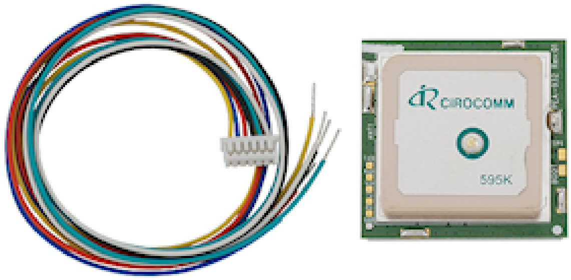
Published on *learn.parallax.com* (http://learn.parallax.com)

GPS Module PMB-648 [1]

Submitted by Gordon McComb on Thu, 05/10/2012 - 11:55



Item code: 28500

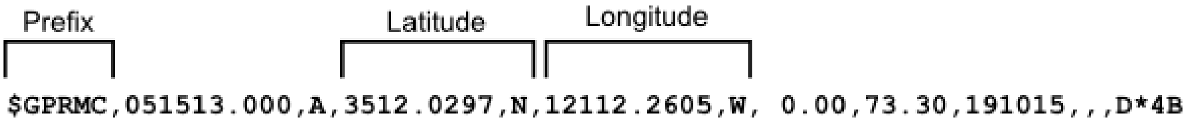
What It Can Do

- Tracks up to 20 satellites, with updates at least once a second
- Provides geo-location data for robots, vehicles, custom GPS units
- Receives signals outdoors or indoors* using a built-in antenna

*Indoor reception is dependent on location within the building, thickness of construction materials, and other factors. For the most consistent results, the GPS receiver should have a clear view of the sky.

The PMB-648 is a self-contained global positioning satellite (GPS) receiver, capable of providing accurate latitude, longitude, altitude, speed, heading, and other information useful for navigation. The data provided by the module is in the industry standard NMEA0183 v2.2 format, making it easy to interpret and use.

GPS data consists of text *sentences* that contain latitude, longitude, and other information. Each sentence consists of a prefix, plus one or more blocks of data, each separated by a comma. Using a microcontroller, you can parse each sentence to extract just the navigation information you’re looking for. For the examples in this KickStart only the latitude and longitude information is parsed and processed.



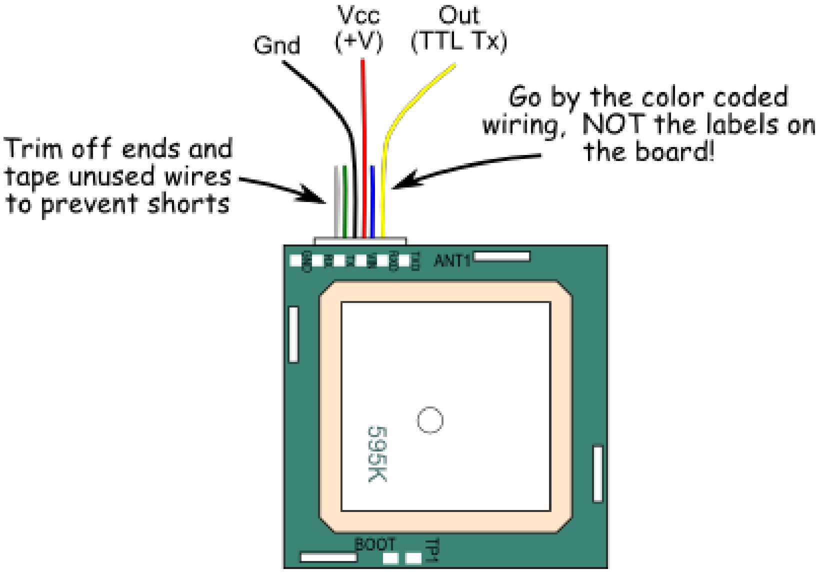
```
$GPRMC,051513.000,A,3512.0297,N,12112.2605,W, 0.00,73.30,191015,,,D*4B
```

While GPS receivers like the PMB-648 follow a standard data protocol, there can be variations between makes and models of receivers in how the data is provided. This KickStart is specifically written for the PMB-648 module, and the programming examples may not work properly with other GPS receivers.

Parts List

- PMB-648 GPS module
- BASIC Stamp HomeWork Board, Propeller BOE, Propeller QuickStart, or Arduino Uno microcontroller
- 22 gauge solid conductor hookup wire

Basic Wiring



Gnd
Vcc (+V)
Out (TTL Tx)
Go by the color coded wiring, NOT the labels on the board!
Trim off ends and tape unused wires to prevent shorts
ANT1
595K
BOOT
TP1

- Power requirements: 3.3 to 5 VDC
- Power consumption: 65 mA @ 5 VDC
- Communications: TTL or RS-232 asynchronous serial @ 4800 bps
- Supported NMEA sentences: GGA, GSV, GSA, RMC (optional VTG, GLL)
- Dimensions: 1.25 x 1.25 x .35 in (32 x 32 x 9 mm)

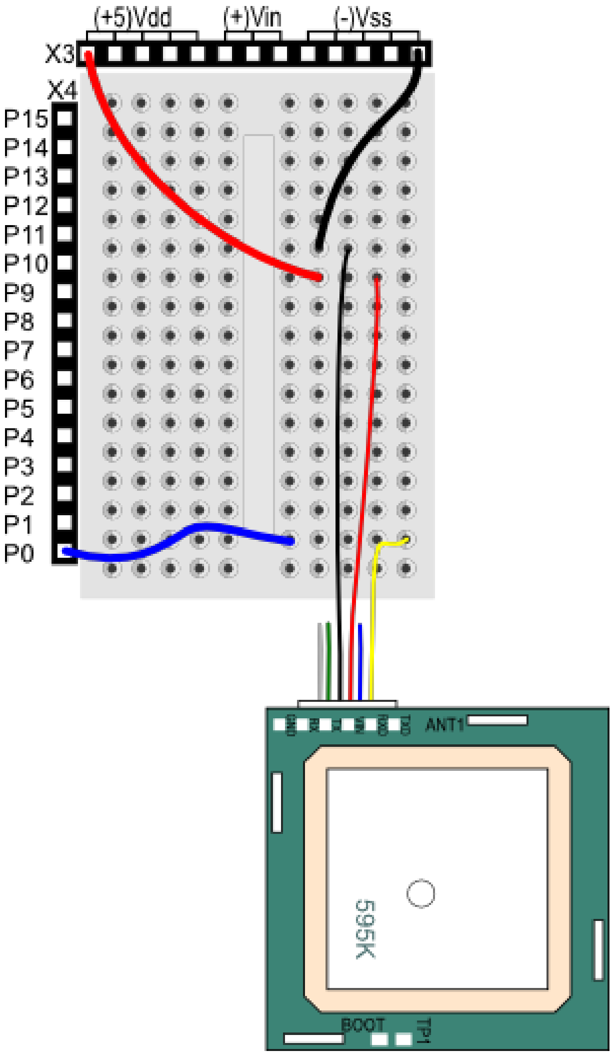
Program KickStarts

For each of the examples in this KickStart the data output is converted to a format that is directly compatible with Google Maps – degrees and minutes for both latitude and longitude. To visually see your location on a map or satellite view, open the Google Maps page, and copy/paste the latitude & longitude data into the location bar.

For proper operation the GPS module requires a 1 to 3 minute warm-up period, in order to acquire satellite data. Acquisition time may be increased if the module is used indoors, or near radio interference.

A red LED on the module indicates satellite lock. When the LED is blinking, the GPS module is still acquiring data, and its output may not be reliable. Wait until the LED is on steadily before reading the data from the module.

BASIC Stamp HomeWork Board



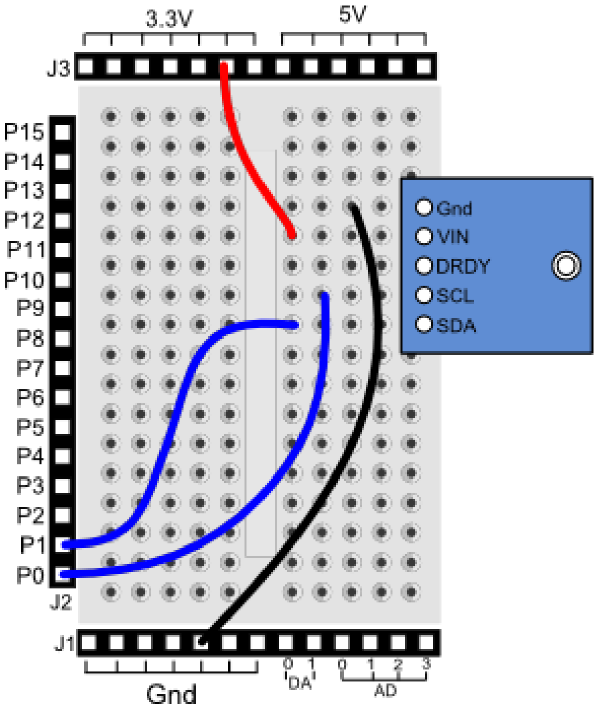
Download BASIC Stamp 2 code for the PMB-648 GPS [2]

```
' {$STAMP BS2}
' {$PBASIC 2.5}

LatDeg      VAR   Byte
LatMin      VAR   Byte
LatMinD     VAR   Word
```

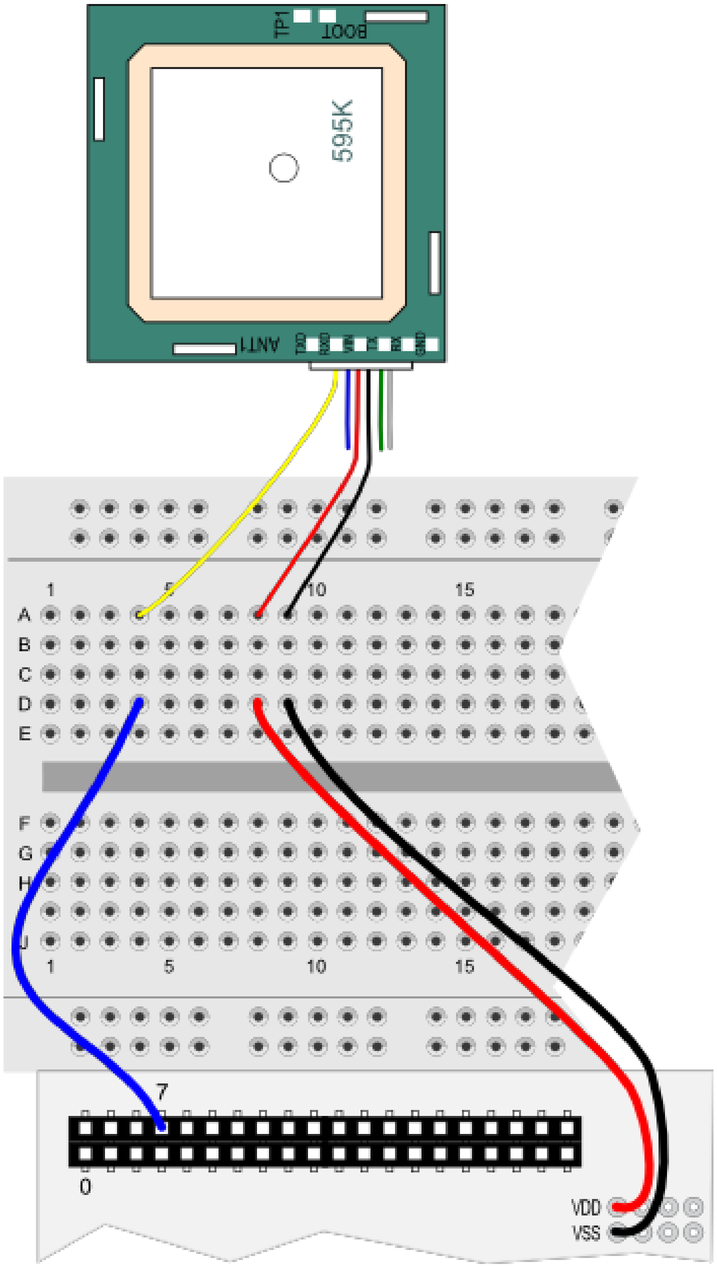
```
LatSign    VAR   Byte
LatDir     VAR   Byte

LonDeg     VAR   Byte
LonMin     VAR   Byte
LonMinD    VAR   Word
LonSign    VAR   Byte
LonDir     VAR   Byte

wVal       VAR   Word

'Baud rates (non-inverted):
n4800 CON 188
n9600 CON 84

Main:
  'Get GPSRMC statement from pin 0
  SERIN 0, n4800, [WAIT("RMC,"),SKIP 13,
      DEC2 LatDeg, DEC2 LatMin, SKIP 1, DEC LatMinD, SKIP 1, LatSign,
      DEC3 LonDeg, DEC2 LonMin, SKIP 1, DEC LonMinD, SKIP 1, LonSign]

  GOSUB AddSign
  PAUSE 760

  wVal = (LatMin * 1000 / 6) + (LatMinD / 60) 'convert to min/seconds
  SEROUT 16, n9600, [REP "-"\LatDir, DEC LatDeg, ".", DEC4 wVal, ", "]

  wVal = (LonMin * 1000 / 6) + (LonMinD / 60) 'convert to min/seconds
  SEROUT 16, n9600, [REP "-"\LonDir, DEC LonDeg, ".", DEC4 wVal, CR]

  GOTO Main
  DEBUG CR        ' Show empty debug window if GPS does not respond

AddSign:
  IF LatSign = "S" THEN
    LatDir = 1
  ELSE
    LatDir = 0
  ENDIF

  IF LonSign = "W" THEN
    LonDir = 1
  ELSE
    LonDir = 0
  ENDIF

  RETURN
```

When this program is run, the BASIC Stamp Debug Terminal will automatically open.

Propeller BOE and Propeller QuickStart



Propeller BOE Wiring Diagram



Propeller QuickStart Wiring Diagram

Download Propeller Spin code for the PMB-648 GPS [3]

```
OBJ

  Debug              : "FullDuplexSerialPlus"
  GPS                : "GPS_Float_Lite"
  FS                 : "FloatString"

CON

  _CLKMODE            = XTAL1 + PLL16x
  _XINFREQ            = 5_000_000

PUB Init |   fv
```

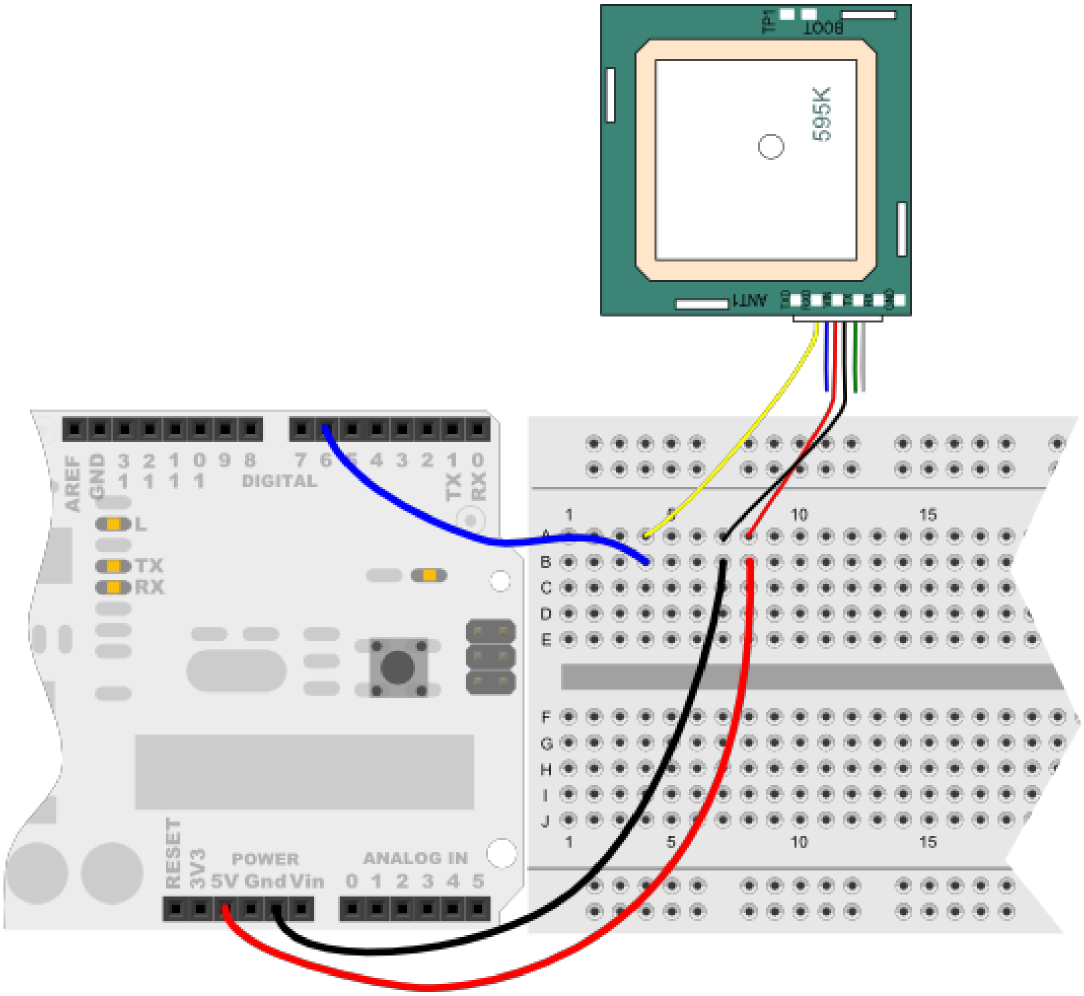
```
  'Start FullDuplexSerialPlus terminal
  Debug.Start(31, 30, 0, 115200)
  Debug.Str(String("Reading GPS..."))

  WaitCnt(ClkFreq * 2 + Cnt)

  'Start GPS object, GPS connected to P7
  GPS.Init

  Repeat
    Debug.Str(String(16, 1))

    FS.SetPrecision(7)
    fv := GPS.Float_Latitude_Deg                ' Get latitude
    If fv <> floatNaN
      Debug.Str(FS.FloatToString(fv))
    Else
      Debug.Str(String("---"))

    Debug.Str(String(","))
    fv := GPS.Float_Longitude_Deg               ' Get longitude
    If fv <> floatNaN
      Debug.Str(FS.FloatToString(fv))
    Else
      Debug.Str(String("---"))
    WaitCnt(ClkFreq / 2 + ClkFreq / 4 + Cnt)


DAT

  floatNaN       LONG $7FFF_FFFF              'Means Not a Number
```

Important! This program uses the following object libraries, which are included in the download. To use this program you must include these libraries in the same folder as the program file. The library contains several files in a zip archive. See instructions for using Propeller object libraries [4] for more information.

- FloatMath.spin
- FloatString.spin
- FullDuplexSerialPlus.spin
- GPS_Float_Lite.spin
- GPS_Str_NMEA_Lite.spin

To view the results of the demonstration, after uploading is complete run the Parallax Serial Terminal from the Run menu, or press F12. Momentarily depress the Reset button on the Propeller QuickStart board to restart the program.

Arduino Uno



Download Arduino 1.0 Code for the PMB-648 GPS [5]

Download Arduino Pre-release Version Code for the PMB-648 GPS [6]

```
#include <SoftwareSerial.h>
#include "./TinyGPS.h"                 // Special version for 1.0

TinyGPS gps;
SoftwareSerial nss(6, 255);            // Yellow wire to pin 6

void setup() {
  Serial.begin(115200);
  nss.begin(4800);
  Serial.println("Reading GPS");
}

void loop() {
  bool newdata = false;
  unsigned long start = millis();
  while (millis() - start < 5000) {  // Update every 5 seconds
    if (feedgps())
      newdata = true;
  }
  if (newdata) {
    gpsdump(gps);
  }
}

// Get and process GPS data
void gpsdump(TinyGPS &gps) {
  float flat, flon;
  unsigned long age;
  gps.f_get_position(&flat, &flon, &age);
  Serial.print(flat, 4); Serial.print(", ");
  Serial.println(flon, 4);
}

// Feed data as it becomes available
bool feedgps() {
  while (nss.available()) {
    if (gps.encode(nss.read()))
      return true;
  }
  return false;
}
```

Important! The Arduino Pre-release versions of the IDE of the sketch uses the following object libraries, which must be downloaded separately, and placed inside the Arduino Library folder:

- NewSoftSerial [7]
- TinyGPS [8]

See instructions for adding Arduino object libraries [9] for additional information.

To view the results of the demonstration, after uploading is complete click the Serial Monitor icon in the Arduino IDE. This displays the Serial Monitor window. Momentarily depress the Reset button on the Arduino board to restart the sketch.

For More Information

See NMEA data [10] for more information on NMEA sentence structures.

- Visit the PMB-648 (#28500) product page for the for the device’s data sheet
- The Parallax VPN 1513 GPS Receiver Module (#28506) provides an enhanced method for retrieving GPS data. The module performs the sentence parsing for you, and outputs individual navigation information upon request.

Source URL: http://learn.parallax.com/KickStart/28500

Links:
[1] http://learn.parallax.com/KickStart/28500
[2] http://learn.parallax.com/sites/default/files/content/kickstart/code/GPS-BS2.bs2
[3] http://learn.parallax.com/sites/default/files/content/kickstart/code/GPS-Propeller.zip
[4] http://learn.parallax.com/KickStart/PropellerObjects
[5] http://learn.parallax.com/sites/default/files/content/kickstart/code/GPS_Arduino_10.zip
[6] http://learn.parallax.com/sites/default/files/content/kickstart/code/GPS_Arduino.zip
[7] http://arduiniana.org/libraries/NewSoftSerial
[8] http://arduiniana.org/libraries/tinygps
[9] http://learn.parallax.com/KickStart/ArduinoLibraries
[10] http://www.gpsinformation.org/dale/nmea.htm