**REAL TIME WATER QUALITY MONITORING SYSTEM**

**INTRODUCTION TO IoT**

**COURSE PROJECT REPORT**

***Submitted by***

**Chirag Jain - 211020415**

**Shresht Mishra - 211020450**

**Srijan Ratrey - 211020454**



**Dr. Shyama Prasad Mukherjee International Institute of Information Technology, Naya Raipur**

**July-2022**

**Abstract**

The need for effective and efficient monitoring, evaluation and control of water quality in rural area has become more demanding in this era. Ensuring safe water supply of drinking water from well is big challenge for rural civilization.

Traditional methods that rely on collecting water samples, testing and analyses in water laboratories are not only costly but also lack capability for real-time data capture, analyses and fast dissemination of information to relevant stakeholders for making timely and informed decisions.

In this project, a real time water quality monitoring system prototype developed for water quality monitoring in well is presented. The system consists of an arduino, Analog to Digital Converter, Water quality measurement sensors. It detects water temperature, dissolved oxygen and pH in real-time and disseminates the information in graphical and numerical formats to relevant stakeholders through a mobile app .

The experimental results show that the system has great prospect and can be used to operate in real world environment for optimum control and protection of water resources by providing key actors with relevant and timely information to facilitate quick action taking.

1. **Introduction**

In the 21st century, there are lots of inventions, but at the same time were pollution, global warming and in rural areas people are dependent on well water , because of this there is no safe drinking water for the rural pollution. Maintaining pure water in well is getting more challenging day by day.

In small towns mainly in wells there are a lot of pollutants that can degrade water quality like acids , plants life , decomposition , algaes etc . And they reserve that water without any test. And they also don’t know the water is either safe for drinking or not. And now a day’s water quality monitoring in real time faces challenges because of lack of facilities and technologies in villages, growing pollutions etc. Hence there is a need to develop better methodologies to monitor the water quality parameters in real time.

The water parameters pH measures the concentration of hydrogen ions. It shows the water is acidic or alkaline. Pure water has 7 pH value, less than 7pH has acidic, more than 7pH has alkaline. The range of pH is 0-14pH. For drinking purpose it should be 6.5-8.5pH. Turbidity measures the large number of suspended particles in water that is invisible. Higher the turbidity higher the risk of diarrhea, cholera. Lower the turbidity then the water is clean. Temperature sensor measures how the water is, hot or cold. Here in this paper we tried to find the problem and then make a solution for it.

* Here we have solved problem for villagers as any one can see water quality of well water through mobile app or they can appoint anyone from panchayat to monitor quality of well water and take required actions.
* Our app will show real time data so that they can see real time quality of water and avoid drinking bad quality of water and take certain actions.

1. **Objectives**

* Due to lack of knowledge in villages drinking safe water is a challenge in many areas. Water can be polluted any time. So the water in well may not be safe. And it’s impossible to check the quality of water manually every time. So an automatic real-time water monitoring system is required to monitor the health of the water in wells .
* So it can warn us automatically if there is any problem with the well water. And we can check the quality of the water anytime and from anywhere. By keeping this mind we designed this system especially for village areas.

1. **Methodology:**

**Sensing Layer:**

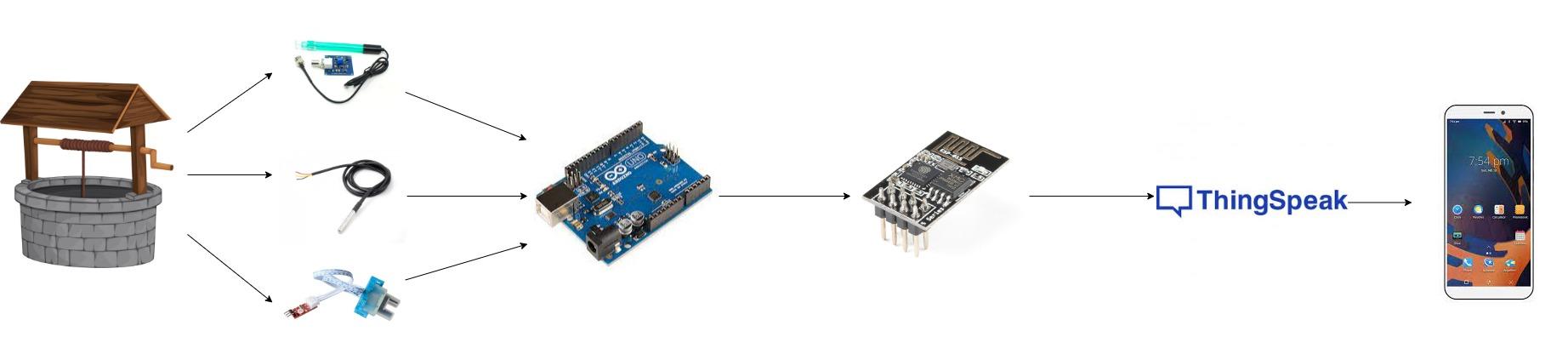
The sensing layer is the heart of this device. In this layer, sensor nodes are placed in the well to sense and collect data. The collected data by sensor are sent to the Arduino which is processed and further proceeded to network layer.

**Network Layer:**

This layer is one of the important layer in the model. In this layer, the processed data is sent to cloud using Wifi module. The data of well in cloud is stored and monitored

**Application Layer:**

The application layer enables the people and the analytics to visualize the collected data in their cell phones and computers. The collected data can be shown in the application. The data from cloud is extracted and send to application.



**Fig 1 Conceptual diagram**

**Design of Sensor Nodes:**

**Components used:**

**Arduino Uno –** It is the central processing location for the IoT prototype. It receives the data from the sensors and ultimately sends it to the cloud via internet with the help of a communication module. Microprocessor inside - Atmega328P.



**Analog pH sensor** - It is the analog pH sensor designed for Arduino and collects the pH information from the water.



**Analog Turbidity sensor** - Arduino Turbidity Sensor is able to detect and verify the quality of the water. The sensor operates on the principle that when the light is passed through a sample of water, the amount of light transmitted through the sample is dependent on the amount of soil in the water*.*

It measures in the range of ~ 1000 NTU

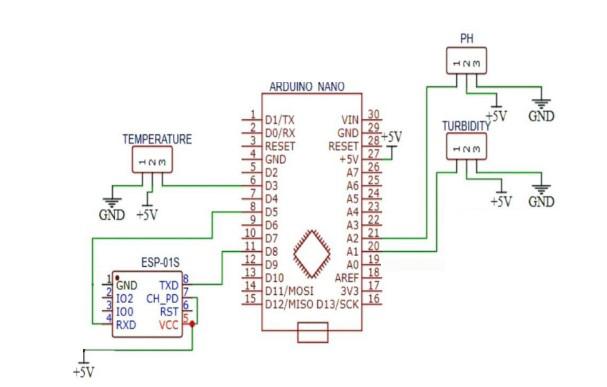


**Temperature sensor - DS18B20 -** Used for measuring the temperature of the Water. It Measures Temperatures from -55°C to +125°C and is a 1-Wire Digital Thermometer. It gives the output in the form of a digital single bus.



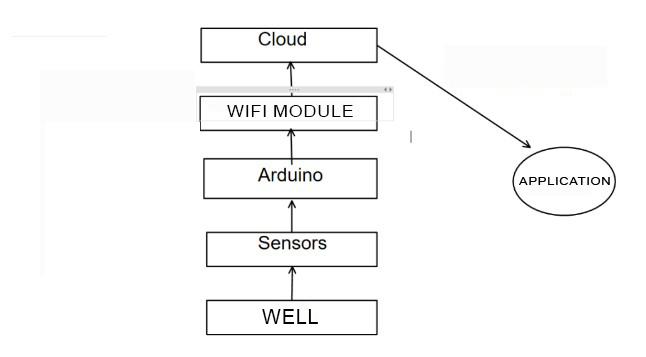
**Esp 8266 WiFi module-** The ESP8266 WiFi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. The ESP8266 is capable of either hosting an application or offloading all WiFi networking functions from another application processor.

We have used Arduino Uno R3 along with ESP8266 module for Wi-Fi connectivity. The choice of using Arduino over other commercial alternatives was driven by the low-cost and availability of the required pins. The ph. sensor, turbidity sensor and temperature sensor are connected to Arduino. Arduino in turn is connected to ESP8266 which received the sensed data in the form of string from Arduino and pushed the data to the cloud for remote access.

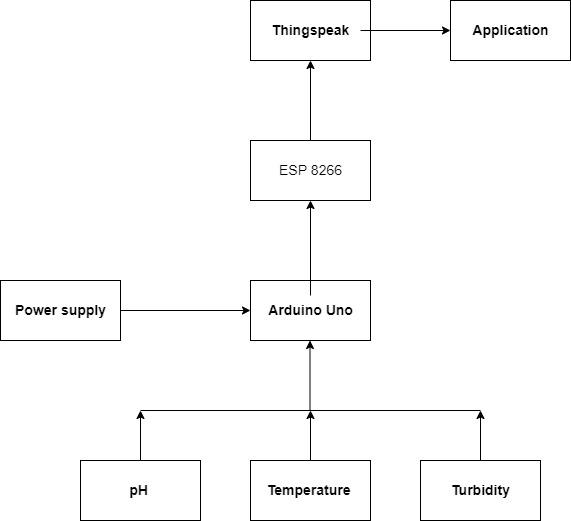
We have precalibrated three sensors for measuring three vital parameters: temperature, turbidity, and ph. Each parameter helps in monitoring the water quality, which can be utilized to the advantage of the local people for a better quality. The circuit diagram of the sensor nodes is presentedin Figure 2

**Fig 2. Circuit diagram**

**Working explanation using flowchart:**

****

**Block diagram :-**

****

**Result:**

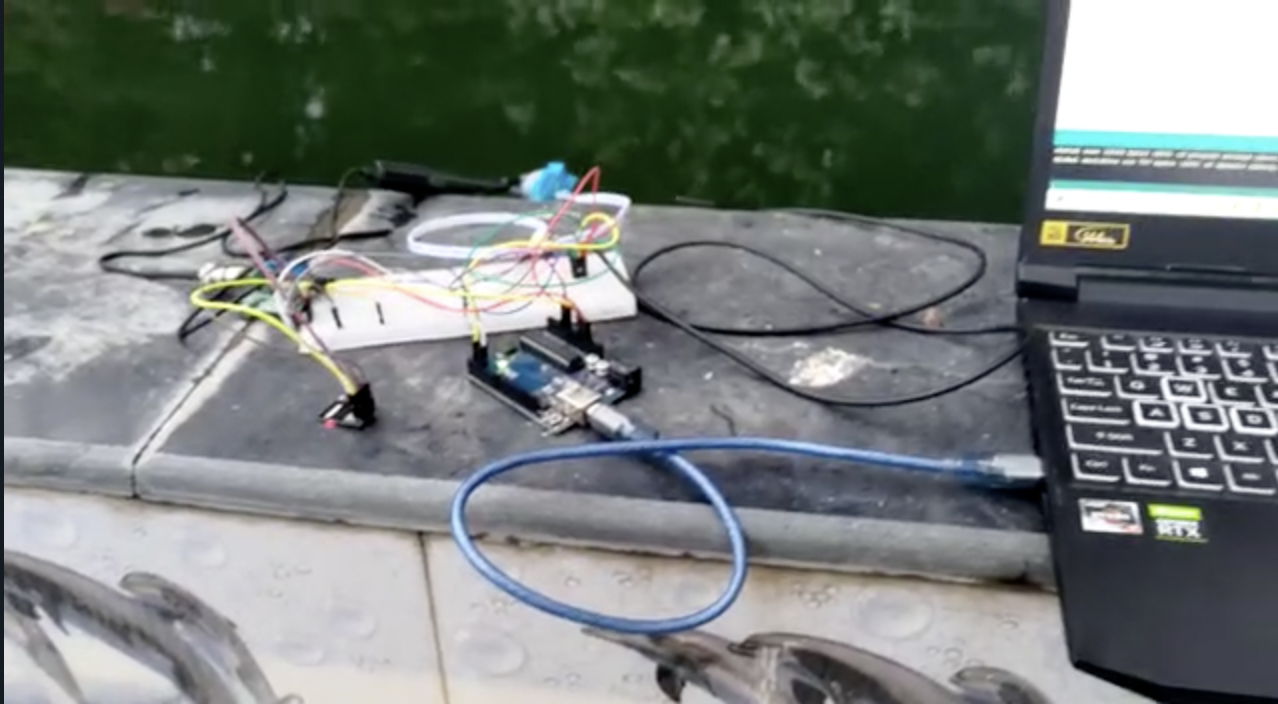
After implementation of the prototype using the above-described methodology, the following result is obtained:

The prototype has been deployed in well.

1. **Results and Discussion**

**4.1. Real Life hardware implementation**

Since we couldn't find a well near us, we took a sample from a pond-like pool.

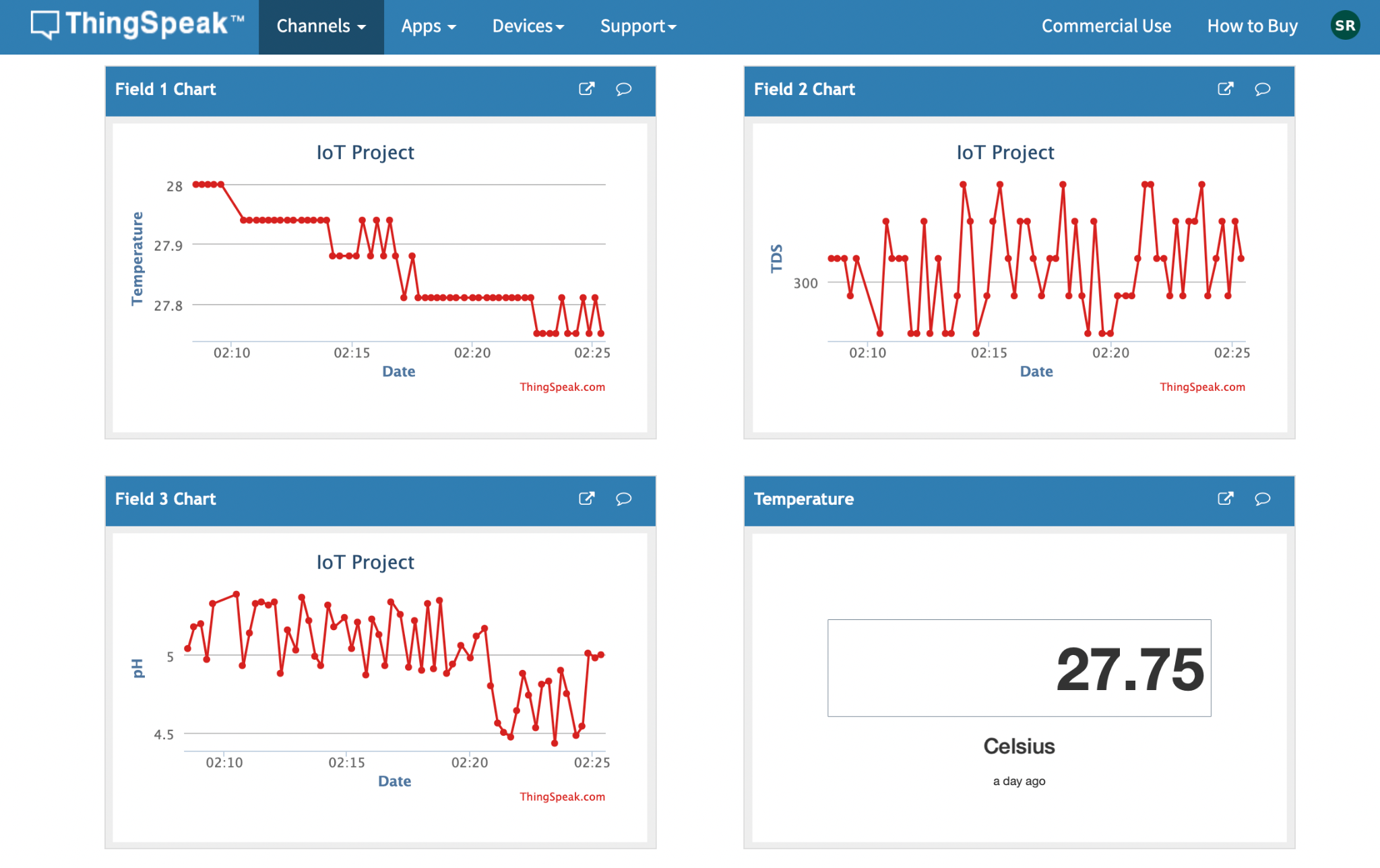
****

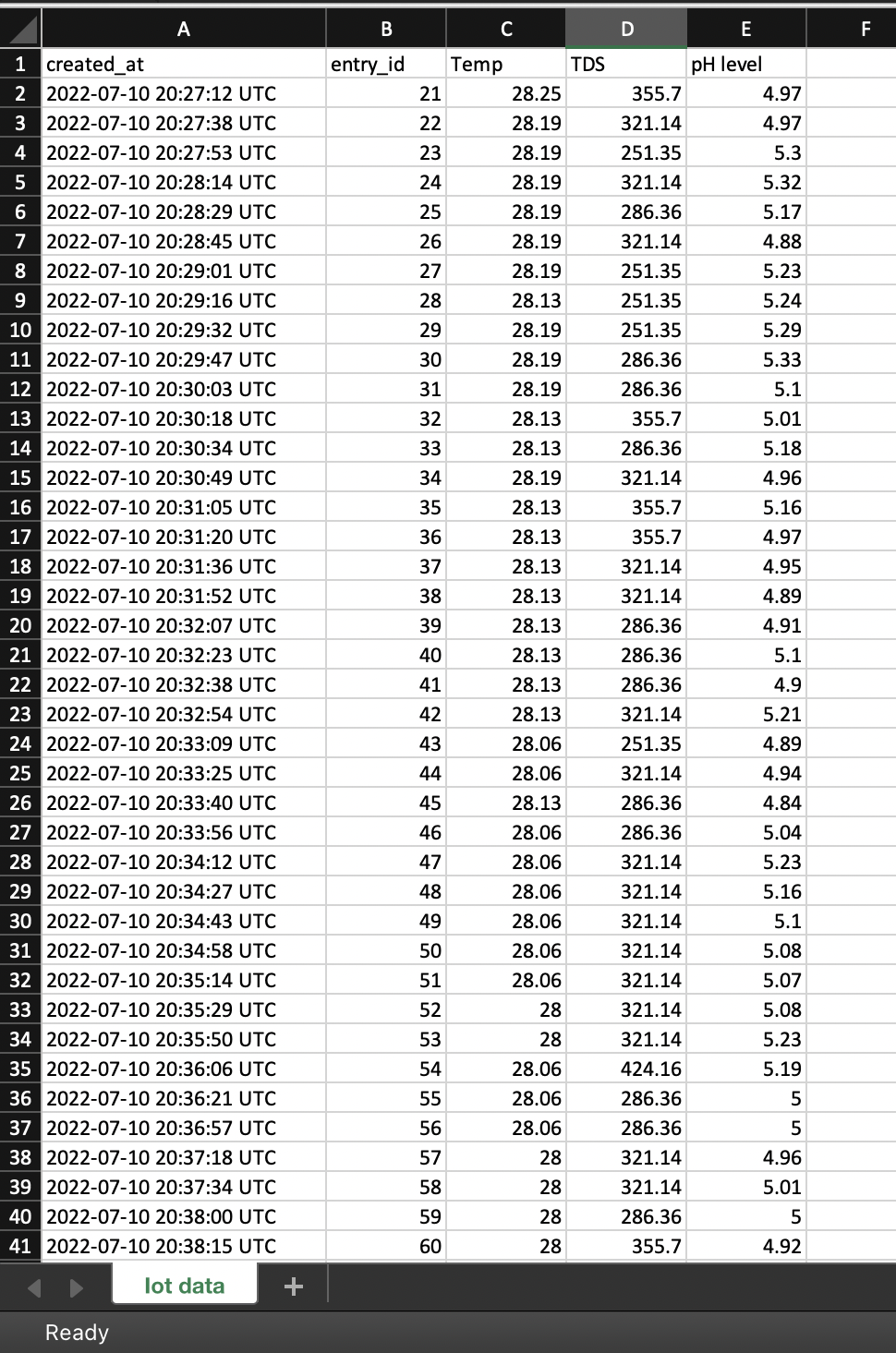
****

**4.2. Results:**

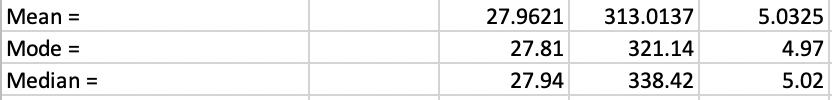
Here, Field 1 chart shows Temperature, Field 2 chart shows TDS(unit of TDS is

NTU), Field 3 chart shows pH level.



After collecting the data from Thingspeak we converted it into a csv File so that we can analyze the data.

**4.3. Analysis of Received data:**

****

* **As we can see the temperature is normal but pH is little acidic and TDS is above 100 NTU. So it is unfit for normal use and should not be used for drinking.**

1. **Comparison of your solution with the existing solutions**

* This project is specifically designed for well water in villages so it is designed according to requirement in villages and keeping in mind all conditions in .
* Our designed system is cost effective , portable and very easy to use .
* The App designed by us is very user friendly , anyone can understand it very easily and can derive conclusion about the quality of well water

.

1. **Conclusion and future scope**

Monitoring of real time quality of Water from well in villages makes use of PH, turbidity and temperature sensor with arduino and existing Cloud system for data analytics. The system can monitor water quality automatically, triggers alarms immediately to prevent any health hazards. So, the system is likely to be more economical, convenient and fast. The system has good flexibility. Only by replacing the corresponding sensors and changing the relevant software programs, this system can be used to monitor other water quality parameters. The operation is simple. The system can be expanded to monitor hydrologic, air pollution, industrial and agricultural production and so on. It has widespread application and extension value.

1. **Learning outcomes**

Making this project pushed us to expand our knowledge. Some of our new learnings are given

below.

● Learned to work with microcontrollers like Arduino UNO.

● Learned to use Arduino IDE how to upload code .

● Learned to make and test prototypes.

● Learned the basics about ic and digital sensors.

● Learned to send and receive data to and from the cloud using microcontrollers

● Learned to send data to thingspeak and to app using wifi module .

● Learned about the working conditions and safety practices .

References

1. <https://how2electronics.com/diy-turbidity-meter-using-turbidity-sensor-arduino/#:~:text=Interfacing%20Turbidity%20Sensor%20with%20Arduino,-Now%20let's%20make&text=You%20can%20do%20this%20by,shown%20in%20the%20image%20above>.
2. <https://create.arduino.cc/projecthub/TheGadgetBoy/ds18b20-digital-temperature-sensor-and-arduino-9cc806>
3. <https://how2electronics.com/ph-meter-using-ph-sensor-arduino-oled/>
4. <https://create.arduino.cc/projecthub/user16726/configuring-the-esp8266-using-an-arduino-0ab2e6>
5. https://create.arduino.cc/projecthub/PatelDarshil/how-to-communicate-with-esp8266-via-arduino-uno-f6e92f

**Source code:-**

#include <SoftwareSerial.h> //Software Serial library

#include <Wire.h>

SoftwareSerial espSerial(3, 4); //Pin 3 and 4 act as RX and TX. Connect them to TX and RX of ESP8266

#define DEBUG true

String mySSID = "ROG Phone 5\_1260"; // WiFi SSID

String myPWD = "sm@90700"; // WiFi Password

String myAPI = "50F4X0DRKBI5QH7J"; // API Key

String myHOST = "api.thingspeak.com";

String myPORT = "80";

String myFIELD = "field1";

String myFIELD2 = "field2";

String myFIELD3 = "field3";

#include <OneWire.h>

#include <DallasTemperature.h>

#define ONE\_WIRE\_BUS 2

OneWire oneWire(ONE\_WIRE\_BUS);

DallasTemperature sensors(&oneWire);

unsigned long int avgValue; //Store the average value of the sensor feedback

float b;

int buf[10],temp;

void setup(){

Serial.begin(9600); //init serial 9600

espSerial.begin(9600);

espData("AT+RST", 1000, DEBUG); //Reset the ESP8266 module

espData("AT+CWMODE=1", 1000, DEBUG); //Set the ESP mode as station mode

espData("AT+CWJAP=\""+ mySSID +"\",\""+ myPWD +"\"", 1000, DEBUG); //Connect to WiFi network

}

void loop(){

float measurement1 = getTempValue();

float measurement2 = gettdsValue();

float measurement3 = getpHValue();

delay(50);

// Serial.print("measurement1 = ");

Serial.println(measurement1);

//Serial.print("measurement2 = ");

Serial.println(measurement2);

// Serial.print("measurement3 = ");

Serial.println(measurement3);

String sendData = "GET /update?api\_key="+ myAPI +"&"+ myFIELD +"="+String(measurement1)+"&"+ myFIELD2 +"="+String(measurement2)+"&"+ myFIELD3 +"="+String(measurement3);

espData("AT+CIPMUX=1", 1000, DEBUG); //Allow multiple connections

espData("AT+CIPSTART=0,\"TCP\",\""+ myHOST +"\","+ myPORT, 1000, DEBUG);

espData("AT+CIPSEND=0," +String(sendData.length()+4),1000,DEBUG);

espSerial.find(">");

espSerial.println(sendData);

}

float getTempValue(){

delay(10);

sensors.requestTemperatures();

float Celsius = sensors.getTempCByIndex(0);

Serial.print("Temperature of water:");

Serial.println(Celsius);

return Celsius;

}

float gettdsValue(){

delay(10);

float volt;

float ntu;

volt = 0;

for(int i=0; i<800; i++)

{

volt += ((float)analogRead(A0)/1023)\*5;

}

volt = (volt/800)+0.6;

volt = round\_to\_dp(volt,2);

if(volt < 2.5){

ntu = 3000;

}else{

ntu = -1120.4\*square(volt)+5742.3\*volt-4353.8;

}

Serial.print("Turbidity of water:");

Serial.println(ntu);

return ntu;

}

float round\_to\_dp( float in\_value, int decimal\_place){

float multiplier = powf( 10.0f, decimal\_place );

in\_value = roundf( in\_value \* multiplier ) /multiplier;

return in\_value;

}

float getpHValue(){

delay(10);

float calibration\_value = 27.54;

int phval = 0;

unsigned long int avgval;

int buffer\_arr[10],temp;

for(int i=0;i<10;i++)

{

buffer\_arr[i]=analogRead(A1);

delay(30);

}

for(int i=0;i<9;i++)

{

for(int j=i+1;j<10;j++)

{

if(buffer\_arr[i]>buffer\_arr[j])

{

temp=buffer\_arr[i];

buffer\_arr[i]=buffer\_arr[j];

buffer\_arr[j]=temp;

}

}

}

avgval=0;

for(int i=2;i<8;i++)

avgval+=buffer\_arr[i];

float volt=(float)avgval\*5.0/1024/6;

float ph\_act = -5.70 \* volt + calibration\_value;

Serial.print("pH of water:");

Serial.println(ph\_act);

return ph\_act;

}

String espData(String command, const int timeout, boolean debug)

{

Serial.print("AT Command ==> ");

Serial.print(command);

Serial.println(" ");

String response = "";

espSerial.println(command);

long int time = millis();

while ( (time + timeout) > millis())

{

while (espSerial.available())

{

char c = espSerial.read();

response += c;

}

}

if (debug)

{

//Serial.print(response);

return response;

}

}