```
#include <SPI.h>
#include <MFRC522.h>
#include <String.h>
#include <TinyGPS++.h>
#include <LiquidCrystal.h> // include the LCD library

LiquidCrystal lcd(PB12, PB13, PB14, PB15, PA8, PA9); //Initialize the LCD


#define SS_PIN PA4
#define RST_PIN PB0

int card1_flag = 0;
int card2_flag = 0;

String data = "\0";
int amount    = 1000;
int amount1    = 1000;

//.....................................
static const uint32_t GPSBaud = 9600;

float sped = 0.0;
double a,b,c,d;

TinyGPSPlus gps;
#define ss Serial1
#define SIM900 Serial2
//=========================================================
/*
void displayInfo()
{
   Serial.print(F("Location: "));
   if (gps.location.isValid())
   {
      Serial.print(gps.location.lat(), 6);
      Serial.print(F(","));
      Serial.print(gps.location.lng(), 6);
   }
   else
   {
      Serial.print(F("INVALID"));
   }

   Serial.print(F("    Date/Time: "));
   if (gps.date.isValid())
   {
      Serial.print(gps.date.month());
      Serial.print(F("/"));
      Serial.print(gps.date.day());
      Serial.print(F("/"));
      Serial.print(gps.date.year());
   }
   else
   {
```

```
      Serial.print(F("INVALID"));
   }

   Serial.print(F(" "));
   if (gps.time.isValid())
   {
      if (gps.time.hour() < 10) Serial.print(F("0"));
      Serial.print(gps.time.hour());
      Serial.print(F(":"));
      if (gps.time.minute() < 10) Serial.print(F("0"));
      Serial.print(gps.time.minute());
      Serial.print(F(":"));
      if (gps.time.second() < 10) Serial.print(F("0"));
      Serial.print(gps.time.second());
      Serial.print(F("."));
      if (gps.time.centisecond() < 10) Serial.print(F("0"));
      Serial.print(gps.time.centisecond());
   }
   else
   {
      Serial.print(F("INVALID"));
   }

   Serial.println();
}
*/
/*
static void printFloat(float val, bool valid, int len, int prec)
{
   if (!valid)
   {
      while (len-- > 1)
         Serial.print('*');
      Serial.print(' ');
   }
   else
   {
      Serial.print(val, prec);
      int vi = abs((int)val);
      int flen = prec + (val < 0.0 ? 2 : 1); // . and -
      flen += vi >= 1000 ? 4 : vi >= 100 ? 3 : vi >= 10 ? 2 : 1;
      for (int i=flen; i<len; ++i)
         Serial.print(' ');
   }
   smartDelay(0);
}

static void printInt(unsigned long val, bool valid, int len)
{
   char sz[32] = "******************";
   if (valid)
      sprintf(sz, "%ld", val);
   sz[len] = 0;
   for (int i = strlen(sz); i < len; ++i)
      sz[i] = ' ';
   if (len > 0)
```

```
      sz[len - 1] = ' ';
   Serial.print(sz);
   smartDelay(0);
}
static void smartDelay(unsigned long ms)
{
   unsigned long start = millis();
   do
   {
      while (ss.available())
         gps.encode(ss.read());
   } while (millis() - start < ms);
}
*/

//=========================================================

//.................................x


MFRC522 rfid(SS_PIN, RST_PIN); // Instance of the class

MFRC522::MIFARE_Key key;
// Init array that will store new NUID
byte nuidPICC[4];

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

void printDec(byte *buffer, byte bufferSize) {
   Serial.println(".....");
   for (byte i = 0; i < bufferSize; i++) {
      Serial.print(buffer[i] < 0x10 ? " 0" : " ");
      Serial.print(buffer[i], DEC);
      data += String(buffer[i] < 0x10 ? " 0" : " ");
      data += String(buffer[i], DEC);
   }
}

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



void setup() {
   // put your setup code here, to run once:
   Serial.begin(9600);
   ss.begin(GPSBaud);
SIM900.begin(9600);
lcd.begin(16, 2);

  lcd.clear();
   lcd.setCursor(0, 0);
   lcd.print("BUS TICKETING SM");
   lcd.setCursor(0, 1);
   lcd.print("...............");
   delay(2000);
```

```
   SPI.begin(); // Init SPI bus
   rfid.PCD_Init(); // Init MFRC522

}

void loop() {
   // put your main code here, to run repeatedly:
   //-----------------------------------
   while (ss.available() > 0)
      if (gps.encode(ss.read()))
         //displayInfo();
         /*
if (gps.speed.isValid())
{
  sped = gps.speed.kmph();
   Serial.print("speed = "); Serial.println(sped);
}
*/
   //--------------------------------------
  lcd.clear();
   lcd.setCursor(0, 0);
   lcd.print("please show card        ");
   lcd.setCursor(0, 1);
   lcd.print("");
   delay(20);


   if ( ! rfid.PICC_IsNewCardPresent())
      return;

   // Verify if the NUID has been readed
   if ( ! rfid.PICC_ReadCardSerial())
      return;

   Serial.println();
   Serial.print(F("In dec: "));
   printDec(rfid.uid.uidByte, rfid.uid.size);
   Serial.println();


   if (data.indexOf("128 48 220 164") != (-1))
   {
       lcd.clear();
   lcd.setCursor(0, 0);
   lcd.print(" CARD-1 MATCHED ");
   lcd.setCursor(0, 1);
   lcd.print("");
   delay(1300);

      Serial.println("card matched");
      data = "\0";
      if (card1_flag == 0)
      {

         if (gps.location.isValid())
         {
```

```
            card1_flag = 1;
              a = gps.location.lat();
              b = gps.location.lng();
            Serial.println("a = " + String(a, 6));
            Serial.println("b = " + String(b, 6));
                      lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("PERSON-1 JOURNEY");
    lcd.setCursor(0, 1);
    lcd.print("      STARTED AT      ");
    delay(1200);
              lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("LAT:"+String(a,6));
    lcd.setCursor(0, 1);
    lcd.print("LNG:"+String(b,6));
    delay(1200);

            sms ("PERSON-1 STARTED JOURNEY \nAT\nLAT:"+String(a,6)+"\nLONG:"+String(b,6));

          }
          else
          {
            Serial.println("gps not working.....");

              lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("INVALID LOCATION");
    lcd.setCursor(0, 1);
    lcd.print("PLEASE CHECK.....");
    delay(1200);

          }
        }//card1_flag== 0
        else
        {
          if (gps.location.isValid())
          {
            card1_flag = 0;
            /*
            float c = gps.location.lat();
            float d = gps.location.lng();
            Serial.println("c = " + String(c, 6));
            Serial.println("d = " + String(d, 6));
            */
            unsigned long distance =

            (unsigned long)TinyGPSPlus::distanceBetween(
          gps.location.lat(),
          gps.location.lng(),
          a,
          b) ;
          //b) / 1000;
    Serial.print("dist = "); Serial.println(distance);
Serial.println("");
amount = amount - (distance *5);
```

```
              lcd.clear();
   lcd.setCursor(0, 0);
   lcd.print("PERSON-1 JOURNEY");
   lcd.setCursor(0, 1);
   lcd.print("       ENDED AT       ");
   delay(1800);

         lcd.clear();
   lcd.setCursor(0, 0);
   lcd.print("LAT:");
   lcd.setCursor(4, 0);
   lcd.print(gps.location.lat(),6);
   lcd.setCursor(0, 1);
   lcd.print("LNG:");
   lcd.setCursor(4, 1);
   lcd.print(gps.location.lng(),6);
   delay(1800);

              lcd.clear();
   lcd.setCursor(0, 0);
   lcd.print("Trav Dist:"+String(distance));
   lcd.setCursor(0, 1);
   lcd.print("Rem Bal:"+String(amount));
   delay(1800);

sms ("PERSON-1 JOURNEY ENDED    \nAT\nLAT:"
+String(gps.location.lat(),6)+"\nLONG:"+String(gps.location.lng(),6)
+"\nAND Travelled Distance of "+String(distance)+"m"
+"\nRemaing Bal:"+String(amount));

          }
          else
          {
             Serial.println("gps not working.....");

              lcd.clear();
   lcd.setCursor(0, 0);
   lcd.print("INVALID LOCATION");
   lcd.setCursor(0, 1);
   lcd.print("PLEASE CHECK.....");
   delay(1200);
         }

       }// card1_flag == 1
       data = "\0";

   }//1st card
   else if (data.indexOf("29 177 111 133") != (-1))
   {
       lcd.clear();
   lcd.setCursor(0, 0);
   lcd.print(" CARD-2 MATCHED ");
   lcd.setCursor(0, 1);
   lcd.print("");
   delay(1300);
```

```
   Serial.println("card-2 matched");
   data = "\0";
   if (card2_flag == 0)
   {

      if (gps.location.isValid())
      {
         card2_flag = 1;
          c = gps.location.lat();
          d = gps.location.lng();
         Serial.println("c = " + String(c, 6));
         Serial.println("d = " + String(d, 6));
                  lcd.clear();
lcd.setCursor(0, 0);
lcd.print("PERSON-2 JOURNEY");
lcd.setCursor(0, 1);
lcd.print("      STARTED AT      ");
delay(1200);
           lcd.clear();
lcd.setCursor(0, 0);
lcd.print("LAT:"+String(c,6));
lcd.setCursor(0, 1);
lcd.print("LNG:"+String(d,6));
delay(1200);

         sms1 ("PERSON-2 STARTED JOURNEY \nAT\nLAT:"+String(c,6)+"\nLONG:"+String(d,6));

      }
      else
      {
         Serial.println("gps not working.....");

           lcd.clear();
lcd.setCursor(0, 0);
lcd.print("INVALID LOCATION");
lcd.setCursor(0, 1);
lcd.print("PLEASE CHECK.....");
delay(1200);

      }
   }//card2_flag== 0
   else
   {
      if (gps.location.isValid())
      {
         card2_flag = 0;
         /*
         float c = gps.location.lat();
         float d = gps.location.lng();
         Serial.println("c = " + String(c, 6));
         Serial.println("d = " + String(d, 6));
         */
         unsigned long distance1 =

         (unsigned long)TinyGPSPlus::distanceBetween(
```

```
          gps.location.lat(),
          gps.location.lng(),
          c,
          d) ;
          //b) / 1000;
   Serial.print("dist1 = "); Serial.println(distance1);
Serial.println("");
amount1 = amount1 - (distance1 *5);

               lcd.clear();
   lcd.setCursor(0, 0);
   lcd.print("PERSON-2 JOURNEY");
   lcd.setCursor(0, 1);
   lcd.print("       ENDED AT       ");
   delay(1800);

         lcd.clear();
   lcd.setCursor(0, 0);
   lcd.print("LAT:");
   lcd.setCursor(4, 0);
   lcd.print(gps.location.lat(),6);
   lcd.setCursor(0, 1);
   lcd.print("LNG:");
   lcd.setCursor(4, 1);
   lcd.print(gps.location.lng(),6);
   delay(1800);

               lcd.clear();
   lcd.setCursor(0, 0);
   lcd.print("Trav Dist:"+String(distance1));
   lcd.setCursor(0, 1);
   lcd.print("Rem Bal:"+String(amount1));
   delay(1800);

sms1 ("PERSON-2 JOURNEY ENDED    \nAT\nLAT:"
+String(gps.location.lat(),6)+"\nLONG:"+String(gps.location.lng(),6)
+"\nAND Travelled Distance of "+String(distance1)+"m"
+"\nRemaing Bal:"+String(amount1));

          }
          else
          {
             Serial.println("gps not working.....");

               lcd.clear();
   lcd.setCursor(0, 0);
   lcd.print("INVALID LOCATION");
   lcd.setCursor(0, 1);
   lcd.print("PLEASE CHECK.....");
   delay(1200);
          }

       }// card1_flag == 1
       data = "\0";

   }//2nd card
```

```
   else
   {
      Serial.println("card not matched");
      delay(2000);
   }

   if (millis() > 5000 && gps.charsProcessed() < 10)
   {
      Serial.println(F("No GPS detected: check wiring."));
      while(true);
   }

}//loop
void sms (String mn)
{
   lcd.clear();
lcd.setCursor(0,0);
            lcd.print("sending sms..... ");
  SIM900.print("AT\r\n");
             SIM900.print('\n');
             ShowSerialData();
             delay(3000);
             SIM900.print("ATE1\r\n");
             ShowSerialData();
             delay(3000);
            SIM900.print("AT&W\r\n");
             SIM900.print('\n');
             ShowSerialData();
             delay(3000);
             SIM900.print("AT+CMGF=1\r\n");
             ShowSerialData();
             delay(3000);
             SIM900.print("AT+CNMI=2,2,0,0,0\r\n");
             ShowSerialData();
             delay(2000);
           // Serial.print("AT+CSMP=17,167,0,0\n");
             delay(2000);
             SIM900.print("AT+CMGS=\"09502570839\"\r");
             ShowSerialData();
           // Serial.print("");
          //Serial.print("9014449822");
          //Serial.print("");
          //Serial.print('\r');
             SIM900.print('\n');
             delay(1000);
             SIM900.print(mn);
           // SIM900.print("EMPTY \n");
        SIM900.print('\r');
          SIM900.print('\n');

        delay(3000);
           SIM900.print((char)26);
           ShowSerialData();
           lcd.setCursor(0,0);
             lcd.print("****sms sent****");
             delay(5000);
```

```
}
void sms1 (String mn)
{
   lcd.clear();
lcd.setCursor(0,0);
             lcd.print("sending sms..... ");
  SIM900.print("AT\r\n");
              SIM900.print('\n');
              ShowSerialData();
              delay(3000);
              SIM900.print("ATE1\r\n");
              ShowSerialData();
              delay(3000);
             SIM900.print("AT&W\r\n");
              SIM900.print('\n');
              ShowSerialData();
              delay(3000);
              SIM900.print("AT+CMGF=1\r\n");
              ShowSerialData();
              delay(3000);
              SIM900.print("AT+CNMI=2,2,0,0,0\r\n");
              ShowSerialData();
              delay(2000);
            // Serial.print("AT+CSMP=17,167,0,0\n");
              delay(2000);
              SIM900.print("AT+CMGS=\"08639268629\"\r");
              ShowSerialData();
            // Serial.print("");
           //Serial.print("9014449822");
           //Serial.print("");
           //Serial.print('\r');
              SIM900.print('\n');
              delay(1000);
              SIM900.print(mn);
            // SIM900.print("EMPTY \n");
         SIM900.print('\r');
           SIM900.print('\n');

         delay(3000);
            SIM900.print((char)26);
            ShowSerialData();
            lcd.setCursor(0,0);
              lcd.print("****sms sent****");
              delay(5000);

}
void ShowSerialData()
{
   while(SIM900.available()!=0)
      Serial.write(char (SIM900.read()));
}
ORG 00H
MOV P1,#11111111B     // initializes P1 as input port
MOV P0,#00000000B     // initializes P0 as output port
MOV P3,#00000000B     // initializes P3 as output port
```

```
MOV DPTR,#LABEL         // loads the address of "LABEL" to DPTR
MAIN: MOV R4,#250D    // loads register R4 with 250D
        CLR P3.7          // makes Cs=0
        SETB P3.6         // makes RD high
        CLR P3.5          // makes WR low
        SETB P3.5         // low to high pulse to WR for starting conversion
WAIT: JB P3.4,WAIT    // polls until INTR=0
        CLR P3.7          // ensures CS=0
        CLR P3.6          // high to low pulse to RD for reading the data from ADC
        MOV A,P1           // moves the digital output of ADC to accumulator A
        MOV B,#10D        // load B with 10D
        DIV AB             // divides the content of A with that in B
        MOV R6,A           // moves the quotient to R6
        MOV R7,B           // moves the remainder to R7
DLOOP:SETB P3.2         // sets P3.2 which activates LED segment 1
        MOV A,R6           // moves the quotient to A
        ACALL DISPLAY // calls DISPLAY subroutine
        MOV P0,A           // moves the content of A to P0
        ACALL DELAY      // calls the DELAY subroutine
        CLR A              // clears A
        MOV A,R7           // moves the remainder to A
        CLR P3.2           // deactivates LED segment 1
        SETB P3.1         // activates LED segment 2
        ACALL DISPLAY
        MOV P0,A
        ACALL DELAY
        CLR A
        CLR P3.1          // deactivates LED segment 2
        DJNZ R4,DLOOP // repeats the loop "DLOOP" until R4=0
        SJMP MAIN         // jumps back to the main loop
DELAY: MOV R3,#255D // produces around 0.8mS delay
LABEL1: DJNZ R3,LABEL1
          RET
DISPLAY: MOVC A,@A+DPTR // converts A's content to corresponding digit drive pattern
           RET
LABEL: DB 3FH            // LUT (look up table) starts here
         DB 06H
         DB 5BH
         DB 4FH
         DB 66H
         DB 6DH
         DB 7DH
         DB 07H
         DB 7FH
         DB 6FH
END
```