Connecting a Parallax GPS module to the Arduino

This tutorial shows how to connect a Parallax GPS module to the Arduino, and how to use Arduino code to read information like date, time, location and satellites in view from the standard NMEA data streams that the module produces.

Hardware Connections:

The module connects to the Arduino through a 4800 bps TTL-level interface (8 data bits, no parity, 1 stop bit, non-inverted). Only four wires are needed to read the module's GPS data.

Understanding NMEA GPS strings

GPS modules typically put out a series of standard strings of information, under something called the National Marine Electronics Association (NMEA) protocol. More information on NMEA standard data strings can be found atthis site

The tutorial code at the bottom of this page demonstrates how to decode and display the most common string, called $GPRMC. If all you need is date, time and position, you can to skip reading this, and just run the code below.

While you can write software to serially request other strings from the Parallax module, the following strings are automatically transmitted when the "/RAW" pin is pulled low.

- $GPGGA: Global Positioning System Fix Data
- $GPGSV: GPS satellites in view
- $GPGSA: GPS DOP and active satellites
- $GPRMC: Recommended minimum specific GPS/Transit data

Each of these sentences contains a wealth of data. For example, here are a few instances of the $GPRMC string, aka the "Recommended minimum specific GPS/Transit data" string:

```
 eg1. $GPRMC,081836,A,3751.65,S,14507.36,E,000.0,360.0,130998,011.3,E*62
 eg2. $GPRMC,225446,A,4916.45,N,12311.12,W,000.5,054.7,191194,020.3,E*68
           225446       Time of fix 22:54:46 UTC
           A            Navigation receiver warning A = Valid position, V =
Warning
           4916.45,N    Latitude 49 deg. 16.45 min. North
           12311.12,W   Longitude 123 deg. 11.12 min. West
           000.5        Speed over ground, Knots
           054.7        Course Made Good, degrees true
           191194       UTC Date of fix, 19 November 1994
           020.3,E      Magnetic variation, 20.3 deg. East
           *68          mandatory checksum
```

```
 eg3. $GPRMC,220516,A,5133.82,N,00042.24,W,173.8,231.8,130694,004.2,W*70
                1    2    3    4    5     6    7    8      9     10  11 12
      1   220516     Time Stamp
      2   A          validity - A-ok, V-invalid
      3   5133.82    current Latitude
      4   N          North/South
      5   00042.24   current Longitude
      6   W          East/West
      7   173.8      Speed in knots
      8   231.8      True course
      9   130694     Date Stamp
      10  004.2      Variation
      11  W          East/West
      12  *70        checksum
 eg4. for NMEA 0183 version 3.00 active the Mode indicator field is added
     $GPRMC,hhmmss.ss,A,llll.ll,a,yyyyy.yy,a,x.x,x.x,ddmmyy,x.x,a,m*hh
 Field #
 1    = UTC time of fix
 2    = Data status (A=Valid position, V=navigation receiver warning)
 3    = Latitude of fix
 4    = N or S of longitude
 5    = Longitude of fix
 6    = E or W of longitude
 7    = Speed over ground in knots
 8    = Track made good in degrees True
 9    = UTC date of fix
 10   = Magnetic variation degrees (Easterly var. subtracts from true
course)
 11   = E or W of magnetic variation
 12   = Mode indicator, (A=Autonomous, D=Differential, E=Estimated, N=Data
not valid)
 13   = Checksum
```

The Arduino Code

```
 /*
 Example code for connecting a Parallax GPS module to the Arduino
 Igor Gonz?lez Mart?n. 05-04-2007
 igor.gonzalez.martin@gmail.com
 English translation by djmatic 19-05-2007
 Listen for the $GPRMC string and extract the GPS location data from this.
 Display the result in the Arduino's serial monitor.
 */
 #include <string.h>
 #include <ctype.h>
 int ledPin = 13;                  // LED test pin
 int rxPin = 0;                    // RX PIN
 int txPin = 1;                    // TX TX
 int byteGPS=-1;
 char linea[300] = "";
 char comandoGPR[7] = "$GPRMC";
 int cont=0;
 int bien=0;
 int conta=0;
 int indices[13];
 void setup() {
   pinMode(ledPin, OUTPUT);       // Initialize LED pin
   pinMode(rxPin, INPUT);
   pinMode(txPin, OUTPUT);
   Serial.begin(4800);
```

```
   for (int i=0;i<300;i++){       // Initialize a buffer for received data
     linea[i]=' ';
   }
 }
 void loop() {
   digitalWrite(ledPin, HIGH);
   byteGPS=Serial.read();         // Read a byte of the serial port
   if (byteGPS == -1) {           // See if the port is empty yet
     delay(100);
   } else {
     linea[conta]=byteGPS;        // If there is serial port data, it is
put in the buffer
     conta++;
     Serial.print(byteGPS, BYTE);
     if (byteGPS==13){            // If the received byte is = to 13, end
of transmission
       digitalWrite(ledPin, LOW);
       cont=0;
       bien=0;
       for (int i=1;i<7;i++){     // Verifies if the received command
starts with $GPR
         if (linea[i]==comandoGPR[i-1]){
           bien++;
         }
       }
       if(bien==6){               // If yes, continue and process the data
         for (int i=0;i<300;i++){
           if (linea[i]==','){    // check for the position of the  ","
separator
             indices[cont]=i;
             cont++;
           }
           if (linea[i]=='*'){    // ... and the "*"
             indices[12]=i;
             cont++;
           }
         }
         Serial.println("");      // ... and write to the serial port
         Serial.println("");
         Serial.println("---------------");
         for (int i=0;i<12;i++){
           switch(i){
             case 0 :Serial.print("Time in UTC (HhMmSs): ");break;
             case 1 :Serial.print("Status (A=OK,V=KO): ");break;
             case 2 :Serial.print("Latitude: ");break;
             case 3 :Serial.print("Direction (N/S): ");break;
             case 4 :Serial.print("Longitude: ");break;
             case 5 :Serial.print("Direction (E/W): ");break;
             case 6 :Serial.print("Velocity in knots: ");break;
             case 7 :Serial.print("Heading in degrees: ");break;
             case 8 :Serial.print("Date UTC (DdMmAa): ");break;
             case 9 :Serial.print("Magnetic degrees: ");break;
             case 10 :Serial.print("(E/W): ");break;
             case 11 :Serial.print("Mode: ");break;
             case 12 :Serial.print("Checksum: ");break;
           }
           for (int j=indices[i];j<(indices[i+1]-1);j++){
             Serial.print(linea[j+1]);
           }
           Serial.println("");
         }
```

```
        Serial.println("---------------");
      }
      conta=0;                     // Reset the buffer
      for (int i=0;i<300;i++){     //
        linea[i]=' ';
      }
    }
  }
}
```

If the above code works, in the serial monitor window you should see the following:

Valid position (status is A):

```
...
...
$GPGGA,154653,4428.2011,N,00440.5161,W,0,00,,-00044.7,M,051.6,M,,*6C
$GPGSA,A,1,,,,,,,,,,,,,,,*1E
$GPGSV,3,1,10,02,50,290,003,10,25,24,045,35,27,56,145,00,,,,,,,,*78
$GPRMC,154653,V,4428.2011,N,00440.5161,W,000.5,342.8,050407,,,N*7F
---------------
Time in UTC (HhMmSs): 154653
Status (A=OK,V=KO): V
Latitude: 4428.2011
Direction (N/S): N
Longitude: 00440.5161
Direction (E/W): W
Speed in knots: 000.5
Direction in degrees: 342.8
Date in UTC (DdMmAa): 050407
Magnetic variation:
Variation (E/W):
Mode: A
---------------
...
...
```

B: Valid positon: The GPS light shines steadily.

```
...
...
$GPGGA,154654,4428.2011,N,00440.5161,W,0,00,,-00044.7,M,051.6,M,,*6B
$GPGSA,A,1,,,,,,,,,,,,,,,*1E
$GPGSV,3,1,10,02,50,290,00
$GPGGA,154655,4328.1874,N,00340.5185,W,1,03,08.5,-00044.7,M,051.6,M,,*79
$GPGSA,A,2,13,23,25,,,,,,,,,08.5,08.5,00.9*0E
$GPGSV,3,1,10,02,50,290,26,04,60,210,26,08,33,173,29,10,21,296,00*7E
$GPGSV,3,2,10,13,58,044,34,16,03,035,00,20,02,109,00,23,26,057,34*7B
$GPGSV,3,3,10,25,24,045,35,27,56,145,27,,,,,,,,*7D
$GPRMC,154655,A,4428.1874,N,00440.5185,W,000.7,000.0,050407,,,A*6C
---------------
Time in UTC (HhMmSs): 154655
Status (A=OK,V=KO): A
Latitude: 4428.1874
Direction (N/S): N
Longitude: 00440.5185
Direction (E/W): W
Speed in Knots: 000.7
Heading in degrees: 000.0
Date in UTC (DdMmAa): 050407
Magnetic variation:
Variation (E/W):
Mode: A
---------------
...
```