Saline Monitoring System

***Mini*** ***Project*** ***Report*** ***submitted*** ***in*** ***partial*** ***fulfillment.***

***of*** ***the*** ***requirement*** ***for*** ***the*** ***degree*** ***of***

**T.** **E.** **(Information** **Technology)**

Submitted By

**Pranali Sharad Darekar**

**Vaishnavi Navnath Lalage**

**Manali Sanjay Surve**

Under the Guidance of

Prof. Vinita Bhandiwad

Department of Information Technology





Vidyalankar Institute of Technology

Wadala(E), Mumbai 400 037

University of Mumbai

2021-22

CERTIFICATE OF APPROVAL

**For**

**Mini** **Project** **Report**

This is to Certify that

**Pranali Sharad Darekar**

**Vaishnavi Navnath Lalage**

**Manali Sanjay Surve**

Have successfully carried out Mini Project entitled

“**Saline Monitoring System**”

In partial fulfillment of degree course in

Information Technology

As laid down by University of Mumbai during the academic year 2021-22

Under the Guidance of

“ Prof. Vinita Bhandiwad ”

Signature of Guide Head of Department

Examiner 1 Examiner 2 Principal

Dr. S. A. Patekar

**ACKNOWLEGEMENT**

We would like to express our deepest appreciation to all those who provided us the possibility to complete this report. We express our profound gratitude we give to our **Prof.** Ma’am, our respectable project guide, for her gigantic support and guidance. Without her counseling our project would not have seen the light of the day.

We extend our sincere thanks to **Dr. Vipul Dalal**, Head of the Department of

Information Technology for offering valuable advice at every stage of this undertaking.

We would like to thank all the staff members who willingly helped us. We are grateful to VIDYALANKAR INSTITUTE OF TECHNOLOGY for giving us this opportunity.

The days we have spent in the institute will always be remembered and also be

reckoned as guiding in our career.

* **Pranali Sharad Darekar**
* **Vaishnavi Navnath Lalage**
* **Manali Sharad Darekar**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| Sr. No. | Topic | Page No. |
| 1 | Introduction | 6 |
| 2 | Aim & Objectives | 7 |
| 3 | Problem Definition | 8 |
| 4 | Proposed System  4.1 Block Diagram  4.2 Flow Chart | 9  10 |
| 5 | Components  5.1 Hardware 5.2 Software | 11 |
| 6 | Project Architecture | 12 |
| 7 | Code | 13 |
| 8 | Implementation  8.1 Working  8.2 Circuit Diagram | 17 |
| 9 | Results | 18 |
| 10 | Conclusion & Future Scope | 20 |
|  | IEEE technical paper on project topic |  |
|  | Tinker cad experiment |  |
|  | Technical poster on Project topic |  |
|  | GitHub repository link | 20 |

**Abstract**

In the process of medication, it is a common practice to treat patients with saline for dehydration and other medical ailments to improve the health condition of the patients. When fed with saline continuous observation of nurses is mandatory in monitoring the level of the saline. There are many cases where patients are being harmed due to the staff inattentiveness, as their absence does not notice the completion of saline level in the container. This arises the problem of back flow of blood immediately after the completion of saline in container. Hence to protect the patient from getting harmed, an IoT based saline level monitoring system has been developed. The proposed model incorporates a sensor which continuously detects the saline drops and three minutes prior to the saline content getting empty an alert is sent to the nurse or doctor.

**Introduction**

Arduino microcontroller is a programmable circuit board; unlike other circuit boards the Arduino does not require separate hardware to upload a code and plays a significant role in developing monitoring systems. In hospitals in the process of medication to patients, when the patients are fed with saline they must be constantly monitored. More often in the busy continuous schedule of the staff attending the large set of patients, the nurse may forget to monitor and change the saline bottle as soon as its completion. Thus, the blood rushes back to saline bottle through the intravenous tube because of the imbalance created between the blood pressure and pressure within the empty saline bottle. This may cause the back flow of blood from their vein through cannula resulting in the reduction of patient hemoglobin levels and shortage of red blood cells (RBC’s).

Patients most frequently face this problem in the hospitals. This may even lead to patient's death. So, to overcome this problem there is a necessity to develop saline level monitoring system which reduces the dependency of patients on nurses. In this system, whenever the saline in the bottle completes, automatically the tube is compressed and additionally it also gives a buzzer alerting the staff about the completion of saline in the bottle. By this, we can avoid the patient's problems and even the hospital staff also feel at ease. This system improves efficiency in the work giving greater results in the hospital maintenance. It is cost effective and even flexible for nurses.

**Aim & Objectives**

Aim:

To construct an IOT based saline monitoring system that sends an automated alert message to the authority in consideration and assure the safety of the patient.

Objectives:

To use IR sensors to detect the dripping rate and level of water in the dripping chamber.

To use esp8266 Wi-Fi module to send sensor information on the network.

To calculate the rate and the time required for completion of the saline.

To send an alert message using wireless technology to the concerned person.

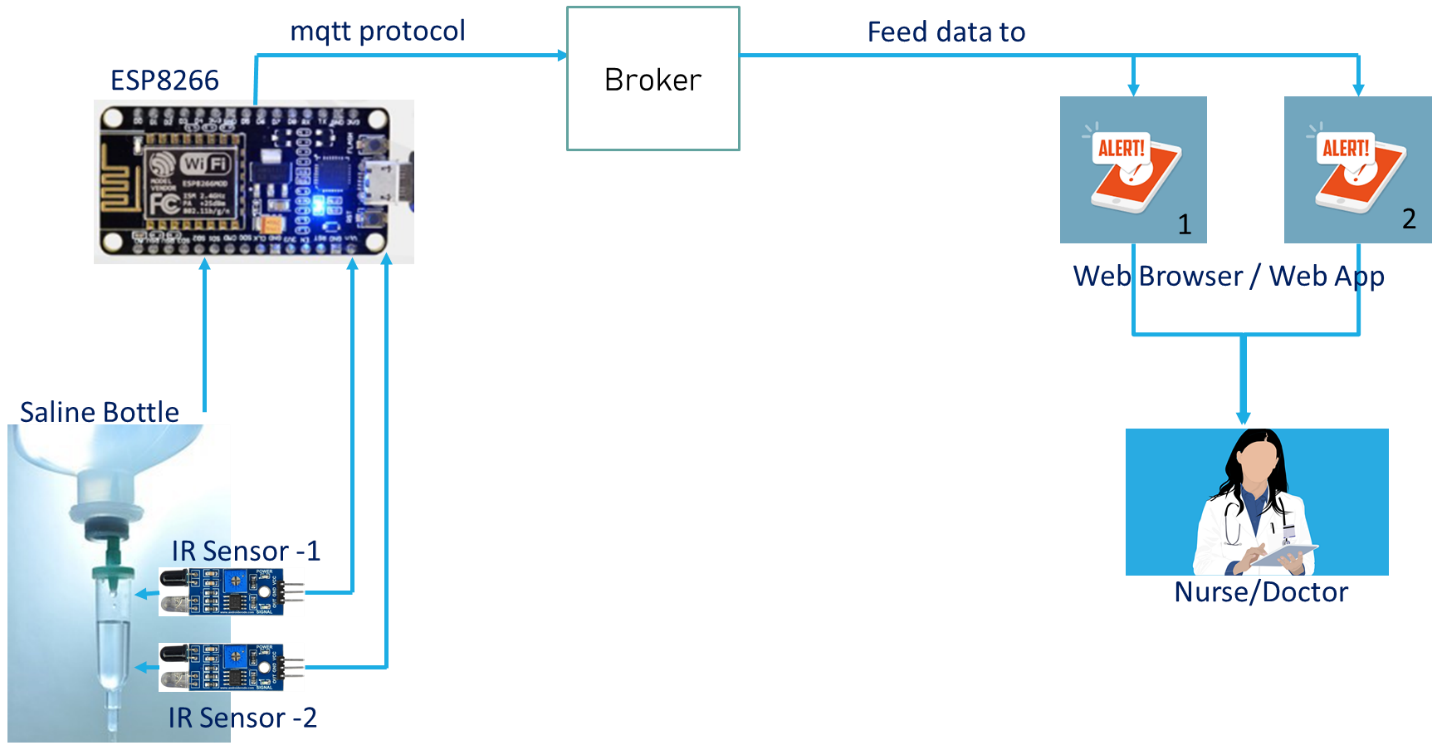
To safeguard the patient.

**Problem Definition**

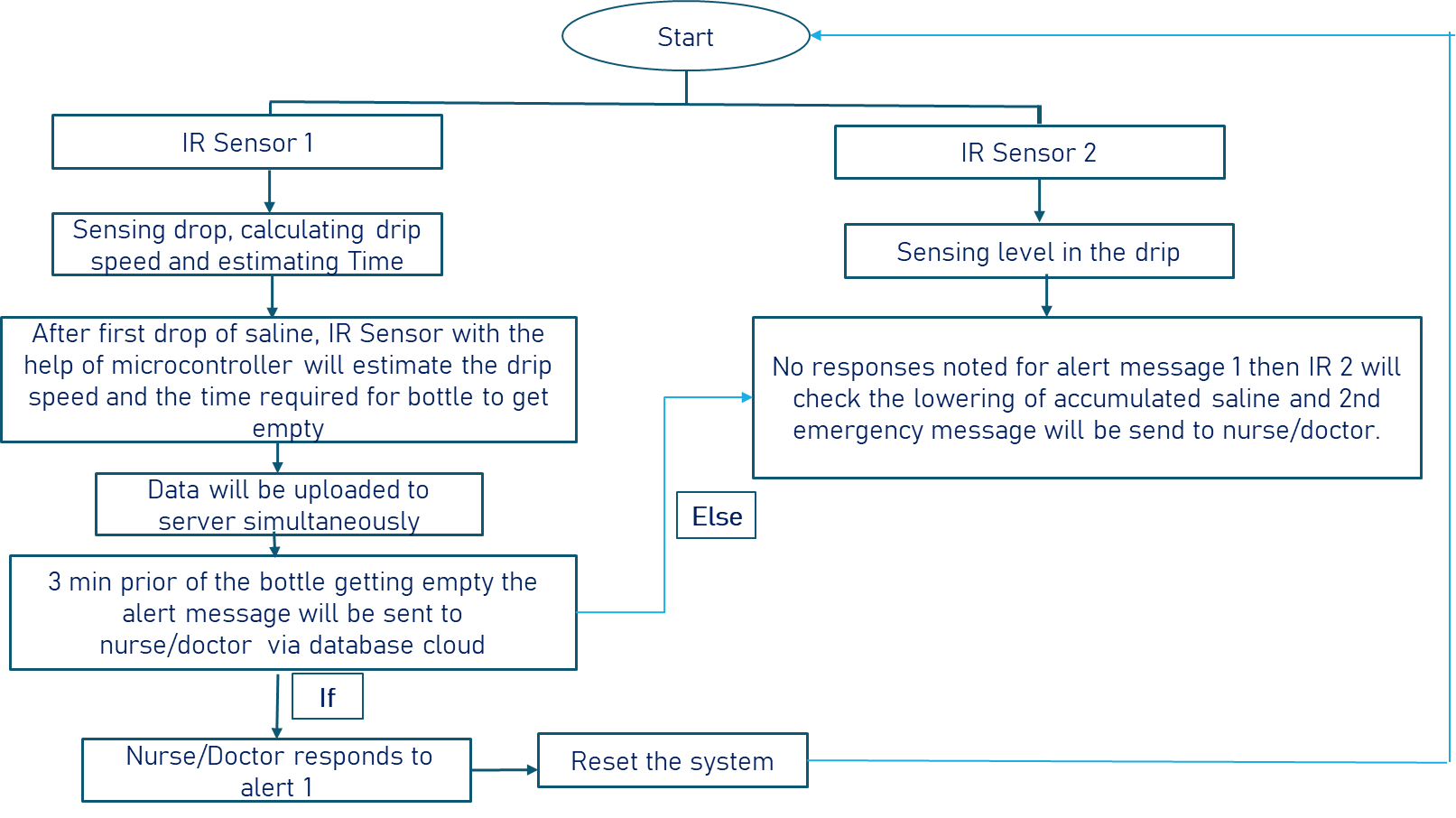
When saline is attached to a patient, the nurse sets the droppage speed and leaves. But when the saline is finished from the bottle there is no way of knowing it unless and until the patient tells the nurse. But till changing or removing saline the whole system will act as a pump in and pump out machine which will cause pain to the patient because it is just pumping the air in and out, in some cases this arises the problem of back flow of blood immediately after the saline bottle is empty. To protect the patient from getting harmed an IOT based saline level monitoring system is proposed. The proposed model incorporates a sensor which continuously detects the saline drops and three minutes prior to the saline content getting empty an alert is sent to the nurse or doctor.

**Proposed System**

4.1 Block Diagram



4.2 Flow Chart



**Components**

5.1 Hardware

* IR Sensors – 2
* ESP8266
* Bread Board
* Jumper Wires

5.2 Software

* Arduino IDE
* Blynk App

**Project Architecture**

1. In this project the saline strip and saline bottle and its pipe are connected to the structure of the project as shown in the picture.

2. The two IR sensors are connected at certain angle to the drip chamber.

I)First IR sensor detects the drops.

II)Second IR sensor detects the level of the liquid.

3. The sensors are connected to the esp8266 wi-fi module through the connecting jumper wires.

4. The esp8266 is connected to the bread board which is again connected to the structure.

**Code**

#include <Blynk.h>

#include <ESP8266WiFi.h>

#define BLYNK\_PRINT Serial

#include <BlynkSimpleEsp8266.h>

int val1 = 0;

int val2 = 0;

int ml = 5;

int ML = 20;

int c1=0;

int c2=0;

int c3=0;

int t1[21];

int TotalTimeRequired, FirstDrop, TwentiethDrop, TimeDifference, MessageTime;

int i=0;

// WiFi

const char \*ssid = "Moto"; // Enter your WiFi name

const char \*password = "chair@12"; // Enter WiFi password

const char \*auth = "FjYFclKFcP-IzoK6vn4fQlLNTl1jzxpK";

void setup() {

Serial.begin(9600);

// connecting to a WiFi network

Blynk.begin(auth, ssid, password, "blynk.cloud", 80);

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

delay(500);

Serial.println("Connecting to WiFi..");

}

Serial.println("Connected to the WiFi network");

}

void loop() {

Blynk.run();

val1 = digitalRead(5); // IR Sensor output pin connected to D1

val2 = digitalRead(4); // IR Sensor output pin connected to D2

if(val1==0){

for(i=0;i<=0;i++){

t1[i] = millis();

Serial.print("Time1=");

Serial.println(t1[i]);

c1++;

while(c1 == 1){

FirstDrop = t1[i]/1000;

Serial.print("FisrtDrop=");

Serial.println(FirstDrop);

break;

}

while(c1 == 20){

TwentiethDrop = t1[i]/1000;

Serial.print("TwentiethDrop=");

Serial.println(TwentiethDrop);

TimeDifference = TwentiethDrop - FirstDrop;

TotalTimeRequired = (ML \* TimeDifference)/ml;

Serial.print("TotalTimeRequired=");

Serial.println(TotalTimeRequired);

break;

}

MessageTime = TotalTimeRequired - 30;

Serial.print("Drop=");

Serial.println(c1);

delay(1000);

if (millis()/1000 >= MessageTime){

Blynk.notify("The Saline Bottle will be empty in next 30 seconds");

break;

}

}

}

else

{

c2++;

}

if (val2==0){

Blynk.notify("The Saline Bottle is empty...");

}

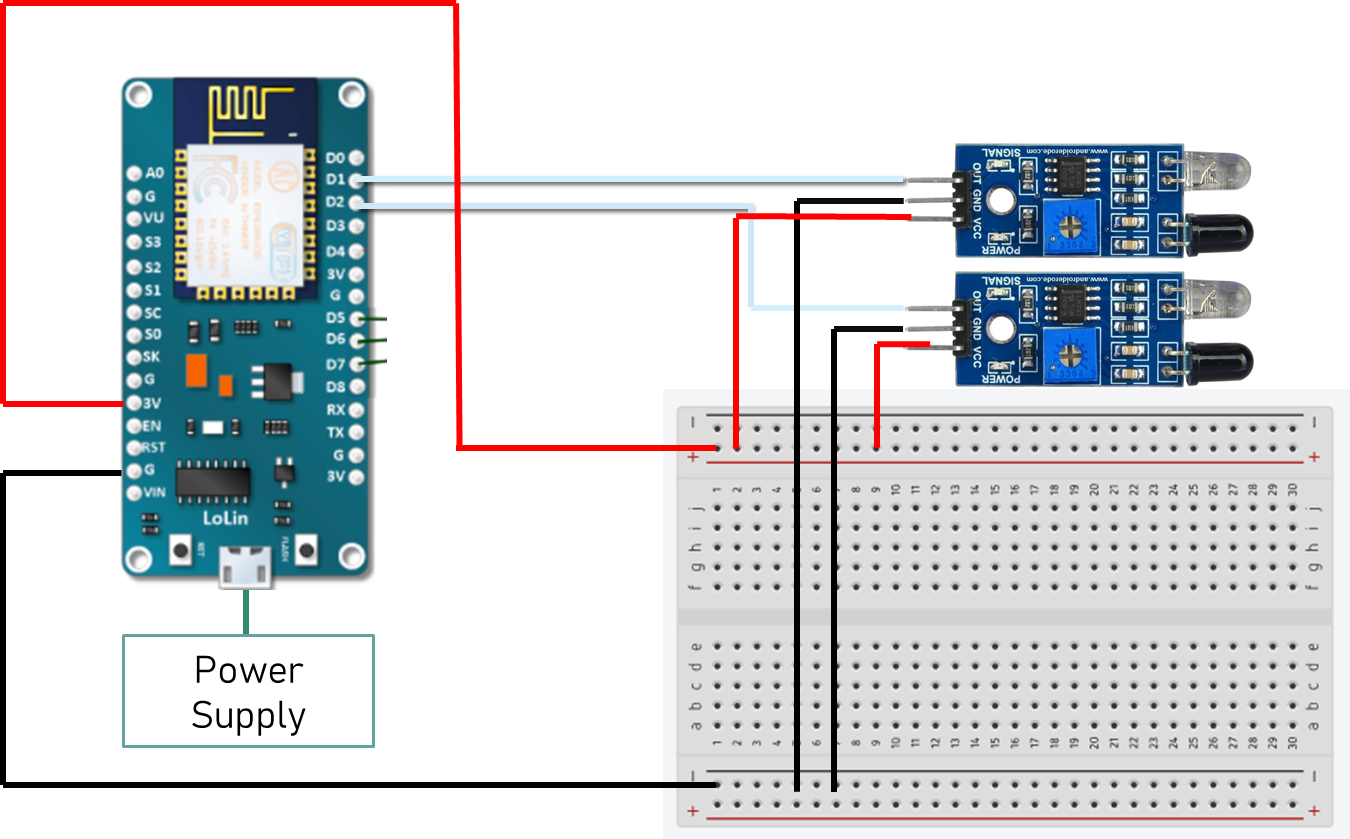
}

**Implementation**

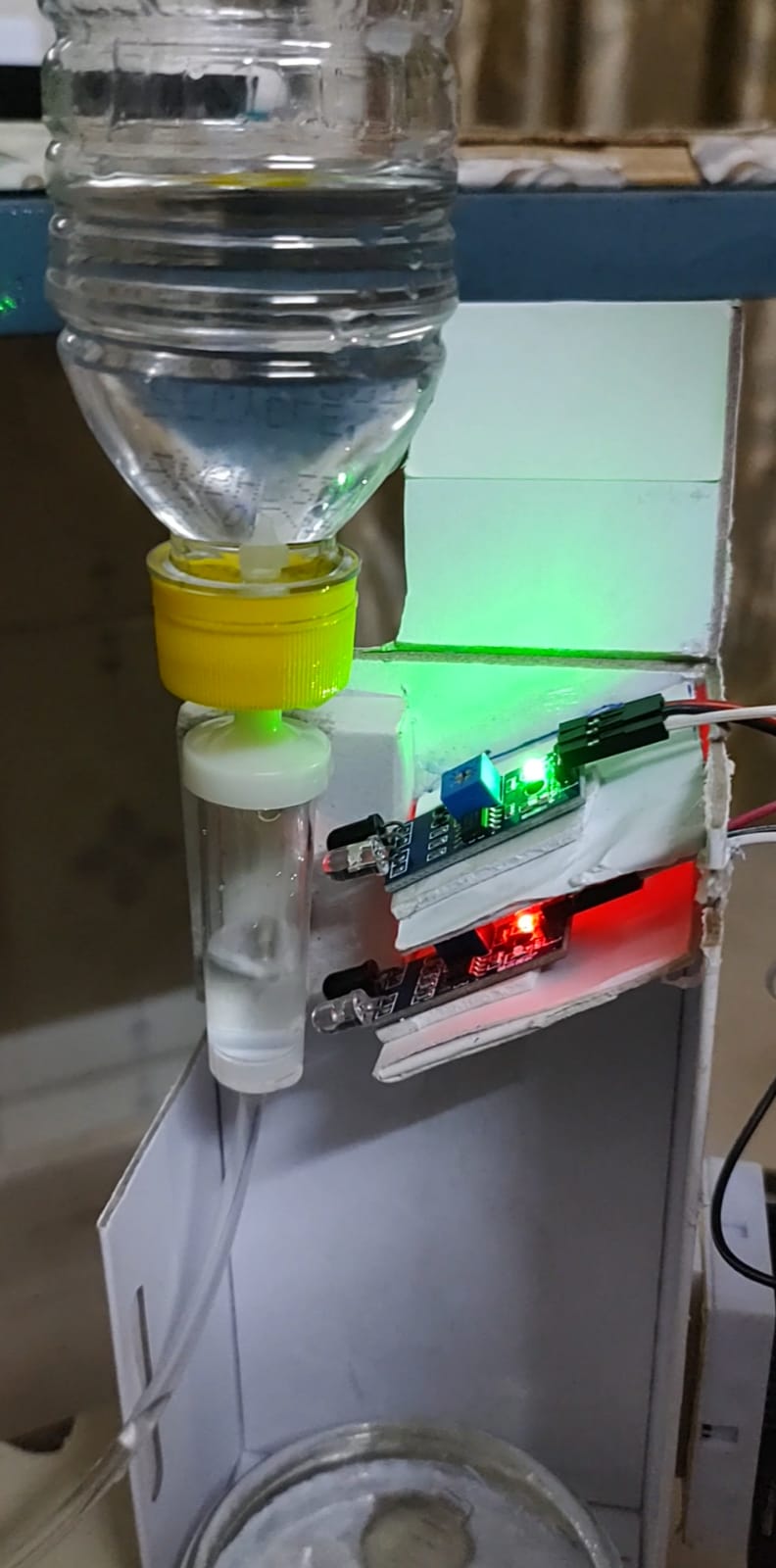
8.1 Working

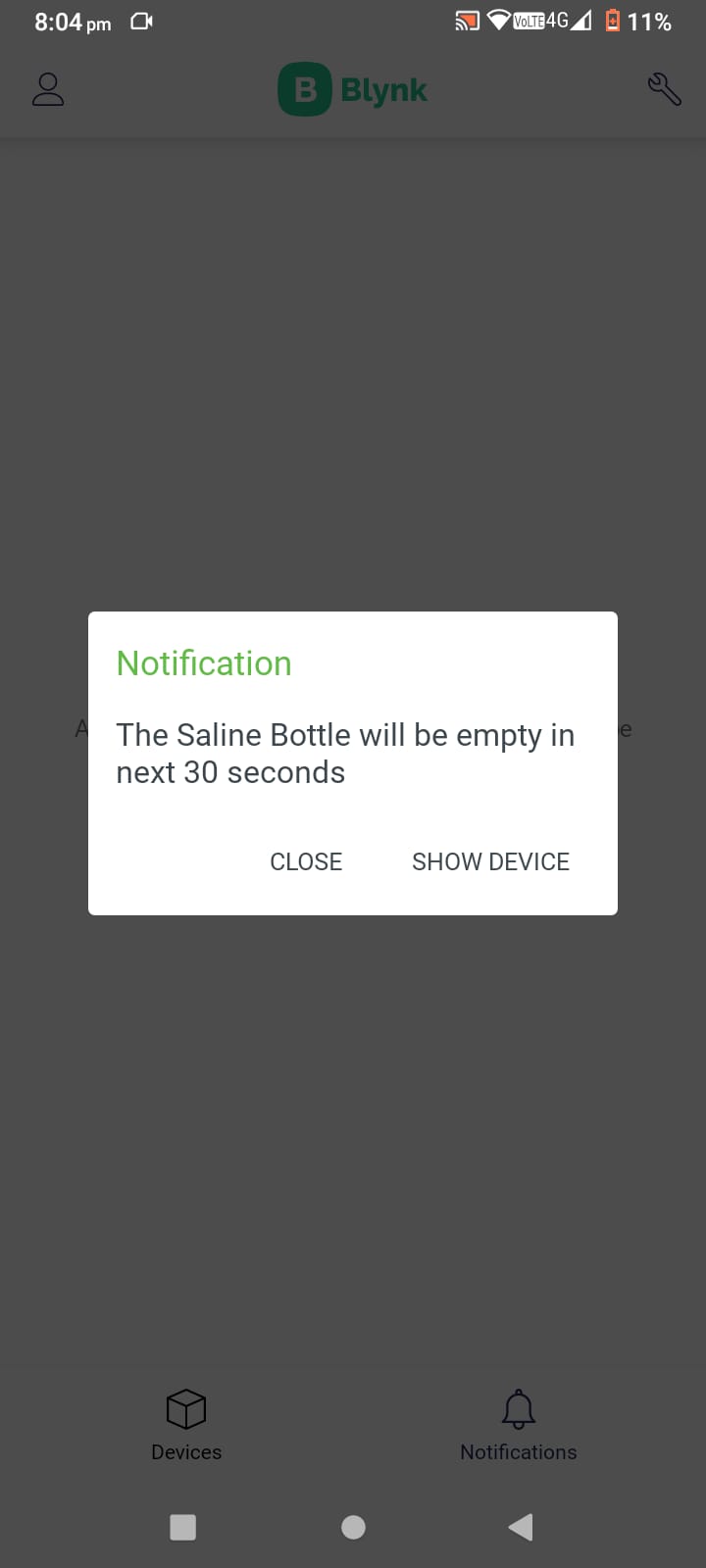
With the help of the first IR sensor will detect the drops in the drip chamber and the program will calculate the drip speed and estimate the time. This will determine the time at which the bottle will get empty. Data will be uploaded to the server simultaneously. The system will then send a message 3 minutes prior of the bottle getting empty. If the Doctor/Nurse responds to the first alert then the system will get reset. If first alert doesn’t get the respond , second IR sensor detects the liquid level and if goes down the certain level then the second alert message is sent.

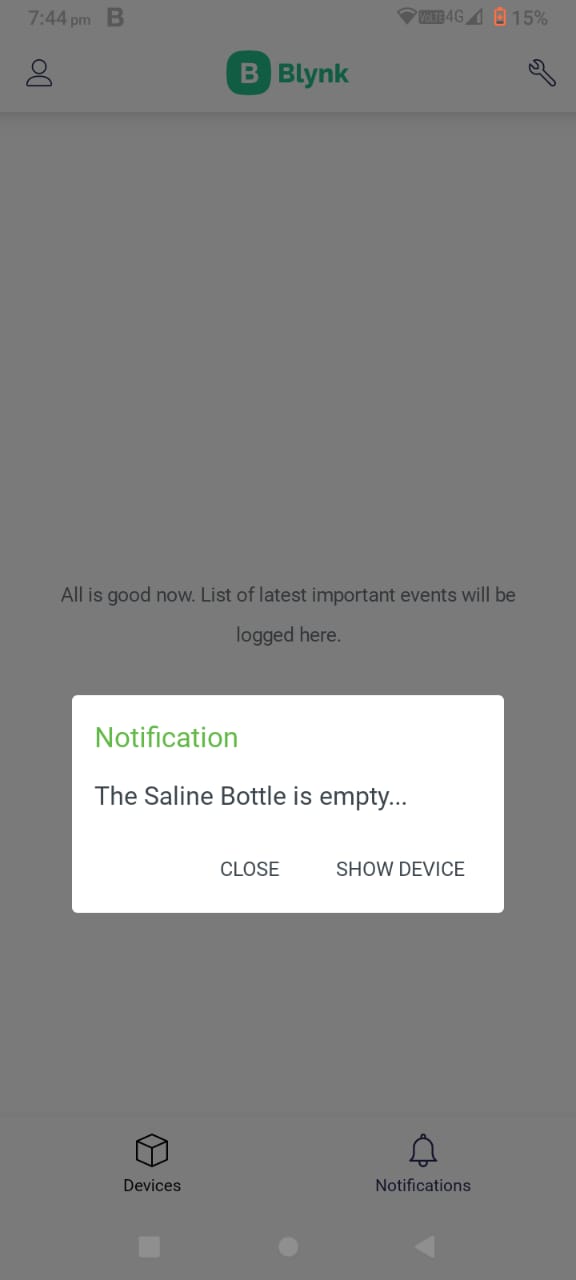
8.2 Circuit Diagram



**Results**







**Conclusion & Future Scope**

With automatic saline monitoring system, the manual effort of continuously monitoring patients injected with saline by the nurses will be reduced. This will lessen the worry in persistent observation by the medical caretaker at a reasonable expense.

In future work, it can send wireless messages to doctors and nurses about the saline droplet rate, it can also include the smart health system, which gives the information about different body parameters. This can help in deciding whether the patient requires another saline bottle or not.

**GitHub Repository Link:**

<https://github.com/Vaishnavi571/Saline-Monitoring-System>