# PORTABLE MEDLL RECEIVER

## Installation and Operation Manual

# Portable MEDLL Receiver

## Installation and Operation Manual

| | | |
|---|---|---|
| *Publication Number:* | ***OM-20000065*** | |
| *Revision Level:* | ***1*** | ***2002/06/21*** |

## Proprietary Notice

The software described in this document is furnished under a license agreement or non-disclosure agreement. The software may be used or copied only in accordance with the terms of the agreement. It is against the law to copy the software on any medium except as specifically allowed in the license or non-disclosure agreement.

No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose without the express written permission of a duly authorized representative of NovAtel Inc.

The information contained within this manual is believed to be true and correct at the time of publication.

GPSAntenna, GPSCard, GPSolution, MEDLL®, and MET® are trademarks of NovAtel Inc.

# TABLE OF CONTENTS

# TABLES

# FIGURES

# NOTICE

**In accordance with the EMC Directive 89/336/EEC of the European Community including amendments by the CE marking Directive 93/68/EEC.**

The Portable MEDLL receiver, in view of its design and type of construction, fully complies with the relevant basic radio interference requirements of the EMC Directive and qualifies to carry the **CE** Marking.

Should this instrument be modified without agreement, this declaration becomes invalid.

Relevant EMC Directive: EMC Directive (89/336/EEC)
Amendment (93/68/EEC)

Applied Harmonized Standard: **EN5022 CLASS A**

CISPR 11/22 CLASS A

**EN50082-2: 1994 R-02**

EN 61000-4-2: 1995

EN 61000-4-3: 1997

EN 61000-4-4: 1995

EN 61000-4-6: 1996

***IMPORTANT: In order to maintain compliance with the limits of EMC Directive 89/336/EEC, it is required to use properly shielded interface cables when using the Serial Ports, such as Belden #9539, or equivalent, double-shielded cables when using the Strobe Port, such as Belden #9945, or equivalent, and Belden #8770 cable for input power source (ensuring the shield is connected to the protection ground).***

# FOREWORD

## SCOPE

The *Portable MEDLL Receiver Installation and Operation Manual* is written for users of the Portable MEDLL receiver.

This manual describes NovAtel's Portable MEDLL receiver in sufficient detail to allow effective integration and operation. The manual is organized into sections, which allow easy access to appropriate information. Acronyms used in this manual may be found in *Appendix D, on Page 92*.

## PREREQUISITES

The Portable MEDLL receiver is a stand-alone fully functional GPS receiver. Refer to *Chapter 2, Installation of Portable MEDLL Receiver, on Page 4*, for more information on installation requirements and considerations. Refer to *Appendix B, on Page 86*, for an overview of GPS.

The NovAtel Portable MEDLL receiver utilizes a comprehensive user interface command structure, which requires communications through its serial (COM) ports. To utilize the built-in command structure to its fullest potential, it is recommended that some time be taken to review and become familiar with *Chapters 6-8* of this manual before operating the Portable MEDLL receiver.

Please also see the *User-Supplied Computer* section on *Page5.*

## COMPLIANCE WITH GPS WEEK ROLLOVER

The GPS week rollover issue refers to the way GPS receivers store information regarding the current GPS week. According to the official GPS system specifications document *(ICD-GPS-200, paragraph 20.3.3.3.1.1)*, "… 10 bits shall represent the number of the current GPS week…" This means an integer number between 0 and 1023 represents the GPS week. As GPS time started on Sunday January 6, 1980 at 0:00 hours Greenwich Mean Time (GMT), week 1023 ended on Saturday August 21, 1999 at 23:59:59 GMT.

According to the ICD-GPS specifications, the receiver should reset the GPS week number. This means that the week number should not advance to 1024, but start back at 0. However, another way to handle this issue is to extend the number of bits used to represent the GPS week number. This way, the GPS week would be able to increment per usual and would not have to be reset.

Per the GPS system specifications document, NovAtel firmware resets the receiver's GPS week number back to zero. Different manufacturers no doubt handled this situation differently. Therefore, users should be aware of this issue and keep in mind that there may be a compatibility issue when purchasing and using different makes of GPS receivers.

# 1 INTRODUCTION

The Portable MEDLL receiver is designed to receive, decode and output GPS signals for analysis by NovAtel's Multipath Assessment Tool (MAT). The primary functions of Portable MEDLL include:

- data collection (for example, multipath characterization data)
- determining satellite orbits
- determining satellite clock corrections
- determining satellite integrity
- independent data verification
- system operations & maintenance

## THE NOVATEL PORTABLE MEDLL RECEIVER

The principal function of the Portable MEDLL receiver is to provide GPS outputs that characterize multipath environments experienced at the receiver antenna. This is particularly important for permanent installations where signal reflections are likely to result in significant multipath effects. The Portable MEDLL receiver together with MAT can be used to analyse and select permanent reference station sites.

***Figure 1 The NovAtel Portable MEDLL Receiver***

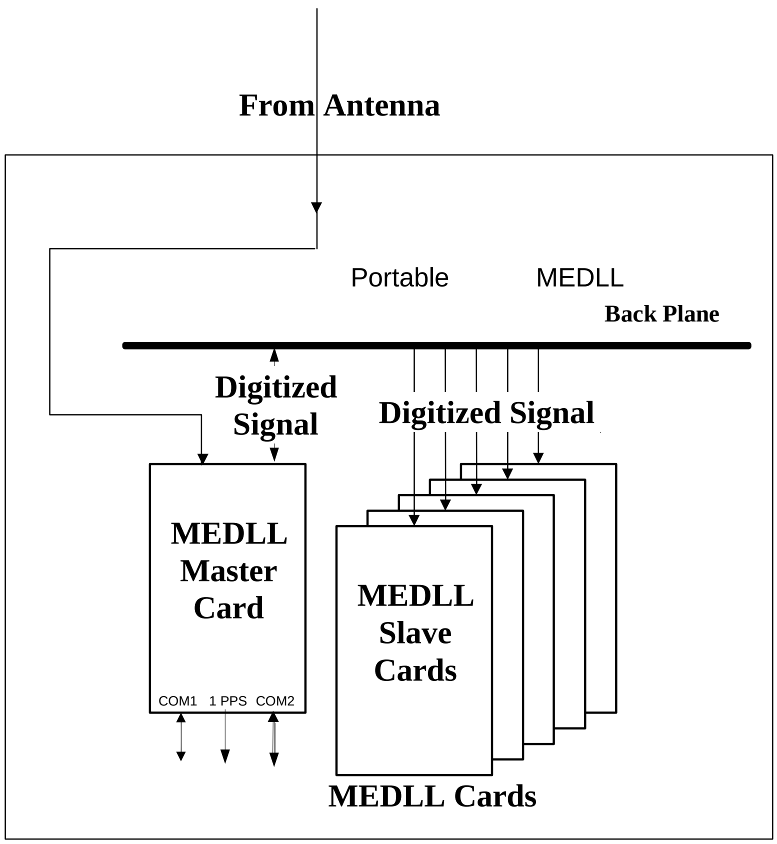

The NovAtel Portable MEDLL receiver is a high-performance GPS & GEO receiver that provides accurate multipath characterization data. NovAtel has developed a multipath elimination technology that approaches the theoretical limits of multipath-free GPS signal reception. This patented technology, known as "Multipath Estimating Delay-Lock-Loop" (MEDLL), uses a combination of hardware and software techniques which together are capable of reducing the combined effects of pseudorange and carrier-phase multipath errors by as much as 90% compared to a system using a standard correlator alone. The MEDLL technology takes advantage of NovAtel's parallel channel correlator sampling techniques. MEDLL uses a proprietary coupled correlator sampling technique combined with "maximum likelihood estimation" techniques to break down the received signals into direct path and reflected path components. MEDLL determines the amplitude, delay, and phase angle of both the direct and multipath signals and analyses the signal with the least delay to determine the direct path. The Portable MEDLL firmware provides multipath amplitude, delay and phase data for subsequent analysis. To do this, MEDLL utilizes a multi-card configuration. Each L1 GPSCard in the MEDLL receiver is linked to one common radio frequency (RF) deck and an oven controlled crystal oscillator (OCXO) that minimizes inter-channel biases.

The Portable MEDLL receiver is packaged in a portable instrumentation case. The rear panel's connectors provide easy access to strobes, communication, power and an antenna.

## HARDWARE OVERVIEW

The NovAtel Portable MEDLL receiver consists of a Master card and five Slave cards, *see Figure 2*.

***Figure 2 Portable MEDLL Receiver Functional Block Diagram***




The receiver provides:

a) L1 GPS and GEO data

b) a 1PPS output

c) COM ports for bi-directional communication with a computer

### MEDLL

MEDLL is implemented across a number of standard NovAtel 12 channel GPS receivers. Through parallel linking of these separate receiver modules, MEDLL may be configured to behave as a single GPS receiver, capable of simultaneously tracking 10 GPS satellites and 2 GEO satellites, or 12 GPS satellites and no GEO satellites.

A single incoming RF signal is routed to a Master Card (Master), which down converts the signal to baseband frequency for parallel processing by five Slave Receivers (Slaves). The baseband signal is then processed by twelve parallel digital signal-processing sections, through Multiple Independent Nomadic Stargazer (MINOS) Application Specific Integrated Circuits (ASICs), and NovAtel patented MEDLL tracking technology.

Across the twelve processing sections, there are a total of 72 tracking channels. Six channels are dedicated to tracking each GPS or GEO satellite, and these channels are positioned around the associated correlation envelope. By a process of continuous comparison of the signals measured by each channel, any distortion from the ideal correlation envelope can be detected, tracked, qualified and removed. This process allows the receiver to isolate and eliminate multipath distortions from the received signal.

The Master and Slave receivers are mounted in a portable instrumentation case, and are supplied power from an integrated power conditioner. Stable clock signals are derived from a precision internal oscillator. Status indicators on the front panel provide visual confirmation of the health of each electronic sub-assembly within the case.

Signals are routed to and from the MEDLL receiver via an RF antenna input, power and digital I/O signal connector on the rear of the unit.

The unit is controlled via RS-232 using standard NovAtel commands, and data is output in NovAtel output logs. Specific MEDLL logs provide access to the MEDLL processing data.

### GEO Processing

Specific channels in the MEDLL receiver have the capability to receive and process the GEO signals. The signal is in-band at L1 and is identified with GEO-specific PRN numbers. The GEO message is decoded and separated into its various components. The GEO message and associated pseudorange is provided as an output.

### Other Outputs & Inputs

- A 1 pulse per second (1PPS) strobe output from the MEDLL master. Timing information of this pulse is available in the TM1A log, see *Page 79*.
- The COM1 and COM2 ports provide: - raw satellite measurements (pseudorange, carrier & time)

  - receiver status data (communications & tracking)
  - raw satellite data (ephemeris & almanac)
  - fast code corrections for signal stability monitoring
  - logs for the MAT software

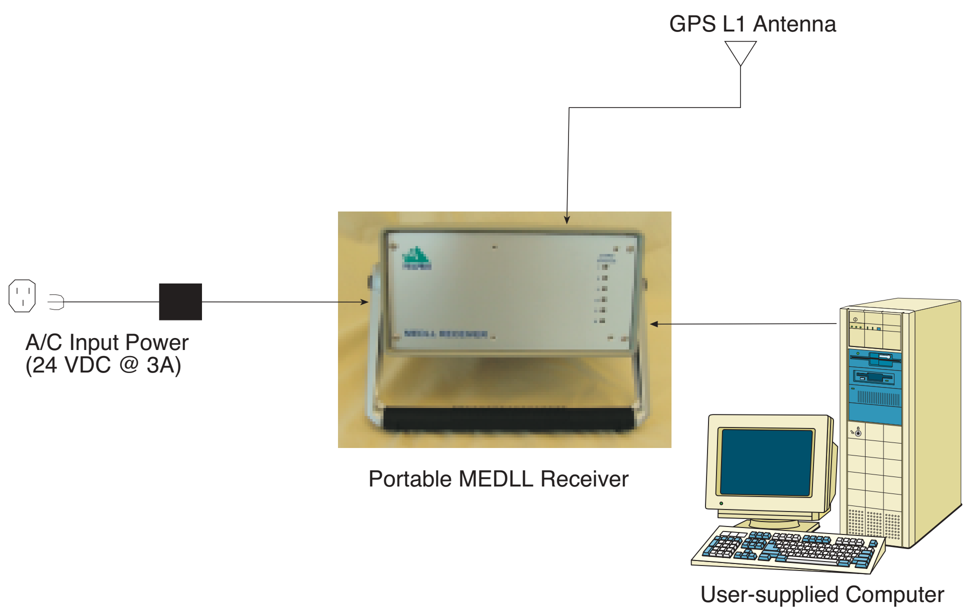
# 2 INSTALLATION OF PORTABLE MEDLL RECEIVER

This chapter provides sufficient information to allow you to set up and prepare the Portable MEDLL receiver for stand-alone operation.

## MINIMUM CONFIGURATION

In order for the Portable MEDLL receiver to function as a complete system, a minimum equipment configuration is required. This is illustrated in *Figure 3*.

***Figure 3 Portable MEDLL Minimum System Configuration***




The recommended minimum configuration and required accessories are listed below:

- NovAtel Portable MEDLL receiver
- L1 GPSAntenna and low noise amplifier (LNA)
- Power supply (22-30 VDC, 3 A maximum)
- User-supplied computer, please see the same named section on the next page
- RF cables
- User-supplied data cable

Of course, your intended set-up may differ significantly from this configuration. This section describes the basic system configuration, which you can modify to meet your specific situation.

For the configuration in *Figure 3*, setting up the Portable MEDLL receiver involves the following steps:

1. Connect your computer to the Portable MEDLL receiver (“COM1” or “COM2”connector)

   Your computer must be fast enough to accept all logs requested at the configured data rates. It is recommended that the serial port be configured to the fastest compatible rate supported by both the receiver and the computer.

   If data collection is planned over an expanded period, 24 hours for example, ensure the data storage capacity is adequate for the logs requested. (NovAtel recommends Microsoft Windows 2000 or later as the computer operating system). If logs are missing in the data record, then the requested logs and logging rates exceed the receiver/computer communications capability. In this case, it is suggested that you reduce the logging rate or increase the serial communication rate.

2. Install the GPS antenna and low noise amplifier, and make the appropriate connections to the Portable MEDLL receiver ("ANT" connector)
3. Supply power in the range 22 to 30 VDC to the Portable MEDLL receiver ("+24 VDC" connector)

The connections on the rear panel are shown in *Figure 4.*


***Figure 4 Rear Panel of Portable MEDLL Receiver***

## *USER-SUPPLIED COMPUTER*

Recommended user supplied computer prerequisites:

a. A laptop computer or workstation with Windows 2000 Professional, SR-1 or later.

b. Hard disk with at least 400MB for free space for each 24 hours of data. Please note that this storage requirement is after subtracting any "Virtual RAM" allocated to the operating system.

c. RAM, 128MB

d. CPU, 300 MHz Pentium 3 or equivalent. Note that the CPU speed requirement depends on other processing requirements. This recommendation assumes the computer is used for data collection only.

e. Archival data storage. It is recommended that any data collected with the receiver be placed on archive media such as a CD-ROM as a contingency against a disk drive failure.

f. A high quality null modem cable for connection between the receiver and the user supplied computer. NovAtel recommends NovAtel part number 60715062 or equivalent.

## LEDS

The LEDs on the front panel indicates the status of the Portable MEDLL receiver's 6 cards. They should turn from red to green to let you know the Portable MEDLL receiver's serial ports are healthy and ready to communicate, see *Figure 5* following.

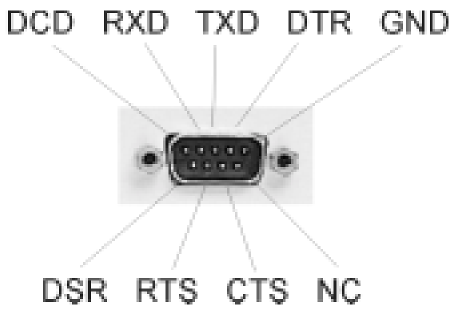
***Figure 5 Lights on Front Panel of Portable MEDLL Receiver***


## CONNECTING DATA COMMUNICATIONS EQUIPMENT

There are two serial ports on the back panel of the Portable MEDLL receiver; both are configured for RS-232 protocol. These ports make it possible for external data communications equipment - such as a personal computer - to communicate with the Portable MEDLL receiver. Each of these ports has a DE9P connector.

- The COM1 and COM2 ports (see *Figure 6*) allow two-way communication. They are connected to the MEDLL master card.

***Figure 6 Pinout for COM Ports - Portable MEDLL***



## CONNECTING THE GPS ANTENNA

Selecting and installing an appropriate antenna system is crucial for proper operation of the Portable MEDLL receiver.

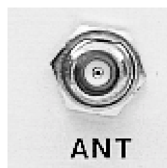
Keep the following points in mind when installing the antenna system:

- Ideally, select an antenna location with a clear view of the sky to the horizon so that each satellite above the horizon can be tracked without obstruction.
- Ensure that the antenna is mounted on a secure, stable structure capable of withstanding relevant environmental loading forces (e.g. due to wind or ice).

Use high-quality coaxial cables (NovAtel 5m cable, part number C005, for example) to minimize signal attenuation. The gain of the LNA must be sufficient to compensate for the cabling loss. The MEDLL receiver supplies 5 A to power the LNA.

The antenna port on the Portable MEDLL receiver has a TNC female connector, as shown in *Figure 7*.

***Figure 7 Antenna Input - Portable MEDLL***

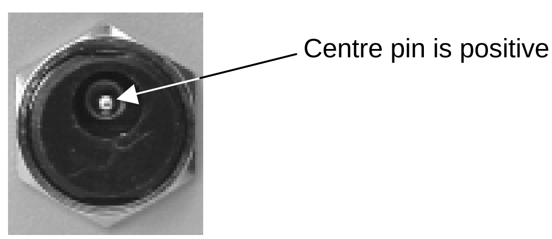



## CONNECTING THE EXTERNAL POWER INPUT

The Portable MEDLL receiver provides one external regulated power supply. The input can be in the 22-30 VDC range. The receiver draws up to 3 A at start-up, but the steady-state requirement is approximately 1.5 A.

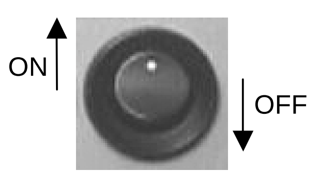
The power-input connector on the Portable MEDLL receiver connects to the Portable MEDLL receiver's internal power supply, which performs filtering and voltage regulation functions. Refer to *Figure 8*, which shows the external power connection on the Portable MEDLL receiver.

***Figure 8 External Power Connections - Portable MEDLL***




To ensure proper operation, the power switch, see *Figure 9*, must be used for applying power to the receiver. First, an external power supply must be connected to the receiver with the power switch in the OFF position (down).

***Figure 9 Power Switch - Portable MEDLL***

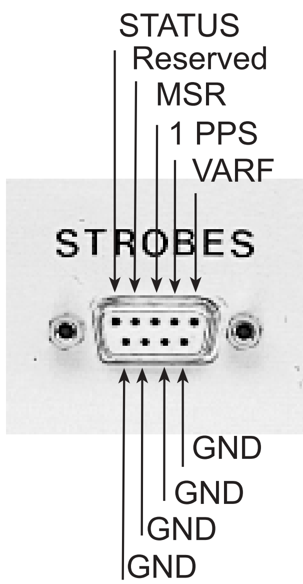


## ACCESSING THE STROBE SIGNALS

The Portable MEDLL receiver's output strobe lines are available on the rear panel from the DE9S connector (see *Figure 10*). The specifications and electrical characteristics of these signals are described in *Appendix A*, starting on *Page 83*. These signals are provided for diagnostic purposes.

The COM ports are connected to the MEDLL master card.

***Figure 10        Strobe 9-pin D-Connector Pinout - Portable MEDLL***



# 3 OPERATION

Before operating the Portable MEDLL receiver for the first time, ensure that you have followed the installation instructions in *Chapter 2*.

From here on, it will be assumed that testing and operation of the Portable MEDLL receiver will be performed while using a personal computer (PC); this will allow the greatest ease and versatility.

## PRE-START CHECK LIST

Before turning on power to the Portable MEDLL receiver, ensure that all of the following conditions have been met:

- The antenna is properly installed and connected.
- The PC is properly connected using a null-modem cable, and its communications protocol has been set up to match that of the Portable MEDLL receiver (see the *Serial Ports* section below).

Supply power to the Portable MEDLL receiver only after all of the above checks have been made. Note that the warm-up process may take several minutes, depending on ambient temperature. Required start-up time (cold start) is 5 minutes. This is for the OCXO to stabilize.

It is recommended that you do a complete receiver reset by sending a RESET command after 5 minutes, following a cold start. This allows a recalibration of the receiver algorithm after RF components have warmed up. It also allows the automatic sky search routine to reset.

If this power-up order is not followed, the start-up times will not be achieved. The limiting factor in achieving the start-up times is the OCXO becoming stable.

### SERIAL PORTS - DEFAULT SETTINGS

Because the Portable MEDLL receiver communicates with the user's PC or core computer via serial ports, both units require the same port settings. The communications settings of the PC should match these on the receiver:

- RS-232 protocol
- 9600 bits per second (bps)
- No parity
- 8 data bits
- 1 stop bit
- No handshaking
- Echo off

Once initial communications are established, the port settings for the Portable MEDLL receiver can be changed using the *COMn* command, which is described on *Page 22*.

### BOOT-UP

The Portable MEDLL receiver's firmware resides in non-volatile memory. Supply power to the unit, wait a few moments for self-boot, and the Portable MEDLL receiver will be ready for command input.

On receiver start-up, the integrity of the random-access memory (RAM) is checked. If the RAM check fails, the receiver waits for operator intervention. If the RAM check completes successfully, the authorization code is verified in order to check the integrity of the software load. If this check fails, an associated message will be output from the receiver (see *Appendix C Information Messages* starting on *Page 90*). Following the authorization code check, the configuration of the receiver is verified. The default configuration for the receiver model will be applied (see *Appendix A Portable*

*MEDLL Receiver* **-** *Technical Specifications* starting on *Page 83)*. After these checks have been completed, the receiver will output the COM port prompt indicating everything has started correctly and the receiver will accept commands.

A read-back check of the ASIC and a loop-back test of the COM ports are also included in the start-up test sequence to make sure that they are functioning properly. In the event of a failure of either of these checks, the associated bit will be set in the receiver status word (see *Table 12* on *Page 70*). If it has been determined that the COM port has failed, the COM prompt may not be output, depending on the nature of the failure.

There are two initial start-up indicators to let you know that the Portable MEDLL receiver's serial ports are ready to communicate:

1. Status lights on the Portable MEDLL receiver's front panel should turn from *red* to *green* to indicate that all cards are healthy. If any one of the LEDs does not turn *green*, then the system should be considered unreliable. If this situation occurs, the receiver requires maintenance.
2. Your external computer screen will display one of the following prompts:

    **Com1>** (if you are connected to the COM1 port.)

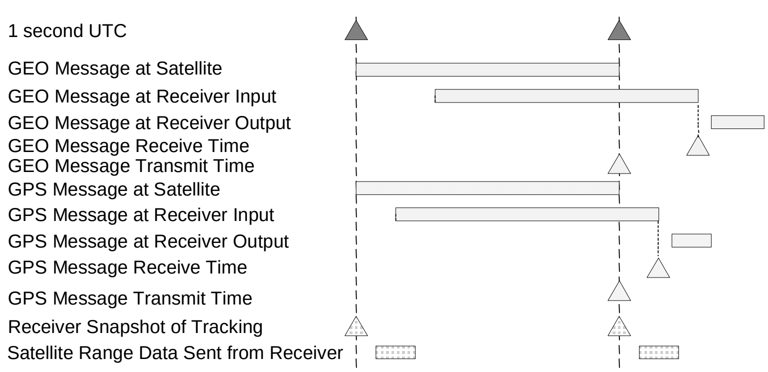
    **Com2>** (if you are connected to the COM2 port.)

The Portable MEDLL receiver is now ready for command input from the COM1 or COM2 port.

## TIMING: RECEIVER TRANSMIT AND RECEIVE TIME

Transmit time is defined as the time when the last bit of the GPS signal leaves the satellite antenna. Receive time is when the last bit is removed from the MINOS chip. For example, the SFD log has receive time and transmit time. The transmit time is when the last bit that belongs to a 1 second accumulation leaves the satellite antenna. Receive time is the time that the last bits of the 1 second accumulation are taken from the MINOS register. *Figure 11* shows the timing relationships within the Portable MEDLL receiver.

***Figure 11 Timing Relationships***




# INITIAL COMMUNICATIONS WITH A PORTABLE MEDLL RECEIVER

Communicating with the Portable MEDLL receiver is accomplished by issuing commands to a COM port from an external serial communications device.

The software on the PC could be one of the following:

1. Terminal Emulator (e.g. HyperTerminal)
2. GPSolution
3. Multipath Assessment Tool (MAT) is recommended

To change the default communication settings, such as bit rate, use the *COMn* command, see *Page 22*.

It is to your advantage to become thoroughly familiar with *Chapters 5* through *7* of this manual to ensure maximum utilization of the Portable MEDLL receiver's capabilities.

When the Portable MEDLL receiver is first powered up, no activity information is transmitted from the COM ports except for the **COM1>** or **COM2>** prompt described in the *Boot-up* section on *Page 9*. Receipt of the prompt confirms the boot process is complete and the receiver is ready to accept commands.

Commands are directly input to the Portable MEDLL receiver using the external computer. It should be noted that most commands do not echo a response to a command input. Your indicator that the command has actually been accepted is a return of the **COM1>** or **COM2>** prompt from the Portable MEDLL receiver. Note that *VERSION* is the only command that does provide an echo response instead of the port prompt.

**Examples:**

1. If you type VERSION <Enter> from a terminal, this will cause the Portable MEDLL receiver to echo the firmware version information.

2. An example of a no-echo response to an input command is the FIX POSITION command. It can be input as follows:

   COM1>**fix position 51.113 -114.043 1060** <Enter>

   This example illustrates command input to the COM1 port, which sets the Portable MEDLL receiver's position. However, your only confirmation that the command was actually accepted is the return of the **COM1>** prompt.

If a command is erroneously input, the Portable MEDLL receiver will respond with either the "Invalid Command Option" or "Invalid Command Name" response followed by the COMn> prompt.

## *VALID DATA AND AVAILABLE LOGS*

15 minutes after a cold start, all logs are available and the data within the logs is valid.

You can also monitor the log output for indication that the receiver output is valid. (i.e. FINETIME is set, position is valid, almanac is valid)

# 4 FIRMWARE UPDATES

The method of updating the firmware in the Portable MEDLL receiver is with a manual update process described following.

GPSCards store their firmware (program software) in on-board, non-volatile memory. This unique feature allows a receiver's firmware to be updated in the field. Thus, a procedure such as updating software model **EGNOSMEDLL rev. 5.4441S7** to **EGNOSMEDLL rev. 5.45** takes only a few minutes instead of the several days which would be required if the receiver had to be sent to a service depot.

The MEDLL system is updated by means of a serial connection to the host PC using a COM port.

When updating the receiver, you will need to transfer the new firmware to the appropriate GPSCard with the aid of the NovAtel-supplied utility program, "**LOADER"**. To update firmware while using LOADER, you will need a personal computer with the following features:

- MS-DOS 6.0 or later
- one available RS-232 serial port
- null-modem cable
- at least 1 MB of available hard drive space

Below is an outline of the procedure for updating your receiver's firmware:

1. Obtain firmware update files from NovAtel
2. Decompress files
3. Run LOADER in one of three modes: Menu, Command Line, or Entire Receiver Update.

## OBTAIN FILES

The files are available in compressed, password-protected file format. The compressed form of the files will have differing names; NovAtel will advise you as to the exact names of the files you need. As well, NovAtel will provide you with a file decompression password.

## DECOMPRESS FILES

After copying the compressed files to an appropriate directory on your PC, each file must be decompressed. The syntax for decompression is as follows:

Syntax:

[*filename*] -s[*password*]

Where:

| | |
|---|---|
| *filename*: | is the name of the compressed file (excluding the extension) |
| *-s*: | is the password command switch |
| *password*: | is the password required to allow decompression |

**Example**:

m54441S7 -s12345678<Enter>

## RUN LOADER

LOADER should be copied to the hard drive of your personal computer and run from the command (DOS) prompt. Once LOADER is installed and running, it allows you to select and configure a PC serial port, as well as choose the directory path and file name of the new program software to be transferred to the GPSCard. After the LOADER parameters have been selected and the auth-code entered, the actual file transfer only takes a few minutes, depending on the data transfer rate selected. LOADER also contains built-in terminal software.

LOADER will instruct you when to turn on and off the Portable MEDLL receiver.

Run LOADER in one of these modes:

1. Command-Line mode: This allows you to set up a batch process to update the multiple cards within the MEDLL.
2. Entire Receiver Update mode: Use this if you wish to update all of the GPSCards in the MEDLL system within the Portable MEDLL in a single operation.

## Command-Line Mode

LOADER may be used with command line options directly at the DOS command prompt. In this mode of operation the filename and authorization code can be specified on the command line. When the program detects a filename and authorization code, it immediately proceeds to read the specified file, authorize it and send it to the Portable MEDLL receiver. The syntax is in the section that follows.

## Updating the GPSCards in the MEDLL Receiver

A batch file can be used to provide a convenient way to program multiple GPSCards with the same or different versions of software. Rather than having to run LOADER once for each card, you can specify the card number, firmware filename and an optional authorization code in a batch file.

The batch file is a simple ASCII text file and should be formatted as shown in the example below. In this example, Card #1 is to be loaded with software in the M544.BIN file, and remaining cards with the S544.BIN file. Cards #2-#8 have no authorization codes. The batch file should be in the same directory as the LOADER program.

Syntax #2:

| | |
|---|---|
| LOADER | -b<batchfile> |

-b<batchfile> Specifies file with multiple load commands.

-h Requests on-line help.

Examples:

```
loader –h
loader -bmedll.txt
```

In the second example, this would be the listing of the MEDLL.TXT file for a MEDLL receiver consisting of one master card and five slave cards:

```
1 m54441S7.bin 1234,5678,9012,3456,EGNOSMEDLL
   2 s54441S7.bin
   3 s54441S7.bin
   4 s54441S7.bin
   5 s54441S7.bin
   6 s54441S7.bin
```

The LOADER program will only initialise the number of cards required to complete the programming. Once the cards have been initialised, the screen will show this:

```
Programming card: 1 with M54441S7.BIN
```

As the card is being programmed, a character to the right of the filename will spin indicating programming activity. When the first card has been programmed, the screen will show this:

```
Programming card: 1 with M54441S7.BIN     Okay
Erasing card:  2    \
```

This process will continue until all of the requested cards are programmed and the screen will look similar to this:

```
Programming card: 1 with M54441S7.BIN     Okay
Programming card: 2 with S54441S7.BIN     Okay
.
.
Programming card: 6 with S54441S7.BIN     Okay
Done.  Resetting Cards
Initialisation took:      27 seconds
   Programming took:     114 seconds
         Total Time:     141 seconds
Press ENTER to exit
```

Once all of the cards have been programmed, they will be reset. Using PC communication software, or the terminal emulator in the Menu mode of LOADER, issue the *VERSION* command to verify your new program version number.

***Warning!*** Master and slave sections need the same software version or the receiver will fail at start-up.

## Entire Receiver Update Mode

If you wish to perform an update of all the GPSCards in a Portable MEDLL receiver, this mode of LOADER will guide you through the process and simplify the operation considerably. Together with the LOADER program and the update files that you received from NovAtel, there will also be a file with a name such as UPDATE.BAT. This is a batch file which runs LOADER in Command Line mode as described above, with all the commands already prepared.

To run the software, you will need to issue the following command: -

Syntax #3:

| | | |
|---|---|---|
| UPDATE | <serial port> | <data transfer rate> |

| | |
|---|---|
| <serial port> | Specifies which serial port (COM port) on your computer is to be used |
| <data transfer rate> | Specifies the data transfer rate that is to be used |

Examples:

```
update 2 115200
```

This example instructs the UPDATE utility to upgrade the GPSCard, which is connected to the PC's serial port 2 (COM2) at 115,200 bits per second. The utility will prompt you to turn the power to the Portable MEDLL receiver on or off; it will also prompt you to connect your personal computer to a COM port on the Portable MEDLL receiver.

By the time this utility completes its task, you will have connected your computer to a COM port on the Portable MEDLL receiver and updated all the GPSCards inside.

# 5 COMMAND DESCRIPTIONS

This chapter describes the commands accepted by the Portable MEDLL receiver. They are listed in alphabetical order.

The Portable MEDLL receiver is capable of responding to many different input commands. You will find that once you become familiar with these commands, the Portable MEDLL receiver offers a wide range in operational flexibility. Commands can be sent to the Portable MEDLL receiver through the COM1 or COM2 serial port.

You can issue these commands to control the following:

1. Overall status of the Portable MEDLL receiver
2. Input & output functions
3. Configuration of a specific channel of the Portable MEDLL receiver

*Table 1* shows the list of commands arranged alphabetically.

When the Portable MEDLL receiver is first powered up, all commands revert to the factory default settings. Each command description in this chapter also lists its default setting.

The following rules apply when communicating with the card:

1. The commands are not case sensitive.
   - e.g. *VERSION* or *version*
   - e.g. *FIX POSITION* or *fix position*

2. All commands and required entries can be separated by a space or a comma.
   - e.g. *fix,position,51.3455323,-117.289534,1002*
   - e.g. *fix position 51.3455323 -117.289534 1002*
   - e.g. *com1,9600,n,8,1,n,off*
   - e.g. *com1 9600 n 8 1 n off*
   - e.g. *log,com1,frmb,onnew*
   - *e.g.* *log com1 frmb onnew*

3. At the end of a command or command string, press <Enter>.

4. Successful entry of a command is verified by receipt of the COM port prompt (i.e. COM1>). Exceptions to this statement are the *VERSION*, *RESET* and *CONFIG* commands where the latter two commands restart the computer. Commands entered in error will return either an "INVALID COMMAND OPTION" or "INVALID COMMAND NAME" string.

**Table 1 Portable MEDLL Command Summary**

| Command | Description | Syntax |
|---|---|---|
| $ALMA | Almanac decoder | $alma *prn,ecc,seconds,week,w,ra,w,$M_0$,$a_{f0}$,$a_{f1}$, corr-mean-motion,A,incl-angle,health-4, health-5,health-alm* |
| ASSIGN | Assign a PRN to a channel # | assign *channel,prn,[doppler], [search window]* |
| $ASSIGNG2TOPRN | Assign a G2 delay to a PRN | $assigng2toprn *prn g2_delay* |
| CLOCKADJUST | Adjust 1PPS continuously | clockadjust *switch* |
| COM1 | Initialize Serial Port 1 | com1 *baud,parity,databits,stopbits, handshake,echo,[fifo]* |
| CONFIG | Implements pre-defined configurations | config *[keyword]* |
| CSMOOTH | Sets code smoothing | csmooth *value,[value2]* |
| ECUTOFF | Set elevation cutoff angle (degrees) | ecutoff angle |
| FIX POSITION | Set antenna coordinates for monitor station | fix position *lat,lon,height,[station id],[health]* |
| FREQUENCY_OUT | | |
| LOG | Choose data logging type | log *[port],datatype,[trigger],[period],[offset],[hold]* |
| RESET | Performs a CPU reset | reset |
| $SETFRAMETYPE | Sets the type of navigation frame data output in the FRMA/B log | $setframetype *type* |
| UNASSIGN | Un-assign a channel | unassign *channel* |
| UNASSIGNALL | Un-assign all channels | unassignall |
| $UNASSIGNG2TOPRN | Un-assign a G2 delay | $unassigng2toprn *prn* |
| UNDULATION | Choose undulation | undulation *separation* |
| UNFIX | Remove all receiver FIX constraints | unfix |
| UNLOG | Cease logging a data log | unlog *[port],data type* |
| UNLOGALL | Cease logging all data logs | unlogall |
| VERSION | Current software level | version |

The syntax tables contain the start-up defaults for each command. If a command has optional parameters, the syntax table for that command will also have a default parameters column.

# *$ALMA*

Enter almanac parameters for sub-frames four and five. Enter parameters for each SV almanac required.

When the $ALMA log is input to the receiver, it should be input in exactly the same format as it was originally logged from the receiver. Therefore, there should be no concerns about the required accuracy of the input parameters.

For more information on Almanac data, refer to the *GPS SPS Signal Specification*.

Syntax:

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| `$ALMA` | `prn` | `ecc` | `seconds` | `week` | $\dot{w}$ | `ra` | `w` | $M_0$ | $a_{f0}$ | $a_{f1}$ | `corr-mean-motion` |
| | `A` | `incl-angle` | `health-4` | `health-5` | `health-alm` | | | | | | |

| Syntax | Range Value | Description | Start-Up Default |
|---|---|---|---|
| $ALMA | N/A | Command | N/A |
| prn | (The $ALMA command is generated directly from the ALMA log and therefore the core computer should never need to change or verify the almanac data.) | Satellite PRN number for message | (When Portable MEDLL receivers start up, there is no almanac available.) |
| ecc | | Eccentricity | |
| seconds | | Almanac reference time, seconds into the week | |
| week | | Almanac reference week (GPS week number) | |
| $\dot{w}$ | | Rate of right ascension, radians/second | |
| ra | | Right ascension, radians | |
| w | | Argument of perigee, radians | |
| $M_0$ | | Mean anomaly, radians | |
| $a_{f0}$ | | Clock aging parameter, seconds | |
| $a_{f1}$ | | Clock aging parameter, seconds/second | |
| cor-mean-motion | | Corrected mean motion, radians/second | |
| A | | Semi-major axis, metres | |
| incl-angle | | Angle of inclination, radians | |
| health-4 | | Anti-spoofing and SV config (sub-frame 4, page 25) | |
| health-5 | | SV health, 6 bits/SV (sub-frame 4 or 5, page 25) | |
| health-alm | | SV health, 8 bits (almanac) | |

Example:

```
$alma 17,1.23692E-002,503808,67,-8.0346E-009,-1.4494328E+000,3.06213275E+000,
-1.4963308E+000,-2.8706E-004,-3.6380E-012,1.45856126E-004,2.65600872E+007,
9.83002E-001,1,0,0
```

## ASSIGN

At start-up, the Portable MEDLL receiver automatically searches for GPS satellites (PRN 1-32). However, the PRN for GEO satellites must be manually assigned to a receiver channel using the ASSIGN command.

This command may be used to aid in the initial acquisition of a satellite by allowing you to override the automatic satellite/channel assignment and reacquisition processes with manual instructions. The command specifies that the indicated tracking channel searches for a specified satellite at a specified Doppler frequency within a specified Doppler window. The instruction will remain in effect for the specified channel and PRN, even if the assigned satellite subsequently sets. If the satellite Doppler offset of the assigned channel exceeds that specified by the 'search-window' parameter of the ASSIGN command, the satellite may never be acquired or re-acquired. To cancel the effects of ASSIGN, you must issue the UNASSIGN or UNASSIGNALL command, or reboot the Portable MEDLL receiver.

When using this command, NovAtel recommends that you monitor the *channel tracking status* of the assigned channel and then use the UNASSIGN or UNASSIGNALL commands to cancel the command once the L1 channel has reached channel tracking state 4, the Steady State Tracking state. Refer to *Table 4, on Page 53,* for an explanation of the various channel states. See also the $ASSIGNG2TOPRN command on *Page 20*. However, this is not recommended for GEO satellites. GEO satellites should be assigned without specifying a Doppler value or a Doppler search-window.

***NOTES:*** Assigning a PRN to a channel does not remove the PRN from the search space of the automatic searcher; only the channel is removed.

By default, the automatic searcher only searches for the GPS satellites (PRNs 1-32).

There are two syntactical forms of this command, as shown following.

Syntax #1:

`ASSIGN` `channel` `prn` `[Doppler]` `[Search-window]`

| Syntax | Range Value | Description | Start-Up Default | Parameter Default |
|---|---|---|---|---|
| ASSIGN | - | Command | UNASSIGNALL [1] | |
| Channel | 0 - highest channel number | Desired channel number from 0 to maximum channel number inclusive | | |
| Prn | 1 - 999 | A satellite PRN integer number from 1 to 32 inclusive (for GPS), 120 to 138 inclusive (for GEO), or 1 to 999 inclusive if PRNs are assigned using the $ASSIGNG2TOPRN command. | | |
| [Doppler] | -100,000 to 100,000 Hz | Current Doppler offset of the satellite<br>Note: Satellite motion, receiver antenna motion and receiver clock frequency error must be included in the calculation for Doppler frequency. | | -100000 < Doppler < 100000 |
| Search-window | 0 - 10,000 | Error or uncertainty in the Doppler estimate above in Hz<br>Note: Any positive value from 0 to 10000 will be accepted. Example: 500 implies ± 500 Hz. | | 10000 |

Example 1:

```
assign 0,29,0,2000
```

In Example 1, the first channel will try to acquire satellite PRN 29 in a range from -2000 Hz to 2000 Hz until the satellite signal has been detected.

Syntax #2:

`ASSIGN` `channel` `keyword`

| Syntax | Range Value | Description | Start-Up Default |
|---|---|---|---|
| ASSIGN | - | Command | UNASSIGNALL [1] |
| Channel | 0 - highest channel number | Desired channel number from 0 to maximum channel number inclusive | |
| Keyword | IDLE | Idles channel (not case sensitive) | |

Example 2:

```
assign 11,idle
```

In Example 2, Channel 11 will be idled and will not attempt to search for satellites.

[1] This is issued automatically at start-up.

## $ASSIGNG2TOPRN

This command allows you to link a satellite PRN with a particular G2 delay. This delay is then added to a user-defined table, which allows up to 100 entries. Hence, when an ASSIGN command is issued, the receiver will check the user-defined table first and if the PRN is listed there, it will use that G2 delay. Otherwise it will use the default values for GPS and GEO satellites. This command affects the C/A code tracking only.

Use the $UNASSIGNG2TOPRN command to delete a G2 delay assignment for a particular PRN from the user-defined table.

Syntax:

```
$ASSIGNG2TOPRN   prn   G2 delay
```

| Syntax | Range Value | Description | Start-Up Default |
|---|---|---|---|
| $ASSIGNG2TOPRN | - | Command | |
| Prn | 1 – 999 | PRN Number | (see footnote) [2] |
| G2 delay | 0 – 1023 | C/A code G2 delay | (see footnote) [2] |

Example:

```
$assigng2toprn 32 862
```

[2] Defaults are as defined in the GPS Interface Control Document (ICD) and the WAAS Minimum Operational Performance Standards (MOPS).

## *CLOCKADJUST*

All oscillators have some inherent drift characterization. This command, if enabled, permits software to model these long-term drift characteristics of the clock. The correction is applied to the 1PPS strobe as well. The clock adjustment is performed digitally. As a result, the 1PPS Strobe (on an L1 card) will have a 49 ns jitter on it due to the receiver's attempts to keep it as close as possible to GPS time.

CLOCKADJUST must be disabled if you wish to measure the drift rate of the oscillator using the CLKA/B data log.

***NOTES:***
1. Do not disable this command after 30 seconds from power-up: unpredictable clock drifts may result.
2. When disabled, the range measurement bias errors will continue to accumulate with clock drift and the 1PPS output from the receiver will drift with the internal clock.
3. This feature should only be changed by advanced users.

Syntax:

```
CLOCKADJUST switch
```

| Syntax | Range Value | Description | Start-Up Default |
|---|---|---|---|
| CLOCKADJUST | - | Command | |
| switch | Enable or disable | Allows or disallows adjustment to the internal clock | ENABLE |

Example:

```
clockadjust disable
```

## COMn

This command permits you to configure the COM port's asynchronous drivers.

The COM1 and COM2 ports allow two-way communications. Thus you may enter *COM1* or *COM2* as the command name.

Syntax:

```
COM1 bps parity databits stopbits handshake echo [FIFO]
```

| Syntax | Range Value | Description | Start-Up Default | Parameter Default |
|---|---|---|---|---|
| COMn | n = 1 or 2 | COM1 or COM2 | com1 | |
| bps | 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600 or 115200 | Specify bit rate | 9600 | |
| parity | N (none), O (odd) or E (even) | Specify parity | N | |
| databits | 7 or 8 | Specify number of data bits | 8 | |
| stopbits | 1 or 2 | Specify number of stop bits | 1 | |
| handshake | N (none), XON (Xon/Xoff) or CTS (CTS/RTS) | Specify handshaking | N | |
| echo | ON or OFF | Specify echo | OFF | |
| FIFO | ON or OFF | Enable or disable the 16550 UART's transmit FIFO | ON | (see footnote) [3] |

Examples:

```
com1 19200,e,7,1,n,on, off
com2 1200,e,8,1,n,on, off
```

[3] The default is set to whatever value was last used.

## *CONFIG*

This command switches the channel configuration of the COMn code receiver between pre-defined configurations. When the command is issued the receiver resets with the new configuration.

Syntax:

```
CONFIG   [keyword]
```

| Syntax | Range Value | Description | Start-Up Default | Parameter Default |
|---|---|---|---|---|
| CONFIG | - | Command | | |
| keyword | G10W2 | Configures the COMn code portion of the MEDLL receiver to track up to 10 GPS satellites & 2 GEO satellites. | G10W2 | (see footnote) [4] |
| | G12W0 | Configures the COMn code portion of the MEDLL receiver to track up to 12 GPS satellites. | | |

Example:

```
config g10w2
```

[4] The default is set to whatever value was last used.

## CSMOOTH

This is the code smoothing command for the DLL filter bandwidth. The 'value' field sets the L1 channel. This value is the reciprocal of the noise equivalent bandwidth of the DLL filter. Upon issuing the command, the locktime for all tracking satellites is reset to zero, and each pseudorange smoothing filter is restarted. You must wait for at least the length of the smoothing time for the new smoothing constant to take full effect. 20 seconds is the default smoothing constant used in the Portable MEDLL receiver.

***NOTES:*** The CSMOOTH command should only be used by advanced users. It may not be suitable for every GPS application. When using CSMOOTH in a differential mode, the same setting should be used at both the monitor and remote station.

The lower the smoothing constant, the noisier the measurement will be.

Syntax:

```
CSMOOTH    value
```

| Syntax | Range Value | Description | Start-Up Default | Parameter Default |
|---|---|---|---|---|
| CSMOOTH | - | Command | | |
| Value | 10 to 1000 | L1 code smoothing constant (in seconds). | 20 | |

Example:

```
csmooth 20
```

# ECUTOFF

This command sets the elevation cut-off angle for usable satellites. The receiver will not start tracking a satellite until it rises above the cut-off angle. If a satellite being tracked drops below this angle, it will no longer be tracked.

This command permits a negative cut-off angle, which could be used in these situations:

- the receiver is at a high altitude, and thus can look below the local horizon
- satellites are visible below the horizon due to atmospheric refraction

***NOTES:***
1. When ECUTOFF is set to zero (0), the receiver will track all satellites in view including some within a few degrees below the horizon.
2. Care should be taken when using ECUTOFF because the signals received from low-elevation satellites travel along a longer atmospheric path and thus tend to experience greater degradation.
3. If a satellite has been assigned to a specific channel (via the ASSIGN command), the receiver will continually attempt to track the satellite regardless of the elevation.

Syntax:

```
ECUTOFF angle
```

| Syntax | Range Value | Description | Start-Up Default |
|---|---|---|---|
| ECUTOFF | - | Command | |
| Angle | -90° to +90° | Angle value in degrees (relative to the horizon). | +5 |

Example:

```
ecutoff 5
```

# FIX POSITION

Invoking this command will result in the Portable MEDLL receiver position being held fixed. A computation will be done to solve local clock offset, pseudorange, and pseudorange differential corrections. This mode of operation can be used for time transfer applications where the position is fixed and accurate GPS time output is required (refer to the CLKA/B log on *Page 45* and the TM1A/B log on *Page 79* for time data).

The values entered into the FIX POSITION command should reflect the precise position of the antenna's phase centre. Any errors in these coordinates will be propagated into the pseudorange corrections calculated by the receiver.

The height is the distance above the geoid. The Portable MEDLL receiver performs all internal computations based on WGS-84. If you are going to input an ellipsoidal height you must first set the undulation to zero; please refer to the UNDULATION command on *Page 33*.

The FIX POSITION command will override any previous FIX POSITION command settings. Use the UNFIX command to disable the FIX POSITION setting.

Syntax:

```
FIX POSITION  lat  lon  height  [Station id]  [RTCM stn health]
```

| Syntax | Range Value | Description | Start-Up Default | Parameter Default |
|---|---|---|---|---|
| FIX POSITION | - | Command | UNFIX | |
| lat | ± 90<br>(only 8 decimal places are shown in the RCCA log, though higher precision is carried internally) | Latitude (in degrees/decimal degrees) of fixed monitor station antenna in current datum. A negative sign implies South latitude. | | |
| lon | ± 360<br>(only 8 decimal places are shown in the RCCA log, though higher precision is carried internally) | Longitude (in decimal degrees) of fixed monitor station antenna in current datum. A negative sign implies West longitude. | | |
| height | -1,000 to 20,000,000 | Height (in metres) above the geoid of the monitor station in current datum. | | |
| station id | Reserved | | | |
| Reference Station Health | | | | |

Example:

```
fix position 51.3455323,-114.289534,1201.123
```

The above example fixes the position of the receiver with fixed coordinates of:

| | |
|---|---|
| Latitude | N 51° 20' 43.9163" (WGS-84 or local datum) |
| Longitude | W 114° 17' 22.3224" |
| Height above sea level | 1201.123 metres |

# *FREQUENCY_OUT*

This command allows you to specify the frequency of the output pulse at the VARF pin of the I/O strobe connector. The frequency in Hz is calculated according to the formula below. The time between pulses may have up to 49 ns jitter variation from the true frequency pulse.

$$\texttt{FREQUENCY\_OUT} = \frac{\left[ 20473000 - \dfrac{20473000}{(n+1)} \right]}{(k+1)}$$

Syntax:

`frequency_out` `n` `k`

OR

`frequency_out` `disable`

| Syntax | Range Value | Description | Default | Example |
|---|---|---|---|---|
| FREQUENCY_OUT | - | Command | disable | frequency_out |
| n | 1 to 65535 | Variable integer | | 1 |
| k | 1 to 65535 | Variable integer | | 65535 |

Example 1:

```
frequency_out 1,65535
```

n=1, k=65535 results in an output pulse frequency of 156.196594 Hz

Example 2:

```
frequency_out 65535,1
```

n=65535, k=1 results in an output pulse frequency of 10,236,343.8034 Hz

As a reference, some n and k selections and their corresponding frequency outputs are listed in the following table:

| n | k | Frequency_Out (Hz) | |
|---|---|---|---|
| 1 | 65535 | 156.1966 | (Minimum) |
| 65535 | 65535 | 312.3884 | |
| 20472 | 20471 | 1 000.0000 | |
| 1569 | 2045 | 9 999.9804 | |
| 346 | 345 | 59 000.0000 | |
| 74 | 201 | 100 000.1320 | |
| 58 | 57 | 347 000.0000 | |
| 1 | 9 | 1 023 650.0000 | |
| 1 | 1 | 5 118 250.0000 | |
| 65535 | 1 | 10 236 343.8034 | (Maximum) |

If the 49 ns jitter is not suitable for your application, the following formula may be used to eliminate the jitter.

$$\texttt{FREQUENCY\_OUT} = \frac{20473000}{(k+1)}$$

where: N is constrained to 0, and K = 1 to 65535

*NOTE:* ***Frequency resolution of this method is not as fine as the original formula but provides jitter-free pulses.***

# LOG

Many different types of data can be logged using several different methods of triggering the log events. Every log element should be directed to a COM port. The ONTIME trigger option requires the addition of the *period* parameter and optionally allows input of the *offset* parameter. See the beginning of *Chapter 7* for further information about the ASCII and binary data log structures.

The '*port*' parameter is optional. If '*port*' is not specified, '*port*' is defaulted to the port that the command was received on. This feature eliminates the need for you to know which port you are communicating on if you want logs to come back to you on the same port by which you are sending commands.

If the LOG syntax does not include a '*trigger*' type, it will be output only once following execution of the LOG command. If '*trigger*' type is specified in the LOG syntax, the log will continue to be output based on the '*trigger*' specification.

The optional parameter '*hold*' will prevent a log from being removed when the UNLOGALL command is issued. To remove a log which was invoked using the '*hold*' parameter requires the specific use of the UNLOG command.

The COM1 and COM2 ports allow two-way communications. Thus, you would enter *COM1* as the port name.

Specific logs can be disabled using the UNLOG command, whereas all enabled logs will be disabled by using the UNLOGALL command (except for those issued with the '*hold*' parameter).

When the LOG command is ONTIME the software will attempt to output the log with the period chosen. The start of the period is the 1PPS signal. For Portable MEDLL the period will be 1 second, therefore, ONTIME 1 will cause logs to be output as soon as the data is available after the 1PPS. The OFFSET option is used to output the log with the period starting, not from the 1PPS but, at 1PPS plus the offset. The OFFSET option value is limited by the measurement frequency. That is, the software will only output a log after a measurement is taken. MEDLL's measurement frequency is 5 Hz. MEDLL's measurement log can be output at 1PPS plus the offset where the offset is one of 0.2, 0.4, 0.6 and 0.8 seconds. If you ask for 1PPS + 0.5 seconds, you will get data at 1PPS + 0.6 seconds. This measurement data is from the 1PPS + 0.6 measurement and not from the time of the 1PPS.

Examples:

```
log com1,etsb,ontime,60,1
log com1 etsb
log etsb ontime 60 1
```

The first example will cause the ETSB log to be sent to COM port 1, recurring every 60 seconds, and offset by 1 second. The second example will cause the ETSB log to be sent only once, by omitting the trigger option. The third example is identical to the first, since the only port on the Portable MEDLL receiver to which one can issue commands is configured as COM1.

Syntax:

`LOG [port] datatype [trigger] [period]`

| Syntax | Range Value | Description | Start-Up Default | Parameter Default |
|---|---|---|---|---|
| LOG | | | UNLOGALL | |
| port | | (Optional) COM1 is the only valid entry | | COM1 |
| datatype | | Enter one of the valid data log names (see *Chapter 7*) | | |
| trigger | | *ONCE*<br>Immediately logs the selected data to the selected port once. Default if trigger field is left blank.<br>*ONMARK*<br>Logs the selected data when a MARKIN electrical event is detected. Outputs internal buffers at time of mark – does not extrapolate to mark time.<br>*ONNEW*<br>Logs the selected data each time the data is new even if the data is unchanged.<br>*ONCHANGED*<br>Logs the selected data only when the data has changed.<br>*ONTIME*[*period*],[*offset*]<br>Immediately logs the selected data and then periodically logs the selected data at a frequency determined by the *period* and *offset* parameters. The logging will continue until an UNLOG command pertaining to the selected data item is received (*see UNLOG Command, on Page 34*).<br>*CONTINUOUSLY*<br>Will log the data all the time. The GPSCard will generate a new log when the output buffer associated with the chosen port becomes empty.<br>This may cause unpredictable results if more than one log is assigned to the port.<br>The *continuously* option was designed for use with differential corrections over low bit rate data links. This will provide optimal record generation rates. The next record will not be generated until the last byte of the previous record is loaded into the output buffer of the UART. | | ONCE |
| Period | 1-3600 | (Optional) Use only with the *Ontime* trigger. Units for this parameter are seconds. The selected period may be any value from 1 second to 3600 seconds. Selected data is logged immediately and then periodic logging of the data will start at the next even multiple of the period. If the period is 15 seconds, then the logger will log the data when the receiver time is at even 1/4 minute marks. The same rule applies even if the chosen period is not divisible into its next second or minute marks. If a period of 7 seconds is chosen, then the logger will log at the multiples of 7 seconds less than 60, that is, 7, 14, 21, 28, 35, 42, 49, 56 and every 7 seconds thereafter. | | |
| Offset | 0 to (period – 1) | (Optional) Use only with the *Ontime* trigger and with *Period*. Units for this parameter are seconds. It provides the ability to offset the logging events from the above start-up rule. If you wished to log data at 1 second after every minute you would set the period to 60 seconds and the offset to 1 second. The default is 0. | | |
| Hold | | (Optional) Prevents a log from being removed when the UNLOGALL command is issued; UNLOG must be used. | | |

## RESET

This command performs a CPU reset. Following a RESET command, the Portable MEDLL receiver will initiate a cold-start boot up. Therefore, the receiver configuration will revert to the factory default.

The RESET command does not reset the clock card.

Syntax:

```
RESET
```

## *$SETFRAMETYPE*

This command controls the type of navigation frame data that is output in the FRMA/B logs.

Syntax:

```
$SETFRAMETYPE      type
```

| Syntax | Range Value | Description | Start-Up Default |
|---|---|---|---|
| $SETFRAMETYPE | - | Command | |
| Type | ALL<br>GPS<br>GEO | Individual navigation frame data can be entered or the 'all' keyword can be used if the all types are needed. | ALL |

Example:

```
$setframetype gps
```

# UNASSIGN

This command cancels a previously issued ASSIGN command and the channel reverts to automatic control. If an L1 channel has reached channel tracking state 4 (L1 steady state tracking, see *Table 4, Page 53*), the satellite being tracked will not be dropped when the UNASSIGN command is issued unless it is below the elevation cut-off angle and there are healthy satellites above the elevation cut-off angle that are not already assigned to other channels.

Syntax:

```
UNASSIGN   channel
```

| Syntax | Range Value | Description | Start-Up Default |
|---|---|---|---|
| UNASSIGN | - | Command | N/A |
| Channel | 0 - highest channel number | Reset channel to automatic search and acquisition mode | |

Example:

```
unassign 11
```

# UNASSIGNALL

This command cancels all previously issued ASSIGN commands for all channels. Tracking and control for each channel reverts to automatic mode. If any of the L1 channels has reached channel tracking state 4 (L1 steady state tracking, see *Table 4, Page 53*), the satellites being tracked will not be dropped when the UNASSIGNALL command is issued unless they are below the elevation cut-off angle and there are healthy satellites above the elevation cut-off angle that are not already assigned to other channels.

Syntax:

```
UNASSIGNALL
```

# $UNASSIGNG2TOPRN

This command deletes a G2 delay assignment for a particular PRN from the user-defined table. It reverses a previous $ASSIGNG2TOPRN command. There are two syntactical forms, as shown below.

Syntax #1:

```
$UNASSIGNG2TOPRN   prn
```

| Syntax | Range Value | Description | Start-Up Default |
|---|---|---|---|
| $UNASSIGNG2TOPRN | - | Command | N/A |
| Prn | 1 – 999 | PRN Number | |

Example:

```
$unassigng2toprn 101
```

Syntax #2:

```
$UNASSIGNG2TOPRN   Keyword
```

| Syntax | Range Value | Description | Start-Up Default |
|---|---|---|---|
| $UNASSIGNG2TOPRN | - | Command | N/A |
| Keyword | ALL | Keyword - 'all' is the only defined keyword at this time. | |

Example:

```
$unassigng2toprn all
```

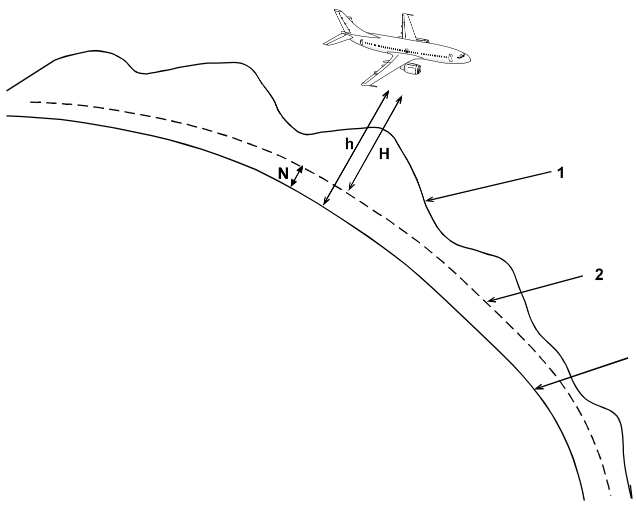
## *UNDULATION*

This command permits you to either enter a specific geoidal undulation value or use the internal table of geoidal undulations. The separation values only refer to the separation between the WGS-84 ellipsoid and the geoid at your location, regardless of the datum chosen. When you are going to input the ellipsoidal height using the FIX POSITION command you must first set the UNDULATION to zero.

*Figure 12* illustrates the various terms used in describing height relationships. In this figure:

1 = topography N = undulation
2 = geoid = mean sea level h = ellipsoidal height = height above ellipsoid
3 = ellipsoid H = height above mean sea level.

***Figure 12 Height Relationships***




Syntax:

```
UNDULATION   Separation
```

| Syntax | Range Value | Description | Start-Up Default |
|---|---|---|---|
| UNDULATION | - | Command | |
| Separation | Keyword "table"<br><br>or<br><br>a value | Selects the internal table of undulations and ignores any previously entered value. The internal table utilizes a grid (OSU - 89B) of approximately 1.5 degrees x 1.5 degrees.<br><br>A numeric entry that overrides the internal table, with a value in metres. | TABLE |

Examples:

```
undulation table
undulation -5.6
```

## UNFIX

This command removes all position constraints invoked with the FIX POSITION command.

Syntax:

```
UNFIX
```

## UNLOG

This command permits you to remove a specific log request from the system. It reverses the effect of a particular LOG command.

The [port] parameter is optional. If [port] is not specified, it is defaulted to the port that the command was received on. This feature eliminates the need for you to know which port you are communicating on if you want logs to come back on the same port you are sending commands on.

Syntax:

```
UNLOG   [port] datatype
```

| Syntax | Range Value | Description | Start-Up Default |
|---|---|---|---|
| UNLOG | - | Command | N/A |
| Port | COM1 | COM1 port from which log originated | |
| datatype | any valid log | The name of the log to be disabled | |

Example:

```
unlog com1,tm1b
```

## UNLOGALL

This command permits you to disable all current logs on the port to which your data communication equipment is connected. It reverses the effects of all LOG commands.

***NOTE:*** This command does not disable logs that have the ‘hold’ attribute (see the description for the LOG command on *Page 28*). To disable logs with the ‘hold’ attribute, use the UNLOG command.

Syntax:

```
UNLOGALL
```

## VERSION

Use this command to determine the current software version of the GPSCard. The response to the VERSION command is logged to the port from which the command originated.

Syntax:

```
VERSION [all]
```

Response:

| card type | model# | S/N | MACH rev | SW rev/boot code rev | date |
|---|---|---|---|---|---|

| Command | Description |
|---|---|
| VERSION | Software version of the master card |
| all | Software version of the master card and the five slave cards |

| Response | Description | Maximum Field Length (bytes) |
|---|---|---|
| card type | Card type | 5 |
| model # | Model number | 16 |
| S/N | Serial number | 16 |
| HW | Characters representing hardware (HW) | 2 |
| MACH rev | Hardware revision | 16 |
| SW | Characters representing software (SW) | 2 |
| SW rev/boot code rev | Firmware revision/ boot code revision | 16 |
| date | Firmware compile date | 12 |

Example:

```
COM1>version all
Card[1]  OEM-2 EGNOSMEDLL SGC00140018  HW 1  SW 5.4441S7/1.03 Sep 25/01
Card[2]  OEM-2  SGD01150132  HW 1  SW 5.4441S7/1.03 Sep 25/01
Card[3]  OEM-2  SGD01150106  HW 1  SW 5.4441S7/1.03 Sep 25/01
Card[4]  OEM-2  SGD01150105  HW 1  SW 5.4441S7/1.03 Sep 25/01
Card[5]  OEM-2  SGD01150119  HW 1  SW 5.4441S7/1.03 Sep 25/01
Card[6]  OEM-2  SGD01150118  HW 1  SW 5.4441S7/1.03 Sep 25/01
```

***NOTE:*** Spaces are delimiters in between fields.

# 6 OUTPUT LOGGING

When outputting data from the Portable MEDLL receiver, the logs must be sent to the COM1 or COM2 serial port.

*Table 2* shows the list of logs, arranged alphabetically.

**Table 2 Portable MEDLL Logs Summary**

| Logs | Descriptions | Message ID | Latency MEDLL Max | Min | Ave. | Std. Dev. |
|---|---|---|---|---|---|---|
| $AGCA/B | AGC and A/D information [6] | 74 | 126 | 88 | 108 | 8 |
| ALMA/B | Decoded almanac [5, 9] | 18 | 46 | 59 | 65 | 3 |
| CDSA/B | Communication and differential decode status [5] | 39 | N/A | | | |
| CLKA/B | Receiver clock offset data [6] | 02 | 181 | 116 | 147 | 15 |
| CRLA/B | Correlator location in C/A chips [5] | 96 | 173 | 75 | 138 | 11 |
| DOPA/B | Dilution of precision [5, 8] | 07 | N/A | | | |
| ETSA/B | Extended channel tracking status [5, 7] | 48 | N/A | | | |
| FRMA/B | Framed raw navigation data [6] | 54 | 492 | 443 | 466 | 10 |
| IONA/B | Decoded almanac-ionospheric model params.[5, 9] | 16 | N/A | | | |
| MPMA/B | Multipath meter [5] | 95 | 200 | 112 | 170 | 8 |
| POSA/B | Computed position [5, 8] | 01 | N/A | | | |
| RBTA/B | Satellite broadcast data: raw bits [5] | 52 | N/A | | | |
| RCCA | Receiver configuration [5] | N/A | N/A | | | |
| REPA/B | Raw ephemeris [5, 7] | 14 | N/A | | | |
| RGEA/B/D | Channel range measurements [5] | B: 32 D: 65 | 61 | 33 | 53 | 3 |
| RVSA/B | Receiver status [6] | 56 | 118 | 81 | 100 | 8 |
| SATA/B | Satellite Specific data [6] | 12 | 174 | 110 | 141 | 15 |
| SBTA/B | Satellite broadcast data: raw symbols [5] | 53 | N/A | | | |
| TM1A/B | Time of 1PPS [6] | 03 | 178 | 145 | 159 | 5 |
| UTCA/B | Decoded almanac-UTC time parameters [5, 9] | 17 | N/A | | | |
| WBCA/B | Wide band carrier range correction [5] | 97 | N/A | | | |
| WRCA/B | Wide band code range correction (inc. L1 & L2) [5] | 67 | 58 | 33 | 53 | 3 |

The data logs available are in NovAtel ASCII and binary format, and are described in *Chapter 7*. They can be logged using several methods of triggering each log event. Each log is initiated using the LOG command. The LOG command and syntax are described in *Chapter 5;* they are of the form ***log*** *[port],datatype,trigger,[period],[offset],[hold]*.

If the LOG syntax does not include a *trigger* type, it will be output only once following execution of the LOG command. If *trigger* type is specified in the LOG syntax, the log will continue to be output based on the *trigger* specification. Specific logs can be disabled using the UNLOG command, whereas all enabled logs will be disabled by using the UNLOGALL command unless a log has the hold attribute (see the UNLOG and UNLOGALL commands on *Page 34*).

The measurement rate on MEDLL is 5 Hz.

---

[5] Latency of logs calculated on a receiver that has only full processing load. The log under test is logged on the receiver under COM1. RVSB, RGEB, SFDB, FRMB and WRCB logs are logged on the receiver under COM2.

[6] Latency of logs calculated on a receiver that has full processing and communication port load. The latency is calculated as per the EGNOS validation test procedure. The active periodic logs are logged on the receiver under COM1.

*The footnotes that follow show the validity of logs when conditions exist.*

[7] Only valid after FINETIME is set.

[8] Only valid after a position solution has been computed.

[9] Only valid after the position of a complete, or partial, almanac.

# 7 NOVATEL FORMAT DATA LOGS

The Portable MEDLL receiver is capable of generating many NovAtel-format output logs, in either ASCII or binary format.

The following log descriptions are listed in alphabetical order. Each log first lists the ASCII format, then the binary format description.

## ASCII LOG STRUCTURE

Log types ending with the letter A are output in ASCII format (e.g., POSA). The structures of all ASCII logs follow the general conventions as noted here:

1. The lead code identifier for each record is '$'.
2. Each log is of variable length depending on amount of data and formats.
3. All data fields are delimited by a comma ',' with the exception of the last data field, which is followed by a '*' to indicate end-of-message data.
4. Each log ends with a hexadecimal number preceded by an asterisk and followed by a line termination using the carriage return and line feed characters, e.g., *xx[CR][LF]. This 8-bit value is an exclusive OR (XOR) of all bytes in the log, excluding the '$' identifier and the asterisk preceding the two checksum digits.

Structure:

```
$xxxx, | data field..., | data field..., | data field... | *xx | [CR][LF]
```

## BINARY LOG STRUCTURE

Log types ending with the letter B are output in binary format (e.g., POSB). The structures of the binary logs follow the general conventions as noted here:

1. Basic format of:

| | |
|---|---|
| Sync | 3 bytes |
| Checksum | 1 byte |
| Message ID | 4 bytes unsigned integer |
| Message byte count | 4 bytes unsigned integer |
| Data | x bytes |

2. The Sync bytes will always be:

| Byte | Hex | Decimal |
|---|---|---|
| First | AA | 170 |
| Second | 44 | 68 |
| Third | 11 | 17 |

3. The Checksum is an XOR of all the bytes, including the 12 header bytes with CRC = 00.
4. The Message ID identifies the type of log to follow.
5. The Message byte count equals the total length of the data block including the header.

**NOTE:** Maximum flexibility for logging data is provided by these logs. You are cautioned however, to recognize that each log requested requires additional CPU time and memory buffer space. Too many logs may result in lost data and degraded CPU performance. Buffer overload can be monitored using the idle time and buffer overload bits from the RVSB log. Please refer to *Table 11, Receiver Self-Test Status Codes on Page 69.*

The following describes the format types used in the description of binary logs.

| Type | Size (bytes) | Size (bits) | Description |
|---|---|---|---|
| char | 1 | 8 | The **char** type is used to store the integer value of a member of the representable character set. That integer value is the ASCII code corresponding to the specified character. |
| int | 4 | 32 | The size of a signed or unsigned **int** item is the standard size of an integer on a particular machine. On a 32-bit processor (such as the NovAtel GPSCard), the **int** type is 32 bits, or 4 bytes. The **int** types all represent signed values unless specified otherwise. Signed integers are represented in two's-complement form. The most-significant bit holds the sign: 1 for negative, 0 for positive and zero. |
| double | 8 | 64 | The **double** type contains 64 bits: 1 for sign, 11 for the exponent, and 52 for the mantissa. Its range is ±1.7E308 with at least 15 digits of precision. |
| float | 4 | 32 | The **float** type contains 32 bits: 1 for the sign, 8 for the exponent, and 23 for the mantissa. Its range is ±3.4E38 with at least 7 digits of precision. |
| long | 4 | 32 | The **long** type is a 32-bit integer in the range –2147483647 to +2147483648 |

Each byte within an **int** has its own address, and the smallest of the addresses is the address of the int. The byte at this lowest address contains the eight least significant bits of the double word, while the byte at the highest address contains the eight most significant bits. Similarly the bits of a "double" type are stored least significant byte first. This is the same data format used by personal computers.

### *COMPRESSED BINARY LOG STRUCTURE*

Log types ending with the letter D are also output in binary format (e.g., RGED). However, the RGED message is a compressed form of the RGEB message.

## *TIME CONVENTIONS*

All logs report GPS time expressed in GPS weeks and seconds into the week. The time reported is not corrected for the local receiver's clock error. To derive the closest GPS time, one must subtract the clock offset shown in the CLKB log (field 4) from GPS time reported.

GPS time is based on an atomic time scale. Universal Time Coordinated (UTC) time is also based on an atomic time scale, with an offset of seconds applied to coordinate Universal Time to GPS time. GPS time is designated as being coincident with UTC at the start date of January 6, 1980 (00 hours) GMT. GPS time does not count leap seconds, and therefore an offset exists between UTC and GPS time. The GPS week consists of 604800 seconds, where 000000 seconds is at Saturday midnight. Each week at this time, the week number increments by one, and the seconds into the week resets to 0. See also the section on *Compliance with GPS Week Rollover on Page viii.*

## *LOG DESCRIPTIONS*

The log references that follow are in alphabetical order as in *Table 2, Portable MEDLL Logs Summary* on *Page 36*.

For binary logs, the 32-bit CRC is calculated with all fields in the log filled except for the checksum field, which is zero. For the ASCII logs, the 32-bit CRC is calculated from all fields of the log after the '$' symbol. Once the CRC has been calculated, the log checksum is calculated in the normal fashion, in order to preserve the standard NovAtel log format.

The GPS seconds into the week value is stored in a number of different ways within the receiver. Depending on which log is being output, the time may be derived from a different source. However, all sources of time are interconnected. Generally, the seconds into the week field is stored as either an integer (in milliseconds) or a floating-point value before being output in the log.

The CRC for the Portable MEDLL logs uses the standardized IEEE 802 CRC algorithm. The expanded polynomial for the algorithm is:

$$X^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1$$

## $AGCA/B AGC AND A/D INFORMATION

The AGCA/B log contains AGC related information. This information includes information about the AGC gain, A/D bin values and AGC control statistics.

### $AGCA

Structure:

```
$AGCA | week | seconds | rec_status | # RF
RF_type | bin 1 | bin 2 | bin 3 | bin 4 | bin 5 | bin 6
Gain | 1ms_AGC | 1ms_Chan | bins_rms | gof_stat
:
next RF deck if required | *xx | [CR][LF]
```

| Field # | Field type | Data Description | Example |
|---|---|---|---|
| 1 | $AGCA | Log header | $AGCA |
| 2 | week | GPS week number | 932 |
| 3 | seconds | GPS seconds into the week | 256542.00 |
| 4 | rec_status | Self-test status of the receiver (see *Table 11, Page 69*) | 43A00FF |
| 5 | # RF | Number of RF decks reported in this message. | 2 |
| 6 | RF_type | 0 = GPS_L1, all others are reserved for future usage. | 0 |
| 7 | bin 1 | A/D Bin 1 | 0.1022 |
| 8 | bin 2 | A/D Bin 2 | 0.1813 |
| 9 | bin 3 | A/D Bin 3 | 0.2380 |
| 10 | bin 4 | A/D Bin 4 | 0.2363 |
| 11 | bin 5 | A/D Bin 5 | 0.1558 |
| 12 | bin 6 | A/D Bin 6 | 0.0864 |
| 13 | gain | AGC Gain | 3125 |
| 14 | 1ms_AGC | Estimate of the receiver noise based on an A/D data histogram. 1 ms noise floor calculated using the A/D bin values. | 1557822.00 |
| 15 | 1ms_Chan | Estimate of the receiver noise that is either an AGC estimate or measured power in a 1 ms post correlation accumulation from a searching channel. | 1557822.00 |
| 16 | bins_rms | Estimate of the variance of the receiver noise based on time smoothed A/D data histograms. Root mean squared value of the A/D bins calculated using the expected values as truth. | 0.9957 |
| 17 | gof_stat | The chi-squared test statistic estimating the goodness of fit between a current A/D data histogram to an A/D data histogram that was obtained at start-up by averaging several A/D histogram samples. Goodness of Fit test statistic for the A/D bins. | 0.000008 |
| 18...29 | .... | Next RF Deck if required. | .... |
| Variable | *xx | Checksum | *66 |
| Variable | [CR][LF] | Sentence terminator | [CR][LF] |

Example:

$AGCA,932,256542.00,43A00FF,2,0,0.1022,0.1813,0.2380,0.2363,0.1558,0.0864,3125,1557822.00,1557822.00,0.9957,0.000008,1,0.0973,0.1722,0.2353,0.2406,0.1637,0.0909,3361,1552060.00,1552060.00,0.9935,0.000042*66 [CR][LF]

### $AGCB

Format: Message ID = 74

Message byte count = 32 + (r * 48) where r is the # of RF decks.

| Field # | Data | Bytes | Format | Units | Offset |
|---|---|---|---|---|---|
| 1 (header) | Sync | 3 | Char | | 0 |
| | Checksum | 1 | Char | | 3 |
| | Message ID | 4 | Integer | | 4 |
| | Message byte count | 4 | Integer | bytes | 8 |
| 2 | Week number | 4 | Integer | weeks | 12 |
| 3 | Time of week | 8 | Double | seconds | 16 |
| 4 | Receiver status | 4 | Integer | (See *Table 11, Page 69*) | 24 |
| 5 | No. of RF decks | 4 | Integer | number of receiver RF decks | 28 |
| 6 | RF Type | 4 | Integer | 0 = GPS_L1, all others are reserved for future usage. | 32 |
| 7 | A/D Bin 1 (Most Neg.) | 4 | Float | percentage | 36 |
| 8 | A/D Bin 2 | 4 | Float | percentage | 40 |
| 9 | A/D Bin 3 | 4 | Float | percentage | 44 |
| 10 | A/D Bin 4 | 4 | Float | percentage | 48 |
| 11 | A/D Bin 5 | 4 | Float | percentage | 52 |
| 12 | A/D Bin 6 (Most Pos.) | 4 | Float | percentage | 56 |
| 13 | AGC Gain | 4 | Integer | 0 to 99,999 (0 is max) | 60 |
| 14 | 1ms Noise Floor From AGC | 4 | Float | | 64 |
| 15 | 1ms Noise Floor From Channels | 4 | Float | | 68 |
| 16 | A/D Bins RMS | 4 | Float | | 72 |
| 17 | A/D Goodness of Fit Test Statistic | 4 | Float | | 76 |
| 18...29 | Next RF Deck if required | 48 | | | 80 |

## ALMA/B DECODED ALMANAC

This log contains the decoded almanac parameters from sub-frames four and five as received from the satellite with the parity information removed and appropriate scaling applied. Multiple messages are transmitted, one for each SV almanac collected.

The Ionospheric Model parameters (IONA/B) and the UTC Time parameters (UTCA/B) are also provided, following the last almanac record. These cannot be logged individually or independently of the ALMA/B message. They are, however, described separately under their respective names.

For more information on Almanac data, refer to the GPS SPS Signal Specification.

***NOTE:*** In order to limit CPU loading while decoding the almanac, all almanac data is decoded from a single tracked satellite, arbitrarily chosen among all tracking channels.

### ALMA

Structure:

| $ALMA | Prn | ecc | seconds | week | $\dot{w}$ | ra | w | $M_0$ | $a_{f0}$ | $a_{f1}$ | cor-mean-motion |
|---|---|---|---|---|---|---|---|---|---|---|---|
| A | incl-angle | health-4 | health-5 | health-alm | *xx | [CR][LF] | | | | | |

| Field # | Field type | Data Description | Example |
|---|---|---|---|
| 1 | $ALMA | Log header | $ALMA |
| 2 | prn | Satellite PRN number for current message (1-32) | 1 |
| 3 | ecc | Eccentricity | 3.55577E-003 |
| 4 | seconds | Almanac reference time, seconds into the week | 32768 |
| 5 | week | Almanac reference week (GPS week number) | 745 |
| 6 | $\dot{w}$ | Rate of right ascension, radians | -7.8860E-009 |
| 7 | ra | Right ascension, radians | -6.0052951E-002 |
| 8 | w | Argument of perigee, radians | -1.1824254E+000 |
| 9 | $M_0$ | Mean anomaly, radians | 1.67892137E+000 |
| 10 | $a_{f0}$ | Clock aging parameter, seconds | -1.8119E-005 |
| 11 | $a_{f1}$ | Clock aging parameter, seconds/second | -3.6379E-012 |
| 12 | cor-mean-motion | Corrected mean motion, radians/second | 1.45854965E-004 |
| 13 | A | Semi-major axis, metres | 2.65602281E+007 |
| 14 | incl-angle | Angle of inclination, radians | 9.55576E-001 |
| 15 | health-4 | Anti-spoofing and SV config from sub-frame 4, page 25 | 1 |
| 16 | health-5 | SV health, 6 bits/SV (sub-frame 4 or 5, page 25) | 0 |
| 17 | health-alm | SV health, 8 bits (almanac) | 0 |
| 18 | *xx | Checksum | *20 |
| 19 | [CR][LF] | Sentence terminator | [CR][LF] |
| : | | | |
| 1 - 19 | $ALMA | Last satellite PRN almanac message | |
| 1 - 11 | $IONA | Ionospheric Model Parameters | |
| 1 - 11 | $UTCA | UTC Time Parameters | |

Example:

```
$ALMA,1,4.99010E-003,503808,67,-7.8975E-009,5.58933014E-001,-1.7435100E+000,
-1.3147095E+000,1.55449E-004,0.00000E+000,1.45861599E-004,2.65594229E+007,
9.62689E-001,1,0,0*22[CR][LF]
```

…

```
$ALMA,31,9.92775E-003,503808,67,-8.1832E-009,-2.6301490E+000,8.33547783E-001,
-2.8544401E-001,2.19345E-005,0.00000E+000,1.45849203E-004,2.65609277E+007,
9.47985E-001,1,0,0*19[CR][LF}
```

```
$IONA,2.3283064365386962E-008,0.0000000000000000E+000,-1.192092895507812E-007,
1.1920928955078122E-007,1.4336000000000018E+005,-1.966080000000002E+005,0.
0000000000000000E+000,1.9660800000000019E+005*0A[CR][LF]
```

```
$UTCA,1.8626451492309570E-008,2.8421709430404010E-014,503808,67,990,13,13,
5*05[CR][LF]
```

## ALMB

Format: Message ID = 18 Message byte count = 120

| Field # | Field Type | | Bytes | Format | Units | Offset |
|---|---|---|---|---|---|---|
| 1 (header) | Sync | | 3 | | | 0 |
| | Checksum | | 1 | | | 3 |
| | Message ID | | 4 | | | 4 |
| | Message byte count | | 4 | integer | bytes | 8 |
| 2 | Satellite PRN number | | 4 | integer | dimensionless | 12 |
| 3 | Eccentricity | | 8 | double | dimensionless | 16 |
| 4 | Almanac ref. time | | 8 | double | seconds | 24 |
| 5 | Almanac ref. week | | 4 | integer | weeks | 32 |
| 6 | Rate of right ascension | $\dot{w}$ | 8 | double | radians/second | 36 |
| 7 | Right ascension | | 8 | double | radians | 44 |
| 8 | Argument of perigee | w | 8 | double | radians | 52 |
| 9 | Mean anomaly | Mo | 8 | double | radians | 60 |
| 10 | Clock aging parameter | $a_{f0}$ | 8 | double | seconds | 68 |
| 11 | Clock aging parameter | $a_{f1}$ | 8 | double | seconds/second | 76 |
| 12 | Corrected mean motion | | 8 | double | radians/second | 84 |
| 13 | Semi-major axis | A | 8 | double | metres | 92 |
| 14 | Angle of inclination | | 8 | double | radians | 100 |
| 15 | SV health from sub-frame 4 | | 4 | integer | | 108 |
| 16 | SV health from sub-frame 4 or 5 | | 4 | Integer | | 112 |
| 17 | SV health from almanac | | 4 | Integer | | 116 |

## CDSA/B COMMUNICATION AND DIFFERENTIAL DECODE STATUS

The Portable MEDLL receiver maintains a running count of a variety of status indicators of the data link. This log outputs a report of those indicators.

Parity and framing errors will occur if poor transmission lines are encountered or if there is an incompatibility in the data protocol. If errors occur, you may need to confirm the baud rate, number of data bits, number of stop bits, and parity of both the transmitting and receiving ends. Overrun errors will occur if more characters are sent to the UART than can be removed by the on-board microprocessor.

### CDSA

Structure:

```
$CDSA  week  seconds  xon1  cts1  parity1  overrun1  framing1  rx1  tx1
Xon2  cts2  parity2  overrun2  framing2  rx2  tx2
Res'd  Res'd  Res'd  Res'd  Res'd  Res'd  Res'd  Res'd  Res'd  Res'd
Res'd  Res'd  Res'd  *xx  [CR][LF]
```

| Field # | Field type | Data Description | Example |
|---|---|---|---|
| 1 | $CDSA | Log header | $CDSA |
| 2 | week | GPS week number | 787 |
| 3 | seconds | GPS seconds into the week | 500227 |
| 4 | xon1 | Flag to indicate that the com1 is using XON/XOFF handshaking protocol and port has received an XOFF and will wait for an XON before sending any more data. | 0 |
| 5 | cts1 | Flag to indicate that com1 is using CTS/RTS handshake protocol and CTS line port has been asserted. The port will wait for the line to de-assert before sending any more data. | 0 |
| 6 | parity1 | A running count of character parity errors from the UART of COM1 | 0 |
| 7 | overrun1 | A running count of UART buffer overrun errors of COM1 | 0 |
| 8 | framing1 | A running count of character framing error from the UART of COM1 | 0 |
| 9 | rx1 | A running count of the characters received from COM1 | 0 |
| 10 | tx1 | A running count of the characters sent out COM1 | 9 |
| 11 | xon2 | Flag to indicate that the COM2 is using XON/XOFF handshaking protocol and port has received an XOFF and will wait for an XON before sending any more data. | 0 |
| 12 | cts2 | Flag to indicate that COM2 is using CTS/RTS handshake protocol and CTS line port has been asserted. The port will wait for the line to de-assert before sending any more data. | 0 |
| 13 | parity2 | A running count of character parity errors from the UART of COM2 | 0 |
| 14 | overrun2 | A running count of UART buffer overrun errors of COM2 | 0 |
| 15 | framing2 | A running count of character framing error from the UART of COM2 | 0 |
| 16 | rx2 | A running count of the characters received from COM2 | 0 |
| 17 | tx2 | A running count of the characters sent out COM2 | 9 |
| 18 | | Reserved. Set to 0. | 0 |
| 19 | | Reserved. Set to 0. | 0 |
| 20 | | Reserved. Set to 0. | 0 |
| 21 | | Reserved. Set to 0. | 0 |
| 22 | | Reserved. Set to 0. | 0 |
| 23 | | Reserved. Set to 0. | 0 |
| 24 | | Reserved. Set to 0. | 0 |
| 25 | | Reserved. Set to 0. | 0 |
| 26 | | Reserved. Set to 0. | 0 |
| 27 | | Reserved. Set to 0. | 0 |
| 28 | | Reserved. Set to 0. | 0 |
| 29 | | Reserved. Set to 0. | 0 |
| 30 | | Reserved. Set to 0. | 0 |

| Field # | Field type | Data Description | Example |
|---|---|---|---|
| 31 | *xx | Checksum | *33 |
| 32 | [CR][LF] | Sentence terminator | [CR][LF] |

Example: $CDSA,787,500227,0,0,0,0,0,0,9,0,0,0,0,0,0,9,0,0,0,0,0,0,0,0,0,0,0,0,0,0*33[CR][LF]

## *CDSB*

Format: Message ID = 39 Message byte count = 128

| Field # | Data | Bytes | Format | Units | Offset |
|---|---|---|---|---|---|
| 1 (header) | Sync | 3 | char | | 0 |
| | Checksum | 1 | char | | 3 |
| | Message ID | 4 | integer | | 4 |
| | Message byte count | 4 | integer | bytes | 8 |
| 2 | Week number | 4 | integer | weeks | 12 |
| 3 | Time of week | 4 | integer | seconds | 16 |
| 4 | Xon COM1 | 4 | integer | 1 or 0 | 20 |
| 5 | CTS COM1 | 4 | integer | 1 or 0 | 24 |
| 6 | Parity errors COM1 | 4 | integer | Total count | 28 |
| 7 | Overrun errors COM1 | 4 | integer | Total count | 32 |
| 8 | Framing error COM1 | 4 | integer | Total count | 36 |
| 9 | Bytes received in COM1 | 4 | integer | Total count | 40 |
| 10 | Bytes sent out COM1 | 4 | integer | Total count | 44 |
| 11 | Xon COM2 | 4 | integer | 1 or 0 | 48 |
| 12 | CTS COM2 | 4 | integer | 1 or 0 | 52 |
| 13 | Parity errors COM2 | 4 | integer | Total count | 56 |
| 14 | Overrun errors COM2 | 4 | integer | Total count | 60 |
| 15 | Framing error COM2 | 4 | integer | Total count | 64 |
| 16 | Bytes received in COM2 | 4 | integer | Total count | 68 |
| 17 | Bytes sent out COM2 | 4 | integer | Total count | 72 |
| 18 | Reserved. Set to 0. | 4 | integer | Total count | 76 |
| 19 | Reserved. Set to 0. | 4 | integer | Total count | 80 |
| 20 | Reserved. Set to 0. | 4 | integer | Total count | 84 |
| 21 | Reserved. Set to 0. | 4 | integer | Total count | 88 |
| 22 | Reserved. Set to 0. | 4 | integer | Total count | 92 |
| 23 | Reserved. Set to 0. | 4 | integer | Total count | 96 |
| 24 | Reserved. Set to 0. | 4 | integer | Total count | 100 |
| 25 | Reserved. Set to 0. | 4 | integer | Total count | 104 |
| 26 | Reserved. Set to 0. | 4 | integer | Total count | 108 |
| 27 | Reserved. Set to 0. | 4 | integer | Total count | 112 |
| 28 | Reserved. Set to 0. | 4 | integer | Total count | 116 |
| 29 | Reserved. Set to 0. | 4 | integer | Total count | 120 |
| 30 | Reserved. Set to 0. | 4 | integer | Total count | 124 |

4

## CLKA/B RECEIVER CLOCK OFFSET DATA

This log is used to monitor the state of the receiver time. Its values will depend on the CLOCKADJUST command. If CLOCKADJUST is enabled, then the offset and drift times will approach zero. If not enabled, then the offset will grow at the oscillator drift rate. Disabling CLOCKADJUST and monitoring the CLKA/B log will allow you to determine the error in your GPSCard receiver reference oscillator as compared to the GPS satellite reference.

All logs report GPS time not corrected for local receiver clock error. To derive the closest GPS time, you must subtract the clock offset (field #4 of the CLKA log) from the reported GPS time.

Field #6 is the output of a Gauss-Markov Selective Availability clock dither estimator. This value reflects both the collective SA-induced short-term drift of the satellite clocks, when SA is active, as well as any range bias discontinuities that would normally affect the clock model's offset and drift states. With SA off this value is much smaller than the example shown.

The CLKA/B log is not synchronous to the 1PPS.

Clock model status is not related to clock type. There is no effect on the validity of the clock model data when the CLOCKADJUST command is set to DISABLE. Upon receiver start-up, the data in the CLKA/B log is not valid. After a position-time solution this log will show the difference between the 1PPS and GPS epoch. The CM STATUS field shows the current stability of the clock model. If CLOCKADJUST is disabled, CM STATUS is meaningless. CLOCKADJUST is disabled by default for Portable MEDLL.

The internal units of the clock model's three states (offset, drift and GM state) are meters, meters per second, and meters. When scaled to time units for the output log, these become seconds, seconds per second, and seconds, respectively.

### CLKA

Structure:

```
$CLKA week seconds offset drift G-M state offset std drift std cm status
*xx [CR][LF]
```

| Field # | Field type | Data Description | Example |
|---|---|---|---|
| 1 | $CLKA | Log header | $CLKA |
| 2 | week | GPS week number | 637 |
| 3 | seconds | GPS seconds into the week | 511323.00 |
| 4 | offset | Receiver clock offset (s). A positive offset implies that the receiver clock is ahead of GPS Time. To derive GPS time, use the following formula:<br>GPS time = receiver time – (offset) | -4.628358547E-003 |
| 5 | drift | Receiver clock drift (s/s). A positive drift implies that the receiver clock is running faster than GPS Time. | -2.239751396E-007 |
| 6 | G-M state | The output value of the Gauss-Markov Selective Availability clock dither estimator (s). | 2.061788299E-006 |
| 7 | offset std | Standard deviation of receiver clock offset (s). | 5.369997167E-008 |
| 8 | drift std | Standard deviation of receiver drift (s/s) | 4.449097711E-009 |
| 9 | cm status | Receiver Clock Model Status (0 = valid, -20 to -1 imply that the model is in the process of stabilization) | 0 |
| 10 | *xx | Checksum | *7F |
| 11 | [CR][LF] | Sentence terminator | [CR][LF] |

Example:

```
$CLKA,841,499296.00,9.521895494E-008,-2.69065747E-008,2.061788299E-006,
9.642598169E-008,8.685638908E-010,0*4F[CR][LF]
```

### CLKB

Format: Message ID = 02 Message byte count = 68

| Field # | Field Type | Bytes | Format | Units | Offset |
|---|---|---|---|---|---|
| 1 (header) | Sync | 3 | char | | 0 |
| | Checksum | 1 | char | | 3 |
| | Message ID | 4 | integer | | 4 |
| | Message byte count | 4 | integer | bytes | 8 |
| 2 | Week number | 4 | integer | weeks | 12 |
| 3 | Seconds of week | 8 | double | s | 16 |
| 4 | Clock offset | 8 | double | s | 24 |
| 5 | Clock drift | 8 | double | s/s | 32 |
| 6 | SA Gauss-Markov state | 8 | double | s | 40 |
| 7 | StdDev clock offset | 8 | double | s | 48 |
| 8 | StdDev clock drift | 8 | double | s/s | 56 |
| 9 | Clock model status | 4 | integer | 0 = good, -1 to -20 = bad | 64 |

## CRLA/B CORRELATOR LOCATION IN C/A CHIPS

It is recommended that this be output only 'once' or 'ontime'. The 'onnew' and 'onchanged' options have been disabled. The order that the correlator locations are output is identical to the order of the residuals in the MPMA/B (multipath meter) log.

### CRLA

Structure:

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| `$CRLA` | `week` | `sec` | `#chans` | `chan#` | `#corr` | `1st corr` | … | `12th corr` | | |
| … | | | | | | | | | | |
| `$CRLA` | `week` | `sec` | `#chans` | `chan#` | `#corr` | `1st corr` | … | `12th corr` | `*xx` | `[CR][LF]` |

| Field # | Field Type | Data Description | Examples |
|---|---|---|---|
| 1 | $CRLA | | $CRLA |
| 2 | Week Number | GPS week number | 2 |
| 3 | Seconds of Week | GPS seconds into the week | 234534.56 |
| 4 | Number of Channels | | 16 |
| 5 | Channel Number | | 0 |
| 6 | Number of Correlators | | 12 |
| 7 | 1st correlator location | Location of correlator 1 (in C/A chips) | -0.04 |
| | … | Repeated for each correlator | … |
| 18 | 12th correlator location | Location of correlator 12 | 1.40 |
| 19… | 5 – 19 Repeated for # of Channels | | … |
| variable | *xx | Checksum | *29 |
| variable | [CR][LF] | Sentence terminator | [CR][LF] |

Example:

```
$CRLA,61,158572.947,16,
0,12,-0.15,-0.10,-0.05,0.00,0.05,0.10,0.15,0.30,0.50,0.70,0.90,1.30,
1,12,-0.15,-0.10,-0.05,0.00,0.05,0.10,0.15,0.30,0.50,0.70,0.90,1.30,
…
15,12,-0.60,-0.40,-0.20,0.00,0.20,0.40,0.60,0.80,1.00,1.20,1.40,1.60*28 [CR][LF]
```

### CRLB

Format: Message ID = 96 Message byte count = 28+(#chans*56)

| Field # | Data | Bytes | Format | Units | Offset |
|---|---|---|---|---|---|
| 1 (header) | Sync | 3 | Char | | 0 |
| | Checksum | 1 | Char | | 3 |
| | Message ID | 4 | Integer | | 4 |
| | Message Byte Count | 4 | Integer | bytes | 8 |
| 2 | Week Number | 4 | Integer | weeks | 12 |
| 3 | Seconds of Week | 8 | Double | seconds | 16 |
| 4 | Number of Channels | 4 | Integer | | 24 |
| 5 | Channel Number | 4 | Integer | | 28 |
| 6 | Number of Correlators | 4 | Integer | | 32 |
| 7 | 1st correlator location | 4 | Float | C/A chips | 36 |
| | … | | | | |
| 18 | 12th correlator location | 4 | Float | C/A chips | 80 |
| 19… | Next channel number<br>Offset = 28 + (chan#*56) where chan# = 0 to (number of channels-1) | | | | |

## DOPA/B DILUTION OF PRECISION

The dilution of precision data is calculated using the geometry of only those satellites that are currently being tracked and used in the position solution by the GPSCard and updated once every 60 seconds. Therefore, the total number of data fields output by the log is variable, depending on the number of satellites being tracked. Twelve is the maximum number of satellite PRNs contained in the list.

***NOTE:*** If insufficient satellites are being tracked to calculate DOP values, the last calculated DOP values are output.

### DOPA

Structure:

```
$DOPA week sec gdop pdop htdop hdop tdop #sats prns *xx [CR][LF]
```

| Field # | Field type | Data Description | Example |
|---|---|---|---|
| 1 | $DOPA | Log header | $DOPA |
| 2 | week | GPS week number | 637 |
| 3 | sec | GPS seconds into the week | 512473.00 |
| 4 | gdop | Geometric dilution of precision - assumes 3-D position and receiver clock offset (all 4 parameters) are unknown | 2.9644 |
| 5 | pdop | Position dilution of precision - assumes 3-D position is unknown and receiver clock offset is known | 2.5639 |
| 6 | htdop | Horizontal position and time dilution of precision | 2.0200 |
| 7 | hdop | Horizontal dilution of precision | 1.3662 |
| 8 | tdop | Time dilution of precision - assumes 3-D position is known and only receiver clock offset is unknown | 1.4880 |
| 9 | #sats | Number of satellites used in position solution (0-12). See the notes above. | 6 |
| 10... | prns | PRN list of SV PRNs tracking (1-32), null field until first position solution available | 18,6,11,2,16,19 |
| variable | *xx | Checksum | *29 |
| variable | [CR][LF] | Sentence terminator | [CR][LF] |

Example:

```
$DOPA,637,512473.00,2.9644,2.5639,2.0200,1.3662,1.4880,6,18,6,11,2,16,19
*29[CR][LF]
```

## DOPB

Format: Message ID = 07 Message byte count = 68+(#sats*4)

| Field # | Data | Bytes | Format | Units | Offset |
|---|---|---|---|---|---|
| 1 (header) | Sync | 3 | char | | 0 |
| | Checksum | 1 | char | | 3 |
| | Message ID | 4 | integer | | 4 |
| | Message byte count | 4 | integer | bytes | 8 |
| 2 | Week number | 4 | integer | weeks | 12 |
| 3 | Seconds of week | 8 | double | seconds | 16 |
| 4 | gdop | 8 | double | | 24 |
| 5 | pdop | 8 | double | | 32 |
| 6 | htdop | 8 | double | | 40 |
| 7 | hdop | 8 | double | | 48 |
| 8 | tdop | 8 | double | | 56 |
| 9 | Number of satellites used | 4 | integer | | 64 |
| 10 | 1st PRN | 4 | integer | | 68 |
| 11... | Next satellite PRN<br>Offset = 68 + (sats*4) where sats = 0 to (number of sats-1) | | | | |

## ETSA/B EXTENDED CHANNEL TRACKING STATUS

These logs provide channel tracking status information for each of the GPSCard parallel channels.

**NOTE:** This log is intended for status display only; since some of the data elements are not synchronized together, they are not to be used for measurement data. Please use the RGEA/B/D and SATA/B logs to obtain synchronized data for post processing analysis.

As shown in *Table 4* (Channel Tracking Status word), bit 19 shows the observation is for L1. This is to aid in parsing the data.

### ETSA

Structure:

```
$ETSA  week  seconds  sol status  # obs
prn  ch tr-status  dopp  C/No  residual  locktime  psr  reject code
:
prn  ch tr-status  dopp  C/No  residual  locktime  psr  reject code  *xx  [CR][LF]
```

| Field # | Field type | Data Description | Example |
|---|---|---|---|
| 1 | $ETSA | Log header | $ETSA |
| 2 | week | GPS week number | 850 |
| 3 | seconds | GPS seconds into the week (receiver time, not corrected for clock error, CLOCKADJUST enabled) | 332087.00 |
| 4 | sol status | Solution status (see *Table 3, on Page 53*) | 0 |
| 5 | # obs | Number of observations to follow | 24 |
| 6 | prn | Satellite PRN number (1-32) (channel 0) [10] | 7 |
| 7 | ch tr-status | Hexadecimal number indicating channel tracking status (See *Table 4, on Page 53*) | 00082E04 |
| 8 | dopp | Instantaneous carrier Doppler frequency (Hz) | -613.5 |
| 9 | C/No | Carrier to noise density ratio (dB-Hz) | 54.682 |
| 10 | residual | Residual from position filter (m) | 27.617 |
| 11 | locktime | Number of seconds of continuous tracking (no cycle slips) | 12301.4 |
| 12 | psr | Pseudorange measurement (m) | 20257359.57 |
| 13 | reject code | Indicates whether the range is valid (code = 0) or not (see *Table 7, on Page 54*) | 0 |
| 14-21<br>..<br>94-101 | ..<br>..<br>.. | next observation<br>..<br>last observation | |
| 102 | *xx | Checksum | *19 |
| 103 | [CR][LF] | Sentence terminator | [CR][LF] |

[10] Satellite PRN = 0 if the channel is idle.

Example (carriage returns have been added between observations for clarity):

```
$ETSA,850,332087.00,0,24,
7,00082E04,-613.5,54.682,27.617,12301.4,20257359.57,0,
7,00582E0B,-478.1,46.388,0.000,11892.0,20257351.96,13,
5,00082E14,3311.2,35.915,1.037,1224.4,24412632.47,0,
5,00582E1B,2580.4,39.563,0.000,1186.7,24412629.40,13,
9,00082E24,1183.1,53.294,-29.857,7283.8,21498303.67,0,
9,00582E2B,921.9,44.422,0.000,7250.2,21498297.13,13,
2,00082E34,-2405.2,50.824,-20.985,19223.6,22047005.47,0,
2,00582E3B,-1874.1,41.918,0.000,19186.7,22046999.44,13,
4,00082E44,3302.8,47.287,7.522,3648.1,22696783.36,0,
4,00582E4B,2573.6,37.341,0.000,3191.2,22696778.15,13,
14,00082E54,2132.7,41.786,-22.388,541.3,25117182.07,0,
14,00582E5B,1661.7,33.903,0.000,500.7,25117179.63,13,
26,00082E64,-3004.3,43.223,2.928,14536.2,25074382.19,0,
26,00582E6B,-2340.9,33.019,0.000,14491.7,25074378.01,13,
15,00082E74,-3037.7,43.669,0.508,12011.5,24104788.88,0,
15,00582E7B,-2367.0,34.765,0.000,11842.4,24104781.53,13,
24,00082E84,3814.0,37.081,7.511,95.7,25360032.49,0,
24,00582E8B,2972.0,24.148,0.000,5.2,25360030.13,13,
28,00082A90,-9800.9,0.000,0.000,0.0,0.00,9,
28,00382A90,-7637.0,0.000,0.000,0.0,0.00,9,
3,000822A0,-3328.3,0.000,0.000,0.0,0.00,9,
3,005828A0,-2593.5,0.000,0.000,0.0,0.00,9,
27,000822B0,-3851.7,0.000,0.000,0.0,0.00,9,
27,005828B0,-3001.7,0.000,0.000,0.0,0.00,9
*41[CR][LF]
```

## ETSB

Format: Message ID = 48
Message byte count = 32 + (n*52) where n is number of channels in receiver

| Field # | Data | Bytes | Format | Units | Offset |
|---|---|---|---|---|---|
| 1 (header) | Sync | 3 | char | | 0 |
| | Checksum | 1 | char | | 3 |
| | Message ID | 4 | integer | | 4 |
| | Message byte count | 4 | integer | bytes | 8 |
| 2 | Week number | 4 | integer | weeks | 12 |
| 3 | Time of week | 8 | double | seconds | 16 |
| 4 | Solution status | 4 | integer | (See *Table 3, on Page 53*) | 24 |
| 5 | No. of channels | 4 | integer | number of channels in receiver | 28 |
| 6 | PRN number (chan 0) | 4 | integer | | 32 |
| 7 | Channel tracking status | 4 | integer | (See *Table 4, on Page 53*) | 36 |
| 8 | Doppler | 8 | double | Hz | 40 |
| 9 | $C/N_0$ (db-Hz) | 8 | double | db Hz | 48 |
| 10 | Residual | 8 | double | metres | 56 |
| 11 | Locktime | 8 | double | seconds | 64 |
| 12 | Pseudorange | 8 | double | metres | 72 |
| 13 | Rejection code | 4 | integer | (See *Table 7, on Page 54*) | 80 |
| 14 ... | Offset = 32 + (chan * 52) where chan varies from 0 - highest channel number | | | | |

### *Table 3 GPSCard Solution Status*

| Value | Description |
|---|---|
| 0 | Solution computed |
| 1 | Insufficient observations |
| 2 | No convergence |
| 3 | Singular AtPA Matrix |
| 4 | Covariance trace exceeds maximum (trace > 1000 m) |
| 5 | Test distance exceeded (maximum of 3 rej if distance > 10 Km) |
| 6 | Not yet converged from cold start |

Higher numbers are reserved for future use

### *Table 4 Channel Tracking Status Bits*

N 7 | N 6 | N 5 | N 4 | N 3 | N 2 | N 1 | N 0 <-<- Nibble Number

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

| Bit | Description | Range Values | Hex. |
|---|---|---|---|
| lsb=0 | | | 1 |
| 1 | Tracking state | 0-11 See below | 2 |
| 2 | | | 4 |
| 3 | | | 8 |
| 4 | | | 10 |
| 5 | | 0-n (0=first, n=last) | 20 |
| 6 | Channel number | (n depends on GPSCard model) | 40 |
| 7 | | | 80 |
| 8 | | | 100 |
| 9 | Phase lock flag[8] | 1=Lock, 0=Not locked | 200 |
| 10 | Parity known flag[8] | 1=Known, 0=Not known | 400 |
| 11 | Code locked flag[8] | 1=Lock, 0=Not locked | 800 |
| 12 | | | 1000 |
| 13 | Correlator spacing type | 0-7 See the Correlator | 2000 |
| 14 | | Spacing Table[1] | 4000 |
| 15 | | 0=GPS 3=Reserved | 8000 |
| 16 | Satellite system | 1=Reserved 4-7 Reserved | 10000 |
| 17 | | 2=GEO[2] | 20000 |
| 18 | Antenna | 1=Secondary, 0=Primary[3] | 40000 |
| 19 | Grouping[4] | 1=Grouped, 0=Not grouped | 80000 |
| 20 | Frequency[5] | 1=L2, 0=L1 | 100000 |
| 21 | Code type[6] | 0=C/A 2=P-codeless | 200000 |
| 22 | | 1=P 3=Reserved | 400000 |
| 23 | Forward error correction | 1=FEC enabled, 0=no FEC | 800000 |
| 24 | | | |
| : | Reserved. Set to 0. | | |
| 29 | | | |
| 30 | External range | 1=Ext. range, 0=Int. range | |
| 31 | Channel assignment | 1=Forced, 0=Automatic | |

1 See ***Table 5****, Bits 12-14: Correlator Spacing* on *Page 54*.
2 GEO will never be set for MEDLL channels 0-9
3 Antenna will always be primary
4 MEDLL is always 0
5 MEDLL is always 0
6 MEDLL is always 0
7 MEDLL channels 10-11 will be 1 and MEDLL channels 0-9 will be 0.
8 When phase, parity, and code lock have been established, the channel has reached steady state tracking in state 4.

### Table 5 Bits 0 - 3 : Channel Tracking State

| Value of Bits 0-3 | Description | Value of Bits 0-3 | Description |
|---|---|---|---|
| 0 | L1 Idle (State 0) | 6 | Reserved. (State 6) |
| 1 | L1 Sky search (State 1) | 7 | L1 Frequency-lock loop (State 7) |
| 2 | L1 Dual frequency band pull-in (State 2) | 8 | Reserved |
| 3 | L1 Narrow frequency band pull-in (State 3) | 9 | |
| 4 | L1 Steady state tracking (State 4) | 10 | |
| 5 | L1 Re-acquisition (State 5) | 11 | |

Higher numbers are reserved for future use

### Table 6 Bits 12-14 : Correlator Spacing

| State | Description |
|---|---|
| 0 | Unknown: this only appears in versions of software previous to x.4x, which didn't use this field |
| 1 | Standard correlator: spacing = 1 chip |
| 2 | Reserved |
| 3 | MET: uses Early-Late Slope Technique to improve correlator performance in reducing errors due to multipath |
| 4 | Reserved. |
| 5 | MEDLL: decomposes the incoming signal into direct-path and reflected-path components to reduce errors due to multipath |

Higher numbers are reserved for future use

### Table 7 Range Reject Codes

| Value | Description |
|---|---|
| 0 | Observations are good |
| 1 | Bad satellite health is indicated by ephemeris data |
| 2 | Old ephemeris due to data not being updated during last 3 hours |
| 3 | Eccentric anomaly error during computation of the satellite's position |
| 4 | True anomaly error during computation of the satellite's position |
| 5 | Satellite coordinate error during computation of the satellite's position |
| 6 | Elevation error due to the satellite being below the cut-off angle |
| 7 | Misclosure too large due to excessive gap between estimated and actual positions |
| 8 | No differential correction is available for this particular satellite |
| 9 | Ephemeris data for this satellite has not yet been received |
| 10 | Invalid IODE due to mismatch between differential stations |
| 11 | Locked out of the position solution by the user |
| 12 | Low Power: satellite rejected due to low signal/noise ratio |
| 13 | Reserved. |
| 14 | |
| 15 | |
| 16 | |
| 17 | GEO satellite not used in the position filter for Portable MEDLL |

Higher numbers are reserved for future use

## FRMA/B FRAMED RAW NAVIGATION DATA

This message contains the raw framed navigation data. An individual message is sent for each PRN being tracked. The message is updated with each new frame, therefore it is best to log the data with the 'onnew' trigger activated.

The types of navigation frame data which are output by this log are defined using the $SETFRAMETYPE command.

**NOTES:** The most significant bits of the data in the message will be padded with zeroes in order to fill the last complete data byte.

When the raw framed navigation data is output from the receiver, any bit errors encountered will have been corrected.

### FRMA

```
$FRMA | week | seconds | prn | cstatus | # of bits | framed raw data
*xx | [CR][LF]
```

| Field # | Field type | Data Description | Example |
|---|---|---|---|
| 1 | $FRMA | Log header | $FRMA |
| 2 | week | GPS week number | 845 |
| 3 | seconds | GPS seconds into the week | 238623.412 |
| 4 | prn | PRN of satellite from which data originated | 120 |
| 5 | cstatus | Channel Tracking Status | 80811F14 |
| 6 | # of bits | Number of bits transmitted in the message. 250 for GEO, 300 for GPS | 250 |
| 7 | framed raw data | One field of raw framed navigation data | 9AFE5354656C2053796E6368726F6E69636974792020202020202020B0029E40 |
| 8 | *xx | Checksum | *3F |
| 9 | [CR][LF] | Sentence terminator | [CR][LF] |

Example:

```
$FRMA,845,238623.412,120,80811F14,250,
9AFE5354656C2053796E6368726F6E696369747920202020202020202020B0029E40*3F[CR][LF]
```

Data transmitted by the SV in this example is hexadecimal 9AFE5354656C2053796E6368726F6E6963697479 2020202020202020B0029E plus 2 bits (01) with the bit on the left sent first. The receiver outputs the log as hexadecimal 9AFE5354656C2053796E6368726F6E69636974792020202020202020B0029E plus hexadecimal 40 (binary 01 with 6 zeroes of padding → 01000000).

### FRMB

Format: Message ID = 54 Message byte count = variable

| Field # | Data | Bytes | Format | Units | Offset |
|---|---|---|---|---|---|
| 1 (header) | Sync | 3 | char | | 0 |
| | Checksum | 1 | char | | 3 |
| | Message ID | 4 | integer | | 4 |
| | Message byte count | 4 | integer | bytes | 8 |
| 2 | Week number | 4 | integer | weeks | 12 |
| 3 | Seconds of week | 8 | double | seconds | 16 |
| 4 | PRN number | 4 | integer | 1-999 | 24 |
| 5 | Channel Status | 4 | integer | N/A | 28 |
| 6 | Number of Bits | 4 | integer | 250 for GEO; 300 for GPS | 32 |
| 7 | Data Sub-frame | 32 for GEO; 38 for GPS | char | N/A | 36 |

## IONA/B DECODED ALMANAC - IONOSPHERIC MODEL PARAMETERS

The Ionospheric Model parameters (IONA/B) are provided following the last almanac records when an ALMA/B message has been logged. The IONA/B message cannot be logged individually or independently of the ALMA/B message.

For more information on Almanac data, refer to the GPS SPS Signal Specification[11].

### IONA

Structure:

```
$IONA  act  a1ot  a2ot  a3ot  bct  b1ot  b2ot  b3ot  *xx  [CR][LF]
```

| Field # | Field type | Data Description | Example |
|---|---|---|---|
| 1 | $IONA | Log header | $IONA |
| 2 | act | Alpha constant term, seconds | 1.0244548320770265E-008 |
| 3 | a1ot | Alpha 1st order term, sec/semicircle | 1.4901161193847656E-008 |
| 4 | a2ot | Alpha 2nd order term, sec/(semic.)$^2$ | -5.960464477539061E-008 |
| 5 | a3ot | Alpha 3rd order term, sec/(semic.)$^3$ | -1.192092895507812E-007 |
| 6 | bct | Beta constant term, seconds | 8.8064000000000017E+004 |
| 7 | b1ot | Beta 1st order term, sec/semicircle | 3.2768000000000010E+004 |
| 8 | b2ot | Beta 2nd order term, sec/(semic.)$^2$ | -1.966080000000001E+005 |
| 9 | b3ot | Beta 3rd order term, sec/(semic.)$^3$ | -1.966080000000001E+005 |
| 10 | *xx | Checksum | *02 |
| 11 | [CR][LF] | Sentence terminator | [CR][LF] |

Example:

```
$IONA,1.0244548320770265E-008,1.4901161193847656E-008,-5.960464477539061E-008,
-1.192092895507812E-007,8.8064000000000017E+004,3.2768000000000010E+004,
-1.966080000000001E+005,-1.966080000000001E+005*02[CR][LF]
```

[11] For copies of the Interface Control Document (ICD)-GPS-200 contact: ARINC Research Corporation, 2551 Riva Road, Annapolis, MD 21401-7465. Their website is at http://www.arinc.com/.

### IONB

Format Message ID = 16 Message byte count = 76

| Field # | Field Type | Bytes | Format | Units | Offset |
|---|---|---|---|---|---|
| 1 (header) | Sync | 3 | char | | 0 |
| | Checksum | 1 | char | | 3 |
| | Message ID | 4 | integer | | 4 |
| | Message byte count | 4 | integer | bytes | 8 |
| 2 | Alpha constant term | 8 | double | seconds | 12 |
| 3 | Alpha 1st order term | 8 | double | sec/semicircle | 20 |
| 4 | Alpha 2nd order term | 8 | double | $sec/(semic.)^2$ | 28 |
| 5 | Alpha 3rd order term | 8 | double | $sec/(semic.)^3$ | 36 |
| 6 | Beta constant term | 8 | double | seconds | 44 |
| 7 | Beta 1st order term | 8 | double | sec/semic | 52 |
| 8 | Beta 2nd order term | 8 | double | $sec/(semic.)^2$ | 60 |
| 9 | Beta 3rd order term | 8 | double | $sec/(semic.)^3$ | 68 |

## MPMA/B MULTIPATH METER

This log outputs information that estimates the amount of multipath the antenna is experiencing and how well MEDLL has modelled the multipath signals.

It is recommended that this log be output only with the 'onnew' trigger option. There will be one log for every tracked satellite per epoch. For example, if eleven satellites are being tracked, there will be eleven instances of this log every epoch. MEDLL runs every second, so one epoch is equivalent to one second.

### MPMA

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $MPMA | week | seconds | prn | chtrstat | medllstat | delay | amplitude | phase |
| 1st in phase | … | 12th in phase | | | | | | |
| 1st quad. phase | … | 12th quad. phase | | | | | | |
| xx | [CR][LF] | | | | | | | |

| Field # | Field Type | Data Description | Example |
|---|---|---|---|
| 1 | $MPMA | Log header | $MPMA |
| 2 | Week Number | GPS week number | 0 |
| 3 | Seconds of Week | GPS seconds into the week | 27.77 |
| 4 | PRN | Satellite identifier | 29 |
| 5 | Channel Tracking Status [12] | Channel tracking status bits, see *Table 4 on 53* | 6A84 |
| 6 | MEDLL Status | MEDLL status bits, see *Table 8 on Page 59* | 103 |
| 7 | Delay | Delay of multipath signal in chips | 1.08154941 |
| 8 | Amplitude | Amplitude of multipath signal | 0.01731431 |
| 9 | Phase | Phase of multipath signal in radians | - 0.00645047 |
| 10 | 1st in phase residual | In phase residual value from correlator 1 | 0.00160142 |
| | … | Repeated for each correlator | … |
| 22 | 12th in phase residual | In phase residual value from correlator 12 | - 0.00196318 |
| 23 | 1st Quadrature phase residual | Quadrature phase residual value from correlator 1 | - 0.00418267 |
| | … | Repeated for each correlator | … |
| 45 | 12th Quadrature phase residual | Quadrature phase residual value from correlator 12 | - 0.00730140 |
| 46 | *xx | Checksum | *71 |
| 47 | [CR][LF] | Sentence terminator | [CR][LF] |

Example:

```
$MPMA,0,27.77,29,6A84,103,1.08154941,0.01731431,-0.00645047,0.00160142,
0.00164832,0.00078163,0.00001205,0.00083644,0.00240084,0.00214321,0.00079274,
-0.00032872,-0.00084985,0.00000891,-0.00196318,-0.00418267,-0.00443155,
-0.00665589,-0.00580890,-0.00078493,-0.00101275,0.00468876,0.00449791,
-0.00126932,-0.00140528,0.00004035,-0.00730140*71[CR][LF]
```

[12] A description can be found on *Page 53, Table 4*

### MPMB

Format: Message ID = 95 Message byte count = 144

| Field # | Data | Bytes | Format | Units | Offset |
|---|---|---|---|---|---|
| 1 | Sync | 3 | Char | | 0 |
| (header) | Checksum | 1 | Char | | 3 |
| | Message ID | 4 | Integer | | 4 |
| | Message Byte Count | 4 | Integer | bytes | 8 |
| 2 | Week Number | 4 | Integer | weeks | 12 |
| 3 | Seconds of Week | 8 | Double | seconds | 16 |
| 4 | PRN | 4 | Integer | | 24 |
| 5 | Channel Status | 4 | Integer | | 28 |
| 6 | MEDLL Status | 4 | Integer | | 32 |
| 7 | Delay | 4 | Float | C/A chips | 36 |
| 8 | Amplitude | 4 | Float | | 40 |
| 9 | Phase | 4 | Float | | 44 |
| 10 | 1st in phase residual | 4 | Float | | 48 |
| | … | | | | |
| 22 | 12th in phase residual | 4 | Float | | 92 |
| 23 | 1st Quadrature phase residual | 4 | Float | | 96 |
| | … | | | | |
| 45 | 12th Quadrature phase residual | 4 | Float | | 140 |

The multipath amplitude and residuals are normalized with respect to the reference correlation function. D/U (desired signal power relative to undesired signal power), in units of decibels (dB), can be calculated from the amplitude of the multipath signal (-20 * log [amplitude of multipath signal]).

**Table 8 MEDLL Status Bits Table**

| Bit Position | Field Description |
|---|---|
| 0 | Sync bit: 1 if MEDLL channels in sync, 0 if not in sync. Sync bit is set to 1 if the different hardware channels use to track the same PRN are aligned. |
| 1 | Phase processing. When phase processing is on, the bit is 1; MEDLL will determine the phase of the multipath signal. If phase processing is set to 0, MEDLL will not process the phase of the multipath signal. For Portable MEDLL, the bit will always be 1 by default. |
| 2-6 | Type of MEDLL will always be 0 for Portable MEDLL. |
| 7-9 | Number of signals will always be 1 for Portable MEDLL. |
| 10-31 | Reserved. Set to 0. |

## POSA/B COMPUTED POSITION

This log will contain the last valid position and time calculated referenced to the antenna phase centre. The position is in geographic coordinates in degrees based on your specified datum (default is WGS-84). The height is referenced to mean sea level. The receiver time is in GPS weeks and seconds into the week. The estimated standard deviations of the solution and current filter status are also included.

### POSA

Structure:

```
$POSA | week | seconds | lat | lon | hgt | undulation | datum ID | lat std | lon std
hgt std | sol status | *xx | [CR][LF]
```

| | Field type | Data Description | Example |
|---|---|---|---|
| 1 | $POSA | Log header | $POSA |
| 2 | week | GPS week number | 637 |
| 3 | seconds | GPS seconds into the week | 511251.00 |
| 4 | lat | Latitude of position in current datum, in degrees (DD.dddddddd). A negative sign implies South latitude | 51.11161847 |
| 5 | lon | Longitude of position in current datum, in degrees (DDD.dddddddd). A negative sign implies West longitude | -114.03922149 |
| 6 | hgt | Height of position in current datum, in metres above mean sea level (MSL) | 1072.436 |
| 7 | undulation | Geoidal separation, in metres, where positive is above spheroid and negative is below spheroid | -16.198 |
| 8 | datum ID | Current datum ID #, see *Table 9, Page 61* and *Table 10, Page 62.* | 61 |
| 9 | lat std | Standard deviation of latitude solution element, in metres | 26.636 |
| 10 | lon std | Standard deviation of longitude solution element, in metres | 6.758 |
| 11 | hgt std | Standard deviation of height solution element, in metres | 78.459 |
| 12 | sol status | Solution status as listed in *Table 3, Page 53.* | 0 |
| 13 | *xx | Checksum | *12 |
| 14 | [CR][LF] | Sentence terminator | [CR][LF] |

Example:

```
$POSA,637,511251.00,51.11161847,-114.03922149,1072.436,-16.198,61,26.636,
6.758,78.459,0*12[CR][LF]
```

## *POSB*

Format: Message ID = 01 Message byte count = 88

| Field # | Data | | Format | Units | Offset |
|---|---|---|---|---|---|
| 1 (header) | Sync | 3 | char | | 0 |
| | Checksum | 1 | char | | 3 |
| | Message ID | 4 | integer | | 4 |
| | Message byte count | 4 | integer | bytes | 8 |
| 2 | Week number | 4 | integer | weeks | 12 |
| 3 | Seconds of week | 8 | double | seconds | 16 |
| 4 | Latitude | 8 | double | degrees (+ is North, - is South) | 24 |
| 5 | Longitude | 8 | double | degrees (+ is East, - is West) | 32 |
| 6 | Height | 8 | double | metres with respect to MSL | 40 |
| 7 | Undulation | 8 | double | metres | 48 |
| 8 | Datum ID | 4 | integer | | 56 |
| 9 | StdDev of latitude | 8 | double | metres | 60 |
| 10 | StdDev of longitude | 8 | double | metres | 68 |
| 11 | StdDev of height | 8 | double | metres | 76 |
| 12 | Solution status | 4 | integer | | 84 |

The following tables contain the internal ellipsoid parameters and transformation parameters used in the GPSCard. The values contained in these tables were derived from the following DMA technical reports:

1. TR 8350.2 Department of Defence World Geodetic System 1984 – Its Definition and Relationships with Local Geodetic Systems - Revised March 1, 1988.
2. TR 8350.2B Supplement to Department of Defence World Geodetic System 1984 Technical Report - Part II - Parameters, Formulas, and Graphics for the Practical Application of WGS-84 - December 1, 1987.

***Table 9 Reference Ellipsoid Constants***

| ELLIPSOID | ID CODE | a (metres) | 1/f | |
|---|---|---|---|---|
| Airy 1830 | AW | 6377563.396 | 299.3249647 | 0.00334085064038 |
| Modified Airy | AM | 6377340.189 | 299.3249647 | 0.00334085064038 |
| Australian National | AN | 6378160.0 | 298.25 | 0.00335289186924 |
| Bessel 1841 | BR | 6377397.155 | 299.1528128 | 0.00334277318217 |
| Clarke 1866 | CC | 6378206.4 | 294.9786982 | 0.00339007530409 |
| Clarke 1880 | CD | 6378249.145 | 293.465 | 0.00340756137870 |
| Everest (India 1830) | EA | 6377276.345 | 300.8017 | 0.00332444929666 |
| Everest (Brunei & E.Malaysia) | EB | 6377298.556 | 300.8017 | 0.00332444929666 |
| Everest (W.Malaysia & Singapore) | ED | 6377304.063 | 300.8017 | 0.00332444929666 |
| Geodetic Reference System 1980 | RF | 6378137.0 | 298.257222101 | 0.00335281068118 |
| Helmert 1906 | HE | 6378200.0 | 298.30 | 0.00335232986926 |
| Hough 1960 | HO | 6378270.0 | 297.00 | 0.00336700336700 |
| International 1924 | IN | 6378388.0 | 297.00 | 0.00336700336700 |
| South American 1969 | SA | 6378160.0 | 298.25 | 0.00335289186924 |
| World Geodetic System 1972 | WD | 6378135.0 | 298.26 | 0.00335277945417 |
| World Geodetic System 1984 | WE | 6378137.0 | 298.257223563 | 0.00335281066475 |

**Table 10 Transformation Parameters (Local Geodetic to WGS-84)**

| GPSCard Datum ID number | NAME | DX | DY | DZ | DATUM DESCRIPTION | ELLIPSOID |
|---|---|---|---|---|---|---|
| 1 | ADIND | -162 | -12 | 206 | Adindan (Ethiopia, Mali, Senegal & Sudan) | Clarke 1880 |
| 2 | ARC50 | -143 | -90 | -294 | ARC 1950 (SW & SE Africa) | Clarke 1880 |
| 3 | ARC60 | -160 | -8 | -300 | ARC 1960 (Kenya, Tanzania) | Clarke 1880 |
| 4 | AGD66 | -133 | -48 | 148 | Australian Geodetic Datum 1966 | Australian National |
| 5 | AGD84 | -134 | -48 | 149 | Australian Geodetic Datum 1984 | Australian National |
| 6 | BUKIT | -384 | 664 | -48 | Bukit Rimpah (Indonesia) | Bessel 1841 |
| 7 | ASTRO | -104 | -129 | 239 | Camp Area Astro (Antarctica) | International 1924 |
| 8 | CHATM | 175 | -38 | 113 | Chatum 1971 (New Zealand) | International 1924 |
| 9 | CARTH | -263 | 6 | 431 | Carthage (Tunisia) | Clarke 1880 |
| 10 | CAPE | -136 | -108 | -292 | CAPE (South Africa) | Clarke 1880 |
| 11 | DJAKA | -377 | 681 | -50 | Djakarta (Indonesia) | Bessel 1841 |
| 12 | EGYPT | -130 | 110 | -13 | Old Egyptian | Helmert 1906 |
| 13 | ED50 | -87 | -98 | -121 | European 1950 | International 1924 |
| 14 | ED79 | -86 | -98 | -119 | European 1979 | International 1924 |
| 15 | GUNSG | -403 | 684 | 41 | G. Segara (Kalimantan - Indonesia) | Bessel 1841 |
| 16 | GEO49 | 84 | -22 | 209 | Geodetic Datum 1949 (New Zealand) | International 1924 |
| 17 | GRB36 | 375 | -111 | 431 | Great Britain 1936 (Ordinance Survey) | Airy 1830 |
| 18 | GUAM | -100 | -248 | 259 | Guam 1963 (Guam Island) | Clarke 1866 |
| 19 | HAWAII | 89 | -279 | -183 | Hawaiian Hawaii (Old) | International 1924 |
| 20 | KAUAI | 45 | -290 | -172 | Hawaiian Kauai (Old) | International 1924 |
| 21 | MAUI | 65 | -290 | -190 | Hawaiian Maui (Old) | International 1924 |
| 22 | OAHU | 56 | -284 | -181 | Hawaiian Oahu (Old) | International 1924 |
| 23 | HERAT | -333 | -222 | 114 | Herat North (Afghanistan) | International 1924 |
| 24 | HJORS | -73 | 46 | -86 | Hjorsey 1955 (Iceland) | International 1924 |
| 25 | HONGK | -156 | -271 | -189 | Hong Kong 1963 | International 1924 |
| 26 | HUTZU | -634 | -549 | -201 | Hu-Tzu-Shan (Taiwan) | International 1924 |
| 27 | INDIA | 289 | 734 | 257 | Indian (India, Nepal, Bangladesh) | Everest (EA) |
| 28 | IRE65 | 506 | -122 | 611 | Ireland 1965 | Modified Airy |
| 29 | KERTA | -11 | 851 | 5 | Kertau 1948 (West Malaysia and Singapore) | Everest (ED) |
| 30 | KANDA | -97 | 787 | 86 | Kandawala (Sri Lanka) | Everest (EA) |
| 31 | LIBER | -90 | 40 | 88 | Liberia 1964 | Clarke 1880 |
| 32 | LUZON | -133 | -771 | -51 | Luzon (Philippines excluding Mindanoa Is.) | Clarke 1866 |
| 33 | MINDA | -133 | -70 | -72 | Mindanoa Island | Clarke 1866 |
| 34 | MERCH | 31 | 146 | 47 | Merchich (Morocco) | Clarke 1880 |
| 35 | NAHR | -231 | -196 | 482 | Nahrwan (Saudi Arabia) | Clarke 1880 |
| 36 | NAD83 | 0 | 0 | 0 | N. American 1983 (Includes Areas 37-42) | GRS-80 |
| 37 | CANADA | -10 | 158 | 187 | N. American Canada 1927 | Clarke 1866 |
| 38 | ALASKA | -5 | 135 | 172 | N. American Alaska 1927 | Clarke 1866 |
| 39 | NAD27 | -8 | 160 | 176 | N. American Conus 1927 | Clarke 1866 |
| 40 | CARIBB | -7 | 152 | 178 | N. American Caribbean | Clarke 1866 |
| 41 | MEXICO | -12 | 130 | 190 | N. American Mexico | Clarke 1866 |
| 42 | CAMER | 0 | 125 | 194 | N. American Central America | Clarke 1866 |
| 43 | MINNA | -92 | -93 | 122 | Nigeria (Minna) | Clarke 1880 |
| 44 | OMAN | -346 | -1 | 224 | Oman | Clarke 1880 |
| 45 | PUERTO | 11 | 72 | -101 | Puerto Rica and Virgin Islands | Clarke 1866 |
| GPSCard Datum ID number | NAME | DX | DY | DZ | DATUM DESCRIPTION | ELLIPSOID |

| | | | | | | |
|---|---|---|---|---|---|---|
| 46 | QORNO | 164 | 138 | -189 | Qornoq (South Greenland) | International 1924 |
| 47 | ROME | -255 | -65 | 9 | Rome 1940 Sardinia Island | International 1924 |
| 48 | CHUA | -134 | 229 | -29 | South American Chua Astro (Paraguay) | International 1924 |
| 49 | SAM56 | -288 | 175 | -376 | South American (Provisional 1956) | International 1924 |
| 50 | SAM69 | -57 | 1 | -41 | South American 1969 | S. American 1969 |
| 51 | CAMPO | -148 | 136 | 90 | S. American Campo Inchauspe (Argentina) | International 1924 |
| 52 | SACOR | -206 | 172 | -6 | South American Corrego Alegre (Brazil) | International 1924 |
| 53 | YACAR | -155 | 171 | 37 | South American Yacare (Uruguay) | International 1924 |
| 54 | TANAN | -189 | -242 | -91 | Tananarive Observatory 1925 (Madagascar) | International 1924 |
| 55 | TIMBA | -689 | 691 | -46 | Timbalai (Brunei and East Malaysia) 1948 | Everest (EB) |
| 56 | TOKYO | -128 | 481 | 664 | Tokyo (Japan, Korea and Okinawa) | Bessel 1841 |
| 57 | TRIST | -632 | 438 | -609 | Tristan Astro 1968 (Tristan du Cunha) | International 1924 |
| 58 | VITI | 51 | 391 | -36 | Viti Levu 1916 (Fiji Islands) | Clarke 1880 |
| 59 | WAK60 | 101 | 52 | -39 | Wake-Eniwetok (Marshall Islands) | Hough 1960 |
| 60 | WGS72 | 0 | 0 | 4.5 | World Geodetic System - 72 | WGS-72 |
| 61 | WGS84 | 0 | 0 | 0 | World Geodetic System - 84 | WGS-84 |
| 62 | ZANDE | -265 | 120 | -358 | Zanderidj (Surinam) | International 1924 |
| 63 | USER | 0 | 0 | 0 | User Defined Datum Defaults | User * |

* Default user datum is WGS 84.

* The POSA/B log reports the Datum used according to the “GPSCard Datum ID” column.

## RBTA/B SATELLITE BROADCAST DATA: RAW BITS

This message contains the satellite broadcast data in raw bits before Forward Error Correction (FEC) decoding or any other processing. An individual message is sent for each PRN being tracked. For a given satellite, the message number increments by one each time a new message is generated. This data matches the SBTA/B data if the message numbers are equal. The data must be logged with the 'onnew' trigger activated to prevent loss of data.

### RBTA

Structure:

```
$RBTA | week | seconds | prn | ch tr-status | message # | # of bits
raw bits | *xx | [CR][LF]
```

| Field # | Field type | Data Description | Example |
|---|---|---|---|
| 1 | $RBTA | Log header | $RBTA |
| 2 | week | GPS week number | 883 |
| 3 | seconds | GPS seconds into the week | 413908.000 |
| 4 | prn | PRN of satellite from which data originated | 115 |
| 5 | ch tr-status | Channel Tracking Status (see *Table 4, on Page 53*) | 80812F14 |
| 6 | message # | Message sequence number | 119300 |
| 7 | # of bits | Number of bits transmitted in the message. At present, always equals 256 bits. | 256 |
| 8 | raw bits | 256 bits compressed into a 32 bytes. Hence, 64 hex characters are output. | 30FB30FB30FB30F878DA62194000F18322931B 9EBDBC1CBC9324B68FBDAEBE8A |
| 9 | *xx | Checksum | *42 |
| 10 | [CR][LF] | Sentence terminator | [CR][LF] |

```
Example:
$RBTA,883,413908.000,115,80812F14,119300,256,30FB30FB30FB30F878DA62194000F1832293
1B9EBDBC1CBC9324B68FBDAEBE8A*42[CR][LF]
```

### RBTB

Format: Message ID = 52 Message byte count = 72

| Field # | Data | Bytes | Format | Units | Offset |
|---|---|---|---|---|---|
| 1 (header) | Sync | 3 | char | | 0 |
| | Checksum | 1 | char | | 3 |
| | Message ID | 4 | integer | | 4 |
| | Message byte count | 4 | integer | bytes | 8 |
| 2 | Week number | 4 | integer | weeks | 12 |
| 3 | Seconds of week | 8 | double | seconds | 16 |
| 4 | PRN number | 4 | integer | 1-999 | 24 |
| 5 | Channel Tracking Status | 4 | integer | n/a | 28 |
| 6 | Message # | 4 | integer | n/a | 32 |
| 7 | # of Bits | 4 | integer | n/a | 36 |
| 8 | Raw Bits | 32 | char | n/a | 40 |

## RCCA RECEIVER CONFIGURATION

This log outputs a list of the current receiver command settings. Observing this log is a good way to monitor the receiver's configuration settings. If these messages are logged directly after powering up the receiver, they will contain the default configuration settings.

***NOTE1:*** In addition to typical log data as described in this manual, there may be additional messages output by the receiver (see *Appendix C on Page 88,* for more on information messages).

***NOTE2:*** Each line is followed by a carriage return ([CR][LF]).

### RCCA

Example default:

```
$RCCA,COM1,9600,N,8,1,N,OFF,ON*2B
$RCCA,COM1_DTR,HIGH*70
$RCCA,COM1_RTS,HIGH*67
$RCCA,ACCEPT,COM1,COMMANDS*5B
$RCCA,COM2,9600,N,8,1,N,OFF,ON*28
$RCCA,COM2_DTR,HIGH*73
$RCCA,COM2_RTS,HIGH*64
$RCCA,ACCEPT,COM2,COMMANDS*58
$RCCA,UNDULATION,TABLE*56
$RCCA,DATUM,WGS84*15
$RCCA,USERDATUM,6378137.000,298.257223563,0.000,0.000,0.000,0.000,0.000,0.000,0.0
    00*6A
$RCCA,SETNAV,DISABLE*5C
$RCCA,MAGVAR,0.000*33
$RCCA,DYNAMICS,AIR*4F
$RCCA,UNASSIGNALL*64
$RCCA,UNLOCKOUTALL*20
$RCCA,RESETHEALTHALL*37
$RCCA,UNFIX*73
$RCCA,SETDGPSID,ALL*1D
$RCCA,RTCMRULE,6CR*32
$RCCA,RTCM16T,*48
$RCCA,CSMOOTH,20.00,20.00*7E
$RCCA,ECUTOFF,5.00*40
$RCCA,FREQUENCY_OUT,DISABLE*12
$RCCA,EXTERNALCLOCK,OCXO*5D
$RCCA,CLOCKADJUST,ENABLE*47
$RCCA,SETTIMESYNC,DISABLE*17
$RCCA,SETL1OFFSET,0.000000*3F
$RCCA,MESSAGES,COM1,ON*56
$RCCA,MESSAGES,COM2,OFF*1B
$RCCA,MESSAGES,CONSOLE,ON*71
$RCCA,DGPSTIMEOUT,60.00,120.00*51
$RCCA,POSAVE,DISABLE*59
$RCCA,CONFIG,G10W2*3A
$RCCA,DIFF_PROTOCOL,DISABLED*47
$RCCA,UNLOGALL*21
```

## REPA/B RAW EPHEMERIS

This log contains the raw binary information for sub-frames one, two and three from the satellite with the parity information removed. Each sub-frame is 240 bits long (10 words – 24 bits each) and the log contains a total 720 bits (90 bytes) of information (240 bits x 3 sub-frames). This information is preceded by the PRN number of the satellite from which it originated. This message will not be generated unless all 10 words from all 3 frames have passed parity.

An individual message is sent for each PRN being tracked. For a given satellite, the message number increments by one each time a new message is generated.

Ephemeris data whose time of ephemeris is older than six hours will not be shown.

### REPA

Structure:

| $REPA | prn | subframe1 | Subframe2 | subframe3 | *xx | [CR][LF] |
|---|---|---|---|---|---|---|

| Field # | Field type | Data Description | Example |
|---|---|---|---|
| 1 | $REPA | Log header | $REPA |
| 2 | prn | PRN of satellite from which data originated | 14 |
| 3 | subframe1 | Sub-frame 1 of ephemeris data (60 hex characters) | 8B09DC17B9079DD7007D5DE404A9B2D 04CF671C6036612560000021804FD |
| 4 | subframe2 | Sub-frame 2 of ephemeris data (60 hex characters) | 8B09DC17B98A66FF713092F12B359D FF7A0254088E1656A10BE2FF125655 |
| 5 | subframe3 | Sub-frame 3 of ephemeris data (60 hex characters) | 8B09DC17B78F0027192056EAFFDF2724C 9FE159675A8B468FFA8D066F743 |
| 6 | *xx | Checksum | *57 |
| 7 | [CR][LF] | Sentence terminator | [CR][LF] |

Example:

```
$REPA,14,8B09DC17B9079DD7007D5DE404A9B2D04CF671C6036612560000021804FD,
8B09DC17B98A66FF713092F12B359DFF7A0254088E1656A10BE2FF125655,
8B09DC17B78F0027192056EAFFDF2724C9FE159675A8B468FFA8D066F743*57[CR][LF]
```

### REPB

Format: Message ID = 14 Message byte count = 108

| Field # | Data | Bytes | Format | Units | Offset |
|---|---|---|---|---|---|
| 1 (header) | Sync | 3 | char | | 0 |
| | Checksum | 1 | char | | 3 |
| | Message ID | 4 | integer | | 4 |
| | Message byte count | 4 | integer | bytes | 8 |
| 2 | PRN number | 4 | integer | 1-32 | 12 |
| 3-4-5 | Ephemeris data | 90 | char | data [90] | 16 |
| | Filler bytes | 2 | char | | 106 |
| 6 | Reserved. Set to 0. | | | | |

## RGEA/B/D CHANNEL RANGE MEASUREMENTS

RGEA/B contains the channel range measurements for the currently observed satellites i.e. those that are in tracking state 4. These logs contain all of the extended tracking status bits. The RGED message is a compressed form of the RGEB message.

It is important to ensure that the receiver clock has been set and can be monitored by the bits in the rec-status field. Large jumps in range as well as Accumulated Doppler Range (ADR) will occur as the clock is being adjusted. If the ADR measurement is being used in precise phase processing, it is important not to use the ADR if the "parity known" flag in the tr-status field is not set as there may exist a half (1/2) cycle ambiguity on the measurement. The tracking error estimate of the pseudorange and carrier phase (ADR) is the thermal noise of the receiver tracking loops only. It does not account for possible multipath errors or atmospheric delays.

As shown in *Table 4* for the Channel Tracking Status word, bit 19 shows the observation is for L1. This is to aid in parsing the data.

### RGEA

Structure:

```
$RGEA  week  seconds  # obs  rec-status
prn  psr  psr std  adr  adr std  dopp  S/No  locktime  ch tr-status
:
prn  psr  psr std  adr  adr std  dopp  S/No  locktime  ch tr-status
*xx  [CR][LF]
```

| Field # | Field type | Data Description | Example |
|---|---|---|---|
| 1 | $RGEA | Log header | $RGEA |
| 2 | week | GPS week number | 845 |
| 3 | seconds | GPS seconds into the week (receiver time, not corrected for clock error) | 511089.00 |
| 4 | # obs | Number of satellite observations with information to follow | 14 |
| 5 | rec-status | Receiver self-test status (see *Table 11 on Page 69*) | 000B20FF |
| 6 | prn | Satellite PRN number (1-32) of range measurement | 4 |
| 7 | psr | Pseudorange measurement (m) | 23907330.296 |
| 8 | psr std | Pseudorange measurement standard deviation (m) | 0.119 |
| 9 | adr | Carrier phase, in cycles (accumulated Doppler range) | -125633783.992 |
| 10 | adr std | Estimated carrier phase standard deviation (cycles) | 0.010 |
| 11 | dopp | Instantaneous carrier Doppler frequency (Hz) | 3714.037 |
| 12 | $C/N_0$ | Signal to noise density ratio $C/N_0 = 10[\log_{10}(S/N_0)]$ (dB-Hz) | 44.8 |
| 13 | locktime | Number of seconds of continuous tracking (no cycle slipping) | 1928.850 |
| 14 | ch tr-status | Channel tracking status: hexadecimal number indicating phase lock, channel number and channel state as shown in *Table 4, Page 53* | 82E04 |
| 15-23 | | Next observation | |
| ... | | Next observation | |
| variable | *xx | Checksum | *30 |
| variable | [CR][LF] | Sentence terminator | [CR][LF] |

Example (carriage returns have been added between observations for clarity):

```
$RGEA,845,511089.00,14,000B20FF
4,23907330.296,0.119,-125633783.992,0.010,3714.037,44.8,1928.850,82E04,
4,23907329.623,1.648,-97896180.284,0.013,2894.285,35.0,1746.760,582E0B,
2,21298444.942,0.040,-111954153.747,0.006,-1734.838,54.2,17466.670,82E14,
2,21298444.466,0.637,-87236867.557,0.006,-1351.607,43.3,17557.260,582E1B,
9,22048754.383,0.063,-115874135.450,0.006,2174.006,50.4,5489.100,82E24,
9,22048754.424,0.641,-90291443.071,0.006,1694.238,43.2,5489.100,582E2B,
15,23191384.847,0.261,-121887295.980,0.017,-2069.744,38.0,9924.740,82E34,
15,23191384.663,0.596,-94977002.452,0.010,-1612.587,43.8,9881.830,582E3B,
26,24063897.737,0.199,-126477739.189,0.014,-2654.682,40.3,12821.640,82E54,
26,24063898.913,1.043,-98553986.239,0.013,-2068.380,39.0,12793.280,582E5B,
7,20213352.139,0.037,-106237901.461,0.005,439.943,55.0,10313.040,82E74,
7,20213351.196,0.498,-82782498.454,0.007,343.020,45.4,9977.400,582E7B,
27,24393726.829,0.123,-128229016.323,0.012,-4047.338,44.5,22354.119,82E94,
27,24393728.057,1.805,-99918535.513,0.013,-3153.559,34.2,22301.830,582E9B
*30[CR][LF]
```

## *RGEB*

Format: Message ID = 32 Message byte count = 32 + (obs x 44)

| Field # | Data | Bytes | Format | Units | Offset |
|---|---|---|---|---|---|
| 1 (header) | Sync | 3 | char | | 0 |
| | Checksum | 1 | char | | 3 |
| | Message ID | 4 | integer | | 4 |
| | Message byte count | 4 | integer | bytes | 8 |
| 2 | Week number | 4 | integer | weeks | 12 |
| 3 | Seconds of week | 8 | double | s | 16 |
| 4 | Number of observations (obs) | 4 | integer | | 24 |
| 5 | Receiver self-test status (*see Table 11*) | 4 | integer | | 28 |
| 6 | PRN | 4 | integer | | 32 |
| 7 | Pseudorange | 8 | double | m | 36 |
| 8 | StdDev pseudorange | 4 | float | m | 44 |
| 9 | Carrier phase – accumulated Doppler range | 8 | double | cycles | 48 |
| 10 | StdDev – accumulated Doppler range | 4 | float | cycles | 56 |
| 11 | Doppler frequency | 4 | float | Hz | 60 |
| 12 | $C/N_0$ | 4 | float | dB-Hz | 64 |
| 13 | Locktime | 4 | float | s | 68 |
| 14 | Channel Tracking status | 4 | integer | | 72 |
| 15... | Next PRN offset = 32 + (obs# x 44) where obs# = 0 to obs | | | | |

## RGED

Format: Message ID = 65 Message byte count = 24 + (obs x 20)

| Field # | Data | Bytes | Format | Units | Offset |
|---|---|---|---|---|---|
| 1 | Sync | 3 | char | | 0 |
| (header) | Checksum | 1 | char | | 3 |
| | Message ID | 4 | integer | | 4 |
| | Message byte count | 4 | integer | bytes | 8 |
| 2 | Number of observations (obs) | 2 | integer | | 12 |
| 3 | Week number | 2 | integer | weeks | 14 |
| 4 | Seconds of week | 4 | integer | s | 16 |
| 5 | Receiver self test status (*see Table 11)* | 4 | integer | | 20 |
| 6 | First PRN range record | 20 | integer | Only bits 0 – 23 are represented | 24 |
| Next PRN offset = 24 + (20 x obs#) where obs# = 0 to obs | | | | | |

### *Table 11 Receiver Self-Test Status Codes*

N7 N6 | N5 | N4 | N3 | N2 | N1 | N0 |<- <- Nibble Number

| Bit | Description | Range Values | Hex Value |
|---|---|---|---|
| lsb = 0 | ANTENNA | 1=good, 0=bad | 00000001 |
| 1 | L1PLL | 1=good, 0=bad | 00000002 |
| 2 | RAM | 1=good, 0=bad | 00000004 |
| 3 | SOFTWARE INTEGRITY | 1=good, 0=bad | 00000008 |
| 4 | MINOS | 1=good, 0=bad | 00000010 |
| 5 | L1AGC | 1=good, 0=bad | 00000020 |
| 6 | COM 1 | 1=good, 0=bad | 00000040 |
| 7 | COM 2 | 1=good, 0=bad | 00000080 |
| 8 | WEEK | 1=not set, 0=set | 00000100 |
| 9 | COARSETIME | 1=not set, 0=set | 00000200 |
| 10 | FINETIME | 1=not set, 0=set | 00000400 |
| 11 | L1JAMMER | 1=present, 0=normal | 00000800 |
| 12 | BUFFER COM 1 | 1=overrun, 0=normal | 00001000 |
| 13 | BUFFER COM 2 | 1=overrun, 0=normal | 00002000 |
| 14 | BUFFER CONSOLE | 1=overrun, 0=normal | 00004000 |
| 15 | RANGE BUFFER OVERLOAD | 1=overload, 0=normal | 00008000 |
| 16 | Reserved | | |
| 17 | | | |
| 18 | | | |
| 19 | | | |
| 20 | | | |
| 21 | | | |
| 22 | | | |
| 23 | | | |
| 24 | | | |
| 25 | | | |
| 26 | | | |
| 27 | | | |
| 28-31 | | | |

GPSCard examples:

MEDLL: All OK = 0000 0000 0000 0000 0000 0000 1111 1111 (binary) = 000000FF (hexadecimal)

***Notes:***

1. ***Bit 3: Refers to the periodic code check.***
2. ***Bits 2, 4, 6, 7: these are set only once when the GPSCard is first powered up. All other bits are set by internal test processes periodically.***
3. ***Bits 12-15: Flag is reset to 0 five minutes after the last overrun/overload condition has occurred.***
4. ***Bits 17-20: These bits are always set to 0 for MEDLL***

## Receiver Status – Detailed Bit Descriptions of Self-Test Word

What follows is a detailed description of each bit setting in the Receiver Self-Test Status word (see *Table 11 on Page 69*). Additional information is also included in *Table 13, on Page 72*.

The following information is output in the receiver status bits:

***Table 12 Receiver Self-Test Status Bits***

| Bit # | Range Value | Name | Description |
|---|---|---|---|
| 0 | 1 = good<br>0 = bad | Antenna | This is only verified if the receiver is supplying power to the antenna. For all EGNOS receiver models, the antenna will not be powered so this bit is not used. |
| 1 | 1 = good<br>0 = bad | L1 PLL | Verifies that the L1 VCO is locked and sets the bit to good if the test passes. |
| 2 | 1 = good<br>0 = bad | RAM | RAM is tested by the assumption that if the code is running and we get to set the RAM good bit, then the RAM itself must be good. The bit is never set to bad in the software. If the RAM is determined to be bad during the boot sequence, the software is not run and the LED on the front panel will be flashed at 4Hz. No additional error message is output when this test fails. |
| 3 | 1 = good<br>0 = bad | Software Integrity | This bit will indicate that the authorization code has successfully been used to check the integrity of the software loaded on the card. On success, the bit is set to good. On failure, the receiver will be reset in an attempt to alleviate the problem after outputting a fatal !ERRA message of type 1, see *Page 88*. If the problem persists, attempt to reload the card. In the event that reloading the receiver does not fix the problem, the unit will need to be replaced. |
| 4 | 1 = good<br>0 = bad | DSP | This function performs a readback of the MINOS Master Control Register using alternating test patterns for both master and slave MINOS on the board and sets the bit to good if the test passes. A failure indicates that there is a problem with the MINOS and the unit should be replaced. No additional error message is output when this test fails. |
| 5 | 1 = good<br>0 = bad | L1 AGC | Verifies that the L1 AGC is currently not being adjusted and is properly calibrated and sets the bit to good if the test passes. This is only done on the MEDLL Master and OEM3 cards, as there is no AGC on the MEDLL Slave cards.<br>Failure of this test could be the result of various possibilities such as: bad antenna LNA, excessive loss in the antenna cable, faulty RF down-converter, or a pulsating or high power jamming signal causing interference. If this bit is continuously set to clear, you cannot identify an external cause for the failed test. |
| 6 | 1 = good<br>0 = bad | COM1 | The DUART is tested using an internal loopback verifying the transmit and receive shift registers, DTR/DSR, RTS/CTS, and RI/DCD lines and sets the bit to good if the test passes. If the test fails, there will most likely be no output from the receiver since the COM port has failed. In the event that the COM port has failed but there are still logs being output, there may be additional problems with the ASIC (since the DUART is integrated into the MINOS) and the unit should be replaced. |
| 7 | 1 = good<br>0 = bad | COM2 | This test is identical to the COM1 test. |
| 8 | 1 = not set<br>0 = set | WEEK | The week is initialised to not set at start-up. Once the week has been determined (by the decoding of navigation data) the bit is set to 0. Once the week has been determined, it will remain set until a receiver reset occurs.<br>If this bit is not set to zero, then the observation data, pseudorange, carrier phase, and Doppler measurements may jump as the clock adjusts itself. |
| 9 | 1 = not set<br>0 = set | Coarse Time | The coarse time is initialised to not set at start-up. Once the coarse time has been determined (by decoding navigation data) the bit is set to 0. Once coarse time has been determined, it will remain set until a receiver reset occurs.<br>If this bit is not set to zero, then the observation data, pseudorange, carrier phase, and Doppler measurements may jump as the clock adjusts itself. |

| Bit # | Range Value | Name | Description |
|---|---|---|---|
| 10 | 1 = not set<br>0 = set | Fine Time | The fine time is initialised to not set at start-up. Once the fine time has been determined (by calculating a position/time solution) the bit is set to 0. Since CLOCKADJUST is disabled, once the fine time has been determined it will remain set until a receiver reset occurs.<br>If this bit is not set to zero, then the observation data, pseudorange, carrier phase, and Doppler measurements may jump as the clock adjusts itself. |
| 11 | 1 = present<br>0 = normal | L1 Jammer | This test checks to see if the L1 AGC has detected a jammer and sets the bit to indicate its presence. You should monitor this bit, and if set to 1, do your best to remedy the cause of the jamming signal. Nearby transmitters or other electronic equipment could be the cause of interference; you may find it necessary to relocate your antenna position if the problem persists. |
| 12 | 1 = overrun<br>0 = normal | Buffer COM1 | The COM1 UART Tx and Rx status lines are checked and if either has experienced an overrun, this status bit is set. If set, it will remain in that state for 2 seconds beyond the last overrun. Over-run is caused by requesting more log data than can be taken from the RIMS-C receiver because of baud rate limitations or slow communications equipment. If this happens, the new data attempting to be loaded into the buffer will be discarded. The receiver will not load a partial data record into an output buffer. The flag resets to zero, 2 seconds after the last overrun occurred. |
| 13 | 1 = overrun<br>0 = normal | Buffer COM2 | The COM2 UART Tx and Rx status lines are checked and if either has experienced an overrun, this status bit is set. If set, it will remain in that state for 2 seconds beyond the last overrun. Over-run is caused by requesting more log data than can be taken from the RIMS-C receiver because of baud rate limitations or slow communications equipment. If this happens, the new data attempting to be loaded into the buffer will be discarded. The receiver will not load a partial data record into an output buffer. The flag resets to zero, 2 seconds after the last overrun occurred. |
| 14 | 1 = overrun<br>0 = normal | Buffer Console | The console Tx and Rx status lines are checked and if either has experienced an overrun, this status bit is set. If set, it will remain in that state for 2 seconds beyond the last overrun. Over-run is caused by requesting more log data than can be taken from the RIMS-C receiver because of baud rate limitations or slow communications equipment. If this happens, the new data attempting to be loaded into the buffer will be discarded. The receiver will not load a partial data record into an output buffer. The flag resets to zero, 2 seconds after the last overrun occurred. |
| 15 | 1 = overrun<br>0 = normal | Range Buffer Overrun [13] | Verifies that there is space in a circular RAM buffer for range measurements and sets the bit to good if the test passes. When the buffer is overloaded (there are too many successive range measurements for the CPU to service) the bit is set to overrun. The bit will remain set for 2 seconds after the last overrun and then the bit will be cleared.<br>Requesting an excessive amount of information from the receiver may cause this. If this condition is occurring, limit redundant data logging or reconfigure the receiver to output binary data output formats, or both. You should attempt to tune the logging requirements to keep the idle time above 20% for best operation. If the average idle CPU drops below 10% for prolonged periods of time (2-5 seconds), critical errors may result in internal data loss and the over-load bit will be set to 1. You can monitor the range buffer CPU idle time by using the RVSA log message. The flag is reset 2 seconds after the last overload occurred. |
| 16-31 | Reserved | | |

[13] As the amount of buffer space becomes limited, the software will begin to slow down the position calculation rate. If the buffer becomes further limited, the software will begin to skip range measurement processing. Priority processing goes to the tracking loops.

**Table 13 Additional Information About Portable MEDLL Receiver Self-Test Status Word**

The bits have been place into 4 categories. They are:

1. **Not Used in Portable MEDLL Rx:** The user can ignore this bit.
2. **Diagnostics:** This bit does not need to be continually monitored by the user but can provide useful information if a problem is detected.
3. **Checked at Start-up:** This test is done at power-up of the receiver. This should show 'valid' status before continuing.
4. **Continuous Monitoring:** This bit should be continually monitored by the user. It either indicates a problem or a change of status.

| Bit | Description | Category | Comments |
|---|---|---|---|
| 0 | ANTENNA | Not Used In Portable MEDLL Rx | No antenna power provided by the receiver. |
| 1 | L1 PLL | Diagnostics | Can be used for diagnostics if a problem is detected. |
| 2 | RAM | Checked at Start-up | Always 1 on start-up. A 0 will remain unseen; the computer is not working. |
| 3 | SOFTWARE INTEGRITY | Continuous Monitoring | ROM CRC failure. |
| 4 | MINOS | Checked at Start-up | |
| 5 | L1 AGC | Diagnostics | Indicates the AGC has been adjusted. If there is a problem it will show up in the C/No and ranges residuals. |
| 6 | COM 1 | Checked at Start-up | |
| 7 | COM 2 | Checked at Start-up | |
| 8 | WEEK | Continuous Monitoring | |
| 9 | COARSETIME | Continuous Monitoring | |
| 10 | FINETIME | Continuous Monitoring | |
| 11 | L1 JAMMER | Continuous Monitoring | Indicates there may be a jammer present on L1. |
| 12 | BUFFER COM1 | Continuous Monitoring | |
| 13 | BUFFER COM2 | Continuous Monitoring | |
| 14 | BUFFER CONSOLE | Continuous Monitoring | |
| 15 | BUFFER OVERLOAD | Continuous Monitoring | |
| 16-31 | Reserved | | |

### *Table 14 Range Record Format (RGED Only)*

| Data | Bits from first to last | Length (bits) | Format | Scale Factor |
|---|---|---|---|---|
| PRN [14] [15] | 0..5 | 6 | integer | 1 |
| $C/N_0$ [16] | 6..10 | 5 | integer | (20 + n) dB-Hz |
| Lock time [17] | 11..31 | 21 | integer | 1/32 s |
| ADR [18] | 32..63 | 32 | integer 2's compliment | 1/256 cycles |
| Doppler Frequency | 68..95 | 28 | integer 2's compliment | 1/256 Hz |
| Pseudorange | 64..67msn; 96..127 lsw | 36 | integer 2's compliment | 1/128 m |
| StdDev – ADR | 128..131 | 4 | integer | (n + 1) / 512 cycles |
| StdDev – pseudorange | 132..135 | 4 | | (see footnote) [19] |
| Channel tracking status [20] | 136..159 | 24 | integer | see *Table 4, Page 53* |

---

14 Only PRNs 1-63 are reported correctly. (Note: while there are only 32 PRNs in the basic GPS scheme, situations exist which require the use of additional PRNs.)

15 The PRN offsets for WAAS have been mapped to the same range as GPS, i.e. 1-19, while the PRN offsets for GLONASS are 1-29.

16 $C/N_0$ is constrained to a value between 20 – 51 dB-Hz. Thus, if it is reported that $C/N_0$ = 20 dB-Hz, the actual value could be less. Likewise, if it is reported that $C/N_0$ = 51 dB-Hz, the true value could be greater.

17 Lock time rolls over after 2,097,151 seconds.

18 ADR (Accumulated Doppler Range) is calculated as follows:

ADR_ROLLS = (-RGED_PSR / WAVELENGTH – RGED_ADR)/MAX_VALUE
*Round to the closest integer*
IF (ADR_ROLLS ≤ -0.5 )
    ADR_ROLLS = ADR_ROLSS –0.5
ELSE
    ADR_ROLLS = ADR_ROLLS + 0.5
*At this point integerise ADR_ROLLS*
CORRECTED_ADR = RGED_ADR – (MAX_VALUE * ADR_ROLLS)
*where:*
ADR has units of cycles
WAVELENGTH = 0.1902936727984 for L1
MAX_VALUE = 8388608

19

| Code | RGED |
|---|---|
| 0 | 0.00 to 0.050 |
| 1 | 0.051 to 0.075 |
| 2 | 0.076 to 0.113 |
| 3 | 0.114 to 0.169 |
| 4 | 0.170 to 0.253 |
| 5 | 0.254 to 0.380 |
| 6 | 0.381 to 0.570 |
| 7 | 0.571 to 0.854 |
| 8 | 0.855 to 1.281 |
| 9 | 1.282 to 2.375 |
| 10 | 2.376 to 4.750 |
| 11 | 4.751 to 9.500 |
| 12 | 9.501 to 19.000 |
| 13 | 19.001 to 38.000 |
| 14 | 38.001 to 76.000 |
| 15 | 76.001 to 152.000 |

20 Only bits 0-23 are represented in the RGED log.

## RVSA/B RECEIVER STATUS

This log conveys various status parameters of the receiver. If the receiver is a multiple-GPSCard unit with a master card (e.g. MEDLL), certain parameters are repeated for each individual GPSCard. If the receiver is composed of only one GPSCard, then only the parameters for that unit are listed.

Note that the number of satellite channels represents the maximum number of channels reporting information in logs such as ETSA/B and RGEA/B/D.

### RVSA

Structure:

```
$RVSA week seconds sat_chan sig_chan num reserved
idle status
:
idle status
*xx [CR][LF]
```

| Field # | Field type | Data Description | Example |
|---|---|---|---|
| 1 | $RVSA | Log header | $RVSA |
| 2 | week | GPS week number | 847 |
| 3 | seconds | GPS seconds into the week. | 318923.00 |
| 4 | sat_chan | Number of satellite channels | 12 |
| 5 | sig_chan | Number of signal channels | 24 |
| 6 | num | Number of cards | 1 |
| 7 | Reserved | Reserved. Set to 0. | |
| 8 | idle | First GPSCard: CPU idle time. Averaged over a 1 second period (percent). | 22 |
| 9 | status | First GPSCard: Self-test status (see *Table 11 on Page 69*) | 000B00FF |
| 10... | | Next GPSCard: CPU idle time & self-test status | |
| ... | | Next GPSCard: CPU idle time & self-test status | |
| variable | *xx | Checksum | *42 |
| variable | [CR][LF] | Sentence terminator | [CR][LF] |

MEDLL Example:

$RVSA,77,162465.00,16,16,8,0,53.00,42000FF,66.00,42000FF,68.00,42000FF,77.00,42000FF,69.00,42000FF,87.00,42000FF,89.00,42000FF,88.00,42000FF*0B

### RVSB

Format: Message ID = 56 Message byte count = 28 + (8 x number of cards )

| Field # | Data | Bytes | Format | Units | Offset |
|---|---|---|---|---|---|
| 1 (header) | Sync | 3 | char | | 0 |
| | Checksum | 1 | char | | 3 |
| | Message ID | 4 | integer | | 4 |
| | Message byte count | 4 | integer | bytes | 8 |
| 2 | Week number | 4 | integer | weeks | 12 |
| 3 | Seconds of week | 8 | double | s | 16 |
| 4 | Number of satellite channels | 1 | char | | 24 |
| 5 | Number of signal channels | 1 | char | | 25 |
| 6 | Number of cards | 1 | char | | 26 |
| 7 | Reserved. Set to 0. | 1 | byte | | 27 |
| 8 | CPU idle time | 4 | float | percent | 28 |
| 9 | Self-test status | 4 | integer | | 32 |
| 8 & 9 are repeated for each card | Next Card offset = 28 + ( 8 x card# ) where card# = 0 – (number of cards – 1) | | | | |

## SATA/B SATELLITE SPECIFIC DATA

This log provides satellite specific data for satellites actually being tracked. The record length is variable and depends on the number of satellites.

Each satellite being tracked has a reject code indicating whether it is used in the solution, or the reason for its rejection from the solution. The reject value of 0 indicates the observation is being used in the position solution. Values of 1 through 13 indicate the observation has been rejected for the reasons specified in *Table 7 on Page 54*.

### SATA

Structure:

```
$SATA | week | Seconds | sol status | # obs
prn | azimuth | elevation | residual | reject code
:
prn | azimuth | elevation | residual | reject code | *xx | [CR][LF]
```

| Field # | Field type | Data Description | Example |
|---|---|---|---|
| 1 | $SATA | Log header | $SATA |
| 2 | week | GPS week number | 637 |
| 3 | seconds | GPS seconds into the week | 513902.00 |
| 4 | sol status | Solution status as listed in *Table 3, Page 53* | 0 |
| 5 | # obs | Number of satellite observations with information to follow | 7 |
| 6 | prn | Satellite PRN number (1-32) | 18 |
| 7 | azimuth | Satellite azimuth from user position with respect to True North, in degrees | 168.92 |
| 8 | elevation | Satellite elevation from user position with respect to the horizon, in degrees | 5.52 |
| 9 | residual | Satellite range residual from position solution for each satellite, in metres | 9.582 |
| 10 | reject code | Indicates that the range is being used in the solution (code 0) or that it was rejected (code 1-13), as shown in *Table 7 on Page 54* | 0 |
| 11... | .. | Next PRN | |
| variable | *xx | Checksum | *1F |
| variable | [CR][LF] | Sentence terminator | [CR][LF] |

Example:

```
$SATA,637,513902.00,0,7,18,168.92,5.52,9.582,0,6,308.12,55.48,0.737,0,
15,110.36,5.87,16.010,0,11,49.63,40.29,-0.391,0,
2,250.05,58.89,-12.153,0,16,258.55,8.19,-20.237,0,
19,118.10,49.46,-14.803,0*1F[CR][LF]
```

## SATB

Format: Message ID = 12 Message byte count = 32 + (obs*32)

| Field # | Data | Bytes | Format | Units | Offset |
|---|---|---|---|---|---|
| 1 (header) | Sync | 3 | char | | 0 |
| | Checksum | 1 | char | | 3 |
| | Message ID | 4 | integer | | 4 |
| | Message byte count | 4 | integer | bytes | 8 |
| 2 | Week number | 4 | integer | weeks | 12 |
| 3 | Seconds of week | 8 | double | seconds | 16 |
| 4 | Solution status | 4 | integer | | 24 |
| 5 | Number of observations (obs) | 4 | integer | | 28 |
| 6 | PRN | 4 | integer | | 32 |
| 7 | Azimuth | 8 | double | degrees | 36 |
| 8 | Elevation | 8 | double | degrees | 44 |
| 9 | Residual | 8 | double | metres | 52 |
| 10 | Reject code | 4 | integer | | 60 |
| 11... | Next PRN offset = 32 + (obs# * 32) where obs# varies from 0 to (obs-1) | | | | |

## SBTA/B SATELLITE BROADCAST DATA: RAW SYMBOLS

This message contains the satellite broadcast data in raw symbols before FEC decoding or any other processing. An individual message is sent for each PRN being tracked. For a given satellite, the message number increments by one each time a new message is generated. This data matches the RBTA/B data if the message numbers are equal. The data must be logged with the 'onnew' trigger activated to prevent loss of data.

### SBTA

Structure:

`$SBTA` `week` `seconds` `prn` `cstatus` `message #` `# of symbols`

`raw symbols` `*xx` `[CR][LF]`

| Field # | Field type | | Example |
|---|---|---|---|
| 1 | $SBTA | Log header | $SBTA |
| 2 | week | GPS week number | 883 |
| 3 | seconds | GPS seconds into the week | 413908.000 |
| 4 | prn | PRN of satellite from which data originated | 120 |
| 5 | cstatus | Channel Tracking Status | 80812F14 |
| 6 | message # | Message sequence number | 119300 |
| 7 | # of symbols | Number of symbols transmitted in the message. At present, always equals 256 bits. | 256 |
| 8 | raw symbols | 256 symbols compressed into a 128 bytes, i.e. 4 bits/symbol. Hence, 256 hex characters are output. If FEC decoding is enabled, soft symbols are output with values ranging from -3 to 3. Otherwise 1's and 0's are output. | EE33EEEE33333E33EE33EEEE33333E33EE33EEEE3333<br>3E33EE33EEEE33333EEEE3333EEE33E33E3EE33EEE3E<br>EEE33EE3E3EEEEEEEEEEEEEE3333EEE33EEEEE33EE<br>3EEE3E3EE3EE33EEE33E333EE3333E3E3333E33E3333<br>EEEEE333EE3E3333EE3EE3EE33EE3EE3EE3E33E33E3<br>EEE33333E3333E33E3E333E3E33333E3EEE3E3E |
| 9 | *xx | Checksum | *4C |
| 10 | [CR][LF] | Sentence terminator | [CR][LF] |

Example:

```
$SBTA,883,413908.000,120,80812F14,119300,256,EE33EEEE33333E33EE33EEEE333
33E33EE33EEEE33333E33EE33EEEE33333EEEE3333EEE33E33E3EE33EEE3EEEE33EE3E3E
EEEEEEEEEEEE3333EEE33EEEEE33EE3EEE3E3EE3EE33EEE33E333EE3333E3E3333E33E3
333EEEEE333EE3E3333EE3EE3EE33EE3EE3EE3E33E33E3EEE33333E3333E33E3E333E3E3
3333E3EEE3E3E*4C[CR][LF]
```

### SBTB

Format: Message ID = 53 Message byte count = 168

| Field # | Data | Bytes | Format | Units | Offset |
|---|---|---|---|---|---|
| 1 (header) | Sync | 3 | char | | 0 |
| | Checksum | 1 | char | | 3 |
| | Message ID | 4 | integer | | 4 |
| | Message byte count | 4 | integer | bytes | 8 |
| 2 | Week number | 4 | integer | weeks | 12 |
| 3 | Seconds of week | 8 | double | seconds | 16 |
| 4 | PRN number | 4 | integer | 1-999 | 24 |
| 5 | Channel Status | 4 | integer | n/a | 28 |
| 6 | Message # | 4 | integer | n/a | 32 |
| 7 | # of Symbols | 4 | integer | n/a | 36 |
| 8 | Raw Symbols | 128 | char | n/a | 40 |

## TM1A/B TIME OF 1PPS

This log provides the time of the GPSCard 1PPS in GPS week number and seconds into the week. It also includes the receiver clock offset, the standard deviation of the receiver clock offset and clock model status. This log will output at a maximum rate of 1 Hz.

### TM1A

Structure:

| $TM1A | week | seconds | offset | offset std | utc offset | cm status | *xx | [CR][LF] |
|---|---|---|---|---|---|---|---|---|

| Field # | Field type | Data Description | Example |
|---|---|---|---|
| 1 | $TM1A | Log header | $TM1A |
| 2 | week | GPS week number | 794 |
| 3 | seconds | GPS seconds into the week at the epoch coincident with the 1PPS output strobe (receiver time) | 414634.999999966 |
| 4 | offset | Receiver clock offset, in seconds. A positive offset implies that the receiver clock is ahead of GPS Time. To derive GPS time, use the following formula:<br>GPS time = receiver time – (offset) | -0.000000078 |
| 5 | offset std | Standard deviation of receiver clock offset, in seconds | 0.000000021 |
| 6 | utc offset | This field represents the offset of GPS time from UTC time, computed using almanac parameters. To reconstruct UTC time, algebraically subtract this correction from field 3 above (GPS seconds):<br>UTC time = GPS time – (utc offset) | -9.999999998 |
| 7 | cm status | Receiver Clock Model Status where 0 is valid and values from -20 to -1 imply that the model is in the process of stabilization | 0 |
| 8 | *xx | Checksum | *57 |
| 9 | [CR][LF] | Sentence terminator | [CR][LF] |

Example:

```
$TM1A,794,414634.999999966,-0.000000078,0.000000021,-9.999999998,0*57[CR][LF]
```

### TM1B

Format: Message ID = 03 Message byte count = 52

| Field # | Data | Bytes | Format | Units | Offset |
|---|---|---|---|---|---|
| 1 (header) | Sync | 3 | char | | 0 |
| | Checksum | 1 | char | | 3 |
| | Message ID | 4 | integer | | 4 |
| | Message byte count | 4 | integer | bytes | 8 |
| 2 | Week number | 4 | integer | weeks | 12 |
| 3 | Seconds of week | 8 | double | seconds | 16 |
| 4 | Clock offset | 8 | double | seconds | 24 |
| 5 | StdDev clock offset | 8 | double | seconds | 32 |
| 6 | UTC offset | 8 | double | seconds | 40 |
| 7 | Clock model status | 4 | integer | 0 = good, -1 to -20 = bad | 48 |

## UTCA/B DECODED ALMANAC - UTC TIME PARAMETERS

The UTC time parameters (UTCA/B) are provided following the last almanac records when an ALMA/B message has been logged. The UTCA/B message cannot be logged individually or independently of the ALMA/B message.

For more information on Almanac data, refer to the GPS SPS Signal Specification.

### UTCA

Structure:

```
$UTCA  pct  p1ot  data-ref  wk#-utc  wk#-lset  delta-time  lsop  day #-lset
*xx  [CR][LF]
```

| Field # | Field type | | Example |
|---|---|---|---|
| 1 | $UTCA | Log header | $UTCA |
| 2 | pct | Polynomial constant term, seconds | -2.235174179077148E-008 |
| 3 | p1ot | Polynomial 1st order term, seconds/second | -1.243449787580175E-014 |
| 4 | data-ref | UTC data reference time, seconds | 32768 |
| 5 | wk #-utc | Week number of UTC reference, weeks | 745 |
| 6 | wk #-lset | Week number for leap sec effect time, weeks | 755 |
| 7 | delta-time | Delta time due to leap sec, seconds | 9 |
| 8 | lsop | For use when leap sec on past, seconds | 10 |
| 9 | day #-lset | Day number for leap sec effect time, days | 5 |
| 10 | *xx | Checksum | *37 |
| 11 | [CR][LF] | Sentence terminator | [CR][LF] |

Example:

```
$UTCA,-2.235174179077148E-008,-1.243449787580175E-014,32768,745,755,9,10,5*37
[CR][LF]
```

### UTCB

Format: Message ID = 17 Message byte count = 52

| Field # | Field Type | Bytes | | Units | |
|---|---|---|---|---|---|
| 1 (header) | Sync | 3 | char | | 0 |
| | Checksum | 1 | char | | 3 |
| | Message ID | 4 | integer | | 4 |
| | Message byte count | 4 | integer | bytes | 8 |
| 2 | Polynomial constant term | 8 | double | seconds | 12 |
| 3 | Polynomial 1st order term | 8 | double | seconds/second | 20 |
| 4 | UTC data reference time | 4 | integer | seconds | 28 |
| 5 | Week number UTC reference | 4 | integer | weeks | 32 |
| 6 | Week number for leap sec effect time | 4 | integer | weeks | 36 |
| 7 | Delta time due to leap sec | 4 | integer | seconds | 40 |
| 8 | For use when leap sec on past | 4 | integer | seconds | 44 |
| 9 | Day number for leap sec effect time | 4 | integer | days | 48 |

## WBCA/B WIDE BAND CARRIER RANGE CORRECTION

This message contains the wide band carrier range correction data. A correction is generated for each PRN being tracked and these are grouped together into a single log. Internally, the correction for each satellite is updated asynchronously at a 1 Hz rate. Therefore, logging this message at a rate higher than 1 Hz will result in duplicate data being output. Each carrier range correction is statistically independent and is derived from the previous 1 second of data.

### WBCA

Structure:

```
$WBCA week seconds # obs
prn ch tr-status tr-BW wide band carrier range correction
:
prn ch tr-status tr-BW wide band carrier range correction *xx [CR][LF]
```

| Field # | Field type | Data Description | Example |
|---|---|---|---|
| 1 | $WRCA | Log header | $WBCA |
| 2 | week | GPS week number | 637 |
| 3 | seconds | GPS seconds into the week | 513902.00 |
| 4 | # obs | Number of satellite observations with information to follow: | 7 |
| 5 | prn | Satellite PRN number | 18 |
| 6 | ch tr-status | Channel Tracking Status: Hexadecimal number indicating phase lock, channel number and channel state as shown in *Table 4* | E04 |
| 7 | tr-BW | PLL tracking loop bandwidth in Hz | 15.000 |
| 8 | wide band carrier correction | Wide band carrier range correction in cycles | 0.00123 |
| 9... | .. | Next observation | |
| Variable | *xx | Checksum | *1F |
| Variable | [CR][LF] | Sentence terminator | [CR][LF] |

### WBCB

Format: Message ID = 97 Message byte count = 28 + (obs*16)

| Field # | Data | Bytes | Format | Units | Offset |
|---|---|---|---|---|---|
| 1 (header) | Sync | 3 | char | | 0 |
| | Checksum | 1 | char | | 3 |
| | Message ID | 4 | integer | | 4 |
| | Message byte count | 4 | integer | bytes | 8 |
| 2 | Week number | 4 | integer | weeks | 12 |
| 3 | Seconds of week | 8 | double | seconds | 16 |
| 4 | Number of observations (obs) | 4 | integer | | 24 |
| 5 | PRN | 4 | integer | | 28 |
| 6 | Channel Tracking Status, see *Table 4, on Page 53* | 4 | - | - | 32 |
| 7 | PLL tracking loop bandwidth | 4 | float | Hz | 36 |
| 8 | Wide Band Range Carrier Correction | 4 | float | cycles | 40 |
| 9... | Next observation offset = 28 + (obs#*16) where obs# = 0 to (obs – 1) | | | | |

## WRCA/B WIDE BAND CODE RANGE CORRECTION

This message contains the wide band code range correction data. A correction is generated for each PRN being tracked and these are grouped together into a single log. Internally, the correction for each satellite is updated asynchronously at a

1 Hz rate. Therefore, logging this message at a rate higher than 1 Hz will result in duplicate data being output. Each code range correction is statistically independent and is derived from the previous 1 second of data.

## WRCA

Structure:

```
$WRCA  week  seconds  # obs
prn  ch tr-status  tr-BW  wide band code range correction
:
prn  ch tr-status  tr-BW  wide band code range correction  *xx  [CR][LF]
```

| Field # | Field type | Data Description | Example |
|---|---|---|---|
| 1 | $WRCA | Log header | $WRCA |
| 2 | week | GPS week number | 637 |
| 3 | seconds | GPS seconds into the week | 513902.00 |
| 4 | # obs | Number of satellite observations with information to follow: | 7 |
| 5 | prn | Satellite PRN number | 18 |
| 6 | ch tr-status | Channel Tracking Status: Hexadecimal number indicating phase lock, channel number and channel state as shown in *Table 4* | E04 |
| 7 | tr-BW | DLL tracking loop bandwidth in Hz | 0.050 |
| 8 | wide band range correction | Wide band code range correction in metres | 1.323 |
| 9... | .. | Next PRN | |
| Variable | *xx | Checksum | *1F |
| Variable | [CR][LF] | Sentence terminator | [CR][LF] |

```
Example:
$WRCA, 637,513902.00,7,18,E04,0.050,1.323,
…
*1F[CR][LF]
```

## WRCB

Format: Message ID = 67 Message byte count = 28 + (obs*16)

| Field # | Data | Bytes | Format | Units | Offset |
|---|---|---|---|---|---|
| 1 (header) | Sync | 3 | char | | 0 |
| | Checksum | 1 | char | | 3 |
| | Message ID | 4 | integer | | 4 |
| | Message byte count | 4 | integer | bytes | 8 |
| 2 | Week number | 4 | integer | weeks | 12 |
| 3 | Seconds of week | 8 | double | seconds | 16 |
| 4 | Number of observations (obs) | 4 | integer | | 24 |
| 5 | PRN | 4 | integer | | 28 |
| 6 | Channel Tracking Status | 4 | - | - | 32 |
| 7 | DLL tracking loop bandwidth | 4 | float | Hz | 36 |
| 8 | Wide Band Code Range Correction | 4 | float | metres | 40 |
| 9... | Next PRN offset = 28 + (obs*16) | | | | |

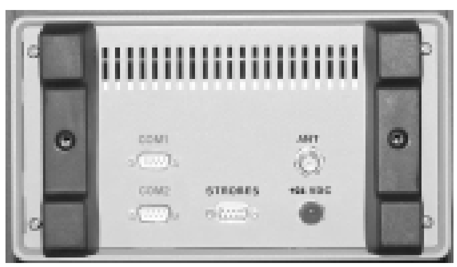
# A PORTABLE MEDLL RECEIVER - TECHNICAL SPECIFICATIONS

| PHYSICAL | |
|---|---|
| Size | 448.8 x 361 x 183.5 mm (without the 19” mounting brackets) |
| Weight | 10.2 kg |

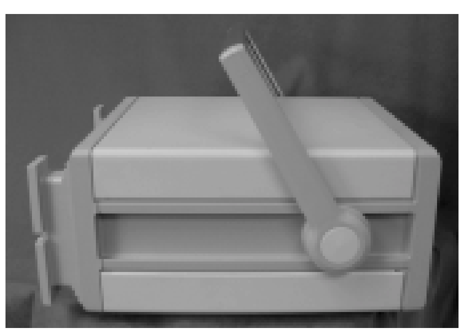
**PORTABLE MEDLL RECEIVER ILLUSTRATIONS**

***Front***


***Back***




***Side***



| | | |
|---|---|---|
| **ENVIRONMENTAL** | | |
| Operating Temperature | | -25° C to +65° C |
| Storage Temperature | | -40° C to +85° C |
| Humidity | | 10-80% |
| Altitude | | 3,000 metres<br>[May operate above 3,000 m in a controlled environment, however is not certified as such.] |
| **POWER INPUT** | | |
| Connector | | 24V power jack |
| Voltage | | 22-30 VDC |
| Current | | maximum 1.5 A continuous; peak 3 A |
| **Portable MEDLL RECEIVER PERFORMANCE** (Subject To GPS System Characteristics) | | |
| Frequency | | L1(1575.42 MHz) |
| Code tracked | | GPS L1-C/A Code, GEO L1-C/A Code, GPS SVN (PRN 1-32), and GEO SVN (PRN 120-138) |
| Satellite Tracking Channels | MEDLL | 12 L1-C/A<br>or<br>10 L1-C/A, 2 L1-C/A GEO |
| Position Accuracy Stand-alone | | 5 metres CEP (SA off), GDOP < 2 |
| Time Accuracy (relative) | | 50 nanoseconds (SA off) |
| Pseudorange Measurement | MEDLL (L1-C/A) | 10 cm RMS, C/N$_0$ > 44 dBHz, BW = 0.05 |
| Single Channel Phase Accuracy | L1 | 3 mm RMS, C/No > 44 dBHz, Loop BW = 3Hz |
| Raw Data Availability Rate | MEDLL | 5 phase and code measurements per second (200 ms) |
| | Almanac data | < 15 minutes after reset |
| Time to First Fix | | 100 seconds (95%) with stabilized internal oscillator.<br>15 minutes maximum from start of cold receiver. No initial time, almanac or position required. |
| Re-acquisition | GPS | 5 seconds C/No = 44 dB-Hz, 10 seconds C/No = 38 dB-Hz |
| | GEO | 10 seconds C/No = 44 dB-Hz |
| Height Measurements | | Up to 18,288 metres (60,000 feet) maximum<br>[In accordance with export licensing, the card is restricted to less than 60,000 feet.] |
| **INPUT/OUTPUT DATA INTERFACE** | | |
| Serial | | Baud rates: 300, 1200, 4800, 9600, 19200, 57600, 115200 bps, user selectable<br>Default: 9600 bps (COM1 and COM2) |
| Connector | | DE9P |
| Electrical format | | RS-232 |
| **OUTPUT STROBES** | | |
| VARF | | A programmable variable frequency output ranging from 0.15 Hz to 5 MHz, with pulse width from 100 ns to 6.55 ms. This is a normally high, active low pulse. There may be as much as 50 ns jitter on this signal. |
| 1PPS | | A one-pulse-per-second Time Sync output. This is a normally high, active low pulse (200 µs for MEDLL) where the falling edge is the reference. |
| MSR | | Measure Output is a 1 or 5 pulses-per-second output, normally high, active low where the pulse width is 200 µs for MEDLL. The falling edge is the receiver's measurement strobe. (Rate is model-dependent.) |

| | | | |
|---|---|---|---|
| STATUS | | Indicates a valid GPS position solution is available. A high level indicates a valid solution or that the FIX POSITION command has been set, see *Page 26.* | |
| Connector | | DE9S | |
| The electrical specifications of the strobe signals are as follows: | | | |
| Output | Voltage | (High) > 2.0 VDC | (Low) < 0.55 VDC |
| | Minimum load impedance | 1 KΩ | |
| **ANTENNA INPUT** | | | |
| Connector | | TNC female | |
| Frequency | | L1(1575.42 MHz), | |
| Power | | 2.1 mm plug (10 – 36 VDC) centre positive | |
| Noise Power Spectral Density | | -140 dBm/Hz to -160 dBm/Hz | |
| Maximum $C/N_0$ | | 65 dB-Hz | |

**DEFAULT CHANNEL ASSIGNMENTS – Portable MEDLL RECEIVER**

| Port | Channel | SV Type | Code | DLL Type | Frame | Nav Type | Symbol Rate | FEC | Sky Search |
|---|---|---|---|---|---|---|---|---|---|
| COM1 | 0 | GPS | L1 C/A | MEDLL | GPS | GPS | 50 | No | Automatic |
| | 1 | GPS | L1 C/A | MEDLL | GPS | GPS | 50 | No | Automatic |
| | 2 | GPS | L1 C/A | MEDLL | GPS | GPS | 50 | No | Automatic |
| | 3 | GPS | L1 C/A | MEDLL | GPS | GPS | 50 | No | Automatic |
| | 4 | GPS | L1 C/A | MEDLL | GPS | GPS | 50 | No | Automatic |
| | 5 | GPS | L1 C/A | MEDLL | GPS | GPS | 50 | No | Automatic |
| | 6 | GPS | L1 C/A | MEDLL | GPS | GPS | 50 | No | Automatic |
| | 7 | GPS | L1 C/A | MEDLL | GPS | GPS | 50 | No | Automatic |
| | 8 | GPS | L1 C/A | MEDLL | GPS | GPS | 50 | No | Automatic |
| | 9 | GPS | L1 C/A | MEDLL | GPS | GPS | 50 | No | Automatic |
| | 10 | GPS | L1 C/A | MEDLL | GPS | GPS | 50 | No | Automatic |
| | 11 | GPS | L1 C/A | MEDLL | GPS | GPS | 50 | No | Automatic |

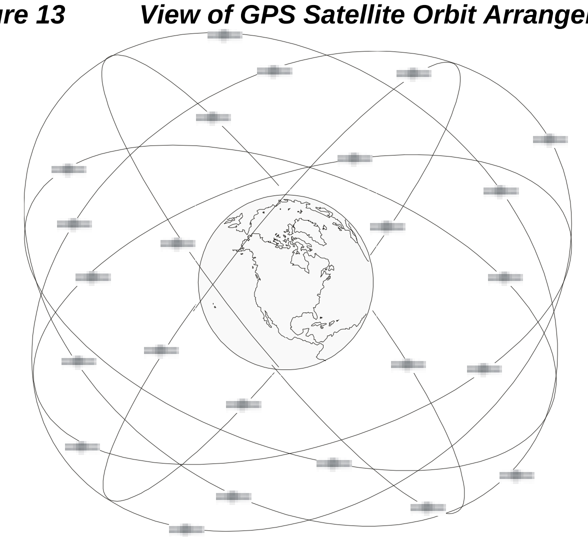
**Or**

| Port | Channel | SV Type | Code | DLL Type | Frame | Nav Type | Symbol Rate | FEC | Sky Search |
|---|---|---|---|---|---|---|---|---|---|
| COM1 | 0 | GPS | L1 C/A | MEDLL | GPS | GPS | 50 | No | Automatic |
| | 1 | GPS | L1 C/A | MEDLL | GPS | GPS | 50 | No | Automatic |
| | 2 | GPS | L1 C/A | MEDLL | GPS | GPS | 50 | No | Automatic |
| | 3 | GPS | L1 C/A | MEDLL | GPS | GPS | 50 | No | Automatic |
| | 4 | GPS | L1 C/A | MEDLL | GPS | GPS | 50 | No | Automatic |
| | 5 | GPS | L1 C/A | MEDLL | GPS | GPS | 50 | No | Automatic |
| | 6 | GPS | L1 C/A | MEDLL | GPS | GPS | 50 | No | Automatic |
| | 7 | GPS | L1 C/A | MEDLL | GPS | GPS | 50 | No | Automatic |
| | 8 | GPS | L1 C/A | MEDLL | GPS | GPS | 50 | No | Automatic |
| | 9 | GPS | L1 C/A | MEDLL | GPS | GPS | 50 | No | Automatic |
| | 10 | GEO | L1 C/A | MEDLL | GEO | GEO | 500 | Yes | Idle |
| | 11 | GEO | L1 C/A | MEDLL | GEO | GEO | 500 | Yes | Idle |

# B GPS OVERVIEW

The Global Positioning System (GPS) is a satellite system capable of providing independent and highly accurate timing and positioning information. GPS provides 24-hour, all-weather, worldwide coverage. Refer to *Figure 13* for a representation of the GPS satellite orbit arrangement.

***Figure 13 View of GPS Satellite Orbit Arrangement***



## GPS SYSTEM DESIGN

The system uses the NAVSTAR (NAVigation Satellite Timing And Ranging) satellites that consist of 24 operational satellites. A minimum of four satellites in view allows the GPSCard to compute its current latitude, longitude, altitude with reference to ellipsoid mean sea level and the GPS system time.

The GPS system design consists of three parts:

- The Space segment
- The Control segment
- The User segment

All these parts operate together to provide accurate three-dimensional positioning, timing and velocity data to users worldwide.

## THE SPACE SEGMENT

The space segment is composed of the NAVSTAR GPS satellites. The specified constellation of the system consists of 24 satellites in six orbital planes, inclined 55° from the equator, with four satellites in each plane. The orbital period of each satellite is approximately 12 hours at an altitude of 20,183 km.

The GPS satellite signal identifies the satellite and provides the positioning, timing, ranging data, satellite status and ephemerides (orbit parameters) of the satellite to the receiver. The satellites can be identified either by the Space Vehicle Number (SVN) or the Pseudorandom Code Number (PRN). The PRN is used by the NovAtel GPSCard.

The GPS satellites transmit on two L-band frequencies; one centred at 1575.42 MHz (L1) and the other at 1227.60 MHz (L2). The L1 carrier is modulated by the C/A code (Coarse/Acquisition) and the P-Code (Precision) that is encrypted for military and other authorized users. The L2 carrier is modulated only with the P-Code. Please refer to *Figure 13, View of GPS Satellite Orbit Arrangement*, on the previous page, for a representation of the GPS satellite orbit arrangement.

## THE CONTROL SEGMENT

The control segment consists of a master control station, five reference stations and three data up-loading stations in locations all around the globe.

The reference stations track and monitor the satellites via their broadcast signals. The broadcast signals contain the ephemeris data of the satellites, the ranging signals, the clock data and the almanac data. These signals are passed to the master control station where the ephemerides are re-computed. The resulting ephemerides corrections and timing corrections are transmitted back to the satellites via the data up-loading stations.

## THE USER SEGMENT

The user segment such as the Portable MEDLL receiver, consists of equipment, which tracks and receives the satellite signals. The user equipment must be capable of simultaneously processing the signals from a minimum of four satellites to obtain accurate position, velocity and timing measurements.

# C INFORMATION MESSAGES

While operating the Portable MEDLL receiver, there may be additional messages output by the receiver that you may observe. This information is in addition to typical log data as described in this manual, and falls into two categories:

Type 1: Messages that occur as a result of an operational error within the receiver over which you have no control. These appear in typical NovAtel log format.

Type 2: Messages that occur in response to your input. These are not in typical log format.

The following sections describe these messages.

## TYPE 1 INFORMATION MESSAGES

As mentioned, Type 1 information messages indicate that there is a problem with the operation of the on-board firmware. To date, the only Type 1 messages in use are the !ERRA and the !MSGA logs.

If you receive the !ERRA message, it may be useful to reload the software using the correct authorization code; see *Table 15, Page 88* for a list of !ERRA message types. Each of these messages causes a "severity fatal" condition to be set, causing the card to be reset. To reload the software, power down the receiver and follow the procedure outlined in *Chapter 4*. Under certain error conditions, the card will reset itself and resolve the difficulty.

If, after verifying that the software has been correctly loaded, and the receiver cannot reset itself to function normally, the receiver requires maintenance.

The !MSGA log would be output if the software is a time-limited version. The log will provide the expiry date of the software. See *Table 16, on Page 89*, for a list of !MSGA message types. Portable MEDLL receivers are not time limited. There will be no expiry date.

### !ERRA

`!ERRA` `type` `severity` `error string` `opt desc` `*xx` `[CR][LF]`

| Field # | Field type | Data Description | Example |
|---|---|---|---|
| 1 | !ERRA | Log header | !ERRA |
| 2 | type | Log type, numbered 0 - 8 (see *Table 15, Page 88*) | 1 |
| 3 | severity | Only one is defined to date: severity_fatal (number = 0); causes the receiver to be reset | 0 |
| 4 | error string | Error message (see *Table 15, Page 88*) | Authorization code invalid |
| 5 | *opt desc* | Optional additional description | |
| 6 | *xx | Checksum | *22 |
| 7 | [CR][LF] | Sentence terminator | [CR][LF] |

Example:

```
!ERRA,1,0,Authorization Code Invalid,*22[CR][LF]
```

**Table 15 Type 1 !ERRA Messages**

| Log type | Error String |
|---|---|
| 0 | Unknown ERRA Type |
| 1 | Authorization Code Invalid |
| 2 | No Authorization Code Found |
| 3 | Invalid Expiry In Authorization Code |
| 4 | Unable To Read ESN |

| | |
|---|---|
| 6 | Card Has Stopped Unexpectedly |
| 7 | Incorrect Number of Cards Found |
| 8 | Software Version Mismatch |

## !MSGA

`!MSGA` `type` `message` `opt desc` `*xx` `[CR][LF]`

| Field # | Field type | | Example |
|---|---|---|---|
| 1 | !MSGA | Log header | !MSGA |
| 2 | type | Log type, numbered from 1000 (see *Table 16*) | 1001 |
| 3 | message | Message (see *Table 16*) | Authorization Code is Time Limited |
| 4 | opt. description | Optional description | Model 3951R expires on 960901 |
| 5 | *xx | Checksum | *6C |
| 6 | [CR][LF] | Sentence terminator | [CR][LF] |

Example:

```
!MSGA,1001,Authorization Code Is Time Limited, Model 3951R Expires on 960901
*6C[CR][LF]
```

***Table 16 Type 1 !MSGA Messages***

| Log type | Message String |
|---|---|
| 1000 | Unknown MSGA Type |
| 1001 | Authorization Code Is Time Limited |

# TYPE 2 INFORMATION MESSAGES

***NOTE:*** Commands referred to in the information messages that do not appear in this manual may be found in a *NovAtel Command Descriptions* manual.

The following is a list of information messages that are generated by the Command Interpreter in the receiver in response to your input. This list is not necessarily complete, but it is the most accurate one available at the time of publication.

***Table 17 Type 2 Information Messages***

| Error Message | Meaning |
|---|---|
| All OK | No errors to report. |
| Argument Must Be Hexadecimal (0-9,A-F) Pairs | An argument which is not hexadecimal was entered. |
| Argument Must Be Numeric | An argument which is not numeric was entered. |
| Authorization Changes Not Available On This Card | An attempt has been made to change the Authorization Code on a card, which is not an OEM card. |
| Authorization Code Entered Incorrectly | The checksum is incorrect for the Authorization Code. The Authorization Code was most likely entered incorrectly. |
| Authorization Code Is Invalid | The existing Authorization Code is invalid. Please contact NovAtel for a new Authorization Code. |
| Can't Change Authorization Code | The existing Authorization Code cannot be changed. Please contact NovAtel for assistance. |
| Clock Model not set TM1A rejected | The clock model status in a $TM1A command is invalid. The $TM1A command is rejected when the clock model has not been set. |
| CLOCK_ADJUST Command Not Available On This Model | The CLOCKADJUST command is not available on this model. |
| Complete Almanac not received yet - try again later | The almanac cannot be saved because a complete almanac has not yet been received. A SAVEALMA command should be performed at a later time when a complete almanac has been received. |
| Data Too Large To Save To NVM | The configuration data being saved is too large. |
| Differential Corrections Not Available On This Model | This model does not have the ability to send or receive differential corrections. |
| EXTERNALCLOCK Command Not Available On This Model | The EXTERNALCLOCK command is not available on this model. |
| FREQUENCY_OUT Command Not Available On This Model | The FREQUENCY_OUT command is not available on this model. |
| FROM port name too LONG | The FROM port name in a SETNAV command is too long. |
| Invalid $ALMA Checksum | The checksum of a $ALMA command is invalid. |
| Invalid $IONA Checksum | The checksum of a $IONA command is invalid. |
| Invalid $REPA Checksum | The checksum of a $REPA command is invalid. |
| Invalid $TM1A Checksum | The checksum of a $TM1A command is invalid. |
| Invalid $UTCA Checksum | The checksum of a $UTCA command is invalid. |
| Invalid ADJUSTCLOCK Option | An invalid CLOCKADJUST switch has been entered. |
| Invalid Baud Rate | The baud rate in a COM1 command is invalid. |
| Invalid Code Smoothing Constant | The code smoothing constant of the CSMOOTH command is invalid. |
| Invalid Channel Number | An invalid channel number has been entered in a command such as ASSIGN. |
| Invalid Command CRC | The received command has an invalid checksum. |
| Invalid Command Name | An invalid command name has been received. |
| Invalid Command Option | One or more arguments of a command are invalid. |
| Invalid Datatype | The data type in an ACCEPT command is invalid. |
| Invalid Datum Offset | The datum offset in a USERDATUM command is invalid. |
| Invalid DATUM Option | An option in a DATUM command is invalid. |
| Invalid Datum Rotation | The datum rotation angle in a USERDATUM command is invalid. |
| Invalid Degree Field | An invalid degree field has been entered in a command such as FIX POSITION. |
| Invalid Doppler | An invalid Doppler has been entered in an ASSIGN command. |

| | |
|---|---|
| Invalid Doppler Window | An invalid Doppler window has been entered in an ASSIGN command. |
| Invalid DYNAMICS Option | The option in a DYNAMICS command is invalid. |
| Invalid Echo Option | The echo option in a COM1 command is invalid. |
| Invalid Elevation Cutoff Angle | The elevation cut-off angle in an ECUTOFF command is invalid. |
| Invalid EXTERNALCLOCK Option | An invalid external clock was entered in the EXTERNALCLOCK command. |
| Invalid EXTERNALCLOCK USER Argument(s) | An invalid argument was entered in the EXTERNALCLOCK command. |
| Invalid FIX Option | An option other than height, position or velocity was specified in a FIX command. |
| Invalid Flattening | The flattening in a USERDATUM command is invalid. |
| Invalid Handshake Option | The handshake option in a COM1 command is invalid. |
| Invalid HEALTH Override | An invalid health has been entered in a SETHEALTH or FIX command. |
| Invalid Height | The height in a FIX HEIGHT command is invalid. |
| Invalid Logger Datatype | An invalid log has been specified in a LOG/UNLOG command. |
| Invalid Logger Offset | An invalid offset has been specified in a LOG command. |
| Invalid Logger Period | An invalid period has been specified in a LOG command. |
| Invalid Logger Port Option | An invalid port number has been specified in a LOG/UNLOG command. |
| Invalid Logger Trigger | An invalid trigger has been specified in a LOG command. |
| Invalid Number of $ALMA Arguments | The number of arguments in a $ALMA command is invalid. |
| Invalid Number of $IONA Arguments | The number of arguments in a $IONA command is invalid. |
| Invalid Number of $REPA Arguments | The number of arguments in a $REPA command is invalid. |
| Invalid Number of $TM1A Arguments | The number of arguments in a $TM1A command is invalid. |
| Invalid Number of $UTCA Arguments | The number of arguments in a $UTCA command is invalid. |
| Invalid Number of Arguments | A command has been received which has an invalid number of arguments. |
| Invalid Number of Databits | The number of data bits in a COM1 command is invalid. |
| Invalid Number of StopBits | The number of stop bits in a COM1 command is invalid. |
| Invalid Parity Option | The parity in a COM1 command is invalid. |
| Invalid Port | The port in a SEND command is invalid. |
| Invalid Port number | The port number in an ACCEPT command is invalid. |
| Invalid RTCA station Name (\XXXX\) | The RTCA station name in a FIX POSITION message is invalid. |
| Invalid Satellite Number | An invalid satellite number has been entered in an ASSIGN command. |
| Invalid Scaling | The scale value in a USERDATUM command is invalid. |
| Invalid Seconds Into Week in TM1A | The time in a $TM1A command is invalid. |
| Invalid SemiMajor Axis | The semi-major axis in a USERDATUM command is invalid. |
| Invalid Symbol Period 1,2,4,5,10,20 | The symbol period is invalid for an ASSIGN on a pseudolite channel. |
| Invalid Token | This error should never occur. If it does, please contact NovAtel. |
| Invalid Week Number in TM1A | The week in a $TM1A command is invalid. |
| MET Command Not Available On This Model | The MET command is not available on this model. |
| Model Invalid | The Authorization Code has an invalid Model. Please contact NovAtel for help. |
| RT20 Logs Not Available On This Model | This model is not able to send or receive RT20 differential corrections. |
| RTCM9 Logs Not Available On This Model | This model does not have the ability to send or receive RTCM9 logs. |
| SETCLOCK disabled TM1A rejected | The $TM1A command is rejected because the user has not enabled clock synchronization using the SETCLOCK command. |
| User Defined DATUM Not Set | This error should not occur. By default the user defined DATUM is set to WGS-84. If you get this error message, please contact NovAtel GPS . |
| Valid Option but Missing Process | This message indicates an error in the software. A command option is valid but software cannot process it |

Example:

```
Argument Must Be Numeric
```

# D ACRONYMS

| | |
|---|---|
| 1PPS | One Pulse Per Second |
| A | Ampere(s) |
| A/D | Analog-to-Digital |
| ADR | Accumulated Doppler Range |
| AGC | Automatic Gain Control |
| ANT | Antenna |
| ASCII | American Standard Code for Information Interchange |
| ASIC | Application Specific Integrated Circuits |
| BIST | Built-In-Self-Test |
| BIT | Built-In Test |
| bps | Bits per Second |
| C/A Code | Coarse/Acquisition Code |
| CEP | Circular Error Probable |
| CPU | Central Processing Unit |
| CR | Carriage Return |
| CRC | Cyclic Redundancy Check |
| CTS | Clear To Send |
| dB | Decibel |
| DC | Direct Current |
| DCD | Data Carrier Detected |
| DGPS | Differential Global Positioning System |
| DLL | Delay Lock Loop |
| DOP | Dilution Of Precision |
| DSP | Digital Signal Processor |
| DSR | Data Set Ready |
| DTR | Data Terminal Ready |
| EGNOS | European Geostationary Navigation Overlay Service |
| ESD | Electrostatic Discharge |
| FEC | Forward Error Correction |
| GDOP | Geometric Dilution Of Precision |
| GEO | Geostationary Satellite |
| GMT | Greenwich Mean Time |
| GND | Ground |
| GPS | Global Positioning System |
| hex | Hexadecimal |
| Hz | Hertz |
| ICD | Interface Control Document |
| I/O | Input/Output |
| IODE | Issue of Data (Ephemeris) |
| L1 | 1575.42 MHz GPS carrier frequency, C/A and P-Code |
| L2 | 1227.60 MHz GPS carrier frequency, P-Code |
| LED | Light-Emitting Diode |

| | |
|---|---|
| LF | Line Feed |
| LNA | Low Noise Amplifier |
| MAT | Multipath Assessment Tool |
| MCC | Master Control Centres |
| MET | Multipath Elimination Technology |
| MEDLL | Multipath Estimating Delay Lock Loop |
| MHz | Megahertz or one million Hertz |
| MINOS | Multiple Independent Nomadic Stargazer |
| MKI | Mark In |
| MOPS | Minimum Operational Performance Standards |
| MSL | Mean sea level |
| MSR | Measurement |
| NAVSTAR | NAVigation Satellite Timing And Ranging (synonymous with GPS) |
| NC | Not Connected |
| NLES | Navigation Land Earth Stations |
| ns | nanosecond |
| OCXO | Oven Controlled Crystal Oscillator |
| OEM | Original Equipment Manufacturer |
| PC | Personal Computer |
| P-Code | Precise Code |
| PLL | Phase Lock Loop |
| PPS | Pulse Per Second |
| PRN | Pseudo Random Noise number |
| RAM | Random Access Memory |
| RIMS | Ranging and Integrity Monitoring Stations |
| RF | Radio Frequency |
| ROM | Read Only Memory |
| RT | Receive Time |
| RTCA | Radio Technical Commission for Aviation Services |
| RTCM | Radio Technical Commission for Maritime Services |
| RTS | Request To Send |
| RXD | Received Data |
| SA | Selective Availability |
| SPS | Standard Positioning Service |
| SV | Space Vehicle |
| SVN | Space Vehicle Number |
| TCXO | Temperature Controlled Crystal Oscillator |
| TT | Transmit Time |
| TXD | Transmitted Data |
| UART | Universal Asynchronous Receiver Transmitter |
| UTC | Universal Time Coordinated |
| V | Volt(s) |
| WGS | World Geodetic System |

# INDEX

**NovAtel Inc.**
**1120 - 68 Avenue N.E.**
**Calgary, Alberta, Canada T2E 8S5**

**GPS Hotline: 1-403-295-4900**
**GPS Fax: 1-403-295-4901**
**E-mail:** mailto:support@novatel.ca
**Web site:** http://www.novatel.ca/

Recyclable

Printed in Canada
on recycled paper