

# APPINDIX

## Arduino Code

```
#include <RFID.h>
#include <TinyGPS.h>
#include <SoftwareSerial.h>
#include <SPI.h>
#include <RFID.h>
#define SS_PIN 10
#define RST_PIN 9
RFID rfid(SS_PIN, RST_PIN);
int serNum[5];
int cards[][5] = {{ 115, 97, 91, 37, 108}};
bool access = false;
SoftwareSerial gpsSerial(6, 7); //RX=11, TX=12
TinyGPS gps; //Creates a new instance of the TinyGPS object
SoftwareSerial mySerial(4, 5);
char msg;
char call;
void setup()
{
  gpsSerial.begin(9600);
  mySerial.begin(9600); // Setting the baud rate of GSM Module
  Serial.begin(9600); // Setting the baud rate of Serial Monitor (Arduino)
```

```
pinMode(3, INPUT);
pinMode(8, OUTPUT);
digitalWrite(8, LOW);
SPI.begin();
  rfid.init();
}
void loop()
{
  rfidd();
  mySerial.begin(9600);
  if (mySerial.available() > 0) {
    Serial.write(mySerial.read());
    if (mySerial.read() > 0) {
      Serial.println("sending.....");
      SendMessage1();
      digitalWrite(8, HIGH);
    }
  }
  int x = digitalRead(3);
  Serial.println(x);
  if (x == 1) {
    SendMessage();
  }
}
```

```

void SendMessage()
{
    mySerial.println("AT+CMGF=1"); //Sets the GSM Module in Text Mode
    delay(1000); // Delay of 1000 milli seconds or 1 second
    mySerial.println("AT+CMGS=\"+249967337336\"\\r"); // Replace x with
mobile number
    delay(1000);
    mySerial.println("someone attack your car");// The SMS text you want to
send
    delay(100);
    mySerial.println((char)26);// ASCII code of CTRL+Z
    delay(1000);
    Serial.println("SendMessage");
}

```

```

void SendMessage1()
{
    gpsSerial.begin(9600);
    float flat, flon;
    unsigned long ag;
    bool newData = false;
    unsigned long chars;
    unsigned short sentences, failed;
    for (unsigned long start = millis(); millis() - start < 1000;)

```

```

{
while (gpsSerial.available())
{
char c = gpsSerial.read();
//Serial.print(c);
if (gps.encode(c))
newData = true;
}
}
if (newData) //If newData is true
{
gps.f_get_position(&flat, &flon, &ag);

Serial.print("lat: ");
Serial.println(flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flat,
6);
Serial.print("lon: ");
Serial.println(flon == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flon,
6); // print longitud

Serial.println("Sending Message...."); //Sets the GSM Module in Text
Mode
mySerial.println("AT+CMGF=1"); //Sets the GSM Module in Text
Mode
delay(1000); // Delay of 1000 milli seconds or 1 second

```

```

    mySerial.println("AT+CMGS=\"+249967337336\"\\r"); // Replace x with
mobile number
    delay(1000);
    mySerial.print("http://maps.google.com/?q=");
    mySerial.print(flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flat,
6);
    mySerial.print(","); // print latitude
    mySerial.print(flon == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 :
flon, 6); // print longitude
    //mySerial.print(",16z"); // print latitude

    delay(100);
    mySerial.println((char)26); // ASCII code of CTRL+Z
    delay(1000);
    Serial.println(" Message Send"); //Sets the GSM Module in Text Mode

}
}

```

```

void get_gps() {
    gpsSerial.begin(9600);
    float flat, flon;
    unsigned long ag;
    bool newData = false;
    unsigned long chars;

```

```

unsigned short sentences, failed;
for (unsigned long start = millis(); millis() - start < 1000;)
{
  while (gpsSerial.available())
  {
    char c = gpsSerial.read();
    //Serial.print(c);
    if (gps.encode(c))
      newData = true;
  }

}

if (newData) //If newData is true
{
  gps.f_get_position(&flat, &flon, &ag);

  Serial.print("lat: ");
  Serial.println(flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flat,
6);
  Serial.print("lon: ");
  Serial.println(flou == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flou,
6); // print longitude
}
}

```

```

void rfidd() {

    if (rfid.isCard()) {

        if (rfid.readCardSerial()) {
            Serial.print(rfid.serNum[0]);
            Serial.print(" ");
            Serial.print(rfid.serNum[1]);
            Serial.print(" ");
            Serial.print(rfid.serNum[2]);
            Serial.print(" ");
            Serial.print(rfid.serNum[3]);
            Serial.print(" ");
            Serial.print(rfid.serNum[4]);
            Serial.println("");

            for (int x = 0; x < sizeof(cards); x++) {
                for (int i = 0; i < sizeof(rfid.serNum); i++) {
                    if (rfid.serNum[i] != cards[x][i]) {
                        access = false;
                        break;
                    } else {
                        access = true;
                    }
                }
            }
        }
    }
}

```

```
    if (access) break;
    //else break;
}

}
if (!access) {
    Serial.println("Welcome!");
    delay(10000000);
}
else {
    Serial.println("Not allowed!");
}
}

rfid.halt();

}
```