Microstack™ GPS
Getting Started

Harness the power of satellites to know exactly where you are, and globally, wirelessly synchronize your time to the accuracy of an atomic clock! http://www.microstack.org.uk

Typical Applications:

- Asset tracking
- High accuracy, remote time synchronisation
- Location aware services
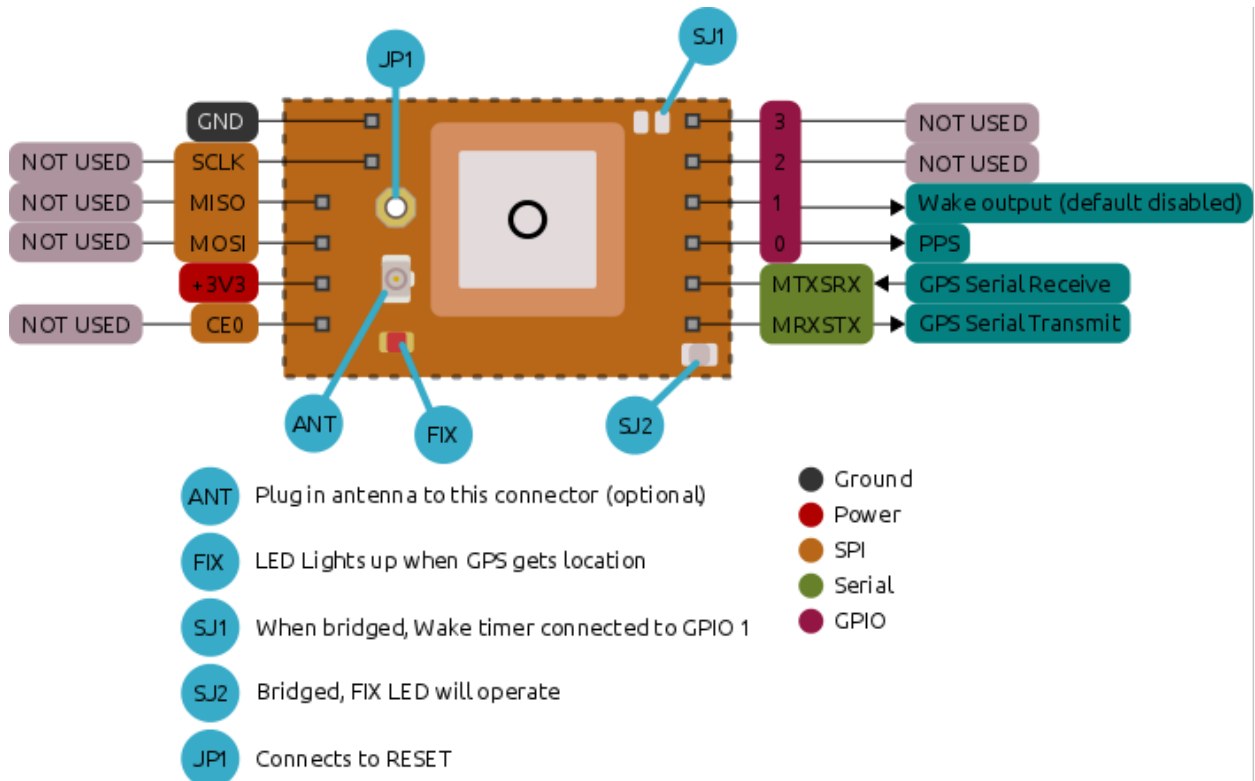- Geofencing
- Position data logging

Features:

- Low power (Acquisition mode 25mA, Tracking 20mA, down to 3mA with AlwaysLocate™ mode)
- Short time to fix
- Embedded, high sensitivity internal patch antenna
- Optional, external antenna connection
- Pulse per second (PPS) signal for precision timing
- High sensitivity receiver with Anti-Jamming
- DGPS, SBAS (WASS,EGNOS/MSAS/GAGAN) compatible
- Built in, self containted data logging
- Programmable automatic wake/sleep periods
- AlwaysLocate™ mode for automatic sleep during inactivity, and wake on movement to save power
- Can be used with Microstack™ adapter baseboards, including Raspberry Pi®
- Can be used directly in breadboards, prototyping and custom designs

Technical Specifications:

- Power 2.8V to 4.3V
- Acquisition mode 25mA
- Tracking mode 20mA
- AlwaysLocate™ mode 3mA
- Update interval 1Hz up to 10Hz
- Operating Temp -40° to 85°
- 1575.42 MHz receiver — 22 Channel Tracking, 66 Channel Acquisition
- Max altitude 18km
- Typical PPS accuracy <15ns
- Serial interface, 4800bps to 115200bps, 9600 by default

Tour of Hardware

ANT	Plug in antenna to this connector (optional)
FIX	LED Lights up when GPS gets location
SJ1	When bridged, Wake timer connected to GPIO 1
SJ2	Bridged, FIX LED will operate
JP1	Connects to RESET

Before using any Microstack™ module always read the instructions and precautions carefully.

Fitting Instructions

Carefully line up the pins of the Microstack™ module with sockets on the appropriate baseboard, then gently push down. The GPS module is fitted in the orange SPI/UART socket. Ensure every pin plugs into a corresponding socket.

Hardware Features

Built-in Logging

The module can act as a self contained data logger, recording position to internal flash memory. See the LOCUS section of the examples for further information.

External Antenna

An external active antenna can provide better performance if the sky is partly obscured and can offer increased sensitivity (ideal if module is used under tree canopies etc.) or inside buildings. An external antenna is connected through the U.FL connector. The GPS will automatically detect if an antenna is connected and report it in the NMEA string. If there is a fault with the antenna this will also be detected and reported.

The following external antennas have been tested with the Microstack™ GPS module. The following antennas are supplied with a male SMA connector, so a U.FL to SMA female adapter is required. A suitable adapter is available from CPC, part code (RF00367).

Farnell Order Code	Price	Gain (approx)	Remarks
190-0077	Low	26dB	Very thin receiver. Low cost. 5m long cable, double sided sticky mounting included. Ideal for mounting inside windows or plastic enclosures.
228-1641	Low	28dB	Waterproof, suitable for external mounting. Higher sensitivity results in fast fixes. Ideal for general purpose applications.
228-1640	Mid	30dB	Waterproof, increased gain enables fast fixes and can allow the use of more satellites for better positioning.
189-9494	Mid	35dB	Magnetic mounting, waterproof housing, very high sensitivity. This is most suitable if there is restricted view of the sky and high precision fixes from many satellites is achieved.

Fix Indication LED

The Fix indicator LED flashes at one second intervals when the module has a valid GPS position. It is driven from the Pulse Per Second pin.

Pulse Per Second

The PPS output can be used to synchronize devices with the highly accurate atomic clocks in the GPS system. The pin goes high for approx 100mS at the start of a second.

Configuration Jumper Pads

For most applications you do not need to change any of the jumper pad settings. However, for maximum flexibility the following options are available:

- SJ1 — Awake/Sleeping open drain output. Not connected by default. Connecting this jumper enables Pin 3 to indicate if the module is sleeping or awake. It may be used to wake custom user circuity or to control the power supply to the module to achieve even lower standby power consumption. It is connected to the TIMER pin of the GPS module. Consult Quectel L80 documentation for more information
- SJ2 — PPS. Connected by default. Connects Pulse Per Second signal of module to flash LED and output on Pin 4
- JP1 — Reset. Not connected by default. Connect to 0V to reset the GPS module.

Installing Software Under Linux

Microstack™ GPS is very easy to use under Linux. It is compatible with the standard gpsd suite of tools, or can be used with our dedicated lightweight and easy to use Microstack™

Node libraries. This gives a consistent interface across all Microstack™ addons. For the GPS module, the Microstack™ Node libraries offer specialised functionality, allowing access to the data logging features built into the GPS module.

Installing the Libraries

This guide assumes you are installing on Raspbian on a Raspberry Pi® connected to the Internet to allow you to download packages with apt-get.

Make sure you are using the lastest version of Raspbian by opening a terminal and typing:

```
sudo apt-get update
```

To install the Microstack™ Node libraries for Python 3, in a terminal, type the following command:

```
sudo apt-get install python3-microstacknode
```

To install the gpsd standard tools enter the following command in a terminal:

```
sudo apt-get install gpsd gpsd-clients python-gps
```

Setting Up the Serial Port

The Microstack™ GPS module communicates using the serial port and you need to ensure that it is not being used by anything else. By default the Raspberry Pi® uses the serial port for debugging, and this functionality needs to be disabled. This can be done manually or automatically in raspi-config. Using raspi-config is usually quicker and easier.

Easy, Automatic Serial Port Setup with raspi-config

The `raspi-config` tool makes it very easy to configure your Raspberry Pi®. Ensure you have the latest version of raspi-config; in a terminal type:

```
sudo apt-get install raspi-config
```

Now run raspi-config by typing in a terminal:

```
raspi-config
```

- From the menu choose `Advanced Options`.
- Then choose `serial`.
- When asked "`would you like a login shell to be accessible over serial?`" choose `<no>`
- A message saying "`serial is now disabled`" will appear.
- Exit raspi-config and reboot the Raspberry Pi®.

You can also manually configure the serial port so it is suitable to be used for Microstack™ GPS.

You need to edit the file `/etc/inittab`. Comment out the following line by putting a hash in front of it so that it goes from this:

```
T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
```

to this:

```
#T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
```

Also, remove all references to ttyAMA0 in `/boot/cmdline.txt` . For example change the file from this:

```
dwc_otg.lpm_enable=0 console=ttyAMA0,115200 console=tty1
root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline rootwait
```

to this:

```
dwc_otg.lpm_enable=0 root=/dev/mmcblk0p2 rootfstype=ext4
elevator=deadline rootwait
```

Automatically Starting gpsd

To automatically start the gps service when the Raspberry Pi® boots, re-configure the GPS daemon by typing into a terminal

```
sudo dpkg-reconfigure gpsd
```

- Choose `<yes>` when asked if you want to start `gpsd` automatically.
- Choose `<no>` when asked "`should gpsd handle attached USB GPS receivers automatically`".
- When asked which "`Device the GPS receiver is attached to`", enter `/dev/ttyAMA0`.
- Accept the defaults for other options.

Testing

With the module fitted and the software installed, you can test the Microstack™ GPS module with the following simple examples.

Note: It may take up to two minutes for the GPS to get a fix. The software may report invalid data or exit unexpectedly up until that point.

Testing the Microstack™ Node Python Libraries

Run the following example program by typing into a terminal:

```
python3 /usr/share/doc/python3-microstacknode/examples
/gpscat.py
```

Python Functions in Microstack™ Node

Full documentation is provided with the Python module, or online. You can also look at the example code in `/usr/share/doc/python3-microstacknode/examples`.

Get GPS data

Note: These examples below use Python interactively. This is shown by the >>> which is presented in the Python interface to indicate it is ready for you to enter a command. Do not type >>> before the commands!

Enter the following to create a gps object and then display the basic GPS data (GPGLL is useful to show location/time):

```
$ python3
>>> import microstacknode.gps.l80gps
>>> # creates a GPS object
>>> gps = microstacknode.gps.l80gps.L80GPS()
>>> # print the latest GPRMC data (errors if poor reception)
>>> gps.gprmc
>>> gps.gpvtg
>>> gps.gpgga
>>> gps.gpgsa
>>> gps.gpgsv
>>> gps.gpgll
>>> gps.gptxt
```

LOCUS Data Logging

The GPS unit can log data to its internal memory without the host computer running using the LOCUS logging system. The following Microstack™ Node Python functions provide a very easy way to use LOCUS logging:

```
>>> gps.locus_query() # Query the status of the LOCUS logger
>>> gps.locus_start() # Start the logger
>>> gps.locus_stop()  # Stop the logger
>>> gps.locus_erase() # Erase all data items in the log
>>> gps.locus_query_data() # Return a list of data items in
the logger
```

For example, you could issue the following command:

```
>>> gps.locus_start()
```

The GPS module will continue to log your position every few seconds until power is removed. Since the data is logged internally in the GPS, you can shutdown or suspend the device running Python.

Note: You must not remove power from the GPS module or else it will stop logging. However, previous logged data is stored in non-volatile memory.

To retrieve data (restart Python and reimport/recreate the gps object if necessary) and issue the following command:

```
>>> gps.locus_query_data()
```

The logger will continue to log GPS data until it runs out of memory, at which point it will stop. You can manually stop the LOCUS logger with the following command in Python:

```
>>> gps.locus_stop()
```

Power Management and Advanced Functions

The Microstack™ Node Python libraries contain functions to save power (e.g. sleep) or customise the operation (e.g. set custom update rate, or even vary update rate dependent on activity). For more information, consult the library documentation.

Testing gpsd

First ensure gpsd is running. If you have not set it to autostart, you can start it manually by typing in a terminal:

```
sudo gpsd /dev/ttyAMA0 -F /var/run/gpsd.sock
```

To display continuously updating data from the GPS, type the following at the command line.

```
cgps -s
```

A table will display the GPS' speed, time, position, visible satellites and quality of the fix.

Note: If the GPS cannot receive data (indicated by lack of flashing fix light) then cgps will exit shortly after starting.

Saving a Track to File

gpsd includes a command that frequently saves the GPS position to an XML file that can be imported into Google Earth or other online mapping programs.

Note: This is separate and does not use the internal LOCUS logging facility of the module.

To begin logging to the test.gpx file, enter the following in a command line:

```
gpxlogger -f test.gpx
```

Take the module for a short journey to log a few points. Press control-c to close the program and file.

There are lots of programs that use gpx track logs. You can visualise it easily by uploading the test.gpx file to GPS Visualizer at http://www.gpsvisualizer.com

Using gpsd with Python bindings
gpsd also has Python bindings. For more info on using gpsd visit the project homepage
http://www.catb.org/gpsd/

Advanced Features
The module can be configured with a variety of options. e.g.
- Interference cancellation can be enabled and disabled
- Fix interval up to 10Hz
- Power saving functions, such as sleep, sleep on inactivity
- Control of NMEA strings
- Baud rate settings

For more information consult the Microstack™ Node Python library documentation or the L80 GPS module protocol specification.

Warnings
- Microstack™ GPS is not intended for use in life critical systems. Do not rely on it for navigation or safety critical positional information. Ensure any application does not cause a distraction to the driver, captain or pilot of any vehicle, vessel or aircraft.
- Do not expose to water, moisture or extremes of temperature (below -40°C or above 85°C).
- Take care whilst handling to avoid mechanical and electrical damage to the module and connectors.
- Take suitable precautions to minimise risk of causing damage by electrostatic discharge.
- Connection to unapproved devices may affect compliance or result in damage to unit and invalidate any warranty.
- Connections to Microstack™ modules should only be made with the power supply disconnected.
- When using the external antenna, ensure connector is positioned such that it does not cause a short circuit with any other metal surface or contact on the module.

- Ensure that Microstack™ modules are powered by a suitably rated power supply that complies with the relevant regulations and standards applicable to the country of intended use.
- It is the user's responsibility to ensure the module is fitted in a suitable enclosure that it offers appropriate protection to ensure safe and proper operation.

Notes

As with any gps, position accuracy will degrade with poor reception of satellites. Poor reception will also cause time to fix to increase.

Compliance Information

- This Microstack™ module complies with the relevant provision of the RoHS Directive for the European Union. In common with all Electronic and Electrical Equipment this Microstack™ module should not be disposed of in household waste. Alternative arrangements may apply in other jurisdictions.
- Microstack™ GPS module is CE marked. It is a class B product. The EMC emission test was performed with a Microstack™ baseboard fitted to a Raspberry Pi® Model B+. Due to the nature of the module, as a prototyping and development board, fast transient immunity tests and conducted radio-frequency immunity tests have not been executed. ESD handling precautions should be observed. The module may be considered a component if integrated into another product. Any person designing or developing a product that uses one or more Microstack™ modules is responsible for ensuring that it is compliant and any modification to a Microstack™ module or inter-connection of other elements and devices with a module does not change compliance.
- This Class B digital apparatus complies with CAN ICES-3 (B). Cet appareil numérique de la classe B est conforme à la norme NMB-003 du Canada.
- This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

http://www.microstack.org.uk

Microstack™ is designed in the UK by OpenLX SP Ltd. Registered Office 145-157 St John Street, London, EC1V 4PW.
Microstack™ is distributed by Premier Farnell UK, 150 Armley Road, Leeds LS12 2QQ, UK

Manufactured in the UK.

Documentation Revision 1.0 October 2014

AlwaysLocate™ is a Trademark of MediaTek Inc. Raspberry Pi is a Trademark of the Raspberry Pi Foundation. All other Trademarks acknowledged.