RSR Transcoder 2.0

User Manual

R003

Notice

Every effort was made to ensure that the information in this manual was accurate at the time of printing. However, information is subject to change without notice, and VIAVI reserves the right to provide an addendum to this manual with information not available at the time that this manual was created.

Copyright/Trademarks

Copyright Release

Reproduction and distribution of this guide is authorized for US Government purposes only.

DFARS Statement

If software is for use in the performance of a U.S. Government prime contract or subcontract, software is delivered and licensed as “Commercial Computer Software” as defined in DFAR 252.227-7014 (Feb 2014), or as a “Commercial Item” as defined in FAR 2.101(a) or as “Restricted Computer Software” as defined in FAR 52.227-19 (Dec 2007) or any equivalent agency regulation or contract clause. Use, duplication or disclosure of Software is subject to VIAVI Solutions' standard commercial license terms, and non-DOD Departments and Agencies of the U.S. Government will receive no greater than Restricted Rights as defined in FAR 52.227-19(c)(1-2) (Dec 2007). U.S. Government users will receive no greater than Limited Rights as defined in FAR 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Patents

Patented as described at www.viavisolutions.com/patents.

Warranty Information

Warranty information for this product is available on the VIAVI website at https://www.viavisolutions.com/en-us/warranty-information.

Terms and conditions

Specifications, terms, and conditions are subject to change without notice. The provision of hardware, services, and/or software are subject to VIAVI’s standard terms and conditions, available at www.viavisolutions.com/en/terms-and-conditions.

Open Source Disclaimer - IMPORTANT READ CAREFULLY

The RSR Transcoder 20 includes third party software licensed under the terms of separate open source software licenses. By using this software you agree to comply with the terms and conditions of the applicable open source software licenses. Software originated by VIAVI is not subject to third party licenses. Terms of the VIAVI Software License different from applicable third party licenses are offered by VIAVI alone.

Federal Communications Commission (FCC) Notice

This product was tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This product generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this product in a residential area is likely to cause harmful interference, in which case you will be required to correct the interference at your own expense.

The authority to operate this product is conditioned by the requirements that no modifications be made to the equipment unless the changes or modifications are expressly approved by VIAVI.

EMC Directive Compliance

This product was tested and conforms to the EMC Directive, 2014/30/EU for electromagnetic compatibility.

Declaration of Conformity

VIAVI recommends keeping a copy of the Declaration of Conformity that shipped with the unit with the device at all times.

Low Voltage Directive Compliance

This product was tested and conforms to the Low Voltage Directive, 2014/35/EU. Conformity with this directive is based upon compliance with the harmonized safety standard, EN60950-1, 3rd Edition.

WEEE and Battery Directive Compliance

This product, and the batteries used to power the product, should not be disposed of as unsorted municipal waste and should be collected separately and disposed of according to your national regulations.

VIAVI has established a take-back process in compliance with the EU Waste Electrical and Electronic Equipment (WEEE) Directive, 2012/19/EU, and the EU Battery Directive, 2006/66/EC.

Information and instructions for returning waste equipment and batteries to VIAVI can be found on the VIAVI website in the WEEE section of the VIAVI Standards and Policies web page at: https://www.viavisolutions.com/en-us/corporate/legal/policies-standards#sustain.

CA Proposition 65

California Proposition 65, officially known as the Safe Drinking Water and Toxic Enforcement Act of 1986, was enacted in November 1986 with the aim of protecting individuals in the state of California and the state's drinking water and environment from excessive exposure to chemicals known to the state to cause cancer, birth defects or other reproductive harm.

VIAVI's position statement on the use of Proposition 65 chemicals in VIAVI products can be found in the Hazardous Substance Control section of the VIAVI Standards and Policies web page at: https://www.viavisolutions.com/en-us/corporate/legal/policies-standards#sustain.

Ordering information

This guide is a product of the VIAVI Technical Publications Department, issued as part of the RSR Transcoder 20.

- The material number associated with this manual is 22177262 R003. This manual is available on the VIAVI website in PDF format.

ii

About this User Guide

This prefix explains how to use this User Guide and includes the following topics:

- “Purpose and scope” on page viii
- “Product Nomenclature” on page viii
- “Intended User” on page viii
- “Related Information” on page viii
- “Contact Information” on page viii
- “Conventions” on page ix
- “Safety and compliance information” on page xi

Purpose and scope

The purpose of this guide is to help you successfully use the RSR Transcoder 2.0 features and capabilities. This guide includes task-based instructions that describe how to install, configure, use, and troubleshoot the RSR Transcoder 2.0. This guide also provides VIAVI warranty, services, and contact information.

> **NOTE**
> The information provided in this manual applies to all Transcoder 2.0 products except as noted for the module and enclosure variants.

Product Nomenclature

The following terms are used in this m annual to refer to the various models in the RSR Transcoder 2.0 family:

- RSR Transcoder 2.0
- Transcoder 2.0
- Transcoder
- Device

Intended User

This guide is intended for users who are familiar with PNT (Position, Navigation, and Timing) technologies and system operation.

This guide is intended for novice, intermediate, and experienced users who want to use the RSR Transcoder 2.0 effectively and efficiently.

Related Information

This manual is contains information about using the instrument. This manual also provides contact information for VIAVI's Technical Assistance Center (TAC).

Contact Information

Contact the Technical Assistance Center (TAC) for technical support or with any questions regarding this or other VIAVI products.

- Phone: 1-844-GO-VIAVI
- Email: Techsupport.Avcomm@viavisolutions.com

For the latest TAC information, go to:

https://www.viavisolutions.com/en-us/support/customer-support/technical-support

Conventions

This manual uses conventions and symbols, as described in the following tables.

Table 1 Text formatting and other typographical conventions

Item(s)	Example(s)
Buttons, keys, or switches that you press or flip on a physical device.	Press the **On** button. – Press the **Enter** key. – Flip the **Power** switch to the on position.
Buttons, links, menus, menu options, tabs, or fields on a PC-based or Web-based user interface that you click, select, or type information into.	Click **Start** – Click **File > Properties**. – Click the **Properties** tab. – Type the name of the probe in the **Probe Name** field.
Directory names, file names, and code and output messages that appear in a command line interface or in some graphical user interfaces (GUIs).	`$NANGT_DATA_DIR/results` (directory) – `test_products/users/defaultUser.xml` (file name) – `All results okay.` (output message)
Text you must type exactly as shown into a command line interface, text file, or a GUI text field.	– Restart the applications on the server using the following command: **`$BASEDIR/startup/npiu_init restart`** Type: **`a:\set.exe`** in the dialog box.
References to guides, books, and other publications appear in *this typeface*.	Refer to *Newton's Telecom Dictionary.*
Command line option separators.	`platform [a|b|e]`
Optional arguments (text variables in code).	`login [platform name]`
Required arguments (text variables in code).	`<password>`

Table 2 Symbol conventions

	This symbol indicates a note that includes important supplemental information or tips related to the main text.
	This symbol represents a general hazard. It may be associated with either a DANGER, WARNING, CAUTION, or ALERT message. See Table 3 for more information.

Table 2 Symbol conventions (Continued)

	This symbol represents an alert. It indicates that there is an action that must be performed in order to protect equipment and data or to avoid software damage and service interruption.
	This symbol represents hazardous voltages. It may be associated with either a DANGER, WARNING, CAUTION, or ALERT message. See Table 3 for more information.
	This symbol represents a risk of explosion. It may be associated with either a DANGER, WARNING, CAUTION or ALERT message. See Table 3 for more information.
	This symbol represents a risk of a hot surface. It may be associated with either a DANGER, WARNING, CAUTION, or ALERT message. See Table 3 for more information.
	This symbol represents a risk associated with fiber optic lasers. It may be associated with either a DANGER, WARNING, CAUTION or ALERT message. See Table 3 for more information.
	This symbol, located on the equipment, battery, or the packaging indicates that the equipment or battery must not be disposed of in a land-fill site or as municipal waste, and should be disposed of according to your national regulations.

Table 3 Safety definitions

Term	Definition
DANGER	Indicates a potentially hazardous situation that, if not avoided, *will* result in death or serious injury. It may be associated with either a general hazard, high voltage, or other symbol. See Table 2 for more information.
WARNING	Indicates a potentially hazardous situation that, if not avoided, *could* result in death or serious injury. It may be associated with either a general hazard, high voltage, or other symbol. See Table 2 for more information.
CAUTION	Indicates a potentially hazardous situation that, if not avoided, could result in minor or moderate injury and/or damage to equipment. It may be associated with either a general hazard, high voltage, or risk of explosion symbol. See Table 2 for more information. When applied to software actions, indicates a situation that, if not avoided, could result in loss of data or a disruption of software operation.
ALERT	Indicates that there is an action that must be performed in order to protect equipment and data or to avoid software damage and service interruption.

Safety and compliance information

The following sections describe the safety and compliance information for the RSR Transcoder 2.0.

California Proposition 65

California Proposition 65, officially known as the Safe Drinking Water and Toxic Enforcement Act of 1986, was enacted in November 1986 with the aim of protecting individuals in the state of California and the state's drinking water and environment from excessive exposure to chemicals known to the state to cause cancer, birth defects or other reproductive harm.

For the VIAVI position statement on the use of Proposition 65 chemicals in VIAVI products, see the Hazardous Substance Control section of the VIAVI Policies & Standards web page.

Federal Communications Commission (FCC)

The equipment was tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case you will be required to correct the interference at your own expense.

The authority to operate this equipment is conditioned by the requirements that no modifications be made to the equipment unless the changes or modifications are expressly approved by VIAVI.

Product Environmental Compliance

VIAVI is committed to compliance with all applicable laws and regulations controlling the use of hazardous substances in its products, as well as the disposal of equipment (including batteries) and waste packaging. For details, see the VIAVI Policies & Standards web page or contact the VIAVI WEEE Program Management team at Global.WEEE@ViaviSolutions.com.

EU REACH

Article 33 of EU REACH regulation (EC) No 1907/2006 requires product suppliers to provide information when a substance included in the list of Substances of Very High Concern (SVHC) is present in an product above a certain threshold.

For information about the presence of REACH SVHC in VIAVI products, see the Hazardous Substance Control section of the VIAVI Policies & Standards web page.

Additional standards compliance

The equipment meets the following standards and requirements:

- Installation Category (Over Voltage Category) II under IEC 60664-1
- Pollution Degree 2 Category under IEC 62368-1 Safety Requirements for Electrical Equipment for Measurement, Control, and Laboratory Use

Electrostatic Discharge (ESD)

CAUTION

Devices are ESD sensitive and should only be installed, removed and/or serviced by Qualified Service Personnel.

Table of Contents

1

Introduction

This chapter discusses the following topics:

About the RSR Transcoder

The RSR Transcoder 2.0 is a compact, self-contained GPS simulator capable of generating real-time, full-constellation, 7-channel GPS RF signals. It can function independently as a GPS L1/L2 source or interface with external Position/Velocity/Navigation/Timing (PVT/PNT) systems to convert NMEA data and 1PPS signals into GPS RF output. The device supports L1 C/A Pcode and L2 Pcode formats and can encode external timing references with nanosecond-level accuracy. No external processing equipment is required, and the unit operates with low power consumption in a small form factor.

The RSR Transcoder 2.0 enables integration with modern navigation standards by converting various data formats into L1/L2 RF signals. It supports real-time transcoding from GNSS and non-GNSS sources, including INS, Satelles/Iridium, SAASM, M-Code, Glonass, Galileo, BeiDou, and QZSS. An optional internal Chip Scale Atomic Clock (CSAC) module provides extended holdover capabilities. The device also includes a built-in 9-DOF Inertial Navigation System to support advanced features such as high-rate position interpolation and short-term navigation continuity.

Typical applications include:

- Enhancing holdover in timing systems
- Relaying secure PVT data across multiple receivers
- Retrofitting legacy GPS units for compatibility with modern standards
- Adding RF modulation capabilities to timing equipment
- Optional ICD-153 interface for receiving data from SAASM or M-code receivers. Contact VIAVI for more information.

The device can be installed by replacing the target receiver's antenna and connecting a suitable data source via USB or RS-232. It can also serve as a coax-connected time-transfer tool or a wireless pseudo-lite transmitter, supporting specifications such as IS-GPS-250A.

The RSR Transcoder 2.0 integrates a complete VIAVI CSAC-capable GPSDO module which allows automatic disciplining of either the internal high-stability TCXO or the optional internal CSAC oscillator to an external 1PPS reference using battle-proven VIAVI disciplining algorithms.

The RSR Transcoder 2.0 is available as both a standalone module and an IP68 rated enclosure.

RSR Transcoder 2.0 module

The RSR Transcoder 2.0 is available as a standalone module. Figure 1 shows the RSR Transcoder 2.0 module.

Figure 1 RSR Transcoder 2.0 module

The RSR Transcoder 2.0 module includes two power supply inputs that may be used separately or simultaneously. The unit operates from a 7.5V to 36V DC power supply. It can accept PVT fixes over its RS-232 or USB serial ports via its built-in NMEA parsing capability, in addition to accepting PVT fixes via standard SCPI commands. Simulation motion control commands can be stored in internal EEPROM and used to automatically start a dynamic simulation scenario with full autonomy from any external control requirements. Front-end GNSS receivers can be connected via the RS-232 serial port. Auto-detection and auto-configuration of uBlox and Rockwell Collins GPS receivers such as the DAGR, MicroGRAM, RSR Puck, and GB-GRAM and various Novatel receivers is supported, along with civilian-grade uBlox 5th- through 8th-generation and later GNSS receiver product lines.

RSR Transcoder 2.0 enclosure

The RSR Transcoder is available in a rugged enclosure. Figure 2 shows the Transcoder 2.0 enclosure.

Figure 2 Transcoder 2.0 enclosure

The RSR Transcoder 2.0 enclosure operates from a 7V to 36V DC power supply. The unit can accept PVT fixes over its RS-232 or USB serial ports via its built-in NMEA parsing capability, in addition to accepting PVT fixes via standard SCPI commands. Simulation motion control commands can be stored in internal EEPROM and used to automatically start a dynamic simulation scenario with full autonomy from any external control requirements. Front-end GNSS receivers can be connected via the RS-232 serial port. Auto-detection and auto-configuration of uBlox and Rockwell Collins GPS receivers such as the DAGR, MicroGRAM, RSR Puck, and GB-GRAM and various Novatel receivers is supported, along with civilian-grade uBlox 5th- through 8th-generation and later GNSS receiver product lines.

Accessories

The following optional accessories are available for the RSR Transcoder 2.0:

- 20 pin pigtail cable (standalone module only)
- Power DIN mating connector (enclosure only)
- Communication 12-pin DIN mating connector (enclosure only)

Software Options

The RSR Transcoder software is available with the following options:

- ICD 153
- 100Hz Refresh Rate

Contact your VIAVI sales representative for more information.

Operating Principles

The RSR Transcoder 2.0 is based on a full-constellation real-time GPS simulator. The unit includes an ARM Cortex main processor that handles communications, calculations, and oscillator disciplining, as well as a high-performance FPGA that implements RF signal generation for up to 8 concurrent GPS channels. Tight coupling between the processor, the FPGA, and the timing reference allows real-time encoding of PVT/PNT data into a GPS L1/L2 C/A and unencrypted P-code RF signal.

In operation, baseband signals in I/Q format representing the individual GPS channels are summed digitally and RF modulated using a TX DAC at the GPS L1/L2 frequency of 1575.42 MHz, taking advantage of sample clock aliasing to avoid the need to generate a high-level RF carrier signal internally. Traditional GNSS simulators that rely on generation and subsequent attenuation of a much-stronger signal must include bulky and costly shielding due to the extremely low power levels of the desired output signal, often well below -120 dBm. In the RSR Transcoder 2.0, the low-level RF signals used internally can be filtered and attenuated both effectively and economically without adding size, weight, or power consumption to the package. The resulting output power level is user-controllable over ~20 dB of range.

Prior to the output jack, an internal RF splitter feeds a portion of the simulated GPS signal to an on-board 8th-generation GNSS receiver which provides signal monitoring and calibration capabilities. The output signal then undergoes final filtering, exiting the transcoder via a resistive pad which includes a 186-ohm DC resistance to ground to simulate a typical GPS antenna load from the perspective of the external target GNSS receiver. The RSR Transcoder 2.0's RF output is compatible with external GPS-provided antenna voltages up to 6V. Care must be taken to avoid supplying antenna voltages higher than 6V to the RF SMA connector.

The RSR Transcoder 2.0 includes circuitry to time-stamp an external 1PPS reference signal to better than +/-2.8ns typically, as well as circuitry to discipline its internal high-stability TCXO or optional CSAC oscillator from this external 1PPS reference signal. Additional features include a USB port for optional PC-based configuration and control.

General Safety Precautions

The following general safety precautions must be observed during all phases of operation of this instrument. Failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture, and intended use of the instrument. VIAVI assumes no liability for the customer's failure to comply with these requirements.

Transmission of synthesized GPS RF signals

NOTE

It is illegal to transmit simulated/synthesized GPS RF signals. The RF output of the RSR Transcoder 2.0 must not be fed to a transmitting antenna. It is intended only to be directly coupled into the RF input of a GPS receiver using shielded coax cables. The RF output must not be amplified or re-transmitted in any way without approval from the appropriate government authorities.

Surge Protector

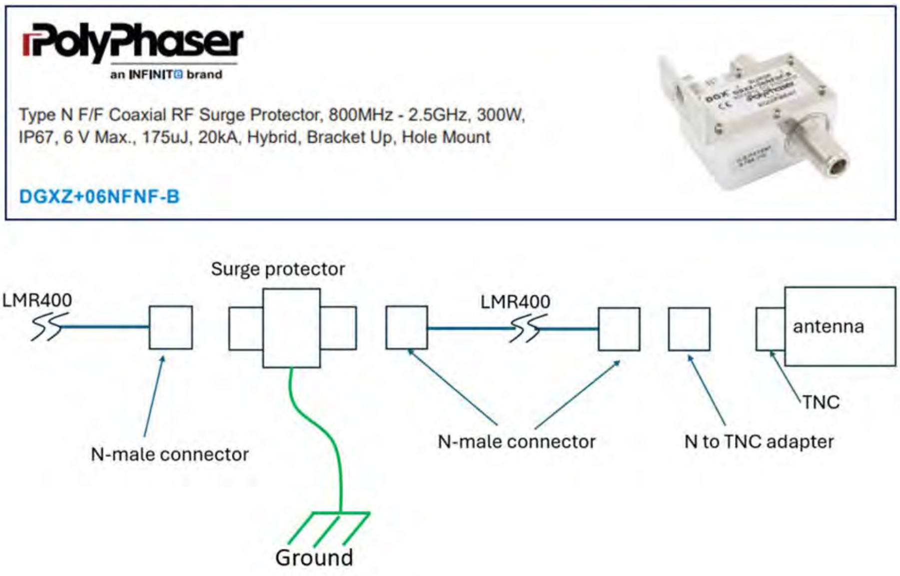
Use a surge protector with the following requirements:

- DC pass voltage threshold supports antenna power
- Clamp voltage low enough to protect downstream equipment
- Frequency response includes L-band

Surge protectors offer various installation methods. Follow the specific installation instructions provided with the surge protector.

Figure 3 shows an example for a 5V antenna power feed.

Figure 3 Surge protector example



Grounding

WARNING

Improper grounding of device can result in electrical shock and damage to the sensitive electronic components in the RSR Transcoder 2.0. To ensure proper grounding, this device should be connected to a grounded source such as a power supply ground.

To avoid damaging the sensitive electronic components in the RSR Transcoder 2.0, always make sure to discharge any built-up electrostatic charge to a good ground source, such as the power supply ground. This should be done before handling the circuit board or anything connected to it such as an external receiver's GNSS antenna.

Power Connections

When connecting DC power to the device, follow the guidance provided in "Powering the device" on page 10.

Environmental Conditions

This instrument is intended for indoor use with a properly-installed GNSS Antenna Lightning Protector. It is designed to operate at a maximum relative non-condensing humidity of 95% and at altitudes of up to 50,000 meters.

Legacy GPS Receiver Compatibility

VIAVI strives to achieve the greatest possible degree of compatibility with legacy GPS receivers. If any incompatibilities are encountered with specific receiver models, please report these to support@viavisolutions.com.

2

Installation and Configuration

The following topics are discussed in this chapter:

Introduction

The RSR Transcoder 2.0 supports various hardware connections and configuration options. This chapter focuses on a basic hardware setup and simulation configuration, allowing you to quickly start using the RSR Transcoder 2.0 as a simulator. Chapter 5 provides a command reference for the supported SCPI command set.

Configuring the RSR Transcoder 2.0 as a simulator requires a USB connection to a computer for communications and an RF connection to the target GPS receiver. Configuration and status commands are sent through a virtual COM port associated with the USB connection to the RSR Transcoder 2.0.

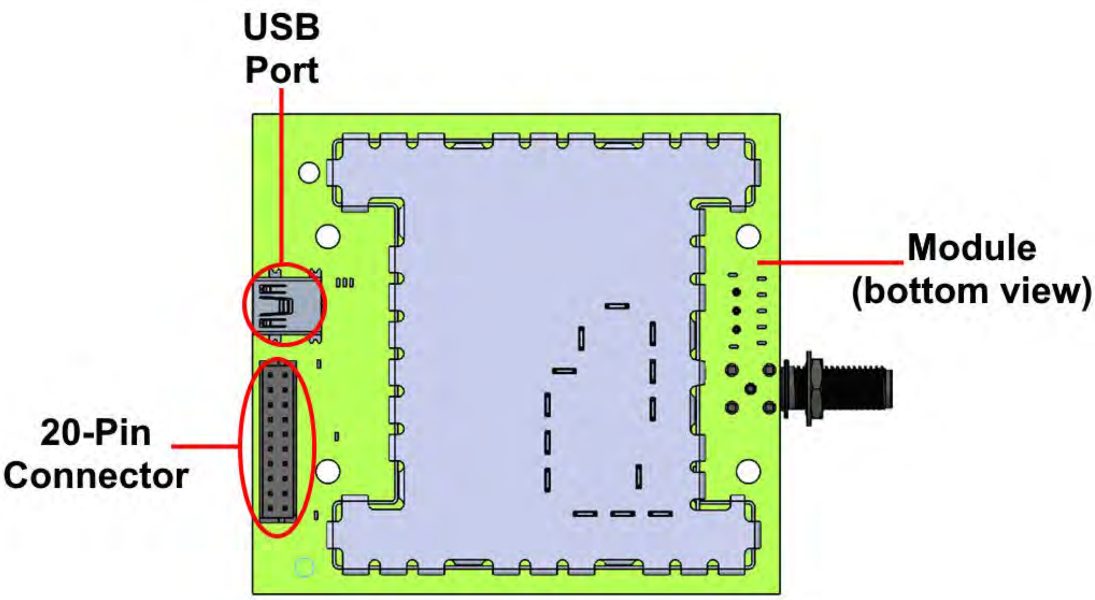
Powering the device

The following sections describe how to power the Transcoder 2.0.

Module

The power and control connection to the RSR Transcoder 2.0 is made through the 20-pin connector on the RSR Transcoder 2.0 and a computer. The location of the 20-pin connector port is shown in Figure 4.

Figure 4 20-pin connector port on RSR Transcoder 2.0 PCB



Power is provided externally from a DC power supply. See "DC Power supply: 7.5V to 36V" for more information.

DC Power supply: 7.5V to 36V

The unit is powered by the DC power input port. DC power should be applied to pin 1 of connector J31, with pin 2 grounded. The allowable voltage range on this DC power port is 7.5V to 36V (12V nominal).

NOTE

Care must be taken to avoid surge voltages higher than 36V to be applied to the unit such as can happen when hot-plugging the unit with voltages higher than 15V due to the lead inductance, and low-ESR internal bypass capacitors. This can typically be accomplished by using a 1 Ohms or higher series damping resistor on the power line.

Typical current consumption of the RSR Transcoder 2.0 at 12V is approximately 0.12A, or approximately 0.14A when equipped with the CSAC option.

Enclosure

The power and control connection to the RSR Transcoder 2.0 is made using the 3-pin circular DC In port. Figure 5 shows the location of the DC In port.

Figure 5 DC In port location on enclosure

Power is supplied to the unit using a shielded, 3-position M8-style circular connector.

The threaded connector has male pins and is IP68 rated. The connector is mounted directly to the PCBA. The part number is ***M8S-03PMMR-SF8001 from Amphenol LTW***. Figure 6 shows the pin locations for the ‘12 VDC Power Input connector.

Figure 6 12 VDC Power Input connector pin locations

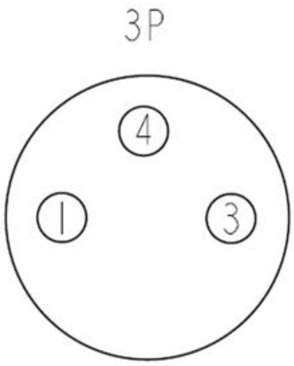


Table 1 describes the pins.

Table 1 12 VDC Power Input Connector Pin descriptions

Pin	Signal	Description
1	GND	Connect to system ground.
3	DC Input	Supply power to the unit. Vin range: – +7.5 to +36VDC – +12VDC nominal – Typical power: 4W
4	N/A	N/A

Connectors

The following sections describe the connectors for the RSR Transcoder 2.0 module and enclosure.

Module

The RSR Transcoder 2.0 module has three major connectors:

- J31 for power and control of the external GNSS receiver or PVT/PNT source, 10 MHz out, and 1PPS In/Out
- J61, which provides the GPS L1/L2 C/A RF output signal to the target GPS receiver
- J2 for USB serial control.

Figure 7 shows the major connections from the top view of the module.

Figure 7 Module connections (top view)

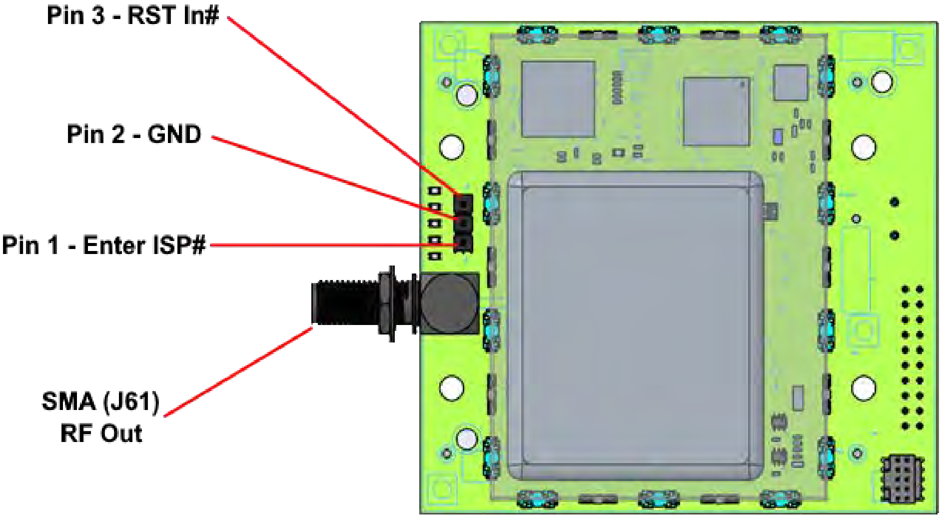


Figure 8 shows the module connections as viewed from the bottom.

Figure 8 Module connections (top view)

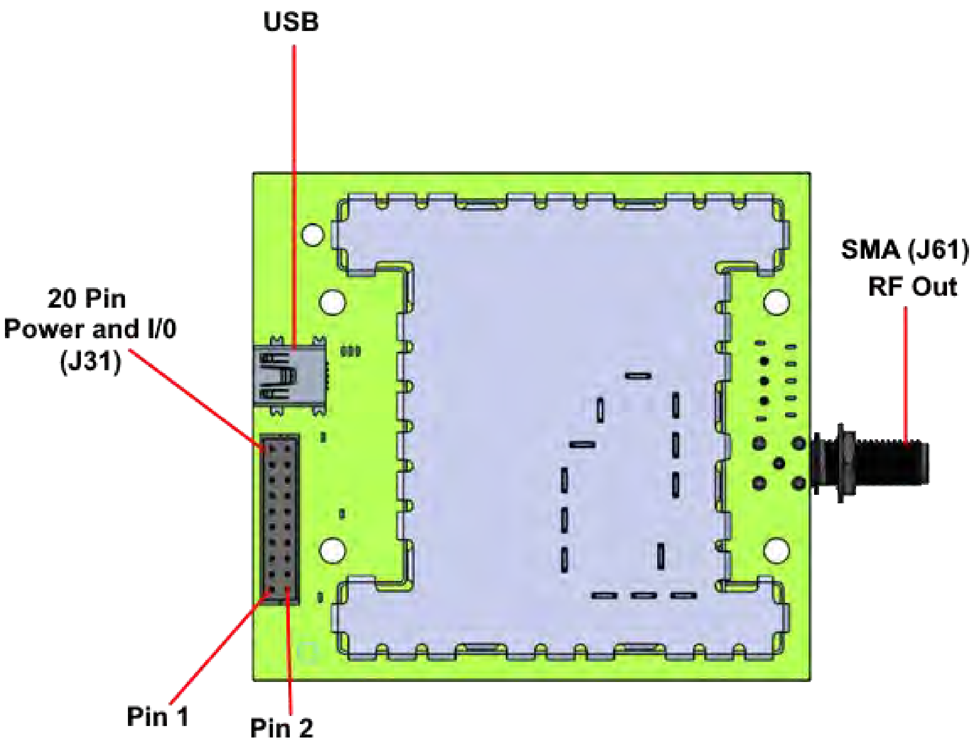


Table 2 describes the connectors.

Table 2 Transcoder 2.0 Module connectors

Name	Connector Type	Function	Notes
RF Out (J61)	SMA	Transcoder RF output	L1 + L2 GPS

Table 2 Transcoder 2.0 Module connectors

Name	Connector Type	Function	Notes
J31	20-pin shrouded header	Power input Data I/O	– Hirose Electric DF11-20DP-2DSA(01) – Mates with cable connector PN: Hirose Electric DF11-20DS-2C – Mates with socket (board to board) PN: Hirose Electric DF11-20DS-2DSA(##)
USB	Data I/O	In/Out	Mini USB

Table 3 shows the signals on connector J31. This connector is a 20-pin male connector from Hirose, part number DF11-20DP-2DSA(01). The use of connector J31 must be used to power the board and control it via the built-in USB connector.

Table 3 20-pin connector (J31) - pinout and signal descriptions

Pin	Signal	Dir	Type	Function
1	PWR_IN	In	DC PWR Input	+7.5VDC to +36VDC – +12VDC nominal – Primary power input for standalone module. – Typical power: 4W
2	GND	n/a	Signal return GND	Connect to system GND **NOTE –** VIAVI recommends the use of a 1A inline fuse or circuit breaker.
3	+5.5Vin	In	Alternate DC Power input	Can be used as a 3.5V to 5.5V Input to board if Primary power is unavailable. Typical power: 3W **WARNING –** Do Not use if Primary power is supplied to Pin 1.
4	10MHZ_BUFF	Out	+3.3V CMOS	3.3V CMOS 10 MHz buffered output from internal TCXO. Must be software-enabled via SCPI command.

Table 3 20-pin connector (J31) - pinout and signal descriptions

Pin	Signal	Dir	Type	Function
5	RS232_RX	Out	RS-232 SIGNAL LEVEL	RS-232 Serial Port transmitter output for module UART3. Connect to RX from external host serial port. This Port can also be used to update firmware ISP-Mode.
6	ENTER ISP#	In	+3.3V CMOS	Places unit into programming mode when shorted to GND during power up.
7	RS232_TX	In	RS-232 SIGNAL LEVEL	RS-232 Serial Port receive for module UART3. Connect to TX of external host serial port. This Port can also be used to update firmware ISP-Mode.
8	1PPS_OUT	Out	+3.3V CMOS	1 Pulse-Per-Second Timing locked to oscillator
9	GPS_1PPS_IN	In	CMOS 3.3V 5V-tol	1PPS Reference Input from external reference or GNSS
10	GPS_PWR_OFF	Out	+3.3V GPIO	Enable for external Legacy equipment
11	LOCK_OK_OUT	Out	CMOS 3.3V	Internal Oscillator LOCK Status: – 3.3V indicates internal oscillator is locked to external 1PPS reference, and no events are pending. – 0V indicates either oscillator is unlocked, or events are pending and can be queried with the SYNC:HEALTH? SCPI command.

Table 3 20-pin connector (J31) - pinout and signal descriptions

Pin	Signal	Dir	Type	Function
12	MCU_PB_SW_O-C	In	CMOS 3.3V	Push button input from membrane keypad in enclosure. Not used for standalone module.
13	Status_LED0	Out	+3.3V CMOS	Status 0 LED. Not used for standalone module.
14	GND	n/a	Signal return GND	Connect to system GND
15	LEDS_ENA	Out	+3.3V CMOS	Enables/disables status LEDs on membrane keypad. Not used for standalone module.
16	USB+	I/O	Pos+ half of Differential USB Signal	USB-Serial Interface. This port can also be used to update firmware in ISP mode.
17	Status_LED1	Out	+3.3V CMOS	Status 1 LED. Not used for standalone module.
18	USB-	I/O	Neg- half of Differential USB Signal	USB-Serial Interface. This port can also be used to update firmware in ISP mode.
19	ALARM	Out	+3.3V CMOS	Alarm Status. Used for LED on enclosure membrane keypad. Not used for standalone module.
20	GND	n/a	Signal return GND	Connect to system GND

Enclosure

The RSR Transcoder 2.0 Enclosure has three major connections:

- RF out SMA (J3)
- I/O DIN 12-pin circular connector (J2)

- DC In 3-pin circular connector (J1)

Figure shows the locations of the connectors on the enclosure

Figure 9 Enclosure connection locations

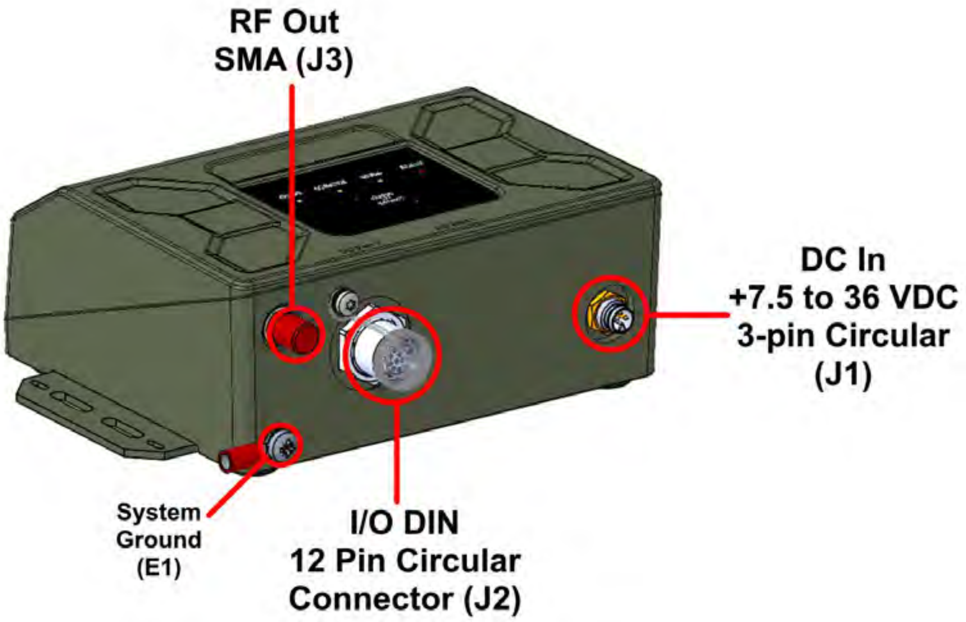


Table describes the connectors.

Table 4 RSR Transcoder 2.0 enclosure connectors

Name	Connector Type	Function	Notes
RF Out (J3)	SMA	Transcoder RF output	L1 + L2 GPS
I/O DIN 12-pin Circular (J2)	12-pin circular connector	Serial programming and data I/O	Norcomp 860-012-213R004 Mates with cable end connectors: – Bulgin PXMB-NI12FIM12ASCPG9 or PXMBNI12FIM12AS-CPG7 as appropriate for cable size. – Other M12 series, A-Code, 12 male contact connector assemblies with appropriate IP6X rating and shielding may be used. **Note** — Norcomp 858-012-103RKTY connector series not recommended.
DC In (J1)	3-pin circular connector	DC Power input.	See “Powering the device” on page 10 for more information.

Figure shows the 12-pin connector (J2) pin locations.

12-pin connector (J2) pin locations

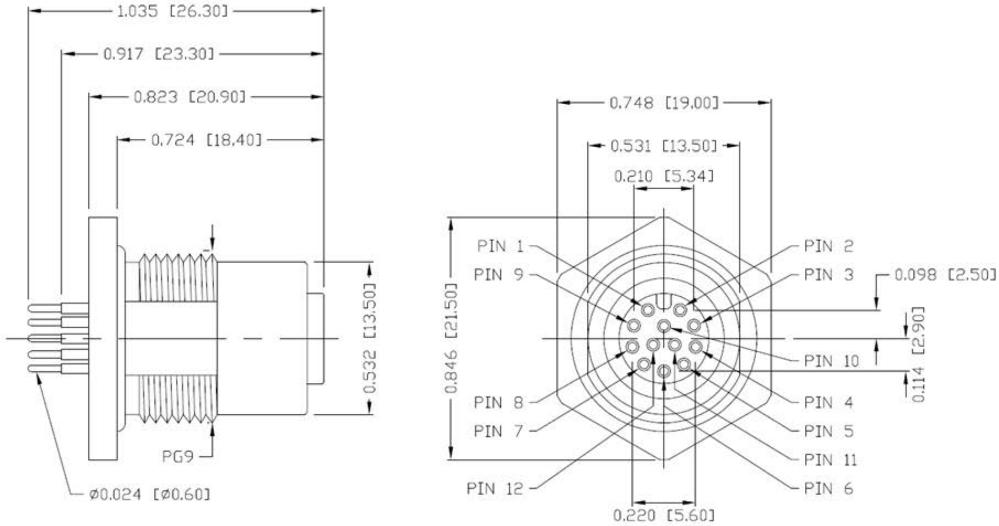


Table describes the 12-pin connector pins.

Table 5 12-pin connector pins

PIN	Signal	Dir	Type	Function
1	1PPS_IN	In	CMOS 3.3V 5V-tol	1PPS Reference input from external reference or GNSS.
2	USB+	I/O	Pos+ half of Differential USB Signal	USB-Serial Interface This Port can also be used to update firmware ISP-Mode.
3	USB-	I/O	Neg- half of Differential USB signal.	USB-Serial Interface This Port can also be used to update firmware ISP-Mode.
4	GND	n/a	Signal return GND	Connect to system GND.
5	LOCK_OK_OUT	Out	CMOS 3.3V	Internal Oscillator LOCK status. 3.3V indicates internal oscillator is locked to external 1PPS reference, and no events are pending. 0V indicates either oscillator is unlocked, or events are pending and can be queried with the SYNC:HEALTH? SCPI command.

Table 5 12-pin connector pins

PIN	Signal	Dir	Type	Function
6	+5.5Vin	In	Alternate Power input	Can be used as a 4V to 5.5V input to power unit if Primary power is unavailable. Typical power: 3W **WARNING –** Do not use if external power is supplied to front panel jack J1.
7	RS232_TX	In	RS-232 – +/- 30Vin max – Input low threshold 0.6V min – 1.3V typical – Input high threshold 1.6V typical – 2.4V max	Ext. Host RS-232 Serial Port TX. Input to Transcoder UART3. This Port can also be used to update firmware ISP-Mode
8	RS232_RX	Out	RS-232 +/- 5.7V typ +/- 5.0V min	Ext. Host RS-232 Serial Port RX. Output from Transcoder UART3. This Port can also be used to update firmware ISP-Mode.
9	1PPS	Out	+3.3V CMOS	1 Pulse-Per-Second. Timing locked to oscillator.
10	ENTER ISP#	In	+3.3V CMOS	Places unit into programming mode when shorted to GND during power up.

Connecting to a target GPS receiver's antenna input

The RF output from the RSR Transcoder 2.0 connects directly to the antenna input on the target GPS receiver. The SMA GPS L1/L2 RF connector is labeled in Figure 7 on page 13 and Figure 9 on page 17. In addition to an RF output, this connection also provides a 186-ohm DC load for the antenna output voltage from the target GPS receiver. Many receivers require an antenna current to operate properly.

NOTE

If the target receiver's antenna output voltage is higher than 6V, a DC block is required between the RSR Transcoder 2.0's RF output and the target receiver's antenna input. For example, an external DC block must be used with the Symmetricom/Microsemi XLI reference, as it outputs 12V DC which would damage the internal 186-ohm DC termination resistor. Use an appropriate RF cable to connect the SMA RF output to the antenna input of the target receiver, and add a DC block as necessary.

Simulation control

This section describes the low-level SCPI commands needed to set up, initiate, and stop a fixed-position simulation at a specified location and start time. These commands may be entered at the SCPI command prompt in the terminal program described in section "Powering the device" on page 10. Each command is terminated with a carriage-return character (Enter key).

NOTE

The SimCon utility available for download at the VIAVI website provides greatly-improved ease of access to these and other features of the RSR Transcoder 2.0. With SimCon, many users will not need to run a separate terminal program or enter SCPI commands directly. Refer to Chapter 3 for more information about downloading and running SimCon.

For manual START/STOP simulation control, set the simulation mode to manual:

```
SIM:MODE MANUAL
```

To set the fixed position of the simulation enter the command:

```
SIM:POS:LLH lat,lon,height
```

where lat and lon are latitude and longitude in decimal degrees and height is in meters above the GPS ellipsoid (not Mean Sea Level height). Any combination of lat/long/height parameters may be provided; for example, to change only the simulated height, enter:

```
SIM:POS:LLH ,,height
```

The simulated time can also be specified manually. This means the GPS receiver will indicate the start time and date as soon as the simulation commences. This allows simulating any time/date in the future or past as required.

> **NOTE**
> It may be necessary to perform a cold start operation on the receiver if the simulation time/date differs substantially from the receiver's last-known PNT fix.

The following procedure describes how to set the assigned time.

To set the assigned start time

1 Set the time mode to *Assigned* mode:

```
SIM:TIME:MODE ASSIGNED
```

2 Next, set the assigned start time and date with the commands:

```
SIM:TIME:START:TIME hh,mm,ss.sss
```

and

```
SIM:TIME:START:DATE yyyy,mm,dd
```

3 Start the simulation with the command:

```
SIM:COMMAND START
```

The simulation will start immediately after the Enter key has been received. The GPS receiver should indicate the selected time/date after acquisition of the simulated signal, typically within less than a minute.

The simulation will run indefinitely or until stopped with the command:

```
SIM:COMMAND STOP
```

The RF output power of the simulated GPS signal can be controlled with the command

```
OUT:POW -xxx
```

where xxx is the desired power level in dBm. The “dBm” abbreviation does not have to be entered; it is assumed. Available power levels range from approximately -90 to -130 dBm.

Steps for programming NMEA

> **NOTE**
> Other transcoding mode options are available. Contact your VIAVI sales representative for more information.

For operating in Transcoding mode, the external GPS receiver input is typically used as the source of PVT data. The input mode on the external GPS port is controlled by the GPS:TYPE:MODE command:

```
GPS:TYPE:MODE <AUTO | NMEA | UBLOX>
```

The default AUTO mode automatically detects the presence of NMEA input or ublox 6th or 8th generation GNSS receivers.

The external GPS port can be configured to accept NMEA data with the GPS:TYPE:MODE NMEA command. For Transcoding, the GPGGA and GPRMC messages are both required at a 1Hz rate or higher, up to the simulation update rate configured with the SIM:TIME:RATE command.

The NMEA port setting GPS:PORT <SCPI | RS232> setting should normally be RS232 to accept NMEA data from the external GPS port. NMEA playback with through the SCPI port is also possible with the SIM:NMEA command.

The external GPS port's baud rate should be manually set with the SYST:COMM:SER:BAUD <AUTO | 9600 | 19200 | 38400 | 57600 | 115200> command. The default AUTO setting will automatically detect between 9600, 38400 and 115200 baud rates. If the baud rate is known, the settings should be set to a fixed value.

Troubleshooting

Descriptions of all error conditions that can result in an error indication on the RSR Transcoder 2.0 enclosure and a non-zero health indicator in the SYNC:HEALTH? query response. “SYNChronization:HEAlth?” on page 74 describes each health status flag that could be high in the SYNC:HEALTH? response indicating an error condition.

Some health status flags are temporary and will usually clear after several minutes, while others may indicate a hardware problem, configuration problem or issue with the PVT input. Table 6 describes the actions that may be required to correct the various categories of health status flags.

Table 6 Corrective actions

Flags	Cause	Action Required
0x1 0x2	On units with only a TCXO clock, these flags indicate that the clock is being steered to the upper or lower limit of its control range. It is possible that the external 1PPS input used for disciplining the clock is off frequency resulting in this error.	To correct this problem, use a GPS-disciplined or other calibrated frequency reference to create the 1PPS input used to discipline the Transcoder 2.0's clock. A hardware problem with the TCXO clock frequency control may be resulting in this issue.

Table 6 Corrective actions

Flags	Cause	Action Required
0x40 0x80	These flags monitor the main 5.5V supply provided by USB power, 36 to 5V input on J31 pin 2 or 5.5V input/ output on J31 pin 11. This voltage should be in the range of 2.4 to 5.8V. Outside of this range, one of these health bits will be high.	This issue can occur if the wrong supply voltage is applied or if the USB 5V supply is not capable of providing enough current to power the Transcoder board.

Table 6 Corrective actions

Flags	Cause	Action Required
0x4 0x8 0x20 0x100 0x200	These flags indicate instability in the oscillator disciplining. Usually these flags are only temporary, and the oscillator discipling will stabilize and lock in several minutes.	If one or more of these health flags remain high after several minutes of disciplining with an external 1PPS input, the frequency calibration error of the Transcoder's clock may be large or the external 1PPS input frequency error may be large, and it will take longer for the clock disciplining to lock to the external 1PPS. Make sure that the external 1PPS input is always from a calibrated time/frequency source and blanked (disabled) when not valid. If the external 1PPS cannot be disabled, the fix status indication in the externally provided PVT data (typically NMEA GPGGA and GPRMC messages) can be used to indicate the validity of the 1PPS input when using the SYNC:SOURCE:MODE GPS setting. With the SYNC:SOURCE:MODE EXT setting, the external 1PPS input is always consider valid. If one or more of these health flags remain high after an hour or more of disciplining with an external 1PPS input, or one or more of these health flags occasionally are high, it is possible that the external 1PPS input is not stable or varies significantly in frequency so that the disciplining loop cannot lock to it. If a more stable 1PPS input is not available, it may be possible to adjust the oscillator disciplining parameters via the SCPI interface, but making these adjustments is not straight forward.

Table 6 Corrective actions

Flags	Cause	Action Required
0x10	This flag indicates oscillator disciplining holdover caused by removal of the external 1PPS, loss of valid fix status in the GPGGA/GPRMC messages with the SYNC:SOURCE:MODE GPS setting or entering manual holdover with the SYNC:HOLD:INIT command. This indication is normal when the clock disciplining is in holdover mode. Some use cases rely on the stability of the Transcoder 2.0's clock during operation if an external 1PPS input is no longer available.	If the external 1PPS is valid, but the fix status in the GPGGA/GPRMC messages does not indicate the 1PPS validity, use the SYNC:SOURCE:MODE EXT instead. To exit manual holdover, use the SYNC:HOLD:RECOVERY:INIT command.
0x800	For ublox external GNSS receivers only, this flag indicates strong jamming signals causing oscillator disciplining holdover.	The indicator will be corrected when the ublox receiver no longer indicates high jamming signals and regains a 3D fix.
0x2000 0x4000	These flags indicate internal errors in the GPS simulation engine.	No action is usually required and these flags usually clear automatically. If these errors persist, there could be a hardware problem.

Table 6 Corrective actions

Flags	Cause	Action Required
0x8000 0x10000 0x20000	These flags indicate issues with the internal GPS monitoring receiver. These flags may be caused by instability of the simulation timing or simulated position. Timing instability often occurs because the Transcoder's clock is still locking to the external 1PPS input. Position instability may occur because of issues with the position filtering or a jump in the position provided in the PVT input data	After periods of timing or simulated position instability, these flags should automatically clear. In addition to position jumps causing temporary simulation position instability, incorrect position filter settings can also cause longer periods of simulated position instability. Make sure to use the recommended position filter settings for the given simulation update rate (SIM:TIME:RATE setting) as described in the "TBD document 2".
0x40000	This flag indicates that the internal GPS monitoring receiver has no 3D fix. Periods of timing or simulated position instability can also cause loss of lock in the internal GPS monitoring receiver. Lack of or invalid GPS constellation data with the SIM:LNAV:SEL LIVE or USER settings can also result in no satellites or invalid satellite orbits being simulated.	After periods of timing or simulated position instability, the internal GPS monitoring receiver should regain a 3D fix. The same actions described above can be used to correct longer periods of instability. If no GPS constellation data is available, correct this issue by providing USER or LIVE data as described in the User Manual, or switch to the SIM:LNAV:SEL AUTO setting which will always result in a valid GPS constellation. If satellites are being simulated (listed in SIM:SV:VIEW? response), but a 3D fix is still not obtained, the validity of the constellation parameters can be checked with the SIM:SV:YUMA? (almanac) and SIM:SV:EPHY? (ephemeris) query responses.

3

GPSCon Utility

The following topics are discussed in this chapter:

Description

GPSCon - VIAVI Edition is a program for the monitoring and control of a VIAVI GPSDO, Simulator and receiver products. It communicates with the receiver using the SCPI command set.

GPSCon has been replaced by the freeware program SimCon available for download from the VIAVI website support page as described in “GPSCon Utility” on page 27, however GPScon is useful in monitoring and controlling the built-in TCXO or CSAC GPSDO features when locking and disciplining the internal oscillators to external 1PPS reference signals.

A free version of the GPSCon utility is compatible with VIAVI products and is available for download from the support section of the VIAVI website:

https://www.viavisolutions.com/en-us/software-download/gpscon-controller-software

Installation

Extract the contents of the ZIP file downloaded from the VIAVI website and execute the MSI installer. Follow the on-screen instructions to complete the installation of GPSCon.

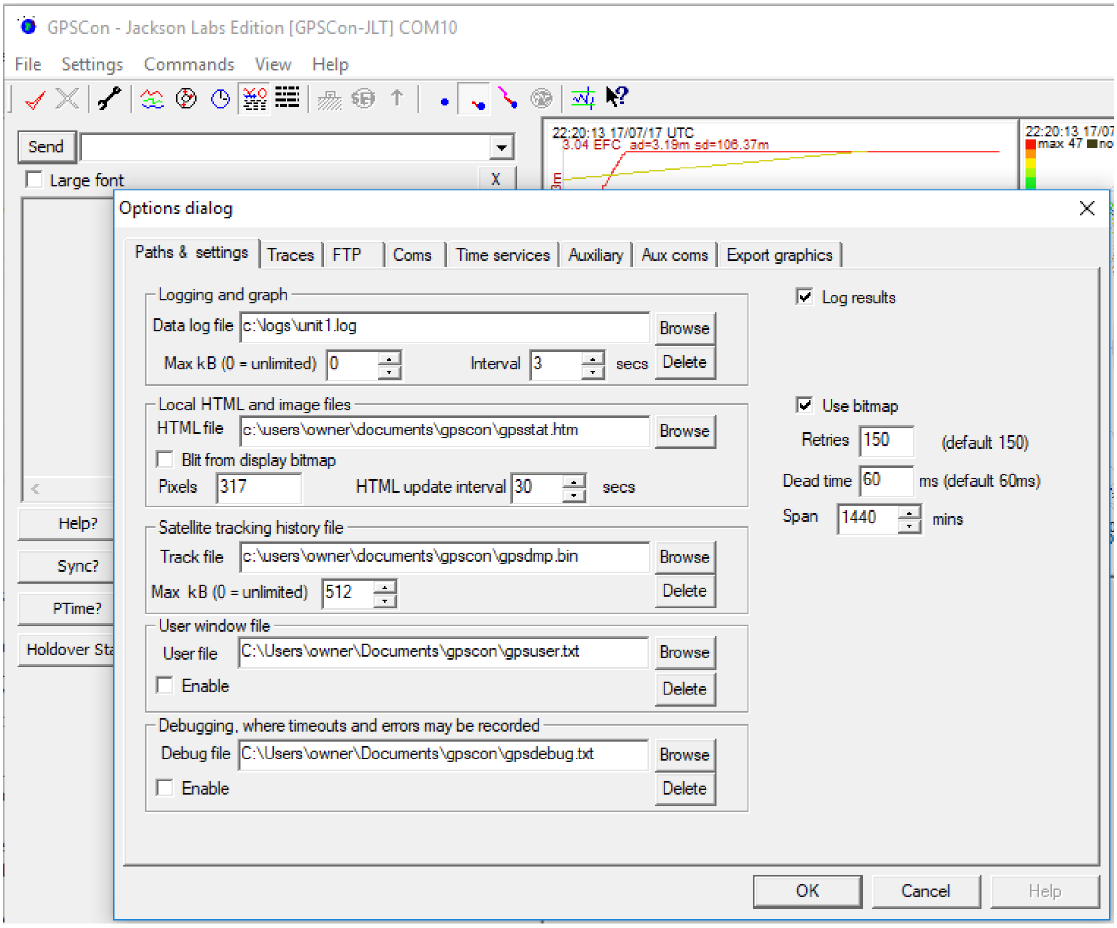
Using GPSCon

The GPSCon utility has a help file that should be consulted in order to get the full functionality of this utility. Only a few of the features and commands are mentioned in this appendix for convenience.

Setting the options

To set up the options for your GPSCon session, press the wrench icon under the menu bar, or select Settings / Options on the menu. The window shown in Figure 10 will appear. You can then select various options on the tabs.

Figure 10 Options window



Communication Parameters

Before you can use GPSCon you must set the communication parameters for your system. Open the dialog box by pressing the wrench icon and select the “Coms” tab. You will see the window shown in Figure 10 on page 29. Available COM ports on your system will be highlighted by an asterisk. Select the correct COM port for your computer and set the baud rate to 115200, parity to None, Data Bits to 8 and Stop Bits to 1. Set Flow Control to “None”. Once you have configured the communication parameters, press the “OK” button to close the window.

Auxiliary Parameters

After pressing the wrench icon, you can select the “Auxiliary” tab to configure auxiliary measurements. See Figure 12 for an example of an auxiliary measurement. You will notice that the “Aux1” request string has been set to `meas:current?<CR>` and the “Trace to go to” is set to trace position 6. See Figure 13 on page 31 for the arrangement of the trace positions in the trace window. In this example the data obtained from the meas:current? query will be plotted in trace position 6.

Traces Parameters

After pressing the  wrench icon, you can select the "Traces" tab to configure the trace labels and vertical plot ranges. See Figure 13 for an example of an auxiliary measurement. The labels and parameters are completed by default for traces 1 through 5. The auxiliary trace defined on the auxiliary tab for trace 6 has the label "Temp" to indicate that the OCXO current from the meas:current? query is a measure of temperature. Any of the eight traces can be replaced by auxiliary traces as described in "Auxiliary Parameters" on page 29. Press the "Help" button for a full description of each option in the Traces tab.

Figure 11 Setting the communications parameters

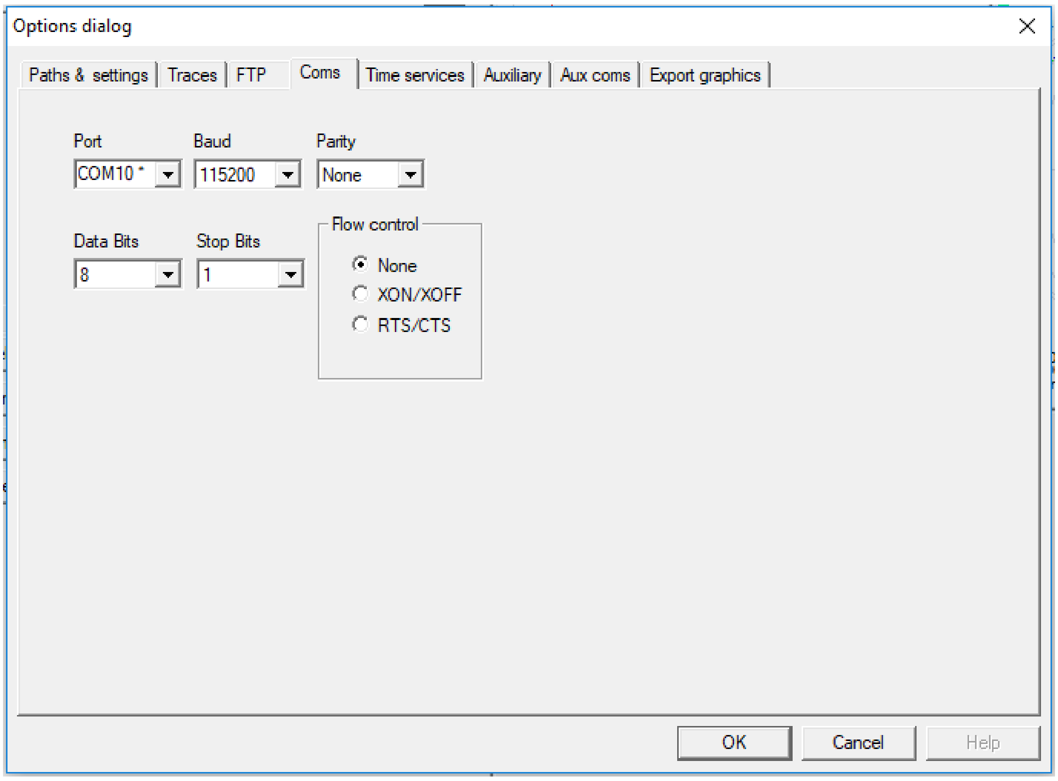


Figure 12 Auxiliary Parameters window

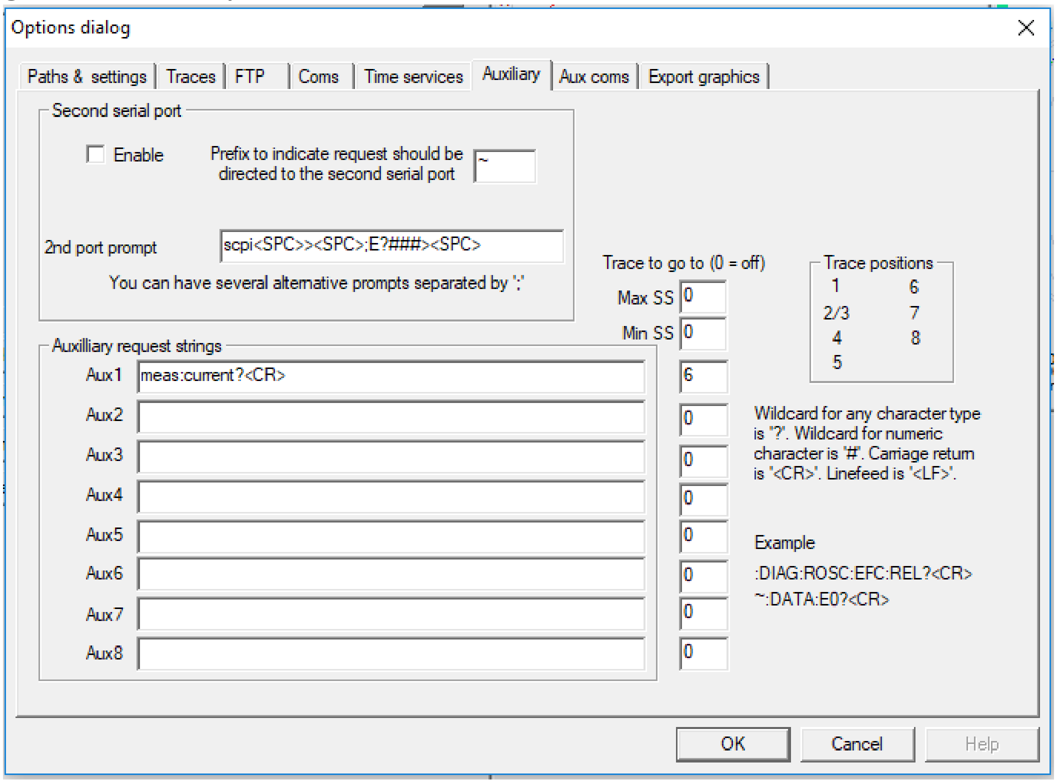
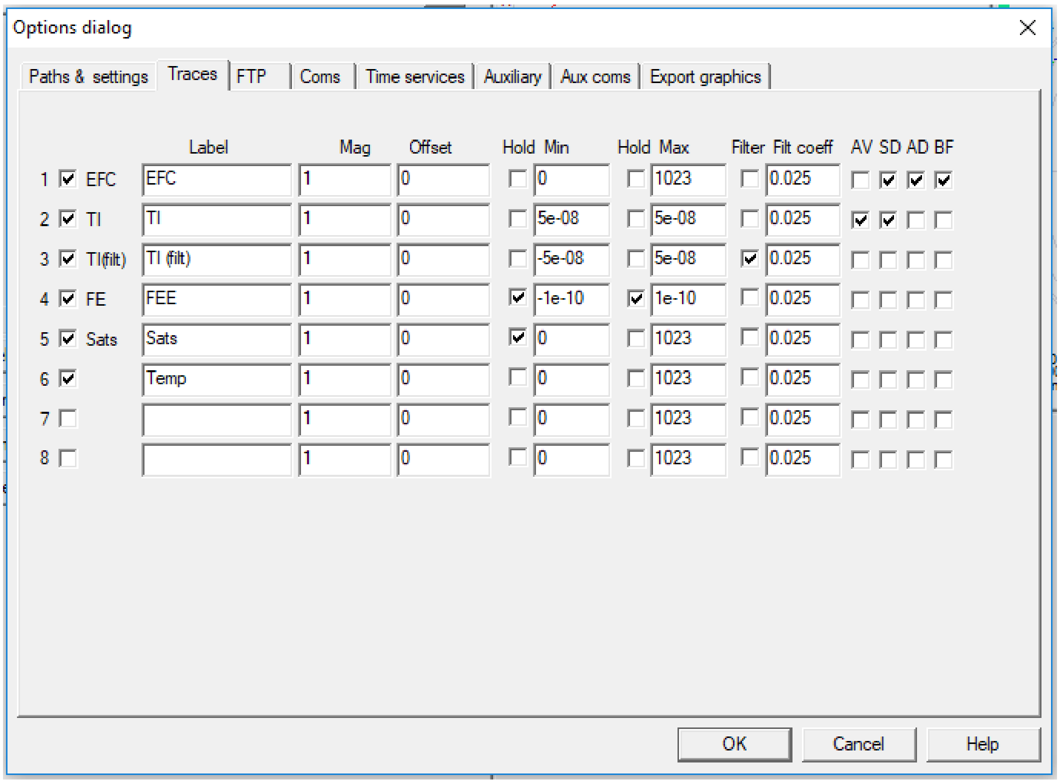


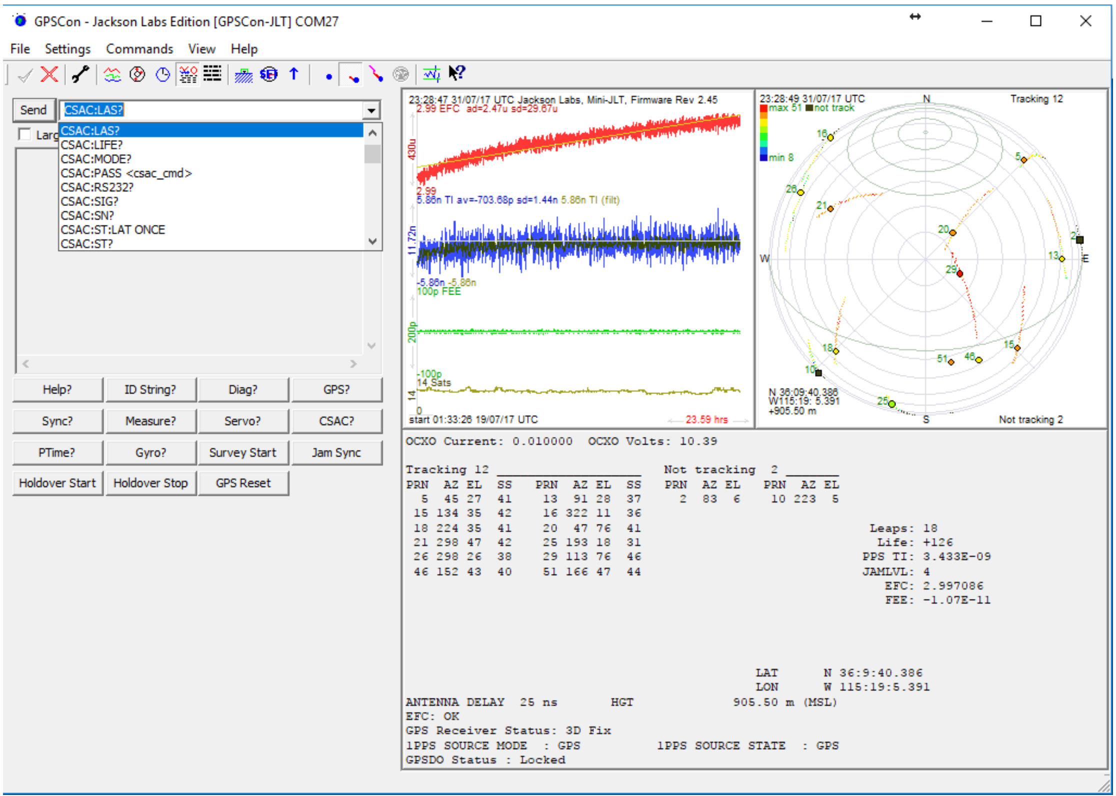
Figure 13 Traces Parameters window



Sending manual commands to the receiver

You can send SCPI commands manually by using the drop-down box in the upper left of the main window as shown in Figure 14. Care must be taken when sending these commands, so be sure that the command that you select is supported by the RSR Transcoder 2.0. Once you've selected the command you can press "Send" to send it to the RSR Transcoder 2.0. You can also send common commands by clicking on the buttons below the message window. You can hover over the buttons to see the exact command that will be sent.

Figure 14 Sending manual commands



Using the Mouse in the Graph Window

Refer to Figure 15 on page 33 for the following description.

The default view in GPSCon is "All" which you can select with the View/All menu option. To see a larger view of the graph, select the View/Graph menu option.

In the graph window the horizontal range of the graph can be set using he mouse. Set the start time by left clicking on the desired start point. If you wish, the stop time may also be set by right clicking the desired stop point. The set start and stop times can be removed by left double-clicking anywhere on the graph.

Since this is harder to describe than to actually do, here is a paraphrase of the above:

"To zoom in: The mouse is used to set the left extent and the right extent of the portion of the curve that the user wants to fill the screen. Click once with the left mouse button on the point that marks the left side of what you want to be the magnified curve.

Immediately that point becomes the left end of the curve. Then similarly click the right mouse button on the curve at the time you wish to be the right most portion of the magnified curve and it immediately becomes the end point on the right side. And, finally to return to the zoomed out ("fit to window") view, left double-click on the curve."

When you have locked the start and stop time using the mouse, you can scroll left or right through the data. To scroll to a later time, use Shift + Left click. To scroll to an earlier time, use Shift + Right click.

Figure 15 Graph display

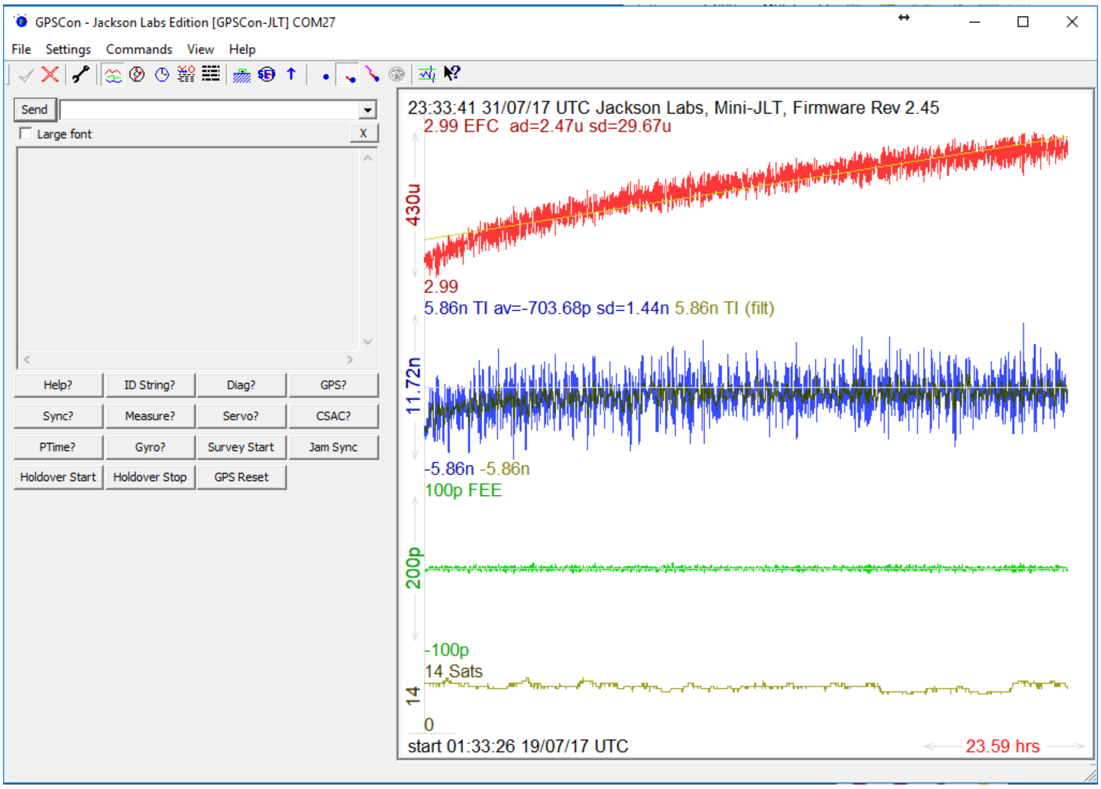
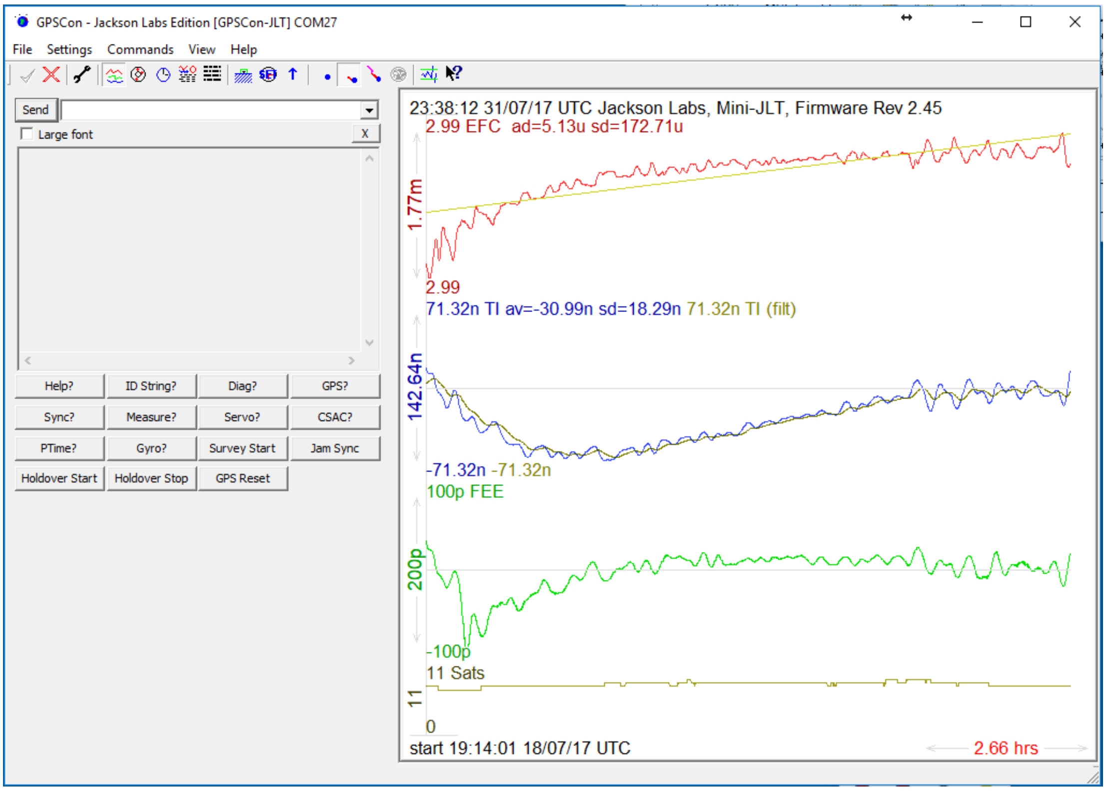


Figure 16 Expanded Graph Display



Exporting the graphics

The settings which control the export function are contained in the “Export graphics” tab in the Options dialog.

Export allows you to create an image file of the graph and/or the satellite trails map. You can select which you want by specifying a file for the Graph path and/or Map path. If you export the graph, you have the option to export only that which is currently visible, or to export the graph which is a plot of the entire logfile contents. Use the checkbox "Export all graph data" to make this choice.

You may select a size of the exported images in X and Y. The file format may be .BMP, .JPG, .GIF, or .PNG. Your settings will be stored and will become the default values the next time you open this dialog.

If you choose to export the graph, you might want to override the TI max setting in force on the screen display. You may do this by entering a non-zero value into the 'Override TI' control. A value of zero causes the export to take the same setting if any as the screen display.

The export may be done automatically on a timed basis. Simply enter a non-zero value in seconds to choose an export time interval. To manually export in accordance with the settings, press the 'Export' button.

Figure 17 Captured Data Example

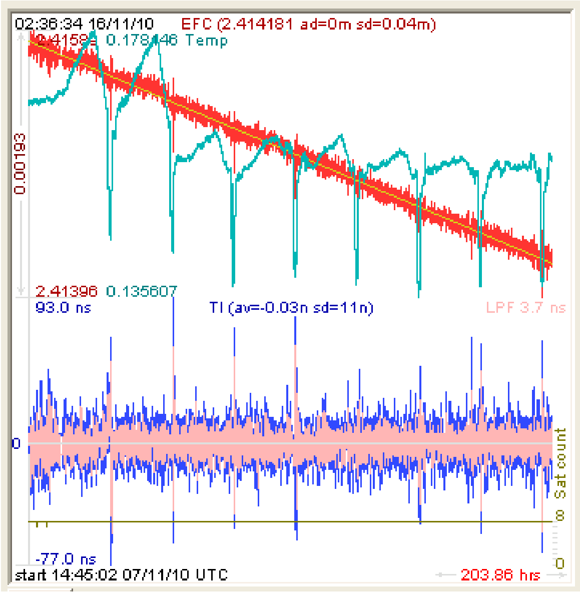
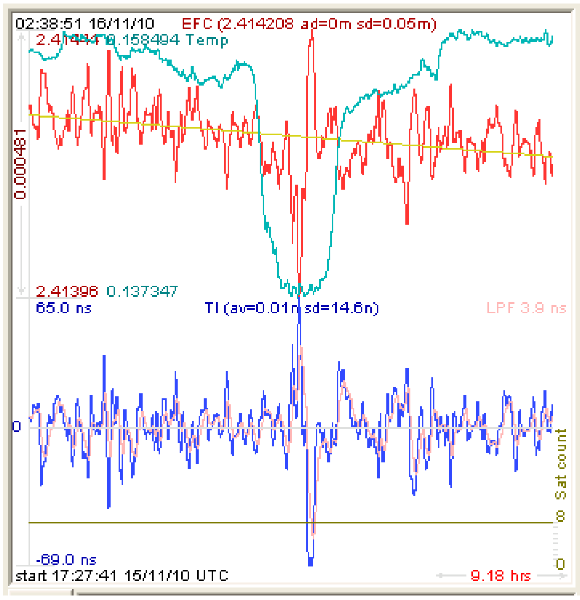


Figure 18 Zoomed Captured Data Example



Interpreting the Data

Figure 17 shows the data acquired by the RSR Transcoder 2.0 unit over a period of more than 200 hours The red trace is EFC (crystal frequency control voltage). The crystal is aging (becoming faster in frequency over time). This requires the control voltage to be lowered to maintain precisely 10.0 MHz. A drift of ~2mV is visible over 200 hours. On the left side of the screen the EFC range over this 200 hour plot is displayed vertically as 0.00193V. This means the drift of the EFC voltage due to aging is ~88mV per year. The EFC sensitivity of the crystal is about 8Hz per volt, so the crystal ages at:

8Hz/V * 0.088V/Year = 0.704Hz/Year drift.

At 10 MHz:

0.704 Hz / 10 MHz = 7.04E-08 aging rate per year.

This is the same as 0.2ppb drift due to aging per day.This crystal aging is fully compensated by the firmware with and without GPS reception of course.

The board temperature is shown in turquoise. We can see it ranges from 0.135607A to 0.178146A. The OCXO current jumps lower every 24 hours because the unit is sitting next to a window, and the sun shines onto the OCXO in the evenings, heating it up, and thus making the unit use lower power during that event.

In Figure 18, which is a zoom of Figure 17 on page 35, we can see the phase offset error of the internal OCXO to the UTC GPS reference. We can see the maximum drift is -77ns to +93ns. The average is (TI av=-0.03ns). The standard deviation over the 200 hour plot is sd=11ns. This means the average error of the 10 MHz phase of this unit over 200 hours is only +/-11ns rms. Or, in other words the average jitter (wander) over 200 hours of operation is:

11ns / 200Hrs = 1.528E-014

or in other words the unit performs as well as a high-quality cesium atomic reference clock over long periods of time. The unit disciplines its internal 10 MHz reference to within less than +/-80ns peak to peak of UTC at all times, which is less than one complete clock cycle at 10 MHz.

4

SCPI Commands

The following topics are discussed in this chapter:

Introduction

The RSR Transcoder 2.0 is typically used in one of three operating modes. This chapter describes the SCPI commands that are used to control and monitor these operating modes. The available modes are:

- Static position simulation with only time/date advancing
- Dynamic position simulation with position, velocity, heading, and time/date advancing based on motion commands stored in internal NV memory
- Real-time PVT transcoding where the RF output simulation follows externally-provided 1Hz PVT NMEA reference strings as well as externally provided 1PPS reference UTC timing pulses

The RSR Transcoder 2.0 includes a USB connector (J2) that provides access to the SCPI (**S**tandard **C**ommands for **P**rogrammable **I**nstrumentation) subsystem through the use of a host system terminal program such as Tera Term or HyperTerminal or a high-level control application such as SimCon or GPSCon. By default the terminal settings are 115200, 8N1, no flow control. The SCPI interface can be used to query the status of the unit, generate NMEA compatible output sentences, or set up and control simulations and transcoding.

More than 150 SCPI commands are supported by the RSR Transcoder 2.0, many of them identical or similar to Symmetricom/Agilent 58503A commands. Additionally, virtually all of the SCPI commands supported by the RSR Transcoder 2.0 are 100% compatible with the CLAW GPS Simulator from VIAVI. To get a listing of the available commands, send the HELP? query. This will return a list of all the available commands and their syntax.

Commands can be entered in either caps or lower-case, and **only the characters shown in capital letters in the command reference below need to be typed in.**

Additional information regarding the SCPI protocol syntax can be found on the following web site:

http://www.ivifoundation.org/scpi/

A basic familiarity with the SCPI protocol is recommended when reading this chapter.

As many setup commands use standard GPS nomenclature, VIAVI recommends review and consideration of the GPS standard specification IS-GPS-200 available here:

http://www.gps.gov/technical/icwg/

As a Quick-Start, the user may want to try sending the following commands to the unit:

- `sim?`
- `syst:stat?`
- `help?`
- `gps?`
- `ptim?`
- `sync?`

General SCPI Commands

*IDN?

This query outputs an identifying string. The response will typically show the following information:

```
<company name> <model name>, <firmware revision> (<FPGA
revision>)
```

HELP?

This query returns a list of the commands available for the RSR Transcoder 2.0.

Simulation Subsystem

The Simulation subsystem groups all of the commands associated with controlling the simulation and transcoding features. The list of commands supported is the following:

- `SIMulation:MODE <AUTO|MANUAL|SIM|TRANSCODE>`
- `SIMulation:STATe?`
- `SIMulation:COMmand <START|STOP>`
- `SIMulation:HOLDover:MODE <OFF|ON|LIMIT>`
- `SIMulation:HOLDover:STATe?`
- `SIMulation:HOLDover:LIMIT <int> [5,86400]`
- `SIMulation:HOLDover:INDicate <ON|OFF>`
- `SIMulation:POSition?`
- `SIMulation:POSition:MODE <FIXed | MOTION>`
- `SIMulation:POSition:LLH <lat>,<lon>,<alt>`
- `SIMulation:POSition:ECEF <x,y,z>`
- `SIMulation:POSition:FILTer?`
- `SIMulation:POSition:FILTer:MODE <OFF | DYNAMIC>`
- `SIMulation:POSition:FILTer:LLH?`
- `SIMulation:POSition:FILTer:ECEF?`
- `SIMulation:POSition:FILTer:VMAX <float> [1.0,600.0]`
- `SIMulation:POSition:FILTer:AMAX <float> [0.1,40.0]`
- `SIMulation:POSition:FILTer:JMAX <float> [0.1,1000]`
- `SIMulation:POSition:MOTION?`
- `SIMulation:POSition:MOTION:WRITE <line>,<command>`
- `SIMulation:POSition:MOTION:READ <line>`

- SIMulation:POSition:MOTION:START <line>
- SIMulation:POSition:MOTION:STOP
- SIMulation:POSition:MOTION:PAUSE
- SIMulation:POSition:MOTION:RESUME
- SIMulation:POSition:MOTION:ZEROize
- SIMulation:TRACe <interval>
- SIMulation:SV?
- SIMulation:SV:VIEW?
- SIMulation:SV:EXCLude <+/-prn>
- SIMulation:SV:PATHloss <ON|OFF>
- SIMulation:SV:ANTenna <UNITY|NORTH|SOUTH|EAST|WEST|PATCH>
- SIMulation:SV:MASK <mask angle>
- SIMulation:SV:HEALTH <sv 1-32> <mask 00-ff> <bits 00-ff>
- SIMulation:SV:ASflag <0 | 1>
- SIMulation:SV:HDOP?
- SIMulation:SV:VDOP?
- SIMulation:SV:TDOP?
- SIMulation:SV:TRACe <prn>
- SIMulation:TIME:MODE <ASSIGNed|CONTinuous|REFerence|TIMer>
- SIMulation:TIME:START:TIME hh,mm,ss.sss
- SIMulation:TIME:START:DATE yyyy,mm,dd
- SIMulation:TIME:LEAPsecond:ACCumulated <seconds>
- SIMulation:TIME:LEAPsecond:DATE yyyy,mm,dd
- SIMulation:TIME:LEAPsecond:DURation <seconds>
- SIMulation:TIME:LEAPsecond?
- SIMulation:TIME:UTCoffset:A0 <a0>
- SIMulation:TIME:UTCoffset:A1 <a1>
- SIMulation:TIME:UTCoffset:DELTATLS <deltatls>
- SIMulation:TIME:UTCoffset:TOT <tot>
- SIMulation:TIME:UTCoffset:WNT <wnt>
- SIMulation:TIME:UTCoffset:WNLSF <wnlsf>
- SIMulation:TIME:UTCoffset:DN <dn>
- SIMulation:TIME:UTCoffset:DELTATLSF <deltatlsf>
- SIMulation:TIME:UTCoffset?
- SIMulation:TIME:CALibrate <float> [-2000000000.0,2000000000.0]
- SIMulation:TIME:FREQuency <float> [-1000.0,1000.0]
- SIMulation:TIME:STEP?
- SIMulation:TIME?
- SIMulation:IONosphere:A <a0>,<a1>,<a2>,<a3>

- `SIMulation:IONosphere:B <b0>,<b1>,<b2>,<b3>`
- `SIMulation:IODCmsb <int> [0,3]`
- `SIMulation:LNAV:IONosphere <subframe> <word> <data>`
- `SIMulation:LNAV:EPHemeris <prn> <subframe> <word> <data>`
- `SIMulation:LNAV:WRITE`
- `SIMulation:LNAV:EEPROM?`
- `SIMulation:LNAV:ACTIVE?`
- `SIMulation:LNAV:SELect <SYNTH | USER | LIVE>`
- `SIMulation:LNAV:HEALTH <0|1>`
- `SIMulation:GPGGA <int> [0,255]`
- `SIMulation:GPRMC <int> [0,255]`
- `SIMulation:NMEA <nmea string>`
- `SIMulation?`

SIMulation:MODE

The SIM:MODE command selects the simulation mode. Available options are Auto, Manual, Simulation, and Transcoding.

This command has the following format:

```
SIMulation:MODE <AUTO|MANUAL|SIM|TRANSCODE>
```

All modes except MANUAL will typically result in an RF output shortly after power up without user intervention. The following table summarizes each mode:

Table 7 Simulation modes

Mode	Description
AUTO	If NMEA data and 1PPS are present from an external reciever at power-up time, then start transcoding. Otherwise, start the configured simulation. This is the default mode.
MANUAL	No simulation from power up. SIM:COMmand must be used to start the simulation
SIM	From power up or if simulation not running, start the configured simulation
TRANSCODE	Wait for NMEA data and 1PPS from an external receiver via the RS-232 port, then start transcoding as NMEA data is being received.

SIMulation:COMmand

To control the RF output state, the SIMulation:COMmand allows both starting and stopping the output. The SIM:COM START command is required to start a simulation and enable RF output in the Manual simulation mode. The SIM:COM STOP command will stop the current simulation and change the simulation mode to Manual.

The command has the following format:

```
SIMulation:COMmand <START|STOP>
```

SIMulation:STATe?

This query responds with the current simulation state. Possible responses include:

```
STOPPED
DETECTING GPS
WAITING GPS FIX
WAITING PPS
STARTING
RUNNING
TRANSCODING
STOPPING
WAITING TIMER
```

The STOPPING and STARTING states each occur for only 0.1 seconds when transitioning to and from RUNNING, so they are unlikely but possible query responses. The RF output is disabled for all states except for RUNNING and TRANSCODING. The RUNNING or TRANSCODING states are indicated by the RF Active LED blinking at 5Hz. The WAITING GPS FIX and WAITING PPS are two states that occur before the TRANSCODING state. The WAITING TIMER state occurs before the RUNNING state when the SIM:TIME:MODE TIMER setting is enabled.

SIMulation:HOLDover:MODE

The simulation holdover mode controls the behavior when the external PVT reference indicates that position information is invalid while operating in Transcoding simulation mode.

The format of this command is:

```
SIMulation:HOLDover:MODE <OFF|ON|LIMIT>
```

When the LIMIT mode is selected, the limit time is specified with the SIM:HOLD:LIMIT command, and the RF output is disabled when the unit reaches the specified holdover time limit described in "SIMulation:HOLDover:LIMIT" on page 43. The SIM:HOLD:STATe? query responds with the current ON/OFF state of the holdover.

SIMulation:HOLDover:STATe?

This command responds with the current ON/OFF state of holdover when operating in transcoding simulation mode. The ON state indicates that the transcoding PVT is based on the local oscillator's holdover performance and/or the INS dead reckoning if operating in the INS simulation position filter mode.

SIMulation:HOLDover:LIMIT

This command specifies the limit in seconds that the holdover state should remain active in the Limit simulation holdover mode. The maximum limit is 1 day or 86400 seconds. Once this holdover limit is reached the RF output will be disabled. For longer holdover periods, use the ON simulation holdover mode.

The format of this command is:

```
SIMulation:HOLDover:LIMIT <int> [5,86400]
```

SIMulation:HOLDover:INDicate

This command selects the holdover indicate mode. With holdover indicate mode ON, the unit decreases the signal strength of the highest elevation satellite by 6dB to indicate that the holdover state is active. An amplitude reduction of 6dB should easily be distinguished as a significant reduction in SNR as reported by the target receiver, while still allowing that signal to be tracked and used in the PVT solution. If the highest elevation satellite changes during holdover then the reduced signal strength satellite will also change.

The format of this command is:

```
SIMulation:HOLDover:INDicate <ON|OFF>
```

SIMulation:POSition:MODE

This command sets the position mode to either the Fixed or Motion mode. The fixed position input is detailed in "SIMulation:POSition:LLH" on page 44 and "SIMulation:POSition:ECEF" on page 44, while the motion commands are detailed starting in "SIMulation:POSition:MOTION:WRITE" on page 47.

The format of this command is:

```
SIMulation:POSition:MODE <FIXed | MOTION>
```

SIMulation:POSition:LLH

This command specifies the current fixed position in Latitude, Longitude and Height (LLH) coordinates. Height is the GPS height or height above the reference ellipsoid and not height above Mean Sea Level (MSL).

The format of this command is:

```
SIMulation:POSition:LLH <lat>,<lon>,<alt>
```

where latitude and longitude are in degrees and altitude is in meters.

The fields are separated by commas. Up to two of the three parameters are optional. Parameters that are not present are not changed. For example, the command:

```
SIM:POS:LLH ,,1000.0
```

sets only the simulated altitude to 1000.0 meters, leaving the latitude and longitude unchanged from their previous settings.

The parameters are immediately applied to the position filter input, but the simulated position may change gradually over time depending upon the SIM:POS:FILTER... settings. Only if the simulation position filter mode is set to OFF will the fixed position be applied immediately as the simulated position. If the simulated position filter mode is set to ON, the position will traverse from the previous simulated position to the new position with the specified dynamics parameters for maximum jerk, maximum acceleration and maximum velocity. See Sections “SIMulation:POSition:FILTer:VMAX” on page 45 to “SIMulation:POSition:FILTer:JMAX” on page 45for more details on the dynamics parameters that affect how the unit simulates a transition from one position to the next.

SIMulation:POSition:ECEF

This command specifies the current fixed position in Earth-Centered-Earth Fixed (ECEF) coordinates. The x,y,z parameters are in meters.

The format of this command is:

```
SIMulation:POSition:ECEF <x,y,z>
```

Unlike the SIM:POS:LLH command, all three x,y,z parameters are required.

SIMulation:POSition:FILTer:MODE

This command selects the position filtering mode that affects how the current input position (fixed position or external PVT source) is applied to the current simulation/ transcoding position.

The format of this command is:

```
SIMulation:POSition:FILTer:MODE <OFF | DYNAMIC>
```

The OFF mode causes the input position to be immediately applied to the simulated position. The DYNAMIC mode moves the simulation position to the input position gradually, limiting velocity, acceleration and jerk to maximum values specified in the VMAX, AMAX and JMAX parameters described below.

The DYNAMIC mode is configured for maximum dynamics using the VMAX, AMAX and JMAX parameters in the commands described in Sections "SIMulation:POSition:FILTer:VMAX" on page 45, "SIMulation:POSition:FILTer:AMAX" on page 45, and "SIMulation:POSition:FILTer:JMAX" on page 45.

SIMulation:POSition:FILTer:LLH?

This query command responds with the current filtered simulation position in LLH format.

SIMulation:POSition:FILTer:ECEF?

This query command responds with the current filtered simulation position in ECEF format.

SIMulation:POSition:FILTer:VMAX

This command specifies the maximum velocity used to configure the position filtering in DYNAMIC and INS position filtering modes. VMAX is then applied to move from the old simulated position to the new simulated position. VMAX is specified in m/s.

The format of this command is:

```
SIMulation:POSition:FILTer:VMAX <float> [1.0,600.0]
```

SIMulation:POSition:FILTer:AMAX

This command specifies the maximum acceleration used to configure the position filtering in DYNAMIC and INS position filtering modes. AMAX is then applied to move from the old simulated position to the new simulated position. AMAX is specified in m/s^2.

The format of this command is:

```
SIMulation:POSition:FILTer:AMAX <float> [0.1,40.0]
```

SIMulation:POSition:FILTer:JMAX

This command specifies the maximum jerk (change in acceleration) used to configure the position filtering in DYNAMIC and INS position filtering modes. JMAX is then applied to move from the old simulated position to the new simulated position. JMAX is specified in m/s^3

The format of this command is:

```
SIMulation:POSition:FILTer:JMAX <float> [0.1,1000]
```

SIMulation:POSition:FILTer:VIN

This command specifies the input velocity filter constant used by the position filter in DYNAMIC position filtering mode. Adjusting this setting can affect the stability of the position filtering and should generally be left at the factory default setting.

The format of this command is:

```
SIMulation:POSition:FILTer:VIN <float> [0.0,1.0]
```

SIMulation:POSition:FILTer:VOUT

This command specifies the output velocity filter constant used by the position filter in DYNAMIC position filtering mode. Adjusting this setting can affect the stability of the position filtering and should generally be left at the factory default setting.

The format of this command is:

```
SIMulation:POSition:FILTer:VOUT <float> [0.0,1.0]
```

SIMulation:POSition:FILTer:POUT

This command specifies the output position filter constant used by the position filter in DYNAMIC position filtering mode. Adjusting this setting can affect the stability of the position filtering and should generally be left at the factory default setting.

The format of this command is:

```
SIMulation:POSition:FILTer:POUT <float> [0.0,1.0]
```

SIMulation:POSition:FILTer:DELay

This command specifies the position filter delay in seconds. Adjusting the delay can calibrate out position error at high velocities due to synchronization errors between the simulation time and position. A positive delay setting will advance the position along the current trajectory. A negative delay setting will set back the position along the current trajectory.

The format of this command is:

```
SIMulation:POSition:FILTer:DELay <float> [-1.0,1.0]
```

SIMulation:POSition:FILTer?

This query responds with the output from the following SIM:POS:FILT... queries:

```
SIM:POS:FILT:LLH?
SIM:POS:FILT:ECEF?
SIM:POS:FILT:MODE?
SIM:POS:FILT:VMAX?
SIM:POS:FILT:AMAX?
SIM:POS:FILT:JMAX?
SIM:POS:FILT:VIN?
SIM:POS:FILT:VOUT?
SIM:POS:FILT:POUT?
SIM:POS:FILT:DELAY?
```

SIMulation:POSition:MOTION:WRITE

This command writes one motion command into non-volatile memory starting at the storage location line number specified. The format of this command is:

```
SIMulation:POSition:MOTION:WRITE <line>,<command>
```

where <line> is the line number and <command> is the command string. Valid starting line numbers are 1 through 100. The final command must always be END which stops the RF output.

When a syntax error occurs with a motion command, a Command Error message is output and that motion command line is ignored. So it is useful to read back the stored commands using the SIM:POS:MOTION:READ command to verify that the correct motion commands have been stored.

SIMulation:POSition:MOTION:READ

This command reads back the motion commands stored starting at the specified storage location line until an END command is reached. The output motion commands follow the motion command syntax in “Simulation Motion Command Language” on page 49, but the format and precision levels of the values originally entered may have changed as the parameters are stored as floating-point numbers and not as the original command text.

The format of this command is:

```
SIMulation:POSition:MOTION:READ <line>
```

where <line> is the motion command storage line to start reading back (1 - 100).

SIMulation:POSition:MOTION:START

This command sets the start position in the motion command storage for playback during a simulation. The simulation will start at the motion command storage line specified by this command and continue until an END command is received or the end of the motion command storage area is reached. Note that in order to play back stored motion commands, the simulation position mode must be set to MOTION with the SIM:POS:MODE command detailed in "SIMulation:POSition:MODE" on page 43.

The format of this command is:

```
SIMulation:POSition:MOTION:START <line>
```

where <line> is the motion command storage line to start the simulation (1 - 100).

SIMulation:POSition:MOTION:STOP

This command stops the current motion command processing and stops the current simulation if in Manual simulation mode.

SIMulation:POSition:MOTION:PAUSE

This command pauses motion command processing and makes the current simulated position the previous simulated position.

SIMulation:POSition:MOTION:RESUME

This command resumes motion command processing after previously being paused with the SIM:POS:MOTION:PAUSE command.

SIMulation:POSition:MOTION:ZEROize

This command erases entire the motion command storage.

SIMulation:POSition:FILTer:SAMPlefactor

This command specifies the factor to reduce the position filter update rate starting with the simulation update rate defined in the SIM:TIME:RATE command. The filter update rate can be calculated as:

```
[filter update rate] = [simulation update rate] / [filter sample factor]
```

Other SIM:POS:FILT… options should be set to appropriate values for the given filter update rate. The sample factor setting is useful if switching between different PVT sources with different message rates that may require different filter update rate for the best performance.

The format of this command is:

```
SIMulation:POSition:FILTer:SAMPlefactor <int> [1,10]
```

Simulation Motion Command Language

The Motion Command Language is designed to define the motion trajectory during the simulation using limits and specified dynamics. The dynamics specified in all commands are also subject to the maximum dynamics of the vehicle specified in the Dynamics (DYN) command. Because the Dynamics command can significantly affect the resulting motion defined by a series of commands, it is highly recommended that the Dynamics command be the first command in a series of Motion Commands. See “Dynamics” on page 49 for a description of the Dynamics command. If no Dynamics command is received, the default dynamics parameters are taken from the filter dynamics parameters defined in the “SIMulation:POSition:FILTer?” commands.

The Reference command specifies the starting location, heading and speed of the simulated motion. Again because this affects the remainder of the simulation, it is highly recommended to include the Reference command as the second command in a series of Motion Commands. See “Reference” on page 50 for a description of the Reference command. If no Reference command is received, the default start position, velocity and heading is the fixed position defined in the SIM:POS:LLH command at zero velocity and heading north (0 degrees).

The commands that define the motion are processed sequentially for the required amount of time to complete the commanded motion. These commands include Straight, Accelerate, Turn, Climb, Waypoint, Combined Accelerate/Turn, Combined Turn/Climb, Goto, Halt, and End. The motion commands are processed until an END command is reached. At this point the simulation stops and the RF output is disabled.

Dynamics

The Dynamics command specifies the linear (in the direction of motion) and lateral (orthogonal to the direction of motion) dynamics limits for all motion. It is recommended that the Dynamics command be the first command in a series of motion commands. The Dynamics command can also be used in the middle of a sequence of Motion Commands to change the vehicle dynamics.

The format of this command is:

```
DYN,<max linear speed>,<max linear accel>,<max linear
jerk>,<max lateral accel>,<max lateral jerk>
```

where

- `<max linear speed>` is in m/s,
- `<max linear accel>` is in m/s^2,

- `<max linear jerk>` is in m/s^3,
- `<max lateral accel>` is in m/s^2, and
- `<max lateral jerk>` is in m/s^3.

Reference

The Reference command specifies the starting position, velocity and heading. It should be used as the second Motion Command after the Dynamics command in a series of commands.

The format of this command is:

```
REF,<latitude>,<longitude>,<altitude>,<heading>,<speed>
```

where

- `<latitude>` is in +/- degrees,
- `<longitude>` is in +/- degrees,
- `<altitude>` is in m above the ellipsoid,
- `<heading>` is in degrees from north, and
- `<speed>` is in m/s.

Straight

The Straight command specifies level (constant altitude) motion at the current heading and speed for the specified duration. The constant heading / great circle option controls if the path over a long distance will maintain a constant heading and spiral towards the poles, or if the path of the largest circle around the earth is followed. With a constant heading, the heading is only reversed 180 degrees when the north or south pole is reached.

The format of this command is:

```
STR,<duration><C|G>
```

where <duration> is in seconds, and C is constant heading and G is great circle.

Accelerate

The Accelerate command specifies a period of level motion at the current heading for the specified duration and positive or negative change in speed.

The format of this command is:

```
ACCEL,<duration>,<change in speed>
```

where `<duration>` is in seconds and `<change in speed>` is in m/s.

Turn

The Turn command specifies a period of level motion at the current speed with the specified heading change and lateral acceleration. The combination of the current speed and lateral acceleration will determine the radius of the curve and the time required to complete the turn.

The format of this command is:

```
TURN,<heading change>,<lateral acceleration>
```

where <heading change> is in degrees and <lateral acceleration> is in g (1 g = 9.81 m/s^2).

Climb

The Climb command specifies a period of increase or decrease in altitude at the current constant heading and speed. The parameters specified include the change in height, height rate, lateral acceleration at the start and lateral acceleration at the end.

The format of this command is:

```
CLIMB,<change in height>,<height rate>,<lateral accel
start>,<lateral accel end>
```

where

- `<change in height>` is in m,
- `<height rate>` is in m/s,
- `<lateral accel start>` is in m/s^2, and
- `<lateral accel end>` is in m/s^2.

Waypoint

The Waypoint command specifies a final position and heading with changes in heading at a specified lateral acceleration at the current speed.

The format of this command is:

```
WAYPT,<latitude>,<longitude>,<altitude>,<final
heading>,<lateral acceleration>
```

where

- `<latitude>` is in degrees,
- `<longitude>` is in degrees,
- `<altitude>` is in m above the ellipsoid,
- `<final heading>` is degrees from north (currently unused),
- `<lateral acceleration>` is in m/s^2 (currently unused).

Combined Accelerate/Turn

The Combined Accelerate/Turn command specifies motion at a constant altitude at the current speeds that is a combination of accelerating linearly and turning. The parameters specified are heading change, lateral acceleration and change in speed. Since with a constant lateral acceleration, the turning radius is affected by the current speed and the current speed is changing with acceleration, the turning radius will also vary. The time required to complete the motion command is determined by the time required to complete the turn.

The format of this command is:

```
ACCEL-TURN,<heading change>,<lateral acceleration>,<change
in speed>
```

where

- <heading change> is in degrees,
- <lateral acceleration> is in g (1 g = 9.81 m/s^2), and
- <change in speed> is in m/s.

Combined Turn/Climb

The Combined Turn/Climb command specifies motion that is a combination of turning and climbing. The parameters specified are heading change, height change, height rate and lateral acceleration.

The format of this command is:

```
TURN-CLIMB,<heading change>,<height change>,<height
rate>,<lateral acceleration>
```

where

- <heading change> is in degrees,
- <height change> is in m,
- <height rate> is in m/s, and
- <lateral acceleration> is in m/s^2.

Halt

The Halt command specifies motion that ends at a specified location. Starting at the current speed and heading, the position changes as quickly as possible to the end location while limited to the maximum dynamics in the Dynamics command.

The format of this command is:

```
HALT,<latitude>,<longitude>,<altitude>
```

where

- <latitude> is in degrees,
- <longitude> is in degrees, and

- `<altitude>` is in m above the ellipsoid.

Goto

The Goto command alters the motion command sequence by executing the command specified by the line number parameter.

The format of this command is:

```
GOTO,<line>
```

This command is useful for creating infinite motion simulations by specifying a previous line number.

End

The End command specifies the end of a sequence of Motion Commands and stops the simulation and RF output.

The format of this command is:

```
END
```

SIMulation:POSition?

The SIM:POS? query combines the query responses from the following queries:

```
SIM:POS:MODE?
SIM:POS:LLH?
SIM:POS:ECEF?
SIM:POS:FILT:LLH?
```

SIMulation:TRACe

This command sets the output rate for the periodically generated statistics output for the simulation. The format of this command is:

```
SIMulation:TRACe <interval>
```

where <interval> is the delay in seconds between simulation trace output in seconds. Zero indicates the simulation trace output is disabled.

The simulation trace output has the form:

```
[yy-mm-dd] [hh:mm:ss.sss] [wn] [tow] [update count] [state]
[sat count]
```

Where

```
[yy-mm-dd] and [hh:mm:ss.sss] are the UTC date and time used
in the simulation,
[wn] and [tow] are the GPS week number and time of week,
[update count] is the 100 Hz simulation update count
starting at 0,
[state] is the simulation state (see table below), and
[sat count] is the number of satellites in the simulation.
```

Below is example output:

```
17-04-27 09:17:35.243 1946 379055.243 6166544 7 10
17-04-27 09:17:36.243 1946 379056.243 6166554 7 10
17-04-27 09:17:37.243 1946 379057.243 6166564 7 10
17-04-27 09:17:38.243 1946 379058.243 6166574 7 10
```

The state number is a numeric representation of the state response from the SIM:STATE? query as defined in the following table:

State	SIM:STATE? output
1	STOPPED
2	DETECTING GPS
4	WAITING GPS FIX
5	WAITING PPS
6	STARTING
7	RUNNING
8	STOPPING
9	TRANSCODING
10	WAITING TIMER

SIMulation:SV:VIEW?

This query responds with a list of the current satellites in view at the GPS RF output with one satellite per line and includes additional statistics. The format of the output is:

```
SV      AZ    EL          RHO    Doppler IODE     TOE
01   258.6  40.7  22011451.0  -1070.07    85  381600
06   127.6  61.0  20849785.2  -1582.25    90  381600
15   343.2  65.0  20704186.0    195.51    99  381600
19   217.7  47.4  21559605.7   2459.13   103  381600
24   117.4  32.3  22646533.1   2050.72   108  381600
```

```
20    46.0   40.6   22029195.3   -2617.34   104   381600
10   314.3   28.0   23029289.3    3055.80    94   381600
```

where SV is the PRN number, AZ is the azimuth in degrees, EL is the elevation in degrees, RHO is the distance in meters, Doppler is the frequency offset from L1 in Hz, IODE is the reference number for the current ephemeris, and TOE is the reference time-of-week in seconds for the current ephemeris.

SIMulation:SV:EXCLude

This command configures specific PRN numbers to be excluded from the simulation. PRNs must be excluded individually with separate SIM:SV:EXCL commands. PRNs can also be individually removed from the exclude list by specifying a negative PRN number. This parameter is stored in NV memory and automatically applied after power-on.

The format of this command is:

```
SIMulation:SV:EXCLude <+/-prn>
```

SIMulation:SV:PATHloss

This command sets enables or disables the attenuation of the satellite signal strength based on pathloss. Satellites farther from the user (i.e. satellites closer to the horizon) will be attenuated more. When pathloss is ON, signal power is reduced as a function of $1/d^2$.

The format of this command is:

```
SIMulation:SV:PATHloss <ON|OFF>
```

SIMulation:SV:ANTenna

This command sets the current selected antenna model. The default antenna model is UNITY. The following table summarizes each option:

Antenna Model	Description
UNITY	Gain of one in all directions
NORTH	North window view: satellites to south attenuated 20 dB
SOUTH	South window view: satellites to north attenuated 20 dB

Antenna Model	Description
EAST	East window view: satellites to west attenuated 20 dB
WEST	West window view: satellites to east attenuated 20 dB
PATCH	Typical patch antenna

The format of this command is:

```
SIMulation:SV:ANTenna <UNITY|NORTH|SOUTH|EAST|WEST|PATCH>
```

SIMulation:SV:MASK

This command sets the minimum elevation angle above the horizon for a satellite to be enabled. The mask angle can be configured with a negative angle, but most receivers will only use satellites above the horizon.

The format of the command is:

```
SIMulation:SV:MASK <mask angle>
```

where <mask angle> is the elevation mask angle in degrees from -90 to 90 degrees. This parameter is stored in NV memory and automatically applied after power-on. The default mask is 10 degrees.

SIMulation:SV:HEALTH

This command causes the health word for the specified satellite to be ANDed with the specified bitmask and OR'ed with the specified word prior to being emitted in the subframe data.

The format of this command is:

```
SIMulation:SV:HEALTH <sv 1-32> <mask 00-ff> <bits 00-ff>
```

The mask and bits parameters should be specified as two-digit hex values. The query SIM:SV:HEALTH? returns the current list of user-specified health values and masks for each satellite.

SIMulation:SV:ASflag

This command sets the Anti-Spoofing flag as broadcast in the satellite navigation data. Some target receivers including most SAASM receivers require the AS flag to be set in order to operate correctly.

The format of this command is:

```
SIMulation:SV:ASflag <0 | 1>
```

SIMulation:SV:HDOP?

This command queries and returns the Horizontal Dillution of Precision (HDOP) provided by the current simulation assuming that all transmitted satellites are received. The reported simulation HDOP may be lower (better) than reported by the target GPS receiver if some satellites are not being used, for example, because of a higher elevation mask angle configured on the target receiver.

SIMulation:SV:VDOP?

This command queries and returns the Vertical Dillution of Precision (VDOP) provided by the current simulation assuming that all transmitted satellites are received. The reported simulation VDOP may be lower (better) than reported by the target GPS receiver if some satellites are not being used, for example, because of a higher elevation mask angle configured on the target receiver.

SIMulation:SV:TDOP?

This command queries and returns the Time Dillution of Precision (TDOP) provided by the current simulation assuming that all transmitted satellites are received. The reported simulation TDOP may be lower (better) than reported by the target GPS receiver if some satellites are not being used, for example, because of a higher elevation mask angle configured on the target receiver.

SIMulation:SV:TRACe

This command enables the trace output for an individual satellite at a 100 Hz serial output rate. Only one satellite can be enabled at one time and setting the selected PRN to 0 will disable the output. Also, the satellite must be in view for the output to be enabled. Use the SIM:SV:VIEW? query to determine which satellites are in view.

The format of this command is:

```
SIMulation:SV:TRACe <prn>
```

where <prn> is the satellite PRN number 1 to 32 or 0 for none. This parameter is stored in NV memory and automatically applied after power-on.

SIMulation:SV?

This query command will output the query results from the following queries:

```
SIM:SV:EXCL?
SIM:SV:MASK?
SIM:SV:TRAC?
```

SIMulation:TIME:MODE

This command sets the simulation Time mode of the when running in Simulation mode. When using the ASSIGNED time mode, the simulation will always start with the time and date assigned with the SIM:TIME:START... commands.

When using the CONTINUOUS time mode, the simulation will use the current time provided by an external GNSS receiver or UTC time manually set with the GPS:INIT:TIME and GPS:INIT:DATE commands. If the current time has not been set, then the time will be set to the value set by the SIM:TIME:START... commands at the beginning of first simulation run. If the simulation is stopped and started again the selected set time will be propagated by the local clock.

When using the REFERENCE time mode, the simulation will use the time provided by the external GPS reference.

When using the TIMER mode, the simulation will delay start until the current system date and time as reported by the PTIME? response matches the start date and time defined by the SIM:TIME:START:TIME and SIM:TIME:START:DATE commands described below. The simulation state as reported by the SIM:STATE? response will be WAITING TIMER before the system date/time and simulation state date/time match. The precision of the simulation start is within +/- 100 ms.

The format of this command is:

```
SIMulation:TIME:MODE <ASSIGNed|CONTinuous|REFerence|TIMer>
```

This parameter is stored in NV memory and automatically applied after power-on. Changing the configured time mode is only allowed when the simulation is stopped.

SIMulation:TIME:START:TIME

This command sets the fixed UTC start time used in the simulation when the simulation Time mode is set to Fixed. The GPS receiver time will reflect this value when the simulation begins. This parameter is stored in NV memory and automatically applied after power-on.

The format of this command is:

```
SIMulation:TIME:START:TIME hh,mm,ss.sss
```

SIMulation:TIME:START:DATE

This command sets the fixed start date of the simulation when in the simulation Time mode is set to Fixed. The GPS receiver will indicate the this time when the simulation begins. This parameter is stored in NV memory and automatically applied after power-on.

The format of this command is:

```
SIMulation:TIME:START:DATE yyyy,mm,dd
```

SIMulation:TIME:STEP?

This command queries the time step in milliseconds between simulatino updates. The simulation update rate is not configurable and depends upon the firmware version loaded.

The format of this command is:

```
SIMulation:TIME:STEP?
```

SIMulation:TIME:LEAPsecond:ACCumulated

This command sets the current UTC to GPS time offset in seconds and provides a simplified method of setting the UTC correction parameters in the SIM:TIME:UTC... commands. This parameter is stored in NV memory and automatically applied after power-on.

When using Transcoding simulation mode or Reference simulation time mode and the time reference is provided by an external ublox GNSS receiver, the SIMulation:TIME:LEAPsecond parameters are automatically updated when the current and pending leap second data is available. This allows the target GPS receiver to maintain the correct UTC time during a leap second event without manually entering the leapsecond data with the SCPI interface.

The format of this command is:

```
SIMulation:TIME:LEAPsecond:ACCumulated <seconds>
```

SIMulation:TIME:LEAPsecond:DATE

This command sets the date of the next or previous pending leap second. If the date is after the simulation time, then a leap second is pending and will be simulated. This command provides a simplified method of setting the UTC correction parameters in the SIM:TIME:UTC... commands. This parameter is stored in NV memory and automatically applied after power-on.

The format of this command is:

```
SIMulation:TIME:LEAPsecond:DATE yyyy,mm,dd
```

SIMulation:TIME:LEAPsecond:DURation

This command sets the duration of a pending leap second event. Valid values are 59, 60 and 61. If no leap second is pending, the value should be set to 60. This command provides a simplified method of setting the UTC correction parameters in the SIM:TIME:UTC... commands. This parameter is stored in NV memory and automatically applied after power-on.

The format of this command is:

```
SIMulation:TIME:LEAPsecond:DURation <seconds>
```

SIMulation:TIME:LEAPsecond?

This query command will output the query results from the following queries:

```
SIM:TIME:LEAP:ACC?
SIM:TIME:LEAP:DATE?
SIM:TIME:LEAP:DUR?
```

SIMulation:TIME:UTCoffset:A0

This command specifies the UTC offset parameter A0. This parameter is stored in NV memory and automatically applied after power-on. The parameter is defined in the IS-GPS-200 specification.

The format of this command is:

```
SIMulation:TIME:UTCoffset:A0 <a0>
```

SIMulation:TIME:UTCoffset:A1

This command specifies the UTC offset parameter A1. This parameter is stored in NV memory and automatically applied after power-on. The parameter is defined in the IS-GPS-200 specification.

The format of this command is:

```
SIMulation:TIME:UTCoffset:A1 <a1>
```

SIMulation:TIME:UTCoffset:DELTATLS

This command specifies the UTC offset parameter DELTATLS. This parameter is stored in NV memory and automatically applied after power-on. The parameter is defined in the IS-GPS-200 specification.

The format of this command is:

```
SIMulation:TIME:UTCoffset:DELTATLS <deltatls>
```

SIMulation:TIME:UTCoffset:TOT

This command specifies the UTC offset parameter TOT. This parameter is stored in NV memory and automatically applied after power-on. The parameter is defined in the IS-GPS-200 specification.

The format of this command is:

```
SIMulation:TIME:UTCoffset:TOT <tot>
```

SIMulation:TIME:UTCoffset:WNT

This command specifies the UTC offset parameter WNT. This parameter is stored in NV memory and automatically applied after power-on. The parameter is defined in the IS-GPS-200 specification.

The format of this command is:

```
SIMulation:TIME:UTCoffset:WNT <wnt>
```

where wnt is a GPS week number (not mod 256 or 1024).

SIMulation:TIME:UTCoffset:WNLSF

This command specifies the UTC offset parameter WNLSF. This parameter is stored in NV memory and automatically applied after power-on. The parameter is defined in the IS-GPS-200 specification.

The format of this command is:

```
SIMulation:TIME:UTCoffset:WNLSF <wnlsf>
```

where wnt is a GPS week number (not mod 256 or 1024).

SIMulation:TIME:UTCoffset:DN

This command specifies the UTC offset parameter DN. This parameter is stored in NV memory and automatically applied after power-on. The parameter is defined in the IS-GPS-200 specification.

The format of this command is:

```
SIMulation:TIME:UTCoffset:DN <dn>
```

SIMulation:TIME:UTCoffset:DELTATLSF

This command specifies the UTC offset parameter DELTATLSF. This parameter is stored in NV memory and automatically applied after power-on. The parameter is defined in the IS-GPS-200 specification.

The format of this command is:

```
SIMulation:TIME:UTCoffset:DELTATLSF <deltatlsf>
```

SIMulation:TIME:UTCoffset?

This query command returns the response of the following individual queries:

```
SIM:TIME:UTC:A0?
SIM:TIME:UTC:A0?
SIM:TIME:UTC:DELTATLS?
SIM:TIME:UTC:TOT?
SIM:TIME:UTC:WNT?
SIM:TIME:UTC:DN?
SIM:TIME:UTC:DELTATLSF?
```

SIMulation:TIME:CALibrate

This command sets a time offset in nanoseconds for the timing simulation. Normally this offset can be set to zero, but nonzero values can be used to compensate for constant timing offsets in the external GPS reference or to precisely calibrate internal hardware delays. A positive time offset results in moving the simulation time forward, while a negative time offset delays simulation time.

The format of this command is:

```
SIMulation:TIME:CALibrate <float> [-
2000000000.0,2000000000.0]
```

SIMulation:TIME:FREQuency

This command sets a frequency offset in parts per billion for the timing simulation. Normally this offset should be set to zero, but nonzero values can be used for some simulation tests to simulate a jump in frequency in the target receiver's oscillator, for example.

The format of this command is:

```
SIMulation:TIME:FREQuency <float> [-1000.0,1000.0]
```

SIMulation:TIME?

This query command returns the response of the following individual queries:

```
SIM:TIME:MODE?
SIM:TIME:START:TIME?
SIM:TIME:START:DATE?
```

SIMulation:IONosphere:A

This command configures the a0 through a3 ionospheric parameters that are part of the GPS navigation message. This parameter is stored in NV memory and automatically applied after power-on. The parameter is defined in the IS-GPS-200 specification.

The format of this command is:

```
SIMulation:IONosphere:A <a0>,<a1>,<a2>,<a3>
```

SIMulation:IONosphere:B

This command configures the b0 through b3 ionospheric parameters that are part of the GPS navigation message. This parameter is stored in NV memory and automatically applied after power-on. The parameter is defined in the IS-GPS-200 specification.

The format of this command is:

```
SIMulation:IONosphere:B <b0>,<b1>,<b2>,<b3>
```

SIMulation:IODCmsb

This command sets the 2 Most Significant Bits (MSBs) of the 10-bit IODC field of the GPS navigation data. The remaining 8 Least Significant Bits (LSBs) are set equal to the IODE field for each satellite ephemeris.

The format of this command is:

```
SIMulation:IODCmsb <int> [0,3]
```

SIMulation:LNAV:IONosphere

The SIM:LNAV:IONosphere command supports the transmission of user-specified ionospheric/UTC correction parameters to the simulator for storage in nonvolatile memory.

The format of this command is:

```
SIMulation:LNAV:IONosphere <subframe> <word> <data>
```

The following describes the parameters accepted by SIM:LNAV:IONosphere:

- **<subframe>** The GPS LNAV subframe number to be addressed by the command. Subframe numbers range from 1 to 5, inclusive, corresponding to the subframes described in IS-GPS-200H Appendix II ("GPS NAVIGATION DATA STRUCTURE FOR LNAV DATA"). Currently the supported only subframe number is 4.
- **<word>** The starting word position within the subframe at which the <data> will be written. Supported word positions range from 2 through 9 inclusive, based on an initial index of zero.

- **<data>** A series of consecutive 24-bit hexadecimal values to be written to the subframe beginning at the specified <word>. Each 24-bit value consists of six ASCII hex characters (nibbles). No spaces or other characters should appear anywhere within the <data> string. Up to eight 24-bit values may be specified as long as the final <word> position does not exceed the maximum index (9).

An example of this command is:

```
SIM:LNAV:ION 4 2
780502FFFF2603FFFB000003000000034EA8120000000000
```

SIMulation:LNAV:EPHemeris

The SIM:LNAV:EPHemeris command supports the transmission of user-specified ephemeris parameters to the simulator for storage in nonvolatile memory.

The format of this command is

```
SIMulation:LNAV:EPHemeris <prn> <subframe> <word> <data>
```

The following describes the parameters accepted by SIM:LNAV:ION:

- <**prn**> The destination satellite identified by its PRN number (1-32 inclusive).
- <**subframe**> The GPS LNAV subframe number to be addressed by the command. Subframe numbers range from 1 to 5, inclusive, corresponding to the subframes described in IS-GPS-200H Appendix II ("GPS NAVIGATION DATA STRUCTURE FOR LNAV DATA"). Supported subframes range from 1 to 3 inclusive.
- <**word**> The starting word position within the subframe at which the <data> will be written. Supported word positions for subframes 2 and 3 range from 2 through 9 inclusive, based on an initial index of zero. Supported word positions for subframe 1 are 2, 6, 7, 8, and 9.
- <**data**> A series of consecutive 24-bit hexadecimal values to be written to the subframe beginning at the specified <word>. Each 24-bit value consists of six ASCII hex characters (nibbles). No spaces or other characters should appear anywhere within the <data> string. Up to eight 24-bit values may be specified as long as the final <word> position does not exceed the maximum index (9).

Data written by SIM:LNAV:EPHemeris is stored in a temporary RAM buffer on the simulator platform. Once a complete set of ephemerides have been transmitted to the simulator, they must be committed to EEPROM storage by issuing a SIM:LNAV:WRITE command.

An example of this command is:

```
SIM:LNAV:EPH 1 1 2 EA1000
SIM:LNAV:EPH 1 1 6 00000C0F2A3000FFFD076908
SIM:LNAV:EPH 1 2 2
0FFE193088C04D8C00FE13038C40781046A10D74D62A3000
```

```
SIM:LNAV:EPH 1 3 2
002B6B8EB403FFD7277453311CDD18355339FFA9590F1238
```

SIMulation:LNAV:WRITE

The SIM:LNAV:WRITE command must be issued to store the ionospheric/UTC parameters and ephemerides most recently uploaded by SIM:LNAV:ION and SIM:LNAV:EPH to nonvolatile (EEPROM) memory. After issuing SIM:LNAV:WRITE, the command SIM:LNAV:SELECT USER may be transmitted to select the uploaded navigation data for simulation.

SIMulation:LNAV:EEPROM?

The SIM:LNAV? query returns a list of all ionospheric/UTC and ephemeris parameters stored in EEPROM. The 24-bit words are returned one at a time in the form of individual SIM:LNAV:EPH and SIM:LNAV:ION commands.

SIMulation:LNAV:ACTIVE?

The SIM:LNAV? query returns a list of all ionospheric/UTC and ephemeris parameters actively used in the current simulation. The 24-bit words are returned one at a time in the form of individual SIM:LNAV:EPH and SIM:LNAV:ION commands.

SIMulation:LNAV:SELect

The SIM:LNAV:SELect command determines whether internally-generated navigation data or user-uploaded navigation data is used for simulation. The format of this command is:

```
SIMulation:LNAV:SELect <SYNTH | USER | LIVE>
```

`SYNTH` mode selects the internally generated synthesized constellation with circular orbits and precisely 12 hour orbit periods. This is the default mode.

`USER` mode selects the user-uploaded ephemeris and ionospheric/UTC parameters.

`LIVE` mode selects ephemeris, almanac and ionospheric/UTC parameters received from the external GPS receiver in Novatel binary command format.

SIMulation:LNAV:HEALTH

This query returns the current health override flag value. 0 is the default, indicating that the health words are modified with the bitmasks and values provided to the SIMulation:SV:HEALTH command, while a value of 1 causes the ephemeris and almanac health values to be used without modification.

The format of this command is:

```
SIMulation:LNAV:HEALTH <0|1>
```

SIMulation:GPGGA

This command instructs the RSR Transcoder 2.0 to send the NMEA standard string $GPGGA every N seconds, with N in the interval [0,255]. The output GPGGA sentence uses the simulated time and position data and is useful for comparison with PVT data output from the target GPS receiver. The GPGGA output is only enabled when the simulation state is RUNNING or TRANSCODING.

This command has the following format:

```
SIMulation:GPGGA <int> [0,255]
```

GPGGA shows height in MSL Meters; this is different from traditional GPS receivers that display height in GPS Meters. The difference between MSL and GPS height can be significant, with 35m or larger discrepancies being common.

SIMulation:GPRMC

This command instructs the RSR Transcoder 2.0 to send the NMEA standard string $GPRMC every N seconds, with N in the interval [0,255]. The output GPRMC sentence uses the simulated time and position data and is useful for comparison with PVT data output from the target GPS receiver. The GPRMC output is only enabled when the simulation state is RUNNING or TRANSCODING.

This command has the following format:

```
SIMulation:GPRMC   <int>  [0,255]
```

SIMulation:NMEA <nmea string>

This command is used to accept NMEA data over the SCPI port. For this command to work, the GPS:PORT setting must be SCPI as described in “GPS:PORT <SCPI | RS232>” on page 89.

SIMulation:TIME:RATE

This command sets the update rate in Hz for the simulation. The default simulation update rate setting is 10Hz. Higher update rates are required for higher dynamic solutions, while lower simulation rates (i.e. 10 Hz) may be needed for compatibility with the original Transcoder.

This command has the following format:

```
SIMulation:TIME:RATE <10|20|50|100>
```

> **NOTE**
> The 50 Hz and 100 Hz options are only available when the 100 Hz option is installed.

This parameter is stored in NV memory and is automatically applied after power-on. Changing the configured simulation rate is only allowed when the simulation is stopped. Changing the update rate also results in a software reset after the setting is stored so that the setting can take effect. The reset behavior is similar to that of the SYSTem:CPURESET command described in "SYSTem:CPURESET" on page 102.

SIMulation:ION:L2MODEL

This command enables and disables the ionospheric correction model difference between L1 and L2 frequencies that many dual frequency receivers user to directly measure the ionospheric corrections. The default setting is OFF/

This command has the following format:

```
SIMulation:ION:L2MODEL <ON|OFF>
```

This parameter is stored in NV memory and is automatically applied after power-on.

SIMulation?

This query responds with the output from the following SIM... queries:

```
SIM:MODE?
SIM:STATE?
SIM:HOLD:MODE?
SIM:HOLD:STATE?
SIM:HOLD:LIMIT?
SIM:HOLD:IND?
SIM:TRACe?
```

This query also returns the firmware version, FPGA version, and FPGA diagnostics code.

Output Subsystem

The commands in the OUTPUT subsystem control various features of the GPS RF and 10 MHz output, as well as the built-in RF synthesizer functionality. The list of commands supported is the following:

```
OUTput:TEST <OFF | SWEEP | TONE | TWOTONE | RAND>
```

```
OUTput:DCBlock
OUTput:POWer <power> dBm
OUTput:OFFset <offset> dBm
OUTput:10Mhz <ON | OFF>
OUTput?
```

OUTput:TEST

This command enables the test signal mode on the SMA RF output that can be used for calibration, spectrum analyzer tests, RF front-end testing or as a jamming test signal. The signal amplitude of the test signal can typically be set as high as -85 dBm. The RF power level is set with the OUT:POWer command.

The format of this command is:

```
OUTput:TEST <OFF | SWEEP | TONE | TWOTONE | RAND>
```

The following table describes each of the test modes:

Antenna Model	Description
OFF	Default setting for normal GPS signal output
SWEEP	Single CW tone, swept +/- 1.023 MHz either side of 1,575.42 MHz over approximately 30 seconds
TONE	Single CW tone at 1,575.42 MHz
TWOTONE	Two CW tones spaced approximately +/- 50 kHz either side of 1,575.42 MHz. Each tone has 6 dB lower amplitude than the OUT:POWER setting.
RAND	Baseband (complex) white noise generated by two independent maximal-length shift registers with cycle times of approximately 30 minutes

OUTput:DCBlock

This command is not supported on the RSR Transcode 2.0. The RF output is always DC-coupled and has a 186-ohm load resistor to ground to simulate an industry-standard GPS antenna load for the target receiver. To avoid damage to the RSR Transcoder 2.0’s internal load resistor, an external DC block rated for operation at 1.5 GHz *must* be used in line with the antenna connection when the receiver supplies more than 6V at its antenna port.

OUTput:POWer?

This command sets the power level at the RF output jack for the L1 C/A code signal. The L1 P-code and L2 P-code levels are a fixed 3 dB and 6 dB lower than the L1 C/A code signal level. The valid range is -90 dBm to -130 dBm (preliminary specification) plus the current OUT:OFFset setting. For example, if an external 10 dB attenuator is attached to the RF output and OUT:OFFset is set to -10 dB, then the valid range for the OUTput:POWer? command is -100 to -140 dBm. Similarly, if a 10 dB amplifier is connected to the RF output, and the OUT:OFFset is set to +10 dB then the valid range for OUTput:POWer command is -80 to -120 dBm.

The OUT:POWer command sets the simulator's RF output power from the perspective of a single satellite, not the entire constellation. When N satellites are transmitting, the total RF output power will be approximately $10*\log_{10}N$ dB greater than specified.

The nominal output power of a live GPS satellite signal at the surface of the earth is typically about -125 dBm, but is often a few dB higher to allow for a reduction in power as the satellite ages. Additionally, target GPS receivers often expect to be used with an amplified antenna, so the output power is typically set higher than -130 dBm for optimum performance of the target GPS receiver.

The format of this command is:

```
OUTput:POWer <power> dBm
```

This parameter is stored in NV memory and automatically applied after power-on. The dBm nomenclature is optional.

OUTput:OFFset

This command sets the output power offset to compensate for an external attenuator (negative offset) or amplifier (positive offset). The offset value adjusts the value provided to the OUT:POWer command so that the true output power from the RF signal chain including the attenuator or amplifier matches that of the OUT:POWer setting. The format of this command is:

```
OUTput:OFFset <offset> dBm
```

The dBm nomenclature is optional.

The offset also affects the valid range of the OUTput:POWer command setting.

OUTput:CAOFFset

This command sets the L1 C/A code signal level offset in dB from the level configured by the OUTput:POWer command. For example, with an OUTput:POWer setting of -120 dBm and OUTput:CAOFFset setting of -3 dB, the effective L1 C/A code output power would be -123 dBm.

This command has the following format:

```
OUTput:CAOFFset <float> [-300,300]
```

This parameter is stored in NV memory and is automatically applied after power-on.

OUTput:L1POFFset

This command sets the L1 P code signal level offset in dB that is normally -3 dB relative to the L1 C/A code level configured by the OUTput:POWer command. For example, with an OUTput:POWer setting of -120 dBm and OUTput:L1POFFset setting of +3 dB, the effective L1 P code output power would be -120 dBm.

This command has the following format:

```
OUTput:L1POFFset <float> [-300,300]
```

This parameter is stored in NV memory and is automatically applied after power-on.

OUTput:L2POFFset

This command sets the L2 P code signal level offset in dB that is normally -6 dB relative to the L1 C/A code level configured by the OUTput:POWer command. For example, with an OUTput:POWer setting of -120 dBm and OUTput:L1POFFset setting of +3 dB, the effective L1 P code output power would be -123 dBm.

This command has the following format:

```
OUTput:L2POFFset <float> [-300,300]
```

This parameter is stored in NV memory and is automatically applied after power-on.

OUTput:L1P

This command enables/disables the L1 P code signal.

This command has the following format:

```
OUTput:L1P <ON|OFF>
```

This parameter is stored in NV memory and is automatically applied after power-on.

OUTput:L2P

This command enables/disables the L2 P code signal.

This command has the following format:

```
OUTput:L2P <ON|OFF>
```

This parameter is stored in NV memory and is automatically applied after power-on.

OUTput:10Mhz

This command enables/disables the buffered 10 MHz internal oscillator CMOS output on J31 pin 12. By default the output is disabled to reduce noise and power consumption and must be enabled with the ON setting. The format of this command is:

```
OUTput:10Mhz <ON | OFF>
```

This parameter is stored in NV memory and automatically applied after power-on.

OUTput?

This query command outputs the query responses from the following separate queries:

```
OUT:POWer?
OUT:OFFset?
OUT:10MHz?
```

Calibration Subsystem

The Calibration Subsystem is used for factory calibration of the output power and should not be user-modified. Please contact VIAVI for information on recalibrating the output power.

SYNChronization Subsystem

This subsystem groups the commands related to the phase and frequency synchronization of the RSR Transcoder 2.0 with the GPS receiver. The list of the commands supported for this subsystem is the following:

- `SYNChronization:SOURce:MODE <GPS | EXTernal | NMEA | HOLDover>`
- `SYNChronization:SOURce:STATE?`
- `SYNChronization:HOLDover:DURation?`
- `SYNChronization:HOLDover:STATe?`
- `SYNChronization:HOLDover:INITiate`
- `SYNChronization:HOLDover:RECovery:INITiate`
- `SYNChronization:OUTput:1PPS:RESET [ON|OFF]`
- `SYNChronization:TINTerval?`
- `SYNChronization:TINTerval:THReshold [50,2000]`
- `SYNChronization:IMMEdiate`
- `SYNChronization:FEEstimate?`
- `SYNChronization:LOCKed?`

- `SYNChronization:HEAlth?`
- `SYNChronization?`

SYNChronization:HOLDover:STATe?

This query returns the current holdover state with possible responses being NONE, MANUAL, or ON. NONE indicates the holdover state is off. MANUAL indicates the holdover is manually enabled with the SYNC:HOLD:INIT command. ON indicates that holdover is enabled due to lack of external GNSS receiver 1PPS or valid GNSS fix.

SYNChronization:HOLDover:DURation?

This query returns the duration of the present or most recent period of operation in the holdover. This is the length of time the reference oscillator was not locked to the external GNSS receiver 1PPS, and is thus “coasting”. The first number in the response is the holdover duration. The duration units are seconds, and the resolution is 1 second. If the receiver is in holdover, the response quantifies the current holdover duration. If the receiver is not in holdover, the response quantifies the length of the previous holdover. The second number in the response identifies the holdover state. A value of 0 indicates that the receiver is not in holdover; a value of 1 indicates that the Receiver is in holdover.

SYNChronization:HOLDover:INITiate

The SYNC:HOLD:INIT and SYNC:HOLD:REC:INIT commands allow the user to manually enter and exit the holdover state, even while GNSS signals are still being properly received. This forced-holdover allows the unit to effectively disable GNSS disciplining of the internal oscillator, while still keeping track of the state of the 1PPS output in relation to the UTC 1PPS signal as generated by the GNSS receiver. When the unit is placed into forced-holdover with this command, the unit will indicate the time interval difference between the 1PPS output and the GNSS receiver 1PPS by using the SYNC:TINT? command up to a limit of +/-2000ns. This allows the user to see the oscillator drift when not locked to GNSS signals for testing purposes, or to prevent the GNSS receiver from being spoofed and affecting the oscillator frequency accuracy. All other frequency-disciplining functions of the unit will behave as if the GNSS antenna was disconnected from the unit while in this forced-holdover state.

SYNChronization:HOLDover:RECovery:INITiate

This command will disable the forced holdover state (see the SYNC:HOLD:INIT command). The unit will resume normal GNSS disciplining operation after this command has been sent.

SYNChronization:SOURce:MODE

This command selects between the EXTernal, GPS, NMEA and HOLDover sync modes. Both the EXTernal and GPS modes use the same 1PPS input for disciplining the oscillator and synchronizing precise timing, but the GPS mode also gates the 1PPS based on the GNSS status of the external receiver as indicated in the NMEA serial sentences from the GNSS receiver. The External mode uses the 1PPS reference input to synchronize precise timing and discipline the oscillator regardless of the GNSS status.

The NMEA sync mode uses the NMEA input to synchronize the system time to millisecond-level precision. The NMEA sync mode is required to transcode in the absence of an externally-supplied 1PPS reference signal.

The Holdover sync mode uses the 1PPS reference input only to synchronize the system time to nanosecond-level precision, but does not discipline the oscillator. The Holdover sync mode is useful when the 1PPS available is not stable enough discipline the oscillator or be used as a continuous timing reference for simulation.

The format of this command is:

```
SYNChronization:SOURce:MODE <GPS | EXTernal | NMEA |
HOLDover>
```

SYNChronization:SOURce:STATE?

This command is implemented, but always reports the same state as the SYNC:SOUR:MODE command setting in the RSR Transcoder 2.0.

SYNChronization:TINTerval?

This query returns the difference or timing shift between the RSR Transcoder 2.0 internal oscillator 1PPS phase and the externally-supplied 1PPS reference phase. The resolution is 1E-010 seconds.

SYNChronization:TINTerval:THReshold [50,2000]

This command selects the oscillator 1PPS phase-offset threshold as compared to the reference 1PPS at which point the unit will initiate a counter-reset (jam-sync) aligning the oscillator generated 1PPS with the reference 1PPS phase. The oscillator phase is slowly and continuously adjusted toward 0ns offset to the externally-supplied reference 1PPS phase as long as the phase difference is less than the THReshold phase limit. The oscillator generated 1PPS phase is allowed to drift up to this threshold before a jam-sync is initiated. The default setting is 220ns, allowing a drift of up to +/-220ns. Reaching this selected threshold will cause a jam-sync phase-normalization to be initiated, which will reset the counter phase that generates the internal 1PPS output. It will also cause the SYNC:HEAlth? Status to indicate 0x200, and the lock status to become “unlocked”. By selecting a larger phase window the user can prevent these phase resets (jam-syncs) of the output 1PPS from happening too frequently in environments where airflow or thermal changes are expected, especially on units without the CSAC option.

SYNChronization:IMMEdiate

This command initiates a near-instantaneous alignment of the GNSS 1PPS and Receiver output 1PPS phases. To be effective, this command has to be issued while the unit is not in holdover.

SYNChronization:FEEstimate?

This query returns the Frequency Error Estimate, similar to the Allan Variance using a 1000s measurement interval, by comparing the internal 1PPS to GNSS 1PPS phase offset.

Values less than 1E-012 are below the noise floor, and are not significant.

SYNChronization:LOCKed?

This query returns the lock state (0=OFF, 1=ON) of the PLL controlling the TCXO.

SYNChronization:OUTput:1PPS:RESET [ON|OFF]

This command allows the generation of the 1PPS output pulse upon power-on without an external GNSS receiver being connected to the unit. By default the unit does not generate a 1PPS pulse until the GNSS receiver has locked onto the satellites. With the command SYNC:OUT:1PPS:RESET ON the unit can be configured to generate an asynchronous 1PPS output after power-on even if a GNSS antenna is not connected to the unit. The ON setting will also allow the NMEA output strings to be generated in the absence of an externally-provided 1PPS reference input. Once the GNSS receiver locks, the 1PPS pulse will align itself to UTC by stepping in 10 equally spaced steps toward UTC alignment. The default setting is OFF which means the 1PPS pulse is disabled until proper external reference GNSS lock is achieved. Changing this setting requires a reset or power-cycle for the new setting to become effective.

SYNChronization:HEAlth?

The SYNChronization:HEAlth? query returns a hexadecimal number indicating the system's health status. Error flags are encoded in a binary fashion so that each flag occupies one single bit of the binary equivalent of the hexadecimal health status flag. This allows for easy query and fully-orthogonal status indications using only one single hex parameter.

The following system parameters are monitored and indicated through the health-status indicator. Individual parameters are ‘ored’ together which results in a single hexadecimal value encoding the following system status information:

If the TCXO coarse-DAC is maxed-out at 255	HEALTH STATUS |= 0x1;
If the TCXO coarse-DAC is mined-out at 0	HEALTH STATUS |= 0x2;
If the phase offset to UTC is >250ns	HEALTH STATUS |= 0x4;
If the run-time is < 300 seconds	HEALTH STATUS |= 0x8;
If the GNSS receiver is in holdover > 60s	HEALTH STATUS |= 0x10;
If the Frequency Estimate is out of bounds	HEALTH STATUS |= 0x20;
If the supply voltage is too high	HEALTH STATUS |= 0x40;
If the supply voltage is too low	HEALTH STATUS |= 0x80;
If the short-term-drift (ADEV @ 100s) > 100ns	HEALTH STATUS |= 0x100;
For the first 3 minutes after a phase reset, or a coarse DAC change:	HEALTH STATUS |= 0x200;
If the GNSS receiver indicates a strong jamming signal of >=50 (range is 0 to 255) (supported only on uBlox external receivers)	HEALTH STATUS |= 0x800;
If FPGA error occurs	HEALTH STATUS |= 0x2000;
If simulation update timeouts occur	HEALTH STATUS |= 0x4000;
If internal GPS receiver C/No values out of range (>2dB error)	HEALTH STATUS |= 0x8000;
If internal GPS receiver position out of range (>10 meters)	HEALTH STATUS |= 0x10000;
If internal GPS receiver 1PPS timing out of range (>100 ns)	HEALTH STATUS |= 0x20000;
If internal GPS receiver has not achieved lock after 100 seconds	HEALTH STATUS |= 0x40000;

As an example, if the unit is in GNSS receiver holdover, and the voltage is too high, and the UTC phase offset is > 250ns then the following errors would be indicated:

1) UTC phase > 250ns: 0x4
2) Voltage too high: 0x40
3) GNSS receiver in holdover: 0x10

‘Oring’ these values together results in:

0x40 | 0x10 | 0x4 = 0x54

The unit would thus indicate: HEALTH STATUS: 0x54

A health status of 0x0 indicates a properly locked, and warmed-up unit that is completely healthy and has no flagged events.

SYNChronization?

This query returns the results of these four queries:

```
SYNChronization:SOURce:MODE?
SYNChronization:SOURce:STATE?
SYNChronization:LOCKed?
SYNChronization:HOLDover:DURation?
SYNChronization:health?
```

DIAGnostic Subsystem

This subsystem groups the queries related to the diagnostic of the internal oscillator. The list of the commands supported for this subsystem is as follows:

```
DIAGnostic:ROSCillator:EFControl:RELative?
DIAGnostic:ROSCillator:EFControl:ABSolute?
DIAGnostic:LIFetime:COUNt?
```

DIAGnostic:ROSCillator:EFControl:RELative?

This query returns the Electronic Frequency Control (EFC) output value of the internal reference oscillator. It returns a percentage value between -100% to +100%.

DIAGnostic:ROSCillator:EFControl:ABSolute?

This query returns the Electronic Frequency Control (EFC) output value of the internal reference oscillator. It returns a value in volts.

DIAGnostic:LIFetime:COUNt?

This command returns the number of hours the unit has been powered-on.

MEASURE Subsystem

This subsystem groups the queries related to some parameters that are measured on-board on the device. The list of the commands supported for this subsystem is the following:

```
MEASure:TEMPerature?
MEASure:VOLTage?
MEASure:POWersupply?
```

```
MEASure:POWersupply:V12?
MEASure:POWersupply:V25?
MEASure:CURRent?
MEASure?
```

MEASure:TEMPerature?

This query returns the internal CSAC temperature if the optional CSAC is enabled.

MEASure:VOLTage?

This query returns the TCXO EFC voltage for units without optional CSAC, and the CSAC telemetry internal TCXO EFC voltage for units with optional CSAC.

MEASure:POWersupply?

This query returns the power supply voltage applied to the prime-power input (~7V to 36V).

MEASure:POWersupply:V12?

This query returns the internally regulated 1.2V supply voltage.

MEASure:POWersupply:V25?

This query returns the internally regulated 2.5V supply voltage.

MEASure:CURRent?

This command is included for backwards compatibility and responds with the CSAC temperature if the optional CSAC is enabled.

MEASure?

This query returns the result of the two following queries:

```
MEASure:TEMPerature?
MEASure:VOLTage?
MEASure:POWersupply?
```

```
MEASure:POWersupply:V12?
MEASure:POWersupply:V25?
```

GPS Subsystem

NOTE

The RSR Transcoder 2.0 displays antenna height in GPS height in meters on the SCPI port rather than in MSL height on all commands that return antenna height except for standard NMEA output sentences with fields defined as MSL height. This corresponds with the GPS height input for all altitude input in the Simulation options.

The list of the commands supported by Generic receivers is the following:

```
GPS:TYPE?
GPS:TYPE:MODE <AUTO | UBLOX | NMEA>
GPS:SATellite:TRAcking:COUNt?
GPS:SATellite:VISible:COUNt?
GPS:GPGGA <int> [0,255]
GPS:GPRMC <int> [0,255]
GPS:GPZDA <int> [0,255]
GPS:GPGSV <int> [0,255]
GPS:PASHR <int> [0,255]
GPS:HEIGHT?
GPS:HEIGHT:MSL?
GPS:HEIGHT:GPS?
GPS:INITial:DATE <yyyy,mm,dd>
GPS:INITial:TIME <hour,min,sec>
GPS:PORT <SCPI | RS232>
GPS:PPSERRor?
GPS:POSERRor?
GPS:VPOSERRor?
GPS?
```

The list of commands supported by externally-connected uBlox and Rockwell GNSS receivers is the following:

```
GPS:REFerence:PULse:SAWtooth?
GPS:RESET ONCE
GPS:FWver?
```

The list of commands supported only by externally-connected uBlox receivers is the following:

```
GPS:XYZSPeed <int> [0,255]
GPS:DYNAMic:MODE <int> [0,8]
GPS:DYNAMic:STATe?
GPS:REFerence:ADELay <float> <s | ns > [-32767ns,32767ns]
GPS:TMODe <ON | OFF | RSTSURV>
GPS:SURVey ONCE
GPS:SURVey:DURation <sec>
GPS:SURVey:VARiance <mm2>
GPS:HOLD:POSition <cm, cm, cm>
GPS:SURVey:STATus?
GPS:SYST:SELect [GPS | SBAS | QZSS | GAL | BD | GLO]
GPS:JAMlevel?
```

GPS:TYPE?

This command queries the current externally-connected GNSS type. The response is one of the following:

- UBLOX
- NMEA

The GPS type is set automatically when in GPS:TYPE:MODE AUTO. Otherwise, the GPS type matches the selected type mode as described below.

GPS:TYPE:MODE

This command selects the GNSS type mode. The default mode is AUTO so that the GNSS type is automatically selected shortly after power up based on serial port queries to the GNSS receiver. To override the automatic selection, use the UBLOX or NMEA option. The NMEA option will require industry-standard NMEA sentences including GPGGA and GPRMC to be sent to the unit's RS-232 serial port at either 9600 or 38400 baud.

The format of this command is:

```
GPS:TYPE:MODE <AUTO | UBLOX | NMEA>
```

GPS:SATellite:TRAcking:COUNt?

This query returns the number of satellites being tracked.

GPS:SATellite:VISible:COUNt?

This query returns the number of satellites (PRN) that the almanac predicts should be visible at the current simulated date, time, and position.

NMEA Support

The following commands allow the RSR Transcoder 2.0 to be used as an industry standard navigation multi-GNSS receiver. The GPGGA, GPGSV, GPRMC, PASHR and GPZDA NMEA commands comprise all necessary information about the antenna position, height, velocity, direction, satellite info, fix quality info, time, date and other information that can be used by standard navigation applications via the USB serial interface.

Once enabled, the unit will send out NMEA sentences on the USB port automatically every N seconds. All incoming serial commands are still recognized by the unit, since the serial interface transmit and receive lines operate completely independently of one another.

For compatibility with existing GPS-only products, the units's NMEA output uses only the GPS NMEA sentence headers (GPGGA, GPGSV, etc.) regardless of the GNSS systems enabled on an external uBlox receiver. Also, the GPGSV output uses a modified satellite numbering scheme as detailed in Section "GPS:GPGSV" on page 81to allow multiple, possibly concurrent GNSS system satellites to be differentiated in the GPGSV message.

> **NOTE**
>
> Due to internal double-buffering that the position, direction, and speed data is delayed by one second from when the external GNSS receiver reported these to the internal microprocessor. Consequently the position is valid for the 1PPS pulse previous to the last 1PPS pulse at the time the data is sent (one second delay). The time and date are properly output with correct UTC synchronization to the 1PPS pulse immediately prior to the NMEA data being sent.

Once set, the following command settings will be stored in NV memory, and generate NMEA output information even after power to the unit has been power cycled.

By default NMEA strings are not generated until at least one external 1PPS reference pulse has been received. The unit can be set to output NMEA data asynchronously and regardless of the external 1PPS pulse status by using the ON setting of the SYNC:OUT:1PPS:RESET command as described in "SYNChronization:OUTput:1PPS:RESET [ON|OFF]" on page 74

GPS:GPGGA

This command instructs the RSR Transcoder 2.0 to send the NMEA standard string $GPGGA every N seconds, with N in the interval [0,255]. The command is disabled during the initial warm-up phase.

This command has the following format:

```
GPS:GPGGA <int> [0,255]
```

GPGGA shows height in MSL Meters.

GPS:GPRMC

This command instructs the RSR Transcoder 2.0 to send the NMEA standard string $GPRMC every N seconds, with N in the interval [0,255]. The command is disabled during the initial warm-up phase.

This command has the following format:

```
GPS:GPRMC   <int>   [0,255]
```

GPS:GPZDA

This command instructs the RSR Transcoder 2.0 to send the NMEA standard string $GPZDA every N seconds, with N in the interval [0,255]. The command is disabled during the initial warm-up phase.

This command has the following format:

```
GPS:GPZDA   <int>   [0,255]
```

GPS:GPGSV

This command instructs the RSR Transcoder 2.0 to send the NMEA standard string $GPGSV every N seconds, with N in the interval [0,255]. The command is disabled until the GNSS receiver achieves a first fix.

```
GPS:GPGSV <int> [0,255]
```

NOTE

Due to the large number of GNSS satellites that can be tracked in this unit, more than the customary four GSV messages can be sent once per second. With multiple GNSS systems enabled, a typical sky view may generate up to six GSV messages per second.

To simultaneously support all available GNSS systems, the following PRN numbering scheme modified from the traditional NMEA standard is being used:

GNSS Type	SV Range	GPGSV PRN vehicle numbering
GPS	G1-G32	1-32
SBAS	S120-S158	33-64,152-158

Galileo	E1-E36	301-336
BeiDou	B1-B37	401-437
IMES	I1-I10	173-182
QZSS	Q1-Q5	193-197
GLONASS	R1-R32, R?	65-96,0

GPS:PASHR

The PASHR command alongside the GPZDA command will return a complete list of relevant parameters including time, date, position, velocity, direction, altitude, quality of fix, and more. As an example, the PASHR string has the following data format:

```
$PASHR,POS,0,7,202939.00,3716.28369,N,12157.43457,W,00087.4
0,????,070.01,000.31,-000.10,05.6,03.5,04.3,00.0,DD00*32
```

> **NOTE**
> The length of the string is fixed at 115 characters plus the two binary 0x0d, 0x0a termination characters.

```
$PASHR,POS,0,aa,bbbbbb.00,cccc.ccccc,d,eeeee.eeeee,f,ggggg.
gg,hhhh,iii.ii,jjj.jj,kkkk.kk,ll.l,mm.m,nn.n,00.0,p.pp,*[ch
ecksum]
```

Where:

aa: Number of Sats

bbbbbb.00: Time of Day UTC

cccc.ccccc,d: Latitude,S/N

eeee.eeeee,f: Longitude,W/E

ggggg.gg: Antenna Height in meters

hhhh: Four fixed '?' symbols

iii.ii: Course Over Ground

jjj.jj: Speed in Knots

kkkk.k: Vertical Velocity in meters/s

ll.l: PDOP

mm.m HDOP

nn.n VDOP

00.0 Static number

p.pp: Firmware version

This command instructs the RSR Transcoder 2.0 to send the NMEA standard string $PASHR every N seconds, with N in the interval [0,255]. The command is disabled during the initial warm-up phase.

This command has the following format:

```
GPS:PASHR   <int>  [0,255]
```

GPS:XYZSPeed

This command is a 3D velocity vector output command. Enabling this command will output a 3-dimensional velocity vector indicating the unit's speed in centimeters per second in the ECEF coordinate system.

X, Y, and Z speed are individually given, and are independent of each other. An accuracy estimate in centimeters per second is also given. The velocity data is time-stamped using the time-of-week with a resolution of milliseconds.

Additionally, the number of accrued leap seconds is indicated in this message, which allows proper calculation of GPS time from UTC time as indicated by other messages, as well as proper handling of leap second events.

Use the following format to generate the velocity vector every N seconds, with N in the interval [0,255]:

```
GPS:XYZSPeed <int> [0,255]
```

This command is supported only with an external uBlox GNSS receiver.

GPS:HEIGHT:MSL?

This query returns the Mean Sea Level height in meters which differs from the GPS ellipsoid height by up to +/-100 meters. This difference varies depending upon the simulated location. The MSL to GPS height differences are calculated by the internal uBlox GNSS monitoring receiver, and reported based on the height delta at the simulated GPS position.

GPS:HEIGHT:GPS?

This query returns the height above the GPS ellipsoid in meters at the currently simulated GPS position.

GPS:HEIGHT?

This command returns the output from the following queries:

```
GPS:HEIGHT:MSL?
GPS:HEIGHT:GPS?
```

GPS:DYNAMic:MODE

This command allows the user to select the dynamic motion model being applied to the Kalman filters in the GNSS receiver. This allows for larger amounts of filtering for lower velocity applications, effectively reducing noise and multipath interference. Applications with high acceleration or velocity can then be used with fast filter settings to allow for the most accurate GNSS coordinates to be provided in high-dynamic applications such as Jet aircraft. Doppler tracking is enabled in all airborne modes, as Carrier Phase tracking is very difficult to achieve in high velocity applications. The GNSS receiver will perform Carrier Phase tracking for non-airborne modes.

The command has the following syntax:

```
GPS:DYNAMic:MODE <int> [0,8]
```

Sending the following command to the RSR Transcoder 2.0 will select a stationary GNSS dynamic model for example:

```
gps:dynam:mode 1
```

The following table lists all available Dynamic modes:

Table 8 Supported Dynamic GNSS operating modes.

Value	Model	Application
0	Portable	Recommended as a default setting
1	Stationary	Used in stationary applications
2	Pedestrian	Used in man-pack, pedestrian settings
3	Automotive	Vehicular velocity applications
4	Sea	Used on ships, where altitude is expected to be constant
5	Airborne <1g	Airborne applications with less than 1g acceleration
6	Airborne <2g	Airborne applications with less than 2g acceleration
7	Airborne <4g	Airborne applications with less than 4g acceleration
8	Automatic Mode	Select one of the above states (0 – 7) based on the actual velocity of the vehicle

If the external GNSS receiver is a timing-grade uBlox receiver it is capable of running in a stationary mode with Position Auto Survey called Position Hold Mode. This mode increases timing stability by storing the position into memory, solving the GNSS signal only for time as the position is not expected to change. This allows operation with only one visible satellite, as well as over-determination of the timing pulse when multiple satellites are tracked. Two modes can be selected for Auto Survey operation (see section “GPS:TMODe <ON | OFF | RSTSURV>” on page 87):

1) Manually setting Timing Mode to ON with a hard-coded position in NVRAM

2) Enabling Auto Survey to start automatically after power-on by setting Timing Mode to RSTSURV

If either one of the above two GPS:TMODE Auto Survey/Position Hold modes is selected, the GPS:DYNAMIC:MODE command is disabled internally and its setting is ignored as the unit does not expect any motion on the antenna. In this case, the dynamic state as programmed into the GNSS receiver is set to STATIONARY independent of the user selection for GPS:DYNAMIC:MODE.

The current dynamic state being applied to the GNSS receiver can be queried with the command

```
GPS:DYNAMIC:STATE?
```

This command is supported only with an external uBlox GNSS receiver.

GPS:DYNAMic:MODE 8 (Automatic Dynamic Mode)

Automatic Dynamic Mode allows the RSR Transcoder 2.0 firmware to automatically configure an external uBlox GNSS receiver's Kalman filter parameters based on actual mission velocities and motion profiles, drastically improving overall performance. The unit will try to set the uBlox GNSS receiver to the optimal setting for any given velocity. The unit is able to set 7 different modes, as shown in "GPS:DYNAMic:MODE" on page 84.

The following table shows the Dynamic mode the unit will program into the uBlox GNSS receiver when Automatic Mode is selected (Dynamic Mode 8).

Table 9 Auto Dynamic Mode Switching Rules

Velocity Threshold	**Selected Dynamic Model**	**Fallback to lower setting**
0 – 2 knots	Stationary	none
>2 knots	Pedestrian	<1 knots
>10 knots	Automotive	<8 knots
>60 knots and >400 Feet/ min climb/descent	Airborne 1g	<50 knots
>150 knots	Airborne 2g	<130 knots
>240 knots	Airborne 4g	<210 knots

In this Automatic mode, the unit will configure the uBlox GNSS receiver based on the actual vehicle velocity.

> **NOTE**
>
> In order to switch from the Automotive mode into the first Airborne (1g) mode, a vehicle velocity greater than 60 knots as well as a climb/descent rate greater than 400 feet per minute are required. Alternatively, a vehicle velocity of greater than 100 knots will also initiate a switch into airborne-1g mode. Without an appropriate climb/descent, the unit will remain in Automotive mode unless the 100-knot velocity threshold is exceeded.

The following command returns the setting of the GNSS dynamics model:

```
GPS:DYNAMic:MODE?
```

The actual state chosen by the firmware for the GNSS receiver based on vehicle velocity can be queried with the command:

```
GPS:DYNAMic:STATe?
```

A value between 0 and 7 is then returned depending on vehicle dynamics.

The dynamic state is always set to STATIONARY if one of the Position Hold Auto Survey stationary modes is selected using the command GPS:TMODE, as the Position Hold mode setting overrides any dynamic state user setting.

Settings will be applied immediately to the uBlox GNSS receiver, and are stored in nonvolatile memory. This command is supported only with an external uBlox GNSS receiver.

GPS:DYNAMic:STATe?

This query returns the dynamic model chosen by the firmware to be applied to the external uBlox GNSS receiver depending on vehicle velocity. It returns a value between 0 and 7 which corresponds to one of the dynamic models defined in the Table in "GPS:DYNAMic:MODE 8 (Automatic Dynamic Mode)" on page 85.

This state can be different from the user-selected Dynamic model mode for two reasons:

- if the dynamic mode is set to 8 (Automatic mode), the state will reflect the dynamic model being applied to the GNSS receiver depending on actual vehicle dynamics
- if the GPS Timing Mode is set to ON or to RSTSURV, the dynamic state will always be set to 1 (Stationary)

This command is supported only with an external uBlox GNSS receiver.

GPS:REFerence:ADELay <float> <s | ns > [-32767ns,32767ns]

The ADELay command allows bidirectional shifting of the 1PPS output in relation to the UTC 1PPS reference in one nanosecond steps. This allows antenna cable delay compensation, as well as retarding or advancing the 1PPS pulse arbitrarily to calibrate different units to each other for example. Typical antenna delays for a 30 foot antenna cable with 1.5ns per foot propagation delay would be compensated with the following command:

```
GPS:REF:ADEL 45ns
```

This command can be used to fine-tune different units to have coincident 1PPS pulse outputs. This command is supported only with an external uBlox GNSS receiver.

GPS:REFerence:PULse:SAWtooth?

This command returns the momentary sawtooth correction factor indicated by the GNSS receiver. This command is supported with both an external uBlox GNSS receiver and a Rockwell Microgram or RSR Puck loaded with custom VIAVI NMEA firmware.

GPS:RESET ONCE

Issues a reset to the external GNSS receiver. This can be helpful when changing the antenna, for example; many GNSS receivers measure the antenna system's C/No right after reset and adjust their internal antenna amplifier gains accordingly. It takes approximately 1 minute for GPS lock to occur after a GNSS receiver reset. The externally-connected GNSS receiver is reconfigured to power-on default specifications when issuing this command.

GPS:TMODe <ON | OFF | RSTSURV>

This command selects the Timing Mode of an external timing-enabled uBlox GNSS receiver.

If the Timing Mode is OFF, the receiver will act as a regular GNSS receiver in 3D mobile mode. This mode has to be chosen if the unit is used with a moving antenna.

If the Timing Mode in ON, the timing features of the GNSS receiver are enabled. At power-up, the Hold position stored in NVRAM will be sent to the GNSS receiver and will be used as the reference. In order to use this mode, the receiver position must be known as accurately and precisely as possible. Errors in the Hold position will translate into time errors depending on the satellite constellation.

The Hold position can be set manually by the user or can be the result of a position Auto Survey executed by the uBlox GNSS receiver.

If the Timing Mode is set to RSTSURV, the uBlox GNSS receiver will start an Auto Survey every time the unit is powered-on and following the Survey sequence, the uBlox GNSS receiver will run with the timing features enabled. Once in Position Hold mode, the antenna location should be held completely stationary. This command is supported only with an external Timing uBlox GNSS receiver.

GPS:SURVey ONCE

This command starts an Auto Survey. At the end of the Survey, the calculated Hold position will be stored in NVRAM. The Survey parameters can be set with the command **GPS:SURVey:DURation** and **GPS:SURVey:VARiance.** This command is supported only with an external Timing uBlox GNSS receiver.

GPS:SURVey:DURation <sec>

This command sets the survey minimal duration. It is supported only with an external Timing uBlox GNSS receiver.

GPS:SURVey:VARiance <mm^2>

This command specifies the minimum variance of the average position computed during the survey. This minimum value is used as a threshold under which the uBlox GNSS receiver can stop the survey. The GNSS receiver will stop the Survey when the minimal duration has been reached and the variance of the average position is under the specified minimum variance. This command is supported only with an external Timing uBlox GNSS receiver.

GPS:HOLD:POSition <cm, cm, cm>

This command allows the user to manually specify the exact position of the antenna. This command will overwrite the Hold position in NVRAM. A subsequent Auto Survey will overwrite this Hold position. The Hold position is stored in ECEF coordinates. This command is supported only with an external Timing uBlox GNSS receiver.

GPS:SURVey:STATus?

This query displays the current status of the Auto Survey. The survey status will be reported as one of the following 3 states:

- ACTIVE: a survey is in progress
- VALID: a survey has been achieved successfully and the uBlox GNSS receiver is now using this Hold position as reference.
- INVALID: no survey is in progress or has been achieved since the last power cycle.

When in ACTIVE or VALID state, this query will also display the duration, the Hold position in ECEF coordinates and the position variance. This command is supported only with an external Timing uBlox GNSS receiver.

GPS:INITial:DATE <yyyy,mm,dd>

This command allows the manual setting of the internal RTC DATE when operating the unit in GNSS-denied environments. This command is compatible with the PTIME:OUT ON command described in section "PTIME:OUTput <ON | OFF>" on page 100 to allow automatic time and date synchronization of two units to each other. This command can also be used to set the date for simulations when using the Continuous Simulation Time mode.

GPS:INITial:TIME <hour,min,sec>

This command allows manual setting of the internal RTC TIME when operating the unit in GNSS-denied environments. This command is compatible with the PTIME:OUT ON command described in section "PTIME:OUTput <ON | OFF>" on page 100 to allow automatic time and date synchronization of two units to each other. This command can also be used to set the time for simulations when using the Continuous Simulation Time mode.

GPS:PORT <SCPI | RS232>

This command selects the source for NMEA data between the SCPI and RS232 ports. When using the SIMulation:NMEA command described in "SIMulation:NMEA <nmea string>" on page 66to provide NMEA data over the SCPI port, the GPS:PORT setting must be SCPI. Normally the setting should be RS232 to support an externally connected GPS receiver.

GPS:PPSERRor?

This command queries and returns the difference in seconds between the external GPS receiver's 1PPS rising edge and the current simulator time.

This query is useful for monitoring the timing accuracy of the target GPS receiver by connecting the 1PPS output of that receiver to the RSR Transcoder 2.0's external 1PPS input. The synchronization source mode should be in NMEA mode (SYNC:SOUR:MODE NMEA) when connecting the target GPS receiver this way.

This query is also useful for monitoring how closely the simulation time matches the reference 1PPS input when the synchronization source mode is EXTernal or GPS.

GPS:POSERRor?

This command queries and returns the horizontal difference in meters between the position reported by the external GPS receiver and the simulated position.

This query is useful for monitoring the horizontal position accuracy of the target GPS receiver with the target GPS receiver connected to the RSR Transcoder 2.0's external GPS receiver RS232 port. The target GPS receiver should not be connected this way in the TRANSCODE simulation mode.

This query is also useful for monitoring the performance of the simulation position filtering when using the TRANSCODE mode with an external GPS receiver providing the input position data.

GPS:VPOSERRor?

This command queries and returns the vertical difference in meters between the position reported by the external GPS receiver and the simulated position.

This query is useful for monitoring the vertical position accuracy of the target GPS receiver with the target GPS receiver connected to the RSR Transcoder 2.0's external GPS receiver RS232 port. The target GPS receiver should not be connected this way in the TRANSCODE simulation mode.

This query is also useful for monitoring the performance of the simulation position filtering when using the TRANSCODE mode with an external GPS receiver providing the input position data.

GPS:SYST:SELect [GPS | SBAS | QZSS | GAL | BD | GLO]

This command selects the GNSS systems that are enabled in an externally connected uBlox-8 GNSS receiver. The command is followed by any combination of the currently supported GNSS system abbreviations GPS, SBAS, QZSS, BD (BeiDou), GAL (Galileo) and GLO (GLONASS). Up to three concurrent GNSS systems plus SBAS may be enabled at any time. This command is supported only with an external uBlox generation 8 or later GNSS receiver.

Only two different L1 carrier frequencies are allowed to be received at any given time. GPS, Galileo, QZSS, and SBAS count as one frequency, while BeiDou and Glonass each count as a separate frequency. For example, while GPS, Galileo, and Glonass can be received concurrently, reception of BeiDou or Glonass will limit the user to only one other GNSS system. Any illegal GNSS system combinations will result in a "command error" response from the system. A typical example of this command is:

```
GPS:SYST:SEL GPS SBAS GAL GLO
```

As of the date of this manual, the GPS:SYST:SEL command is not currently supported on the RSR Transcoder.

GPS:RATE [1|10]

This command selects the solution rate and GPGGA/GPRMC NMEA output rate in Hz enabled on an externally connected uBlox receiver. Only 1Hz and 10Hz rates are supported.

GPS:JAMlevel?

Externally-connected uBlox GNSS receivers will detect and flag jamming interference at levels ranging from 0 (no jamming) to 255 (strong jamming), and will report the observed jamming level with this command. This command is supported only with an external uBlox GNSS receiver.

GPS:FWver?

This command returns the firmware version used inside the externally-connected GNSS receiver itself. This command is supported with both an external uBlox GNSS receiver and a Rockwell Microgram or RSR Puck.

GPS?

This query displays the configuration, position, speed, height and other relevant data of the external GNSS receiver in one convenient location.

INTGPS Subsystem

The INTGPS subsystem groups all the commands related to the control and status of the internal monitoring GNSS receiver. The internal GNSS receiver is used to monitor the RF output signal on the SMA connector, providing a sanity check for the simulated position and calibrating the timing for the simulated signal. The list of the commands supported is the following:

```
INTGPS:SATellite:TRAcking:COUNt?
IINTGPS:SATellite:VISible:COUNt?
INTGPS:GPGGA <int> [0,255]
INTGPS:GPRMC <int> [0,255]
INTGPS:GPGSV <int> [0,255]
INTGPS:GPZDA <int> [0,255]
INTGPS:PASHR <int> [0,255]
INTGPS:XYZSPeed <int> [0,255]
INTGPS:HEIGHT?
INTGPS:HEIGHT:MSL?
```

```
INTGPS:HEIGHT:GPS?
INTGPS:DYNAMic:MODE <int> [0,8]
INTGPS:DYNAMic:STATe?
INTGPS:REFerence:PULse:SAWtooth?
INTGPS:RESET ONCE
INTGPS:JAMlevel?
INTGPS:FWver?
INTGPS:PPSERRor?
INTGPS:POSERRor?
INTGPS:VPOSERRor?
INTGPS:SYSTem:STATus?
INTGPS?
```

INTGPS:SATellite

This group of commands describe the satellite constellation as seen by the internal GNSS receiver.

INTGPS:SATellite:TRAcking:COUNt?

This query returns the number of satellites being tracked by the internal GNSS receiver.

INTGPS:SATellite:VISible:COUNt?

This query returns the number of satellites that the almanac predicts should be visible at the current simulated date, time, and position.

Internal GNSS Receiver NMEA Support

The following commands allow the RSR Transcoder 2.0 to provide standard NMEA data from the internal GNSS receiver. The GPGGA,GPRMC, GPGSV, PASHR and GPZDA NMEA commands comprise all necessary information about the antenna position, height, velocity, direction, satellite info, fix info, time, date and other information that can be used by standard navigation applications or as a comparison with the simulation results from the target GPS receiver.

Once enabled, the RSR Transcoder 2.0 will send out information on the USB serial interface automatically every N seconds. All incoming serial commands are still recognized by the RSR Transcoder 2.0, since the serial interface transmit and receive lines are completely independent of one another.

> **NOTE**
>
> NMEA output configured for the internal GNSS receiver appears identical to the NMEA output for the optional external GNSS receiver. To prevent confusing the output, only one (internal or external) NMEA sentence of each type should be enabled.

The position, direction, and speed data is delayed by one second from when the GPS receiver internally reported these to the RSR Transcoder 2.0's microprocessor, so the position is valid for the 1PPS pulse previous to the last 1PPS pulse at the time the data is sent (one second delay). The time and date are properly output with correct UTC synchronization to the 1PPS pulse immediately prior to the data being sent.

Once set, the following two commands will be stored in NV memory, and will generate output information even after power to the unit has been cycled.

INTGPS:GPGGA

This command instructs the RSR Transcoder 2.0 to send the NMEA standard string $GPGGA every N seconds, with N in the interval [0,255]. The command is disabled until the internal GNSS receiver achieves a first fix.

This command has the following format:

```
INTGPS:GPGGA   <int>  [0,255]
```

> **NOTE**
>
> GPGGA shows height in MSL Meters; this is different from traditional GPS receivers that display height in GPS Meters. The difference between MSL and GPS height can be significant, with discrepancies of 35m or more being common.
>
> **NOTE**
>
> The simulation height is specified in GPS height above the reference ellipsoid, so the height returned by this NMEA message will vary from the simulated height by the GPS to MSL height difference at the particular location that is being simulated.

INTGPS:GPRMC

This command instructs the RSR Transcoder 2.0 to send the NMEA standard string $GPRMC every N seconds, with N in the interval [0,255]. The command is disabled until the internal GNSS receiver achieves a first fix.

This command has the following format:

```
INTGPS:GPRMC   <int>  [0,255]
```

INTGPS:GPGSV

This command instructs the RSR Transcoder 2.0 to send the NMEA standard string $GPGSV every N seconds, with N in the interval [0,255]. The command is disabled until the internal GPS receiver achieves a first fix.

```
INTGPS:GPGSV <int> [0,255]
```

INTGPS:XYZSPeed

This command is a 3D velocity vector output command. Enabling this command will output a 3 dimensional velocity vector indicating the unit's speed in centimeters per second in the ECEF coordinate system.

X, Y, and Z speed are individually given, and are independent of each other. An accuracy estimate in centimeters per second is also given. The velocity data is time-stamped using the time-of-week with a resolution of milliseconds.

Additionally, the number of accrued Leapseconds is indicated in this message, which allows proper calculation of GPS time from UTC time as indicated by other messages, as well as proper handling of Leapsecond events.

Use the following format to generate the velocity vector every N seconds, with N in the interval [0,255]:

```
INTGPS:XYZSPeed <int> [0,255]
```

INTGPS:GPZDA

This command instructs the RSR Transcoder 2.0 to send the NMEA standard string $GPZDA every N seconds, with N in the interval [0,255]. The command is disabled until the internal GNSS receiver achieves a first fix.

This command has the following format:

```
INTGPS:GPZDA   <int>  [0,255]
```

INTGPS:PASHR

The NMEA string $PASHR,POS has been added for compatibility to legacy GPS hardware. The PASHR command alongside the GPZDA command will give all relevant parameters such as time, date, position, velocity, direction, altitude, quality of fix, and more. As an example, the String has the following data format:

```
$PASHR,POS,0,7,202939.00,3716.28369,N,12157.43457,W,00087.4
0,????,070.01,000.31,-000.10,05.6,03.5,04.3,00.0,DD00*32
```

The length of the string is fixed at 115 characters plus the two binary 0x0d, 0x0a termination characters.

```
$PASHR,POS,0,aa,bbbbbb.00,cccc.ccccc,d,eeeee.eeeee,f,ggggg.
gg,hhhh,iii.ii,jjj.jj,kkkk.kk,ll.l,mm.m,nn.n,00.0,p.pp,*[ch
ecksum]
```

Where:

- aa:Number of Sats
- bbbbbb.00:Time of Day UTC
- cccc.ccccc,d:Latitude,S/N
- eeee.eeeee,f:Longitude,W/E
- ggggg.gg:Antenna Height in meters
- hhhh:Four fixed ‘?’ symbols
- iii.ii:Course Over Ground
- jjj.jj:Speed in Knots
- kkkk.k:Vertical Velocity in meters/s
- ll.l:PDOP
- mm.mHDOP
- nn.nVDOP
- 00.0Static number
- p.pp:Firmware Version (1.05 and above)

This command instructs the RSR Transcoder 2.0 to send the NMEA standard string $PASHR every N seconds, with N in the interval [0,255]. The command is disabled until the internal GNSS receiver achieves a first fix.

This command has the following format:

```
INTGPS:PASHR   <int>  [0,255]
```

INTGPS:HEIGHT:MSL?

This query returns the Mean Sea Level height in meters which differs from the GPS ellipsoid height by up to +/-100 meters. This difference varies depending upon the location as reported by the internal GPS receiver. The simulation uses GPS height in commands such as the SIM:POS:LLH command described in “SIMulation:POSition:LLH” on page 44

INTGPS:HEIGHT:GPS?

This query returns the GPS ellipsoid height in meters as reported by the internal GPS reference receiver. It is calculated from the internal uBlox GNSS receiver’s MSL height parameter by subtracting the MSL to GPS ellipsoid height difference for the simulated position. This GPS height should reflect the simulated height as set in the simulation commands such as the SIM:POS:LLH command described in “SIMulation:POSition:LLH” on page 44.

INTGPS:HEIGHT?

This command returns the output from the following queries:

```
INTGPS:HEIGHT:MSL?
INTGPS:HEIGHT:GPS?
```

INTGPS:DYNAMic:MODE

This command allows the user to select the dynamic motion model applied to the Kalman filters in the internal monitoring GNSS receiver. This allows for larger amounts of filtering for lower velocity applications, effectively reducing noise and multipath interference. Applications with high acceleration can be used with fast filter settings to allow for the most accurate GPS coordinates to be provided in high-dynamic applications such as jet aircraft. Doppler tracking is enabled in all airborne modes, as Carrier Phase tracking is very difficult to achieve in high velocity applications. The GNSS receiver will perform Carrier Phase tracking for non-airborne modes.

The command has the following syntax:

```
INTGPS:DYNAMic:MODE <int> [0,8]
```

Sending the following command to the RSR Transcoder 2.0 will select a stationary GNSS dynamic model, for example:

```
INTGPS:DYNAM:MODE 1
```

The following table lists all available modes:

Value	**Model**	**Application**
0	Portable	Recommended as a default setting
1	Stationary	Used in stationary applications
2	Pedestrian	Used in man-pack, pedestrian settings
3	Automotive	Vehicular velocity applications
4	Sea	Used on Ships, where altitude is expected to be constant
5	Airborne <1g	Airborne applications with less than 1g acceleration
6	Airborne <2g	Airborne applications with less than 2g acceleration
7	Airborne <4g	Airborne applications with less than 4g acceleration
8	Automatic Mode	Select one of the above states (0 – 7) based on the actual velocity of the vehicle

INTGPS:DYNAMic:MODE 8 (Automatic Dynamic Mode)

Automatic Dynamic Mode allows the RSR Transcoder 2.0 firmware to automatically configure the internal monitoring GNSS receiver Kalman filter parameters based on simulated/transcoded velocities and motion profiles. The unit will try to set the internal GNSS receiver to the optimal setting for any given velocity. The unit is able to set 7 different modes, as shown in Section “INTGPS:DYNAMic:MODE” on page 95.

The following table shows the Dynamic mode that the unit will program into the internal GNSS receiver when Automatic Mode is selected (Dynamic Mode 8).

Table 10 Dynamic modes

Velocity Threshold	Selected Dynamic Model	Fallback to lower setting
0 – 2 knots	Stationary	none
>2 knots	Pedestrian	<1 knots
>10 knots	Automotive	<8 knots
>60 knots and >400 Feet/ min climb/descent	Airborne 1g	<50 knots
>150 knots	Airborne 2g	<130 knots
>240 knots	Airborne 4g	<210 knots

In this Automatic mode, the unit will configure the internal GNSS receiver based on the simulated/transcoded vehicle velocity.The following command returns the setting of the

> **NOTE**
>
> In order to switch from the Automotive mode into the first Airborne (1g) mode, a vehicle velocity greater than 60 knots as well as a climb/descent rate greater than 400 feet per minute are required. Alternatively, a vehicle velocity of greater than 100 knots will also initiate a switch into airborne-1g mode.
>
> Without an appropriate climb/descent, the unit will remain in Automotive mode.

GNSS dynamic model:

```
INTGPS:DYNAMic:MODE?
```

The actual state chosen by the firmware for the internal GNSS receiver based on vehicle velocity can be obtained with the command:

```
INTGPS:DYNAMic:STATe?
```

A value between 0 and 7 is then returned depending on simulated/transcoded vehicle dynamics.

Settings will be applied immediately to the internal GNSS receiver, and are stored in Non Volatile memory.

INTGPS:DYNAMic:STATe?

This query returns the actual state of the dynamic model chosen by the firmware to be applied to the internal GNSS receiver depending on simulated/transcoded vehicle velocity. It returns a value between 0 and 7, corresponding to one of the dynamic models defined in the table in "GPS:DYNAMic:MODE 8 (Automatic Dynamic Mode)" on page 85.

This state can be different from the user-selected Dynamic model mode if the dynamic mode is set to 8 (Automatic mode). In this case, the state will reflect the dynamic model being applied to the internal GNSS receiver depending on simulated/transcoded vehicle dynamics

INTGPS:REFerence:PULse:SAWtooth?

This command returns the momentary sawtooth correction factor from the internal monitoring GNSS receiver.

INTGPS:RESET ONCE

Issues a reset to the internal GNSS receiver. This can be helpful when changing amplitude settings on the simulation, since the internal GNSS receiver measures the RF output's C/No immediately after reset and adjusts its internal antenna amplifier gains accordingly. It takes approximately 1 minute for locking to commence after a GNSS receiver reset, as indicated by the internal GNSS 1PPS LED.

INTGPS:JAMlevel?

This command provides a signal-jamming indicator from the internal monitoring GNSS receiver. The internal GNSS receiver will detect and flag jamming interference at levels ranging from 0 (no jamming) to 255 (strong jamming). Normally the jamming level is low on a correctly-operating RSR Transcoder 2.0.

INTGPS:FWver?

This command queries and returns the Firmware version used inside the internal monitoring uBlox GNSS receiver.

INTGPS:PPSERRor?

This command queries and returns the difference in seconds between the internal GPS receiver's 1PPS rising edge and the current simulator time

INTGPS:POSERRor?

This command queries and returns the difference in meters between the position reported by the internal GPS receiver and the transmitted 3D position.

INTGPS:VPOSERRor?

This command queries and returns the difference in meters between the altitude reported by the internal GPS receiver and the transmitted altitude.

INTGPS:SYSTem:STATus?

This query returns a full page of internal GPS receiver status in ASCII format. This command returns a comprehensive internal GPS receiver status pages.

INTGPS?

This query displays the configuration, position, speed, height and other relevant data of the internal monitoring GNSS receiver in one convenient location.

PTIME Subsystem

The PTIME subsystem groups all the commands related to the management of the internal RTC time including simulation time when running in the Continuous simulation time mode.

NOTE

The RSR Transcoder 2.0 does not contain a battery-backup to maintain RTC time/date. Consequently the time/date is lost when power is removed, or when the unit is reset.

The following commands are supported:

```
PTIMe:DATE?
PTIMe:TIME?
PTIMe:TIME:STRing?
PTIMe:TINTerval?
PTIME:OUTput <ON | OFF>
PTIMe:LEAPsecond?
PTIMe:LEAPsecond:PENDing?
PTIMe:LEAPsecond:ACCumulated?
PTIMe:LEAPsecond:DATE?
```

```
PTIMe:LEAPsecond:DURation?
PTIME?
```

PTIMe:DATE?

This query returns the current calendar date referenced to UTC time. The year, month, and day are returned.

PTIMe:TIME?

This query returns the current 24-hour time referenced to UTC. The hour, minute, and second is returned.

PTIMe:TIME:STRing?

This query returns the current 24-hour time suitable for display (for example, 13:24:56).

PTIMe:TINTerval?

This query is identical to the SYNChronization:TINTerval? query.

PTIME:OUTput <ON | OFF>

This command allows connecting two units together through the USB serial port, with the master unit sending time and date information to the slave unit. This allows time synchronization between two units which can be useful when operating in GPS/GNSS denied environments.

Sending the command PTIM:OUT ON will cause the unit to automatically generate GPS:INIT:DATE and GPS:INIT:TIME sentences on the USB serial port once per second.

PTIMe:LEAPsecond?

This command returns the results of the four following queries:

```
PTIMe:LEAPsecond:PENDing?
PTIMe:LEAPsecond:ACCumulated?
PTIMe:LEAPsecond:DATE?
PTIMe:LEAPsecond:DURation?
```

PTIMe:LEAPsecond:PENDing?

This command returns 1 if the GPS almanac data contains a future pending leap second date and 0 if no future leap second is pending or almanac data is not available. The GNSS receiver must have the GPS system enabled for the GPS almanac to be available. This command is compatible only with an external ublox GNSS receiver.

PTIMe:LEAPsecond:ACCumulated?

This command returns the internally applied leapsecond offset between GPS time and UTC time as stored in the EEPROM (GPS almanac not received yet) or as indicated by the GNSS receiver (GPS almanac is available).

PTIMe:LEAPsecond:DATE?

This command returns the date of the pending leap second, if any. This command is compatible onlywith an external ublox GNSS receiver.

PTIMe:LEAPsecond:DURation?

This command returns the duration of the last minute of the day during a leap second event. The returned value is 59, 60 or 61 if GPS almanac data is available, and 0 otherwise. A response of 60 indicates that no leap second is pending. This command is compatible only with an external uBlox GNSS receiver.

PTIME?

This query returns the result of the following queries:

```
PTIME:DATE?
PTIME:TIME?
PTIME:TINTerval?
PTIME:OUTput?
PTIME:LEAPsecond:ACCumulated?
```

SYSTEM Subsystem

This subsystem groups the commands related to the general configuration of the RSR Transcoder 2.0. The list of the commands supported for this subsystem follows:

```
SYSTem:COMMunicate:SERial:ECHO <ON | OFF>
SYSTem:COMMunicate: SERial:PROmpt <ON | OFF>
```

```
SYSTem:COMMunicate:SERial:BAUD <AUTO | 9600 | 19200 | 38400
| 57600 | 115200>
SYSTem:COMMunicate:USB:BAUD <9600 | 19200 | 38400 | 57600 |
115200>
SYSTem:CPURESET
SYSTem:FACToryreset ONCE
SYSTem:USERreset ONCE
SYSTem:ISP
SYSTem:ID:SN?
SYSTem:ID:HWrev?
SYSTem:ID:UID?
SYSTem:ID:SN:BOX?
SYSTem:ID:TYPE?
SYSTem:ID:MODELname?
SYSTem:STATus?
SYSTem:FLAGS?
SYSTem:FEATure?
```

SYSTem:COMMunicate:SERial:ECHO

This command enables/disables echo on the serial port. Echo should be turned off when using the Z38xx application program. This command has the following format:

```
SYSTem:COMMunicate:SERial:ECHO   <ON | OFF>
```

SYSTem:COMMunicate:SERial:PROmpt

This command enables/disables the prompt “scpi>” on the SCPI command lines. The prompt must be enabled when used with the GPSCon software. This command has the following format:

```
SYSTem:COMMunicate: SERial:PROmpt   <ON | OFF>
```

SYSTem:COMMunicate:SERial:BAUD

This command sets the external GPS serial speed. The serial configuration is always 8 bit, 1 stop bit, no parity, no flow control. The factory default setting is AUTO which will detect between 9600 and 38400 baud. This command has the following format:

```
SYSTem:COMMunicate:SERial:BAUD   <AUTO | 9600 | 19200 |
38400 | 57600 | 115200>
```

SYSTem:COMMunicate:USB:BAUD

This command sets the USB serial speed. The serial configuration is always 8 bit, 1 stop bit, no parity, no flow control. The factory default rate is 115200 baud, and it should not be changed in most circumstances.

This command has the following format:

```
SYSTem:COMMunicate:USB:BAUD <9600 | 19200 | 38400 | 57600 |
115200>
```

SYSTem:CPURESET

This command resets the unit.

SYSTem:FACToryreset ONCE

This command applies the Factory Reset setting to the NVRAM.

> **WARNING**
>
> This operation resets all user parameters, erases stored motion commands and navigation data, erases installed option keys, and clears factory calibration parameters.
>
> Use of this command will require the device to be returned to VIAVI for recalibration.

Sending this command once results in a warning being output. Sending the command a second time performs the factory reset.

SYSTem:USERreset ONCE

This operation resets all user parameters to their factory default values, and erases any motion commands and navigation data uploaded to the device by the user. Factory calibration is unaffected.

SYSTem:ISP

This command sets causes the unit to enter ISP mode for reprogramming.

SYSTem:ID:SN?

This query returns the serial number of the module.

SYSTem:ID:HWrev?

This query returns the Hardware version of the module.

SYSTem:ID:UID?

This query returns four 32-bit hexadecimal words that are unique to the CPU in each individual unit.

SYSTem:ID:SN:BOX?

This query returns the serial number of the box/enclosure. No box serial number is assigned on stand-alone modules.

SYSTem:ID:TYPE?

This query returns the oscillator type installed on the board. Possible values include:

- TCXO
- MEMS
- CSAC
- TCSAC
- UNKNOWN

If UNKNOWN is returned, an error has occurred in detecting the oscillator type.

SYSTem:ID:MODELname?

This command returns the product model name.

SYSTem:STATus?

This query returns a full page of external GNSS receiver status data in ASCII format. The output is compatible with GPSCon. This command returns one of the most comprehensive external GNSS and oscillator status pages.

SYSTem:FLAGS?

The SYSTem:FLAGS? query returns two 32-bit hexadecimal values indicating the system's operational status. Each flag occupies a single bit position in one of the two words.

The first word contains the following status bits:

External 10 MHz reference present	FLAGS |= 0x1;
1PPS source present	FLAGS |= 0x2;
1PPS servo lock in progress	FLAGS |= 0x4;
1PPS servo lock complete	FLAGS |= 0x8;
CSAC present (RSR Transcoder only)	FLAGS |= 0x10;
CSAC locked (RSR Transcoder only)	FLAGS |= 0x20;
Valid GPS signal has been transmitted for >100 seconds	FLAGS |= 0x40;
Valid 3D fix reported by internal GNSS receiver	FLAGS |= 0x80;

The second word returned by SYSTem:FLAGS? is identical to the value returned by the SYNChronization:HEAlth? query. See “SYNChronization:HEAlth?” on page 74 for descriptions of these flag values.

SYSTem:FEATure?

This query returns a list of supported options and whether they are installed.

This command has the following format:

```
SYSTem:FEATure?
```

The following is an example of the query output:

```
=== Feature Status ===
ID   Status      Description
1    DISABLED    Transcoder 2.0 100 Hz Refresh Software Option
2    DISABLED    Transcoder 2.0 ICD 153 Software Option
```

SERVO Subsystem

This subsystem groups all the commands related to the adjustment of the servo loop:

```
SERVo:LOOP?
SERVo:COARSedac <int> [0,255]
SERVo:DACGain <int> [0.1,10000]
```

```
SERVo: EFCScale <float>[0.0 , 500.0]
SERVo:EFCDamping <float>[0.0 , 4000.0]
SERVo:SLOPe <NEG | POS >
SERVo:TEMPCOmpensation
SERVo:AGINGcompensation <float> [-10.0, 10.0]
SERVo:PHASECOrrection <float> [-100.0, 100.0]
SERVo:1PPSoffset <int> ns
SERVo:TRACe <int > [0,255]
SERVo?
```

SERVo:LOOP?

This command returns the currently enabled servo loop, either TCXO or CSAC. This command is supported only on the RSR Transcoder 2.0 with the internal CSAC option.

SERVo:COARSedac

This command sets the coarse DAC that controls the TCXO EFC on units without the internal CSAC option. The RSR Transcoder 2.0 control loop automatically adjusts this setting during disciplining. The user should not have to change this value.

This command has the following format:

```
SERVo:COARSedac  <int>  [0,255]
```

SERVo:DACGain

This command is used for factory setup. The format of this command is:

```
SERVo:DACGain <int> [0.1,10000]
```

SERVo:EFCScale

This command controls the Proportional part of the PID loop. Typical values are 0.5 to 3.0. Larger values increase the loop control at the expense of increased noise while locked. Setting this value too high can cause loop instability.

This command has the following format:

```
SERVo: EFCScale   <float>[0.0 , 500.0]
```

SERVo:EFCDamping

This command adjusts the damping factor of the lowpass filter in the DAC control loop. Values from 2.0 to 50 are typically used. Larger values result in less noise at the expense of phase delay.This command has the following format:

```
SERVo:EFCDamping <float>[0.0 , 4000.0]
```

SERVo:SLOPe

The parameter determines the sign of the slope between the EFC and the frequency variation of the TCXO. This parameter should be set to match the TCXO's EFC frequency slope and should not be changed from factory settings. This command has the following format:

```
SERVo:SLOPe <NEG | POS >
```

SERVo:TEMPCOmpensation

This command is not supported on the RSR Transcoder 2.0 board.

SERVo:AGINGcompensation

This parameter is a coefficient that represents the drift of the EFC needed to compensate the natural drift in frequency of the TCXO or CSAC due to aging. This coefficient is automatically computed and adjusted over time by the VIAVI firmware. This command has the following format:

```
SERVo:AGINGcompensation <float> [-10.0, 10.0]
```

SERVo:PHASECOrrection

This parameter sets the Integral part of the PID loop. Loop instability will result if the parameter is set too high. Typical values are 10.0 to 30.0. This command has the following format:

```
SERVo:PHASECOrrection <float> [-100.0, 100.0]
```

SERVo:1PPSoffset

This command sets the RSR Transcoder 2.0 1PPS signal's offset to UTC in 5.6ns steps.

Using the SERV:1PPS command results in immediate phase change of the 1PPS output signal.

This command has the following format:

```
SERVo:1PPSoffset <int> ns
```

SERVo:TRACe

This command sets the period in seconds for the debug trace output on the USB port. Debug trace data can be used with Ulrich Bangert's "Plotter" freeware utility to display UTC tracking and other parameters versus time.

This command has the following format:

```
SERVo:TRACe <int > [0,255]
```

An example of the generated output is:

```
08-07-31 373815 60685 -32.08 -2.22E-11 14 10 6 0x54
[date][1PPS Count][Fine DAC][UTC offset ns][Frequency Error
Estimate][Sats Visible][Sats Tracked][Lock State][Health
Status]
```

See "SYNChronization:HEAlth?" on page 74 for information on how to decode the health status indicator values. The Lock State variable indicates one of the following states:

Value	State
0	TCXO/CSAC warmup
1	Holdover
2	Locking (TCXO/CSAC training)
4	[Value not defined]
5	Holdover, but still phase locked (stays in this state for about 100s after GNSS lock is lost)
6	Locked, and external GNSS active

SERVo?

This command returns the result of the following queries:

```
SERVo:COARSedac?
SERVo:DACGain?
SERVo: EFCScale?
SERVo:EFCDamping?
```

```
SERVo:SLOPe?
SERVo:TEMPCOmpensation?
SERVo:AGINGcompensation?
SERVo:PHASECOrrection?
SERVo:1PPSoffset?
SERVo:TRACe?
```

CSAC Subsystem

The following commands are used to query the microcontroller built into the optional Chip-Scale Atomic Clock (CSAC) oscillator. These commands are available only on RSR Transcoder 2.0 units equipped with the optional CSAC oscillator. The supported commands include:

```
CSAC:POWer <ON | OFF>
CSAC:RS232?
CSAC:STeer?
CSAC:STATus?
CSAC:ALarm?
CSAC:MODE?
CSAC:CONTrast?
CSAC:LASer?
CSAC:TCXO?
CSAC:SIGnal?
CSAC:HEATpackage?
CSAC:TEMP?
CSAC:FWrev?
CSAC:SN?
CSAC:LIFEtime?
CSAC:STeer:LATch ONCE
CSAC?
```

CSAC:POWer

This command enables/disables the low power mode on the CSAC oscillator. The low power mode is enabled when the power is set to OFF, at which point the CSAC will disable all its internal circuitry and the module will operate as a TCXO based unit. The default setting is ON.

When the CSAC power is OFF, the internal TCXO is used instead of the CSAC as the timing reference for the system. The SERVo:LOOP? query response outputs the current timing reference for the system (TCXO or CSAC).

The format of this command is:

```
CSAC:POWer <ON | OFF>
```

CSAC:RS232?

This query returns the state (OK or FAIL) of the serial communication between the main CPU and the CSAC internal microcontroller. When the state is FAIL, a communication interruption has occurred. The unit should be power cycled to clear the communication error.

CSAC:STeer?

This query returns the current Frequency Adjustment in units of parts-per-trillion (1E-012).

CSAC:STATus?

This query returns the status value in [0,9] of the CSAC oscillator as shown below:

Alarm	Definition
0	Locked
1	Microwave Frequency Steering
2	Microwave Frequency Stabilization
3	Microwave Frequency Acquisition
4	Laser Power Acquisition
5	Laser Current Acquisition
6	Microwave Power Acquisition
7	Heater equilibration
8	Initial warm-up
9	Asleep (ULP mode only)

CSAC:ALarm?

This query returns the CSAC oscillator Alarm value as shown below:

Alarm	Definition
0x0001	Signal Contrast Low
0x0002	Synthesizer tuning at limit
0x0010	DC Light level Low
0x0020	DC Light level High
0x0040	Heater Power Low
0x0080	Heater Power High
0x0100	uW Power control Low
0x0200	uW Power control High
0x0400	TCXO control voltage Low
0x0800	TCXO control voltage High
0x1000	Laser current Low
0x2000	Laser current High
0x4000	Stack overflow (firmware error)

CSAC:MODE?

This query returns the CSAC oscillator mode as shown below:

0x0001	Analog tuning enable
0x0002	Reserved
0x0004	Reserved
0x0008	1 PPS auto-sync enable
0x0010	Discipline enable
0x0020	Ultra-low power mode enable
0x0040	Reserved
0x0080	Reserved

CSAC:CONTrast?

This query returns an indication of signal level, typically ~4000 when locked and ~0 when unlocked.

CSAC:LASer?

This query returns the current (in mA) driving the laser.

CSAC:TCXO?

This query returns the TCXO tuning voltage. The tuning voltage range is 0-2.5 VDC, reported at +/- 10 ppm absolute accuracy.

CSAC:SIGnal?

This query returns an indication of signal level.

CSAC:HEATpackage?

This query returns the physics-package heater power, typically 15mW under normal operating conditions.

CSAC:TEMP?

This query returns the temperature measured by the CSAC unit in °C. Absolute accuracy is +/- 2°C.

CSAC:FWrev?

This query returns the firmware version of the CSAC's internal microcontroller.

CSAC:SN?

This query returns the serial number of the CSAC in the format YYMMCSNNNNN, where YYMM is the year and month of production and NNNNN is the serialized unit of that month.

CSAC:LIFEtime?

This query returns the accumulated number of hours that the CSAC has been powered on since the last factory reset of the RSR Transcoder 2.0 board. The value is stored in the external NVRAM and updated every hour when the CSAC is powered on.

CSAC:STeer:LATch ONCE

This command stores the momentary steering offset into the CSAC internal NVRAM. This is done automatically by the firmware once every 24 hours to avoid exhausting the limited number of write cycles available to the CSAC NVRAM. The user may force this value to be stored into the CSAC by issuing the CSAC:STeer:LATch ONCE command.

CSAC?

This query displays all the CSAC queries defined above.

5

Firmware Upgrade

The following topics are discussed in this chapter:

Introduction

The following sections provide instructions on how to upgrade the RSR Transcoder 2.0 firmware. Please follow the instructions in order to prevent accidentally corrupting the RSR Transcoder 2.0 Flash.

ISP Flash Loader Utility installation

VIAVI recommends using the JLTerm application to upgrade the contents of Flash memory on the RSR Transcoder 2.0. The JLTerm application can be downloaded for free from the VIAVI support page at the link provided below. Alternatively, you can also use the Flash magic utility to perform the same upgrade.

The JLTerm application is available for download at:

https://www.viavisolutions.com/en-us/software-download/jlterm-software

The Flash Magic utility is available for download on the Flash Magic website:

http://www.flashmagictool.com/

Follow the directions given on the website for installing the utility on your computer.

Putting the PCB into In-Circuit Programming (ISP) mode

Two methods are supported for enabling the RSR Transcoder 2.0 in System Programming (ISP) mode. Issuing the SCPI command `SYST:ISP` discussed in "SYSTem:ISP" on page 103 in JLTerm or another terminal program causes the board to reset into ISP mode from normal operation.

If Flash Magic is used instead of the JLTerm, a power cycle is required after programming has finished to restart the board with the updated firmware. Although the SYST:ISP command is recommended, Flash Magic will be useful if the RSR Transcoder 2.0 has stopped communicating.

The board can also be put into ISP mode by shorting-out the ISP# pin 2 of header J5 to pin 1 (Ground) of header J5 of the RSR Transcoder 2.0 board during power-on.

Downloading the firmware

Download the latest version of RSR Transcoder 2.0 firmware as well as JLTerm from the VIAVI support page on the JLT website and store it in a place that will be remembered. The firmware file should be in .hex format.

The unit needs to be connected to the computer's USB serial port prior to firmware download. The RSR Transcoder 2.0 is Micro-USB compatible.

Using the JLTerm programming terminal

1. Download the JLTerm application from https://www.viavisolutions.com/en-us/software-download/jlterm-software.
2. Install and open the JLTerm application.
3. Select the COM port in JLTerm as needed on your PC.

   Once a successful connection is established, the connection icon becomes green, as shown in Figure 19. The RSR Transcoder 2.0 must be in normal operation with working SCPI communication prior to JLTerm connection.

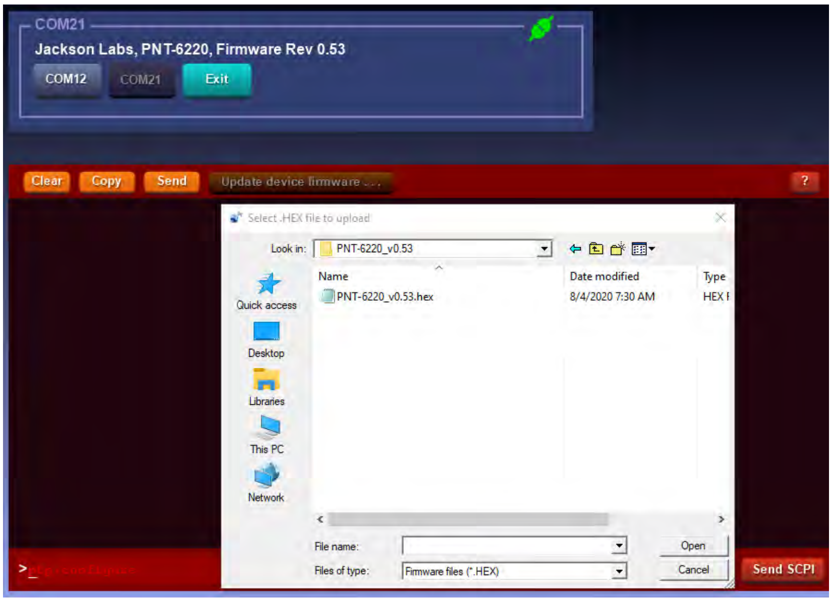
   NOTE

   If there is no valid response from the COM port, check for valid driver and port number for the COM port in the Device Manager on your computer. The COM poer number may be conflicting with another COM port device. Each device should have a different COM port. Ensure GPSCon is not running in the background and using the same COM port.

   Figure 19 Successful connection

   If necessary, change the COM port number, install or update the driver, and unplug and reconnect the serial cable, then try to establish a connection again. In the case of an unsuccessful connection to JLTerm after several attempts, follow the instructions in “Putting the PCB into In-Circuit Programming (ISP) mode” on page 114 and “Using the Flash Magic programming utility” on page 117 for an alternative firmware upgrade method.
4. Once the device is connected in JLTerm, click the orange **Update device firmware...** button.
5. Choose the correct hex file to program the device and click **Open** in the pop-up, as shown in Figure 20. The firmware automatically downloads and the board is reset.

Figure 20 Open Hex file and download firmware



6 If an error occurs during the programming process:

a Check the USB cable connection.

b While the unit is still in ISP mode after the error occurred, follow the steps in "Using the Flash Magic programming utility" on page 117 and attempt to finish the firmware upgrade using Flash Magic.

Figure 21 shows an error in programming.

Figure 21 Error in programming

Using the Flash Magic programming utility

Perform the following steps to use the Flash Magic programming utility.

> **NOTE**
> The following procedure is for Flash Magic version 12.1 or later. For version 11.20 or older, refer to "Using the Flash Magic classic version" on page 118.

1 Put the RSR Transcoder 2.0 into ISP mode, as described in "Putting the PCB into In-Circuit Programming (ISP) mode" on page 114.

2 Open the Flash Magic utility.

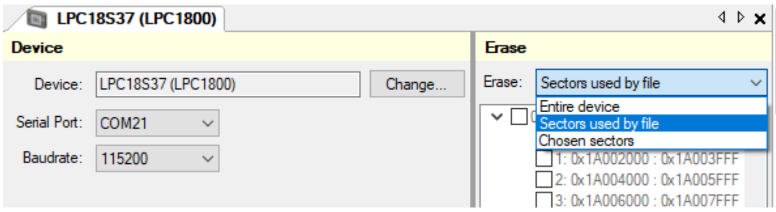
3 Select Device LPC18S37:

  a Click **Change...**

  b Select **LPC1800 > LPC18S37**.

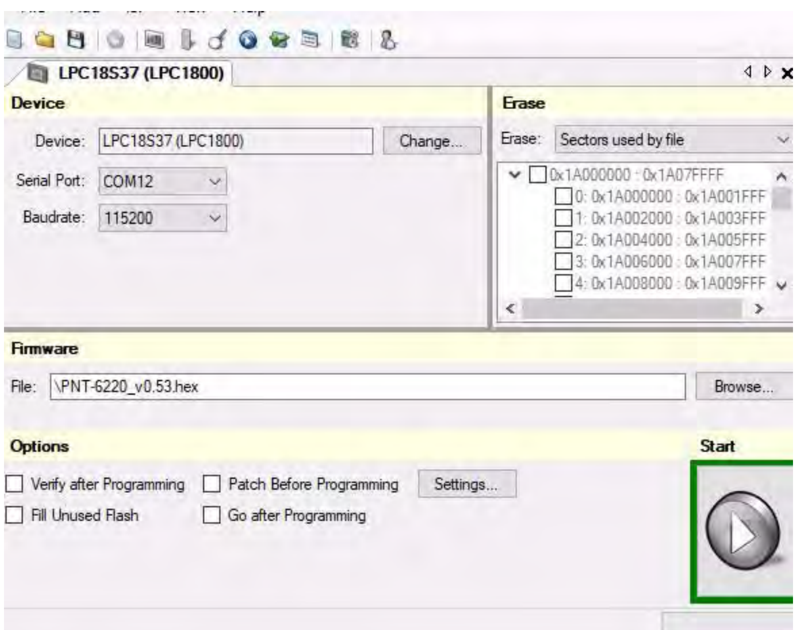
4 Select Erase > Sectors used by file, as shown in Figure 22.

Figure 22 Erase Sectors used by file



5 Change the COM port in the Flash Magic application as needed and select Baud Rate 115200, as shown in Figure 23.

Figure 23 Baud rate



6 Under **Step 3 – Hex File**, browse for the hex file that you downloaded in "Downloading the firmware" on page 114.

7 Check again for the Erase option **Sectors used by file** in Figure 22.

8 In the Start section, press the gray start button .

The firmware is downloaded to the processor.

9 Verify the firmware update as described in “Verifying the firmware update” on page 120.

Using the Flash Magic classic version

> **NOTE**
> The following procedure is for Flash Magic version 11.20 or earlier. For version 12.1 or later, refer to “Using the Flash Magic programming utility” on page 117.

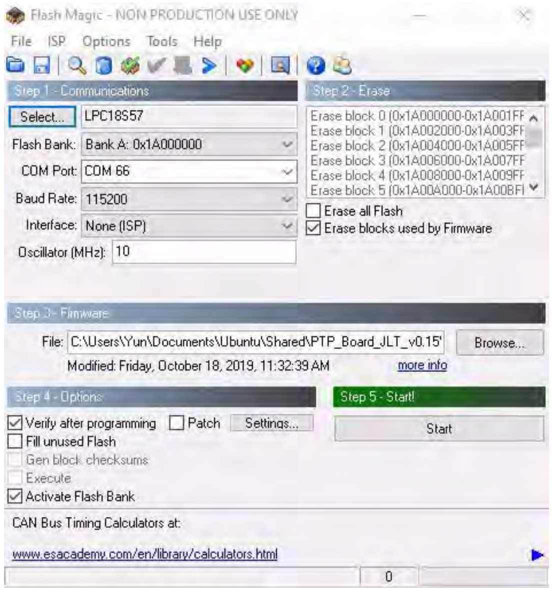
1 Put the RSR Transcoder 2.0 into ISP mode, as described in “Putting the PCB into In-Circuit Programming (ISP) mode” on page 114.

2 Open the Flash Magic application.

3 Set the COM port in the Flash Magic application as needed on our PC.

4 Set **Interface** to **None (ISP)**, as shown in Figure 24.

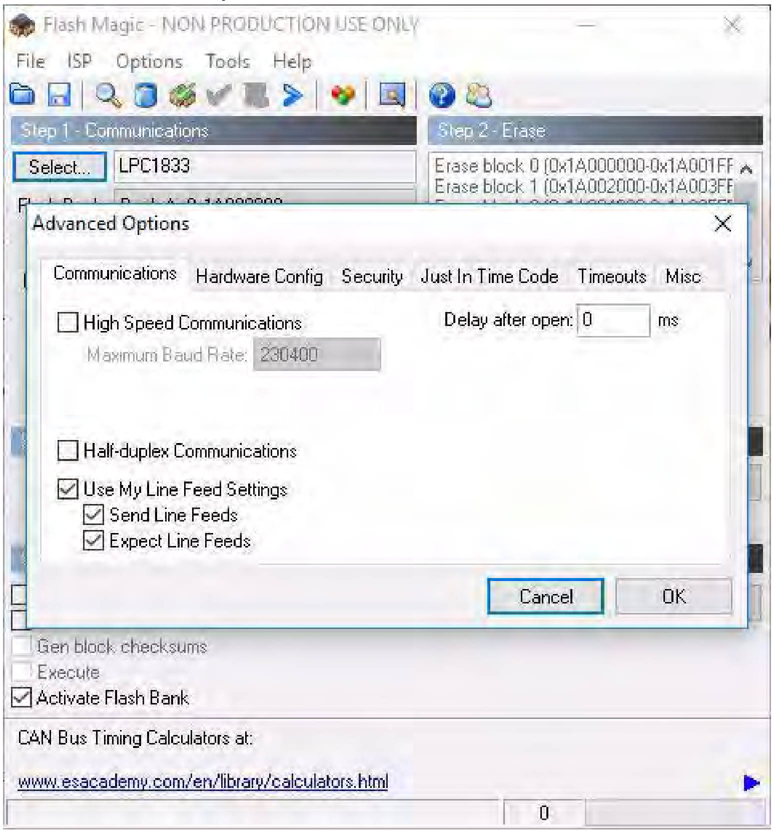
Figure 24 Flash Magic Programming Utility Classic Version



5 Open the **Options** and **Advanced Options** tabs on the application.

6 Ensure that the **Use my Line Feed settings**, Send Line Feeds, and Expect Line Feeds options are all highlighted and selected, as shown in Figure 25.

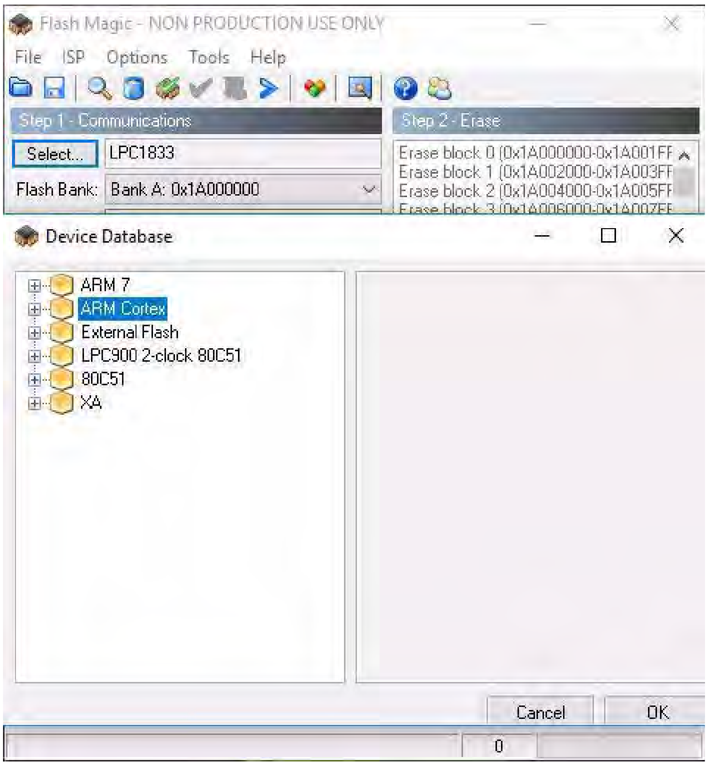
Figure 25 Communications options



7 Click **OK** to return to the main menu.

8 Click the **Select Device** button. The Device Selection window appears, as shown in Figure 26.

Figure 26 Device selection



9 Expand the **ARM CORTEX** folder and select the appropriate processor, in this case **LPC18S37**.

10 Configure the following parameters, as shown in Figure:

- **Baud Rate**: 115200
- **Oscillator (MHz)**: 10.0

11 Check the **Erase blocks used by Hex File** box.

> **ALERT**
>
> Do NOT check the box marked **Erase all Flash**. This will erase factory calibration data, and the unit will not operate and will have to be returned to the factory. Checking this box on the ISP utility will void the warranty.

12 Under **Step 3 – Hex File**, browse for the hex file that you downloaded in "Downloading the firmware" on page 114.

13 Under **Step 4 – Options**, check **Activate Flash Bank**.

14 Proceed to **Step 5** and press **Start**. The firmware is downloaded to the processor.

15 Verify the firmware update as described in "Verifying the firmware update" on page 120.

Verifying the firmware update

Power cycle the unit with the pin 1 of J5 and pin 2 of J5 (ISP#) left floating.

During power on, the unit sends an ID string out of the serial port at 115200 Baud by default. The firmware version can also be queried by sending the *IDN? command. Verify that the firmware version is the version that was downloaded.

22177262
R003, December 2025
English

Viavi Solutions

North America:	1.844.GO VIAVI / 1.844.468.4284
Latin America	+52 55 5543 6644
EMEA	+49 7121 862273
APAC	+1 512 201 6534
All Other Regions:	viavisolutions.com/contacts
email	TAC@viavisolutions.com