1300 Henley Court
Pullman, WA 99163
509.334.6306
www.digilentinc.com

# PmodGPS Library Reference Manual

Revised July 2, 2014

## Overview

The PmodGPS library provides an interface to a GlobalTop Gms-u1LP PmodGPS antenna. Once the module connects to at least four satellites, it sends several messages containing GPS data to the microcontroller. This data is then formatted into structs that can be accessed at will.

## 1 Library Operation

### 1.1 Library Interface

The header file PmodGPS.h defines the structs used in the PmodGPS library, as well as a PmodGPS class containing the functions used to format the structs, which are stored inside the class. To instantiate a GPS object, simply include the library and instantiate a GPS object.

### 1.2 GPS Initialization

The PmodGPS is initialized by calling the function:

```
GPSinit(<UART Serial#>, <baud rate>, <3DF pin>, <1PPS pin>) ;
```

This function sets the pin directions and starts serial communications. If the reset pin is used, it also sends the PmodGPS through a soft reset.

After the PmodGPS pins are set and communications are started, the PmodGPS will begin to send sentences through the serial port. These sentences are based on National Marine Electronics Association (NMEA) protocols. These sentences are sent every second, and must be received by calling the getData() function periodically. *getData() should be called whenever there is data available on the Serial port to avoid data corruption.*

Four general data sentences will be sent out every ping. Every 5 pings, several (3 or 4) additional sentences will be sent containing Satellite info (GSV). getData() will return an enumerated NMEA value. This NMEA value can be used to check what type of sentence was received (and what data was updated).

The PmodGPS must first find at least 4 satellites before it will send out valuable GPS data. Calling isFixed() will return true if the PmodGPS has a position fix.

A table of the output sentences can be found in the PmodGPS reference manual.

# 2 Enumerated Types

**NMEA:** Enumerates the types of sentences sent from the PmodGPS, these are the last 3 characters in the message ID.

| | | | | | |
|---|---|---|---|---|---|
| Invalid=0 | GGA | GSA | GSV | RMC | VTG |

# 3 Structs

**Note**: Data is stored as either a char or a char array. These values can be retrieved and converted with their corresponding get functions.

**GGAdata**

| Member | Description |
|---|---|
| UTC | UTC Time |
| LAT | Latitude |
| NS | North or south indicator |
| LONG | Longitude |
| EW | East or west indicator |
| PFI | Position fixed indicator |
| NUMSAT | Number of satellites used |
| HDOP | HDOP |
| ALT | MSL Altitude |
| AUNIT | Altitude units |
| GSEP | Geoidal separation |
| GUNIT | Units |
| AODC | Time since last DGPS update |
| CHECKSUM | Checksum |

**GSAdata**

| Member | Description |
|---|---|
| MODE1 | 'M' for manual: force to operate in 2D or 3D mode, 'A' for 2D automatic: switches automatically between 2D/3D |
| MODE2 | 1-Fix not available, 2-2D (<4 SVs used), 3-3D (4 or more SVs used) |
| SAT1 | Satellite used (SV) (channel 1) |
| SAT2 | Satellite used (SV) (channel 2) |
| SAT3 | Satellite used (SV) (channel 3) |
| SAT4 | Satellite used (SV) (channel 4) |
| SAT5 | Satellite used (SV) (channel 5) |
| SAT6 | Satellite used (SV) (channel 6) |
| SAT7 | Satellite used (SV) (channel 7) |
| SAT8 | Satellite used (SV) (channel 8) |
| SAT9 | Satellite used (SV) (channel 9) |
| SAT10 | Satellite used (SV) (channel 10) |
| SAT11 | Satellite used (SV) (channel 11) |
| SAT12 | Satellite used (SV) (channel 12) |
| PDOP | Position dilution of precision |
| HDOP | Horizontal dilution of precision |
| VDOP | Vertical dilution of precision |
| CHECKSUM | Checksum |

## VTGdata

| Member | Description |
|---|---|
| COURSE_T | Measured heading |
| REF_T | True(T) |
| COURSE_M | Measured heading |
| REF_M | Magnetic(M) |
| SPD_N | Measured speed |
| UNIT_N | Knots(N) |
| SPD_KM | Measured speed |
| UNIT_KM | Km/hr(K) |
| MODE | A: Autonomous mode, D: Differential mode, E: Estimated mode |
| CHECKSUM | Checksum |

## RMCdata

| Member | Description |
|---|---|
| UTC | UTC Time |
| STAT | Status: A = Data valid, V = data not valid |
| LAT | Latitude |
| NS | North or south indicator |
| LONG | Longitude |
| EW | East or west indicator |
| SOG | Speed over ground (knots) |
| COG | Course over ground (degrees) |
| DATE | Date |
| MVAR | Magnetic variation (degrees) |
| MVARDIR | Magnetic variation direction |
| MODE | A: Autonomous mode, D: Differential mode, E: Estimated mode |
| CHECKSUM | Checksum |

## GSVdata

| Member | Description |
|---|---|
| NUMM | Number of messages |
| MESNUM | Message number |
| SATVIEW | Satellites in view |
| SAT[15] | Satellite info struct array (see below) |
| CHECKSUM | Checksum |

## GSVdata.SAT

| Member | Description |
|---|---|
| ID | Satellite identification |
| ALT | Satellite altitude in degrees |
| AZM | Satellite azimuth, degrees from true north |
| SNR | Signal to noise ratio |

# 4 GPS Library Functions

## 4.1 Public Functions

**GPSinit(HardwareSerial &serPort, unsigned long baud, uint8_t DF, uint8_t PPS)**

Parameters:

| | |
|---|---|
| serPort | The serial UART interface to use (Serial or Serial1) |
| baud | The baud rate of the serial port |
| DF | 3DS pin number |
| PPS | 1PPS pin number |

Return Value:

- None

Starts the serial port using the specified baud rate, and sets the pin direction of the 3DS pin and the 1PPS pin to inputs.

**GPSinit(HardwareSerial &serPort, unsigned long baud, uint8_t DF, uint8_t PPS, uint8_t RST)**

Parameters:

| | |
|---|---|
| serPort | The serial UART interface to use (Serial or Serial1) |
| baud | The baud rate of the serial port |
| DF | 3DS pin number |
| PPS | 1PPS pin number |
| RST | RST pin number |

Return Value:

- None

Starts the serial port with the specified baud rate, and then sets 3DS and 1PPS pins to inputs and sets RST pin to output.

**getData(HardwareSerial &serPort)**

Parameters:

| | |
|---|---|
| serPort | The serial UART interface to use (Serial or Serial1) |

Return Value:

| | |
|---|---|
| NMEA mode | The type of sentence that was received |

If there is a message from the PmodGPS, the function collects it. When the message is collected, this function finds out what type of sentence was sent and formats the data into the corresponding struct. The function returns the type of sentence.

## 4.2 Get Functions for Structs

These functions can be used to retrieve the full data structs containing char and char strings.

**getGGA()**

Return Value:

- GGA_DATA* GGAdata The GGAdata struct pointer within the class

A get function for the private pointer to the GGAdata struct.

**getGSA()**

Return Value:

- GSA_DATA* GSAdata The GSAdata struct pointer within the class

A get function for the private pointer to the GSAdata struct.

**getGSV()**

Return Value:

- GSV_DATA* GSVdata The GSVdata struct pointer within the class

A get function for the private pointer to the GSVdata struct.

**getRMC()**

Return Value:

- RMC_DATA* RMCdata The RMCdata struct pointer within the class

A get function for the private pointer to the RMCdata struct.

**getVTG()**

Return Value:

- VTG_DATA* VTGdata The VTGdata struct pointer within the class

A get function for the private pointer to the VTGdata struct.

## 4.3 Get Functions for Data in Structs

Data in structs are originally in either char or char string form, some get functions return these in integer or double form.

**bool isFixed()**

Return Value:

- True, if PmodGPS fixed. False, if otherwise.

A check to see if the PmodGPS has a position fix.

**char* getLatitude()**

Return Value:

- char GGAdata.LAT[14] A formatted string representing latitude

A get function for the formatted latitude string in GGAdata.

**char* getLongitude()**

Return Value:

- char GGAdata.LONG[15] A formatted string representing longitude

A get function for the formatted longitude string in GGAdata.

**double getDate()**

Return Value:

- double heading The direction(in degrees) the PmodGPS is heading (True north)

A get function that gets GGAdata.COURSE_T and returns its value in double form.

**double getAltitude()**

Return Value:

- Double altitude Altitude

A get function that gets GGAdata.ALT and returns its value in double form.

**char* getAltitudeString()**

Return Value:

- char altitude[11] A formatted string showing altitude and units

A get function that gets GGAdata.ALT and formats it into a string with units.

**double* getTime()**

Return Value:

- Double UTC        Double value representing UTC time

A get function that returns a floating point value of UTC time.

**int getNumSats()**

Return Value:

- int NumSats        Number of satellites the PmodGPS is connected to

A get function that gets the number of satellites the PmodGPS is connected to and returns it in integer form.

**double getPDOP()**

Return Value:

- double PDOP        Position dilution of position

A get function that gets GSAdata.PDOP and returns its value in double form.

**double getSpeedKnots()**

Return Value:

- double speed_n        Speed data in knots

A get function that gets VTGdata.SPD_N and returns its value in double form.

**double getSpeedKM()**

Return Value:

- double speed_km        Speed data in kilometers

A get function that gets GGAdata.SPD_KM and returns its value in double form.

**double getHeading()**

Return Value:

- double heading        The direction (in degrees) the PmodGPS is heading (True north)

A get function that gets GGAdata.COURSE_T and returns its value in double form.

**SATELLITE getSatelliteInfo()**

Return Value:

- SATELLITE sat[15]        The array of satellite structs containing satellite info

A get function that returns the array of Satellite info structs containing ID, ELV, AZM, and SNR.

## 4.4 Private Functions

**NMEA chooseMode (char recv[MAX_SIZE])**

Parameters:

- recv The full sentence received from the PmodGPS.

Return Value:

- NMEA mode The sentence format of the sentence

Reads the ID of the sentence and returns what type of sentence was sent.

**void formatXXX (char* data_array)**

XXX = <GGA, GSA, GSV, RMC, VTG>

Parameters:

- data_array The full sentence received from the PmodGPS.

Reads the sentence, tokenizes it using commas as delimiters, and formats the data into the corresponding data struct within the GPS class.