```
#include <SoftwareSerial.h>
#include <AltSoftSerial.h>
#include <TinyGPSPlus.h>
#include <TinyGPS++.h>
//TinyGPSPlus gps;
//#include <TinyGsmClient.h>
const String PHONE = "+91**********";
#define ignition_switch 4
#define ignition_sensor A0
#define rxPin 11
#define txPin 10
//#define sendSms
//#define sendGpsToServer
//#define parseData
SoftwareSerial sim900A(rxPin, txPin);
AltSoftSerial neogps;
TinyGPSPlus gps;
String sms_status, sender_number, received_date, msg;
boolean ignition_status = false;
boolean tracking_status = false;
boolean reply_status = true;
boolean anti_theft = false;
unsigned long previousMillis = 0;
long interval = 60000;
void setup() {
// put your setup code here, to run once:
delay(7000);
Serial.begin(115200);
//Serial.println("Arduino serial initialize");
Sim900.begin(9600);
neogps.begin(9600);
```

```
//Serial.println("neogps Software serial initialize");
//delay(1000);
pinMode(ignition_switch, OUTPUT);
pinMode(ignition_sensor, INPUT);
sms_status = "";
sender_number = "";
received_date = "";
msg = "";
sim900.print("At+CMGF=1\r"); //SMS text mode
delay(1000);
//sendATcommand("AT+CSCLK=2","OK",1000);
}
void loop() {
// put your main code here, to run repeatedly:
ignition_status = getIgnitionStatus();
if (tracking_status == true && ignition_status == true) {
unsigned long currentMillis = millis();
if (currentMillis - previousMillis > interval) {
previousMillis = currentMillis;
//sendGpsToServer();
}
}
if (anti_theft == true && ignition_status == true) {
digitalWrite(ignition_switch, HIGH);
}
while (sim900A.available()) {
parseData(sim900A.readString());
}
while (Serial.available()) {
sim900A.println(Serial.readString());
}
```

```
}
void parseData(String buff) {
Serial.println(buff);
unsigned int len, index;
index = buff.indexOf("\r");
buff.remove(0, index + 2);
buff.trim();
if (buff != "OK") {
index = buff.indexOf(":");
String cmd = buff.substring(0, index);
cmd.trim();
buff.remove(0, index + 2);
//Serial.println(buff);
if (cmd == "+CMTI") {
//get newly arrived memory location and store it in temp
//temp=4
index = buff.indexOf(",");
String temp = buff.substring(index + 1, buff.length());
temp = "AT+CMGR=" + temp + "\r";
//AT+CMGR=4 i.e get message stored at memory location 4
sim900A.println(temp);
} else if (cmd == "+CMGR") {
extractSms(buff);
//Serial.println("extractSms Success.");
if (sender_number == PHONE) {
//Serial.println("doAction");
doAction();
//deleteSms();
}
}
} else {
}
}
void extractSms(String buff) {
unsigned int index;
Serial.println(buff);
index = buff.indexOf(",");
sms_status = buff.substring(1, index - 1);
buff.remove(0, index + 2);
sender_number = buff.substring(0, 13);
buff.remove(0, 19);
received_date = buff.substring(0, 20);
buff.remove(0, buff.indexOf("\r"));
buff.trim();
index = buff.indexOf("\n\r");
buff = buff.substring(0, index);
buff.trim();
msg = buff;
buff = "";
msg.toLowerCase();
Serial.println("______________________");
Serial.println(sms_status);
Serial.println(sender_number);
Serial.println(received_date);
Serial.println(msg);
Serial.println("______________________");
}
void doAction() {
//case sensitive
if (msg == "bike on") {
digitalWrite(ignition_switch, HIGH);
Serial.println("Prateek Your Bike has ON");
```

```
if (reply_status == true) {
sendSms("Prateek Your Bike is ON");
}
} else if (msg == "bike off") {
digitalWrite(ignition_switch, LOW);
Serial.println("Prateek Your Bike has OFF");
if (reply_status == true) {
sendSms("Prateek Your Bike is OFF");
}
} else if (msg == "get ignition") {
sendSmsGPS("Location");
} else if (msg == "anti theft on") {
anti_theft = true;
if (reply_status == true) {
sendSms("Hi Golden Your Anti Theft System on And Bike Will Be OFF");
}
} else if (msg == "anti theft off") {
anti_theft = false;
if (reply_status == true) {
sendSms("Anti Theft System OFF");
} else if (msg == "reply ON") {
reply_status = true;
sendSms("Reply has ON");
} else if (msg == "reply off") {
reply_status = false;
}
} else if (msg == "tracking on") {
tracking_status = true;
if (reply_status == true) {
sendSms("Live Tracking has ON");
}
```

```
}
//yet to be implemented
else if (msg == "tracking off") {
tracking_status = false;
if (reply_status == true) {
sendSms("Live Tracking has OFF");
}
} else if (msg == "tracking status") {
if (tracking_status == false) {
sendSms("Live Tracking has OFF");
} else {
sendSms("Live Tracking has ON");
}
}
sms_status = "";
sender_number = "";
received_date = "";
msg = "";
}
void deleteSms() {
sendATcommand("AT+CMGD=1,4", "OK", 2000);
Serial.println("All SMS are deleted");
}
void sendSmsGPS(String text) {
//Can take upt ot 60 seconds
boolean newData = false;
for (unsigned long start = millis(); millis() - start < 2000;) {
while (neogps.available()) {
if (gps.encode(neogps.read())) {
newData = true;
}
```

```
}
}
if (newData) { //If newData id true
float flat, flon;
unsigned long age;
Serial.print("latitude=");
Serial.print(gps.location.lat(), 6);
Serial.print("Longitude=");
Serial.println(gps.location.lng(), 6);
Serial.print(gps.location.rawLat().negative ? "-" : "+");
Serial.print(gps.location.rawLat().deg);
Serial.print(gps.location.rawLat().billionths);
Serial.println(gps.location.rawLng().negative ? "-" : "+");
Serial.println(gps.location.rawLng().deg
Serial.println(gps.location.rawLng().billionths);
Serial.println(gps.date.value());
Serial.println(gps.date.year());
Serial.println(gps.date.month());
Serial.println(gps.date.day());
Serial.println(gps.time.value());
Serial.println(gps.time.hour());
Serial.println(gps.time.minute());
Serial.println(gps.time.second());
Serial.println(gps.time.centisecond());
Serial.println(gps.speed.value());
Serial.println(gps.speed.knots());
Serial.println(gps.speed.mph());
Serial.println(gps.speed.mps());
Serial.println(gps.speed.kmph());
Serial.println(gps.course.value());
Serial.println(gps.course.deg());
Serial.println(gps.altitude.value());
Serial.println(gps.altitude.meters());
Serial.println(gps.altitude.miles());
Serial.println(gps.altitude.kilometers());
Serial.println(gps.altitude.feet());
Serial.println(gps.satellites.value());
Serial.println(gps.hdop.value());
Serial.println(TinyGPSPlus::libraryVersion());
newData = false;
delay(300);
sim900A.print("AT+CMGF=1\r");
delay(1000);
sim900A.print("AT+CMGS=\"" + PHONE + "\"\r");
delay(1000);
sim900A.print("http://maps.google.com/maps?q=loc:");
sim900A.print(gps.location.lat(), 6);
sim900A.print(",");
sim900A.print(gps.location.lng(), 6);
delay(1000);
sim900A.write(0x1A); //ASCII code for ctrl-26//sim900A.println((char26));//ASCII code for ctrl-26
delay(1000);
}
}
void sendSms(String text) {
sim900A.print("AT+CMGF=1\r");
delay(1000);
sim900A.print("AT+CMGS=\"" + PHONE + "\"\r");
delay(1000);
sim900A.print(text);
delay(1000);
sim900A.write(0x1A); //ASCII code for ctrl-26//sim900A.println((char26));//ASCII code for ctrl-26
```

```
delay(1000);
sim900A.print("SMS sent Successfully");
}
int8_t sendATcommand(char* ATcommand, char* expected_answer, unsigned int timeout) {
uint8_t x = 0, answer = 0;
char response[100];
unsigned long previous;
memset(response, '\0', 100); //Initializethe string
delay(1000);
while (sim900A.available() > 0) sim900A.read(); //Clean the input buffer
if (ATcommand[0] != '\0') {
sim900A.println(ATcommand); //Send the AT command
}
x = 0;
previous = millis();
//this loop waits for the answer
do {
if (sim900A.available() != 0) { //if there are data in the URAT input buffer,reads it and checks for the answer
response[x] = sim900A.read();
//Serial.print(response[x]);
x++;
if (strstr(response, expected_answer) != NULL) //check if the desired answer (OK) is in tne response of the
module
{
answer = 1;
}
}
} while ((answer) == 0 && ((millis() - previous) < timeout)); //Waits for the answer with time out
return answer;
}
```

```
boolean getIgnitionStatus() {
float val = 0;
for (int i = 0; i <= 10; i++) {
val = val + analogRead(ignition_sensor);
}
val = val / 100;
/Serial.println(val);
if (val > 90) {
sendSms("Hi Golden Someone Trying To Start Your Bike ");
sendSmsGPS("Location");
return true;
} else if (val < 50) {
return false;
}
}
void setIgnition() {
ignition_status = getIgnitionStatus;
if (ignition_status == false) {
//sim900A1 sleep mode turned off
//must send sim900A.print("AT");here
sim900A.print("AT");
sendATcommand("AT+CSCLK=0", "OK", 1000);
} else if (ignition_status == true) {
//sim900A1 sleep mode turned on
sendATcommand("AT+CSCLK=2", "OK", 1000);
}
else
{
int sendGpsToServer();
}
}
```