Readwrite permission (LInux - ubuntu)

sudo chmod a+rw /dev/ttyACM0

Message only

```
#include "SoftwareSerial.h"

SoftwareSerial mySerial(2, 3);

String cmd = "";

void setup()
{
 mySerial.begin(9600);
 Serial.begin(9600);
 Serial.println("Initializing...");
 delay(1000);

 mySerial.println("AT");                 // Sends an ATTENTION command,
reply should be OK
 updateSerial();
 mySerial.println("AT+CMGF=1");          // Configuration for sending SMS
 updateSerial();
 mySerial.println("AT+CNMI=1,2,0,0,0");  // Configuration for receiving
SMS
 updateSerial();
}

void loop()
{
 updateSerial();
}

void updateSerial()
{
 delay(500);
```

```
 while (Serial.available())
 {

   cmd+=(char)Serial.read();
   if(cmd!=""){
     cmd.trim();  // Remove added LF in transmit
     if (cmd.equals("S")) {
       sendSMS();
     } else {
       mySerial.print(cmd);
       mySerial.println("");
     }
   }
 }
  while(mySerial.available())
 {
   Serial.write(mySerial.read());//Forward what Software Serial received
to Serial Port
 }
}

void sendSMS(){
 mySerial.println("AT+CMGF=1");
 delay(500);
 mySerial.println("AT+CMGS=\"+918130887106\"\r");
 delay(500);
 mySerial.print("Hi! TechToTinker!");
 delay(500);
 mySerial.write(26);
}
```

GPS MODULE ONLY

```
#include<SoftwareSerial.h>
SoftwareSerial NEO6M(2, 3);

void setup() {
 Serial.begin(115200);

 NEO6M.begin(9600);
}

void loop() {
 while (NEO6M.available() > 0) {
   Serial.write(NEO6M.read() );
 }
}
```

The NMEA GGA sentence is **one of the most common sentences used with GPS receivers**. It contains information about position, elevation, time, number of satellites used, fix type, and correction age.

Code (Send repeated locations)

```
#include <SoftwareSerial.h>
SoftwareSerial sim800l(10, 11);
#include <SoftwareSerial.h>
#include <TinyGPS.h>
SoftwareSerial mySerial(7, 8);
TinyGPS gps;

void gpsdump(TinyGPS &gps);
void printFloat(double f, int digits = 2);
void setup()
{
 sim800l.begin(9600);
 mySerial.begin(9600);
 Serial.begin(9600);
 delay(1000);
}
void loop()
{
 bool newdata = false;
 unsigned long start = millis();
 // Every 5 seconds we print an update
 while (millis() - start < 5000)
 {
   if (mySerial.available())

   {
     char c = mySerial.read();
     //Serial.print(c);  // uncomment to see raw GPS data
     if (gps.encode(c))
     {
       newdata = true;
       break;  // uncomment to print new data immediately!
     }
   }
 }
  if (newdata)
 {
   gpsdump(gps);
```

```
   Serial.println();
 }
 }
void gpsdump(TinyGPS &gps)
{
 long lat, lon;
 float flat, flon;
 unsigned long age;
 gps.f_get_position(&flat, &flon, &age);
   Serial.println("Sending SMS...");
 sim800l.print("AT+CMGF=1\r");
 delay(100);
 sim800l.print("AT+CMGS=\"+918130887106\"\r");//EX +919876543210
 delay(500);
 sim800l.print("http://maps.google.com/maps?q=loc:");
 sim800l.print(flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flat, 6);
 sim800l.print(",");
 sim800l.print(flon == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flon, 6);
 sim800l.print((char)26);
 delay(500);
 sim800l.println();
 Serial.println("Text Sent.");
 delay(5000);
}

void printFloat(double number, int digits)
{
 // Handle negative numbers
 if (number < 0.00)
 {
    Serial.print('-');
    number = -number;
 }

 // Round correctly so that print(1.9990, 2) prints as "2.000"
 double rounding = 0.50;
 for (uint8_t i=0; i<digits; ++i)
   rounding /= 10.00;
  number += rounding;
```

```
 // Extract the integer part of the number and print it
 unsigned long int_part = (unsigned long)number;
 double remainder = number - (double)int_part;
 Serial.print(int_part);

 // Print the decimal point, but only if there are digits beyond
 if (digits > 00)
   Serial.print(".");

 // Extract digits from the remainder one at a time
 while (digits-- > 0)
 {
   remainder *= 10.00;
   int toPrint = int(remainder);
   Serial.print(toPrint);
   remainder -= toPrint;
 }
}
```