PSG COLLEGE OF TECHNOLOGY

(AUTONOMOUS INSTITUTION)

COIMBATORE – 641004

BRANCH: ELECTRONICS AND COMMUNICATION ENGINEERING

19L503 – MICROPROCESSORS AND MICROCONTROLLERS

(BATCH – 8)

TOPIC: BUS ADVANCEMENT SYSTEM

BY:

20L147 – SURESHKUMAR R

20L151 – THARUN BALAJI M N

21L401 – AKASH R

BUS ADVANCEMENT SYSTEM

AIM:

To track the bus location of the system using Arduino and GPS and displaying consecutive stops route of the bus.

SOFTWARE USED:

ARDUINO IDE 2.0.1

PROTEUS 8.13

ARDUINO, GPS AND LCD LIBRARIES

OBJECTIVE:

As the advancement of technology in the current era, the automation has been brought into the scope improvement and implementation, we especially look into the bus advancement and automation.

The current bus passengers has the information about their bus departure, bus starting and more importantly of their timing brought by their bus conductor when they are traveling in the bus. As the future has important factor of automation, we brought an idea and new scope of bus advancement simulation.

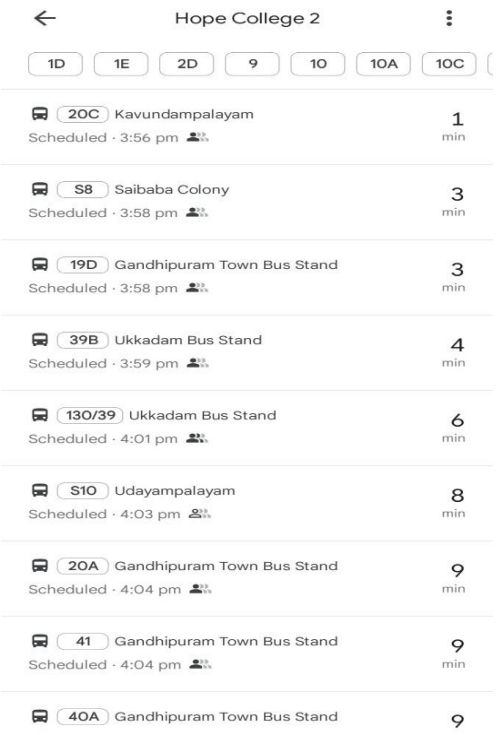
By using the Arduino and GPS with help of longitudinal and latitude location we track the location of the bus and further display by LCD to passenger, without any external factors they get to know their location, next stops and previous destination. We bring you the general prototype simulation of Bus tracking advancement system.

EXISTING SYSTEM:

The current system has major human errors even though there google maps and many other current useful system. There may be time delay between real arrival of bus timings compared to estimated time in the maps, since they are scheduled.

In practical situations, we may don’t know the inbetween stoppings of the bus and may depart at wrong places. To avoid these both confusions, we have proposed a system to solve these issues.

Below figure shows the scheduled timings of google maps which has a less accuracy since bus may arrive at different timings due to traffic. To solve this we can use gsm module to in this same project as a future scope also.



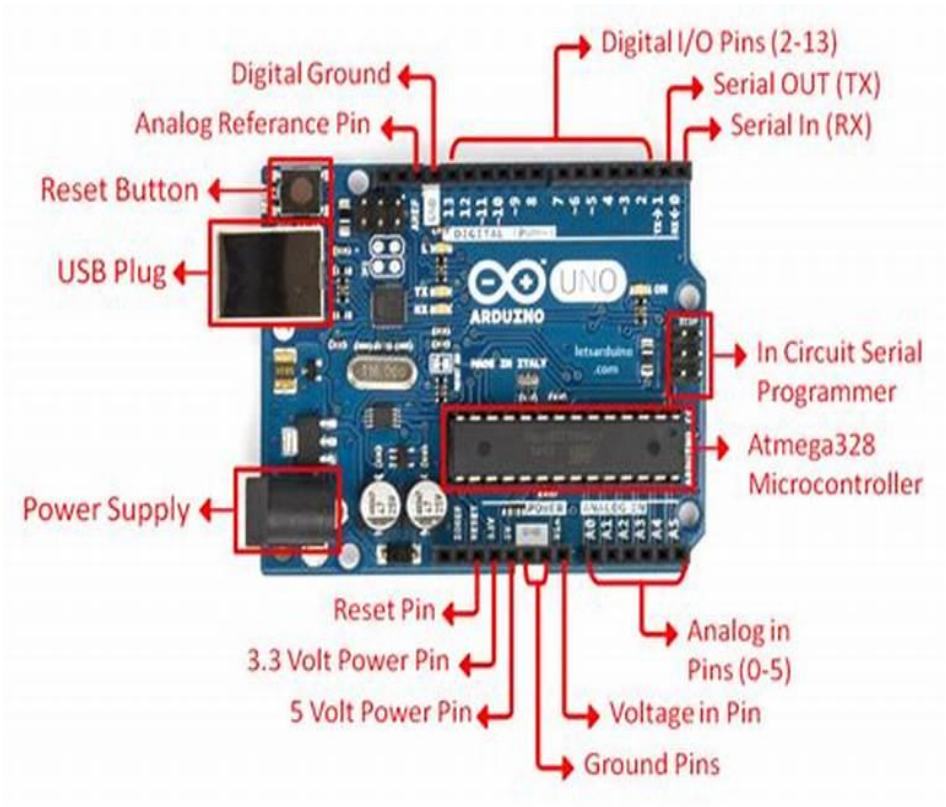
PROPOSED SYSTEM :

We are using GPS to transfer the data to Arduino and we are determining the next stopping of the bus by using that data and displaying the next stop in LCD.

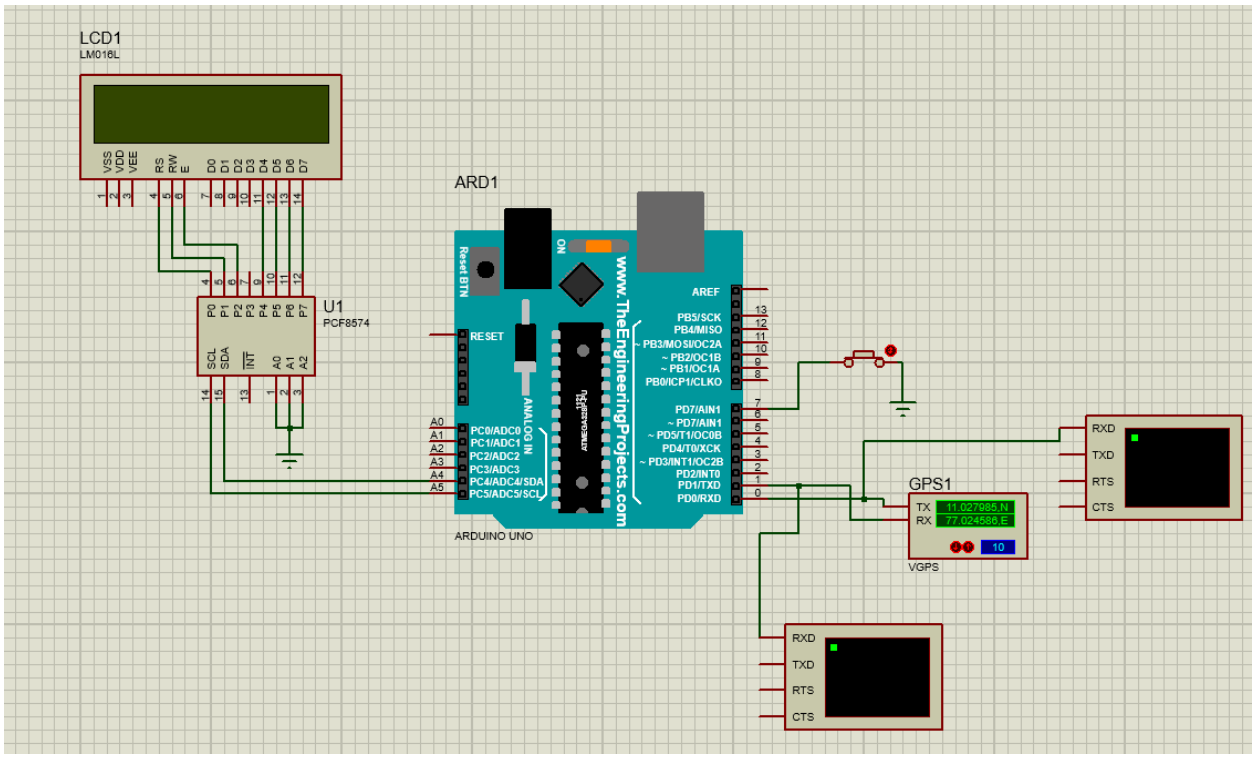
We have included the simulation circuit, code and the results of the code to show how our proposed system works.

We have also included the database we have used in this project.

PIN DIAGRAM OF ARDUINO UNO:



SIMULATION CIRCUIT:



DATASHEET:

We are taken a bus which is going from Karumathampatti to Peelamedu and returning from Peelamedu to Karumathampatti. We have also included 9 more stops along with Karumathampatti and Peelamedu.

PLACE	**LAT, LON**	**NEXT STOP**
KARUMATHAMPATTI	11.107241511785308, 77.17684512614439	KANIYUR
KANIYUR	11.095331344893252, 77.15186782349558	NEELAMBUR
NEELAMBUR	11.060802628384506, 77.08557953883675	CHINNAYAMPALAYAM
CHINNAYAMPALAYAM	11.054939068579541, 77.06509628047094	KMCH
KMCH	11.040500891509737, 77.04100392801953	SITRA
SITRA	11.038571800529647, 77.03824228413662	ARAVIND
ARAVIND	11.034941681420454, 77.03396671217283	CIT
CIT	11.028023073994838, 77.02456334813105	HOPES
HOPES	11.025930839812604, 77.01900756349369	PSG HOSPITAL
PSG HOSPITAL	11.024555131303154, 77.00847764867117	PEELAMEDU
PEELAMEDU	11.022826094673313, 77.0011435018567	RETURN

We have used character array inside the code to demonstrate this database. This is the database we have used to provided the particular stop’s latitude and longitude to Arduino and to compare them through the realtime gps data.

CODE:

```
#include <TinyGPS++.h>
#include <Wire.h>
#include <math.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x20,16,2);
TinyGPSPlus gps;
void getgps(TinyGPSPlus &gps);
float la,lo,t1,t2;
long int lt,lg;
int i;
#define button1 7
bool button_State;
char
p[12][50]={"KARUMATHAMPATTI","KANIYUR","NEELAMBUR","C
HINNAYAMPALAYAM","KMCH","SITRA","ARAVIND","CIT","HOPE
S","PSG HOSPITAL","PEELAMEDU"};


void setup() {
  // put your setup code here, to run once:
  pinMode(button1, INPUT_PULLUP);
  Serial.begin(9600);
  lcd.begin(16,2);
  lcd.init();                        // initialize the
lcd
  lcd.init();
  // Print a message to the LCD.
  lcd.backlight();
```

```
}
void getgps(TinyGPSPlus &gps)
{
  if  (gps.location.isValid()  and  gps.date.isValid()
and gps.time.isValid())
  {
    la=gps.location.lat();
    lo=gps.location.lng();
    Serial.print("Latitude: ");
    Serial.println(la,20);
    Serial.print("Longitude: ");
    Serial.println(lo);
    t1=round(la*1000);
    lt=(long int) t1;
    t2=round(lo*1000);
    lg=(long int) t2;
    if(lt==11107 and lg==77177)
    {
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("TRIP STARTS");
      delay(5000);
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("NEXT STOP:");
      lcd.setCursor(0,1);
      lcd.print("KANIYUR");
```

```
  delay(2000);
}
else if(lt==11095 and lg==77152)
{
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("NEXT STOP:");
  lcd.setCursor(0,1);
  lcd.print(p[i+2]);
  delay(2000);
}
else if(lt==11061 and lg==77086)
{
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("NEXT STOP:");
  lcd.setCursor(0,1);
  lcd.print(p[i+3]);
  delay(2000);
}
else if(lt==11055 and lg==77065)
{
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("NEXT STOP:");
  lcd.setCursor(0,1);
  lcd.print(p[i+4]);
```

```
  delay(2000);
}
else if(lt==11040 and lg==77041)
{
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("NEXT STOP:");
  lcd.setCursor(0,1);
  lcd.print(p[i+5]);
  delay(2000);
}
else if(lt==11039 and lg==77038)
{
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("NEXT STOP:");
  lcd.setCursor(0,1);
  lcd.print(p[i+6]);
  delay(2000);
}
else if(lt==11035 and lg==77034)
{
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("NEXT STOP:");
  lcd.setCursor(0,1);
  lcd.print(p[i+7]);
```

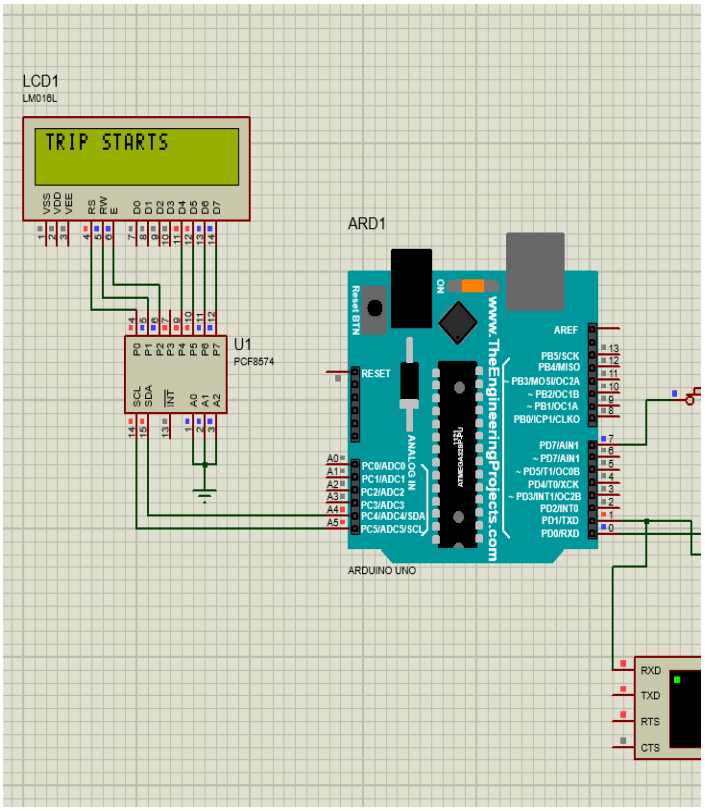
```
  delay(2000);
}
else if(lt==11028 and lg==77025)
{
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("NEXT STOP:");
  lcd.setCursor(0,1);
  lcd.print(p[i+8]);
  delay(2000);
}
else if(lt==11026 and lg==77019)
{
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("NEXT STOP:");
  lcd.setCursor(0,1);
  lcd.print(p[i+9]);
  delay(2000);
}
else if(lt==11025 and lg==77008)
{
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("NEXT STOP:");
  lcd.setCursor(0,1);
  lcd.print(p[i+10]);
```

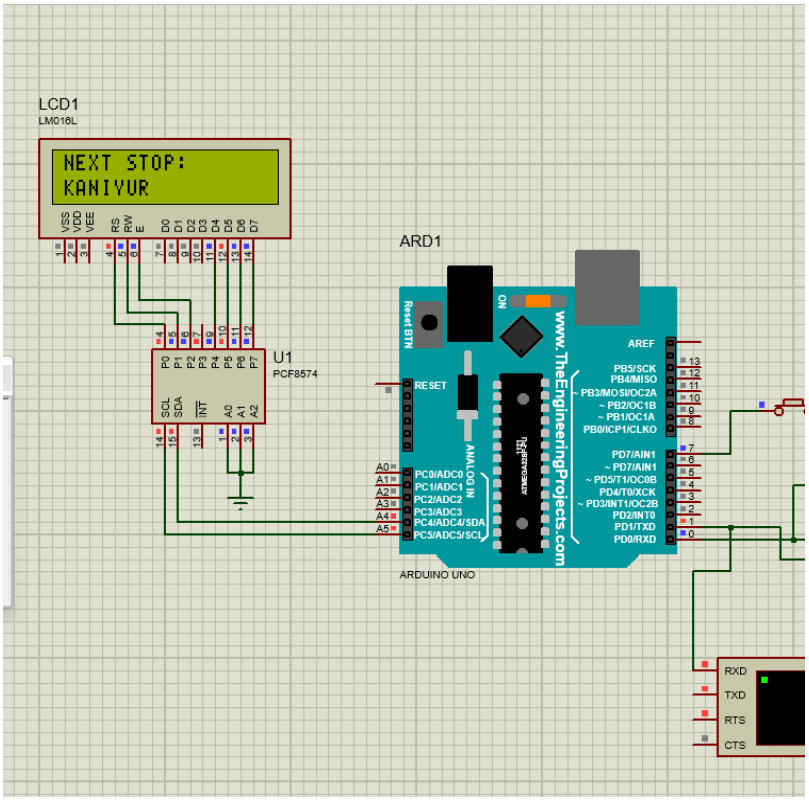
```
      delay(2000);
    }
    else if(lt==11023 and lg==77001)
    {
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("RETURN TRIP");
      delay(5000);
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("NEXT STOP:");
      lcd.setCursor(0,1);
      lcd.print("PSG HOSPITAL");
      delay(2000);
    }
    else
    {
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print(lt);
      lcd.setCursor(0,1);
      lcd.print(lg);

    }
  }
}
void loop() {
```

```
  // put your main code here, to run repeatedly:
  button_State  =  digitalRead(button1);     //We  are
constantly reading the button State


  i=0;
  byte a;
  if (Serial.available() > 0 )
  {
    a = Serial.read();
    if (gps.encode(a))
    {
      if (button_State == LOW)          //PRESSED
      {
        Serial.println("TRIP STARTS");
        i=0;
        delay(200);
      }
      else
      {
        Serial.println("RETURN TRIP");
        i=-2;
        delay(200);
      }
      getgps(gps);
    }
  }
}
```

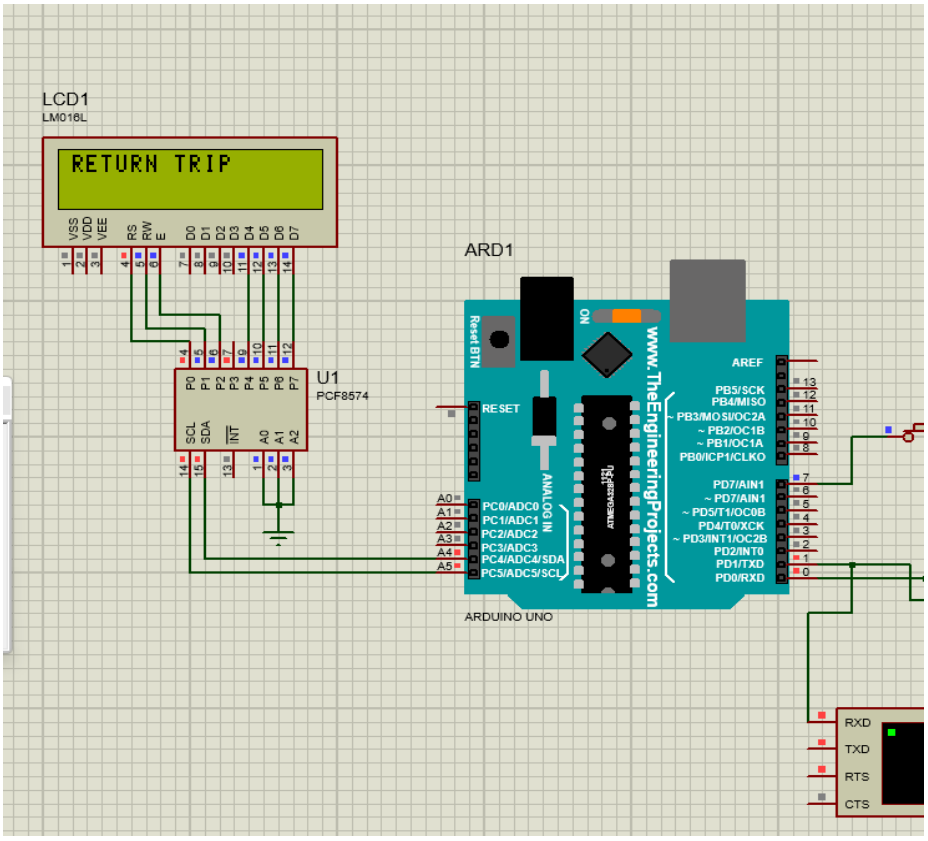
OUTPUT:

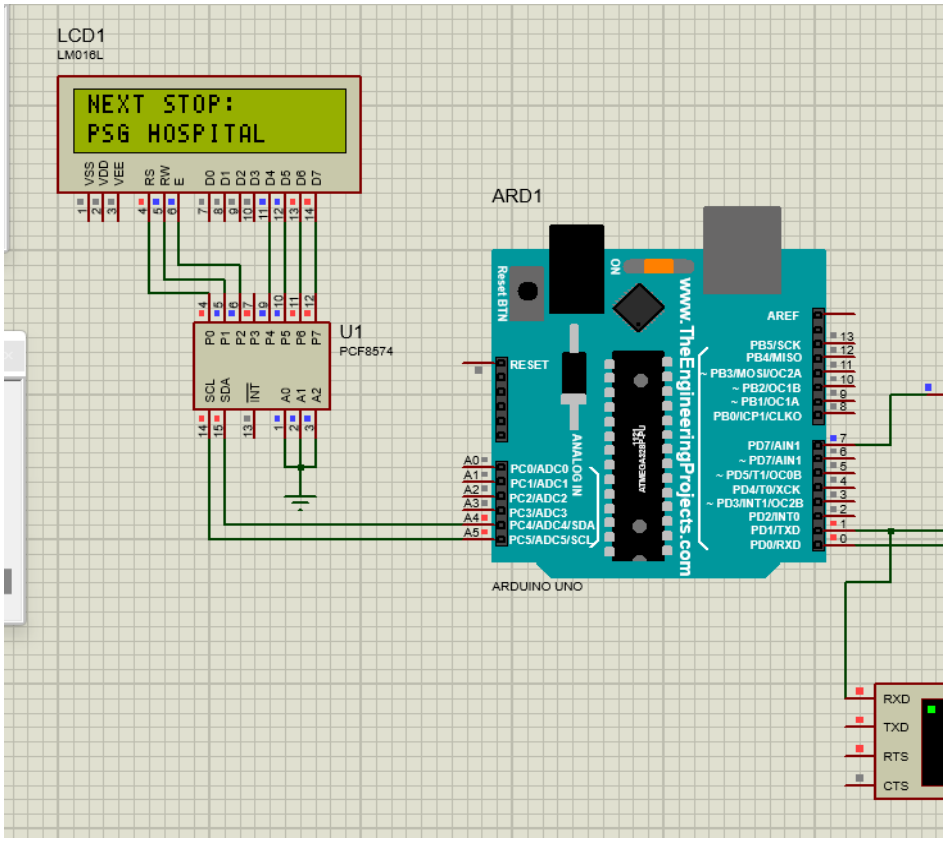
STARTING FROM KARUMATHAMPATTI:



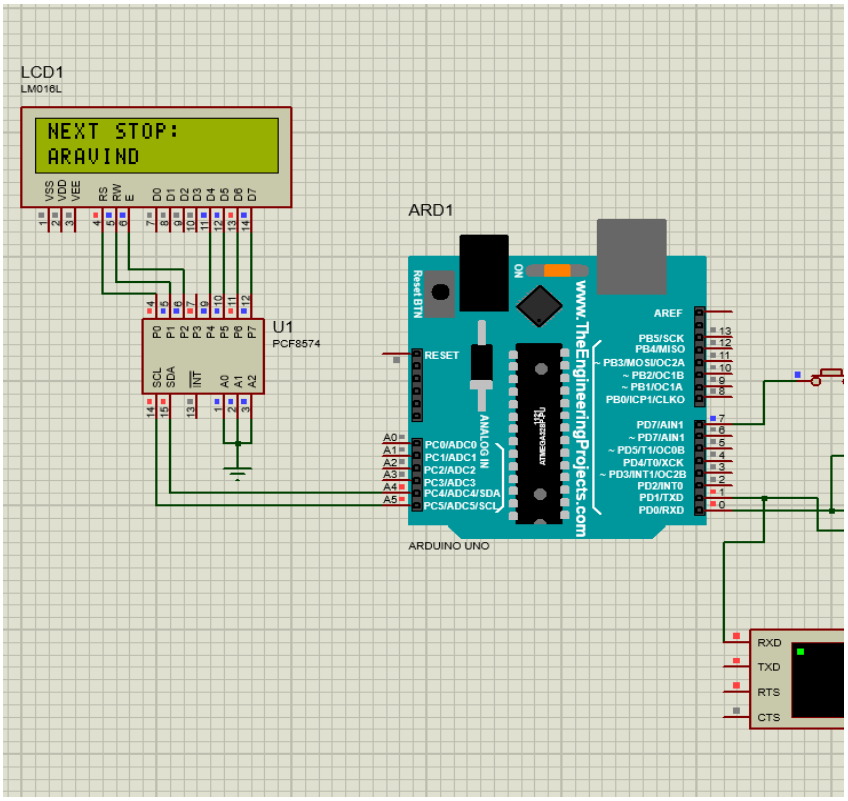


RETURNING FROM PEELAMEDU:

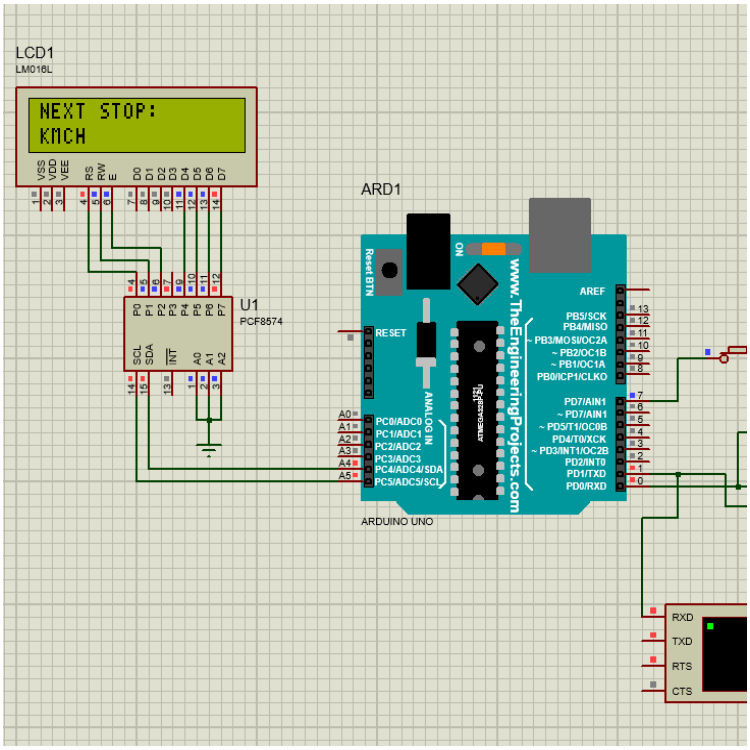




DISPLAY AFTER SITRA WHILE COMING FROM KARUMATHAMPATTI TO PEELAMEDU:



DISPLAY AFTER SITRA WHILE COMING FROM PEELAMEDU TO KARUMATHAMPATTI:



CONCLUSION:

We have made the circuit and simulation with minimal database to show that it will be helpful in the future bus automation industry. To implement it in a practical situation, we have to take atleast 20 to 30 latititude and longitude points in the same region of bus stop to get more accuracy of data. These information can be further transmitted using gsm module for developing app or website regarding realtime bus tracking.