SMART BUS SYSTEM

A Project Report

Submitted to the APJ Abdul Kalam Technological University in partial fulfillment of requirements for the award of degree

Bachelor of Technology

in

Electronics and Communication Engineering

By

NIMIA DAS A(NSS21EC057)

NIRUPAMA V NAIR(NSS21EC058)

O B KEERTHANA(NSS21EC060)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING NSS COLLEGE OF ENGINEERING

PALAKKAD

KERALA

DEPT. OF ELECTRONICS & COMMUNICATION ENGINEERING

NSS COLLEGE OF ENGINEERING PALAKKAD

2023 - 24

CERTIFICATE

This is to certify that the report entitled **SMART BUS SYSTEM** submitted by **NIMIA DAS A** (NSS21EC057), **NIRUPAMA V NAIR** (NSS21EC058), **O B KEERTHANA** (NSS21EC060) to the APJ Abdul Kalam Technological University in partial fulfillment of the B.Tech. degree in Electronics and Communication Engineering is a bonafide record of the project work carried out by him under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Prof. Sruthi N
(Project Guide)
Assistant Professor
Dept.of ECE
NSS College of Engineering
Palakkad

Prof. Reshmi S
(Project Coordinator)
Assistant Professor
Dept.of ECE
NSS College of Engineering
Palakkad

Vinod G
Associate Professor and
Head Dept.of ECE
NSS College of Engineering
Palakkad

DECLARATION

We hereby declare that the project report **SMART BUS SYSTEM**, Submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by us under the supervision of Prof. Sruthi N.

This submission represents our ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources.

We also declare that we have adhered to the ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not previously formed the basis for the award of any degree, diploma, or similar title of any other University.

Palakkad

02-05-2024

NIMIA DAS A

NIRUPAMA V NAIR

OB KEERTHANA

ACKNOWLEDGMENT

We take this opportunity to express our deepest sense of gratitude and sincere thanks to everyone who helped us to complete this work successfully. We express our sincere thanks to Vinod G, Head of Department, Electronics and Communication Engineering, NSS College of Engineering Palakkad for providing us with all the necessary facilities and support.

We would like to express our sincere gratitude to Prof. Reshmi S, department of Electronics and Communication Engineering, NSS College of Engineering Palakkad Palakkad for the support and cooperation.

We would like to place on record our sincere gratitude to our project guide Prof. Sruthi N, Assistant Professor, Electronics and Communication Engineering, NSS College of Engineering Palakkad for the guidance and mentorship throughout this work.

Finally, we thank our family, and friends who contributed to the successful fulfillment of this seminar work.

NIMIA DAS A

NIRUPAMA V NAIR

OB KEERTHANA

ABSTRACT

The Smart Bus System presents a transformative approach to modernizing public transportation, leveraging a sophisticated integration of hardware and software components to enhance passenger experience, operational efficiency, and system management. At its core, the system comprises an array of hardware elements including an RFID reader, LCD display, GPS module all orchestrated by an Arduino Uno microcontroller. These components work in concert to facilitate key functions such as destination selection, fare calculation, and capacity management. In practice, passengers interact with the system through RFID card tapping, destination selection via the Keypad, and seamless payment confirmation. Integration of hardware components is achieved through meticulous wiring and programming, enabling seamless communication and centralized control. The Smart Bus System promises to revolutionize urban transit, offering a more convenient, efficient, and sustainable mode of public transportation.

CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

In the bustling urban environment, public transportation stands as a lifeline, facilitating the movement of millions daily. Yet, amidst the hustle and bustle, traditional bus systems often find themselves grappling with inefficiencies that hinder both passenger experience and operational efficiency. Current bus systems are plagued by several shortcomings, each posing its own challenge to the smooth functioning of public transit. One glaring issue lies in the archaic methods of passenger counting.

Manual headcounts fail to provide accurate real-time data, leading to overcrowding and discomfort for passengers(Liu et al., 2023). Moreover, the lack of a systematic approach to fare collection results in long queues, fare evasion, and revenue losses, resulting in financial strain on transit authorities.

In light of these challenges,We present the "**SMART BUS SYSTEM**". Through our Smart Bus System, we aim to address these shortcomings head-on. Our system calculates the total time that the bus should stop at a particular destination.

Additionally, we integrate RFID-based fare deduction mechanisms, coupled with GPS technology, to streamline the fare collection process and enhance transaction security(Holia et al., 2019; Ehn et al., 2021).

LITERATURE REVIEW

1. ***Bus Fare Collection System Using RFID and GPS***
    - June 2023
    - June 2023

    DOI:10.13140/RG.2.2.16620.13449

    Authors:

    1. Shazid Bin Zaman - American International University-Bangladesh
    2. Richard Victor Biswas - University of California, Los Angeles
    3. Eftakharul Islam Emon

This paper discusses the development of an Automatic Fare Collection (AFC) combining the Radio Frequency Identification (RFID) and Global Positioning System (GPS) technologies for public transportation buses in Dhaka, Bangladesh. The paper begins with an introduction highlighting the need for such a system due to the dissatisfaction of passengers and high fare collection.The methodology section explains the working principle, components, implementation, and experimental setup of the proposed AFC system. The report then presents a cost analysis and discusses the results of the project, highlighting its limitations. The paper concludes with recommendations for future expansion of the project and references used in the report. The proposed system utilizes RFID and GPS technologies to record passenger boarding and arrival locations and automatically deduct fares. The system eliminates the need for paper tickets, offers onboard ticket inspection, and can introduce spatial validation elements to enhance its usefulness.Buses are currently monitored by conductors who collect fares and distribute paper tickets. However, this method has drawbacks such as time and paper consumption. Portable devices are used, but can be time-consuming. An alternative is the Automatic Fare Collection (AFC) system, which records transaction details and eliminates paper tickets. This system can also introduce spatial validation elements.

2. ***RFID based Smart Bus using Embedded System***

International Journal of Engineering Research & Technology (IJERT)

ISSN: 2278-0181

Published by, www.ijert.org

COCODANTR - 2016 Conference Proceedings

Authors:

1. Mr. D. Baskaran - Assistant Professor, Nandha College of Technology,Erode, India
2. Mr. M. Pattumuthu - UG Scholar, Nandha College of Technology,Erode, India
3. Ms. B. Priyadharshini - UG Scholar, Nandha College of Technology,Erode, India
4. Mr. P. Shabab Akram - UG Scholar, Nandha College of Technology,Erode, India
5. Ms .S. Sripriya - UG Scholar, Nandha College of Technology,Erode, India

Ticket-friendly machines use smart cards to store event details and enable consumers to purchase tickets through their personal accounts. These cards store and transact data, enhancing convenience and security. When the RFID reader reads the user's account, the destination is displayed. This system is more efficient than traditional paper ticketing, saving time and personnel costs while allowing for more efficient fare collection. The system also includes a real-time bus information system using satellite technology to predict bus arrival times. In smartcard pay mode, the RFID reader transfers data to the bank, and the system is controlled by a PIC microcontroller. GPS signals allow passengers to identify their destination via voice-talking GPS. The proposed system integrates RFID and GSM for low-cost transportation management. Displays at bus stops inform passengers of the bus's location. The system handles errors and emergency situations, reducing departure times and underutilization of buses.

3. *Smart bus fare ticketing system using RFID technology and GSM Module*

National Conference on Emerging Technologies for Sustainable Engineering & Management (NCETSEM'18)-2018.

Authors:

1. Antony Fernandas - Assistant Professor
2. Lokesh - Final Year U.G Student
3. Harikrishnan - Final Year U.G Student
4. Karthikraja - Final Year U.G Student

Affiliation:Department of Electronics andCommunication Engineering,SRM Institute of Science and Technology.

This paper's goal is to issue tickets to each passenger and collect money.It will take a great deal of time and lead to human mistake.A novel system is suggested as a solution to this issue. An RFID smart card is used in this proposed system.This device is easy to use; it will recognize the passenger automatically and deduct the fare based on the distance traveled.A message with a well-secured system will be sent to the concerned individual's mobile device.The bus's infrared sensor counts the number of people getting on and off.One-time passwords (OTPs) are secure and valid for only one login or transaction, avoiding vulnerabilities associated with static passwords. RFID-based automatic bus ticketing uses RFID technology to identify and credit passengers' account balances. A smart card with passenger details and account balance is used, and the RFID reader system detects the card's location and calculates the bus fare. The RFID reader then requires an OTP from the passenger's mobile device. A messaging system is introduced to communicate boarding points and destinations, and the number of passengers entering and exiting the bus is calculated. This system helps prevent misuse of smart cards by strangers.

Chapter 2

BLOCK DIAGRAM AND DESCRIPTION

2.1 BLOCK DIAGRAM

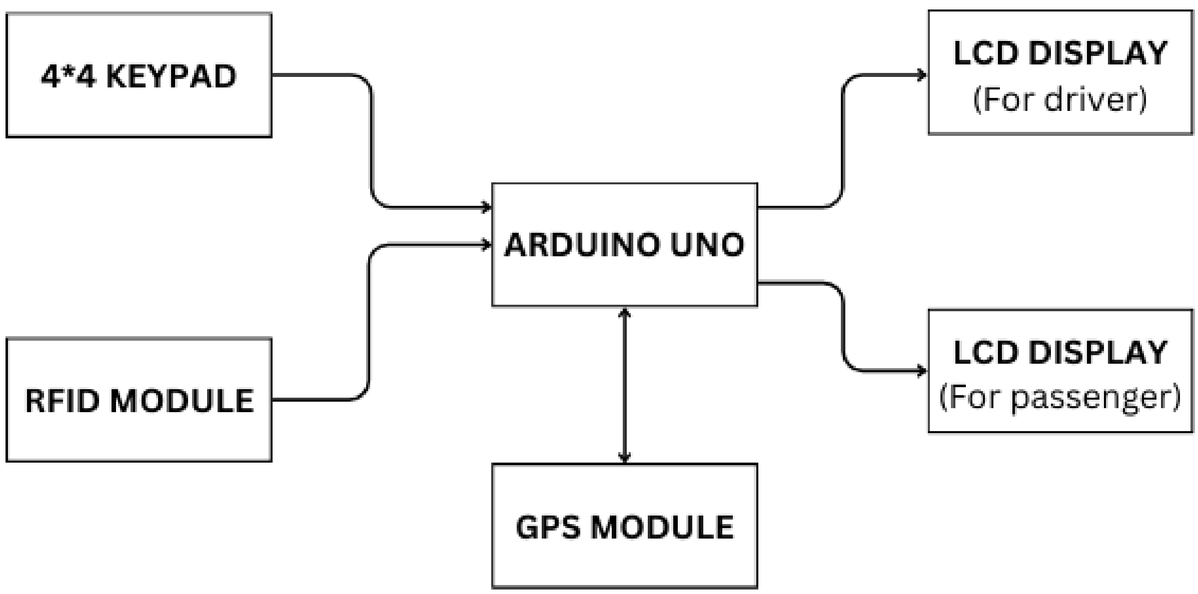

Figure 2.1: Block Diagram

2.2 BLOCK DIAGRAM DESCRIPTION

1. RFID Reader:

- **Functionality**: The RFID reader module serves as the gateway for passenger authentication and fare management within the transit system. Employing advanced radio frequency identification (RFID) technology, it enables seamless interaction between passengers and the fare collection infrastructure(Haibi et al., 2022).

- **Operation**:
  - **Detection and Communication**: Equipped with specialized antennas, the reader swiftly detects the presence of RFID cards presented by passengers. Upon detection, it establishes a secure communication channel with the card, extracting essential information such as unique identifiers and possibly passenger-specific data.
  - **Data Transmission**: Acting as a conduit for data transfer, the reader efficiently transmits the extracted card information to the central processing unit, an Arduino board, via a wired or wireless connection.
  - **Boarding and Fare Deduction**: During boarding, the reader plays a pivotal role in verifying passenger identities and logging entries into the system. Similarly, during exits, it records passenger departures and initiates fare deductions based on predefined fare rules and parameters stored within the system's database.

2. Arduino UNO Board:

- **Role**: Positioned as the nerve center of the transit system, the Arduino board orchestrates the seamless integration and operation of all system components, ensuring efficient data processing, decision-making, and control.
- **Functions**:
  - **Data Aggregation**: Serving as a data hub, the board aggregates incoming signals and information from diverse sources, including the RFID reader, GPS module, keypad, and other peripheral devices.
  - **Processing and Decision-Making**: Leveraging its computational capabilities, the board executes predefined algorithms to interpret data, perform calculations (e.g., fare computation based on distance), and make real-time decisions for passenger management and system operation.
  - **Control and Communication**: Acting as a master controller, the board governs the operation of output devices such as LCD screens, ensuring synchronized dissemination of critical information to passengers and the driver. Furthermore, it facilitates seamless communication with external interfaces for data exchange and system integration purposes.

3. Keypad:

- **Purpose**: Designed to facilitate intuitive user interaction, the keypad empowers passengers to input their stop selections and other relevant information, enhancing the overall user experience and system usability.
- **Operation**:
  - **User Input**: Equipped with tactile keys or buttons, the keypad provides passengers with a familiar interface for selecting their desired stop or performing other interactive actions.
  - **Data Transmission**: Upon receiving input from passengers, the keypad translates the input into digital signals and transmits them to the Arduino board via a wired or wireless connection for further processing.
  - **Feedback and Confirmation**: To ensure user confidence and satisfaction, the keypad provides audible or visual feedback to passengers, confirming the acceptance of their input and facilitating smooth interaction with the system.

4. GPS Module:

- **Function**: Empowered by global navigation satellite systems (GNSS) technology, the GPS module serves as a vital source of real-time positioning data, enabling accurate route monitoring and dynamic fare calculation within the transit system.
- **Operation**:
  - **Satellite Reception**: Utilizing an array of satellites in orbit, the module continuously receives signals from global navigation satellite systems such as GPS, GLONASS, or Galileo, enabling precise determination of the vehicle's current geographical coordinates.
  - **Data Provision**: Transmits location data to the Arduino board, enabling dynamic fare calculation based on actual distance traveled, ensuring fair and accurate fares for passengers.
  - **Integration and Accuracy**: Seamlessly integrated into the system architecture, the GPS module delivers high levels of accuracy and reliability in determining the vehicle's position, thereby enhancing overall system performance and passenger satisfaction(Ramesh et al., 2021).

5. LCD Displays:

- **Purpose**: Serving as the primary means of information dissemination, LCD displays play a crucial role in providing real-time updates and operational feedback to both passengers and the driver, fostering situational awareness and enhancing the overall travel experience.
- **Features**:
    - **Passenger Displays:** Tailored for passengers, these displays provide comprehensive information on stop times, upcoming stops,calculated fares, and other details, ensuring passengers are well-informed throughout their journey.
    - **Driver Displays:** Equipped with features for the driver, these displays offer essential metrics such as passenger counts, system status updates, enabling informed decision-making and efficient vehicle management.

6. Power Supply:

- **Function**: Serving as the lifeblood of the transit system, the power supply ensures continuous and reliable electrical power to all system components, safeguarding against interruptions and maintaining operational efficiency throughout transit operations(Sibikin, 2022).
- **Features**:
    - **Stability and Reliability**: Delivering stable and consistent power to all system components, the power supply mitigates the risk of system malfunctions or downtime, thereby ensuring seamless operation and passenger satisfaction.
    - **Redundancy and Backup**: Incorporating redundant power sources or backup mechanisms, such as uninterruptible power supplies (UPS) or backup generators, the power supply enhances system resilience and reliability, safeguarding against power outages or failures.
    - **Efficiency and Optimization**: Employing energy-efficient principles and power management techniques to optimize power distribution and consumption, reducing energy waste, environmental impact, and prolonging component lifespan

2.3 CIRCUIT DIAGRAM

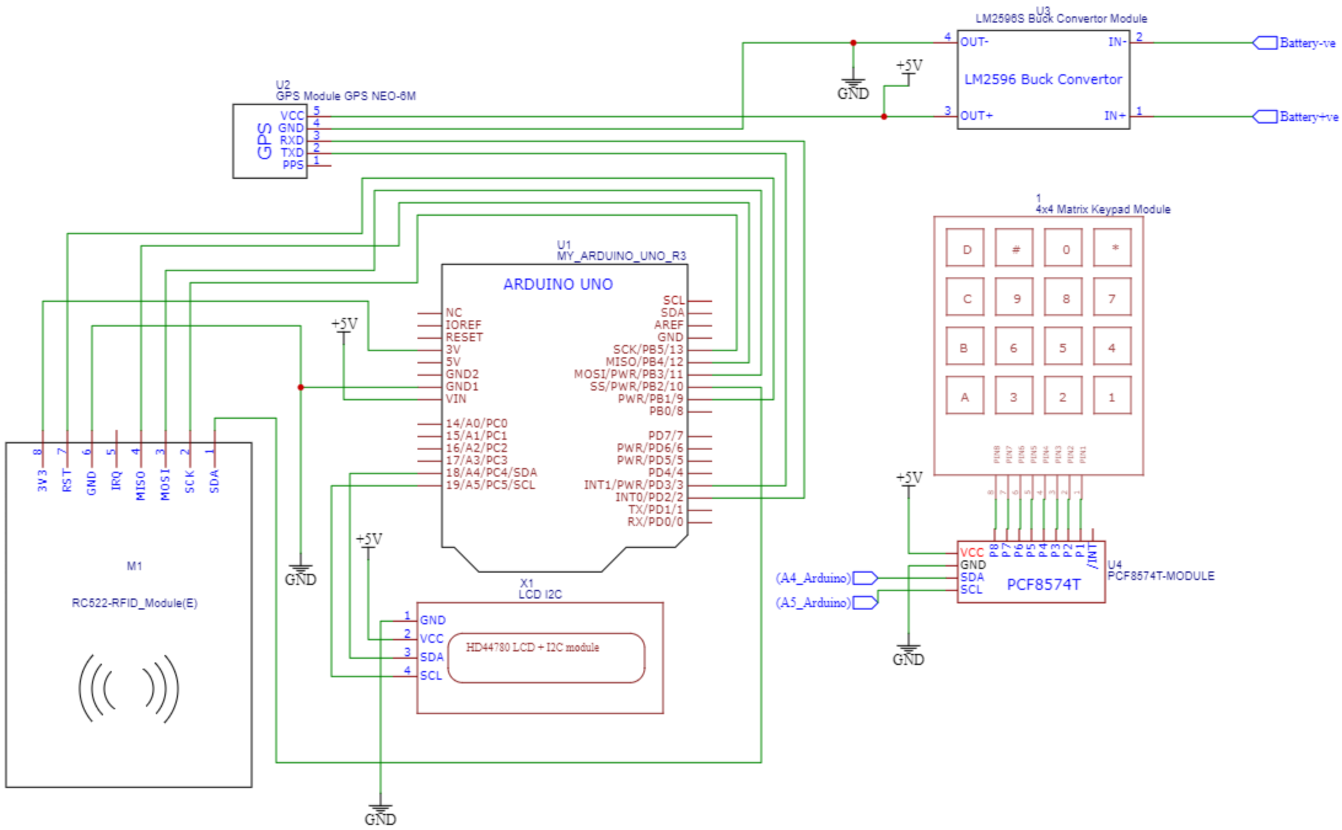

Figure 2.2: Circuit Diagram

2.4 SOURCE CODE

```
#include "I2CKeyPad.h"
#include <SPI.h>
#include <MFRC522.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#define RST_PIN    9     // Configurable, see typical pin layout above
#define SS_PIN     10    // Configurable, see typical pin layout above
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
#define GPS_RX_PIN 2
```

```
#define GPS_TX_PIN 3
TinyGPSPlus gps;
MFRC522 mfrc522(SS_PIN, RST_PIN);   // Create MFRC522 instance
SoftwareSerial gpsSerial(GPS_RX_PIN, GPS_TX_PIN); // RX, TX for GPS
LiquidCrystal_I2C lcd1(0x27, 16, 2);
LiquidCrystal_I2C lcd2(0x26, 16, 2);
const uint8_t KEYPAD_ADDRESS = 0x38;
////////////////////////////////////
int flag1 = 0;
int flag2 = 0;
int flag3 = 0;
char ch;
int s1=0;
int s2=0;
int s3=0;
int Time1=5;
int Time2=5;
int Time3=5;
int fare_p1,fare_p2,fare_p3;

////////////////////////////////////
String latitude;
String longitude;
float clat_p1 = 0.0;
float clong_p1 = 0.0;
float elat_p1 = 0.0;
float elong_p1 = 0.0;
float clat_p2 = 0.0;
float clong_p2 = 0.0;
float elat_p2 = 0.0;
float elong_p2 = 0.0;
float clat_p3 = 0.0;
float clong_p3 = 0.0;
float elat_p3 = 0.0;
float elong_p3 = 0.0;

///////////////////////////////////////
I2CKeyPad keyPad(KEYPAD_ADDRESS);

char keymap[19] = "123A456B789C*0#DNF";  // N = NoKey, F = Fail


void setup()
{
  Serial.begin(9600);
  gpsSerial.begin(9600);
  ////////////////////////////////////////////////////
    lcd1.init();
```

```
  lcd1.backlight();
  lcd1.setCursor(0, 0);
  lcd1.print("Welcome");
  lcd1.setCursor(0, 1);
  lcd1.print("Enter_stp_1,2,3");
  lcd2.init();
  lcd2.backlight();
  lcd2.setCursor(0, 0);
  lcd2.print("Smart bus system");
  delay(1000);
  lcd2.clear();
  lcd2.setCursor(0, 0);
   lcd2.print("Stp1=");
     lcd2.setCursor(8, 0);
   lcd2.print("Stp2=");
     lcd2.setCursor(0, 1);
   lcd2.print("Stp3=");
  ////////////////////////////////////////////////////
  SPI.begin();         // Init SPI bus
  mfrc522.PCD_Init();
  ////////////////////////////////////////////////////
  Serial.println(_FILE_);

  Wire.begin();
  Wire.setClock(400000);
  if (keyPad.begin() == false)
  {
    Serial.println("\nERROR: cannot communicate to keypad.\nPlease
reboot.\n");
    while (1);
  }

  keyPad.loadKeyMap(keymap);
}

void loop()
{
  while (gpsSerial.available() > 0)
    if (gps.encode(gpsSerial.read()))
      {
        if (gps.location.isValid())
  {

    latitude=String(gps.location.lat(), 6);
    longitude=String(gps.location.lng(), 6);
    Serial.println("Lat"+(String)latitude);
    Serial.println("Long"+(String)longitude);
  }
```

```
  else
  {
    //Serial.println("Location: Not Available");
  }
      }

    if (keyPad.isPressed())
  {
   ch = keyPad.getChar();     // note we want the translated char
    int key = keyPad.getLastKey();
//    Serial.print(key);
//    Serial.print(" \t");
//    Serial.println(ch);

    if(ch=='1')
    {
    delay(1000);
    s1=s1+1;


      lcd1.clear();
        lcd1.setCursor(0, 0);
  lcd1.print("Tap card");

      lcd2.clear();
       lcd2.setCursor(0, 0);
   lcd2.print("Stp1="+(String)s1);
     lcd2.setCursor(8, 0);
   lcd2.print("Stp2="+(String)s2);
     lcd2.setCursor(0, 1);
   lcd2.print("Stp3="+(String)s3);


   Serial.println(ch);

    }
    if(ch=='2')
    {
   delay(1000);
      s2=s2+1;
//      Time2=Time2*s2;

       lcd1.clear();
      lcd1.setCursor(0, 0);
  lcd1.print("Tap card");

      lcd2.clear();
       lcd2.setCursor(0, 0);
```

```
  lcd2.print("Stp1="+(String)s1);
    lcd2.setCursor(8, 0);
  lcd2.print("Stp2="+(String)s2);
    lcd2.setCursor(0, 1);
  lcd2.print("Stp3="+(String)s3);


   }
   if(ch=='3')
   {
     delay(1000);
     s3=s3+1;
//       Time3=Time3*s3;
     lcd1.clear();
     lcd1.setCursor(0, 0);
 lcd1.print("Tap card");
     lcd2.clear();
      lcd2.setCursor(0, 0);
  lcd2.print("Stp1="+(String)s1);
    lcd2.setCursor(8, 0);
  lcd2.print("Stp2="+(String)s2);
    lcd2.setCursor(0, 1);
  lcd2.print("Stp3="+(String)s3);

  Serial.println(ch);

   }
 }


 if(! mfrc522.PICC_IsNewCardPresent())return;
 if(! mfrc522.PICC_ReadCardSerial())return;
 //Serial.print("UID :");
 String content="";
 byte letter;
 for(byte i=0;i<mfrc522.uid.size;i++)
 {
   //Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0 ":" ");
   //Serial.print(mfrc522.uid.uidByte[i],HEX);
   content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0 " : " "));
   content.concat(String(mfrc522.uid.uidByte[i],HEX));
 }
 Serial.println();
 content.toUpperCase();

 if((content.substring(1) == "43 5C B6 FA")&&(flag1==0))
 {
   elat_p1=latitude.toFloat();
```

```
    elong_p1=longitude.toFloat();
    if(elat_p1==0&&elong_p1==0)
    {
    elat_p1=10.46336;
  elong_p1=76.394104;
    }
    else
    {
      elat_p1=latitude.toFloat();
    elong_p1=longitude.toFloat();
    }
   lcd1.clear();
   lcd1.setCursor(0, 0);
   lcd1.print("passager_1");
   lcd1.setCursor(0, 1);
   lcd1.print("location_updt");

    delay(2000);
    lcd1.clear();
    lcd1.setCursor(0, 0);
lcd1.print("Welcome");
lcd1.setCursor(0, 1);
lcd1.print("Enter_stp_1,2,3");

    flag1=1;
  ////////////////////////////////////

}

else if((content.substring(1) == "43 5C B6 FA")&&(flag1==1))
{


    clat_p1=latitude.toFloat();
    clong_p1=longitude.toFloat();
    if(clat_p1==0&&clong_p1==0)
    {
    clat_p1=10.739357;
  clong_p1=76.648128;
    }
    else
    {
      clat_p1=latitude.toFloat();
    clong_p1=longitude.toFloat();
    }
    fare_p1=calculateFareAndDisplay_1();
    Time1=Time1*s1;
    Time2=Time2*s2;
```

```
      Time3=Time3*s3;

       lcd1.clear();
//     lcd.setCursor(0, 0);
//     lcd.print("Stop arrived");
     lcd1.setCursor(0, 0);
     lcd1.print("Fare="+(String)fare_p1);
     lcd1.setCursor(9, 0);
     lcd1.print("T1="+(String)Time1);
     lcd1.setCursor(0, 1);
     lcd1.print("T2="+(String)Time2);
     lcd1.setCursor(6, 1);
     lcd1.print("T3="+(String)Time3);
     delay(2000);
     lcd1.clear();
     lcd1.setCursor(0, 0);
  lcd1.print("Welcome");
  lcd1.setCursor(0, 1);
  lcd1.print("Enter_stp_1,2,3");
  if(Time1>=5)
  {
    s1=0;
  }
 if(Time2>=5&&Time1==0)
 {
  s2=0;
 }
 if(Time3>=5&&Time1==0&&Time2==0)
 {
  s3=0;
 }

   flag1=0;
   Time1=5;
   Time2=5;
   Time3=5;

  }
  //}
//  Serial.println(f);

  ///////////////////////////////////////////////////////////////////////
   if((content.substring(1) == "E3 B7 9F C5")&&(flag2==0))
  {
     elat_p2=latitude.toFloat();
      elong_p2=longitude.toFloat();
      if(elat_p2==0&&elong_p2==0)
      {
```

```
      elat_p2=10.46336;
    elong_p2=76.394104;
      }
      else
      {
        elat_p2=latitude.toFloat();
      elong_p2=longitude.toFloat();
      }
     lcd1.clear();
     lcd1.setCursor(0, 0);
     lcd1.print("passager_2");
     lcd1.setCursor(0, 1);
     lcd1.print("location_updt");

       delay(2000);
       lcd1.clear();
     lcd1.setCursor(0, 0);
  lcd1.print("Welcome");
  lcd1.setCursor(0, 1);
  lcd1.print("Enter_stp_1,2,3");


     flag2=1;
  // lcd1.print("enter_stp_1,2,3");

  }

  else if((content.substring(1) == "E3 B7 9F C5")&&(flag2==1))
  {

      clat_p2=latitude.toFloat();
      clong_p2=longitude.toFloat();
      if(clat_p2==0&&clong_p2==0)
      {
      clat_p2=10.773238;
    clong_p2=76.662907;
      }
      else
      {
        clat_p2=latitude.toFloat();
      clong_p2=longitude.toFloat();
      }
      fare_p2=calculateFareAndDisplay_2();
      Time1=Time1*s1;
      Time2=Time2*s2;
      Time3=Time3*s3;
      lcd1.clear();
//     lcd.setCursor(0, 0);
```

```
//     lcd.print("Stop arrived");
     lcd1.setCursor(0, 0);
     lcd1.print("Fare="+(String)fare_p2);
     lcd1.setCursor(9, 0);
     lcd1.print("T1="+(String)Time1);
     lcd1.setCursor(0, 1);
     lcd1.print("T2="+(String)Time2);
     lcd1.setCursor(6, 1);
     lcd1.print("T3="+(String)Time3);
//     lcd1.setCursor(0,1);
//     lcd1.print("Thank You");
     delay(2000);
     lcd1.clear();
     lcd1.setCursor(0, 0);
  lcd1.print("Welcome");
  lcd1.setCursor(0, 1);
  lcd1.print("Enter_stp_1,2,3");

     if(Time1>=5)
  {
    s1=0;
  }
 if(Time2>=5&&Time1==0)
 {
  s2=0;
 }
  if(Time3>=5&&Time1==0&&Time2==0)
  {
    s3=0;
  }
   flag2=0;
   Time1=5;
   Time2=5;
   Time3=5;
  }
 /////////////////////////////////////////////////////////
 if((content.substring(1) == "1A CD A1 EE")&&(flag3==0))
  {

     elat_p3=latitude.toFloat();
      elong_p3=longitude.toFloat();
      if(elat_p3==0&&elong_p3==0)
      {
      elat_p3=10.46336;
    elong_p3=76.394104;
      }
      else
      {
```

```
        elat_p3=latitude.toFloat();
      elong_p3=longitude.toFloat();
      }
     lcd1.clear();
     lcd1.setCursor(0, 0);
     lcd1.print("passager_3");
     lcd1.setCursor(0, 1);
     lcd1.print("location_updt");

      delay(2000);
      lcd1.clear();
      lcd1.setCursor(0, 0);
  lcd1.print("Welcome");
  lcd1.setCursor(0, 1);
  lcd1.print("Enter_stp_1,2,3");
  flag3=1;
    ////////////////////////////////////

  }

  else if((content.substring(1) == "1A CD A1 EE")&&(flag3==1))
  {



      clat_p3=latitude.toFloat();
      clong_p3=longitude.toFloat();
      if(clat_p3==0&&clong_p3==0)
      {
      clat_p3=10.751923;
    clong_p3=76.664836;
      }
      else
      {
        clat_p3=latitude.toFloat();
      clong_p3=longitude.toFloat();
      }

      fare_p3=calculateFareAndDisplay_3();
      Time1=Time1*s1;
      Time2=Time2*s2;
      Time3=Time3*s3;
       lcd1.clear();
//     lcd.setCursor(0, 0);
//     lcd.print("Stop arrived");
     lcd1.setCursor(0, 0);
     lcd1.print("Fare="+(String)fare_p3);
     lcd1.setCursor(9, 0);
```

```
    lcd1.print("T1="+(String)Time1);
    lcd1.setCursor(0, 1);
    lcd1.print("T2="+(String)Time2);
    lcd1.setCursor(6, 1);
    lcd1.print("T3="+(String)Time3);
    delay(2000);
    lcd1.clear();
    lcd1.setCursor(0, 0);
 lcd1.print("Welcome");
 lcd1.setCursor(0, 1);
 lcd1.print("Enter_stp_1,2,3");
 if(Time1>=5)
 {
   s1=0;
 }
if(Time2>=5&&Time1==0)
{
 s2=0;
}
if(Time3>=5&&Time1==0&&Time2==0)
{
 s3=0;
}

  flag3=0;
  Time1=5;
  Time2=5;
  Time3=5;

 }

 if(s1>0&&s2>0&&s3>0)
 {
  lcd2.setCursor(7, 1);
  lcd2.print("Psgr_full");
 }
 else
 {
  lcd2.setCursor(7, 1);
  lcd2.print("          ");
 }

/////////////////////////////////////////////////////////

   //delay(100);
   //lcd2.clear();
    lcd2.setCursor(0, 0);
  lcd2.print("Stp1="+(String)s1);
```

```
      lcd2.setCursor(8, 0);
    lcd2.print("stp2="+(String)s2);
      lcd2.setCursor(0, 1);
    lcd2.print("Stp3="+(String)s3);
    delay(1000);
}
int calculateFareAndDisplay_1()
{

  float lat1_p1 = clat_p1 * PI / 180.0;
  float lon1_p1 = clong_p1 * PI / 180.0;
  float lat2_p1 = elat_p1 * PI / 180.0;
  float lon2_p1 = elong_p1 * PI / 180.0;

  float dlon_p1 = lon2_p1 - lon1_p1;
  float dlat_p1 = lat2_p1 - lat1_p1;
  float a_p1 = pow(sin(dlat_p1 / 2), 2) + cos(lat1_p1) * cos(lat2_p1) *
pow(sin(dlon_p1 / 2), 2);
  float c_p1 = 2 * atan2(sqrt(a_p1), sqrt(1 - a_p1));
  float distance_p1 = 6371 * c_p1;
  Serial.println(distance_p1);
  int f_p1 = 1  * distance_p1;

//      lcd.setCursor(0, 0);
//     lcd.print("           ");


//       lcd.clear();
 return f_p1;
  }

  int calculateFareAndDisplay_2()
{

  float lat1_p2 = clat_p2 * PI / 180.0;
  float lon1_p2 = clong_p2 * PI / 180.0;
  float lat2_p2 = elat_p2 * PI / 180.0;
  float lon2_p2 = elong_p2 * PI / 180.0;

  float dlon_p2 = lon2_p2 - lon1_p2;
  float dlat_p2 = lat2_p2 - lat1_p2;
  float a_p2 = pow(sin(dlat_p2 / 2), 2) + cos(lat1_p2) * cos(lat2_p2) *
pow(sin(dlon_p2 / 2), 2);
  float c_p2 = 2 * atan2(sqrt(a_p2), sqrt(1 - a_p2));
  float distance_p2 = 6371 * c_p2;
  //Serial.println(distance_p2);
  int f_p2 = 1  * distance_p2;
```

```
 return f_p2;
  }

  int calculateFareAndDisplay_3()
{

  float lat1_p3 = clat_p3 * PI / 180.0;
  float lon1_p3 = clong_p3 * PI / 180.0;
  float lat2_p3 = elat_p3 * PI / 180.0;
  float lon2_p3 = elong_p3 * PI / 180.0;

  float dlon_p3 = lon2_p3 - lon1_p3;
  float dlat_p3 = lat2_p3 - lat1_p3;
  float a_p3 = pow(sin(dlat_p3 / 2), 2) + cos(lat1_p3) * cos(lat2_p3) *
pow(sin(dlon_p3 / 2), 2);
  float c_p3 = 2 * atan2(sqrt(a_p3), sqrt(1 - a_p3));
  float distance_p3 = 6371 * c_p3;
  //Serial.println(distance_p3);
  int f_p3 = 1  * distance_p3;


 return f_p3;
  }
```

Chapter 3

COMPONENTS USED AND DESCRIPTION

3.1 ARDUINO UNO

Figure 3.1: Arduino Uno Board

The Arduino Uno is a widely-used microcontroller board based on the ATmega328P microchip. This board is equipped with numerous features that make it ideal for a variety of applications, including embedded systems and prototyping.

Key Features:

- **Digital Input/Output Pins:** The Arduino Uno has 14 digital I/O pins, 6 of which can be used for Pulse Width Modulation (PWM) outputs. These pins allow the board to interface with various digital devices and components.
- **Analog Inputs:** There are 6 analog input pins that can read sensors and convert analog signals to digital data, useful for interfacing with a wide range of analog sensors.
- **Clock Speed:** It operates at a clock speed of 16 MHz, facilitated by a ceramic resonator, providing sufficient processing power for most applications.
- **Connectivity:** The board includes a USB connection for programming and serial communication, a power jack for external power supply, and an ICSP (In-Circuit Serial Programming) header for programming the microcontroller directly.
- **Reset Button:** A reset button is available to restart the microcontroller, which is useful during debugging and testing.

Use in the Project: In this transit system project, the Arduino Uno serves as the central processing unit. It collects and processes data from various inputs such as the RFID reader, keypad, and GPS module. It executes algorithms for fare calculation based on the distance traveled and controls the output to devices like the LCD display. The Arduino Uno ensures all components work together seamlessly, enabling efficient operation of the transit system.

3.2 RFID

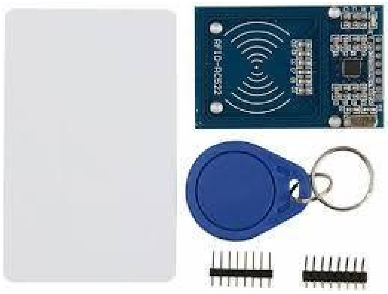

Figure 3.2: RFID Reader and Tag

RFID (Radio Frequency Identification) technology is a form of wireless communication that uses electromagnetic fields to automatically identify and track tags attached to objects. The system comprises two main components: the RFID reader and the RFID tag.

Components:

- **RFID Reader:** The reader generates an electromagnetic field to communicate with the RFID tag. It reads the data encoded in the tag and transmits it to the Arduino or another processing unit.
- **RFID Tag:** Comprising an integrated circuit (IC) and an antenna, the tag stores identifying information. The RFID inlay is the part of the tag that encodes this information. Tags can be passive (powered by the reader's electromagnetic field) or active (with their own power source).

Use in the Project: In this transit system, the RFID reader is used for passenger authentication and fare management. When passengers board, they present their RFID cards to the reader, which identifies them and logs their entry into the system. During exit, the reader again scans the RFID card to log the departure and initiates fare deduction based on the journey details. This system allows for seamless, contactless transactions and efficient tracking of passenger movements.

3.3 16*2 LCD DISPLAY

Figure 3.3: LCD Display

An LCD (Liquid Crystal Display) is a flat-panel display technology commonly used in many devices for displaying information in a clear, readable format.

Key Features:

- **Display Capacity:** The 16x2 LCD display can show 16 characters per line on 2 lines. Each character is displayed using a 5x8 pixel matrix.
- **Customization:** These displays support custom characters, special characters, and simple animations, making them versatile for different applications.
- **Ease of Use:** The displays are easy to program and interface with microcontrollers like the Arduino Uno. They require minimal power and can be driven with a few digital I/O pins.
- **Applications:** Commonly used in devices such as calculators, clocks, various consumer electronics, and project prototypes, due to their low cost and ease of integration.

Use in the Project: In the transit system, the 16x2 LCD display is used to provide real-time information to both passengers and the driver. For passengers, the display shows essential details such as remaining stop times, upcoming stops, and fare information. For the driver, it can display operational data such as passenger counts, system status, and calculated fares. This information dissemination enhances situational awareness and improves the overall travel experience.

3.4 GPS MODULE

Figure 3.4: GPS Module

The GPS (Global Positioning System) module is a device that receives signals from GPS satellites to determine the precise location, speed, and time.

Functionality:

- **Satellite System:** The GPS system consists of a constellation of at least 24 satellites orbiting Earth. These satellites continuously transmit signals that GPS receivers use to calculate their position.
- **Location Calculation:** A GPS module calculates its position by triangulating signals from at least three satellites, while a fourth satellite can provide time correction and enhance accuracy.
- **Components:** Modern GPS modules are compact, integrating tiny processors and antennas to receive satellite data and compute position coordinates.
- **Applications:** GPS modules are essential in navigation systems for vehicles, smartphones, and various tracking devices. They provide reliable location data regardless of weather conditions.

Use in the Project: The GPS module in this transit system provides real-time location data of the vehicle. This information is crucial for dynamically calculating the fare based on the distance traveled. The Arduino Uno uses the GPS data to determine the vehicle's position and integrates this with passenger entry and exit logs to compute accurate fares. Additionally, the GPS data can be used to update passengers and the driver about the current location and expected arrival times at upcoming stops.

Chapter 4

SYSTEM DESIGN AND WORKING

4.1 SYSTEM DESIGN

Hardware Components

1. **RFID Reader:**
    - **Location and Installation:** Installed near the entrance of the bus.
    - **Function:** Reads RFID cards carried by passengers to authenticate their entry into the bus system. The reader detects the unique identifier encoded in the RFID card and sends this data to the Arduino Uno for processing.
2. **4x4 Keypad:**
    - **Location and Installation:** Mounted inside the bus, easily accessible to passengers.
    - **Function:** Allows passengers to select their destination by inputting the corresponding key. The keypad inputs are sent to the Arduino Uno for processing to determine the fare and update the system accordingly.
3. **GPS Module:**
    - **Location and Installation:** Installed in a secure location on the bus.
    - **Function:** Determines the current geographic location of the bus using satellite signals. The GPS data is used to calculate the distance to the passenger's selected destination and to assist in fare calculation.
4. **16x2 LCD Display:**
    - **Location and Installation:** Passenger display positioned inside the bus where passengers can easily view it and Driver display installed on the driver's dashboard.
    - **Function:** The passenger display shows information such as passenger count, fare details, and system messages, providing feedback after RFID card taps and destination selection. The driver display provides essential

operational data, including passenger count, system status, calculated fares, and alerts, helping the driver manage bus operations efficiently and respond to issues.

5. **Arduino Uno Microcontroller:**

- **Location and Installation:** Centrally located within the bus's electronic control system.
- **Function:** Acts as the central processing unit. It collects data from the RFID reader, keypad, GPS module, and other sensors, processes this data, executes fare calculation algorithms, and controls the output to the LCD displays and driver display.

4.2 WORKING

(a) Passenger boarding and stop selection

Upon entering the bus, passengers use keypads to select their intended stop. This input is sent to the Arduino board, where the total count of passengers getting off at each stop is updated. The system displays this information on both Passenger and Driver Displays, providing real-time tracking of passenger destinations. The keypads communicate with the Arduino via the I2C protocol, sending stop selection data that updates internal counters (s1, s2, s3) for each stop.

(b) RFID Card Swiping and GPS Location Tracking

After selecting their stop, passengers swipe their RFID cards, which triggers the system to record their entry GPS location. The RFID reader (MFRC522) reads the unique identifier of each card, and the Arduino captures this data, updating the current GPS coordinates using the TinyGPS++ library. The GPS module (connected via SoftwareSerial on pins 2 and 3) continuously streams location data to the Arduino, which processes and stores valid GPS coordinates.

(c) Real-Time Updates on Driver Display

The Driver Display is updated in real-time with the number of passengers for each stop. The Arduino uses this data to manage stop times effectively, reducing unnecessary delays. The system calculates stop times by multiplying the number of passengers by five seconds. This information is crucial for the driver to maintain an efficient schedule and manage the flow of passengers.

(d) Passenger Exit and Fare Calculation

When passengers disembark, they swipe their RFID cards again, capturing the exit GPS coordinates. The system uses these coordinates to calculate the distance traveled using the Haversine formula, implemented in functions like calculateFareAndDisplay_1(). The calculated distance determines the fare, which is displayed to the passenger on the Passenger Display. The fare calculation process involves converting latitude and longitude values into radians, computing the differences, and applying the Haversine formula to get the distance in kilometers.

4.3 SYSTEM FLOW

1. **Passenger Entry:**
   - **Process:** When a passenger boards the bus, they tap their RFID card on the RFID reader.
   - **Action:** The RFID reader captures the card's unique identifier and sends it to the Arduino Uno. The Arduino logs the entry and updates the passenger count.
2. **Destination Selection:**
   - **Process:** After boarding, the passenger uses the 4x4 keypad to select their destination.
   - **Action:** The passenger inputs their destination using the keypad, and the input is sent to the Arduino Uno. The selected destination is displayed on the LCD screen for confirmation.
3. **Fare Calculation:**
   - **Process:** The system calculates the fare based on the distance from the current location to the selected destination.

    - **Action:** The GPS module provides the current location data, which the Arduino Uno uses along with the destination input to calculate the fare. The fare is then deducted from the passenger's account linked to the RFID card, and the LCD display shows the deducted fare.

4. **Capacity Management:**
    - **Process:** The system monitors the number of passengers on the bus to ensure it does not exceed its maximum capacity.
    - **Action:** The Arduino Uno updates the passenger count each time an RFID card is read. If the bus reaches its maximum capacity, the system displays a message on the LCD screen indicating that the bus is full.
5. **Integration:**
    - **Process:** All hardware components are connected to the Arduino Uno for data exchange and control.
    - **Action:** The Arduino Uno collects data from various sensors, processes it, and controls output to LCD displays and the driver display. It executes algorithms for fare calculation and capacity management, serving as the system's central processing unit.
6. **User Interface:**
    - **Passenger Interface:**
        - **RFID Reader:** Used for tapping RFID cards upon entry.
        - **4x4 Keypad:** Used for selecting the destination.
        - **16x2 LCD Display:** Provides real-time feedback and information on fare deduction, passenger count, and any system messages.
    - **Driver Interface:**
        - **Driver Display:** Shows operational data such as passenger count, system status, calculated fares, and alerts, enabling the driver to manage the bus effectively.

Chapter 5
RESULT AND CONCLUSION

5.1 RESULT

A Smart Bus System incorporating passenger tracking and RFID-based payment has been successfully implemented. proving the viability and effectiveness of the technology. It demonstrated significant improvements in efficiency, convenience, and security within the public transportation sector. Through the utilization of RFID technology, passengers were able to seamlessly access bus services and make payments without the need for physical tickets or cash transactions.

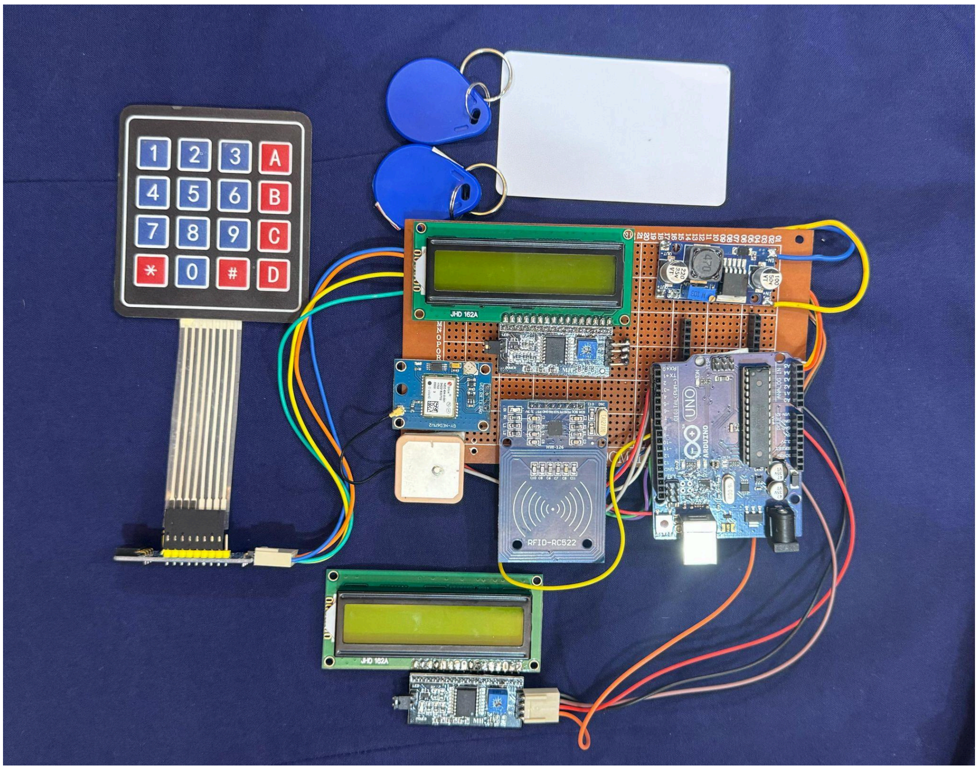

Figure 5.1: Circuit Implementation

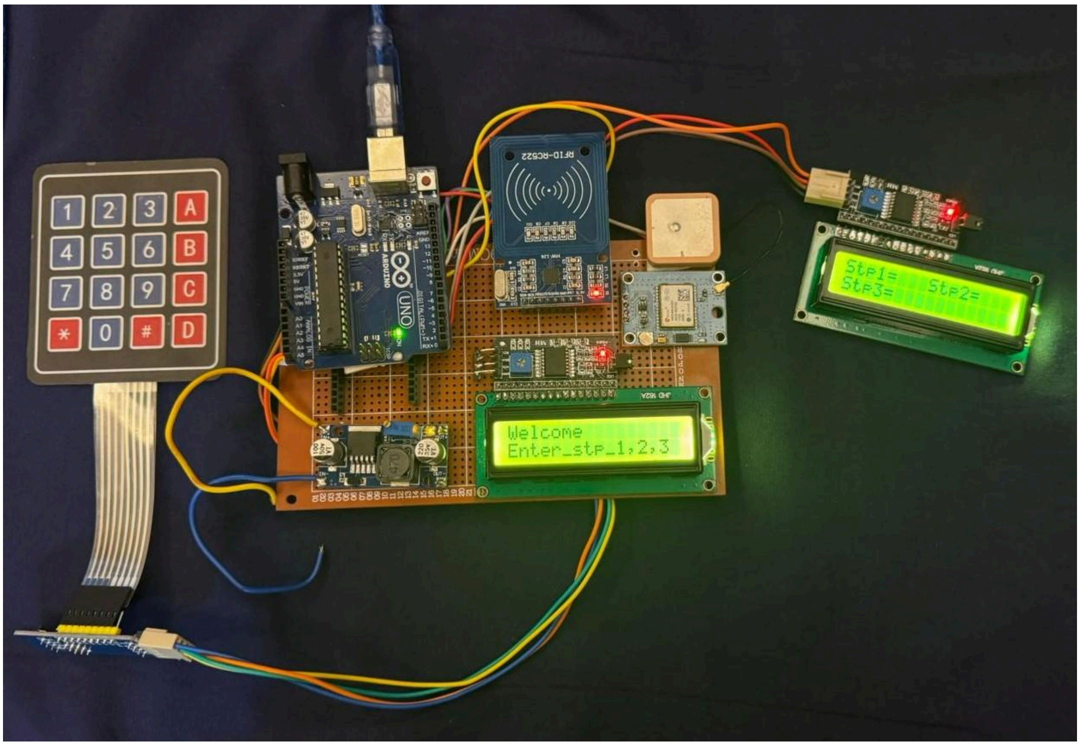

Figure 5.2: Output

5.2 CONCLUSION

The integration of passenger tracking and RFID-based payment systems in our smart bus project not only modernizes public transportation but also lays the foundation for a more connected, efficient, and sustainable urban mobility ecosystem. Through continuous innovation and collaboration, we can further enhance the effectiveness and impact of smart technologies in revolutionizing public transit systems worldwide. As we look ahead, it is imperative to continue exploring novel technologies and collaborative partnerships to further optimize public transit systems, making them more accessible, sustainable, and responsive to the needs of commuters and communities alike.

Chapter 6

MERITS,LIMITATIONS AND FUTURE SCOPES

6.1 MERITS

1. **Efficiency in Operations:** Integrating passenger count control helps optimize bus operations by preventing overloading. This ensures that buses operate within safe capacity limits, improving safety and reducing delays caused by overcrowding.

2. **Reduced Congestion and Emissions**: By preventing overloading and optimizing bus capacity, the project contributes to reducing congestion and carbon emissions associated with public transportation. By encouraging more efficient use of buses, it supports efforts to create a cleaner and more sustainable urban environment.

3. **Scalability and Adaptability:** The system's modular design allows for scalability and adaptability to evolving transportation needs. As demand changes, the system can be easily expanded or modified to accommodate additional buses or routes, ensuring its effectiveness in meeting the needs of growing urban populations.

4. **Improved Revenue Collection:** The RFID-based payment system enables more accurate and efficient revenue collection for transportation authorities. With automated fare calculation and payment deduction, there is reduced risk of fare evasion and greater assurance of collecting the appropriate fare for each

journey, ultimately contributing to improved financial sustainability of the public transportation system.

5. **Compatibility and Integration**: RFID technology is compatible with existing infrastructure and can be seamlessly integrated into the smart bus system without significant modifications. This allows for easy adoption and scalability, ensuring that the payment system can be expanded or upgraded as needed to meet evolving passenger needs and technological advancements.

6.2 LIMITATIONS

1. **Initial Implementation Costs:** Implementing a smart bus system with RFID-based payment and passenger tracking technologies requires a significant upfront investment in hardware, software, and infrastructure. These costs may pose a financial barrier to some transportation authorities or operators, especially those with limited budgets.

2. **Technological Dependence:** The success of the project relies heavily on the reliability and availability of technology. Any disruptions or malfunctions in the RFID payment system, passenger tracking sensors, or communication networks could lead to service interruptions or delays, impacting the overall effectiveness of the system.

3. **Equity Issues:**RFID-based payment systems might not be fair for everyone because some people might not know how to use them or have the technology needed. This could make it harder for certain groups, like those with less money, to use public transportation, making inequalities worse.

4. **Maintenance and Support:** Smart bus systems require ongoing maintenance, upgrades, and technical support to ensure optimal performance and reliability. Adequate resources and expertise must be allocated for routine maintenance and troubleshooting to address any issues that may arise during operation.

6.3 FUTURE SCOPES

1. **Mobile Payment Options:** Introducing payment methods using smartphones or other devices for bus fares.

2. **Advanced Passenger Assistance:** Implementing features like real-time bus tracking and seat reservation systems.

3. **Accessibility Improvements:** Making the system more user-friendly for people with disabilities or those unfamiliar with technology.

4. **Keeping the Air Clean:** Thinking about ways to use buses that don't cause pollution, like using electric buses or fuels that are better for the environment

7.REFERENCES

1. Smart Bus System Sudhir N. Divekar , Sagar R. Patil, Satish A. Shelke HSBPVT'S COE, Kashti, SPPU, Maharashtra, India

2. Smart Bus System Using IoT Prof. V. B. Kumbhar1, Prof. Mrs. Y.V. Sawant 2, Pranav Kavathekar3, Tanmay Joshi4 Asst. Professor, Department of Electronics, DKTE Society's Textile and Engineering Institute Ichalkaranji, Maharashtra, India1,2 UG Students, Department of Electronics DKTE Society's Textile and Engineering Institute Ichalkaranji, Maharashtra, India3,

3. Smart Bus Tracking System (Keerthana. M1, Rama.B2, Priyadharshini.M3, Vijayalakshmi.S4) Dept.of Information Technology M.A.M. College of Engineering and Technology, Tiruchirappalli, India.

4. Gantimurova, J., & Ermolina, V. (2023). TECHNICAL AND ECONOMIC ASSESSMENT OF THE INTRODUCTION OF ELECTRONIC FARE COLLECTION AS A SUBSYSTEM OF INTELLIGENT TRANSPORT SYSTEM. Scientific Papers Collection of the Angarsk State Technical University.

5. Liu, Y., Lyu, Q., Zhang, J., Suo, X., Hong, C., Lin, L., & Wang, D. (2023). APC: Automatic Passenger Counting and Seat Occupancy Detection for Sightseeing Tram based on 3D LiDAR and Camera. 2023 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM).

6. RFID-based Automated Toll Collection System, authored by M. Holia and Z. Shah and M. Solanki, was published in 2019. falls under volume 9. It can be found on pages 106-110.

7. Ramesh, G., Sivaraman, K., Subramani, V., Vignesh, P., Bhogachari, S. V. V., & ... Farhani, S. (2021). Farm Animal Location Tracking System Using

Arduino and GPS Module. 2021 International Conference on Computer Communication and Informatics (ICCCI), 1-4

8. Haibi, A., Oufaska, K., Yassini, K. E., Boulmalf, M., & Bouya, M. (2022). Systematic Mapping Study on RFID Technology. IEEE Access, 10, 6363-6380.
9. Sibikin, Y. (2022). Power supply.