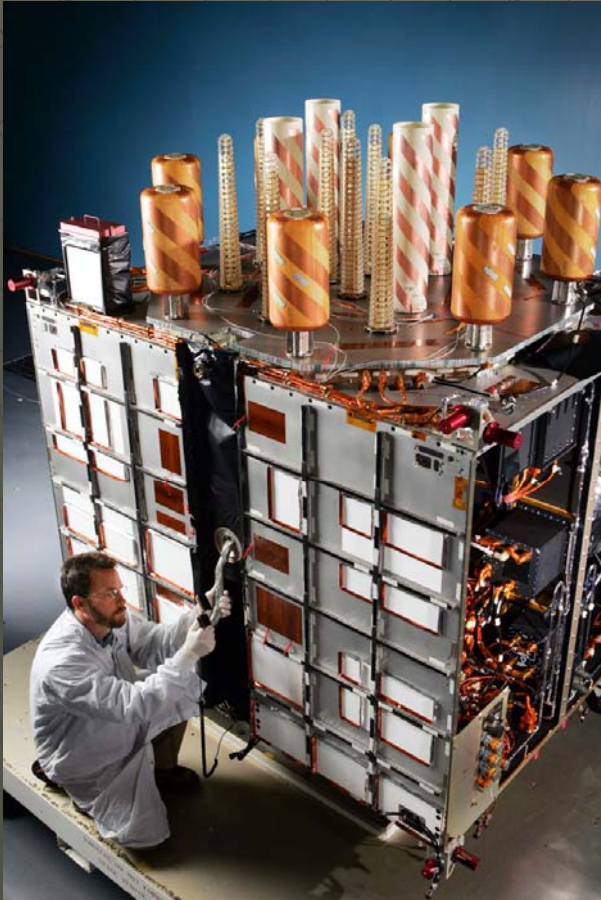




GPS – Tracking Position

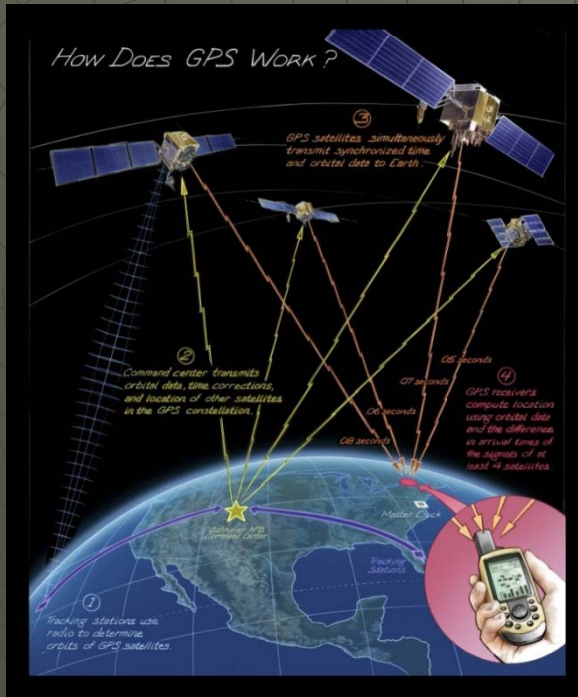


This tutorial will introduce you to GPS and hopefully help demystify it. You will learn:

- How it works
- What data is available from your GPS receiver
- How to interpret that data
- How to program the Arduino to use and display that data

Image From: <http://www.gpsreview.net/next-gps-satellite-to-launch-monday>

GPS – Tracking Position (User Space & Segments)



Your GPS Receiver measures the distance to several satellites to determine where you are using a process called trilateration

Each satellite sends its position and the time of that position to your GPS receiver

Your receiver calculate the distance you are from each satellite – that puts your location on a sphere

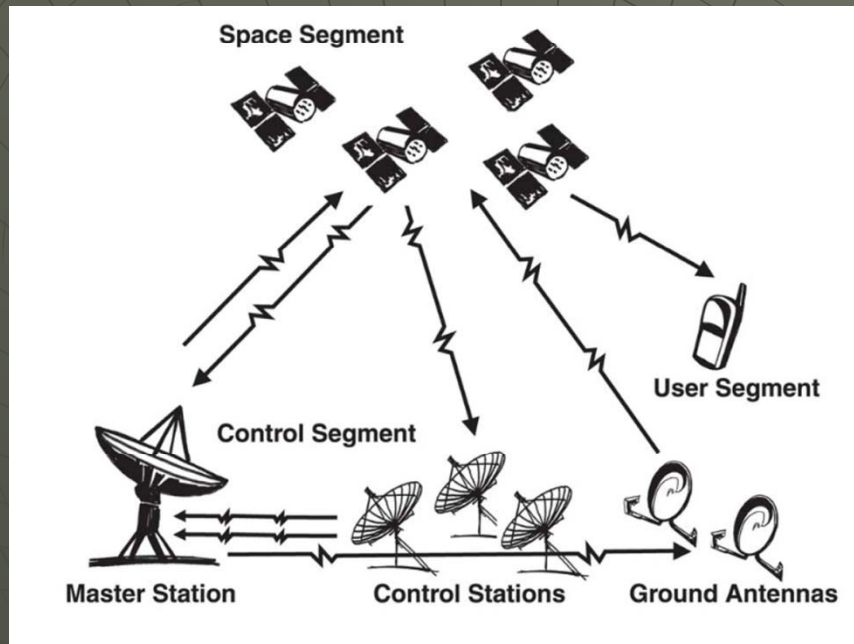
The intersection of several spheres locates where you are

There is a video describing this process at:

http://www.nasa.gov/mov/255363main_134_GPS_Work.mov

Image From: <http://timeandnavigation.si.edu/multimedia-asset/how-does-gps-work>

GPS – Three Segment System



Each satellite has a very accurate atomic clock (down to a few nanoseconds)

The control segment is five ground stations that send corrections in location and time to each of the satellites

Your GPS receiver uses the position and time sent from each satellite and a lot of math to determine your position

Your GPS receiver then reports that position and time in a simple format (we use the NMEA 0183 format, default for our GPS receivers)

Learn more at:

http://www.nwcg.gov/pms/pubs/475/PMS475_chap5.pdf

<http://www.gpsinformation.org/dale/theory.htm>

<http://www.montana.edu/gps/understd.html>

Image From: http://www.nwcg.gov/pms/pubs/475/PMS475_chap5.pdf

Hardware Required



-OPTIONAL: USB GPS receiver for direct connect to your PC

- Purchase: http://www.amazon.com/GlobalSat-BU-353-S4-USB-Receiver-Black/dp/B008200LHW/ref=sr_1_1

- GPS Receiver such as the EM-506 receiver from USGlobalSatellite: From Sparkfun and many more

- Purchase: <https://www.sparkfun.com/products/12751>
- Datasheet: http://cdn.sparkfun.com/datasheets/GPS/EM506_um.pdf
- Cable Purchase (6 pin JST for GPS to bare wire): <https://www.sparkfun.com/products/10361>

-Arduino Uno: From Arduino, Amazon, Sparkfun, MP3Cars, many more:

- Purchase: <http://www.amazon.com/Arduino-UNO-board-DIP-ATmega328P/dp/B006H06TVG>
- Schematic: http://arduino.cc/en/uploads/Main/Arduino_Uno_Rev3-schematic.pdf

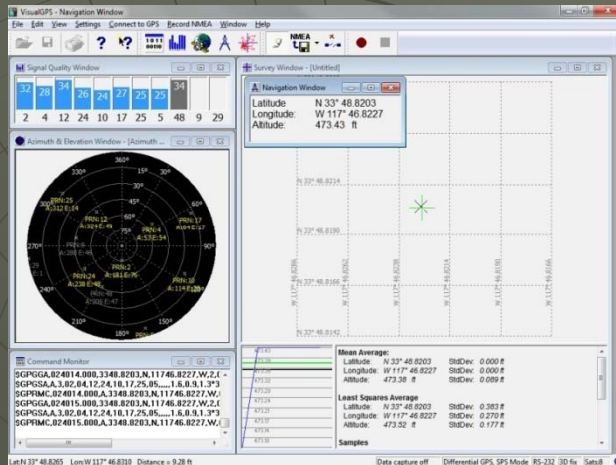
- A PC or laptop

AIAA OC Rocketry

AIAA OC Section – NAR #718



Software Required



Arduino Integrated Development Environment (IDE):

http://arduino.cc/en/main/software#_Uy4WgU1OUpA

VisualGPSView or VisualGPS (PC Software to view data from GPS Receiver)

<http://www.visualgps.net/VisualGPSView/Download/default.html>

<http://www.visualgps.net/visualgps/Download/VisualGPSInstall.exe>

Arduino Pass-through Sketch(Arduino sketch to pass on received data)

http://aiaaocrocketry.org/AIAAOCRocketryDocs/SPARC2014/Sketches/GPSPass_Through.zip

Arduino TinyGPS Library with examples:

https://github.com/mikalhart/TinyGPS_ (select "Download .zip" on the right below the list of files)

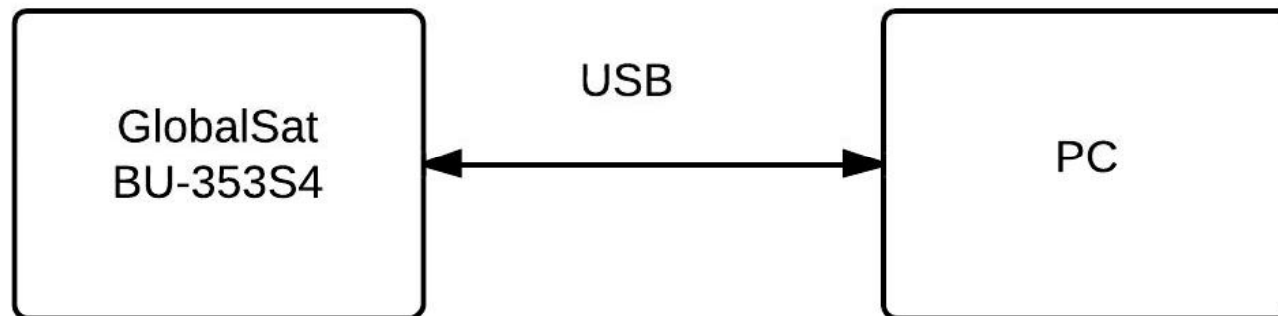
- Install the .zip file as the TinyGPS Library
- Test_with_gps_device is in the examples folder
- Note that this also uses the SoftwareSerial library that is included in the Arduino IDE V1.0 & later



Connecting the GPS to the PC

Block Diagram

In this first exercise, we will connect the PC to a commercial USB module meant to connect to a PC via USB and examine the output – both in a useful human readable form as well as the raw data coming from the module



Optional – USB GPS on PC



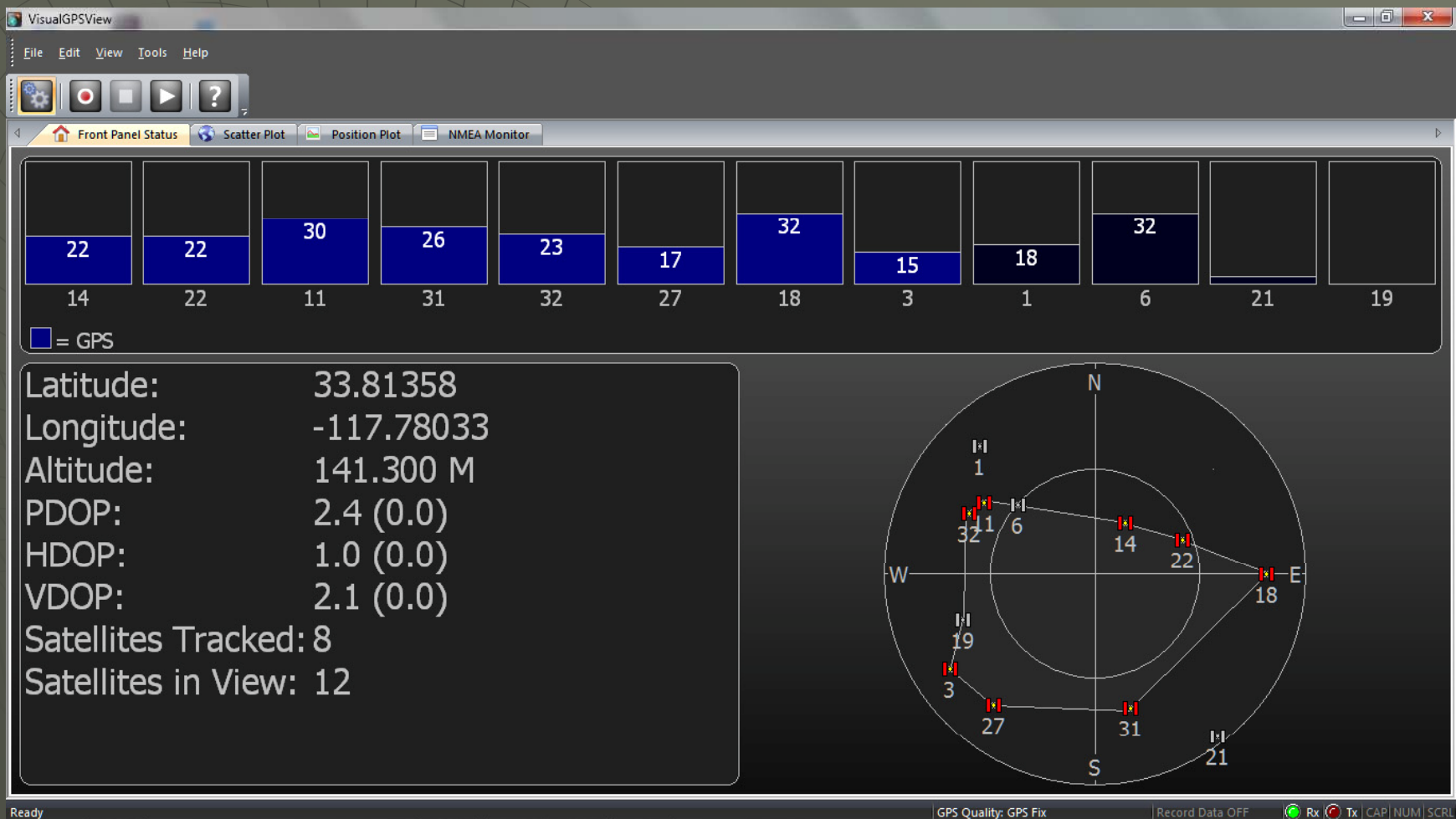
Connect a USB GPS receiver with NMEA 0183 messaging to your receiver (e.g. GlobalSat BU-353S4) Drivers, if required, are on the included mini CD (you will need to be outside or next to a window)

Install and start the VisualGPSView software on your PC

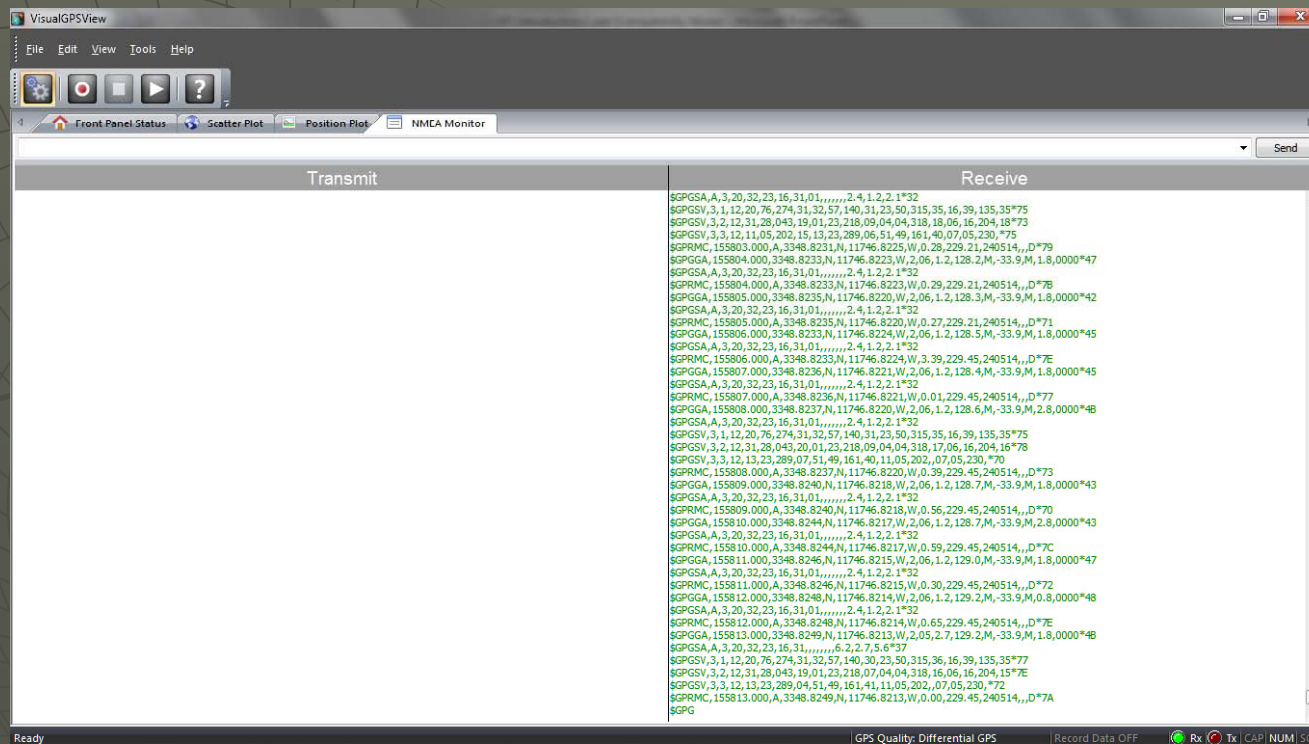
To connect and start the GPS, click on the gears icon in the upper left and select your COM port (ours shows "Prolific USB-to-Serial Comm Port")

You should see a screen showing your current location, numbers of satellites, and the dilution of precision for the Positional, Vertical, and Horizontal locations (see next slide)

GPS Software (VisualGPSView)



GPSView NMEA Monitor Tab



The full NMEA 0183 Specification that explains the messages from the GPS receiver is available from NMEA but expensive. The subset that you need (albeit a little dated) in the SiRF NMEA manual is available at:

http://www.usglobalsat.com/store/downloads/NMEA_commands.pdf



NMEA 0183 GGA Message (Captured Data from the GPS Receiver)

```
$GPGGA,024230.000,3348.8189,N,11746.8215,W,2,08,0.9,147.8,M,-33.9,M,2.8,0000*4E
$GPGSA,A,3,02,04,12,24,10,17,25,05,,,,,1.6,0.9,1.3*38
$GPRMC,024230.000,A,3348.8189,N,11746.8215,W,0.15,346.87,230514,,,D*7A
$GPGGA,024231.000,3348.8184,N,11746.8219,W,2,08,0.9,147.8,M,-33.9,M,2.8,0000*4E
$GPGSA,A,3,02,04,12,24,10,17,25,05,,,,,1.6,0.9,1.3*38
$GPGSV,3,1,11,02,78,180,29,04,53,051,22,12,50,325,35,24,47,236,11*74
$GPGSV,3,2,11,10,25,113,31,17,16,065,16,25,15,313,11,05,10,163,23*79
$GPGSV,3,3,11,48,47,206,32,09,46,277,,29,02,263,*44
$GPRMC,024231.000,A,3348.8184,N,11746.8219,W,2.52,285.67,230514,,,D*7B
$GPGGA,024232.000,3348.8182,N,11746.8222,W,2,08,0.9,147.9,M,-33.9,M,2.8,0000*42
$GPGSA,A,3,02,04,12,24,10,17,25,05,,,,,1.6,0.9,1.3*38
$GPRMC,024232.000,A,3348.8182,N,11746.8222,W,0.27,285.67,230514,,,D*76
$GPGGA,024233.000,3348.8182,N,11746.8223,W,2,08,0.9,147.8,M,-33.9,M,3.8,0000*42
$GPGSA,A,3,02,04,12,24,10,17,25,05,,,,,1.6,0.9,1.3*38
$GPRMC,024233.000,A,3348.8182,N,11746.8223,W,0.46,285.67,230514,,,D*71
$GPGGA,024234.000,3348.8184,N,11746.8221,W,2,08,0.9,147.6,M,-33.9,M,4.8,0000*48
$GPGSA,A,3,02,04,12,24,10,17,25,05,,,,,1.6,0.9,1.3*38
$GPRMC,024234.000,A,3348.8184,N,11746.8221,W,1.71,313.26,230514,,,D*7C
$GPGGA,024235.000,3348.8183,N,11746.8221,W,2,08,0.9,147.3,M,-33.9,M,2.8,0000*4D
$GPGSA,A,3,02,04,12,24,10,17,25,05,,,,,1.6,0.9,1.3*38
$GPRMC,024235.000,A,3348.8183,N,11746.8221,W,1.14,313.26,230514,,,D*79
$GPGGA,024236.000,3348.8182,N,11746.8222,W,2,08,0.9,147.2,M,-33.9,M,2.8,0000*4D
$GPGSA,A,3,02,04,12,24,10,17,25,05,,,,,1.6,0.9,1.3*38
$GPGSV,3,1,11,02,78,180,29,04,53,051,23,12,50,325,34,24,47,236,17*72
```



NMEA 0138 GGA Message Decoded (Global Positioning System Fixed Data)

We can use the information in the SiRF NMEA Manual to decode the information received from the GPS Receiver. The TinyGPS library will handle these details for you – but this is how TinyGPS does its job. It uses two messages – the GGA and the RMC

```
$GPGGA,024230.000,3348.8189,N,11746.8215,W,2,08,0.9,147.8,M,-33.9,M,2.8,0000*4E<CRLF>
```

- \$GPGGA -> Message ID = GGA (Time, Position, and Fix Type Data)
- 024230.000 -> UTC Time hh.mmss.ss = 24 hrs 42 minutes 30.000 seconds
- 3348.8149 -> Latitude ddmm.mmmm = 33 degrees 48.8149 seconds
- N -> N/S Indicator (N=North S=South) = North
- 11746.8215 -> Longitude dddmm.mmmm = 117 degrees 46.8215 minutes
- W -> E/W Indicator (E=East W=West) = West
- 2 -> Position Fix Indicator (0=Inv or N/A, GPS SPS=1, Diff GPS SPS=2, DeadRek=6) = GPS Valid
- 08 -> Satellites Used (0 to 12) = 8
- 0.9 -> HDOP Horizontal Dillution of Precision = 0.9
- 147.8 -> MSL Altitude = 147.8 m
- M -> Units = Meters
- -33.9 -> Geoid Separation (Ellipsoid Altitude = MSL Altitude + Geoid Separation) = -33.9 m
- M -> Units = Meters
- 2.8 -> Age of Differential Correction = 2.8 seconds
- 0000 -> Diff Ref Station ID = 0000
- 4E -> Checksum (XOR of each character after \$ and before *) = 4E
- <CRLF> -> Carriage Return Line feed End Of Message Terminator (0x0d, 0x0a)



NMEA 0138 RMC Message Decoded (Recommended Minimum Specific GNSS Data)

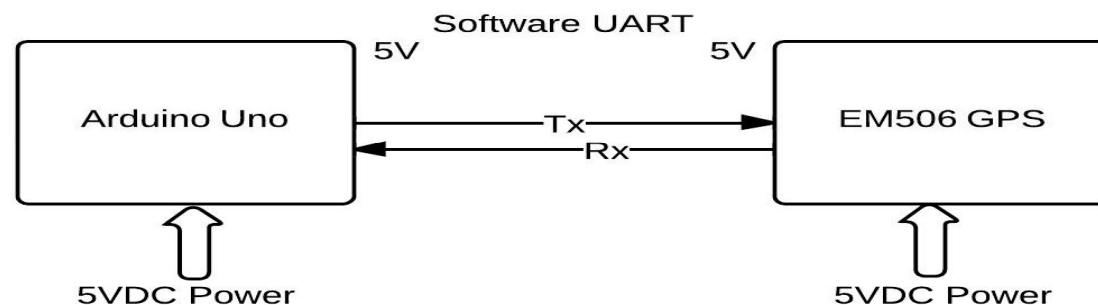
```
$GPRMC,024230.000,A,3348.8189,N,11746.8215,W,0.15,346.87,230514,,,D*7A<CRLF>
```

- \$GPGGA -> Message ID = RMC (Time, Date, Position, and Fix Type Data)
- 024230.000 -> UTC Time hh.mmss.ss = 24 hrs 42 minutes 30.000 seconds
- A -> Status (A=Data Valid V=Data Not Valid) = Valid
- 3348.8189 -> Latitude ddmm.mmmm = 33 degrees 48.8189 seconds
- N -> N/S Indicator (N=North S=South) = North
- 11746.8215 -> Longitude dddmm.mmmm = 117 degrees 46.8215 minutes
- W -> E/W Indicator (E=East W=West) = West
- 0.15 -> Speed Over Ground = 0.15 Knots
- 346.87 -> Course Over Ground = 346.87 degrees true
- 230514 -> Date ddmmyy = 05/23/14
- <empty> -> Magnetic Variation Degrees E=East W=West = <empty>
- 7A -> Checksum (XOR of each character after \$ and before *) = 7A
- <CRLF> -> Carriage Return Line feed End Of Message Terminator (0x0d, 0x0a)

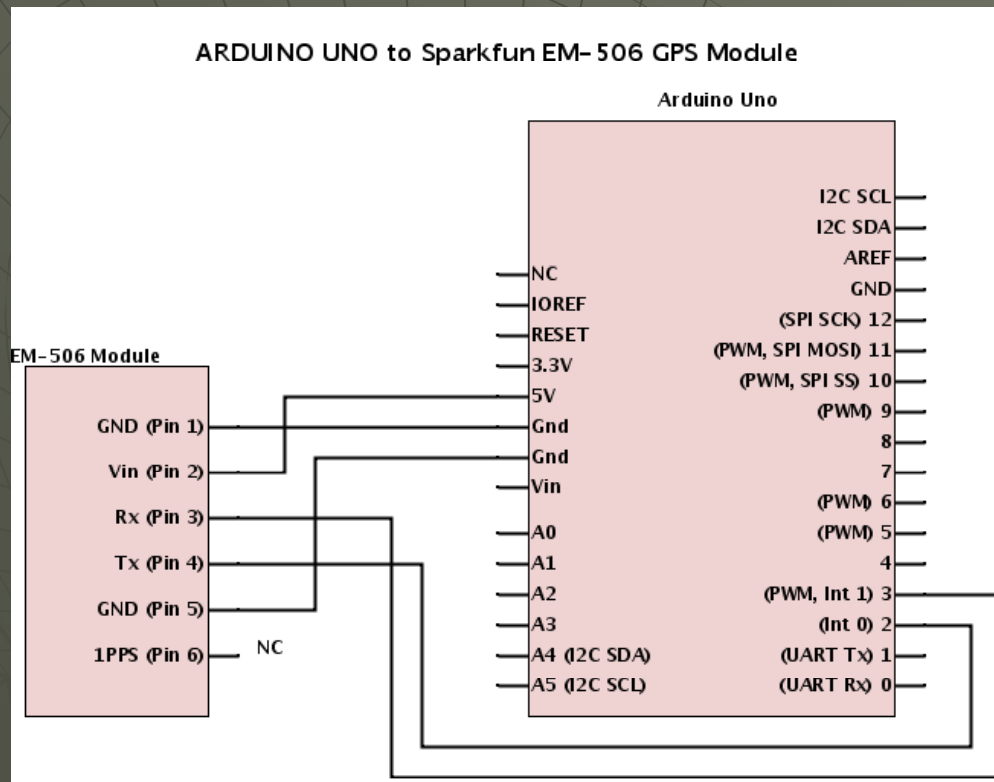
As you can see, extracting your location from the data provided by the USB GPS Receiver is straightforward. In the next steps, we'll see that exactly the same data is provided by the EM-406A module for the Arduino – and we can use an Arduino program instead of the PC to interpret and display the data in a very human readable and understandable format

Connecting the EM506GPS to the Arduino Uno Block Diagram

In this final exercise, we will connect the EM506 module to an Arduino Uno, then connect the Arduino Uno to a PC using the USB cable. The Arduino Uno will be in pass-through mode (it accepts a character from one device and passes it on to another device unmodified). The results should be identical to those received on the commercial module connected directly to the PC



Connecting the EM506GPS to the Arduino Uno Schematic



EM-506 Pinout

EM-506 Datasheet: http://www.usglobalsat.com/store/download/717/EM506_um.pdf
Both the Arduino Uno and the EM-506 run at 5VDC to no level shifting is required



Connecting the EM506GPS to the PC



It may make prototyping easier for the GPS if you modify the GPS cable by cutting the jumpers in half and soldering them to the ends of the cable

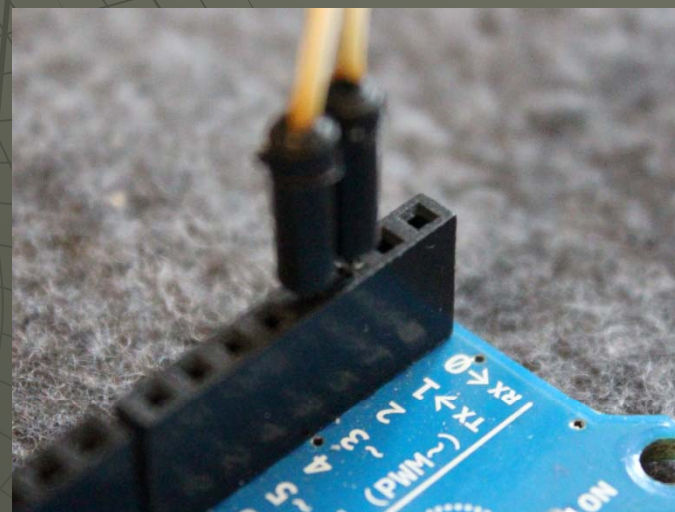
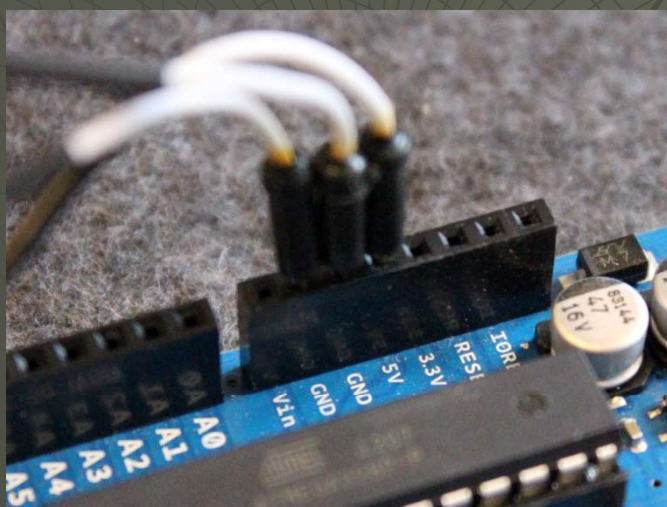
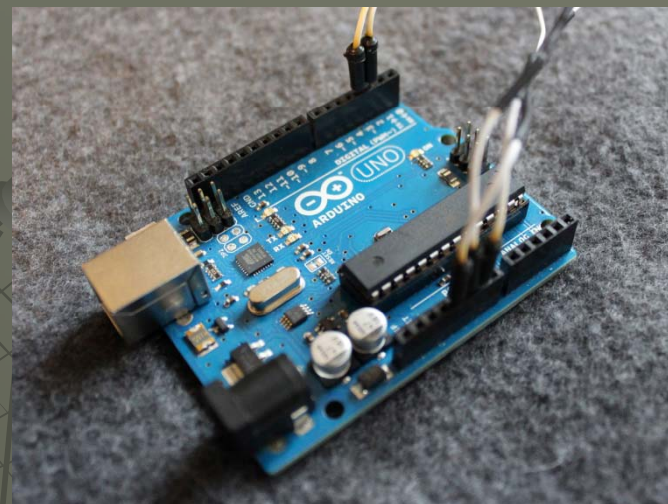
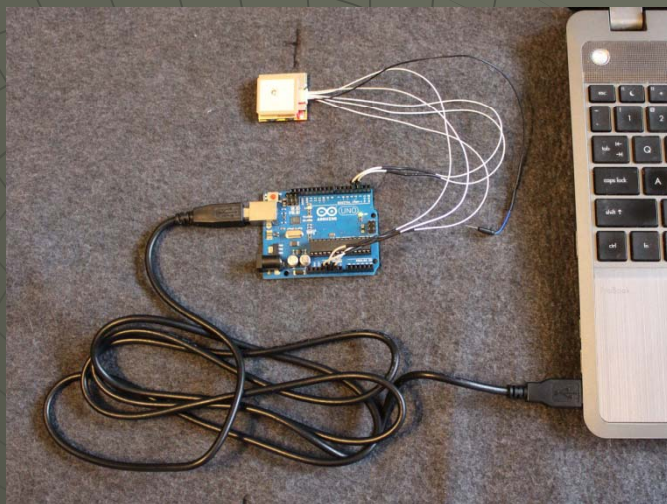
6 pin JST for GPS to bare wire: <https://www.sparkfun.com/products/10361>

Male to Male jumpers: <https://www.sparkfun.com/products/12795>

Note: There are many options for jumper wires



Connecting the EM506GPS to the PC





Download and run the GPS_Passthrough sketch from the AIAA OC Rocketry Website:

http://aiaaocrocketry.org/AIAAOCRocketryDocs/SPARC2014/Sketches/GPSPass_Through.zip

Unzip the folder with the .ino file into the Sketches folder under Arduino in (My)Documents. This program will just pass the data from the GPS receiver to the PC via the USB cable. It uses a software UART on pins 2 and 3 on the Arduino to talk to the GPS receiver and the hardware UART to talk to the PC via USB

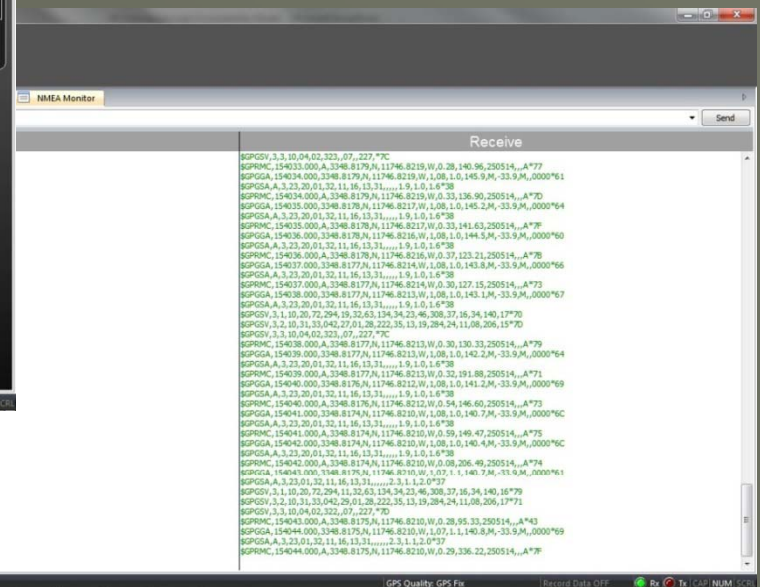
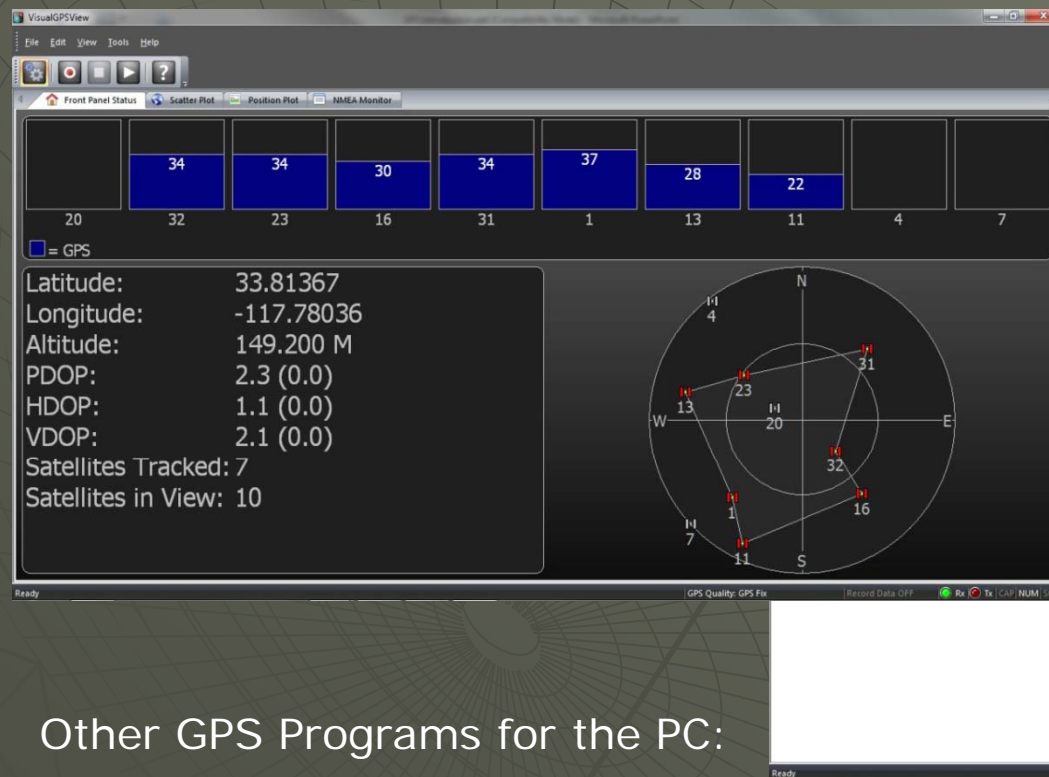
```
#include <SoftwareSerial.h>

// GPS Baud Rate is 4800
#define GPSBAUD 4800
#define RXPIN 2
#define TXPIN 3

SoftwareSerial uart_gps(RXPIN, TXPIN);
void setup()
{
  uart_gps.begin(GPSBAUD);
  Serial.begin(4800);
}
void loop()
{
  byte a;
  if ( uart_gps.available() > 0 )
  {
    a = uart_gps.read(); // get the byte of data from the GPS
    Serial.write(a);
  }
}
```


GPS Software

(VisualGPSView – Looks the same as previous module)



Other GPS Programs for the PC:

GPS Info Program

<http://www.usglobalsat.com/store/downloads/GPSInfo.zip>

SiRF Demo Program

<http://www.usglobalsat.com/store/downloads/SiRFDemo387.zip>