# Global Positioning System (GPS) Module

# Technical Data

DS-GPM
DS-GPMe
Firmware version 1

## Features

- Give your robot the ability to know where it is, how fast its moving and in what direction*.
- Fast 8-channel position acquisition with battery backup for fast restart.
- Simple register based data retrieval of latitude, longitude, heading, altitude, speed, time, date & satellites in view.
- Integral low power antenna (external on GPMe).
- Form factor identical to OOPic embedded control module (51mm x 89mm high quality PCB)
- Low power consumption
- RS232 or I$^2$C communication interface for simple connection to PC, OOPic, BASIC Stamp etc.
- Built in 4 line IO port and 4 input 8bit ADC port for local sensors.
- Raw GPS data output to NMEA 0183 standard.

## Description

The Designer Systems DS-GPM is a highly integrated Global Positioning System allowing your robotic application to determine its location on the earth's surface. Specifically targeted at OOPic / BASIC Stamp type embedded control applications and the Personal Computer user the GPM features I$^2$C and RS232 communication.

GPS data received by the DS-GPM is stored within internal registers which are updated once per second and include:

- Latitude (i.e. vertical)
- Longitude (i.e. horizontal)
- Altitude (metres)
- Time & date (UTC)
- Heading (True & Magnetic)
- Speed (kilometres per hour)
- Satellites detected

In addition the DS-GPM features an on-board fully configurable four line TTL IO port and a four line analogue input port with automatic measurement.

An auxiliary connection is also provided that delivers an accurate one pulse-per-second signal and the raw NMEA 0183 data.

## Applications

The DS-GPM has many applications in robotics, security and timing. For example the module could be used to send a rover to a particular position or the DS-GPMe + antenna be used to form a vehicle security solution in-conjunction with an embedded controller and GSM modem. Application notes for the OOPic controllers are provided.

COMMS MODULES

## Selection Guide

| Description | Part Number |
| --- | --- |
| Global Position System Module | DS-GPM |
| Global Position System Module with MCX antenna connection (no antenna included) | DS-GPMe |
| DS-GPMe magnetic-mount antenna with 3metre lead | DS-GPA1 |
| OOPic I$^2$C / power 'Y' cable 150mm | DS-WI2CCAB |
| PC (DB9) to GPM RS232 cable 1.5m | DS-W232CAB |

* Note: GPS information cannot be collected without a clear view of the sky.

## GPS basics

The heart of the DS-GPM is a Global Positioning System receiver module and antenna (antenna external in GPMe) that receive signals from satellites orbiting the earth.
There are 24 of these satellites, each sending its own unique signal to the earth's surface for pickup by any GPS receiver, which searches the sky for available satellites.
Upon detecting the satellites in view and their current position the receiver uses the satellites with highest signal strength to calculate, using triangulation, the receiver's latitude, longitude & altitude** (position).
Latitude is measured in degrees and minutes either North or South of the equator.
Longitude is measured in degrees and minutes either West or East of an imaginary line drawn vertically through Greenwich in the UK.
Altitude is measured in metres above sea level.
For example the offices of Designer Systems in Truro, UK are located 50 degrees, 15.817 minutes North latitude and 5 degrees, 3.549 minutes West longitude.
Should the receiver also be moving, speed in kilometres per hour, and heading, in degrees true north and magnetic north, can also be determined.
To gain the best reception the GPM should be used outside with a good view of the sky. Trees and buildings will cause the GPS signals being received to degrade and positional/speed information may be lost. To greatly improve reception the GPM should be mounted above a metal base, the X4 rover is an idea platform due to its aluminium base.
** LLA format to WGS-84 ellipsoid.

## Operation

When power is applied to the GPM the unit immediately starts to search for satellites. The GPM can start in one of three (3) modes, as follows:

*Cold start mode:*
This mode only applies when the GPM has been powered-up for the first time after being removed from its packaging. As the GPM does not know where it is on the earth's surface, it starts to hunt for groups of satellites to determine its location. This process may take up to 30 minutes before positional information is available; it is suggested that a battery be connected and the unit left in the open air until the STATUS indicator starts to flash.

*Warm start mode:*
This mode applies to a GPM that has already been 'cold-started' and whose location has not changed significantly when powered up again or has been powered down for at least one (1) hour. Positional information is normally available again within 45 seconds of power re-application.

*Hot start mode:*
This mode applies when the GPM has been powered off for less than 60 minutes. Positional information is normally available again within 5-20 seconds of power re-application.

The warm and hot start power-up modes are possible due to an internal backup battery which powers the Real Time Clock (RTC) and almanac memory when external power is removed.

*STATUS indication…*

The STATUS indicator is used to provide visual feedback of the current GPM condition. There are three (3) conditions as follows:

| | |
|---|---|
| ON Steady | Power applied and no positional information. |
| Flashing slowly | Positional information received. |
| Flashing fast | GPM in motion. |

These conditions will change as the GPM moves around its location and under objects that may block the satellite signals.

## Power requirements

The DS-GPM may take the power necessary for operation (approx. 60mA) from an OOPic-R +5V supply, using a DS-ORI2CCAB, or from an external battery or power adaptor supplying between 6 and 16V DC, which is recommended if the OOPic or OOPic II is connected. The two pin DC power input connector marked '6 – 16V DC' is marked '+' & '-' which should be connected to positive and negative battery/power supply terminals respectively. *Warning: Misconnection of this connection may damage the DS-GPM.*

## Analogue input port

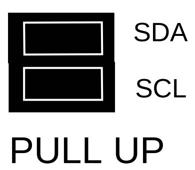
The DS-GPM features a four input 8bit Analogue to Digital Converter port 'AN0' to 'AN3' (see Fig. 4.0). Each input is automatically updated every 100mS from an external input voltage of 0 - 5V and the result stored in internal registers which can be read by the connected RS232 or I$^2$C device (see register details further on in this datasheet). The port also incorporates a ground and V+ bus that allows sensors to be directly connected and powered (see Fig. 4.0)
*Warning: These inputs are not over-voltage protected and should not be subjected to voltages over 5V.*

## Input/Output port

The DS-GPM also features a four line logic level IO port marked 'RAW GPS & IO' (8pin header, see Fig. 4.0). Each line can be individually configured as input or output with outputs capable of driving a maximum load of 20mA and inputs capable of reading the state of switches, sensors and other input devices. Full control of IO direction, input and output can be accessed by the connected RS232 or I$^2$C device (see register details further on in this datasheet).
*Warning: These inputs are not over-voltage protected and should not be subjected to voltages over 5V*

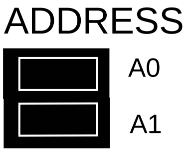
## I$^2$C connector(s)

The I$^2$C connector marked 'I$^2$C Input' (4pin header) allows connection to the OOPic controller – using a DS-WI2CCAB cable – or another I$^2$C Master device. There is also an additional connector marked 'I$^2$C Output' (4pin header) that can be used to connect to additional I$^2$C devices such as the DS-WCM, DS-LCDD2, DS-SXM16S etc.
The DS-GPM is fitted with pull-up jumpers that can be configured to provide the source current necessary for I$^2$C communication. The following jumpers should <u>ONLY</u> be set if the I$^2$C bus does not have existing pull-up's:



# I$^2$C communication

Up to four DS-GPM modules may be connected to the same I$^2$C bus and accessed individually using their own individual address.
The address is configured with the following jumpers:




The following table shows how the jumpers are placed for the different binary addresses:

| Address xx | A0 | A1 |
|---|---|---|
| 00 (default) | ON | ON |
| 01 | OFF | ON |
| 10 | ON | OFF |
| 11 | OFF | OFF |

The binary address (xx) above is used in conjunction with the device ID 11010xxD to form the complete device address i.e. if both jumpers are left connected (default) then the device address would be $1101000D_{binary}$.
The 'D' bit determines if a read or a write to the GPM is to be performed. If the 'D' bit is set '1' then a register read is performed or if clear '0' a register write.

To access individual registers a device write must be undertaken by the OOPic / I$^2$C Master which consists of a Start condition, device ID ('D' bit cleared), register to start write, one or more bytes of data to be written and a stop condition (see Figure 1.0 for I$^2$C write protocol).

There are 2 individual registers that can be written within the GPM that control local IO port setup and output as follows:

| | $N_7$ | $N_6$ | $N_5$ | $N_4$ | $N_3$ | $N_2$ | $N_1$ | $N_0$ |
|---|---|---|---|---|---|---|---|---|

GPM I2C address

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1. | 1 | 1 | 0 | 1 | 0 | X | X | 0 |

XX = Address select pins A1 & A0

Register address

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 2. | U | U | U | U | U | U | U | B |

B..B = 0 to 1
U..U = unused on this implementation

Local I/O port direction register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R0 | U | U | U | U | X | X | X | X |

X = 1 or 0 (1 = I/O is input, 0 = I/O is output)
U..U = unused on this implementation

Local I/O port output data register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R1 | U | U | U | U | X | X | X | X |

X = 1 or 0 (1 = output pin is high, 0= output pin is low)
U..U = unused on this implementation

To read individual data and status registers a device write then read must be undertaken by the OOPic / I$^2$C Master.
The write consists of a Start condition, device ID ('D' bit cleared), register to start read and a Stop condition.
This is followed by a read, which consists of a Start condition, device ID ('D' bit set), followed by data from the register specified and terminated with a Stop condition. The GPM also auto increments the register specified for every additional read requested by the Master I$^2$C device, which allows more than one register to be read in one transaction. This allows for example Register 0 to Register 5, current UTC time, to be read in one transaction (see Figure 1.1 for I$^2$C read protocol).
There are 94 individual registers that can be read within the GPM as follows:

| | $N_7$ | $N_6$ | $N_5$ | $N_4$ | $N_3$ | $N_2$ | $N_1$ | $N_0$ |
|---|---|---|---|---|---|---|---|---|

GPM Address

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1. | 1 | 1 | 0 | 1 | 0 | X | X | 1 |

XX = Address select pins

Hours tens register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R0 | X | X | X | X | X | H | H | H |

H..H = Tens of hours (24 hour clock UTC time)
X..X = not used

Hours units register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R1 | X | X | X | X | H | H | H | H |

H..H = Units of hours (24 hour clock UTC time)
X..X = not used

Minutes tens register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R2 | X | X | X | X | X | M | M | M |

M..M = Tens of minutes (UTC time)
X..X = not used

Minutes units register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R3 | X | X | X | X | M | M | M | M |

M..M = Units of minutes (UTC time)
X..X = not used

Seconds tens register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R4 | X | X | X | X | X | S | S | S |

S..S = Tens of seconds (UTC time)
X..X = not used

Seconds units register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R5 | X | X | X | X | S | S | S | S |

S..S = Units of seconds (UTC time)
X..X = not used

Day of month tens register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R6 | X | X | X | X | X | X | D | D |

D..D = Tens of day of month
X..X = not used

Day of month units register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R7 | X | X | X | X | D | D | D | D |

D..D = Units of day of month
X..X = not used

Month tens register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R8 | X | X | X | X | X | X | M | M |

M..M = Tens of months
X..X = not used

Month units register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R9 | X | X | X | X | M | M | M | M |

M..M = Units of months
X..X = not used

Years thousands register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R10 | X | X | X | X | X | X | Y | Y |

Y..Y = Thousands of years
X..X = not used

Years hundreds register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R11 | X | X | X | X | Y | Y | Y | Y |

Y..Y = Hundreds of years
X..X = not used

Years tens register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R12 | X | X | X | X | Y | Y | Y | Y |

Y..Y = Tens of years
X..X = not used

Years units register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R13 | X | X | X | X | Y | Y | Y | Y |

Y..Y = Units of years
X..X = not used

Latitude degrees tens register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R14 | X | X | X | X | D | D | D | D |

D..D = Tens of degrees
X..X = not used

Latitude degrees units register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R15 | X | X | X | X | D | D | D | D |

D..D = Units of degrees
X..X = not used

Latitude minutes tens register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R16 | X | X | X | X | M | M | M | M |

M..M = Tens of minutes
X..X = not used

Latitude minutes units register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R17 | X | X | X | X | M | M | M | M |

M..M = Units of minutes
X..X = not used

Latitude minutes tenths register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R18 | X | X | X | X | M | M | M | M |

M..M = Tenths of minutes
X..X = not used

Latitude minutes hundredths register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R19 | X | X | X | X | M | M | M | M |

M..M = Hundredths of minutes
X..X = not used

Latitude minutes thousandths register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R20 | X | X | X | X | M | M | M | M |

M..M = Thousandths of minutes
X..X = not used

Latitude direction character

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R21 | X | D | D | D | D | D | D | D |

D..D = ASCII Character (N = North, S = South)
X..X = not used

Longitude degrees hundreds register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R22 | X | X | X | X | X | X | D | D |

D..D = Hundreds of degrees
X..X = not used

Longitude degrees tens register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R23 | X | X | X | X | D | D | D | D |

D..D = Tens of degrees
X..X = not used

Longitude degrees units register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R24 | X | X | X | X | D | D | D | D |

D..D = Units of degrees
X..X = not used

Longitude minutes tens register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R25 | X | X | X | X | M | M | M | M |

M..M = Tens of minutes
X..X = not used

Longitude minutes units register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R26 | X | X | X | X | M | M | M | M |

M..M = Units of minutes
X..X = not used

Longitude minutes tenths register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R27 | X | X | X | X | M | M | M | M |

M..M = Tenths of minutes
X..X = not used

Longitude minutes hundredths register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R28 | X | X | X | X | M | M | M | M |

M..M = Hundredths of minutes
X..X = not used

Longitude minutes thousandths register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R29 | X | X | X | X | M | M | M | M |

M..M = Thousandths of minutes
X..X = not used

Longitude direction character

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R30 | X | D | D | D | D | D | D | D |

D..D = ASCII Character (W = West, E = East)
X..X = not used

GPS quality indicator

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R31 | X | X | X | X | X | X | D | D |

D..D = 0 - 2 (0 = No GPS, 1 = GPS, 2 = DGPS)
X..X = not used

Satellites in use tens register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R32 | X | X | X | X | X | X | S | S |

S..S = Tens of satellites in use
X..X = not used

Satellites in use units register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R33 | X | X | X | X | S | S | S | S |

S..S = Units of satellites in use
X..X = not used

HDOP units register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R34 | X | X | X | X | H | H | H | H |

H..H = Units of HDOP
X..X = not used

HDOP tenths register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R35 | X | X | X | X | H | H | H | H |

M..M = Tenths of HDOP
X..X = not used

HDOP hundredths register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R36 | X | X | X | X | H | H | H | H |

M..M = Hundredths of HDOP
X..X = not used

Altitude metres tens of thousands register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R37 | X | X | X | X | X | X | X | A |

A = Tens of thousands of metres
X..X = not used

Altitude metres thousands register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R38 | X | X | X | X | A | A | A | A |

A..A = Thousands of metres
X..X = not used

Altitude metres hundreds register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R39 | X | X | X | X | A | A | A | A |

A..A = Hundreds of metres
X..X = not used

Altitude metres tens register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R40 | X | X | X | X | A | A | A | A |

A..A = Tens of metres
X..X = not used

Altitude metres units register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R41 | X | X | X | X | A | A | A | A |

A..A = Units of metres
X..X = not used

Heading degrees (true North) hundreds register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R42 | X | X | X | X | X | X | H | H |

H..H = Hundreds of degrees
X..X = not used

Heading degrees (true North) tens register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R43 | X | X | X | X | H | H | H | H |

H..H = Tens of degrees
X..X = not used

Heading degrees (true North) units register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R44 | X | X | X | X | H | H | H | H |

H..H = Units of degrees
X..X = not used

Heading degrees (true North) tenths register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R45 | X | X | X | X | H | H | H | H |

H..H = Tenths of degrees
X..X = not used

Heading degrees (Magnetic North) hundreds register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R46 | X | X | X | X | X | X | H | H |

H..H = Hundreds of degrees
X..X = not used

Heading degrees (Magnetic North) tens register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R47 | X | X | X | X | H | H | H | H |

H..H = Tens of degrees
X..X = not used

Heading degrees (Magnetic North) units register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R48 | X | X | X | X | H | H | H | H |

H..H = Units of degrees
X..X = not used

Heading degrees (Magnetic North) tenths register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R49 | X | X | X | X | H | H | H | H |

H..H = Tenths of degrees
X..X = not used

Speed hundreds register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R50 | X | X | X | X | X | X | S | S |

S..S = Hundreds of kilometres per hour
X..X = not used

Speed tens register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R51 | X | X | X | X | S | S | S | S |

S..S = Tens of kilometres per hour
X..X = not used

Speed units register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R52 | X | X | X | X | S | S | S | S |

S..S = Units of kilometres per hour
X..X = not used

Speed tenths register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R53 | X | X | X | X | S | S | S | S |

S..S = Tenths of kilometres per hour
X..X = not used

GPS Mode character

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R54 | X | D | D | D | D | D | D | D |

D..D = ASCII character (A = Autonomous Mode, D = Differential Mode, E = Estimated (dead reckoning) Mode, M = Manual Input Mode, S = Simulated Mode, N = Data Not Valid

Satellites in view tens register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R55 | X | X | X | X | X | X | S | S |

S..S = Tens of satellites in view
X..X = not used

Satellites in view units register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R56 | X | X | X | X | S | S | S | S |

S..S = Units of satellites in view
X..X = not used

Satellite 1 ID number tens register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R57 | X | X | X | X | X | X | S | S |

S..S = Tens of satellite ID number
X..X = not used

Satellite 1 ID number units register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R58 | X | X | X | X | S | S | S | S |

S..S = Units of satellite ID number
X..X = not used

Satellite 1 signal level tens register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R59 | X | X | X | X | L | L | L | L |

L..L = Tens of satellite signal level
X..X = not used

Satellite 1 signal level units register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R60 | X | X | X | X | L | L | L | L |

L..L = Units of satellite signal level
X..X = not used

Satellite 2 ID number tens register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R61 | X | X | X | X | X | X | S | S |

S..S = Tens of satellite ID number
X..X = not used

Satellite 2 ID number units register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R62 | X | X | X | X | S | S | S | S |

S..S = Units of satellite ID number
X..X = not used

Satellite 2 signal level tens register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R63 | X | X | X | X | L | L | L | L |

L..L = Tens of satellite signal level
X..X = not used

Satellite 2 signal level units register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R64 | X | X | X | X | L | L | L | L |

L..L = Units of satellite signal level
X..X = not used

Satellite 3 ID number tens register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R65 | X | X | X | X | X | X | S | S |

S..S = Tens of satellite ID number
X..X = not used

Satellite 3 ID number units register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R66 | X | X | X | X | S | S | S | S |

S..S = Units of satellite ID number
X..X = not used

Satellite 3 signal level tens register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R67 | X | X | X | X | L | L | L | L |

L..L = Tens of satellite signal level
X..X = not used

Satellite 3 signal level units register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R68 | X | X | X | X | L | L | L | L |

L..L = Units of satellite signal level
X..X = not used

Satellite 4 ID number tens register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R69 | X | X | X | X | X | X | S | S |

S..S = Tens of satellite ID number
X..X = not used

Satellite 4 ID number units register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R70 | X | X | X | X | S | S | S | S |

S..S = Units of satellite ID number
X..X = not used

Satellite 4 signal level tens register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R71 | X | X | X | X | L | L | L | L |

L..L = Tens of satellite signal level
X..X = not used

Satellite 4 signal level units register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R72 | X | X | X | X | L | L | L | L |

L..L = Units of satellite signal level
X..X = not used

Satellite 5 ID number tens register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R73 | X | X | X | X | X | X | S | S |

S..S = Tens of satellite ID number
X..X = not used

Satellite 5 ID number units register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R74 | X | X | X | X | S | S | S | S |

S..S = Units of satellite ID number
X..X = not used

Satellite 5 signal level tens register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R75 | X | X | X | X | L | L | L | L |

L..L = Tens of satellite signal level
X..X = not used

Satellite 5 signal level units register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R76 | X | X | X | X | L | L | L | L |

L..L = Units of satellite signal level
X..X = not used

Satellite 6 ID number tens register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R77 | X | X | X | X | X | X | S | S |

S..S = Tens of satellite ID number
X..X = not used

Satellite 6 ID number units register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R78 | X | X | X | X | S | S | S | S |

S..S = Units of satellite ID number
X..X = not used

Satellite 6 signal level tens register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R79 | X | X | X | X | L | L | L | L |

L..L = Tens of satellite signal level
X..X = not used

Satellite 6 signal level units register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R80 | X | X | X | X | L | L | L | L |

L..L = Units of satellite signal level
X..X = not used

Satellite 7 ID number tens register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R81 | X | X | X | X | X | X | S | S |

S..S = Tens of satellite ID number
X..X = not used

Satellite 7 ID number units register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R82 | X | X | X | X | S | S | S | S |

S..S = Units of satellite ID number
X..X = not used

Satellite 7 signal level tens register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R83 | X | X | X | X | L | L | L | L |

L..L = Tens of satellite signal level
X..X = not used

Satellite 7 signal level units register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R84 | X | X | X | X | L | L | L | L |

L..L = Units of satellite signal level
X..X = not used

Satellite 8 ID number tens register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R85 | X | X | X | X | X | X | S | S |

S..S = Tens of satellite ID number
X..X = not used

Satellite 8 ID number units register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R86 | X | X | X | X | S | S | S | S |

S..S = Units of satellite ID number
X..X = not used

Satellite 8 signal level tens register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R87 | X | X | X | X | L | L | L | L |

L..L = Tens of satellite signal level
X..X = not used

Satellite 8 signal level units register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R88 | X | X | X | X | L | L | L | L |

L..L = Units of satellite signal level
X..X = not used

Local analogue input AN0 value

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R89 | D | D | D | D | D | D | D | D |

D..D = 0 to 255 (Analogue input value for AN0 input)

Local analogue input AN1 value

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R90 | D | D | D | D | D | D | D | D |

D..D = 0 to 255 (Analogue input value for AN1 input)

Local analogue input AN2 value

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R91 | D | D | D | D | D | D | D | D |

D..D = 0 to 255 (Analogue input value for AN2 input)

Local analogue input AN3 value

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R92 | D | D | D | D | D | D | D | D |

D..D = 0 to 255 (Analogue input value for AN3 input)

Local I/O port input value

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R93 | X | X | X | X | D | D | D | D |

D = 1 or 0 (1 = input pin is high, 0= input pin is low)

DS-GPM Status

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R94 | U | U | B | A | V | V | V | V |

V..V = Firmware version number 1-15
A = Position found bit (0 = Not found, 1 = Found)
B = Motion bit (0 = Standstill, 1 = Moving)

Registers R0 to R88 may contain invalid data until satellite information has been gained and stored.

*Register restoration…*

All received data is formatted into decimal units (i.e. hundreds, tens & units) and stored in individual registers to facilitate either value or character restoration.
Value restoration can be undertaken by multiplying the required register by its multiplier e.g. to restore the value of register R0 'Hours tens' the register contents are multiplied by ten (10).
Character restoration, to allow the out to a PC via. RS232 or display of data on a LCD panel etc., can be undertaken by the addition of the constant value $48_{decimal}$, $30_{hex}$.

*UTC Time format…*

The standard GPS time coordinate system is called Universal Coordinated Time or UTC.
This time format replaced Greenwich Mean Time (GMT) in 1986 and is of the same value. Time zones relative to GMT should add or subtract a standard value to gain the correct time.

Example.
To read the complete time from registers 0 to 5 (Current time = 14:32:56, Device address = default) write:

'Point to register 0
Byte 1 (GPM Adr) $11010000_{binary}$
Byte 2 (Set register) $0_{decimal}$, $00_{hex}$

'Read register 0 - 5
Byte 1 (GPM Adr) $11010001_{binary}$
Byte 2 Hours tens $1_{decimal}$, $01_{hex}$
Byte 3 Hours units $4_{decimal}$, $04_{hex}$
Byte 4 Minutes tens $3_{decimal}$, $03_{hex}$
Byte 5 Minutes units $2_{decimal}$, $02_{hex}$
Byte 6 Seconds tens $5_{decimal}$, $05_{hex}$
Byte 7 Seconds units $6_{decimal}$, $06_{hex}$

See further on in this data sheet for a sample OOPic application.

## RS232 connection & setup

The RS232 connector marked 'RS232 COMMS' (6pin header) allows connection to a Personal Computer – using a DS-W232CAB cable – or serial RS232 device.

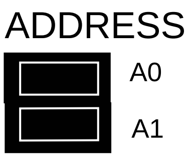
Connection is via. a six (6) pin vertical header, pinned as follows:

| Header connection | Description |
|---|---|
| 1 | Serial output |
| 2 | Serial input |
| 3 | Ground (AN0-3 GD) |
| 4 | Serial output |
| 5 | Serial input |
| 6 | AN0-3 V+ connection |

The connection is pinned to allow the maximum of four (4) DS-GPM modules to be connected in a 'daisy chain' configuration similar to an $I^2C$ bus.
This is accomplished by connecting all 'Serial inputs' together, all 'Serial outputs' together (see Figure 3.0) and changing the GPM 'ADDRESS' links on each module.
The RS232 connection will support any modern RS232D/E compliant device and must be configured for 9600 baud, 8 data bits, 1 stop bit, no parity, no handshaking required.

## RS232 communication

Up to four GPM modules may be connected to the same RS232 device and accessed individually using their own individual address. The address is configured with the following jumpers:




The following table shows how the jumpers are placed for the different binary addresses:

| Address xx | A0 | A1 |
|---|---|---|
| 00 (default) | ON | ON |
| 01 | OFF | ON |
| 10 | ON | OFF |
| 11 | OFF | OFF |

The binary address (xx) above is used in conjunction with the device ID 11010xxD to form the complete device address i.e. if both jumpers are left connected (default) then the device address would be $1101000D_{binary}$.
The 'D' bit determines if a read or a write to the GPM is to be performed. If the 'D' bit is set '1' then a register read is performed or if clear '0' a register write.

To write to individual registers a device write must be undertaken by the RS232 device which consists of a Prefix character, device ID ('D' bit cleared), register to start write, one or more bytes of data to be written and two terminator characters.

There are 2 individual registers that can be written within the GPM that control local IO port setup and output as follows:

| | $N_7$ | $N_6$ | $N_5$ | $N_4$ | $N_3$ | $N_2$ | $N_1$ | $N_0$ |
|---|---|---|---|---|---|---|---|---|
| RS232 command prefix | | | | | | | | |
| 1. | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| GPM RS232 address | | | | | | | | |
| 2. | 1 | 1 | 0 | 1 | 0 | X | X | 0 |
| Register address | | | | | | | | |
| 3. | U | U | U | U | U | U | U | B |
| Local I/O port direction register | | | | | | | | |
| R0 | U | U | U | U | X | X | X | X |
| Local I/O port output data register | | | | | | | | |
| R1 | U | U | U | U | X | X | X | X |
| RS232 command terminator | | | | | | | | |
| n. | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| RS232 command terminator | | | | | | | | |
| n. | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |

Notes:
1. ASCII character ' [ '
2. XX = Address select pins A1 & A0
3. B..B = 0 to 1; U..U = unused on this implementation
R0: X = 1 or 0 (1 = I/O is input, 0 = I/O is output); U..U = unused on this implementation
R1: X = 1 or 0 (1 = output pin is high, 0= output pin is low); U..U = unused on this implementation
n. ASCII character ' ] '
n. ASCII character ' ] '

To read the individual data and status registers a device read must be requested by the RS232 device. The read request consists of a Prefix character, device ID ('D' bit set), register to read and two terminator characters as follows:

| | $N_7$ | $N_6$ | $N_5$ | $N_4$ | $N_3$ | $N_2$ | $N_1$ | $N_0$ |
|---|---|---|---|---|---|---|---|---|
| RS232 command prefix | | | | | | | | |
| 1. | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| GPM RS232 address (Read requested) | | | | | | | | |
| 2. | 1 | 1 | 0 | 1 | 0 | X | X | 1 |
| Register to read | | | | | | | | |
| 3. | U | B | B | B | B | B | B | B |
| RS232 command terminator | | | | | | | | |
| n. | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| RS232 command terminator | | | | | | | | |
| n. | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |

Notes:
1. ASCII character ' [ '
2. XX = Address select pins A1 & A0
3. B..B = 0 to 94; U..U = unused on this implementation
n. ASCII character ' ] '
n. ASCII character ' ] '

*Note: The register number to read is the same as that defined for the $I^2C$ interface.*

Upon receipt of the above command the GPM will reply with a response packet containing the contents of the requested register as follows:

| | $N_7$ | $N_6$ | $N_5$ | $N_4$ | $N_3$ | $N_2$ | $N_1$ | $N_0$ |
|---|---|---|---|---|---|---|---|---|
| RS232 command prefix | | | | | | | | |
| 1. | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| ASCII character ' [ ' | | | | | | | | |
| GPM RS232 address value returned from | | | | | | | | |
| 2. | 1 | 1 | 0 | 1 | 0 | X | X | 0 |
| XX = Address select pins A1 & A0 | | | | | | | | |
| Register data value | | | | | | | | |
| 3. | D | D | D | D | D | D | D | D |
| D..D = 0 to 254 (If value = 255 then error) | | | | | | | | |
| RS232 command terminator | | | | | | | | |
| 4. | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| ASCII character ' ] | | | | | | | | |

*Note: If data from an invalid register is requested the returned register value is set to $255_{decimal}$, $FF_{hex}$.*

*Common register set access…*

To speed register access for common register sets (i.e. complete time & date) the GPM also supports the return of multi-value packets. This is accomplished by specifying a 'Register to read' value of 128 to 132 which requests the following Register sets:

128 – Time & Date (R0-R13)
129 – Position & altitude (R14-R41)
130 – Heading & speed (R42-R54)
131 – Satellites in view (R55-R88)
132 – IO and ADC values (R89-R94)

Example:
To setup IO lines 0 & 1 as inputs, lines 2 & 3 as outputs and read IO line 0 & 1 status (Device address = default) write:

[ #208 #0 #3 ] ]

Where #nn is a value not characters.

Then write:

[ #209 #93 ] ]

The GPM will respond to the request with the following:

[ #208 #x ]

where x = the status of the IO lines (0-3).

Example:
To request the return of the current GPS mode character register R54 (Device address = default) write:

[ #209 #54 ] ]

Where #nn is a value not characters.

The GPM will respond to the request with the following:

[ #208 A ]

where 'A' is the returned mode character.

Example:
To request the return of the current time and date register set (Device address = default) write:

[ #209 #128 ] ]

Where #nn is a value not characters.
The GPM will respond to the request with the following:

[ #208 #1 #4 #5 #5 #2 #7 #2 #4 #0 #7 #2 #0 #0 #2 ]

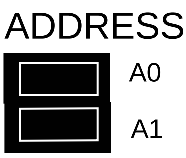
where the returned values correspond to the time/date of 14:55:27, 24 day, 07 month, 2002 year.

## SCP communication

The Savage Innovations SCP allows a remote PC, Pocket PC, Palm Pilot, or any other device with a serial port to control and read the GPM. The serial protocol is as follows:

9600 baud (bps), 8 Data bits
1 Stop Bit, No Parity
No handshaking (if configurable)

Up to four DS-GPM units may be connected to the same RS232 port and accessed individually with their own node address. The node address is configured with the following jumpers:




The following table shows how the jumpers are placed for the different node addresses:

| Node Address | A0 | A1 |
|---|---|---|
| '0' (default) | ON | ON |
| '1' | OFF | ON |
| '2' | ON | OFF |
| '3' | OFF | OFF |

The entire character set used by SCP is composed of human readable characters so that a serial terminal program can be used to manually read GPS information. The following commands are only briefly described as the full SCP is not within the scope of this data sheet, a full explanation being available from Savage Innovations.

*To enable SCP:*
Send "\0V" replace 0 with node address 0 - 3
Receive "v" indicates GPM is functional.

*To set Memory type:*
Send "128H" 128 + number of registers to access 0 = 1, 1=2
Receive "h" confirms set.

*To set register location to start access:*
Send "15J" 15 is register number 0 to 31.
Receive "j" confirms set.
*To write to previously selected register:*
Send "80N" 80 is sample hexadecimal value to write to register (must be in two character notation)
Receive "n" confirms write.

*To read previously selected register:*
Send "M" Request register read.

Receive "80m" 80 is sample hexadecimal value.

*To read register location:*
Send "I" Request register location.
Receive "16i" 16 is current register location.

*To read memory type:*
Send "G" Request memory type.

Receive "128g" 128 is current memory type.

*To reset GPM:*
Send "W" GPM is reset.

No response generated.

Use the $I^2C$ register set for SCP.

*To query SCP buffer:*
Send "Q" Request SCP buffer contents.

Receive "124q" Characters in buffer are returned e.g. 124 followed by "q". This command does not effect the buffers contents.

*To disable SCP:*
Send "X" Request exit from SCP.

Receive "x" SCP has exited.

If command format or value is not correct then a "!" response will be received and the command will not be executed.

Example:
To enable SCP, setup memory type register, location register and read hours registers:

Send "\0V" Enable SCP.
Receive "v" SCP enabled.
Send "129H" Set memory type to 128 + 1=129 (2 registers to be accessed).
Receive "h" Command confirmed.

| | |
|---|---|
| Send “0J” | Set register location to 0 to allow access to R0 & R1. |
| Receive “j” | Command confirmed. |
| Send “M” | Request read of hours value registers. |
| Receive “0104m” | Hours tens value 01hex and hours units 04hex is returned. |

## Raw GPS port

The DS-GPM features a GPS port marked ‘RAW GPS & IO’ (8pin header) that provides raw NMEA 0183 GPS data strings in packets using the following serial protocol:

9600 baud (bps), 8 Data bits
1 Stop Bit, No Parity
No handshaking (if configurable)

The connection is pinned as follows:

| Header connection | Description |
|---|---|
| 1 | Pulse-Per-Second out |
| 2 | GPS control data input |
| 3 | GPS data output |
| 4 | Ground |
| 5 | I/O line 0 |
| 6 | I/O line 1 |
| 7 | I/O line 2 |
| 8 | I/O line 3 |

*WARNING: Connections 1 – 3 are 3.3V TTL levels and may be damaged if connected directly to 5V TTL signals without suitable level shifting. It is recommended that these connections are not used unless you have a good understanding of electronics.*

The DS-IM II, or any other MAX232 based unit, is a suitable interface, when powered from a 3.3V power supply, to allow the GPS data IO to be connected to a standard RS232 device such as a Personal Computer. For an application note please contact your distributor.

Electrical Characteristics ($T_A$ = 25°C Typical)

| Parameter | Minimum | Maximum | Units | Notes |
|---|---|---|---|---|
| Supply Voltage (+5V) (Vcc) | 4.5 | 5.5 | V | 1 |
| Supply Voltage (6-16V) | 6 | 16 | V | |
| Supply Current | 50 | 60 | mA | |
| RS232 TX data output level | 0.3 | Vcc-0.8V | V | |
| RS232 RX data input level | -15 | +15 | V | |
| RS232 speed | - | 9600 | bps | |
| GPS port TX data output level | 0.1 | 3.3 | V | |
| GPS port RX data input level | 0 | 3.3 | V | |
| GPS port speed | - | 9600 | bps | |
| GPS PPS output level | 0.1 | 3.3 | V | |
| GPS PPS duration | - | 4 | uS | |
| GPS PPS accuracy | - 95 | 95 | nS | |
| $I^2C$ speed | - | 400 | kHz | |
| $I^2C$ pull-up resistance | - | 4700 | Ω | 3 |
| GPS positional accuracy | 2.5 | 5 | Metres | |
| GPS frequency band | - | 1575.42 | MHz | 2 |
| GPS channels | - | 8 | | |
| ADC input voltage | 0 | Vcc | V | |
| ADC measurement cycle | - | 100 | mS | |
| IO line output voltage | 0.3 | Vcc-0.8V | V | |
| IO line output current | - | 20 | mA | |
| IO line input voltage | 0 | Vcc+0.3V | V | |

## Absolute Maximum Ratings

| Parameter | Minimum | Maximum | Units |
|---|---|---|---|
| Supply Voltage (+5V) | -0.5 | +6 | V |
| Supply Voltage (6-16V) | -0.5 | +20 | V |

## Environmental

| Parameter | Minimum | Maximum | Units |
|---|---|---|---|
| Operating Temperature | 0 | 70 | °C |
| Storage Temperature | -10 | 80 | °C |
| Humidity | 0 | 80 | % |
| Dimensions | Length 89mm, Width 51mm, Height 30mm | | |
| Weight | 50g | | |
| Immunity & emissions | EMC compliance to 89/336/EEC | | |

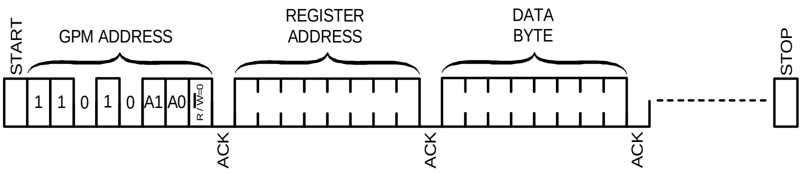
Notes:
1. Vcc is supply rail from OOPic or any other +5V supply.
2. L1 frequency, C/A code (Standard Positioning Service)
3. Value given is to Vcc when activated with appropriate jumpers.

# Example files for OOPic communication:

The following BASIC code can be compiled and downloaded to the OOPic, OOPic II or OOPic-R to allow the acquisition of time/date and heading/speed information from the DS-GPM and its display on a RS232 device such as a PC, Palm Pilot etc.
Please see the DS-GPM demonstration diskette for the following code sample.

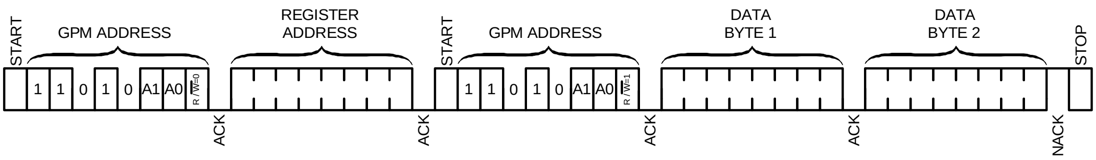
```
' Example : DS-GPM information retrieval demonstration
'
' Platform : OOPic, OOPic II & OOPic-R
'
' Description : Retrieves Time, Date, Heading and Speed information from the DS-GPM, stores
' it and outputs it to the OOPic RS232 port for display on a connected Personal Computer through
' a DS-IM II (RS232 port on OOPic-R).
'
' Created : 8/10/02           Revision : 1.00               Written by : Designer Systems
'
Dim GPM As New oi2c
Dim RS232 As New oSerial
Dim RTS As New oDio1
Dim Hours As New oByte                          'Hours storage register
Dim Minutes As New oByte                        'Minutes storage register
Dim Seconds As New oByte                        'Seconds storage register
Dim Day As New oByte                            'Day storage register
Dim Month As New oByte                          'Month storage register
Dim Year As New oWord                           'Year storage register
Dim Heading As New oWord                        'Heading storage register (1 degree accuracy)
Dim Speed As New oWord                          'Speed storage register (1 kmh accuracy)

Sub Main()

          oopic.delay = 400                     'Wait 4 seconds for GPM to initialise
          Const GPMAdr = &h68                   'GPM A0 & A1 jumpers ON (Range &h68-&h6B)
          RS232.Mode = 0
          RS232.Baud = cv9600
          RS232.Operate = cvOn                  'Setup RS232 port 9600 baud
          RTS.IOline = 24
          RTS.Direction = cvOutput              'Ensure RTS line is high (ready for communication)
          RTS = cvOff
          Call SetUpGPM                         'Setup GPM ready for information retrieval

          do
          '
          ' Stores time, date, heading and speed in registers defined above
          '
          call      StoreGPTD                   'Store time and date
          call      StoreGPHS                   'Store heading and speed

          ' Send time, date, heading & speed to RS232 port as displayable output
          '
          GPM.Location = 0                      'Point to R0
          RS232.String = "Time = "+chr$(GPM+48)+chr$(GPM+48)+":"+chr$(GPM+48)+chr$(GPM+48)+":"
          +chr$(GPM+48)+chr$(GPM+48)+chr$(&h0d)+chr$(&h0a)
          RS232.String = "Date = "+chr$(GPM+48)+chr$(GPM+48)+"/"+chr$(GPM+48)+chr$(GPM+48)+"/"
          +chr$(GPM+48)+chr$(GPM+48)+chr$(GPM+48)+chr$(GPM+48)+chr$(&h0d)+chr$(&h0a)

          GPM.Location = 46                     'Point to R46
          RS232.String = "Heading = "+chr$(GPM+48)+chr$(GPM+48)+chr$(GPM+48)+"."+chr$(GPM+48)+" degrees"+chr$(&h0d)+chr$(&h0a)
          RS232.String = "Speed = "+chr$(GPM+48)+chr$(GPM+48)+chr$(GPM+48)+"."+chr$(GPM+48)+" kmh"+chr$(&h0d)+chr$(&h0a)+chr$(&h0a)

          oopic.delay = 200                     'Wait 2 seconds

          loop

End Sub
'
' Subroutine to setup I2C communication to DS-GPM
'
Sub SetUpGPM()

          'Set the DS-GPM I2C address shifted right by 1 bit

          GPM.Node = GPMAdr                     'Setup I2C address for GPM

          'Setup I2C addressing to GPM

          GPM.Width = cv8bit                    'Control Info is 1-byte
          GPM.Mode = cv10bit                    'I2C mode is 10-Bit Addressing
          GPM.NoInc = cvFalse                   'Increment on every read/write

End Sub
'
' Subroutine to store time and date as values
'
Sub StoreGPTD()

          GPM.Location = 0                      'Start at R0
          Hours = (GPM*10)+GPM                  'Store hours
          Minutes = (GPM*10)+GPM                'Store minutes
          Seconds = (GPM*10)+GPM                'Store seconds

          Day = (GPM*10)+GPM                    'Store day
          Month = (GPM*10)+GPM                  'Store month
          Year = (GPM*1000)+(GPM*100)+(GPM*10)+GPM 'Store year

End Sub
'
' Subroutine to store heading and speed as values
'
Sub StoreGPHS()

          GPM.Location = 46                     'Start at R46
          Heading = (GPM*100)+(GPM*10)+GPM      'Store heading in degrees
          Speed = (GPM*100)+(GPM*10)+GPM        'Store speed in kmh

End Sub
```

Figure 1.0 ($I^2C$ write protocol)




Multiple bytes may be written before the ‘STOP’ condition. Data is written into registers starting at ‘REGISTER ADDRESS’, then ‘REGISTER ADDRESS’ +1, then ‘REGISTER ADDRESS’ +2 etc.
Each byte transfer is acknowledged ‘ACK’ by the GPM until the ‘STOP’ condition.

Figure 1.1 ($I^2C$ read protocol)




‘DATA BYTE 1 & 2’ are register values returned from the GPM. Each byte written is acknowledged ‘ACK’ by the GPM , every byte read is acknowledged ‘ACK’ by the $I^2C$ Master. A Not-acknowledge ‘NACK’ condition is generated by the $I^2C$ Master when it has finished reading.

Figure 2.0 (Connection Schematic for OOPic (II) $I^2C$ communication)

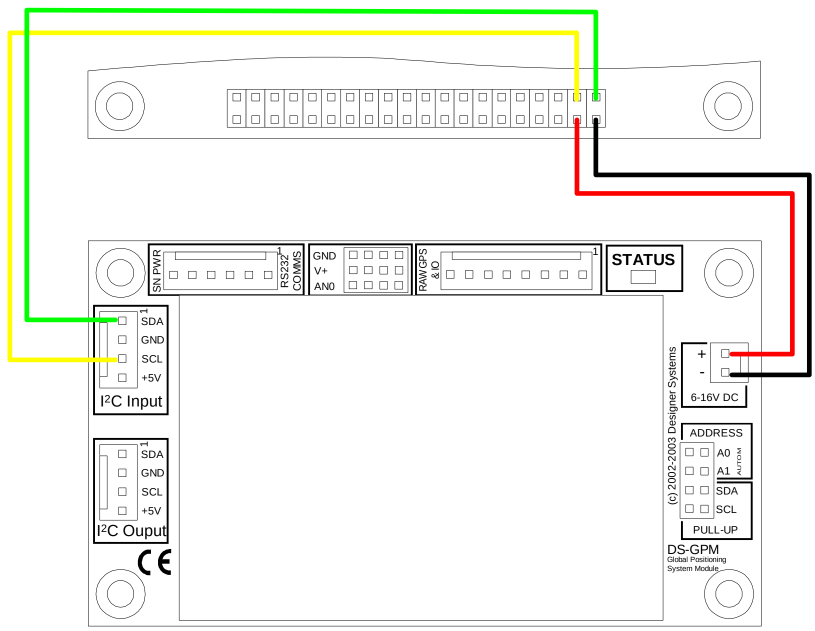



Figure 3.0 (Multi-drop RS232 connection)

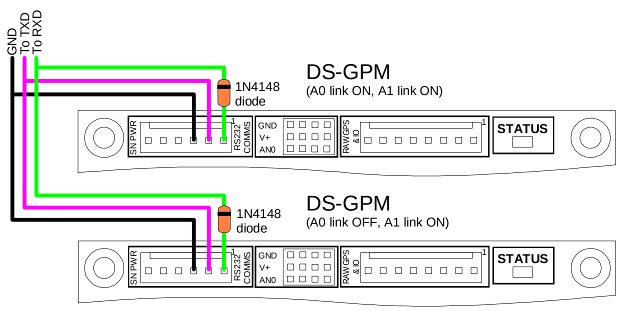
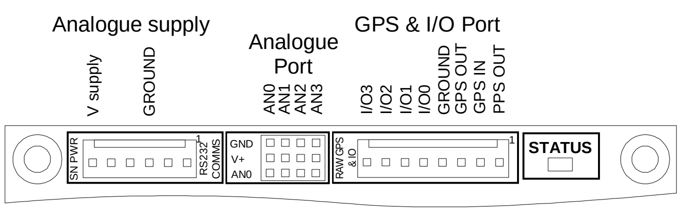
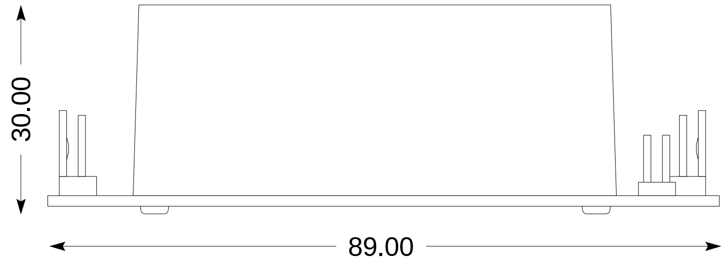
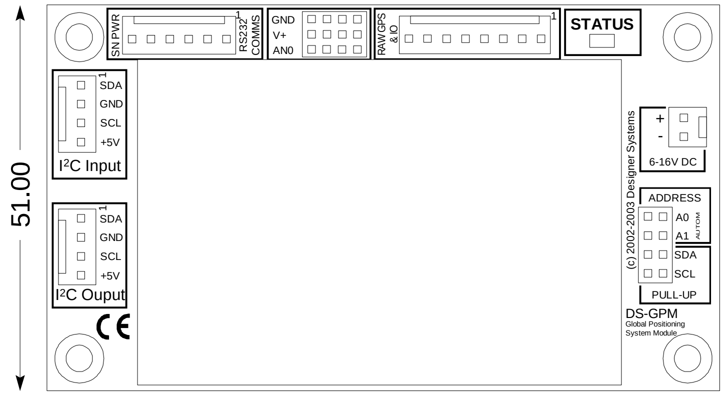



Figure 4.0 (I/O connections)



# Mechanical Specifications – Units millimetres



Page intentionally left blank

Page intentionally left blank