Gps neo 6m code:

#include <TinyGPS++.h>

#include <SoftwareSerial.h>

/\* Create object named bt of the class SoftwareSerial \*/

SoftwareSerial GPS\_SoftSerial(4, 3);/\* (Rx, Tx) \*/

/\* Create an object named gps of the class TinyGPSPlus \*/

TinyGPSPlus gps;

volatile float minutes, seconds;

volatile int degree, secs, mins;

void setup() {

Serial.begin(9600); /\* Define baud rate for serial communication \*/

GPS\_SoftSerial.begin(9600); /\* Define baud rate for software serial communication \*/

}

void loop() {

smartDelay(1000); /\* Generate precise delay of 1ms \*/

unsigned long start;

double lat\_val, lng\_val, alt\_m\_val;

uint8\_t hr\_val, min\_val, sec\_val;

bool loc\_valid, alt\_valid, time\_valid;

lat\_val = gps.location.lat(); /\* Get latitude data \*/

loc\_valid = gps.location.isValid(); /\* Check if valid location data is available \*/

lng\_val = gps.location.lng(); /\* Get longtitude data \*/

alt\_m\_val = gps.altitude.meters(); /\* Get altitude data in meters \*/

alt\_valid = gps.altitude.isValid(); /\* Check if valid altitude data is available \*/

hr\_val = gps.time.hour(); /\* Get hour \*/

min\_val = gps.time.minute(); /\* Get minutes \*/

sec\_val = gps.time.second(); /\* Get seconds \*/

time\_valid = gps.time.isValid(); /\* Check if valid time data is available \*/

if (!loc\_valid)

{

Serial.print("Latitude : ");

Serial.println("\*\*\*");

Serial.print("Longitude : ");

Serial.println("\*\*\*");

}

else

{

DegMinSec(lat\_val);

Serial.print("Latitude in Decimal Degrees : ");

Serial.println(lat\_val, 6);

Serial.print("Latitude in Degrees Minutes Seconds : ");

Serial.print(degree);

Serial.print("\t");

Serial.print(mins);

Serial.print("\t");

Serial.println(secs);

DegMinSec(lng\_val); /\* Convert the decimal degree value into degrees minutes seconds form \*/

Serial.print("Longitude in Decimal Degrees : ");

Serial.println(lng\_val, 6);

Serial.print("Longitude in Degrees Minutes Seconds : ");

Serial.print(degree);

Serial.print("\t");

Serial.print(mins);

Serial.print("\t");

Serial.println(secs);

}

if (!alt\_valid)

{

Serial.print("Altitude : ");

Serial.println("\*\*\*");

}

else

{

Serial.print("Altitude : ");

Serial.println(alt\_m\_val, 6);

}

if (!time\_valid)

{

Serial.print("Time : ");

Serial.println("\*\*\*");

}

else

{

char time\_string[32];

sprintf(time\_string, "Time : %02d/%02d/%02d \n", hr\_val, min\_val, sec\_val);

Serial.print(time\_string);

}

}

static void smartDelay(unsigned long ms)

{

unsigned long start = millis();

do

{

while (GPS\_SoftSerial.available()) /\* Encode data read from GPS while data is available on serial port \*/

gps.encode(GPS\_SoftSerial.read());

/\* Encode basically is used to parse the string received by the GPS and to store it in a buffer so that information can be extracted from it \*/

} while (millis() - start < ms);

}

void DegMinSec( double tot\_val) /\* Convert data in decimal degrees into degrees minutes seconds form \*/

{

degree = (int)tot\_val;

minutes = tot\_val - degree;

seconds = 60 \* minutes;

minutes = (int)seconds;

mins = (int)minutes;

seconds = seconds - minutes;

seconds = 60 \* seconds;

secs = (int)seconds;

}