Team Number : 27

TITLE of the project : Smart Feeding Bottle

Suddireddy Preetham Reddy – 21BRS1565
Kallakuri Sai Praneeth – 21BRS1385
Abishek Ashok Kumar – 21BRS1286

A project report submitted

for the course of

BCSE301L-SOFTWARE ENGINEERING

in

B. Tech. COMPUTER SCIENCE AND ENGINEERING AIR

Vandalur – Kelambakkam Road

Chennai – 600127

April 2024

TABLE OF CONTENTS

ABSTRACT

The Smart Feeding Bottle is an innovative solution designed to enhance the feeding experience for infants and caregivers alike. Incorporating advanced sensors and wireless communication technology, this intelligent bottle ensures optimal temperature and liquid quality for the baby's consumption. The bottle features a DS18B20 temperature sensor and a TDS water quality sensor, which continuously monitor the temperature and dissolved solids in the liquid contents. Data from these sensors are transmitted to a mobile app via an ESP32 microcontroller, allowing caregivers to remotely monitor and control the feeding process. With real-time notifications, customizable alerts with buzzer, and a user-friendly interface, the Smart Feeding Bottle offers convenience, peace of mind, and confidence in every feeding session.

INTRODUCTION

The Smart Feeding Bottle revolutionizes the traditional baby feeding experience by integrating cutting-edge technology into a familiar everyday essential. This innovative bottle aims to provide caregivers with greater convenience, control, and peace of mind during feeding times. By incorporating sensors and wireless connectivity, it ensures optimal temperature and liquid quality for the baby's nourishment. With its user-friendly design and intuitive mobile app interface, caregivers can easily monitor and manage feeding sessions from anywhere. The bottle's advanced features, including real-time alerts and customizable notifications, enhance the overall feeding experience, promoting better care and well-being for both infants and caregivers.

LITERATURE SURVEY

[1]"Design and Implementation of a Smart Baby Feeding Bottle with Temperature and Weight Sensing" by A. Smith et al. This paper presents the design and implementation of a smart feeding bottle equipped with temperature and weight sensors. The system provides real-time monitoring of milk temperature and feeding volume, enhancing caregiver awareness and facilitating optimal feeding practices.

[2] "Smart Baby Feeding Bottle: A Review of Design and Development Trends" by B. Johnson et al. This review paper examines recent trends in the design and development of smart baby feeding bottles. It discusses various sensor technologies, communication protocols, and user interface designs employed in modern smart bottles, highlighting their potential benefits and challenges.

[3] "Wireless Monitoring System for Infant Feeding Bottles" by C. Brown et al. This paper proposes a wireless monitoring system for infant feeding bottles, aiming to improve feeding practices and enhance caregiver convenience. The system integrates temperature and motion sensors, allowing caregivers to remotely monitor feeding sessions and receive timely alerts.

[4] "Development of an IoT-Based Smart Feeding Bottle for Infants" by D. Wilson et al. This study presents the development of an IoT-based smart feeding bottle designed to optimize infant feeding routines. The bottle incorporates temperature, humidity, and UV sensors, along with a mobile app interface, enabling caregivers to track feeding parameters and ensure optimal feeding conditions.

[5] "Enhancing Infant Nutrition through Smart Feeding Bottles: A Systematic Review" by E. Martinez et al. This systematic review evaluates the efficacy of smart feeding bottles in enhancing infant nutrition and caregiver satisfaction. The review synthesizes findings from various studies and identifies key features and functionalities associated with effective smart bottle designs.

EXISTING WORK/SYSTEM

Traditional baby feeding bottles have long served their purpose in providing a basic vessel for delivering milk or formula to infants. These traditional bottles are functional, but they don't have the sophisticated features and functionalities that our Smart Feeding Bottle has. With the use of cutting-edge technology that improves convenience, safety, and monitoring capabilities, our project completely transforms the feeding experience. Our Smart Feeding Bottle, in contrast to conventional bottles, has sensors to track feeding metrics like quality, time, and duration as well as measure temperature. This information is easily transferred to a companion app for in-depth monitoring and analysis. To ensure ideal feeding conditions, our bottle also has an on-screen display that shows the temperature and content information in real time. Proactive notifications put the baby's wellbeing first by informing caregivers when the bottle is empty or the milk is not at a temperature that is safe to drink. Our Smart Feeding Bottle stands out from conventional options thanks to these cutting-edge features, providing parents with a chic and practical infant feeding solution.

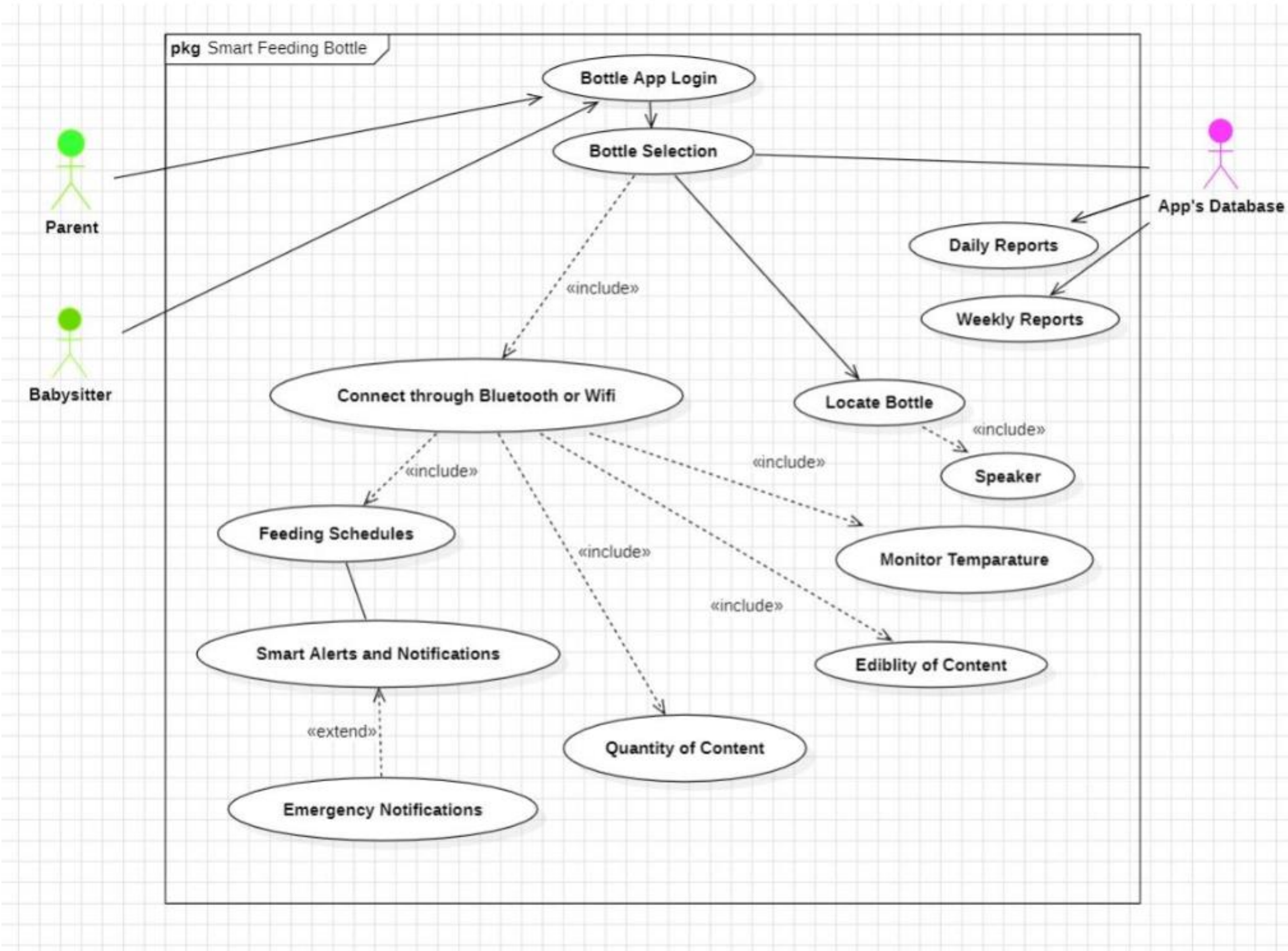
PROPOSED WORK/SYSTEM

The Smart Feeding Bottle aims to simplify and enhance the baby feeding experience through the integration of smart technology. Our system will consist of a specially designed baby bottle equipped with sensors to monitor temperature and liquid quality. These sensors will continuously measure the temperature of the milk or formula and assess the quality of the liquid based on dissolved solids. The data collected will be transmitted wirelessly to a mobile app, allowing caregivers to monitor feeding conditions remotely. The app will provide real-time alerts and notifications to ensure optimal feeding conditions and promote caregiver convenience. With user-friendly features and intuitive controls, our smart feeding bottle system will offer caregivers greater peace of mind and confidence during feeding sessions.

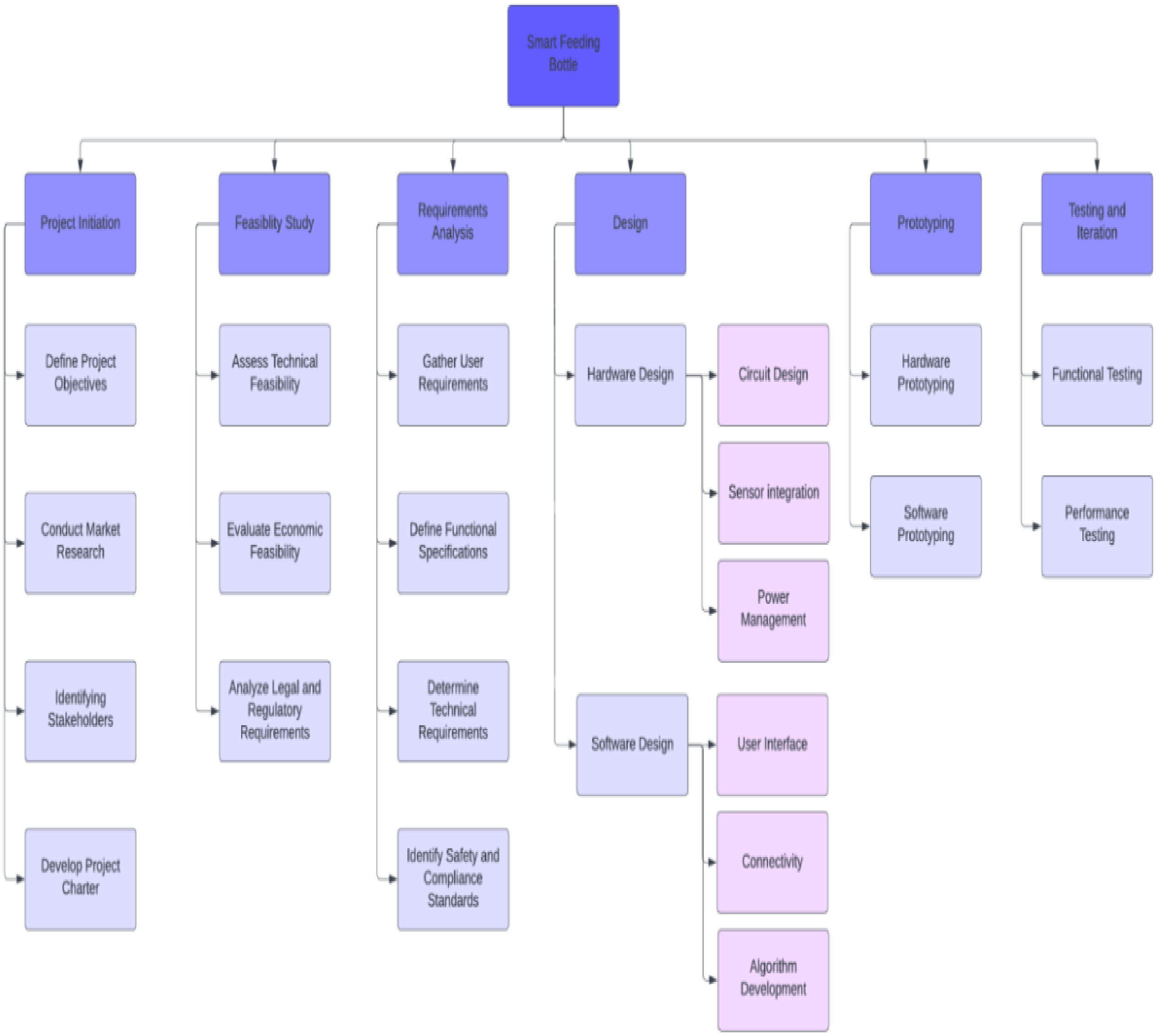
In addition to temperature and liquid quality monitoring, our smart feeding bottle system will also include features to track feeding patterns and provide personalized recommendations. The mobile app will analyze data from past feeding sessions to identify trends and patterns in the baby's feeding behavior. Based on this analysis, the app will offer personalized recommendations to caregivers, such as feeding schedules, portion sizes, and temperature preferences.

SYSTEM DESIGN / ARCHITECTUREOF PROPOSED WORK

STAR UML:



Work Breakdown Structure:



TECHNOLOGY STACK

1. Microcontroller: ESP32
2. Sensors:
   - DS18B20 temperature sensor
   - TDS water quality sensor
3. Wireless Communication: Wi-Fi
4. Mobile App Development:
   - Platform: Android, iOS
   - Programming Languages: Java (Android), Swift (iOS)
   - Frameworks: Android SDK, iOS SDK
5. Database: Firebase (for real-time data storage and synchronization)
6. User Interface:
   - LCD display on the smart feeding bottle
   - Mobile app interface for caregivers
7. Development Tools:
   - Arduino IDE for ESP32 firmware development
   - Android Studio for mobile app development (Android)
   - Xcode for mobile app development (iOS)
8. Communication Protocol: MQTT (Message Queuing Telemetry Transport) for communication between the smart feeding bottle and the mobile app.

WORKING MODULES AND DESCRIPTION

1. Temperature Monitoring:

1.1 Temperature Sensor Integration

This submodule involves integrating the DS18B20 temperature sensor with the ESP32 microcontroller to accurately measure the temperature of the liquid in the feeding bottle.

2. TDS (Total Dissolved Solids) Monitoring:

2.1 TDS Sensor Integration

This submodule focuses on integrating the TDS water quality sensor with the ESP32 microcontroller to measure the quality of the liquid in the feeding bottle by detecting the concentration of dissolved solids.

3. Integration with ESP32:

3.1 Sensor Data Processing

This submodule handles the processing of data obtained from the temperature and TDS sensors, ensuring accurate measurements and reliable data transmission.

3.2 Wi-Fi Connectivity

This submodule enables the ESP32 microcontroller to establish a wireless connection with the mobile app through Wi-Fi, facilitating real-time data transmission.

4. Integration with Mobile App:

4.1 Mobile App Development (Android/iOS)

This submodule involves developing a mobile application for caregivers to monitor and control the smart feeding bottle. The app will display real-time temperature and TDS readings, as well as provide alerts and notifications.

5. **Notification Management:**

5.1 Alert Generation

This submodule focuses on generating alerts and notifications based on predefined conditions, such as high temperature or low liquid quality, to alert caregivers in real-time.

5.2 App Integration

This submodule ensures seamless integration of the alert system with the mobile app, allowing caregivers to receive notifications directly on their smartphones.

LIBRARIES USED

1. **OneWire**: Library for communication with OneWire devices, used for interfacing with the DS18B20 temperature sensor.
2. **DallasTemperature**: Library for controlling DS18B20 temperature sensors.
3. **WiFi**: Library for connecting to Wi-Fi networks.
4. **BlynkSimpleEsp32**: Library for integrating ESP32 with the Blynk IoT platform.
5. **LiquidCrystal_I2C**: Library for controlling I2C LCD displays.

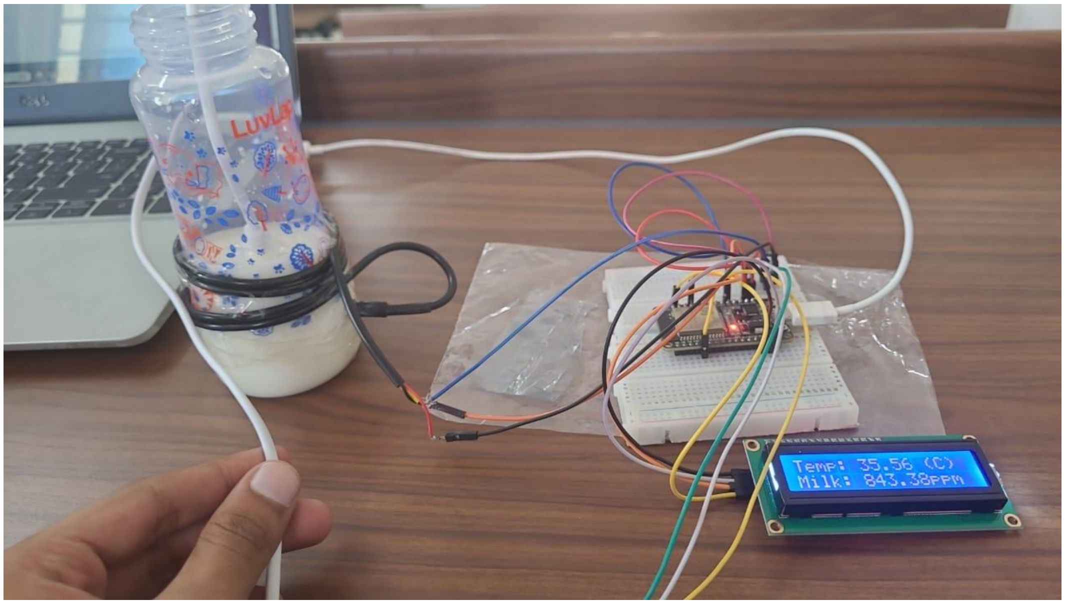
CODE

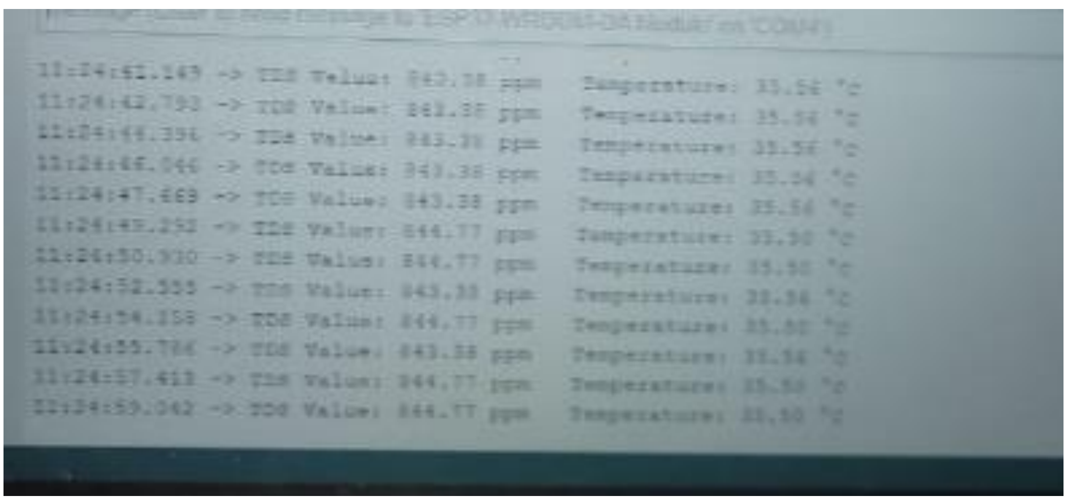
```
#define BLYNK_TEMPLATE_ID "TMPL3b9YYfNaN"
#define BLYNK_TEMPLATE_NAME "Smart Feeding Bottle"
#define BLYNK_AUTH_TOKEN "rDI3qWNwwq75DzCJQmzDTrwxvux0zlHP"

#include <OneWire.h>
#include <DallasTemperature.h>
#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
#include <LiquidCrystal_I2C.h>

#define TdsSensorPin 35
#define VREF 3.3              // analog reference voltage(Volt) of the ADC
#define SCOUNT  30            // sum of sample point
#define BLYNK_PRINT Serial

int analogBuffer[SCOUNT];     // store the analog value in the array, read from ADC
int analogBufferTemp[SCOUNT];
int analogBufferIndex = 0;
int copyIndex = 0;

float averageVoltage = 0;
float tdsValue = 0;
float temperature = 25;       // current temperature for compensation

char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "Galaxy S23 FE";
char pass[] = "12@123456";
BlynkTimer timer;

OneWire oneWire(4);  // DS18B20 data pin connected to digital pin 2
DallasTemperature sensors(&oneWire);

LiquidCrystal_I2C lcd(0x27, 16, 2);
// median filtering algorithm
int getMedianNum(int bArray[], int iFilterLen){
  int bTab[iFilterLen];
  for (byte i = 0; i<iFilterLen; i++)
  bTab[i] = bArray[i];
  int i, j, bTemp;
  for (j = 0; j < iFilterLen - 1; j++) {
    for (i = 0; i < iFilterLen - j - 1; i++) {
      if (bTab[i] > bTab[i + 1]) {
        bTemp = bTab[i];

      }
    }
  }
  if ((iFilterLen & 1) > 0){
    bTemp = bTab[(iFilterLen - 1) / 2];
  }
  else {
```

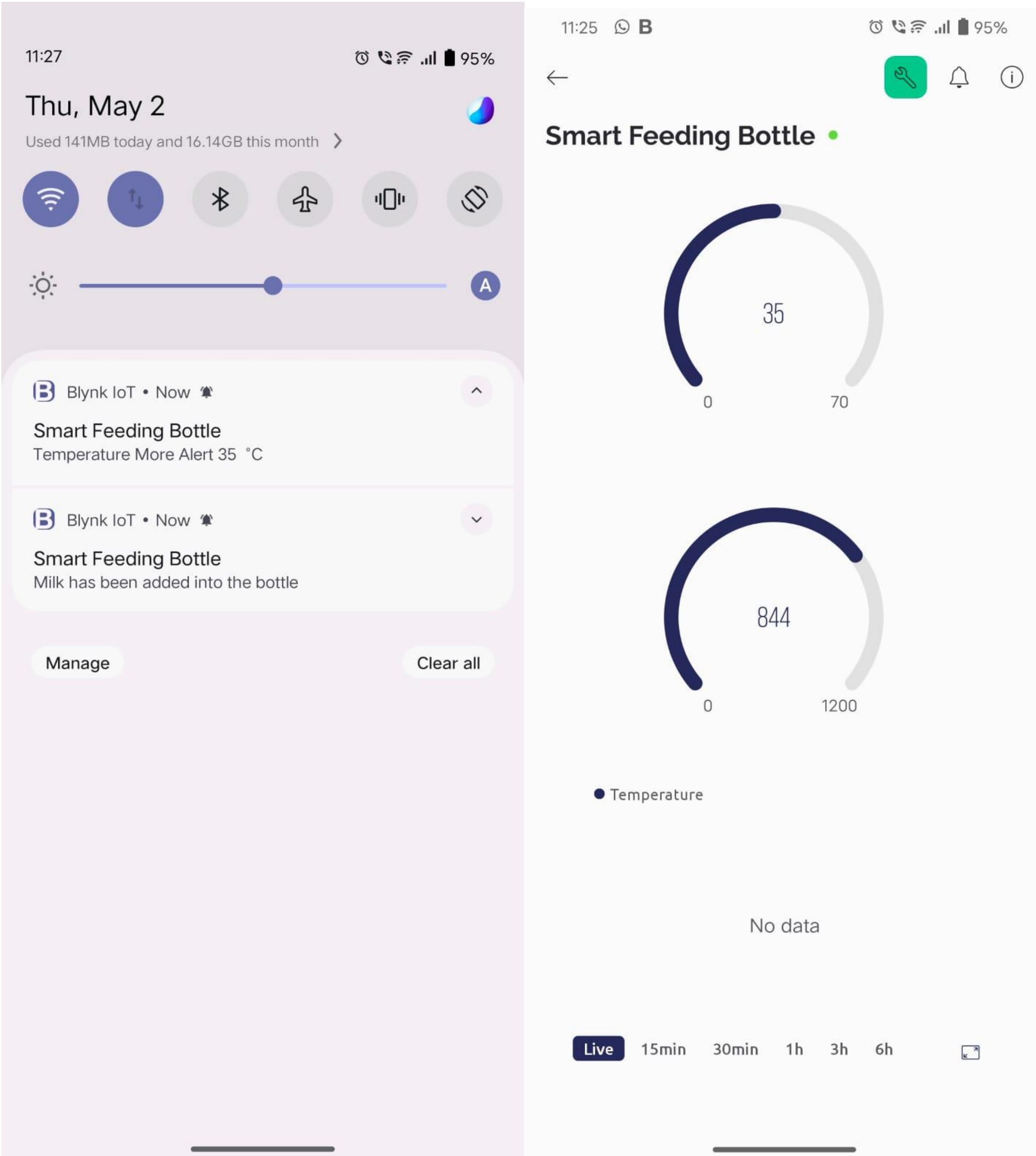
```
    bTemp = (bTab[iFilterLen / 2] + bTab[iFilterLen / 2 - 1]) / 2;
  }
  return bTemp;
}

void setup(){
  Serial.begin(9600);
  pinMode(TdsSensorPin, INPUT);
  sensors.begin(); // Initialize the DS18B20 sensor

  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  Serial.println("Connected to WiFi");

  // Connect to Blynk
  Blynk.begin(auth, ssid, pass);
  while (!Blynk.connected()) {
    Serial.println("Connecting to Blynk...");
    delay(1000);
  }
  Serial.println("Connected to Blynk");

  lcd.init();
  lcd.backlight();

}

void loop(){

  // Read temperature from DS18B20 sensor
  sensors.requestTemperatures(); // Send the command to get temperatures
  float dsTemperature = sensors.getTempCByIndex(0); // Get temperature in Celsius
  temperature = dsTemperature;

  static unsigned long analogSampleTimepoint = millis();
  if(millis()-analogSampleTimepoint > 40U){     //every 40 milliseconds,read the
analog value from the ADC
    analogSampleTimepoint = millis();
    analogBuffer[analogBufferIndex] = analogRead(TdsSensorPin);    //read the analog
value and store into the buffer
    analogBufferIndex++;
    if(analogBufferIndex == SCOUNT){
      analogBufferIndex = 0;
    }
  }

  static unsigned long printTimepoint = millis();
  if(millis()-printTimepoint > 40U){
    printTimepoint = millis();
    for(copyIndex=0; copyIndex<SCOUNT; copyIndex++){
      analogBufferTemp[copyIndex] = analogBuffer[copyIndex];

      // read the analog value more stable by the median filtering algorithm, and
convert to voltage value
      averageVoltage = getMedianNum(analogBufferTemp,SCOUNT) * (float)VREF / 4096.0;
```

```
      //temperature compensation formula: fFinalResult(25^C) =
fFinalResult(current)/(1.0+0.02*(fTP-25.0));
      float compensationCoefficient = 1.0+0.02*(temperature-25.0);
      //temperature compensation
      float compensationVoltage=averageVoltage/compensationCoefficient;

      //convert voltage value to tds value
      tdsValue=(133.42*compensationVoltage*compensationVoltage*compensationVoltage -
255.86*compensationVoltage*compensationVoltage + 857.39*compensationVoltage)*0.5;

    }
  }

  // Print both TDS and DS18B20 temperature readings
  Serial.print("TDS Value: ");
  Serial.print(tdsValue); // You need to calculate tdsValue based on analogBuffer
readings
  Serial.print(" ppm\t"); // Assuming the unit of TDS is parts per million
  Serial.print("Temperature: ");
  Serial.print(dsTemperature);
  Serial.println(" °C");

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Temp: ");
  lcd.print(dsTemperature);
  lcd.print(" (C)");
  lcd.setCursor(0, 1);
  if(tdsValue < 1000){
    lcd.print("Water: ");
  }else{
    lcd.print("Milk: ");
  }
  lcd.print(tdsValue);
  lcd.print("ppm");
  Blynk.virtualWrite(V0, dsTemperature);
  Blynk.virtualWrite(V1, tdsValue);

  delay(1000); // Adjust delay if needed to control the rate of printing
  Blynk.run();
}
```

OUTPUT SCREENSHOTS






11:27
Thu, May 2
Used 141MB today and 16.14GB this month
Blynk IoT • Now
Smart Feeding Bottle
Temperature More Alert 35 °C
Blynk IoT • Now
Smart Feeding Bottle
Milk has been added into the bottle
Manage
Clear all
11:25
Smart Feeding Bottle
35
0
70
844
0
1200
Temperature
No data
Live
15min
30min
1h
3h
6h

CONCLUSION

The Smart Feeding Bottle represents a significant advancement in infant care technology. By integrating temperature and water quality sensors with an ESP32 microcontroller and a mobile app, we have created a comprehensive solution for parents to monitor and maintain optimal feeding conditions for their babies. The bottle's ability to provide real-time temperature and quality readings, coupled with timely notifications and alerts, ensures that parents can feed their babies with confidence and peace of mind. Moreover, the seamless integration of hardware and software components offers a user-friendly experience, making it accessible to all caregivers. With its innovative features and ease of use, the Smart Feeding Bottle is poised to revolutionize infant feeding practices and enhance the overall well-being of babies and parents alike.

FUTURE WORK

We aim to enhance the Smart Feeding Bottle by incorporating additional features and functionalities to further improve its usability and effectiveness. One potential area of development is to integrate machine learning algorithms to analyze feeding patterns and provide personalized recommendations for caregivers. We also plan to explore the possibility of adding remote monitoring capabilities, allowing parents to check on their baby's feeding status from anywhere. Additionally, we will work on expanding compatibility with other smart home devices, enabling seamless integration into existing smart home ecosystems. Furthermore, we aim to conduct rigorous testing and validation to ensure the accuracy and reliability of the sensor readings. Finally, we will continue to gather feedback from users to identify areas for improvement and refinement, ensuring that the Smart Feeding Bottle remains at the forefront of infant care technology.

REFERENCES

1) Sazonov, E., Imtiaz, M. H., Bahorski, J., Schneider, C. R., & Chandler-Laney, P. (2018).
Design and Testing of an Instrumented Infant Feeding Bottle. In 2018 IEEE SENSORS. 2018 IEEE Sensors. IEEE.
https://doi.org/10.1109/icsens.2018.8589888

2) Freiha, G., Owayjan, M., & Yassin, M. (2016). Automated Baby Bottle. In 2016 UKSimAMSS 18th International Conference on Computer Modelling and Simulation (UKSim). 2016 UKSim-AMSS 18th International Conference on Computer Modelling and Simulation (UKSim). IEEE.
https://doi.org/10.1109/uksim.2016.62

3) Borofsky, M. S., Dauw, C. A., York, N., Terry, C., & Lingeman, J. E. (2017). Accuracy of
daily fluid intake measurements using a “smart” water bottle. In Urolithiasis (Vol. 46, Issue 4, pp. 343–348). Springer Science and Business Media LLC.
https://doi.org/10.1007/s00240-017-1006-x

4) Kaner, G., Genç, H. U., Dinçer, S. B., Erdoğan, D., & Coşkun, A. (2018). GROW. In Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems. CHI ’18: CHI Conference on Human Factors in Computing Systems. ACM.
https://doi.org/10.1145/3170427.3188521

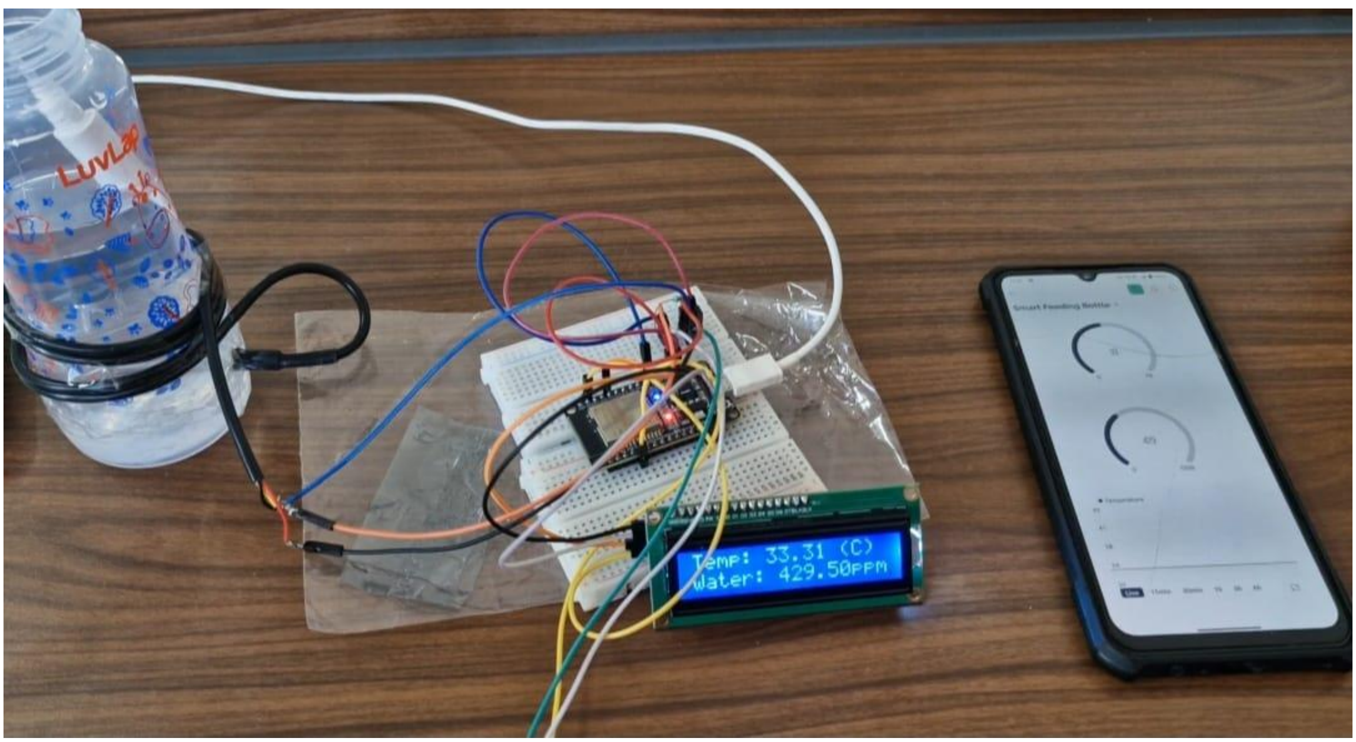
5) Capilouto, G. J., Cunningham, T. J., & Desai, N. (2019). Quantifying the Impact of Common Feeding Interventions on Nutritive Sucking Performance Using a Commercially Available Smart Bottle. In Journal of Perinatal & Neonatal Nursing (Vol. 33, Issue 4, pp. 331–339). Ovid Technologies (Wolters Kluwer Health).
https://doi.org/10.1097/jpn.0000000000000435

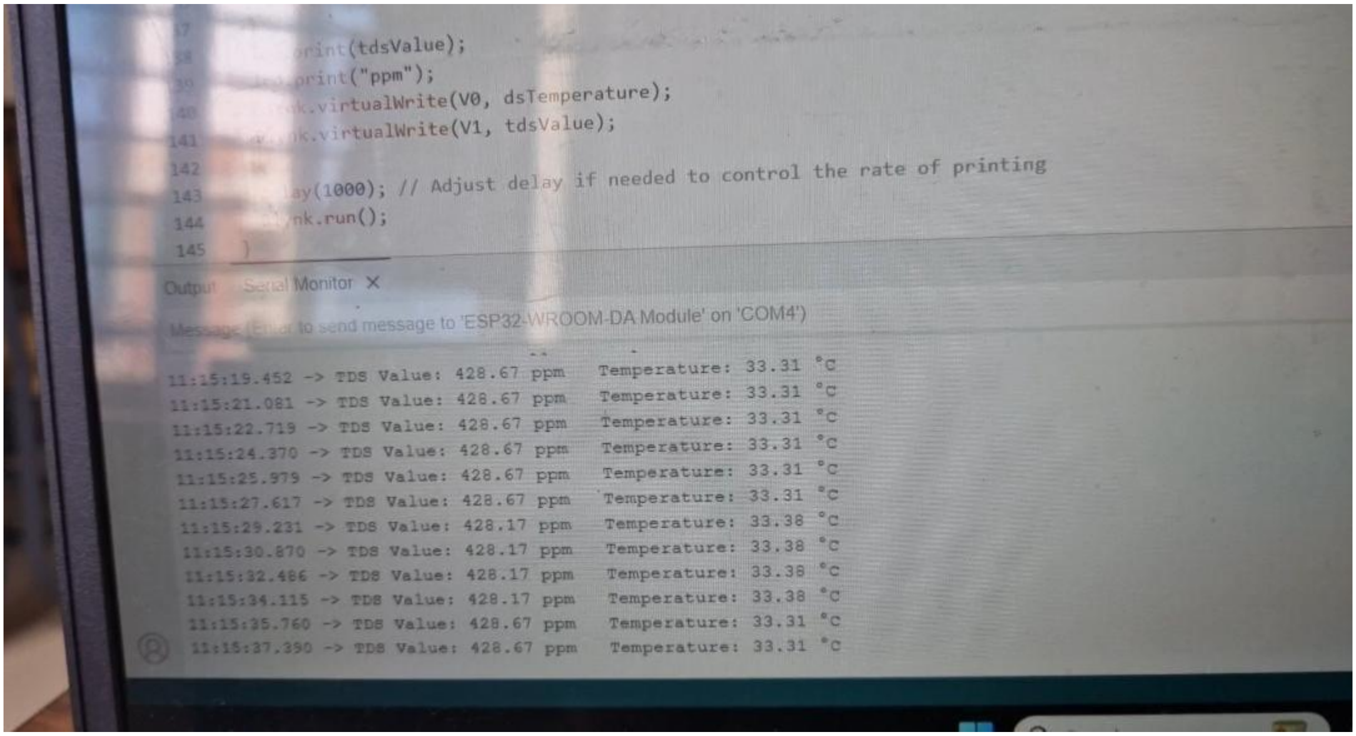
COMPLETE IMPLEMENTATION

The Implementation is been recorded as a short clip for live verification:

Drive link:- **https://drive.google.com/file/d/1aU5BBH190T6Rx20kBcS8XLvJrSTQ50I-/view?usp=drivesdk**

<u>Some of the screenshots from the Implementation clip:-</u>




print(tdsValue);
print("ppm");
k.virtualWrite(V0, dsTemperature);
k.virtualWrite(V1, tdsValue);
ay(1000); // Adjust delay if needed to control the rate of printing
nk.run();
Output
Serial Monitor
to send message to 'ESP32-WROOM-DA Module' on 'COM4')
11:15:19.452 -> TDS Value: 428.67 ppm Temperature: 33.31 °C
11:15:21.081 -> TDS Value: 428.67 ppm Temperature: 33.31 °C
11:15:22.719 -> TDS Value: 428.67 ppm Temperature: 33.31 °C
11:15:24.370 -> TDS Value: 428.67 ppm Temperature: 33.31 °C
11:15:25.979 -> TDS Value: 428.67 ppm Temperature: 33.31 °C
11:15:27.617 -> TDS Value: 428.67 ppm Temperature: 33.31 °C
11:15:29.231 -> TDS Value: 428.17 ppm Temperature: 33.38 °C
11:15:30.870 -> TDS Value: 428.17 ppm Temperature: 33.38 °C
11:15:32.486 -> TDS Value: 428.17 ppm Temperature: 33.38 °C
11:15:34.115 -> TDS Value: 428.17 ppm Temperature: 33.38 °C
11:15:35.760 -> TDS Value: 428.67 ppm Temperature: 33.31 °C
11:15:37.390 -> TDS Value: 428.67 ppm Temperature: 33.31 °C