SYNERGY SYSTEMS, LLC
Time proven products and support

Motorola ONCORE™ GPS Receiver NMEA Protocol Communications

rev 15DEC04

by Randy Warner
Senior Applications Engineer, Synergy Systems, LLC

Introduction

Although the Motorola binary protocol is the native communication standard used with the Motorola Oncore series of GPS receivers, many of the receivers (M12 and M12+ Positioning receivers, GT+, etc.) also support the National Marine Electronics Association (NMEA) standard. In a nutshell, NMEA is a 4800 baud, N/8/1 ASCII text based messaging protocol. The fact that it is ASCII text based makes it useful for many users who simply want navigation and time information without going through all of the mathematical calisthentics of parsing out the information provided by the receiver in binary mode. Also, once placed in NMEA mode the receiver can be interfaced with most of the commercial GPS display/mapping programs such Street Atlas, Expedia Streets, and Map ‘n Go, among others.

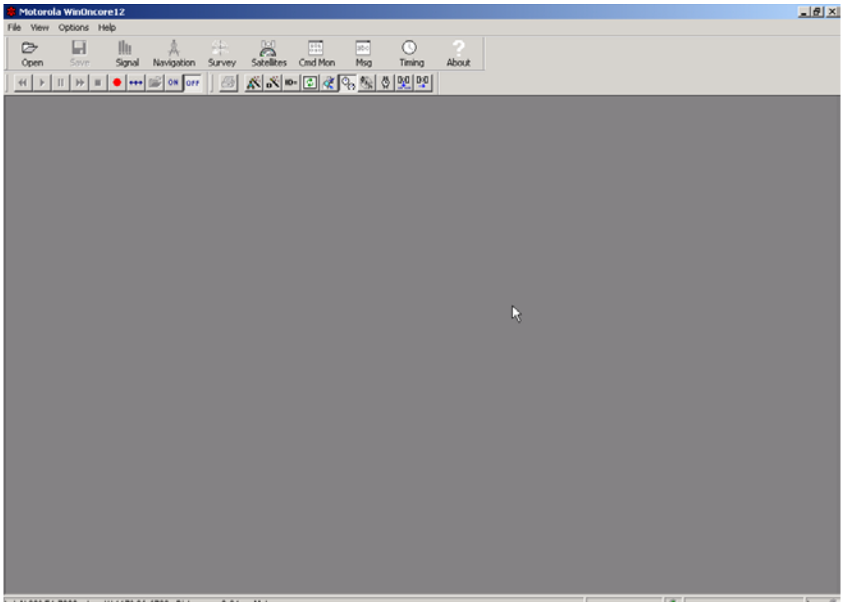
The limitation of operating in NMEA mode is that it only outputs GPS data in a limited number of formats, and there is no facility for CONTROLLING the receiver as there is in binary mode. If you want to change receiver parameters (Satellite Mask Angle, for instance), you must revert back to binary, change the mask angle, and then switch back to NMEA. Actually, this is not such a bad thing. 99% of the time leaving the receiver with the factory default values for things such as Mask Angle is the smart move. It is extremely easy for a user to unintentionally degrade a GPS receiver’s performance by accidentally changing operating parameters to unusable values.

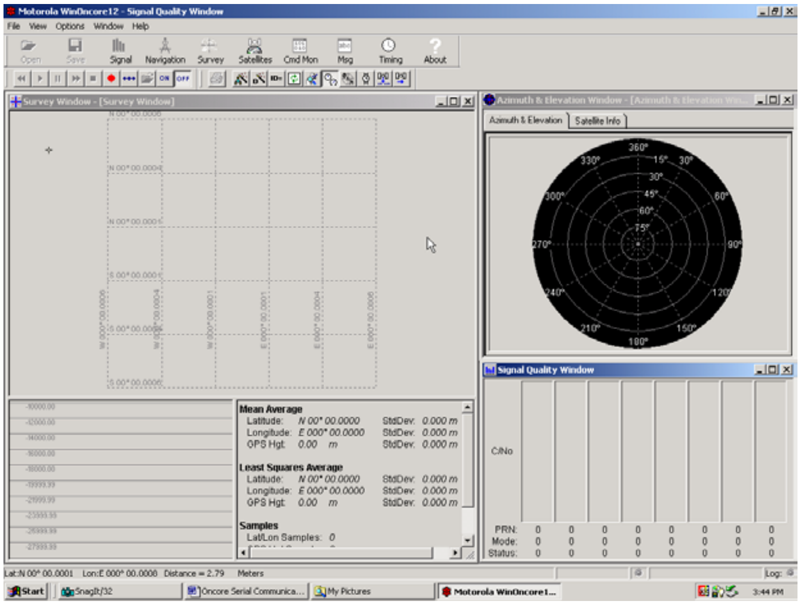
Communicating in NMEA

Using the WinOncore12 software is certainly the quickest way to start out. Once you have seen how these commands are formatted using WinOncore12 you should be able to format your own command strings for use with your own application's software.

Initialization Using WinOncore12

Before we can go much further, we need to get WinOncore set up. Shown below is the screen displayed the first time WinOncore is run after being installed. As you can see, it is not really ready to be used yet as none of the user windows are open. By clicking on the icons on the main Toolbar I usually open the <Signal>, <Survey>, and <Satellites> windows as they display the information I usually find most useful. After pulling the windows around and resizing them the screen should look something like this:


Motorola WinOncore12
File View Options Help
Open Save Signal Navigation Survey Satellites Cmd Mon Msg Timing About


Motorola WinOncore12 - Signal Quality Window
File View Options Window Help
Open Save Signal Navigation Survey Satellites Cmd Mon Msg Timing About
Survey Window - [Survey Window]
Azimuth & Elevation Window - [Azimuth & Elevation Win...
Azimuth & Elevation
Satellite Info
360°
15°
30°
330°
300°
45°
60°
60°
75°
270°
90°
240°
120°
210°
150°
180°
Signal Quality Window
C/No
Start
SnagIt/32
Oncore Serial Communica...
My Pictures
Motorola WinOncore1...
3:44 PM
Log:

Mean Average
Latitude: N 00° 00.0000 StdDev: 0.000 m
Longitude: E 000° 00.0000 StdDev: 0.000 m
GPS Hgt: 0.00 m StdDev: 0.000 m

Least Squares Average
Latitude: N 00° 00.0000 StdDev: 0.000 m
Longitude: E 000° 00.0000 StdDev: 0.000 m
GPS Hgt: 0.00 m StdDev: 0.000 m

Samples
Lat/Lon Samples: 0

PRN:	0	0	0	0	0	0	0	0
Mode:	0	0	0	0	0	0	0	0
Status:	0	0	0	0	0	0	0	0

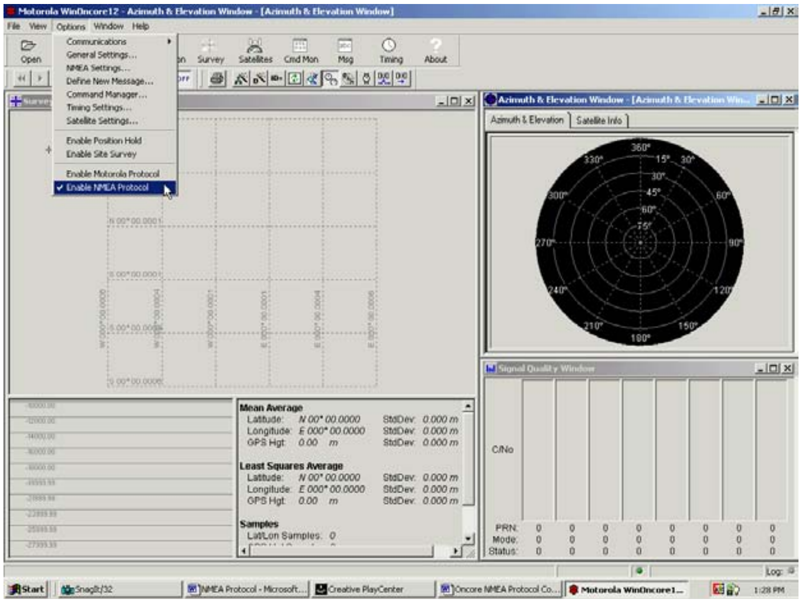
Lat:N 00° 00.0001 Lon:E 000° 00.0008 Distance = 2.79 Meters

As you can see, I now have three windows open that can show average receiver position, satellite locations, and the C/No numbers for the satellites being tracked. Note that the <Survey>

window displays running mean and least squares averages of the position reports. If you want to see un-averaged numbers you can open the <Navigation> window. This will display second by second positions along with other useful information about the current status of the receiver.

Now that we have a usable screen displayed we need to request that the receiver send out data for display. The simplest way to initialize the receiver is to do the following:

1. Click on <Options> and make sure that "Enable NMEA Protocol" is checked as shown below:

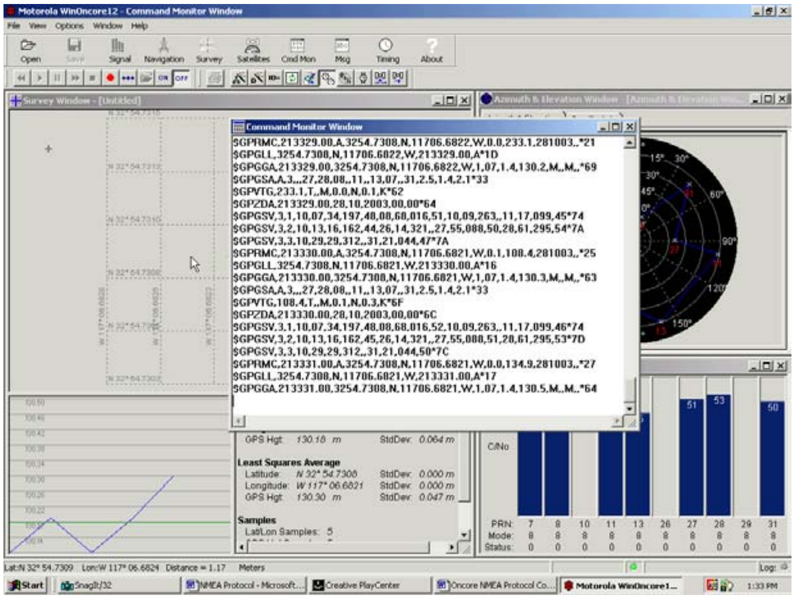


Mean Average
Latitude: N 00° 00.0000 StdDev: 0.000 m
Longitude: E 000° 00.0000 StdDev: 0.000 m
GPS Hgt: 0.00 m StdDev: 0.000 m

Least Squares Average
Latitude: N 00° 00.0000 StdDev: 0.000 m
Longitude: E 000° 00.0000 StdDev: 0.000 m
GPS Hgt: 0.00 m StdDev: 0.000 m

Samples
Lat/Lon Samples: 0

2. As soon as you click on the <Enable NMEA Protocol> text, WinOncore12 will send out NMEA commands to turn on specific NMEA messages. WinOncore12's default state is to turn on all seven NMEA messages as a 1 Hz rate. If you open the <Cmd Mon> window you should see something like this:


Motorola WinOncore12 - Command Monitor Window
File View Options Window Help
Open Save Signal Navigation Survey Satellites Cmd Mon Msg Timing About
Survey Window - [Untitled]
Command Monitor Window
C/No
PRN: 7 8 10 11 13 26 27 28 29 31

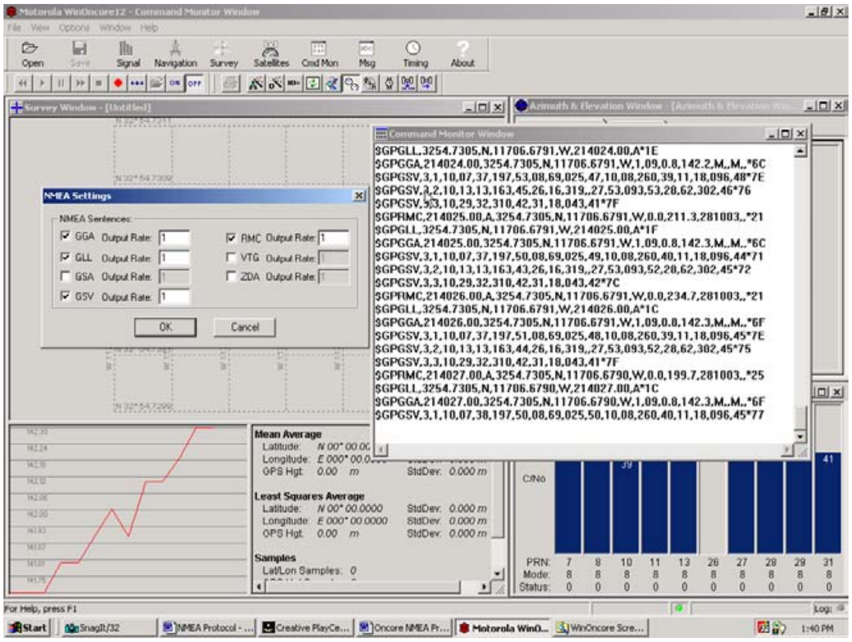
```
$GPRMC,213329.00,A,3254.7308,N,11706.6822,W,0.0,233.1,281003,,*21
$GPGLL,3254.7308,N,11706.6822,W,213329.00,A*1D
$GPGGA,213329.00,3254.7308,N,11706.6822,W,1,07,1.4,130.2,M,,M,,*69
$GPGSA,A,3,,,27,28,08,,11,,13,07,,31,2.5,1.4,2.1*33
$GPVTG,233.1,T,,M,0.0,N,0.1,K*62
$GPZDA,213329.00,28,10,2003,00,00*64
$GPGSV,3,1,10,07,34,197,48,08,68,016,51,10,09,263,,11,17,099,45*74
$GPGSV,3,2,10,13,16,162,44,26,14,321,,27,55,088,50,28,61,295,54*7A
$GPGSV,3,3,10,29,29,312,,31,21,044,47*7A
$GPRMC,213330.00,A,3254.7308,N,11706.6821,W,0.1,108.4,281003,,*25
$GPGLL,3254.7308,N,11706.6821,W,213330.00,A*16
$GPGGA,213330.00,3254.7308,N,11706.6821,W,1,07,1.4,130.3,M,,M,,*63
$GPGSA,A,3,,,27,28,08,,11,,13,07,,31,2.5,1.4,2.1*33
$GPVTG,108.4,T,,M,0.1,N,0.3,K*6F
$GPZDA,213330.00,28,10,2003,00,00*6C
$GPGSV,3,1,10,07,34,197,48,08,68,016,52,10,09,263,,11,17,099,46*74
$GPGSV,3,2,10,13,16,162,45,26,14,321,,27,55,088,51,28,61,295,53*7D
$GPGSV,3,3,10,29,29,312,,31,21,044,50*7C
$GPRMC,213331.00,A,3254.7308,N,11706.6821,W,0.0,134.9,281003,,*27
$GPGLL,3254.7308,N,11706.6821,W,213331.00,A*17
$GPGGA,213331.00,3254.7308,N,11706.6821,W,1,07,1.4,130.5,M,,M,,*64
```

GPS Hgt: 130.18 m StdDev: 0.064 m

Least Squares Average
Latitude: N 32° 54.7308 StdDev: 0.000 m
Longitude: W 117° 06.6821 StdDev: 0.000 m
GPS Hgt: 130.30 m StdDev: 0.047 m

Samples
Lat/Lon Samples: 5

3. Typically, there is no need to send out all seven messages, the most popular being the GGA, GSA, and GSV as these are the messages used by most mapping programs. The other messages may be turned off quite simply in WinOncore12 using the <NMEA Settings> dialog, which is accessed through the <Options> menu. Once open, the <NMEA Settings> dialog looks like this:



NMEA Settings

NMEA Sentences:

☑ GGA Output Rate: 1
☑ GLL Output Rate: 1
☐ GSA Output Rate: 1
☑ GSV Output Rate: 1
☑ RMC Output Rate: 1
☐ VTG Output Rate: 1
☐ ZDA Output Rate: 1

OK Cancel

Command Monitor Window

```
$GPGLL,3254.7305,N,11706.6791,W,214024.00,A*1E
$GPGGA,214024.00,3254.7305,N,11706.6791,W,1,09,0.8,142.2,M,,M,,*6C
$GPGSV,3,1,10,07,37,197,53,08,69,025,47,10,08,260,39,11,18,096,48*7E
$GPGSV,3,2,10,13,13,163,45,26,16,319,,27,53,093,53,28,62,302,46*76
$GPGSV,3,3,10,29,32,310,42,31,18,043,41*7F
$GPRMC,214025.00,A,3254.7305,N,11706.6791,W,0.0,211.3,281003,,*21
$GPGLL,3254.7305,N,11706.6791,W,214025.00,A*1F
$GPGGA,214025.00,3254.7305,N,11706.6791,W,1,09,0.8,142.3,M,,M,,*6C
$GPGSV,3,1,10,07,37,197,50,08,69,025,49,10,08,260,40,11,18,096,44*71
$GPGSV,3,2,10,13,13,163,43,26,16,319,,27,53,093,52,28,62,302,45*72
$GPGSV,3,3,10,29,32,310,42,31,18,043,42*7C
$GPRMC,214026.00,A,3254.7305,N,11706.6791,W,0.0,234.7,281003,,*21
$GPGLL,3254.7305,N,11706.6791,W,214026.00,A*1C
$GPGGA,214026.00,3254.7305,N,11706.6791,W,1,09,0.8,142.3,M,,M,,*6F
$GPGSV,3,1,10,07,37,197,51,08,69,025,48,10,08,260,39,11,18,096,45*7E
$GPGSV,3,2,10,13,13,163,44,26,16,319,,27,53,093,52,28,62,302,45*75
$GPGSV,3,3,10,29,32,310,42,31,18,043,41*7F
$GPRMC,214027.00,A,3254.7305,N,11706.6790,W,0.0,199.7,281003,,*25
$GPGLL,3254.7305,N,11706.6790,W,214027.00,A*1C
$GPGGA,214027.00,3254.7305,N,11706.6790,W,1,09,0.8,142.3,M,,M,,*6F
$GPGSV,3,1,10,07,38,197,50,08,69,025,50,10,08,260,40,11,18,096,45*77
```

Note that here I have unchecked the GSA, VTG, and ZDA dialog boxes. In addition, their update rate dialog boxes have been grayed out, indicating that they are not active. The Command Monitor window also shows that the three messages have been turned off.

Switching to NMEA

Switching your receiver to NMEA mode is deceptively easy, but seems to cause users no end of confusion. In order to switch to NMEA, you must successfully send the proper binary string to the receiver at 9600 baud, 8/N/1, change your com port to support 4800 baud, 8/N/1 communications, and then start communications in NMEA.

The command to switch the receiver into NMEA mode is quite short. In the classical Motorola notation the entire string is:

@@Ci012B<CR><LF>

where

“@@Ci” is the command header in ASCII
“01” is the command mode in hex
“2B” is the checksum in hex, and
<CR><LF> are the ASCII Carriage Return and Linefeed characters which terminate the command

I tend to find the Motorola notation (jumping back and forth between ASCII and HEX characters in the string) rather confusing, so I like to convert all characters to their hex equivalents:

$40 $40 $43 $69 $01 $2B $0D $0A

This is probably how you would encode this string into memory of a microcontroller for use with the Oncore anyway.

Communicating in the NMEA Protocol

OK great, you send the string to the receiver, so now what happens? Well, if you have done everything right, ABSOLUTELY NOTHING. This is one of the few Oncore commands where the receiver does not acknowledge a properly formatted message. In fact, the best indication that the Oncore has switched over to NMEA is to note if all binary message outputs have STOPPED. In other words, if your receiver was happily putting out binary messages BEFORE you issued the NMEA command, and is now silent, you can be pretty sure that the receiver is in NMEA mode, patiently awaiting further instructions.[1]

OK, no problem. We now switch our communications to 4800 baud and issue a NMEA command, such as:

$PMOTG,GGA,0001<CR><LF>

Note that we are in pure ASCII now. The receiver should immediately start issuing “GGA” messages once every second. You can change the update rate to anything you want by changing the '0001' characters to something else. Also note that I have ignored the checksum character in the GGA command string. This is permissible as the checksums are optional in the NMEA format.

This general command structure can be repeated for any or all of the seven supported NMEA messages.

Operating in Polled Mode

Many users are surprised (and pleased) when they discover that the messages may be requested in a 'polled' mode. In polled mode the receiver will send one string when requested and then cease further output of that string. For instance, the GGA example shown above could be converted to polled mode by sending the string:

$PMOTG,GGA,0000<CR><LF>

Every time this command is sent to the receiver, the receiver will output a GGA string and then cease further GGA outputs. This '0000' argument can be repeated for any or all of the seven NMEA output strings. Operating in polled mode generally makes life a lot easier for users trying to write code for small micro-controllers. Instead of having to worry about writing resource draining interrupt routines to keep the serial buffer from overflowing, one can simply tell the receiver to be quiet until data is requested.

---

[1] The exception to this behavior is the case in which the receiver was previously sending out NMEA strings, was commanded into binary mode, and then back into NMEA. So long as the receiver has not been turned off (or has a backup battery), the receiver will output the same NMEA strings that it was before the switch to binary.

Switching back to Binary

Getting back to binary is just as easy. With your serial port still set for 4800 baud, send the ASCII string:

$PMOTG,FOR,0<CR><LF>

If the receiver was previously outputting binary strings before being changed to NMEA mode, these same strings will be active when you switch it back. Remember, now that you are back in binary you have to set your serial port to 9600 baud......

This raises the question: why would I want to switch to binary if I plan to operate in NMEA? The answer is that you need to switch into binary mode if you need to change any of the operating parameters of the receiver, run Self-Tests, etc, as these operations can only be accomplished in binary mode.

Constructing Your Own Messages

Now that we have been through the basics of NMEA communications with the receiver using WinOncore12, let's try writing our own commands and receiving data back from the receiver. Shown below is a little test program I wrote some time ago that shows how to initiate basic communications with the M12+ extract NMEA data, and then send it to a serial 16 character x 2 line LCD for display. The target for this program is the Basic Stamp IISX which has several unique serial communications macros that automate a lot of the port setup chores in the 'Constant Declaration' section, but no matter which processor you use the basic message formatting will be the same. I have added numbers to the lines that we will be discussing in detail. The comments in the code should explain everything else.

```
'NMEA to LCD Display Program - Randy Warner, Synergy Systems, LLC    29NOV01

'       constant declarations

n9600          con     $4054          'set serial port for 9600 baud using hex constant
n4800          con     188            'set serial port for 4800 baud using dec constant
LCD            con     0              'define LCD port as Pin P0
I              con     254            'LCD Instruction Prefix
CLR            con     1              'LCD Clear Instruction
Row1           con     128            'set LCD cursor to row 1, char 1
Row2           con     192            'set LCD cursor to row 2, char 1
GPSCMD         con     2              'define GPS command port as Pin P2
GPSDATA        con     1              'define GPS data port as Pin P1
RET            con     $0D            'Carriage Return
LF             con     $0A            'Line Feed

'       variable declarations
deg            var     byte(3)        'whole degrees of current position
minute         var     byte(2)        'whole minutes of current position
dec_minute     var     byte(2)        'decimal minutes of current position
time           var     byte(6)        'current UTC time
dir            var     byte           'direction (N, S, E, W)of current position data
track          var     byte(2)        'number of satellites being tracked

'       main program

(1) serout GPSCMD,n9600,[$40,$40,$43,$69,$01,$2B,C_R,L_F]   'if in binary, set to NMEA
serout LCD,n9600,[I,CLR]                                     'clear LCD display

       loop
              serout LCD,n9600,[I,Row1]                      'set LCD to Row 1, char 1
              (2) serout GPSCMD,n4800,["$PMOTG,GGA,0000",RET,LF]
```

```
        (3) serin GPSDATA, n4800,[skip 7, str time\6, skip 4, str deg\2, str
        minute\2, skip 1, str dec_minute\2, skip 3, dir]
        (4) serout LCD,n9600,[str deg\2, 223, str minute\2, 46, str dec_minute\2,
        dir, 32, str time\6]
        pause 100
        serout LCD,n9600,[I,Row2]                    'set LCD to Row 2, char 1
        (5) serout GPSCMD,n4800,["$PMOTG,GGA,0000",RET,LF]
        (6) serin GPSDATA, n4800,[skip 29, str deg\3, str minute\2, skip 1, str
        dec_minute\2, skip 3, dir, skip 3, str track\2]
        (7)serout LCD,n9600,[str deg\3, 223, str minute\2, 46, str dec_minute\2,
        dir, 32, str track\2, "SVs"]
        pause 100
goto loop
```

OK, let's dissect this little bit of code:

Line 1- directs the microcontroller to send out a binary string to tell the receiver to switch to the NMEA protocol. If the receiver is in binary, it will switch into NMEA. If the receiver is in NMEA already the command will simply be ignored.

Line 2 - the micro-controller requests a GGA string from the receiver. Note that the update rate argument is set to '0000'. This orders the receiver to output only one GGA string and then stop further GGA output. Operating the receiver in this 'polled' mode makes life easier on simple micro-controllers as their serial buffers don't get inundated with a lot of extraneous data that the micro-controller can't handle with limited assets.

Line 3 - the micro-controller takes in the time and latitude information from the Oncore receiver. Notice that I am “cherry picking” the data I do want by telling the Basic Stamp to “SKIP” characters I don’t want.

Line 4 - the micro-controller sends latitude and time stamp information to the LCD

Lines 5, 6, and 7 - the micro-controller requests a second 'GGA' message. This time the Basic Stamp picks off the longitude and number of tracked satellites information for display on the second line of the LCD.

Conclusion

That's really all there is to NMEA communications with Oncore receivers. Things to remember are that the Oncore always defaults to binary protocol when power is disconnected, and that the receiver must be commanded into NMEA mode before NMEA communications can be initiated. It is ALWAYS good design practice to ASSUME that the receiver has powered up in the default mode and include code in your application to ensure that the receiver makes the switch to NMEA.