficult to detect if there is any leak in the pipelines. And this situation leads to a huge loss of water. And this in turn leads to water scarcity. There are many proposed systems to detect water leakage but either they are too costly or they lack efficiency. Due to leakage in agricultural supply pipes, huge amount of water may be wasted if not identified at the earliest and this may also lead to loss of yield.

Some leaks may be small and some leaks may be huge. In the case of huge leaks, if the water stops flowing then the surrounding dust and mud enters into the pipe which in turn leads to the transport of dusty and mud water to the households and consumption of these dusty water may lead to illness. If identified in households, this water is thrown useless and are hence wasted. If these leaks are not identified and hence ignored, this may lead to the permanent damage of pipes due to crumbling of mud and small rocks.

The aim of this project is basically works on the principle of pressure variation along the pipeline. This system detects the leaks pretty faster and with approximately more than 90% accuracy. If the maintenance teams use this model, they can even identify the leaks very faster thus results in saving of huge amount of water which is being wasted otherwise.

OBJECTIVE

To save the huge amount of water which is being wasted because of the unidentified leaks in the underground pipeline system. And to detect the leaks much faster and all the above with cost efficiency. And thus, to save the tremendous loss of the most required natural resource.

The main objective of this project is to stop the tremendous loss of water by these leakages and to reduce the man power required to detect for any leaks in the pipelines.

APPLICATIONS

- It can be used to detect leaks in the underground pipelines which carry water for our daily usage.
- It can also be used in the agricultural pipelining to prevent the wastage of water that is meant for crops.
- It can be used to prevent the contamination of water through the leaks in the pipes.
- It can also be used in the industries which requires huge pipelining system to carry liquid essentials.
- It can be used in the normal household where there is an overhead tank which supplies water into rooms using pipelining system. This can detect leaks in the pipes which are also mounted into walls.

CHAPTER-2

LITERATURE REVIEW

INTRODUCTION

Fresh water is one of the most important natural resources required for the human survival. Due to rapid urbanisation, requirement of water has been increased rapidly. To meet the huge fresh water requirement of humans, water is supplied through the underground pipeline system. And the pipelines used are somehow damaged and the leaks are unidentified and water is lost in a huge amount. To save the loss of water some devices are proposed which involves different methodologies and algorithms.

LITERATURE ANALYSIS

This project has been implemented by collecting and analyzing from various resources some of resources are research papers that havebeen published in various renowned journals and websites.

1. Due to ageing or pressure above ground, pipes may crack or may have a leak. These leaks may be small or big and are unidentified because they are under the ground. Because of this situation, huge amount of water may be wasted until the leaks are identified. They used the concept of pressure variation along the pipe. A robot is sent into the pipe which contains two microcontrollers, one for communicating with the device above the ground which controls the movement of robot and other to process the data and check whether there is a leak in the pipeline. The robot uses a High Watt (HW) battery for power supply and runs for a maximum time of 20 minutes per battery. This robot consumes more battery because of the motors used in the robot to travel and the components used also need power to continuously detect the pressure.

2. Another Reference uses an entirely different approach which requires a huge data analysis. They record the sound signals that are produced by the water fixtures and different appliances used in the household. Every sound signal is recorded and stored in a separate database. Then the data is processed and now, it is checked for any disruption of signals which represents the leak situation. In the recording step the recordings are sent remotely to the dataset for various purposes. A software tool for sound signal identification is used to analyse and process the data. The analysis is represented in graph format. This requires a very high-level software involvement which requires higher level dataset training and representation.

3. This method uses a different setup which involves the use of a self-built Monitoring Application system which involves server and a real- time monitoring client which is used to send and receive data from the server. Arduino Ethernet shield and Arduino board included with Node MCU is used combinedly which takes the data from the Global Positioning System (GPS) and Flow rate meter to determine the position of the leak. The range of this device is approximately 2 metres and the data from the Arduino ethernet shield (taken from the Flow Rate meter) is then transferred to the Arduino (with built in Node MCU) which transfer the data to the Monitoring Application system from which the data is requested and monitored by a real time monitoring client.

CHAPTER-3

PROPOSED SYSTEM:

- This project is made based on some simple concepts like, the pressure inside the flowing water pipe remains constant throughout the pipe, unless there is a leak or loss of water by any other means. That means, if there is a leak in the pipe, there will be certain amount of pressure drop between both ends if the pipe. We check pressures at both ends of the pipe to check whether there is a leak or not in a pipe. Another concept used is, data transfer from processor to the web server using Internet.

- This project consists of two flow sensors which are connected to each end of the pipe which continuously monitor the flowrate/pressure values at both the ends and if there is a drop of certain amount of pressure at the other end, then it shows that there is a leak in the pipe. It can also be used for household pipeline systems easily.

- The Flow sensor data is sent to thing speak webserver. Which we can see the water flowrate of two sensors from far distances.

MODEL DIAGRAM:

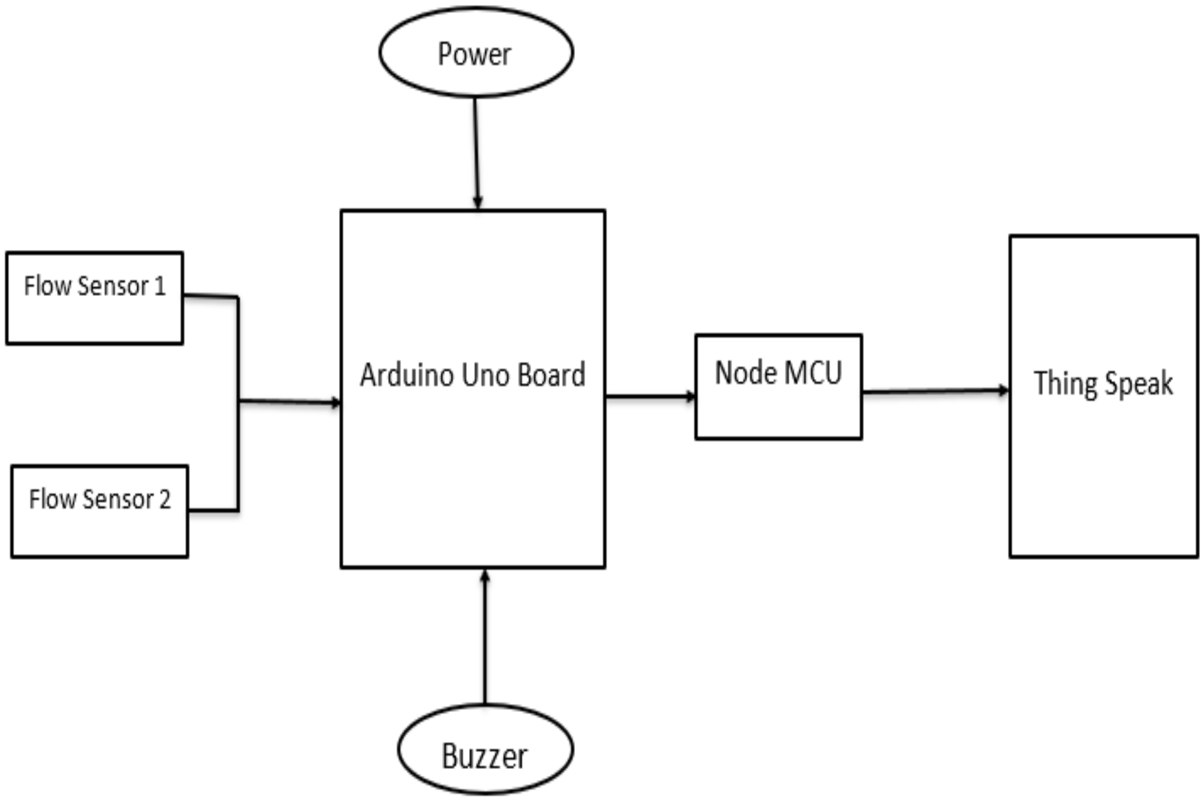


Fig 3.1 module diagram

ARDUINO:

Fig 3.2 Arduino uno

The above figure shows Arduino board which is used to collect data from flow sensors and then process the data and determine whether there is a leak or not. And then sends this data to Node MCU

NODE MCU:

Fig 3.3 NODE MCU

Above figure shows Node MCU is used to connect the device to the internet and transfer the data from Arduino to Web page whether there is a leak or not

BREAD BOARD & JUMPER WIRES:

Fig 3.4 Bread Board & Jumperwires

These are used for the connection between Hardware components. Thisis used for easier connection of Hardware components.

Water flow sensors:

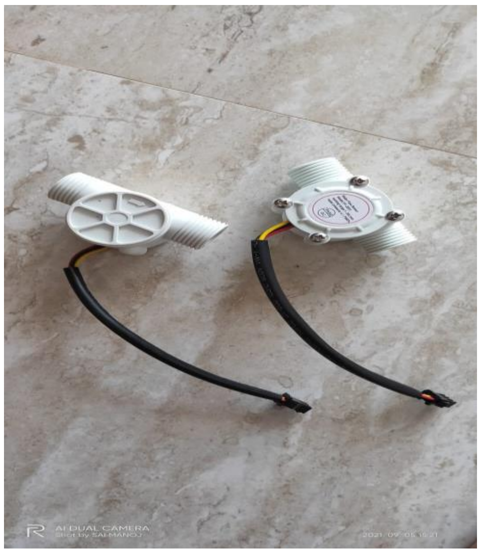


Fig 3.5 Water flow sensors

The above figure shows the flow sensors used in the device to detect the flowrate at the ends of the pipe.

LED & BUZZER:

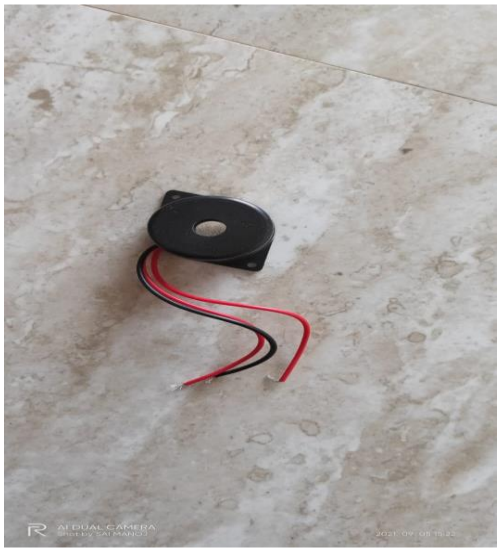


Fig 3.6 Buzzer

The above figure is a buzzer or beeper is an audio signaling device, which acts as actuator in this project.

CHAPTER – 4

EXPERIMENTAL SETUP:

CONNECTING SYSTEM

The main part of the data processing in the device is done by the Arduino microcontroller. Two flow sensors are connected to the either sides of the water pipe. In the middle of the pipe a leak is created .The first flow sensor is connected to the Arduino using jumper wires and bread board. The power pin of the flow sensor is connected in parallel to the 5v power pin of the Arduino board and the ground pins of flow sensor and Arduino are connected together. Now, the signal pin of the flow sensor is connected to any of the digital input pins of the Arduino and in between the wires, a 1K ohm resistor is connected using bread board.

Similarly, the power pin of the flow sensor is connected in parallel to the 5v power pin of the Arduino board and the ground pins of flow sensor and Arduino are connected together. Now, the signal pin of the flow sensor is connected to any of the digital input pins of the Arduino and in between the wires, a 10K ohm resistor is connected using bread board. Now, both the flow sensors are connected to the Arduino and the sample pipe used for leak detection.

This makes a basic connection wherein, now the system can be able to detect the leak but there is no means for it to show the output. And for that Nodemcu is connected which takes the data from the Arduino and sends it to a cloud platform to store and show the output whether there is a leak or not.

Now, for connecting the Nodemcu, the RX and TX pins of the Nodemcu are connected to the Arduino digital pins and the ground pin is connected in the bread board which is then connected in parallel to the ground pin of the Arduino board and the power is left empty as we used a separate power supply for the Nodemcu.

Now, for connecting the buzzer, the 5v wire and ground wire is connected to buzzer. The buzzer pin is connected to digital pin on Arduino board which activate the buzzer when leakage is detected and in between 10ohm resistor acts as power resistence.

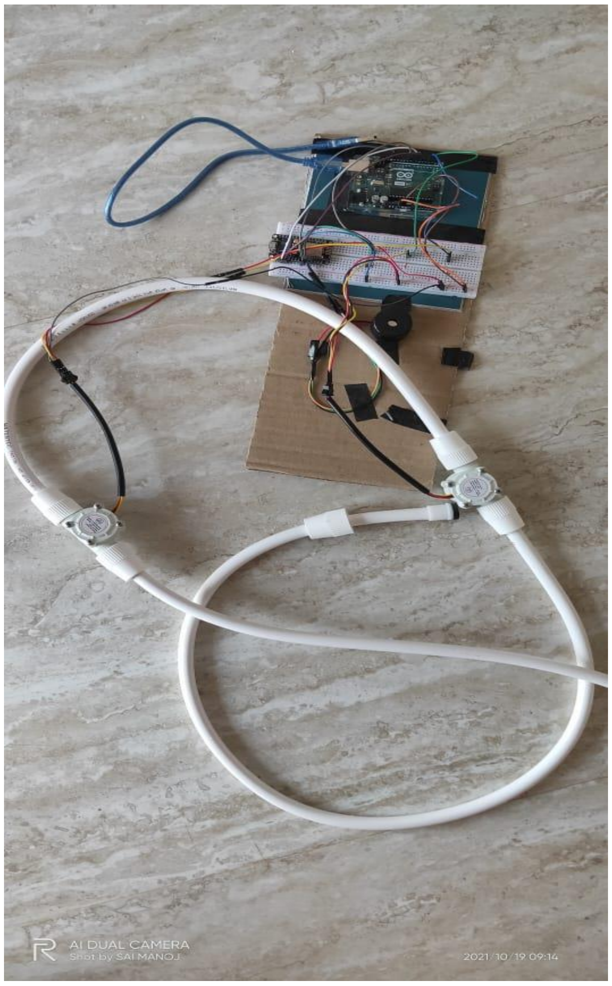


Figure 4.1 Basic connection

The above picture shows the rough circuit of basic connections which are required in the device. All the connections used are parallel connections as there is only one power supply pin in the Arduino. This circuit made can detect the leak if connected to the pipe and water supply and can be able to process the data and send the output to a cloud server (Thingspeak IOT is used here) and if leak is detected the the ardunio will activate the buzzer and the buzzer will give long beep sound.after completion of all the coding required whose process is discussed further and .

DATA PROCESSING AND TRANSFER

The two flow sensors which are connected on the either sides of the pipe which counts the pulses (when the water enters the one end of the flow sensor, there will be a small fan which rotates inside the flow sensor due to the water flow and this reading is counted as pulse) which are in turn modified into flowrate of water using mathematical calculations. And the same goes on with the other flow sensor. Thus, the values of both the flow sensors are recorded. Normally, when there is a normal flow in the pipe, the pressure is maintained equal throughout the pipe under normal conditions. But, when there is a leak in pipe between both the flow sensors, the pressure values vary as the water is lost in between. By using this concept coding is done.

The readings of the flow sensors are recorded in variables like "FlowRate1" and "FlowRate2". The readings of the flow sensors are first counted as pulses which are then modified into litres per minute or millilitres per minute based on the requirement using mathematical calculations. And a new variable is created for recording the difference between the readings of both the flow sensors as "Flow Difference". Now, by testing several times a mean difference value is created which is considered as a border value for the maximum difference to be allowed.

Now, the if condition arrives wherein, if there is a leak in the pipeline, then the difference in readings of the flow sensors will move above the mean difference value and then the Arduino determines that there is a leak in the pipeline. And if the difference in flowrates is below the mean difference, then the Arduino determines that there is no leak in the pipeline if the difference is greater then there is a leak and the buzzer gives beep sound. This data whether there is a leak or not, is to be shown as output.

Now, the data which is processed by the Arduino should now be transferred to the Nodemcu. For connecting the Nodemcu, the RX and TX pins of the Nodemcu are connected to the Arduino digital pins and the ground pin is connected in the bread board which is then connected in parallel to the ground pin of the Arduino board and the power is left empty as we used a separate power supply for the Nodemcu. The data which is processed by the Arduino is now transferred into the Nodemcu using the serial communication method. In serial communication method, the data from Arduino is sent to Nodemcu using both the serial monitors in the Arduino Terminal. The RX and TX pins of the Nodemcu are declared as input pins and the Arduino pins are declared as output pins to send data to the Nodemcu.

The flow of data from Arduino to the Nodemcu using the serial communication method. Here, both the hardware serial and software serial are used and in the Arduino code software serial is declared by a local name for RX and TX pins and now the hardware serial and software serial are initiated with serial begin command and 9600 standard delay value. In the Nodemcu code, while the serial is receiving data it is reflected as true and if there is no data received, it returns false. So, if there is incoming data, the serial monitors with respect to the port connected with Nodemcu to the device is now activated with 9600 monitor and the incoming data can be viewed on the serial monitor of Nodemcu. Now, the data in the Nodemcu must be transferred to a web server/ cloud service by connecting the Nodemcu to the internet.

DATA TO WEB SERVER

After the transfer of data from Arduino to Nodemcu, the Nodemcu should be connected to the active internet connection. For that, in the Nodemcu code two strings are to be initiated and the name of the Wi-fi and password and then wifi.begin() keyword is used with username and password as arguments in it. For the web server and cloud services a platform called Thingspeak.com is used where in that you can create a database with desired number of fields. And the data sent to the server is displayed in the graph format.

In the Thingspeak website, when you create a personal database with required fields, after the creation API keys will be created for reading the data as well as writing the data. The API key for writing data is used in the code after mentioning the host link in the same code. Then, we should specify the port connected and the names of all the fields used in the Thingspeak website. Then the command for writing the data in the server is given in the Nodemcu code. Now the if condition comes into action in which if the data is received properly, it displays the data. In the other case it shows failed to connect. Now, the data from sensors to Arduino then to Nodemcu now reaches the final stage/output stage which is displayed in the web server and also stored in cloud service of Thingspeak.

Now, we can check the data which is received in the Thingspeak web page by opening the sensor page and then opening the visualise page and then going to show graph. This displays the data received from the Nodemcu in a graphical format. These values can be crosschecked by opening the serial monitor and checking the values printed in the Arduino terminal. The figure for the data flow from Nodemcu to web server is shown below.

CHAPTER – 5

RESULTS AND DISCUSSION:

The working of two flow sensor with Arduino is checked and the flow sensor readings are shown successfully in the serial monitor in litres per minute units. And now the two flow sensors integration with the Arduino is checked and the data from both the flow sensors is displayed successfully.

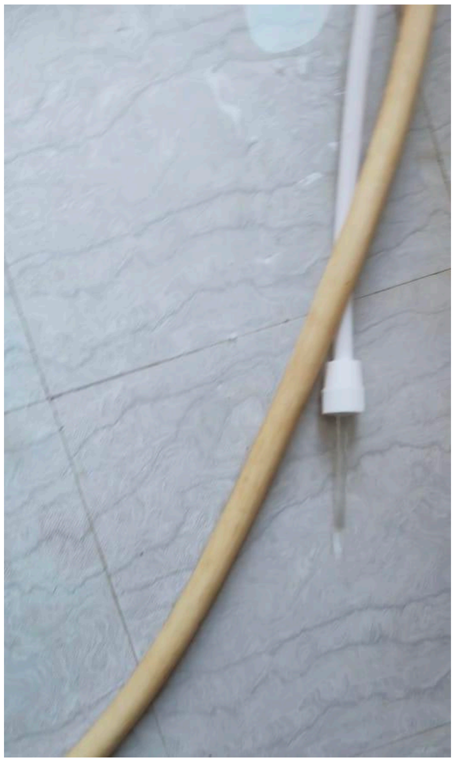


Fig 5.1 Water Flow

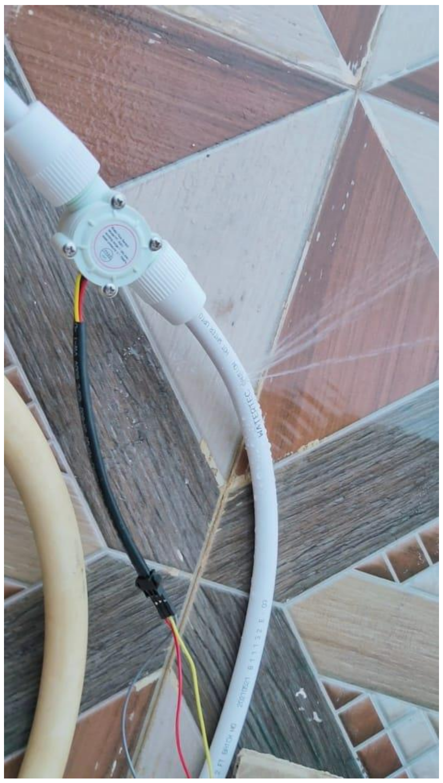
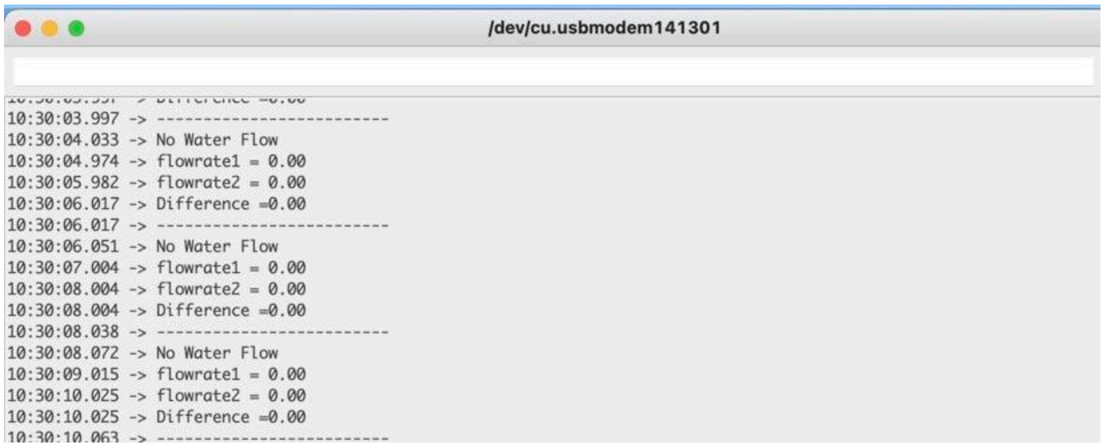
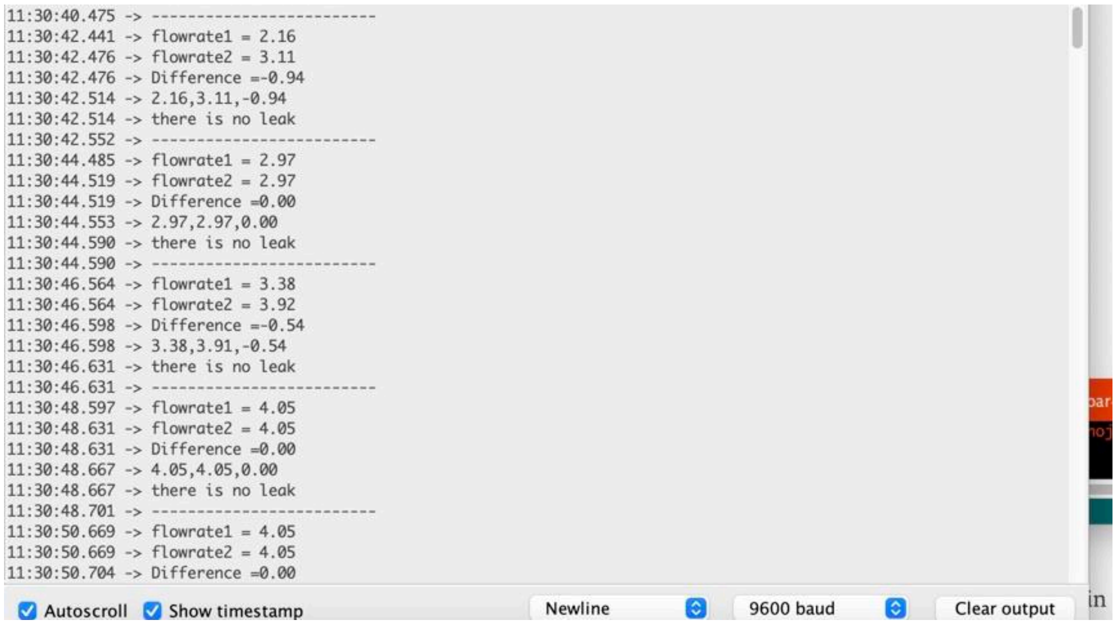


Fig 5.2 Water Leakage


/dev/cu.usbmodem141301

```
10:30:03.997 -> ------------------------
10:30:04.033 -> No Water Flow
10:30:04.974 -> flowrate1 = 0.00
10:30:05.982 -> flowrate2 = 0.00
10:30:06.017 -> Difference =0.00
10:30:06.017 -> ------------------------
10:30:06.051 -> No Water Flow
10:30:07.004 -> flowrate1 = 0.00
10:30:08.004 -> flowrate2 = 0.00
10:30:08.004 -> Difference =0.00
10:30:08.038 -> ------------------------
10:30:08.072 -> No Water Flow
10:30:09.015 -> flowrate1 = 0.00
10:30:10.025 -> flowrate2 = 0.00
10:30:10.025 -> Difference =0.00
10:30:10.063 -> ------------------------
```

Fig 5.3 No water flow



```
11:30:40.475 -> ------------------------
11:30:42.441 -> flowrate1 = 2.16
11:30:42.476 -> flowrate2 = 3.11
11:30:42.476 -> Difference =-0.94
11:30:42.514 -> 2.16,3.11,-0.94
11:30:42.514 -> there is no leak
11:30:42.552 -> ------------------------
11:30:44.485 -> flowrate1 = 2.97
11:30:44.519 -> flowrate2 = 2.97
11:30:44.519 -> Difference =0.00
11:30:44.553 -> 2.97,2.97,0.00
11:30:44.590 -> there is no leak
11:30:44.590 -> ------------------------
11:30:46.564 -> flowrate1 = 3.38
11:30:46.564 -> flowrate2 = 3.92
11:30:46.598 -> Difference =-0.54
11:30:46.598 -> 3.38,3.91,-0.54
11:30:46.631 -> there is no leak
11:30:46.631 -> ------------------------
11:30:48.597 -> flowrate1 = 4.05
11:30:48.631 -> flowrate2 = 4.05
11:30:48.631 -> Difference =0.00
11:30:48.667 -> 4.05,4.05,0.00
11:30:48.667 -> there is no leak
11:30:48.701 -> ------------------------
11:30:50.669 -> flowrate1 = 4.05
11:30:50.669 -> flowrate2 = 4.05
11:30:50.704 -> Difference =0.00
```

Fig 5.4 flowrates between sensors

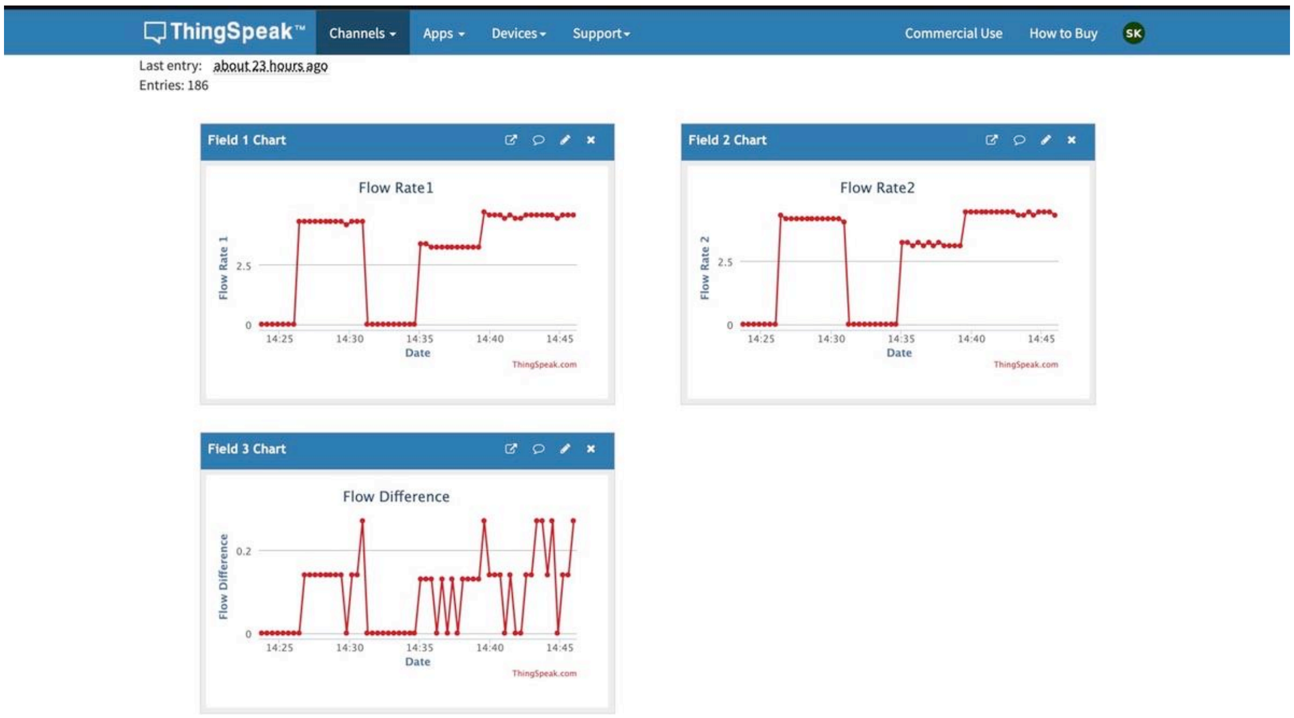


Fig 5.5 Values are updating into Thingspeak

In the above figure the output screen in the Thingspeak web server. Two values are read in the server and shown in graph format. In the three fields, one is for the readings from the flow sensor 1 and other field is for the readings from the flow sensor 2 and the other is for the difference of both the readings. Thus, the data is received in the web server and is stored.

CHAPTER 6
CONCLUSION & FUTURE SCOPE

CONCLUSION:

Using of this device can save huge amount of water which is otherwise wasted. This device makes the identification of leaks much easier for the maintenance people. This also save a lot of manpower and saves a huge time which is otherwise wasted in finding leaks. This device also makes identification of leaks much easier, faster and cost efficient.

FUTURE WORK:

Based on this project, an android application can be made which makes notifying of leaks easier and better and also recent leaks can be viewed in a lot easier way in the recent tab of android application.

REFERENCES

Reference papers:

1) Dvajasvie.G, Farisha Banu PK, Sachin N Babu, Saheen K P, Nikhil Binoy C, "Leak Detection in Water-Distribution Pipe System", Proceedings of the Second International Conference on Intelligent Computing and Control Systems (ICICCS 2018) IEEE Xplore Compliant Part Number: CFP18K74-ART; ISBN:978-1-5386-2842-3, PAGE NO-1151

2) Solomon Seyouma, Leonardo Alfonsoa, Schalk Jan van Andela, Wouter Kooleb, Ad Groenewegenb, Nick van de Giesenb, "A Shazam-like household water leakage detection method", Proceedings of the XVIII International Conference on Water Distribution Systems Analysis, WDSA2016, Procedia Engineering 186 (2017) 452 – 459

3) R F Rahmat, I S Satria, B Siregar, R Budiarto, "Water Pipeline Monitoring and Leak Detection using Flow Liquid Meter Sensor", Proceedings of the International Conference on Recent Trends in Physics 2016 (ICRTP2016) IOP Publishing Journal of Physics: Conference Series 755 (2016) 011001 doi:10.1088/1742-6596/755/1/011001

4) https://www.hawaiiplumbingservices.com/underground-leaks.html
5) https://bc-robotics.com/tutorials/using-a-flow-sensor-with-arduino/

6) https://www.google.com/url?sa=i&url=https%3A%2F%2Fcreate.arduino.cc%2Fprojecthub%2Fpawan-kumar3%2Fserial-communication-between- nodemcu-and-arduino-640819&psig=AOvVaw1aCsRqXe-0K7OTZFB8C3Gt&ust=1609414296720000&source=images&cd=vfe&ved=0CAIQjRxqFwoTCKDZl9PN9e0CFQAAAAAdAAAAABAD

}

SOURCE CODE:

Arduino code:

```
#include <SoftwareSerial.h>
int flowPin1 = 2;    //This is the input pin on the Arduino
int flowPin2 = 3;    //This is the input pin on the Arduino
const int buzzer= 9;//buzzer to arduino pin 9
SoftwareSerial mySerial(6, 7);
double flowRate1,flowRate2,difference;
String values;
//This is the value we intend to calculate.
volatile int count1,count2;//This integer needs to be set as volatile to ensure it updates
correctly during the interrupt process.

void setup() {
  pinMode(buzzer,OUTPUT);
  // put your setup code here, to run once:
  pinMode(flowPin1, INPUT);           //Sets the pin as an input
  attachInterrupt(0, Flow, RISING);  //Configures interrupt 0 (pin 2 on the Arduino
Uno) to run the function "Flow"
  Serial.begin(9600);
  mySerial.begin(115200);//Start Serial


  pinMode(flowPin2, INPUT);
  attachInterrupt(0, Flow, RISING);
  Serial.begin(9600);


}
void loop1(){
  {
    tone(buzzer,2000);
    delay(1000);

}
```

```
  }
void loop2(){
  noTone(buzzer);
}

void loop() {
   // ...for 1 sec
  count1 = 0;      // Reset the counter so we start counting from 0 again
  interrupts();   //Enables interrupts on the Arduino
  delay (1000);   //Wait 1 second
  noInterrupts(); //Disable the interrupts on the Arduino

  //Start the math
  flowRate1 = (count1 * 2.25);        //Take counted pulses in the last second and
multiply by 2.25mL
  flowRate1 = flowRate1 * 60;
  flowRate1 = flowRate1/1000;



  //Convert mL to Liters, giving you Liters / Minute//Convert seconds to minutes,
giving you mL / Minute



  count2 = 0;
  interrupts();
  delay (1000);
  noInterrupts();
  flowRate2 = (count2 * 2.25);
  flowRate2 = flowRate2 * 60;
  flowRate2= flowRate2/1000;
  Serial.print("flowrate1 = ");
  Serial.println(flowRate1);
 //Convert mL to Liters, giving you Liters / Minute
  Serial.print("flowrate2 = ");
  Serial.println(flowRate2);
  difference = flowRate1 - flowRate2;
  Serial.print("Difference =");
  Serial.println(difference);
  values= (String(flowRate1) + ","+ String(flowRate2) +","+ String(difference));
  Serial.println(values);
  //------Sending Data to ESP8266--------// // Starting char
    // removed any buffered previous serial data.
       // sent sensors data to serial (sent sensors data to ESP8266)
       mySerial.println(values);
   //2 digit data // Ending char
    //-----------------------------------//
if(difference ==0.00)
{
    Serial.println("there is a leak");
    loop1
    ();
}
else if (flowRate1 == 0.00){
      Serial.println("No Water Flow");
}
else
{
  Serial.println("there is no leak");
  loop1();
}
Serial.println("------------------------");
}
void Flow()
{
   count1++;
   count2++;//Every time this function is called, increment "count" by 1
}
```

Node Mcu code:

```
#include "ThingSpeak.h"
#include <ESP8266WiFi.h>
const char *ssid = "Manoj123";   // your network SSID (name)
const char *pass= "manoj159";   // your network password
int keyIndex = 0;            // your network key Index number (needed only for WEP)
WiFiClient  client;

unsigned long myChannelNumber = 1543778;
const char * myWriteAPIKey = "B48RMV63BMKNBVNG";
//------- WI-FI details ----------//
```

```
String values,data;
float f1,f2,f3;

void setup()
{
    Serial.begin(115200);
    delay(1000);
    WiFi.mode(WIFI_STA);
    ThingSpeak.begin(client);
}
void loop() {
  if(WiFi.status() != WL_CONNECTED){
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(ssid);
    while(WiFi.status() != WL_CONNECTED){
      WiFi.begin(ssid, pass);  // Connect to WPA/WPA2 network. Change this line if
using open or WEP network
      Serial.print(".");
      delay(5000);
    }
    Serial.println("\nConnected.");
  }
  data=Serial.readString();
  values=data;
  //get comma indexes from values variable
  int fristCommaIndex = data.indexOf(',');
  int secondCommaIndex = data.indexOf(',', fristCommaIndex+1);
  int thirdCommaIndex = data.indexOf(',', secondCommaIndex + 1);

  //get sensors data from values variable by  spliting by commas and put in to
variables
  String flowsensor1 = data.substring(0, fristCommaIndex);
  String flowsensor2 = data.substring(fristCommaIndex+1, secondCommaIndex);
  String difference = data.substring(secondCommaIndex+1);
  f1=flowsensor1.toFloat();
  f2=flowsensor2.toFloat();
  f3=difference.toFloat();
  // set the fields with the values
  ThingSpeak.setField(1, f1);
  ThingSpeak.setField(2, f2);
  ThingSpeak.setField(3, f3);
  // write to the ThingSpeak channel
  int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
  if(x == 200){
    Serial.println("Channel update successful.");
  }
  else{
    Serial.println("Problem updating channel. HTTP error code " + String(x));
  }

  delay(20000); // Wait 20 seconds to update the channel again
}

      //get sensor data from serial put in sensor_data
```

TEAM DETAILS:

1. NAME: P. SAI MANOJ

CLASS: CSE 5B

ROLL NUMBER: 19113101

EMAIL: 19113101@student.hindustanuniv.ac.in

2. NAME: K. SUMANTH KUMAR REDDY

CLASS: CSE 5B

ROLL NUMBER: 19113101

EMAIL: 19113105@student.hindustanuniv.ac.in

3. NAME: P. UMESH CHANDRA

CLASS: CSE 5B

ROLL NUMBER: 19113118

EMAIL: 19113118@student.hindustanuniv.ac.in