STL-2600/STL-1400

Users Guide

R001

About this User Guide

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

Purpose and scope

This manual is intended to help you use the capabilities of the STL-2600 and STL-1400.

This manual includes task-based instructions that describe how to configure, use, and troubleshoot the test capabilities available on your instrument assuming it is configured and optioned to support the capabilities.

Assumptions

This manual is intended for novice, intermediate, and experienced users who want to use their instrument effectively and efficiently. We are assuming that you have basic computer experience and are familiar with basic telecommunication concepts, terminology, and safety.

Related Information

This manual is application-oriented and contains information about using these instruments to test service carried on each of the listed networks. It includes an overview of testing features, instructions for using the instruments to generate and transmit traffic over a circuit, and detailed test result descriptions. This manual also provides contact information for VIAVI's Technical Assistance Center (TAC).

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)

Table 1 Text formatting and other typographical conventions (Continued)

Item(s)	Example(s)
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.
	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 STL-2600 and STL-1400.

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.

Bands of Operation

The equipment operates on the following bands.

Table 4 Bands of operation

Receiver	Frequency Range
GNSS[1]	1164 MHz to 1300 MHz 1559 MHz to 1610 MHz
STL	1610 MHz to 1626.5 MHz

1. GNSS is only available on the STL-2600. Contact your VIAVI representative for more information.

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

Technical assistance

If you require technical assistance, call 1-844-GO-VIAVI. For the latest TAC information, go to https://support.viavisolutions.com.

Table of Contents

1

Introduction

This chapter discusses the following topics:

About the STL-2600/STL-1400

The VIAVI STL-2600/STL-1400 is a receiver built to receive Low Earth Orbit (LEO) satellite signals, in particular Satelles-STL signals transmitted by the Iridium constellation. The STL signals allow Positioning, Navigation, and Timing (PNT) solutions anywhere on earth to be generated within minutes after power-on, and completely independent of any GNSS signal. STL signals are subscription-based, and allow assured PNT performance in environments where GNSS signals cannot perform. The STL-2600/STL-1400 receiver excels at this due to the 1,000x (30dB) higher signal power and inherent constellaion spoofing resilience of the STL signals used. The STL-2600/STL-1400 board is available in a commercial Timing-only variant, and a commercial Timing and Positioning variant. The board is optionally available with either one or two (default configuration) STL processors allowing the user to scale power-consumption/cost versus number of “channels” (Bursts Per Minute - BPM) tracked per minute. The STL-2600/STL-1400 board is particularly well suited to fulfill the recent Executive Order 13905 mandates that require a GNSS-independent national timing capability, and with a mature LEO non-GNSS satellite constellation, is deployable today anywhere in the world.

The STL-2600/STL-1400 receiver can run in stationary (Position Hold mode) and optionally in 3D Mobile mode. It uses UTC(STL) derived reference timing signals to discipline a built-in ultra-high-stability TCXO local oscillator to better than 1ppb frequency accuracy typically, and with typically better than 100ns rms phase error to UTC(USNO). It supplies a disciplined, low phase-noise 10MHz output as well as a 1PPS pulse derived from the 10MHz oscillator and synchronized to UTC(STL). The STL-2600/STL-1400 board provides industry-standard NMEA PNT output sentences, and thus can be a drop-in replacement for legacy GPS receiver products. The board includes both dual-TTL-level and a USB serial control port for NMEA and SCPI communication, and a third serial port to optionally control an externally-connected atomic holdover oscillator. The board can optionally be fully powered and controlled through its Micro-USB connector. The board usually operates from a single +3.3V supply and offers a low power mode that operates at less than 1.1W. Alternatively the board can be operated in power saving mode with further power reduction discussed in “SYSTem:PWRSAVing:MODE [OFF|DUTY-CYCLE|RF-RECEIVER|SECOND-MCU|GNSS|ALL]” on page 85. Complete communications and powering the board can be achieved by connecting the on-board Micro-USB port to a PC. The board also includes a high-performance commercial 8th-gen GNSS receiver that can acquire and track up to 72 GNSS signals down to a state of the art –167dBm, supports concurrent GNSS reception of up to three GNSS constellations plus SBAS, and that allows auto- or manual switch-over between GNSS and STL signals depending on GNSS signals being jammed/spoofed, or per user-set priority preference. One or optionally two powered antenna input connectors can be used to supply the Satellite signals to the receivers through one or two diversity antennae with these connectors also providing power to the external antennae.

Additional capability is available in the STL-2600/STL-1400receiver by upgrading the frequency/phase-noise/holdover performance of the unit gluelessly by using an optional external oscillator. The unit supports either a Microchip or Teledyne CSAC oscillator, a Microchip or SpectraTime Rubidium oscillator, or high-end DOCXO analog oscillators.

> **NOTE**
> A separate analog high-resolution DAC output voltage connector supplie EFC steering voltage to an externally connected analog OCXO for this purpose.

Connecting one of these high-end oscillator solutions to the STL-2600/STL-1400receiver creates a state-of-the-art GNSS/STL-disciplined Timing and Frequency reference at a fraction of the cost of lesser competitive solutions that do not offer GNSS-independent operation, and industry-leading holdover performance in completely GNSS/STL denied operation. External oscillators can be connected with as little as three wires (ground, 10MHz, EFC OCXO steering voltage).

The STL-2600/STL-1400 board can either be designed-into a customers' PCB as a plug-in module using a Samtec board-to-board connector, or it can be fully operated by just plugging in the Micro-USB cable. The USB cable supplies 1PPS and NMEA PNT information transferred via the DCD signal and the serial port in the USB connection. The unit is thus fully GPSD and NTP compliant under Linux. The unit can also be supplied with a cable harness that provides power and connectorized access to a number of I/O signals and thus allows easy evaluation and design-in. The unit can be powered by both the 3.3V external power supply, or the USB power, or by both of these two supplies at the same time, and it can be controlled by the freeware GPSCon Windows application available on the Support page of the JLT website. Other third-party NMEA-compatible monitoring programs are also usable, such as the freeware uBlox uCenter console program. The STL-2600/STL-1400 receiver is fully in-field, in-situ software updatable through the free JLTerm application program also available on the support page of the JLT website, or through a JLT-supplied command line program.

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. Jackson Labs Technologies, Inc. assumes no liability for the customer's failure to comply with these requirements.

Antenna lightning protector

Always use a UL approved and properly installed Iridium/GNSS Antenna Lightning protector on the coaxial antenna feed(s) to prevent damage, injury, and/or death in case of a lightning strike.

Grounding

To avoid damaging the sensitive electronic components in the STL-2600/STL-1400 always make sure to discharge any built-up electrostatic charge to a good ground source, such as power supply ground. This should be done before handling the circuit board or anything connected to it, i.e. the L-band antennae.

Power Connections

Make sure to connect the DC power to the device following the polarity indicated in "Pre-installation" on page 6. Do not reverse the power pins as this will cause serious damage to the circuit board.

Environmental Conditions

This instrument is intended for indoor use. It is designed to operate at a maximum relative non-condensing humidity of 95% and at altitudes of up to 50,000 meters. Refer to the specifications tables for the DC voltage requirements and ambient operating temperature range, as well as the requirements for the optional external AC Wall Wart power supply.

Bands of Operation

Table lists the bands of operation.

Table 1 Bands of operation

Receiver	Frequency Range
GNSS[1]	1164 MHz to 1300 MHz 1559 MHz to 1610 MHz
STL	1610 MHz to 1626.5 MHz

1. GNSS availability is module dependent. Contact your VIAVI representative for more information.

2

Quick-Start Instructions

The following topics are discussed in this chapter:

Overview

The following sections describe how to install the STL-2600/STL-1400.

Pre-installation

Ensure the following materials and resources are present before proceeding:

Delivered with module (refer to packing list):

- STL-2600/STL-1400 Receiver Module
- SMA to U.FL adapter cable

Optional accessories:

- Multi-connector power/comm harness (includes AC/DC adapter)
- Tallysman GNSS/Iridium active antenna
- 4-meter SMA antenna cable
- Micro-USB serial communication cable (STL-2600 only)
- USB-to-TTL serial communication cable

Provided separately (by user):

- Windows PC (required to support firmware upgrades and monitoring/control)
- Frequency/time measurement equipment as required for evaluation
- RS232-to-TTL or USB-to-TTL serial adapter (if communicating to module UART from a PC)
- 50-ohm BNC interface cables for time/frequency signals (if measuring time/frequency outputs, or supplying time/frequency reference inputs)

Powering the Unit

The following procedure describes how to power the unit.

To power the unit

1 Connect a 3.3V compatible active Iridium antenna to U.FL (UMCC) connector J4.

> **NOTE**
> The antenna should be an active type with at least 25dB gain. The antenna can also be a combination-antenna that supplies GNSS signals and STL signals at the same time to the on-board GNSS receiver on the STL-2600/ STL-1400 module. JLT recommends the GNSS/Iridium combo antenna Tallysman PN: 2643A or the Iridium-only Tallysman antenna PN: 33-2600A-07 available at Digikey and other vendors. For better indoor STL reception, VIAVI also recommends Iridium-only Tallysman helical active antenna 33-HC610. The STL-2600/STL-1400 board will supply 3.3V power to the antenna.

2 Connect +3.3V (+/-0.15V) DC Power to J3 pins 2 and 4 on the unit, and ground to pin 1 and pin 5 of J3.

> **NOTE**
> The Samtec connector J3 indicator-notch indicates pin 20, not pin 1 by Samtec default. Pins 1 and 2 are marked in Silkscreen for reference on the bottom side of the PCB right next to connector J3.

The unit will consume up to 1.6W of power, or the equivalent of up to 510mA when operated at 3.15V.

3 Alternatively the unit can also be controlled and powered by connecting the Micro-USB serial port to a PC.

> **NOTE**
> Overall power consumption and heat generation is slightly higher when powering the unit only through the Micro-USB connector.

The unit will now lock to the STL signals (the Green LED D5 will blink every time an STL packet has been received and has been successfully processed). Combo bi-color LED D23 will indicate proper lock and no events pending when the Green LED lights up, and will blink Red whenever the board is generating a UTC(STL or GNSS as selected by the user with STL set as default) 1PPS output pulse. Once the green LED D23 is on, the unit will output 10MHz with significantly better than 1ppb frequency accuracy. Green LED D21 is connected to the 1PPS raw output of the on-board GNSS receiver and indicates whenever the GNSS receiver has achieved a 3D Fix.

Green LED D18 indicates when the unit is tracking an externally-supplied 10MHz signal by pulling J3 Pin 8 to ground. This is usually connected to, and driven by the BIST (BITE) output of an externally-connected CSAC oscillator and indicates when this

CSAC oscillator has achieved atomic lock. Until atomic lock is achieved the internal TCXO will be controlled by the modules' EFC DAC and loop software.

NOTE

When tracking STL, it can take several minutes or in worst-case conditions tens of minutes to achieve a first 1PPS UTC output pulse as indicated by Red LED D23 - this is normal. The time-to-first-1PPS-fix can be shortened by increasing the PFOM and TFOM thresholds described in "RECEiver:STL:PPSFilter:PFOM <int> [1,9]" on page 62 and "RECEiver:STL:PPSFilter:TFOM <int> [1,9]" on page 63 required for the unit to start generating a 1PPS UTC(STL) fix.

NOTE

If the reference oscillator frequency is unstable or reference CSAC oscillator is not locked yet, any large frequency jump may confuse the STL Kalman Filter and result in longer time to achieve first 1PPS UTC(STL) fix. See "Initial Operation" on page 25 for detailed operation with external oscillator.

To avoid confusing the STL Kalman Filter, the J3 Pin 8 should not be pulled low until reference frequency is stable or delayed STL Kalman Filter processing can be configured with RECEiver:STL:KFPROCessing <ON|OFF|DELAYed> command.

NOTE

The GPS receiver establishes the internal antenna gain immediately after power-on. For proper operation the GPS antenna should always be connected prior to turning on the modules' power. Alternatively the internal GNSS receiver can be manually forced into reset and therefore to re-measure the antenna gain after the antenna has been connected with the RECEiver:GNSS:RESET <ONCE|HOLD|OFF> command.

Connect a terminal program (the latest version of TeraTerm is recommended as older versions of TeraTerm do not support baud rates higher than 115.2K) to the unit via the SCPI TTL serial connector pins 16 and 18 on connector J3, or to the Micro-USB serial port - both with 8 bit, 1 stop bit, no parity, and no flow-control. The default baud rate for USB serial port is 921,600, and the default baud rate for the SCPI TTL serial port is 115,200.

The unit can also be controlled, monitored, and performance-plotted via the free application program GPSCon available on the support page of the JLT website. The latest version of GPSCon now supports up to 921,600 baud. For more information about GPSCon and how to use it, please refer to Chapter 5.

WARNING

Do not connect RS-232 serial levels to connector J3. Doing so will damage the unit from RS-232 serial levels on connector J3, as the default configuration is TTL/LVCMOS levels only.

Both the TTL serial port J3 and the Micro-USB port operate orthogonally, so the user may send and receive commands to/from both ports at the same time. Try some of these SCPI commands:

- `HELP?`

- `SYSTem:STATus?`
- `RECEiver?`
- `SYNChronization?`
- `DIAG?`

PCB photos

Figure 1 shows the STL-2600 PCB.

Figure 1 STL-2600 PCB

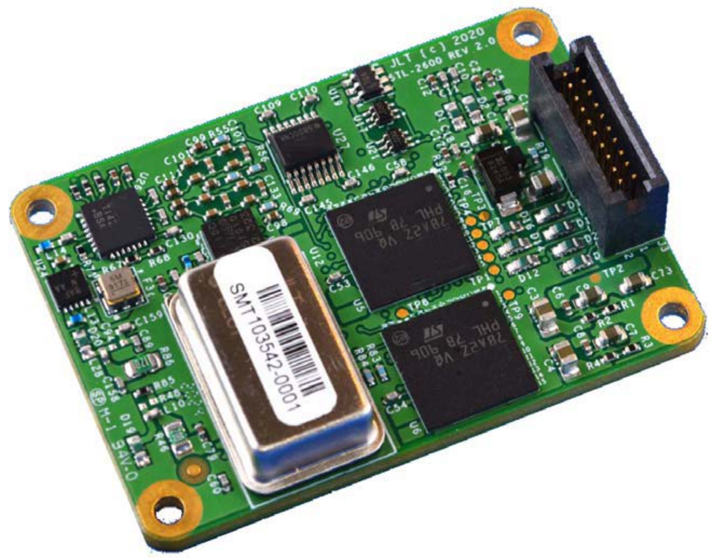
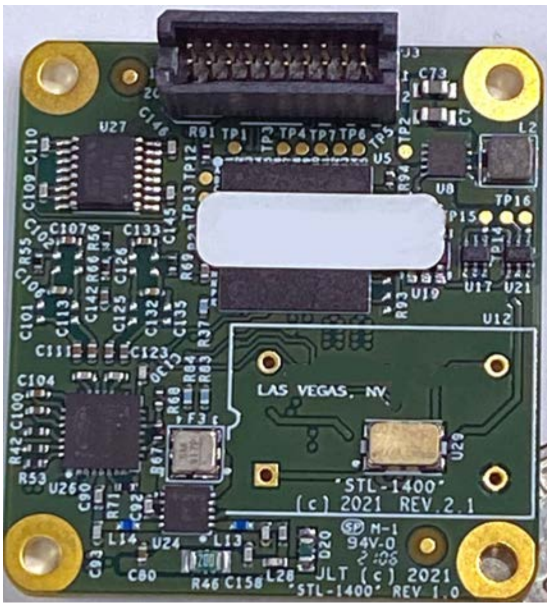


Figure 2 shows the STL-1400 PCB.

Figure 2 STL-1400 PCB



Mechanical drawings

The following drawings and table show the mechanical dimensions and the hardware connections of the STL-2600 and STL-1400 PCBs.

Figure 3 STL-2600 mechanical dimension top view

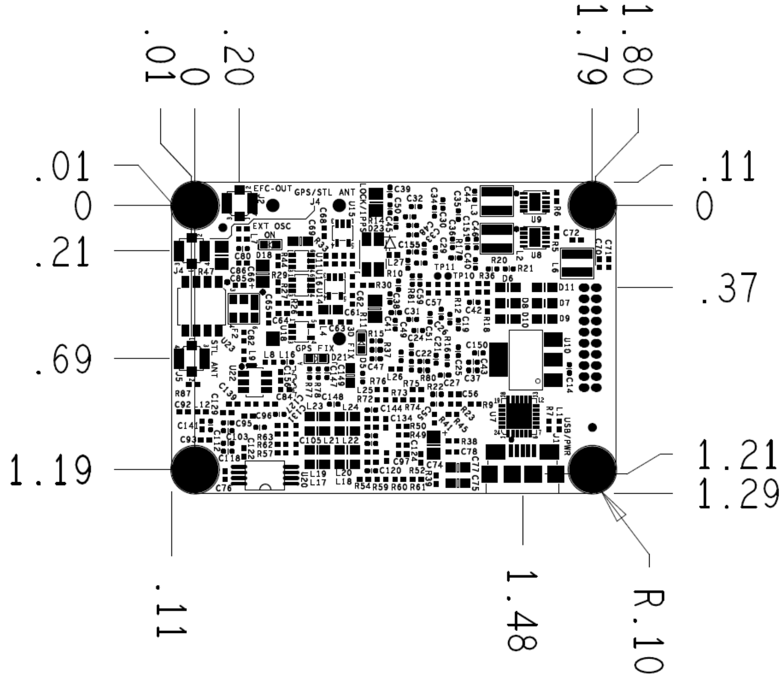


Figure 4 STL-2600 perspective mechanical drawings

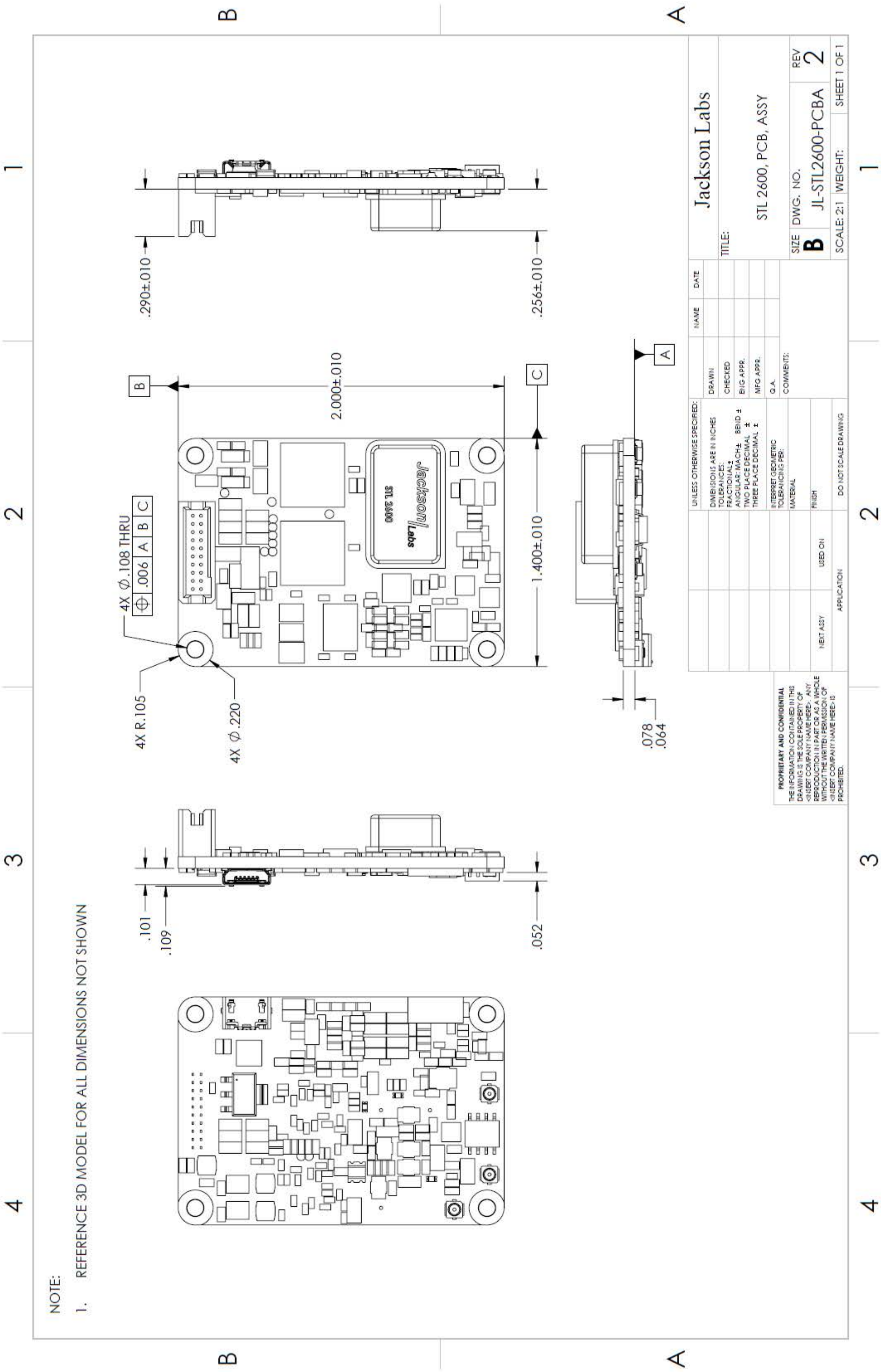


Figure 5 STL-1400 Mechanical Drawing Top View

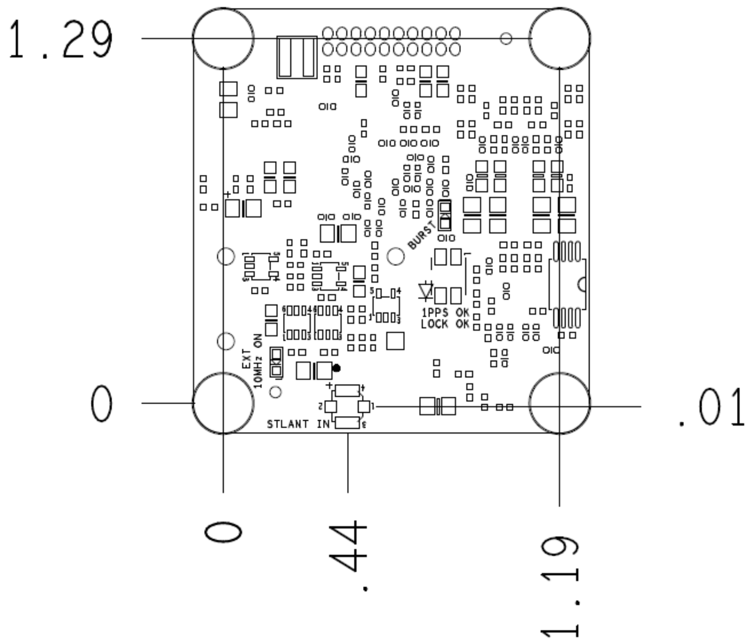
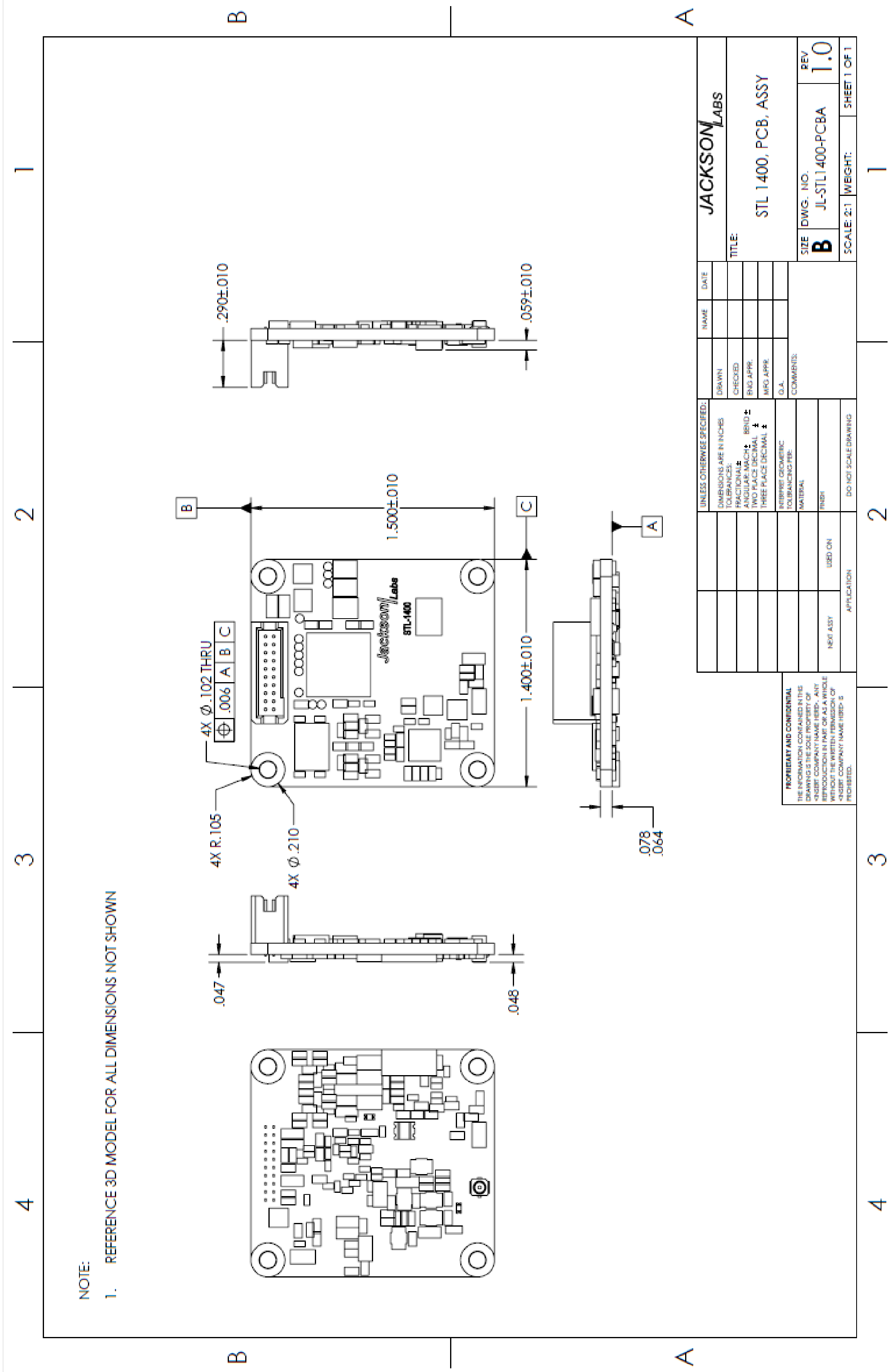


Figure 6 STL-1400 perspective mechanical drawings



Sec. 2, page 14 shows the STL-2600/STL-1400 hardware connectors.

Table 2 STL-2600/STL-1400 hardware connectors

Ref	Name	Function	Specification	Description
J3 Pin 1	Ground	Ground	Ground	Prime Power and 10MHz output Ground Return. All ground pins on connector J3 are connected together on the PCB, and can be used interchangeably
J3 Pin 2	3.3V Power In	+3.3V Prime Power Input A	3.3V +/- 0.15V, <0.55A	3.3V Prime Power Input, used to power the module if USB power is not available. Using the 3.3V Prime Power input will result in the overall lowest power consumption compared to USB power. **Note –** This pin will carry 3.15V on it if the unit is powered by the USB connector and will back-feed power into the user's PCB if the user-applied voltage is < 3.2V on the prime-power pins 2 and 4 of connector J3.
J3 Pin 3	10MHz OUT	10MHz LVCMOS output from internal TCXO	3.3V LVCMOS, <1ns Rise/Fall time typically, Internally buffered and series-terminated to 50 Ohms	10MHz LVCMOS output from internal TCXO with low phase-noise and jitter. Disciplined to UTC(STL/GNSS). While this signal can drive 50 ohms coax cables, do not end-terminate this signal with less than 1K Ohms

Table 2 STL-2600/STL-1400 hardware connectors

Ref	Name	Function	Specification	Description
J3 Pin 4	3.3V Power In	+3.3V Prime Power Input B	3.3V +/- 0.15V, <0.55A	3.3V Prime Power Input, used to power the module if USB power is not available. Using the 3.3V Prime Power input will result in the overall lowest power consumption compared to USB power. Pins 2 and 4 are internally connected to each other. **NOTE** – This pin will carry 3.15V on it if the unit is powered by the USB connector and will back-feed power into the users' PCB if the user-applied voltage is <3.2 on the prime-power pins 2 and 4 of connector J4.
J3 Pin 5	Ground	Ground	Ground	Prime Power and 10MHz output Return
J3 Pin 6	10MHz Input	Optional 10MHz LVCMOS Reference Input	10MHz input, 3.3V LVCMOS input with 5V compatibility <10ns rise/ fall time recommended. 0V to 5.0V min/max. 10MHz +/-5Hz lock-range	Used to connect an external CMOS 10MHz input signal from an oscillator that should be more stable and accurate than the internal 10MHz TCXO. The internal TCXO is phase-locked to this reference signal when Pin 8 (CSAC-LOCK#) is pulled to Ground. The module can discipline the external 10MHz oscillator either through serial commands on pins 12 and 14 of J3, or through an analog steering voltage on connector J2 to assure the externally-supplied 10MHz reference is accurate, stable, and UTC-disciplined

Table 2 STL-2600/STL-1400 hardware connectors

Ref	Name	Function	Specification	Description
J3 Pin 7	1PPS OUT	1PPS UTC(STL/ GNSS) Output	3.3V LVCMOS. <5mA	UTC-aligned 1PPS output generated by internal 10MHz oscillator signal, and aligned to UTC via either STL or GNSS received signals. This is the main 1PPS UTC output signal of the unit. This signal is generated during holdover by either the internal or the optional external 10MHz reference. This signal is also connected to the Red LED D23, and will not become active until a user-adjustable PFOM threshold is reached in STL reception mode.
J3 Pin 8	8 CSAC LOCK#	CSAC LOCK#	0V to 3.3V. Pulled internally to +3.3V via 4.7K resistor	A 0V signal on this pin enables the external 10MHz reference input to the module, and switches the internal TCXO to phase-lock to the external 10MHz reference signal rather than follow the internal DAC control voltage. The user must ensure that a stable and accurate 10MHz signal is applied to pin 6 of connector J3 prior to or simultaneously to pulling this pin low. This pin can be connected to Ground, be driven by a GPIO pin on the customers' PCB, or optimally be connected to the BITE/BIST output signal of externally connected CSAC or Rubidium oscillators. The latter case will allow the module to work from the internal 10MHz TCXO signal as long as the external CSAC/Rubidium has not achieved atomic lock or if the oscillator indicates a fault, and will automatically switch-over when the external CSAC achieves atomic lock as indicated by its BITE/BIST output signal. Leave floating or pull to 3.3V when no external 10MHz reference is to be used.

Table 2 STL-2600/STL-1400 hardware connectors

Ref	Name	Function	Specification	Description
J3 Pin 9	1PPS IN	Optional external 1PPS Input	3.3V LVCMOS, pulled to Ground via internal 4.7K resistor. Rising-edge aligned. <10ns rise-time recommended	Optional external 1PPS reference input. Internal oscillator can be locked to this reference input via SCPI-controlled disciplining algorithm
J3 Pin 10	GPS 1PPS OUT	Raw 1PPS Output from internal GNSS receiver	+3.3V LVCMOS, Rising-edge UTC(GNSS) aligned, <5mA	This UTC(GNSS) 1PPS output is directly connected to the internal uBlox GNSS receiver 1PPS output and Green LED D21. It is a rising-edge UTC(GNSS) aligned 3.3V signal that is gated by the GNSS receiver and indicates the GNSS receiver has achieved a 3D Fix
J3 Pin 11	LOCK_OK and NO_EVENT#	LOCK OK and EVENT#	0V or 3.3V CMOS	3.3V: Unit Locked to either STL, GPS, or external 1PPS reference input (selected via SCPI command) and no events are pending 0V: Event is pending (events can be queried via the SYNC:HEAlth? command) or units' phase/ frequency is out of bounds. Connected to Green LED D23 in parallel
J3 Pin 12	CSAC TX	TX Serial Input FROM optional external CSAC	0V to 3.3V TTL serial signal. Internally pulled to 3.3V via 4.7K resistor. Do not drive with 5V CMOS levels.	This is connected to the serial output from an externally connected CSAC or Rubidium oscillator. It is connected to the module processor's serial RX input
J3 Pin 13	ENTER_ISP	Enable ISP Mode	Pull to 3.3V during Reset to enter In System Programming (ISP) mode. Internally pulled to Ground via 4.7K resistor	Pull this signal to 3.3V during Reset or Power-On to enable the internal MCU to enter ISP mode manually for in-situ firmware upgrade programming.
J3 Pin 14	CSAC RX	Serial output to optional external CSAC RX pin	3.3V LVCMOS, <5ns rise-time	Connect this output signal to TTL level RX serial inputs on optional externally-connected CSAC or Rubidium oscillators. This signal is a driven output from the internal MCU and can drive <5mA

Table 2 STL-2600/STL-1400 hardware connectors

Ref	Name	Function	Specification	Description
J3 Pin 15	RESET#	External RESET# input	Pull to ground to Reset the module. Internally pulled to 3.3V via 100K resistor	Pull this pin to Ground via an open-collector output or a switch to reset the MCU on the module. Can also be used to force the module into ISP mode by pulling pin 13 of J3 to 3.3V while this reset pin is cycled to Ground. Leave open if not used.
J3 Pin 16	SCPI TX	SCPI TTL Serial OUTPUT	3.3V LVCMOS Output	Main SCPI TX control output from the MCU. Can also generate NMEA messages on this pin
J3 Pin 17	Ground	Ground	Ground	Ground
J3 Pin 18	SCPI RX	SCPI TTL Serial INPUT	0V to 3.3V max LVCMOS input. Do not drive with 5V levels.	Main SCPI RX serial input to the module. Accepts SCPI control commands
J3 Pin 19	Ground	Ground	Ground	Ground
J3 Pin 20	Ground	Ground	Ground	Ground

Table 2 STL-2600/STL-1400 hardware connectors

Ref	Name	Function	Specification	Description
J1	Micro-USB	Micro-USB power and serial port	3.6V to 5.5V Micro-USB Power and Control	The Micro-USB port allows power, SCPI serial control, and NMEA output from the module. Default baud-rate is 921,600 baud, 8N1. The modules' power consumption when using only USB connector J1 to power the board is slightly higher then when also applying 3.3V as the unit has an internal 5V to 3.15V LDO regulator that generates additional heat in this mode. This heat is not generated and the unit will not draw any current from the USB port if the board is also powered from an external 3.3V Prime Power on connector J3. The Micro-USB connector is diode-ored to prevent back-feeding of power into the Micro-USB cable. **NOTE –** The unit provides 3.15V to the prime power pins whenever USB power is applied and the user-supplied voltage on the prime power pins is <3.2V. Therefore, if the unit is powered through USB and plugged-into a users' PCB, that PCB will receive 3.15V power from pins 2 and 4 of connector J3. The module is not designed to power external circuitry when powered from the USB port, and care should be taken to avoid operating the unit with external circuitry connected to the 3.3V Prime Power pins when external power is not supplied to the Prime Power pins.

Table 2 STL-2600/STL-1400 hardware connectors

Ref	Name	Function	Specification	Description
J2	EFC OUT	EFC output voltage to OCXO	0V to 3.0V Frequency Steering output to external OCXO/TCXO. The input impedance of the external OCXO is recommended to be >40K. U.Fl (UMCC) connector type.	This pin supplies the EFC steering voltage to an externally-connected analog oscillator. It is driven by the modules' DAC output from a high-stability 3.0V reference. This output voltage controls the optional externally-connected analog oscillator frequency to steer the external OCXO onto UTC via the JLT software disciplining loops. Leave open if not used.
J4	STL/GNSS Antenna IN	Combined GNSS and STL antenna input (default factory configuration) or GNSS-only antenna (custom factory option)	L1 RF input with 3.3 V antenna supply. Supplies up to 60mA at 2.7V to an external STL/GNSS antenna such as Tallysman PN: 2643A or 2600A or HC610. This is the default-configuration antenna input for both STL and GNSS signals. U.Fl (UMCC) connector type. GNSS-only antenna input as a factory configuration option.	L1 1574MHz to 1626MHz input. Antennae with >25dB RF gain and less than 1.5dB NF are recommended. This combined STL/GNSS antenna input is the factory-default setup, and the RF signal is internally split-up, amplified, and SAW-filtered for the two different L1 receivers on the module (STL and GNSS). This is the GNSS-only antenna input for boards configured in the default dual-antenna connector input mode at the factory.

Table 2 STL-2600/STL-1400 hardware connectors

Ref	Name	Function	Specification	Description
J5	Optional STL-Only Antenna IN	Optional STL-Only antenna input connector. Requires custom factory-configured option.	L1 RF input with 3.3V antenna supply. Supplies up to 45mA at 2.7V to an external STL antenna such as Tallysman PN: 2643A or 2600A. U.FI (UMCC) connector type. REMAINS UNUSED on factory-configured boards.	This is the optional external STL antenna input from the recommneded Tallysman PN: 2643A antenna if the unit is factory-configured to accept separate GNSS and STL antennae feeds for diversity-antenna function. 1620MHz to 1626MHz input. **REMAINS UNUSED ON BOARDS DELIVERED IN STANDARD FACTORY CONFIGURATION.** Antennae with >25dB RF gain and less than 1.5dB recommended. When the board is configured for dual STL/GNSS input on connector J4 (factory default) this connector is internally disconnected from the receivers, but is still useful to optionally ouput 3.3V power to an external antenna amplifier etc.

Prime connector J3

The 20-pin main connector J3 is a Samtec connector Part Number: TFM-110-11-H-D with a height of 0.29 inches above the PCB. See Figure 2.3 for more details.

The following parts are recommended to mount the module onto a users' PCB:

- 8pcs M2 Steel Washer, Flat, Digikey PN: H766-ND
- 4pcs Hex Nut, 0.157" M2, Digikey PN: H761-ND
- 4pcs8mm Hex Stand-Off, M2, Mouser PN: 710-971080244
- 4pcs M2 screw 5mm, Digikey PN: H738-ND
- Mating Female connector: Samtec SFM-110-02 series

A screw-retaining compound such as those typically manufactured by Loctite is recommended to secure the screws and nuts in place.

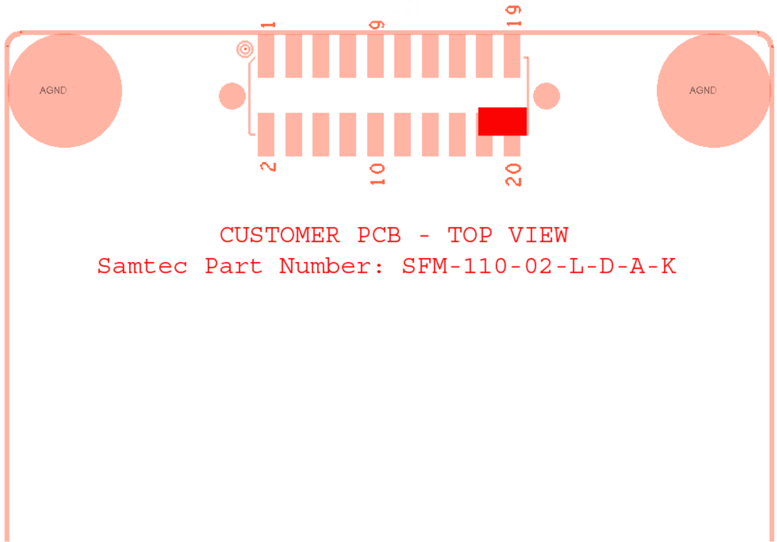
A suitable mating connector is Samtec Part Number: SFM-110-02-L-D-A-K.

Figure 7 shows a customer PCB layout with SFM-110-02 connector.

> **NOTE**
>
> The same SAMTEC connector TFM and SFM series have a mechanical key indicator at Pin Location 20 and not PIN-1. This is non-standard and can lead to assembly of the connector onto a PCB in the wrong direction.
>
> Pin 1, 2, and 19,20 are clearly marked in Figure 7 which depicts how a PCB should be laid out when using the SFM-110-02 SMT type mating female connector (top view.)

> **NOTE**
>
> The large red square indicates the location of the keying notch in the SAMTEC connectors.

Figure 7 Customer PCB Mating (Female) Connector Layout. TOP VIEW



Notes on signal interfacing

The optional LOCK_OK signal on J3 pin 11 is a 3.3V LVCMOS output requiring a series resistor of typically 390 to 470 Ohms when used to drive an external LED.

The serial port RX and TX lines cannot be directly connected to a typical RS-232 level DB-9 connector as these are driven with TTL signal levels and doing so would damage the board. Connecting these pins to an RS-232 serial interface may seriously damage the board.

The SCPI and NMEA outputs can be sent to either the SCPI TTL or the USB serial ports, as both ports operate orthogonally to each other. The CSAC TX and RX lines are designed to directly connect to the CSAC RX and TX signals on the optional external CSAC or Rubidium oscillators. Drivers are available for optional Microchip, Teledyne, and Spectratime atomic clocks.

Connect an active or passive L1 Iridium/GNSS antenna that is compatible to 2.5V to 3.3V antenna power to connector J4. Connect this antenna prior to turning-on the power to the board. Use a properly installed lightning arrestor on the antenna feed for safety and to prevent damage, injury, or death in case of a lightning strike.

For units that are optionally factory-modified to separate the Iridium and GNSS antennae to two separate antennae connectors: use J4 for the GNSS antenna and J5 for the Iridium antenna. Use only active Iridium antennae with at least 25dB antenna gain and 1.5dB NF or less.

Coaxial connectors

The RF antenna connectors and the EFC Analog OCXO steering output connector are U.FL (UMCC) types. Suitable coax U.FL to SMA connector adaptor cables are readily available from sources such as Digikey or Mouser etc. Taoglass cable PN: CAB.011 works well for example.

Power

Power can be provided by an external 3.3V DC power supply, the Micro-USB connector, or both at the same time. The unit is typically powered from a +3.3VDC source Prime Power with +/-0.15V allowable range. The current is typically less than 0.55A at 3.3V. Connect a clean +3.3V power supply to pin 2 and pin 4 of J3, and ground to any two of the ground pins on J3 as all ground pins are connected together on the PCB. Both USB power and external +3.3V power can be connected at the same time to the unit. Running with +3.3V applied at any time will result in lower power consumption compared to running only with USB 5V power. The current when running just from USB power is also about 0.55A max, hence since the USB voltage is higher the overall power consumption is higher when running from USB power only.

NOTE

The +3.3V power pins on connector J3 are diode-ored with a regulated +3.15V power supply derived from the USB connector 5V input. The unit will inject 3.15V into a users' PCB through the two power pins on connector J3 if USB power is applied, and the users' 3.3V power supply is turned-off. As the USB 3.15V power circuit is diode-protected, the USB power supply will be disabled as soon as the user applies greater than +3.2V to the prime power pins of connector J3, and thus no USB power is consumed when the unit is also powered by an external +3.3V power supply on pins 2 and 4 on connector J3. The unit is not designed to supply power from the USB connector to external circuitry through the 33V Prime Power pins, and may overheat if current is drawn this way.

The unit provides 3.3V via 10 Ohms series over-current protection resistors to the antenna connector J4 when running from external 3.3V power, and ~3.15V when running from USB power only.

Connecting the Combined GNSS/Iridium Antenna

Connect the STL/GNSS antenna to the U.FI connector J4.

CAUTION

Use Lightning Arrestor on your Antenna setup. Use an amplified antenna that is 3V LNA compatible such as the Tallysman 2643A part number, or the Iridium-only Tallysman 33-2600A-07 or 33-HC610 part number. Both Tallysman 2643A or 33-2600A-07 are available from Digikey.

The STL-2600/STL-1400 module includes an antenna splitter, SAW filters, and RF amplifiers that take the single antenna signal on connector J4 and internally split this up to feed the GNSS and STL receivers on the pcb. This together with the SAW filters located inside the typical amplified antennae from Tallysman improves adjacent channel rejection and increases overall sensitivity while at the same time allowing loss from substantial antenna cable lengths to be compensated.

The STL-2600/STL-1400 receiver is capable of generating standard navigation messages (for example, see RECE:GPGGA, and RECE:GPRMC serial commands in Chapter 3) that are compatible with most GPS based navigation software.

The receiver generates two 1PPS time signal that are phase synchronized to UTC(STL) and UTC(GPS). These 1PPS signals are used to frequency-lock the 10MHz output of the STL-2600/STL-1400 GPSDO to UTC, thus disciplining the unit's 10MHz frequency output to the US Naval master clock for very high frequency accuracy (typically better than 10 digits of frequency accuracy when locked to GPS or STL). On the main 1PPS output the user can select which domain the 1PPS is generated from and which domain is used for oscillator disciplining (GNSS or STL) with a planned auto-switchover feature if one domain loses lock. The GNSS 1PPS secondary output signal is directly driven by the internal uBlox GNSS receiver, and operates independently from the main 1PPS system output. The user can further select if the main 1PPS signal is generated by the 10MHz reference oscillator and thus phase-stable (GPSDO mode) or if the 1PPS output is allowed to phase-jump second to second just like a GNSS receiver does, and thus provide the lowest average phase offset to UTC(STL/GNSS) possible at the expense of phase jitter. The latter mode is called GPS Mode versus GPSDO mode. In GPSDO mode the unit keeps the 1PPS output tightly-aligned to the 10MHz output, and slowly adjusts the phase of the 10MHz clock back and forth to keep the 1PPS output aligned to UTC. This adjustment can take minutes to hours, but results in extremely low second-to-second phase jitter. At the same time the unit can be queried for the UTC to OCXO phase offset using the SYNC:TINT? command, or this phase offset can be output over the serial ports via the SERVo:TRACe 1 pushed output sentence. In GPS 1PPS mode the 1PPS signal is phase-aligned to UTC from second to second, independent of the 10MHz oscillator output phase, and the 1PPS phase can thus jump nanoseconds in relationship to the 10MHz phase from second to second in order to maintain the best overall averaged phase offset to UTC.

Remote Serial Control

The unit is controlled via the SCPI TTL serial port at 115,200 baud or USB serial port at 921,600 baud, 8N1. Other Baud Rates can be set via SCPI commands.

Attach the STL-2600/STL-1400 unit to your PC's TeraTerm (recommended), or to the optional freeware GPSCon software package that can be downloaded from the support page on the JLT website (see also Chapter 5 of this manual).

Initial Operation

After connecting a suitable Iridium antenna and applying power to the unit through either the USB port or the optional +3.3V cable harness, the unit will search for STL satellite signals. The user can improve initial lock time by manually typing in the initial Guess Position as explained in "RECEiver:STL:GUESSPOSition:MANUAL <double> <aa.aaaaaa,bbb.bbbbbb,h.hh>" on page 51. Green LED D5 will blink whenever the unit has received and successfully processed an STL burst-packet. Expect this to happen anywhere from once every couple of seconds to several times a second for outdoor antenna locations with clear view of the sky. After the unit establishes an initial STL Position fix and the unit resolves for UTC time, the user will see the Red combo LED D23 blink once per second. This indicates that the unit has achieved a suitable Figure Of Merit (FOM) and is generating a UTC(STL) aligned 1PPS output. This can take anywhere from several minutes to tens of minutes after power-on depending on the quality of the initial guess-position and the STL satellite signals being received. The user might also see Green LED D21 blink at 1Hz - this indicates that the on-board GNSS receiver is generating a 3D Fix and alternate UTC(GPS) 1PPS output signal on pin 10 of J3. After typically ten to thirty minutes or more after the Red combo LED starts indicating a 1PPS output the unit will have disciplined the internal TCXO or the optional externally-connected oscillators to better than 1ppb frequency accuracy, and the Green combo LED D23 will light-up indicating the unit is fully locked to UTC, the oscillator has been disciplined, and there are no pending events that are being flagged.

Operating With An External 10mhz Reference Or Oscillator

The STL-2600/STL-1400 module can accept an external asynchronous 10MHz reference signal or an external synchronous 10MHz oscillator signal that is being disciplined by the module via TTL serial signals or the analog EFC output signal on connector J2. Oscillator disciplining can be disabled whenever the user can provide a highly-stable and accurate 10MHz reference source to pin 6 of J3. The external 10MHz input is enabled by pulling the CSAC_LOCK# pin 8 of J3 to ground. This can be done through hard-wiring, or by connecting pin 8 to the CSAC/Rubidium oscillator BITE/BIST output pin for automatic switchover when the external oscillator achieves atomic lock. Pin 8 should only be pulled to ground when the external 10MHz signal is sufficiently stable and within +/-5Hz offset of true 10MHz to avoid jumping the internal TCXO frequency excessively and causing a processor crash.

Also note that if the reference oscillator frequency is unstable or reference CSAC oscillator is not locked yet, any large frequency jump may confuse the STL Kalman Filter and result in longer time to achieve first 1PPS UTC(STL) fix. To avoid confusing the STL Kalman Filter, the J3 Pin 8 should not be pulled low until reference frequency is stable or delayed STL Kalman Filter processing can be configured with RECEiver:STL:KFPROCessing <ON|OFF|DELAYed> command.

Once the CSAC_LOCK# pin is pulled low the module disconnects the internal DAC (that usually steers the internal TCXO frequency) from the TCXO, and connects the TCXO steering input to an analog Phase Locked Loop (PLL) with less than 20Hz loop bandwidth. The external 10MHz reference signal now “steers” the internal TCXO by phase-locking the TCXO to the 10MHz reference. This is useful for reducing the CSAC phase noise for example, as the internal TCXO has a much lower spur and phase noise levels compared to a typical CSAC 10MHz output. The low-phase-noise filtered CSAC-derived 10MHz signal is available on pin 3 of connector J3 and typically has ~90 degrees phase shift to the 10MHz reference input signal.

External reference oscillators can either be digitally steered via the CSAC TX/RX pins 12 and 14 on J3, or via the analog DAC steering output on U.FI connector J2. The EFC output voltage on J2 can range from 0V to 3V and can directly drive the EFC inputs of positive-slope analog DOCXO’s that operate within the 0V to 3V EFC compliance range. It is recommended that a 220pF capacitor is connected between the EFC pin of the oscillator and Ground close to the OCXO EFC pin. SCPI commands are used to select analog or digital steering methods for the external oscillator, as well as the type of oscillator that is being connected (Microchip, Teledyne, Spectratime CSAC or Rubidium etc.). The externally connected oscillator is automatically phase- and frequency-steered by the proven JLT steering algorithms to UTC(STL) or UTC(GPS) or UTC(1PPS Ref-In) via SCPI user-selection as explained in Chapter 3.

Operating With Dual Processors

The STL-2600/STL-1400 with firmware 4.001 or later adds the support for utilizing the second MCU for burst processing to achieve higher count in received and processed bursts. The second MCU is enabled by default and can be configured in “SYSTem:SLAVECONfig <ON|OFF>” on page 88 .

With the second MCU enabled and Iridium antenna having clear view of sky, the STL-2600/STL-1400 can achieve up to 400 processed burst-per-minute count and up to 500 received burst-per-minute count.

The STL-2600/STL-1400 firmware also includes the automatic fail-safe switching to inactive the second MCU in case of hardware failure in second MCU or lost communication with the second MCU. When the second MCU is being inactivated, the master CPU will seamlessly take over and resume the burst processing operation, with no need for firmware reset or any other interruption.

NOTE

Enabling the second MCI will consume additional power than single MCU operation. To reduce power consumption, the SYSTem:PWRSAVing:SECONDMCU <ON|OFF> command may be configured to ON which will automatically disable the second MCU until returning to normal power mode.

Operating with Jackson Labs Transcoder/CLAW Simulator

The STL-2600/STL-1400 with firmware 4.012 or later adds the support for the external connected JLT Transcoder/CLAW Simulator option. An example diagram for connection between the JLT Micro Transcoder and STL-2600/STL-1400 is shown in Figure 2.5. The JLT STL-3200 module also has the same connection between the STL-2600/STL-1400 and Micro Transcoder as shown in Figure 2.5 and provides position and timing solution from STL-2600/STL-1400 to the target GNSS receiver via RF output port.

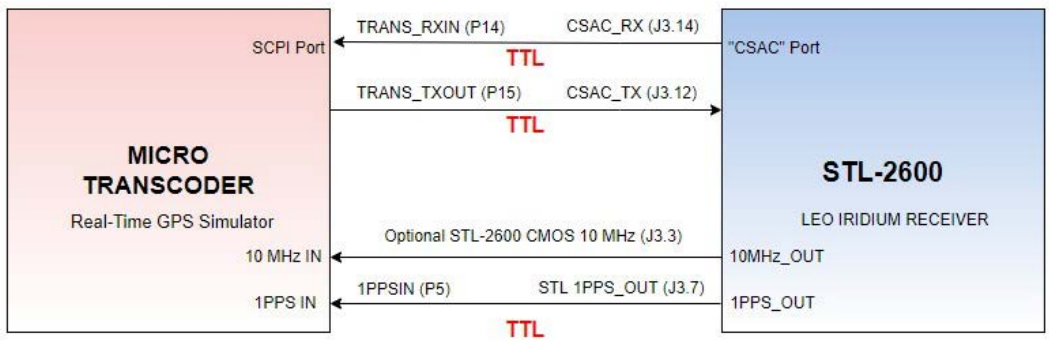
When TRANSCODER settings are configured in "SYSTem:COMMunicate:CSAC:MODE <OFF|QUERY|TRANScoder>" on page 83, the STL-2600/STL-1400 will configure the Transcoder/CLAW on start-up and provide STL NMEA data with position and time for transcoding. The 1PPS connection will allow transcoding with UTC aligned 1PPS timing. The 10MHz connection as shown is optional in which the user may provide more stable clock reference to STL-2600/STL-1400 or vice versa. See "SYSTem:COMMunicate:CSAC:MODE <OFF|QUERY|TRANScoder>" on page 83 and "Initial Operation" on page 25 for available commands related to Transcoder/CLAW operation with STL-2600/1400.

WARNING
Do not connect RS-232 serial levels to any serial port of connector J3. The unit will be damaged by RS-232 serial levels on connector J3, as the default configuration is TTL/LVCMOS levels only.

When connecting external module such as JLT RSR Transcoder or CLAW Simulator to the STL-2600/1400, ensure that all signals are (or converted to) TTL/LVCMOS levels such as the RS-232 level serial ports before connection is established.

Figure shows an example diagram for connection between the VIAVI Micro Transcoder and STL-2600.

Figure 8 STL-3200 Micro Transcoder and STL-2600/STL-1400 connection



Loop Parameter Adjustment

All loop parameters can be controlled via the USB or SCPI TTL serial port.

Loop parameters are optimized for the TCXO on the board, and changing the factory settings may result in the unit's performance to deteriorate. Loop time constants change automatically based on which 1PPS reference source is selected for disciplining (STL, GPS, or External). Since the STL signal is less stable than the typical GNSS signal, the time constants are extended when STL mode is selected.

Table 2.2 shows the recommended loop time constants for different 1PPS reference inputs in the STL-2600/STL-1400 board.

Table 3 Recommended loop settings for default on-board TCXO

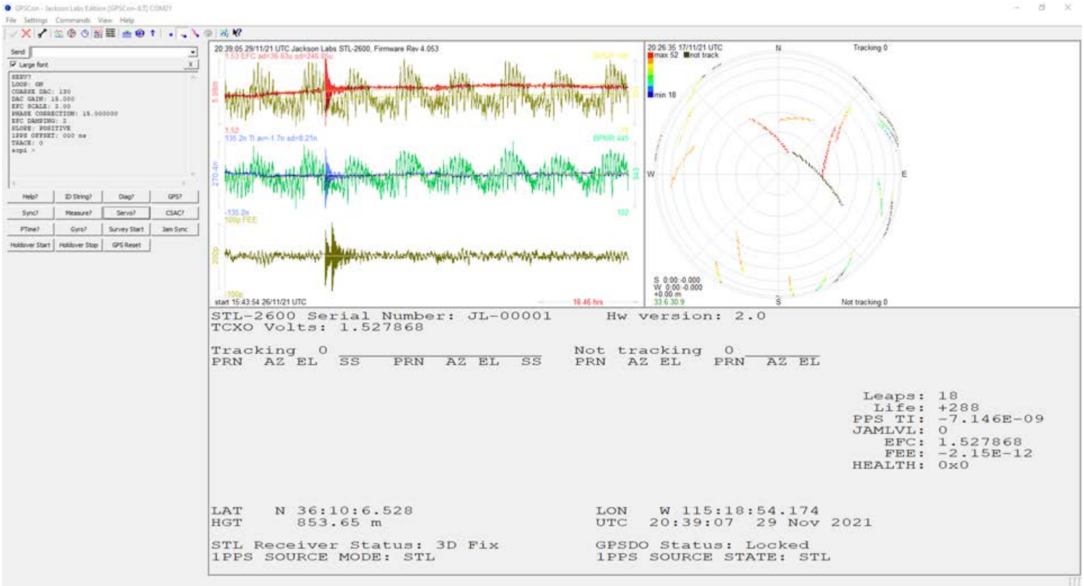
Usage	STL 1PPS	GNSS 1PPS	EXT 1PPS
SERV:EFCS	3.0	2.0	2.0
SERV:PHASECO	15.0	40.0	40.0
SERV:EFCD	2	2	2
SYNC:TINT:THR	1000	300	300

The commands to control the loop parameters are part of the SERVo? command. See "SERVO Subsystem" on page 90 for more information. The individual commands are:

- **EFC Scale:** this is the proportional gain of the PI loop. Higher values will give quicker convergence, and faster locking of the GPS time (lower loop time constant), lower values give less noise. Typical values are 0.2 to 3.0 for different 1PPS reference inputs in the STL-2600/STL-1400 board.
- **EFC Damping:** overall IIR filter time constant. Higher values increase loop time constant. Jackson Labs Technologies, Inc. typically uses values between 2 to 20. Setting this value too high may cause loop instability.
- **Phase compensation:** this is the Integral part of the PID loop. This corrects phase offsets between the STL-2600/STL-1400 1PPS signal and the UTC 1PPS signal as generated by the STL/GNSS receiver. Set higher values for tighter phase-following at the expense of frequency stability. Typical values range from 4.0 - 40.0. Setting this value too high may cause loop instability.

A well-compensated unit will show performance similar to the plot shown in Figure 2.6 when experiencing small perturbations.

Figure 9 STL-2600/STL-1400 phase compensation plot



3

SCPI-Control Quick Start

The following topics are discussed in this chapter:

Introduction

The SCPI (Standard Commands for Programmable Instrumentation) subsystem is accessed via the SCPI TTL or USB serial interfaces and a terminal program. By default the terminal settings are 921,600 baud on USB port (115,200 on SCPI TTL port), 8N1, no flow-control.

There are a number of commands that can be used as listed below. Most of these are identical or similar to HP/Symmetricom 58503A commands. To get a listing of the available commands, send the HELP? query. This will return a list of all the available commands for the STL-2600™ board.

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.

NOTE

The symbols “<“, “>”, “[“, “]”, “|”, and “,” in the parameter field listed in some commands of Chapter 3 are used for ranging or separating ONLY. Do not include these symbols when sending the SCPI commands to avoid a Command Error response.

NOTE

Only the upper-case letters have to be typed-in. Lowercase letters in the commands are optional.

General SCPI Commands

The following sections describe general SCPI commands.

Quick Start

For a quick start, try the following SCPI serial port commands:

- `HELP?`
- `SYSTem:STATus?`
- `RECEiver?`
- `SYNChronization?`
- `MEASure?`

*IDN?

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

`<company name> <model number>, <firmware revision>`

> **NOTE**
>
> The firmware revision reported by this command identifies the firmware revision number for the entire STL-2600 board and is different than the response of the `RECEiver:FWver?` command.

HELP?

This query returns a list of the commands available for the STL-2600™ board. The list of commands may be increased or may change with future firmware releases, please consult the HELP? command for the latest command set available on the installed firmware.

RECEiver Subsystem

> **NOTE**
>
> The STL-2600 can display antenna height in MSL Meters or in GPS Meters on all commands that return antenna height, with STL height being GPS height and the GNSS receiver outputting MSL height. The NMEA position fixes are in the WGS84 coordinate system, while the X,Y, and Z velocity vectors are given in the ECEF coordinate system.

The RECEiver subsystem regroups all the commands related to the control and status of the STL/GNSS receiver. The list of the commands supported is the following:

- `RECEiver?`
- `RECEiver:TYPE?`
- `RECEiver:GPGGA <int> [0,255]`
- `RECEiver:GPRMC <int> [0,255]`
- `RECEiver:GPZDA <int> [0,255]`
- `RECEiver:PJLTPOS <int> [0,255]`
- `RECEiver:PJLTPPS <int> [0,255]`
- `RECEiver:PJLTSTAT <int> [0,255]`
- `RECEiver:PJLTSUB <int> [0,255]`
- `RECEiver:PJLTVEL <int> [0,255]`
- `RECEiver:NMEA?`
- `RECEiver:NMEA:OUTput <ON|OFF|USBON|USBOFF|SCPION|SCPIOFF>`

- RECEiver:NMEA:OUTput:ZEROPOSition <ON|OFF>
- RECEiver:NMEA:SCPI:GPGGA <int> [0,255]
- RECEiver:NMEA:SCPI:GPRMC <int> [0,255]
- RECEiver:NMEA:SCPI:GPZDA <int> [0,255]
- RECEiver:NMEA:SCPI:PJLTPOS <int> [0,255]
- RECEiver:NMEA:SCPI:PJLTPPS <int> [0,255]
- RECEiver:NMEA:SCPI:PJLTSTAT <int> [0,255]
- RECEiver:NMEA:SCPI:PJLTSUB <int> [0,255]
- RECEiver:NMEA:SCPI:PJLTVEL <int> [0,255]
- RECEiver:POSition?
- RECEiver:POSition:ECEF?
- RECEiver:HEIGHT?
- RECEiver:INITial:DATE <yyyy,mm,dd>
- RECEiver:INITial:TIME <hour,min,sec>
- RECEiver:STL:AGC <int> [400,3333]
- RECEiver:STL:AGC:MODE <AUTO|MANUAL>
- RECEiver:STL:DGAIN <int> [0,15]
- RECEiver:STL:CLOCK
- <TCXO|OCXO|DOCXO|HS-DOCXO|CSAC|RUBIDIUM|[1.0E-20,1.0]>
- RECEiver:STL:GUESSPOSition <MANUAL|EEPROM|GNSS|BOOTSTRAP|TRUEPOS>
- RECEiver:STL:GUESSPOSition:MANUAL <double> <aa.aaaaaa,bbb.bbbbbb,h.hh>
- RECEiver:STL:GUESSPOSition:UNCertainty <float> <eee.e,nnn.n,uuu.u>
- RECEiver:STL:TRUEPOSition <double> <aa.aaaaaa,bbb.bbbbbb,h.hh>
- RECEiver:STL:GEOLOCation <STATIC>
- RECEiver:STL:AUTHSN?
- RECEiver:STL:EIN?
- RECEiver:STL:AUTHKEY <string>
- RECEiver:STL:AUTHKEY:REStore ONCE
- RECEiver:STL:SUBScription?
- RECEiver:STL:SUBScription:VALID?
- RECEiver:STL:SUBScription:STATus?
- RECEiver:STL:SUBScription:FEATure?
- RECEiver:STL:SUBScription:STARTdate?
- RECEiver:STL:SUBScription:ENDdate?
- RECEiver:STL:KFPROCessing <ON|OFF|DELAYed>
- RECEiver:STL:KFPROCessing:START ONCE
- RECEiver:STL:KFPROCessing:DELAY <int> [0,604800]

- RECEiver:STL:SENSitivity <HIGH|MEDium|LOW>
- RECEiver:STL:SENSitivity:DOPpler <HIGH|MEDium|LOW>?
- RECEiver:STL:DOPpler?
- RECEiver:STL:DOPpler:MINimum <double> [-36000.000000,0.0]
- RECEiver:STL:DOPpler:MAXimum <double> [0.0,36000.000000]
- RECEiver:STL:CNO?
- RECEiver:STL:CNO:MINTHReshold <float> [1.0,100.0]
- RECEiver:STL:CNO:MAXTHReshold <float> [1.0,100.0]
- RECEiver:STL:CNO:MAXimum?
- RECEiver:STL:PPSFilter?
- RECEiver:STL:PPSFilter:PFOM <int> [1,9]
- RECEiver:STL:PPSFilter:TFOM <int> [1,9]
- RECEiver:STL:PPSFilter:TFOM:UNCertainty <float> [0.1,2000.0]
- RECEiver:STL:PPSFilter:HOLDover <int> [0,604800]
- RECEiver:STL:PHASECOmpensation:OFFset <ns> [-1000000000.0,1000000000.0]
- RECEiver:STL:PHASECOmpensation:CALibrate ONCE
- RECEiver:STL:PHASECOmpensation:GNSSOFFSet?
- RECEiver:STL:STeer <double> <ppt> [-9000000.0,9000000.0]
- RECEiver:STL:BPMProcessed?
- RECEiver:STL:BPMReceived?
- RECEiver:STL:PFOM?
- RECEiver:STL:TFOM?
- RECEiver:GNSS:RESET <ONCE|HOLD|OFF>
- RECEiver:FWver?

RECEiver?

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

RECEiver:TYPE?

This command queries the currently selected receiver type. The response is one of the following:

- GNSS
- STL

The default receiver type is STL.

NMEA Support

The following six commands allow the STL-2600™ board to be used as an industry standard navigation GPS receiver. The GPGGA,GPRMC, PJLTPOS, PJLTPPS, PJLTSTAT, and PJLTVEL 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 via the STL-2600™ serial interfaces.

Both SCPI TTL and USB serial ports are capable of transmitting NMEA output messages. Once enabled (valid command received on the SCPI TTL or USB serial receive pin), STL-2600™ will send out information on the selected (SCPI TTL or USB) serial transmit pin automatically every N seconds. All incoming serial commands are still recognized by STL-2600™ since the serial interface transmit and receive lines are completely independent of one another.

Please note that the position, direction, and speed data is delayed by one second from when the STL/GNSS receiver internally reported these to the STL-2600™ 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 for the selected serial port, the following six commands will be stored in NV memory, and generate output information even after power to the unit has been cycled.

RECEiver:GPGGA <int> [0,255]

This command will configure the NMEA output message for the currently selected serial port receiving this command. This command instructs the STL-2600™ to send the NMEA standard string $GPGGA every N seconds, with N in the interval [0,255]. The output message configured by this command is invalid until the receiver achieves a first fix.

This command has the following format:

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

When operating from the GNSS receiver, the GPGGA sentence shows height in MSL Meters, this is different from traditional GPS receivers that display height in GPS Meters. When operating from the STL receiver height is indicated in GPS height. The difference between MSL and GPS height can be significant, 35m or more are common.

The following command will query the state of GPGGA output for the serial port receiving this command:

```
RECEiver:GPGGA?
```

RECEiver:GPRMC <int> [0,255]

This command will configure the NMEA output message for the currently selected serial port receiving this command. This command instructs the STL-2600™ to send the NMEA standard string $GPRMC every N seconds, with N in the interval [0,255]. The output message configured by this command is invalid until the receiver achieves a first fix.

This command has the following format:

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

The following command will query the state of GPRMC output for the serial port receiving this command:

```
RECEiver:GPRMC?
```

RECEiver:GPZDA <int> [0,255]

This command will configure the NMEA output message for the currently selected serial port receiving this command. This command instructs the STL-2600™ to send the NMEA standard string $GPZDA every N seconds, with N in the interval [0,255].

This command has the following format:

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

The following command will query the state of GPZDA output for the serial port receiving this command:

```
RECEiver:GPZDA?
```

RECEiver:PJLTPOS <int> [0,255]

This command will configure the NMEA output message for the currently selected serial port receiving this command. This command instructs the STL-2600™ to send the proprietary NMEA messages $PJLTPOS every N seconds, with N in the interval [0,255]. The output message configured by this command is invalid until the receiver achieves a first fix.

This command has the following format:

```
RECEiver:PJLTPOS <int> [0,255]
```

The following command will query the state of PJLTPOS output for the serial port receiving this command:

```
RECEiver:PJLTPOS?
```

The PJLTPOS string has the following data format:

```
$PJLTPOS,v,ssssssssss,f,ll.llllllll,yyy.yyyyyyyy,aaa.aa,xxx
xxxxxx,yyyyyyyyy,zzzzzzzzz,uu.uuuu,uu.uuuu,uu.uuuu,uu.uuuu,
uu.uuuu,uu.uuuu,b.b,s.s,p,g*[checksum]
```

where:

- `v` is the message version
- `ssssssssss` is the current time with "format of time" below
- `f` is the format of time:
  - `0` = GPS time and
  - `1` = Unix time
- `ll.llllllll` is the latitude in signed decimal degree
- `yyy.yyyyyyyy` is the longitude in signed decimal degree
- `aaa.aa` is the antenna altitude above/below GPS height in meters when receiving STL signals and MSL height when operating from the GNSS receiver
- `xxxxxxxxx` is the position in ECEF-X axis, in meter
- `yyyyyyyyy` is the position in ECEF-Y axis, in meter
- `zzzzzzzzz` is the position in ECEF-Z axis, in meter
- `uu.uuuu` are the uncertainty represented as local level covariance for EE, NN, UU, EN, EU, and NU, respectively
- `b.b` is the age of position in seconds since last burst
- `s.s` is the age of position in seconds since last sensor update
- `p` is the position figure of merit (PFOM) of current solution
- `g` is the current geo-location model configured in "RECEiver:STL:GEOLOCation <STATIC>" on page 53 with 1 = STATIC.

RECEiver:PJLTPPS <int> [0,255]

This command will configure the NMEA output message for the currently selected serial port receiving this command. This command instructs the receiver to send the proprietary NMEA messages $PJLTPPS every N seconds, with N in the interval [0,255]. The output message configured by this command is invalid until the receiver achieves a first fix.

This command has the following format:

```
RECEiver:PJLTPPS <int> [0,255]
```

The following command will query the state of PJLTPOS output for the serial port receiving this command:

```
RECEiver:PJLTPPS?
```

The PJLTPPS string has the following data format:

```
$PJLTPPS,v,ssssssssss,b,t,a.aaaaaa,r,ll,p,q*[checksum]
```

where

- `v` is the message version
- `ssssssssss` is the GPS time (number of seconds since GPS epoch in 1980)
- `b` is the number of seconds since last burst
- `t` is the number of seconds since last PPS timing update
- `a.aaaaaa` is the estimation of PPS accuracy in microseconds
- `r` is the PPS signal output ready status (1 = ready, 0 = not ready)
- `ll` is the leap second difference between GPS time and UTC time (in seconds)
- `p` is the pending leap second event (+# = positive leap second, 0 = no pending leap second -#= negative leap second)
- `q` is the current Time Figure of Merit (TFOM) value in the STL solution. This value is equivalent to the "RECEiver:STL:TFOM?" query.

RECEiver:PJLTSTAT <int> [0,255]

This command will configure the NMEA output message for the currently selected serial port receiving this command. This command instructs the receiver to send the proprietary NMEA messages $PJLTSTAT every N seconds, with N in the interval [0,255]. The GPS time in the $PJLTSTAT output message is invalid until the receiver achieves a first fix.

This command has the following format:

```
RECEiver:PJLTSTAT <int> [0,255]
```

The following command will query the state of PJLTSTAT output for the serial port receiving this command:

```
RECEiver:PJLTSTAT?
```

The PJLTSTAT string has the following data format:

```
$PJLTSTAT,v,ssssssssss,b,s*[checksum]
```

where

- `v` is the message version
- `ssssssssss` is the GPS time (number of seconds since GPS epoch in 1980)
- `b` is the number of received bursts in last 60 seconds (bursts per minute, or BPM)
- `s` is the number of received strong (processed) bursts in last 60 seconds

RECEiver:PJLTSUB <int> [0,255]

This command will configure the NMEA output message for the currently selected serial port receiving this command. This command instructs the receiver to send the proprietary NMEA messages $PJLTSUB every N seconds, with N in the interval [0,255]. The subscription status in the $PJLTSUB output message will stay unknown until the GPS time is received.

This command has the following format:

```
RECEiver:PJLTSUB <int> [0,255]
```

The following command will query the state of PJLTSUB output for the serial port receiving this command:

```
RECEiver:PJLTSUB?
```

The PJLTSUB string has the following data format:

```
$PJLTSUB,v,rrrrr,iiiiiiii,ssssssssss,V,S,ff,yyyy,mm,dd,YYYY
,MM,DD*[checksum]
```

where

- `v` is the message version
- `rrrrr` is the firmware revision reported in *IDN? response
- `iiiiiiii` is the serial number for the receiver described in "RECEiver:STL:AUTHSN?" on page 54
- `ssssssssss` is the GPS time (number of seconds since GPS epoch in 1980)
- `V` is the validity flag of subscription key
- `S` is the subscription status described in Table 4
- `ff` is the activated features bit-field described in Table 5
- `yyyy` is the starting year of subscription
- `mm` is the starting month of subscription
- `dd` is the starting day of subscription
- `YYYY` is the ending year of subscription
- `MM` is the ending month of subscription
- `DD` is the ending day of subscription

Table 4 describes the subscription status.

Table 4 Subscription Status

Status Value	Description
-1	Unknown: GPS time is not yet known and cannot compare range of subscription date
0	Inactive: receiver now knows GPS time and determines subscription date is outside of the range
1	Active: receiver now knows GPS time and determines subscription date is within the range

Table 5 describes the subscription feature codes.

Table 5 Subscription features

Feature Code	Description
ff |= 0x1	Location
ff |= 0x2	Timing
ff |= 0x4	Validated Location
ff |= 0x8	Validated Timing

RECEiver:PJLTVEL <int> [0,255]

This command will configure the NMEA output message for the currently selected serial port receiving this command. This command instructs the receiver to send the proprietary NMEA messages $PJLTVEL every N seconds, with N in the interval [0,255]. The output message configured by this command is invalid until the receiver achieves a first fix.

This command has the following format:

```
RECEiver:PJLTVEL <int> [0,255]
```

The following command will query the state of PJLTVEL output for the serial port receiving this command:

```
RECEiver:PJLTVEL?
```

The PJLTVEL string has the following data format:

```
$PJLTVEL,v,iiiiiiiiii,e.eeeeee,n.nnnnnn,u.uuuuuu,ssssssssss
*[checksum]
```

where:

- `v` is the message version,
- `iiiiiiiiii` is the serial number for the receiver described in "RECEiver:STL:AUTHSN?" on page 54
- `e.eeeeee` is the velocity in east direction, in m/s,
- `n.nnnnnn` is the velocity in north direction, in m/s,
- `u.uuuuuu` is the velocity in up direction, in m/s,
- `ssssssssss` is the GPS time (number of seconds since GPS epoch in 1980)

RECEiver:NMEA?

This query will display results from the following commands:

- `RECEiver:NMEA:OUTput?`
- `RECEiver:GPGGA?`

- `RECEiver:GPRMC?`
- `RECEiver:GPZDA?`
- `RECEiver:PJLTPOS?`
- `RECEiver:PJLTPPS?`
- `RECEiver:PJLTSTAT?`
- `RECEiver:PJLTSUB?`
- `RECEiver:PJLTVEL?`
- `RECEiver:NMEA:SCPI:GPGGA?`
- `RECEiver:NMEA:SCPI:GPRMC?`
- `RECEiver:NMEA:SCPI:GPZDA?`
- `RECEiver:NMEA:SCPI:PJLTPOS?`
- `RECEiver:NMEA:SCPI:PJLTPPS?`
- `RECEiver:NMEA:SCPI:PJLTSTAT?`
- `RECEiver:NMEA:SCPI:PJLTSUB?`
- `RECEiver:NMEA:SCPI:PJLTVEL?`

RECEiver:NMEA:OUTput <ON|OFF|USBON|USBOFF|SCPION|SCPIOFF>

This command will control the output of all available NMEA messages on one or both of the USB and TTL SCPI serial ports. The following NMEA output messages are supported: GPGGA, GPRMC, GPZDA, PJLTPOS, PJLTPPS, PJLTSTAT, and PJLTVEL. Sending the `RECE:NMEA:OUT USBON` command will turn on all NMEA outputs on the USB port. The USBOFF settings will turn off all NMEA outputs if any enabled on the USB port. The SCPION settings will turn on all NMEA outputs and the SCPIOFF settings will turn off all NMEA outputs if any enabled on the TTL SCPI port. The `RECE:NMEA:OUT ON` command will turn on all NMEA outputs on both USB and TTL SCPI ports. The RECE:NMEA:OUT OFF command will turn off all NMEA outputs if any enabled on both USB and TTL SCPI ports.

This command has the following format:

```
RECEiver:NMEA:OUTput <ON|OFF|USBON|USBOFF|SCPION|SCPIOFF>
```

The following command will query the state of this command and indicating if any NMEA output is enabled on USB or TTL SCPI ports:

```
RECEiver:NMEA:OUTput?
```

RECEiver:NMEA:OUTput:ZEROPOSition <ON|OFF>

This command will disable the output of $GPGGA and $GPRMC NMEA messages on both USB and TTL SCPI serial ports if the position of $GPGGA and/or $GPRMC messages become invalid due to loss of fix. The default settings is ON which will continue outputting $GPGGA and $GPRMC messages after losing fix.

When configuring OFF, the device will stop outputting $GPGGA and $GPRMC messages containing zero latitude and/or longitude values to avoid jump in output position when lose fix, which may be useful for position plotting applications utilizing $GPGGA/GPRMC messages.

This command has the following format:

```
RECEiver:NMEA:OUTput:ZEROPOSition <ON|OFF>
```

The following command will query the settings of this command:

```
RECEiver:NMEA:OUTput:ZEROPOSition?
```

RECEiver:NMEA:SCPI:GPGGA <int> [0,255]

This command will configure the NMEA output message for the TTL SCPI serial port. This command can be instantiated on either USB or TTL SCPI serial port. This command instructs the device to send the NMEA standard string $GPGGA every N seconds, with N in the interval [0,255]. This message will be sent through the TTL SCPI port. The output message configured by this command is invalid until the receiver achieves a first fix.

This command has the following format:

```
RECEiver:NMEA:SCPI:GPGGA <int> [0,255]
```

The following command will query the state of this command for the TTL SCPI serial port:

```
RECEiver:NMEA:SCPI:GPGGA?
```

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

RECEiver:NMEA:SCPI:GPRMC <int> [0,255]

This command will configure the NMEA output message for the TTL SCPI serial port. This command can be instantiated on either USB or TTL SCPI serial port. This command instructs the STL-2600™ to send the NMEA standard string $GPRMC every N seconds, with N in the interval [0,255]. This message will be sent through the TTL SCPI port. The output message configured by this command is invalid until the receiver achieves a first fix.

This command has the following format:

```
RECEiver:NMEA:SCPI:GPRMC <int> [0,255]
```

The following command will query the state of this command for the TTL SCPI serial port:

```
RECEiver:NMEA:SCPI:GPRMC?
```

RECEiver:NMEA:SCPI:GPZDA <int> [0,255]

This command will configure the NMEA output message for the TTL SCPI serial port. This command can be instantiated on either USB or TTL SCPI serial port. This command instructs the STL-2600/STL-1400 to send the NMEA standard string $GPZDA every N seconds, with N in the interval [0,255]. This message will be sent through the TTL SCPI port.

This command has the following format:

```
RECEiver:NMEA:SCPI:GPZDA <int> [0,255]
```

The following command will query the state of this command for the TTL SCPI serial port:

```
RECEiver:NMEA:SCPI:GPZDA?
```

RECEiver:NMEA:SCPI:PJLTPOS <int> [0,255]

This command will configure the NMEA output message for the TTL SCPI serial port. This command can be instantiated on either USB or TTL SCPI serial port. This command instructs the receiver to send the proprietary NMEA messages $PJLTPOS every N seconds, with N in the interval [0,255]. This message will be sent through the TTL SCPI port. The output message configured by this command is invalid until the receiver achieves a first fix.

This command has the following format:

```
RECEiver:NMEA:SCPI:PJLTPOS <int> [0,255]
```

The following command will query the state of this command for the TTL SCPI serial port:

```
RECEiver:NMEA:SCPI:PJLTPOS?
```

The PJLTPOS string has the following data format:

```
$PJLTPOS,v,ssssssssss,f,ll.llllllll,yyy.yyyyyyyy,aaa.aa,xxx
xxxxxx,yyyyyyyyy,zzzzzzzzz,uu.uuuu,uu.uuuu,uu.uuuu,uu.uuuu,
uu.uuuu,uu.uuuu,b.b,s.s,p,g*[checksum]
```

where:

- `v` is the message version,
- `ssssssssss` is the current time with "format of time" below,
- `f` is the format of time with 0 = GPS time and 1 = Unix time,
- `ll.llllllll` is the latitude in signed decimal degree,
- `yyy.yyyyyyyy` is the longitude in signed decimal degree,

- `aaa.aa` is the antenna altitude above/below GPS height in meters when receiving STL signals, and MSL height when operating from the GNSS receiver,
- `xxxxxxxxx` is the position in ECEF-X axis, in meter,
- `yyyyyyyyy` is the position in ECEF-Y axis, in meter,
- `zzzzzzzzz` is the position in ECEF-Z axis, in meter,
- `uu.uuuu` are the uncertainty represented as local level covariance for EE, NN, UU, EN, EU, and NU, respectively
- `b.b` is the age of position in seconds since last burst,
- `s.s` is the age of position in seconds since last sensor update,
- `p` is the position figure of merit (PFOM) of current solution,
- `g` is the current geo-location model configured in "RECEiver:STL:GEOLOCation <STATIC>" on page 53 with 1 = STATIC.

RECEiver:NMEA:SCPI:PJLTPPS <int> [0,255]

This command will configure the NMEA output message for the TTL SCPI serial port. This command can be instantiated on either USB or TTL SCPI serial port. This command instructs the receiver to send the proprietary NMEA messages $PJLTPPS every N seconds, with N in the interval [0,255]. This message will be sent through the TTL SCPI port. The output message configured by this command is invalid until the receiver achieves a first fix.

This command has the following format:

```
RECEiver:NMEA:SCPI:PJLTPPS <int> [0,255]
```

The following command will query the state of this command for the TTL SCPI serial port:

```
RECEiver:NMEA:SCPI:PJLTPPS?
```

The PJLTPPS string has the following data format:

```
$PJLTPPS,v,ssssssssss,b,t,a.aaaaaa,r,ll,p*[checksum]
```

where:

- `v` is the message version
- `ssssssssss` is the GPS time (number of seconds since GPS epoch in 1980)
- `b` is the number of seconds since last burst
- `t` is the number of seconds since last PPS timing update
- `a.aaaaaa` is the estimation of PPS accuracy in microseconds
- `r` is the PPS signal output ready status (1 = ready, 0 = not ready)
- `ll` is the leap second difference between GPS time and UTC time (in seconds)
- `p` is the pending leap second event (+# = positive leap second, 0 = no pending leap second -#= negative leap second)

RECEiver:NMEA:SCPI:PJLTSTAT <int> [0,255]

This command will configure the NMEA output message for the TTL SCPI serial port. This command can be instantiated on either USB or TTL SCPI serial port. This command instructs the receiver to send the proprietary NMEA messages $PJLTSTAT every N seconds, with N in the interval [0,255]. This message will be sent through the TTL SCPI port. The output message configured by this command is invalid until the receiver achieves a first fix.

This command has the following format:

```
RECEiver:PJLTSTAT <int> [0,255]
```

The following command will query the state of this command for the TTL SCPI serial port:

```
RECEiver:NMEA:SCPI:PJLTSTAT?
```

The PJLTSTAT string has the following data format:

```
$PJLTSTAT,v,ssssssssss,b,s*[checksum]
```

where

- `v` is the message version,
- `ssssssssss` is the GPS time (number of seconds since GPS epoch in 1980),
- `b` is the number of received bursts in last 60 seconds (bursts per minute, or BPM),
- `s` is the number of received strong (processed) bursts in last 60 seconds.

RECEiver:NMEA:SCPI:PJLTSUB <int> [0,255]

This command will configure the NMEA output message for the TTL SCPI serial port. This command can be instantiated on either USB or TTL SCPI serial port. This command instructs the receiver to send the proprietary NMEA messages $PJLTSUB every N seconds, with N in the interval [0,255]. This message will be sent through the TTL SCPI port. The subscription status in the $PJLTSUB output message will stay unknown until the GPS time is received.

This command has the following format:

```
RECEiver:NMEA:SCPI:PJLTSUB <int> [0,255]
```

The following command will query the state of this command for the TTL SCPI serial port:

```
RECEiver:NMEA:SCPI:PJLTSUB?
```

The PJLTSUB string has the following data format:

```
$PJLTSUB,v,rrrrr,iiiiiii,ssssssssss,V,S,ff,yyyy,mm,dd,YYYY
,MM,DD*[checksum]
```

where

- `v` is the message version,
- `rrrrr` is the firmware revision reported in *IDN? response,
- `iiiiiiii` is the serial number for the receiver described in "RECEiver:STL:AUTHSN?" on page 54
- `ssssssssss` is the GPS time (number of seconds since GPS epoch in 1980),
- `V` is the validity flag of subscription key,
- `S` is the subscription status described in Table 4,
- `ff` is the activated features bit-field described in Table 5,
- `yyyy` is the starting year of subscription,
- `mm` is the starting month of subscription,
- `dd` is the starting day of subscription,
- `YYYY` is the ending year of subscription,
- `MM` is the ending month of subscription,
- `DD` is the ending day of subscription.

RECEiver:NMEA:SCPI:PJLTVEL <int> [0,255]

This command will configure the NMEA output message for the TTL SCPI serial port. This command can be instantiated on either USB or TTL SCPI serial port. This command instructs the receiver to send the proprietary NMEA messages $PJLTVEL every N seconds, with N in the interval [0,255]. This message will be sent through the TTL SCPI port. The output message configured by this command is invalid until the receiver achieves a first fix.

This command has the following format:

```
RECEiver:NMEA:SCPI:PJLTVEL <int> [0,255]
```

The following command will query the state of this command for the TTL SCPI serial port:

```
RECEiver:NMEA:SCPI:PJLTVEL?
```

The PJLTVEL string has the following data format:

```
$PJLTVEL,v,iiiiiiiiii,e.eeeeee,n.nnnnnn,u.uuuuuu,ssssssssss
*[checksum]
```

where

- `v` is the message version,
- `iiiiiiiiii` is the serial number for the receiver described in "RECEiver:STL:AUTHSN?" on page 54
- **`e.ee`**`eeee` is the velocity in east direction, in m/s,
- `n.nnnnnn` is the velocity in north direction, in m/s,
- `u.uuuuuu` is the velocity in up direction, in m/s,
- `ssssssssss` is the GPS time (number of seconds since GPS epoch in 1980).

This query will return the current latitude and longitude in signed decimal degree, and altitude above/below GPS height in meters being reported from the STL or GNSS receiver. The GNSS receiver may also report the MSL height in meters.

RECEiver:POSition:ECEF?

This query will return the earth-centered, earth-fixed (ECEF) X, Y, and Z coordinates as well as the position accuracy (PFOM for STL receiver) of the STL or GNSS receiver.

RECEiver:HEIGHT?

This command returns the current altitude of the STL receiver in GPS Meters or current altitude of the GNSS receiver in both MSL and GPS Meters.

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

This command allows setting the internal RTC DATE manually when operating the unit in GNSS denied environments. This command is compatible to the PTIMe:OUTput ON command available in other JLT products to allow automatic time and date synchronization of two units to each other.

For compatibility purpose, the STL-2600/STL-1400 also supports the GPS:INITial:DATE and GPS:INITial:TIME input messages from other JLT products. The internal RTC is driven by the highly stable oscillator 10MHz signal, and thus has very high time accuracy.

This command has the following format:

```
RECEiver:INITial:DATE <yyyy,mm,dd>
```

The following command will query the state of this command:

```
RECEiver:INITial:DATE?
```

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

This command allows setting the internal RTC TIME manually when operating the unit in GNSS denied environments. This command is compatible to the PTIMe:OUTput ON command available in other JLT products to allow automatic time and date synchronization of two units to each other.

For compatibility purpose, the STL-2600/STL-1400 also supports the GPS:INITial:DATE and GPS:INITial:TIME input messages from other JLT products. The internal RTC is driven by the highly stable oscillator 10MHz signal, and thus has very high time accuracy.

This command has the following format:

```
RECEiver:INITial:TIME <hour,min,sec>
```

The following command will query the state of this command:

```
RECEiver:INITial:TIME?
```

RECEiver:STL:AGC <int> [400,3333]

This command will configure the manual RF Analog Gain Control (AGC) for the STL receiver.

Sending minimum value of 400 mV in this command results in maximum analog gain. The maximum value of 3333 mV results in minimum analog gain, with a nearly linear change of gain for intermediate values. Use this command to compensate for excessive cable losses or indoor reception with low signal levels, or non-amplified antennae. Observe the number of processed bursts per minute (BPM) in $PJLTSTAT message after changing the analog gain, and slowly adjust the value to achieve maximum number of processed BPM. A typical value when using the Tallysman 2643A amplified antenna with short antenna cabling is 1350 mV.

This command has the following format:

```
RECEiver:STL:AGC <int> [400,3333]
```

The following command will query the settings of this command:

```
RECEiver:STL:AGC?
```

RECEiver:STL:AGC:MODE <AUTO|MANUAL>

This command will determine the RF analog gain control mode for the STL receiver. Currently the AUTO mode is still work in progress and disabled until further notice.

The default and supported settings is MANUAL mode where the user can manually configure and determine the STL receiver’s analog gain control with RECEiver:STL:AGC <int> [400,3333] command.

This command has the following format:

```
RECEiver:STL:AGC:MODE <AUTO|MANUAL>
```

The following command will query the settings of this command:

```
RECEiver:STL:AGC:MODE?
```

RECEiver:STL:DGAIN <int> [0,15]

This command will configure the IF signal digital gain for the STL receiver. The default settings is 0 dB to avoid introducing extra noise into the STL receiver. JLT thus recommends configuring the RECEiver:STL:AGC <int> [400,3333] command before changing the settings of this command.

This command has the following format:

```
RECEiver:STL:DGAIN [0,15]
```

The following command will query the settings of this command:

```
RECEiver:STL:DGAIN?
```

RECEiver:STL:CLOCK <TCXO|OCXO|DOCXO|HS-DOCXO|CSAC|RUBIDIUM|[1.0E-20,1.0]>

This command allows the user to choose a clock model for the STL receiver. The default clock model is TCXO with fastest convergence rate at the expense of potential instability in the position/timing solution. The RUBIDIUM clock model minimize the instability with slower convergence rate. As listed in Table 6, smaller numeric value would instruct the STL receiver to expect higher stability in reference clock.

Note that the consequence to stability of configuring a somewhat optimistic, i.e., thinking that the reference clock can maintain higher stability than its actual specification, clock model is significantly worse than if configuring the clock model somewhat pessimistic.

This command supports both ASCII parameter for pre-defined clock model and numeric parameter for customized clock model. This command has the following formats:

```
RECEiver:STL:CLOCK <TCXO|OCXO|DOCXO|HS-DOCXO|CSAC|RUBIDIUM>
RECEiver:STL:CLOCK <float> [1.0E-20,1.0]
```

The following command will query the settings of this command:

```
RECEiver:STL:CLOCK?
```

Table 6 lists the pre-defined Clock Models.

Table 6 Pre-defined Clock Models

Pre-defined Clock Models	Numeric Values
TCXO	5.0E-7
OCXO	1.0E-9
DOCXO	1.0E-12
HS-OCXO	1.0E-14
CSAC	1.0E-15
RUBIDIUM	1.0E-17

RECEiver:STL:GUESSPOSition <MANUAL|EEPROM|GNSS|BOOTSTRAP|TRUEPOS>

This command will configure the source of initial guess position for STL solution. By providing an accurate initial guess position, the Time to First Fix is reduced. Note that an incorrect guess position may further increase the time to first fix.

The default guess position source is BOOTSTRAP mode where the STL receiver will compute the initial position without the information of guess position. The MANUAL or EEPROM settings require the user input of initial guess position with RECEiver:STL:GUESSPOSition:MANUAL command.

The TRUEPOS settings allows the STL solution to configure the user input position described in "RECEiver:STL:TRUEPOSition <double> <aa.aaaaaa,bbb.bbbbbb,h.hh>" on page 53 as the true antenna location. The TRUEPOS mode will stop computing new position in the solution attempting to improve timing performance in stationary application. The GNSS settings is currently not supported with the future plan of providing initial guess position from the on-board GNSS receiver.

The settings changed in this command will be effective only after the next power cycle or when the SYSTem:CPURESET command is sent.

This command has the following format:

```
RECEiver:STL:GUESSPOSition <MANUAL|EEPROM|GNSS|BOOTSTRAP>
```

The following command will query the settings of this command:

```
RECEiver:STL:GUESSPOSition?
```

RECEiver:STL:GUESSPOSition:MANUAL <double> <aa.aaaaaa,bbb.bbbbbb,h.hh>

This command allows the user to manually configure the initial guess position for STL solution.

Initial Time to First Fix and lock time is reduced the closer the guess position is to the true location of the antenna. Note that an incorrect guess position may further increase the time to first fix.

The settings changed in this command will be effective only if the RECE:STL:GUESSPOS MANUAL settings is also selected in "RECEiver:STL:GUESSPOSition <MANUAL|EEPROM|GNSS|BOOTSTRAP|TRUEPOS>" on page 51 and after the next power cycle or when the SYSTem:CPURESET command is sent.

This command has the following format:

```
RECEiver:STL:GUESSPOSition:MANUAL
<aa.aaaaaa,bbb.bbbbbb,h.hh>
```

Where:

- `aa.aaaaaa` is the latitude of the guess position in signed decimal format.
- `bbb.bbbbbb` is the longitude of the guess position in signed decimal format.
- `h.hh` is the altitude of the guess position above/below GPS height in meters.

The following command will query the settings of this command:

```
RECEiver:STL:GUESSPOSition:MANUAL?
```

RECEiver:STL:GUESSPOSition:UNCertainty <float> <eee.e,nnn.n,uuu.u>

This command allows the user to specify the standard deviation/uncertainty for the initial guess position configured in "RECEiver:STL:GUESSPOSition:MANUAL <double> <aa.aaaaaa,bbb.bbbbbb,h.hh>" on page 51. Narrowing down the uncertainty settings with knowledge of actual true location of the antenna will reduce the Initial Time to First Fix and lock time.

> **NOTE**
> An optimistic uncertainty setting may further increase the time to first fix.

The default settings for this command are 1000.0m, 1000.0m, 100.0m for position uncertainty in EE, NN, UU axis, respectively.

The settings changed in this command will be effective only if the RECE:STL:GUESSPOS MANUAL settings is also selected in "RECEiver:STL:GUESSPOSition <MANUAL|EEPROM|GNSS|BOOTSTRAP|TRUEPOS>" on page 51 and after the next power cycle or when the SYSTem:CPURESET command is sent.

This command has the following format:

```
RECEiver:STL:GUESSPOSition:UNCertainty <float>
<eee.e,nnn.n,uuu.u>
```

Where:

- `eee.e` is the uncertainty represented as standard deviation in EE axis of the guess position, in meters
- `nnn.n` is the uncertainty represented as standard deviation in NN axis of the guess position, in meters.
- `uuu.u` is the uncertainty represented as standard deviation in UU axis of the guess position, in meters

The following command will query the settings of this command:

```
RECEiver:STL:GUESSPOSition:UNCertainty?
```

RECEiver:STL:TRUEPOSition <double> <aa.aaaaaa,bbb.bbbbbb,h.hh>

This command allows the user to manually configure the hold position for STL solution. Initial Time to First Fix and lock time is reduced significantly by configuring the hold position as the true location of the antenna and stop computing new position in the solution.

> **NOTE**
> The hold position must be known as closer to the true antenna location as possible. Errors in hold position will translate into time errors to the STL-2600 1PPS output.
>
> The antenna location should be held completely stationary in TRUEPOS mode.

The settings changed in this command will be effective only if the RECE:STL:GUESSPOS TRUEPOS setting is also selected in "RECEiver:STL:GUESSPOSition <MANUAL|EEPROM|GNSS|BOOTSTRAP|TRUEPOS>" on page 51 and after the next power cycle or when the SYSTem:CPURESET command is sent.

This command has the following format:

```
RECEiver:STL:TRUEPOSition <aa.aaaaaa,bbb.bbbbbb,h.hh>
```

Where:

- `aa.aaaaaa` is the latitude of the true position in signed decimal degree
- `bbb.bbbbbb` is the longitude of the true position in signed decimal degree
- `h.hh` is the altitude of the true position above/below GPS height in meters

The following command will query the settings of this command:

```
RECEiver:STL:TRUEPOSition?
```

RECEiver:STL:GEOLOCation <STATIC>

This command allows the user to configure geo-location model being applied to the Kalman Filter of the STL receiver. The default and supported geo-location model is STATIC which assumes stationary Iridium antenna location and the STL solution will not compute updated position output when the antenna is moving.

Please contact your STL subscription/authentication key provider on potential support of other geo-location models of the STL receiver.

This command has the following format:

```
RECEiver:STL:GEOLOCation <STATIC>
```

The following command will query the settings of this command:

```
RECEiver:STL:GEOLOCation?
```

RECEiver:STL:EIN?

> **NOTE**
> This command is only available for Release 5.x and above.

This query will return the Equipment Identification Number (EIN). The EIN contains product ID, model number, and SDPM3 license authorization serial number. The EIN is programmed during manufacturing.

The format of this command is:

```
RECEiver:STL:EIN?
```

For example:

```
RECEiver:STL:EIN?


51190
STL2600
XLSL1234567890
```

RECEiver:STL:AUTHSN?

This query returns the license authorization serial number of the STL receiver. The Authorization serial number represents the license authorization key and is programmed during manufacturing. This serial number cannot be changed by a customer as it is tied to the hardware board and must be unique across all equipment connected to Iridium.

The Serial Number has different formats for Releases 4.x and below (SDPM1) and Releases 5.x and above (SDPM3).

Release 4.x

This query displays the serial number for the STL receiver. The STL serial number should be 8 characters with prefix “JL” unless otherwise specified.

If this command returns the response “INVALID SN!”, please contact VIAVI immediately with the serial number information shown on the label of STL-2600/STL-1400 board.

Release 5.x

The STL serial number is 14 characters that starts with 4 characters and has 10 more digits. If this command returns the response "INVALID SN!", please contact VIAVI immediately with the serial number information shown on the label of STL-2600/STL-1400 board.

Example:

```
RECEiver:STL:AUTHSN?
XLSL1234567890
```

RECEiver:STL:AUTHKEY <string>

This command allows the user to configure an authorization key for the STL receiver. The Authorization key has different formats for Releases 4.x and below (SDPM1) and Releases 5.x and above (SDPM3).

Release 4.x

> **NOTE**
>
> The string entered should be 28 characters long unless otherwise specified.

This command allows the user to configure an authentication key for the STL receiver. The string entered should be 28 characters long unless otherwise specified.

> **ALERT**
>
> This command will overwrite the existing key. System reset is required for new key activation.

> **NOTE**
>
> The over-the-air (OTA) subscription update may be available upon request without sending this command. Contact your STL subscription/authentication key provider for more information.

> **ALERT**
>
> If a new over-the-air key is received, the existing key will be overwritten.
>
> The Subscription state may have changed and affected receiver operation; this is expected behavior in response to an over-the-air subscription update.

Once new key is accepted, an encrypted copy of previous existing key will be stored to the EEPROM for backup. The RECEiver:STL:AUTHKEY:REStore ONCE command can be used to retrieve the previous key, if available.

This command has the following format:

```
RECEiver:STL:AUTHKEY <string>
```

The following command will query the state of the key:

```
RECEiver:STL:AUTHKEY?
```

Release 5.x

NOTE
Updates to the Authorization Keys take effect upon device reboot.

The Authorization Key represents the 24-character license authorization key provided by Iridium per authorization serial number. Manufacturing requests authorization keys per serial number and license type (Commercial, Locust, ANS) from Iridium. The Authorization key is programmed during manufacturing after Iridium has provided the key.

In the event of an expired time-limited license, a new license (obtained from Iridium) can be entered using the RECEiver:STL:AUTHKEY command.

The format of this command is:

```
RECEiver:STL:AUTHKEY <string>
```

For example:

```
RECE:STL:AUTHKEY 0123456789ABCDEF01234567
```

The following command will query the state of the key:

```
RECEiver:STL:AUTHKEY?
```

RECEiver:STL:AUTHKEY:REStore ONCE

This command allows the user to retrieve the stored previous existing key from the EEPROM, if available. When a new key is received and accepted, either from the RECEiver:STL:AUTHKEY

<string> command or the over-the-air subscription update, an encrypted copy of previous existing key will be stored to the EEPROM for backup.

ALERT
This command will overwrite the existing key. System reset is required for new key activation.

After sending this command and system reset, use the RECEiver:STL:SUBScription? commands to query the STL subscription status and confirm that the subscription is valid and active (not expired).

This command has the following format:

```
RECEiver:STL:AUTHKEY:REStore ONCE
```

RECEiver:STL:SUBScription?

This query will display the results of the following commands:

- `RECEiver:STL:SUBScription:VALID?`
- `RECEiver:STL:SUBScription:STATus?`
- `RECEiver:STL:SUBScription:FEATure?`
- `RECEiver:STL:SUBScription:STARTdate?`
- `RECEiver:STL:SUBScription:ENDdate?`

RECEiver:STL:SUBScription:VALID?

This query displays the validity of the STL subscription key indicating whether the key matches with the STL serial number listed in RECEiver:STL:AUTHSN? command.

The response of this command should display “VALID” for normal operation. If unmatched key is found, this command will display “INVALID” response and please contact your STL subscription/authentication key provider.

RECEiver:STL:SUBScription:STATus?

This query displays the current status of the STL subscription described in Table 4.

RECEiver:STL:SUBScription:FEATure?

This query will return the current STL subscription features that are enabled. Subscription features include: timing, location, or combination of both features. Also refer to the activated features bit-field described in Table 5 for $PJLTSUB message.

RECEiver:STL:SUBScription:STARTdate?

This query displays the start date of current STL subscription on the STL-2600/STL-1400 board.

RECEiver:STL:SUBScription:ENDdate?

This query displays the end/expiration date of current STL subscription on the STL-2600/STL-1400 board.

RECEiver:STL:KFPROCessing <ON|OFF|DELAYed>

This command allows the user to configure the initial burst processing mode for Kalman Filter of the STL solution. The default oscillator option is TCXO with fast/minimal warm-up time. However some oscillators such as OCXO, double-oven OCXO, CSAC, and Rubidium will require certain warm-up duration to stabilize the frequency. The Kalman

Filter may get confused by rapid change in frequency and result in large error in initial solution which requires even longer time to converge than expected.

The DELAYED settings in this command will temporarily prevent the Kalman Filter from processing bursts for the period configured in RECEiver:STL:KFPROCessing:DELAY <int> [0,604800] command. After the configured delay period, the STL-2600/STL-1400 will resume burst processing indicated by non-zero value in RECEiver:STL:BPMProcessed? query response for proper operation. The user may also send the RECEiver:STL:KFPROCessing:START ONCE command to manually terminate the delay period and resume burst processing for STL solution.

One example usage with CSAC oscillator reference is to pre-configure DELAYED settings in this command and delay period of 5-minute in "RECEiver:STL:KFPROCessing:DELAY <int> [0,604800]" on page 59. After power cycle, monitor the BITE pin status for atomic lock. Once atomic lock is detected, send the RECEiver:STL:KFPROCessing:START ONCE command to start burst processing for STL solution.

The default settings is ON assuming the on-board TCXO oscillator is used as reference. The OFF settings will stop the burst processing operation until the settings in this command is changed to ON or DELAYED.

This command has the following format:

```
RECEiver:STL:KFPROCessing <ON|OFF|DELAYed>
```

The following command will query the settings of this command:

```
RECEiver:STL:KFPROCessing?
```

RECEiver:STL:KFPROCessing:START ONCE

This command allows the user to manually resume the burst processing for STL solution, if DELAYED settings is configured in "RECEiver:STL:KFPROCessing <ON|OFF|DELAYed>" on page 57 and delay period configured in "RECEiver:STL:KFPROCessing <ON|OFF|DELAYed>" on page 57 is currently active.

This command has no effect if the RECEiver:STL:KFPROCessing <ON|OFF|DELAYed>command is configured to ON or OFF.

> **NOTE**
> This command is required if the delay period is configured to 0 in "RECEiver:STL:KFPROCessing:DELAY <int> [0,604800]" on page 59 and DELAYED settings is configured in "RECEiver:STL:KFPROCessing <ON|OFF|DELAYed>" on page 57.

This command has the following format:

```
RECEiver:STL:KFPROCessing:START ONCE
```

RECEiver:STL:KFPROCessing:DELAY <int> [0,604800]

This command allows the user to specify the period in seconds that the STL-2600/STL-1400 will postpone burst processing for Kalman Filter to avoid confusing Kalman Filter before the oscillator is stable. Refer to RECEiver:STL:KFPROCessing <ON|OFF|DELAYed> command for more details.

The default delay period is 180 seconds and the period can be configured up to one week of delay before resuming burst processing for STL solution. The special settings of 0 indicated by "START ONCE NEEDED" response and DELAYED settings configured in "RECEiver:STL:KFPROCessing <ON|OFF|DELAYed>" on page 57 will instruct the STL-2600/STL-1400 to postpone burst processing indefinitely until the RECEiver:STL:KFPROCessing:START ONCE command is received or the settings in "RECEiver:STL:KFPROCessing <ON|OFF|DELAYed>" on page 57 is changed to ON.

This command has the following format:

```
RECEiver:STL:KFPROCessing:DELAY <int> [0,604800]
```

The following command will query the settings of this command:

```
RECEiver:STL:KFPROCessing:DELAY?
```

RECEiver:STL:SENSitivity <HIGH|MEDium|LOW>

This command allows the user to configure three levels of recommended receiver sensitivity settings for the following commands with single command:

- `RECEiver:STL:AGC <int> [400,3333]`
- `RECEiver:STL:CNO:MINTHReshold <float> [1.0,100.0]`
- `RECEiver:STL:CNO:MAXTHReshold <float> [1.0,100.0]`
- `RECEiver:STL:DOPpler:MINimum <double> [-36000.000000,0.0]`
- `RECEiver:STL:DOPpler:MAXimum <double> [0.0,36000.000000]`

The default sensitivity level is HIGH with highest default analog gain settings and widest acceptable range for C/No and doppler listed in Table 7. The HIGH level is recommended for applications that require maximum burst-per-minute count such as indoor reception.

The MEDIUM and LOW levels, on the other hands, have lower analog gain settings and narrower range for C/No and doppler listed in Table 7 which may help rejecting some bursts with worse measuring quality and recommended for improving the timing solution, if the current processed burst-per-minute count can be reduced without frequent loss of fix.

Table 7 describes the recommended sensitivity settings.

Table 7 List of recommended sensitivity settings.

Sensitivity Levels	HIGH	MEDIUM	LOW
AGC (mV)	1350	1400	1450

Table 7 List of recommended sensitivity settings.

Sensitivity Levels	HIGH	MEDIUM	LOW
Minimum C/No Threshold (dB)	40.0	50.0	60.0
Maximum C/No Threshold (dB)	100.0	100.0	100.0
Minimum Doppler Threshold (Hz)	-36,000.000000	-35,500.000000	-35,500.000000
Maximum Doppler Threshold (Hz)	-36,000.000000	-35,500.000000	-35,500.000000

RECEiver:STL:SENSitivity:DOPpler <HIGH|MEDium|LOW>?

This command allows the user to configure three levels of recommended doppler sensitivity settings for both RECEiver:STL:DOPpler:MINimum and RECEiver:STL:DOPpler:MAXimum commands at the same time.

The default sensitivity level is HIGH with widest minimum and maximum doppler threshold range listed in Table 7. The HIGH level is recommended for applications that require maximum burst-per-minute count such as indoor reception.

The MEDIUM and LOW levels, on the other hands, have narrower minimum and maximum doppler threshold range listed in Table 7which may help rejecting some bursts with worse measuring quality and recommended for improving the timing solution, if the current processed burst-per-minute count can be reduced without frequent loss of fix.

RECEiver:STL:DOPpler?

This query will return the doppler value of last processed burst.

RECEiver:STL:DOPpler:MINimum <double> [-36000.000000,0.0]

This command allows the user to configure the minimum doppler value for any burst to be further processed in the STL solution. The value must be between -36,000.000000 Hz and 0.0 Hz.

All bursts used in STL solution will need to have doppler value inside the range of minimum threshold configured in this command and maximum threshold configured in RECEiver:STL:DOPpler:MAXimum <double> [0.0,36000.000000] command.

This command has the following format:

```
RECEiver:STL:DOPpler:MINimum <double> [-36000.000000,0.0]
```

The following command will query the settings of this command:

```
RECEiver:STL:DOPpler:MINimum?
```

RECEiver:STL:DOPpler:MAXimum <double> [0.0,36000.000000]

This command allows the user to configure the maximum doppler value for any burst to be further processed in the STL solution. The value must be between 0.0 Hz and 36,000.000000 Hz.

All bursts used in STL solution will need to have doppler value inside the range of minimum threshold configured in RECEiver:STL:DOPpler:MINimum <double> [-36000.000000,0.0] command and maximum threshold configured in this command.

This command has the following format:

```
RECEiver:STL:DOPpler:MAXimum <double> [0.0,36000.000000]
```

The following command will query the settings of this command:

```
RECEiver:STL:DOPpler:MAXimum?
```

RECEiver:STL:CNO?

This query will return the carrier to noise (C/No) value of the last decoded burst that will be further processed by the STL receiver.

RECEiver:STL:CNO:MINTHReshold <float> [1.0,100.0]

This command allows the user to configure the minimum carrier to noise (C/No) threshold for any burst to be further processed in the STL solution. The default minimum C/No threshold value is 40.0 dB.

All bursts used in STL solution will need to have C/No value inside the range of minimum threshold configured in this command and maximum threshold configured in RECEiver:STL:CNO:MAXTHReshold <float> [1.0,100.0] command.

This command has the following format:

```
RECEiver:STL:CNO:MINTHReshold <float> [1.0,100.0]
```

The following command will query the settings of this command:

```
RECEiver:STL:CNO:MINTHReshold?
```

RECEiver:STL:CNO:MAXTHReshold <float> [1.0,100.0]

This command allows the user to configure the maximum carrier to noise (C/No) threshold for any burst to be further processed in the STL solution. The default maximum C/No threshold value is 100.0 dB.

All bursts used in STL solution will need to have C/No value inside the range of minimum threshold configured in RECEiver:STL:CNO:MINTHReshold <float> [1.0,100.0] command and maximum threshold configured in this command.

This command has the following format:

```
RECEiver:STL:CNO:MAXTHReshold <float> [1.0,100.0]
```

The following command will query the settings of this command:

```
RECEiver:STL:CNO:MAXTHReshold?
```

RECEiver:STL:CNO:MAXimum?

This query will return the current maximum C/No value for previous decoded bursts since last STL fix is achieved.

RECEiver:STL:PPSFilter?

This query will display the results of the following commands:

- `RECEiver:STL:PPSFilter:PFOM?`
- `RECEiver:STL:PPSFilter:TFOM?`
- `RECEiver:STL:PPSFilter:HOLDover?`

RECEiver:STL:PPSFilter:PFOM <int> [1,9]

This command allows the user to configure the Position Figure of Merit (PFOM) threshold before the STL 1PPS output is enabled. The PFOM threshold is another way to set the Estimated Position Error (EPE) threshold. The default threshold value is 2 in which the default EPE threshold is < 50 meters before the STL 1PPS output is enabled after STL fix in static geo-location mode. PFOM values range from 1 to 9 and correspond to these industry-standard error estimates:

- 1: EPE < 25m
- 2: EPE < 50m
- 3: EPE < 75m
- 4: EPE < 100m
- 5: EPE < 200m
- 6: EPE < 500m
- 7: EPE < 1,000m
- 8: EPE < 5,000m
- 9: EPE >= 5,000m

This command has the following format:

```
RECEiver:STL:PPSFilter:PFOM [1,9]
```

The following command will query the settings of this command:

```
RECEiver:STL:PPSFilter:PFOM?
```

RECEiver:STL:PPSFilter:TFOM <int> [1,9]

This command allows the user to configure the Time Figure of Merit (TFOM) threshold before the STL 1PPS output is enabled. The TFOM threshold is another way to set the Estimated Time Error (ETE) threshold. The default threshold value is 4 in which the default ETE threshold is < 1 microsecond before the STL 1PPS output is enabled after STL fix in static geolocation mode.

TFOM values range from 1 to 9 and correspond to these industry-standard error estimates:

- 1: EPE < 1ns
- 2: EPE < 10ns
- 3: EPE < 100ns
- 4: EPE < 1us
- 5: EPE < 10us
- 6: EPE < 100us
- 7: EPE < 1ms
- 8: EPE < 10ms
- 9: EPE >= 10ms

This command has the following format:

```
RECEiver:STL:PPSFilter:TFOM [1,9]
```

The following command will query the settings of this command:

```
RECEiver:STL:PPSFilter:TFOM?
```

RECEiver:STL:PPSFilter:TFOM:UNCertainty <float> [0.1,2000.0]

This command allows the user to specify the standard deviation/uncertainty for all sources of short-term variability in the timing path such as 1PPS timer resolution, delay uncertainties, etc. This uncertainty does not include temperature, aging, or predicted holdover specs. This timing uncertainty and clock covariance reported in STL solution are then used to compute the TFOM for RECEiver:STL:TFOM? command.

Narrowing down the uncertainty settings with knowledge of all variables in the 1PPS timing path will provide more accurate estimate in TFOM computation. Note that an optimistic uncertainty settings may result in inaccurate TFOM reporting.

The recommended experimental uncertainty settings for the on-board TCXO is 100 ns and 20 ns with HS-DOCXO clock and disciplining to the STL 1PPS.

This command has the following format:

```
RECEiver:STL:PPSFilter:TFOM:UNCertainty <float> [0.1,2000.0]
```

The following command will query the settings of this command:

```
RECEiver:STL:PPSFilter:TFOM:UNCertainty?
```

RECEiver:STL:PPSFilter:HOLDover <int> [0,604800]

This command allows the user to specify the period in seconds that the holdover STL 1PPS output should remain active after losing the PPS timing update in the STL solution. The period or age information in seconds since last PPS timing update is available in the $PJLTPPS NMEA message described in "RECEiver:PJLTPPS <int> [0,255]" on page 38 and "RECEiver:NMEA:SCPI:PJLTPPS <int> [0,255]" on page 45.

The default holdover period is 30 seconds and the period can be configured up to one week of holdover STL 1PPS output from the internal or externally-connected holdover oscillator. The special settings of 0 indicated by "ON" response will instruct the STL-2600/STL-1400 to output STL 1PPS signal indefinitely after first PPS timing update.

This command has the following format:

```
RECEiver:STL:PPSFilter:HOLDover [0,604800]
```

The following command will query the settings of this command:

```
RECEiver:STL:PPSFilter:HOLDover?
```

RECEiver:STL:PHASECOmpensation:OFFset <ns> [-1000000000.0,1000000000.0]

This command configures the offset relative to overall constant phase offset in nanoseconds for the overall delay introduced in the analog filter and burst processing in the STL receiver. The STL-2600/STL-1400 board is typically calibrated by compensating the STL 1PPS output offset comparing to the on-board GNSS 1PPS output at factory, upon request. See the "RECEiver:STL:PHASECOmpensation:CALibrate ONCE" on page 65 command for phase offset calibration.

By sending this command, the STL-2600/STL-1400 board will add the relative offset to the current overall phase offset. The updated overall phase offset is then stored to the EEPROM. The maximum configurable phase compensation offset is +/- 1,000,000,000 ns or +/- one second.

This command has the following format:

```
RECEiver:STL:PHASECOmpensation:OFFset <ns> [-
1000000000.0,1000000000.0]
```

The following command will return the current overall phase offset:

```
RECEiver:STL:PHASECOmpensation:OFFset?
```

RECEiver:STL:PHASECOmpensation:CALibrate ONCE

This command allows the user to query and calibrate the STL 1PPS phase offset comparing to the GNSS 1PPS reference from the on-board GNSS receiver. For STL-2600/STL-1400 modules factory-configured with combined GNSS/Iridium antenna input at J4 connector discussed in "Connecting the Combined GNSS/Iridium Antenna" on page 24, both STL and GNSS receivers share the same antenna cable delay and thus the phase relationship is relative with the GNSS 1PPS aligned to the UTC time within 10s of ns in normal operation.

The STL-2600/STL-1400 will compute the running average of phase offset between the STL 1PPS and GNSS 1PPS reference and the phase offset can be queried with the following command:

```
RECEiver:STL:PHASECOmpensation:CALibrate?
```

By sending this command, the STL-2600/STL-1400 will update the phase offset value returned in RECEiver:STL:PHASECOmpensation:OFFset? command. This command will result in immediate phase change of the STL 1PPS to align with the GNSS 1PPS reference.

This command has the following format:

```
RECEiver:STL:PHASECOmpensation:CALibrate ONCE
```

The following command will return the current count of valid phase offset samples and running average of phase offset between the STL 1PPS and GNSS 1PPS reference:

RECEiver:STL:PHASECOmpensation:CALibrate?

RECEiver:STL:PHASECOmpensation:GNSSOFFSet?

This command allows the user to query the average STL 1PPS phase offset comparing to the GNSS 1PPS reference from the on-board GNSS receiver. For STL-2600/STL-1400 modules with combined GNSS/Iridium antenna input configuration at J4 connector discussed in "Connecting the Combined GNSS/Iridium Antenna" on page 24, both STL and GNSS receivers share the same antenna cable delay and thus the phase relationship is relative with the GNSS 1PPS aligned to the UTC time within 10s of ns in normal operation.

The STL-2600/STL-1400 will compute the running average of phase offset between the STL 1PPS and GNSS 1PPS reference and the phase offset can be queried with this command:

```
RECEiver:STL:PHASECOmpensation:GNSSOFFSet?
```

RECEiver:STL:STeer <double> <ppt> [-9000000.0,9000000.0]

This command allows the user to notify the STL Kalman filter whenever the external connected reference oscillator is being steered. This command aims to mitigate the frequency feedback loop avoiding the STL Kalman filter attempting to re-adjust the clock drift due to the external oscillator disciplining.

This command accepts the frequency steering value in units of parts-per-trillion (1E-12) related to the reference oscillator frequency (e.g., 10 MHz). The steering value range is +/- 9,000,000 ppt or +/- 90.0 Hz frequency adjustment for 10 MHz oscillator.

This command has the following format:

```
RECEiver:STL:STeer <double> <ppt> [-9000000.0,9000000.0]
```

The following command will query the last steering value received with this command:

```
RECEiver:STL:STeer?
```

RECEiver:STL:BPMProcessed?

This query will return the number of processed STL bursts in last 60 seconds. Typical values range from 0 to 400 depending on the quality of STL signal reception at the time of query.

RECEiver:STL:BPMReceived?

This query will return the number of received STL bursts in last 60 seconds. Typical values range from 0 to 400 depending on the quality of STL signal reception at the time of query.

RECEiver:STL:PFOM?

This query will return the current Position Figure of Merit (PFOM) value in the STL solution.

RECEiver:STL:TFOM?

This query will return the current Time Figure of Merit (TFOM) value in the STL solution.

RECEiver:STL:EIN?

> **NOTE**
> This command is only available for Release 5.x and above.

This query will return the Equipment Identification Number (EIN). The EIN contains product ID, model number, and SDPM3 license authorization serial number. The EIN is programmed during manufacturing.

The format of this command is:

```
RECEiver:STL:EIN?
```

For example:

```
RECEiver:STL:EIN?

51190
STL2600
XLSL1234567890
```

RECEiver:GNSS:RESET <ONCE|HOLD|OFF>

This command will configure the reset state of the on-board GNSS receiver if available.

The default settings is OFF which the on-board GNSS receiver is enabled in normal operation. The HOLD settings will keep and hold the on-board GNSS receiver on reset.

The ONCE settings will issue a reset sequence to the on-board GNSS receiver. This can be helpful when changing the antenna for example, since the GNSS receiver measures the antenna system's C/No right 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 green blinking LED D21 for GNSS Fix.

RECEiver:FWver?

This query will return the firmware version of the STL receiver or the GNSS receiver module.

> **NOTE**
> The response of this command is different than the firmware revision reported by the *IDN? command, which identifies the firmware revision number for the entire STL-2600 board.

PTIME Subsystem

The PTIME subsystem regroups all the commands related to the management of the internal RTC time.

> **NOTE**
> The STL-2600 does not contain a battery backup to maintain RTC time/date. As a result, the time/date is lost when power is removed or when the unit is reset.

The list of the commands supported is the following:

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

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 time. The hour, minute, and second are 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 adds support for auto-initialization of time and date between two Jackson Labs Technologies, Inc. GPSDO units. This allows connecting two units together through the currently selected serial port with a serial cable, and having the master unit send time and date information to the slave unit. The slave unit's 1PPS reference input can also be driven by the master unit's 1PPS output signal, by setting the slave unit to external 1PPS sync mode using the SYNC:SOUR:MODE EXT command. This allows time synchronization within seconds at the nanosecond level between two units which can be useful when operating in GNSS denied environments.

This command has the following format:

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

The following command returns the PTIME output settings for the serial port receiving this command:

```
PTIMe:OUTput?
```

Sending the following command will cause the unit to automatically generate GPS:INIT:DATE and GPS:INIT:TIME sentences described in "RECEiver:INITial:DATE <yyyy,mm,dd>" on page 48 and "RECEiver:INITial:TIME <hour,min,sec>" on page 48 on the selected serial port receiving this command once per second:

```
PTIMe:OUTput ON
```

PTIMe:OUTput:SCPI <ON|OFF>

This command will configure the PTIMe output messages for the TTL SCPI serial port. This command adds support for auto-initialization of time and date between two Jackson Labs Technologies, Inc. GPSDO units. This allows connecting two units together through the currently selected serial port with a serial cable, and having the master unit send time and date information to the slave unit. The slave unit's 1PPS reference input can also be driven by the master unit's 1PPS output signal, by setting the slave unit to external 1PPS sync mode using the SYNC:SOUR:MODE EXT command. This allows time synchronization within seconds at the nanosecond level between two units which can be useful when operating in GNSS denied environments.

This command has the following format:

```
PTIMe:OUTput:SCPI <ON|OFF>
```

The following command returns the PTIME output settings for the TTL SCPI serial port:

```
PTIMe:OUTput:SCPI?
```

Sending the following command will cause the unit to automatically generate GPS:INIT:DATE and GPS:INIT:TIME sentences described in "RECEiver:INITial:DATE <yyyy,mm,dd>" on page 48 and "RECEiver:INITial:TIME <hour,min,sec>" on page 48 on the TTL SCPI serial port once per second:

```
PTIMe:OUTput ON
```

PTIMe:LEAPsecond?

This command returns the leapseconds accrued between the GPS and UTC time systems.

PTIMe:LEAPsecond:PENDing?

This command returns 1 if the GPS almanac data contains a future pending leap second date or if the STL solution indicates an incoming leap second event and 0 if no future leap second is pending or almanac data or STL solution is not available. The internal GNSS receiver must have the GPS system enabled for the GPS almanac to be available when GNSS receiver is selected.

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 or no STL solution yet) or as indicated by the GNSS receiver (GPS almanac is available) or the STL receiver.

PTIME?

This query returns at once the result of the following queries:

- `PTIMe:DATE?`
- `PTIMe:TIME?`
- `PTIMe:TINTerval?`
- `PTIMe:OUTput?`
- `PTIMe:LEAPsecond:ACCumulated?`

SYNChronization Subsystem

This subsystem regroups the commands related to the synchronization of the STL-2600/STL-1400 with the STL/GNSS receiver. The list of the commands supported for this subsystem is the following:

- `SYNChronization?`
- `SYNChronization:SOURce:MODE [STL|GPS|EXTernal]`
- `SYNChronization:SOURce:STATE?`
- `SYNChronization:HOLDover:DURation?`
- `SYNChronization:HOLDover:STATe?`
- `SYNChronization:TINTerval?`
- `SYNChronization:TINTerval:THReshold <int> [50,2000]`
- `SYNChronization:IMMEdiate`
- `SYNChronization:FEEstimate?`
- `SYNChronization:LOCKed?`
- `SYNChronization:LOCKed:LEDMODE <LOCK|BURST-LOCK>`
- `SYNChronization:OUTput:1PPS:RESET [ON|OFF]`
- `SYNChronization:OUTput:1PPS:HOLDover <int> <s> [0,604800]`
- `SYNChronization:OUTput:1PPS:DOMAin [STL|SERVo]`
- `SYNChronization:HEAlth?`

SYNChronization:SOURce:MODE [STL|GPS|EXTernal]

This command selects between the EXTernal, GNSS, and STL 1PPS sync modes. The EXTernal and GPS modes use the external or GNSS 1PPS reference input to synchronize precise timing and to discipline the oscillator.

The STL mode uses the STL 1PPS output to synchronize precise timing and discipline the oscillator when the STL 1PPS output is reported as valid. If STL mode is selected and the STL 1PPS output is no longer valid, the STL-2600/STL-1400 board will go into holdover mode.

Changing the settings of this command will also automatically configure the servo loop parameters described in “Loop Parameter Adjustment” on page 27 and settings in “SYNChronization:TINTerval:THReshold <int> [50,2000]” on page 72 suitable for oscillator disciplining with STL/GNSS/EXTernal 1PPS source.

The format of this command is:

```
SYNChronization:SOURce:MODE <STL|GPS|EXTernal>
```

The following command will query the settings of this command:

```
SYNChronization:SOURce:MODE?
```

SYNChronization:SOURce:STATE?

This command is implemented, but always reports the same state as the SYNC:SOUR:MODE command setting in the STL-2600/STL-1400 board.

SYNChronization:HOLDover:DURation?

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

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 STL/GNSS receiver 1PPS or valid STL/GNSS fix.

SYNChronization:TINTerval?

This query returns the difference or timing shift between the oscillator and the UTC (STL/GNSS) 1PPS reference signals. The resolution is 1E-10 seconds.

This value is essentially the phase offset from the unit's 1PPS output to UTC and can be used to externally compensate for this phase offset. It is also pushed out in the SERVo:TRACe 1 command described in "SERVo:TRACe <int> [0,255]" on page 93.

> **NOTE**
> If the SERVo:LOOP? settings is OFF, the value returned by this query is unchanged and invalid since the oscillator disciplining is disabled with SERVo:LOOP OFF settings.

SYNChronization:TINTerval:THReshold <int> [50,2000]

This command selects the oscillator 1PPS phase-offset threshold as compared to the [EXT/STL/GNSS] reference 1PPS at which point the unit will initiate a counter-reset (jam-sync) aligning the TCXO 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 300ns for GNSS/EXT 1PPS and 1,000ns for STL 1PPS, allowing a drift of up to +/-1,000ns when in STL disciplining mode. 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 be "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 or the satellite signals are very weak.

The format of this command is:

```
SYNChronization:TINTerval:THReshold <int> [50,2000]
```

The following command will query the settings of this command:

```
SYNChronization:TINTerval:THReshold?
```

SYNChronization:IMMEdiate

This command initiates a near-instantaneous alignment (jam-sync) of the [EXT/GNSS/STL] 1PPS phase and Receiver output 1PPS phase. To be effective, this command has to be issued while not in holdover.

SYNChronization:FEEstimate?

This query returns the Frequency Error Estimate, similar to the Allan Variance using a 1,000s measurement interval and comparing the internal 1PPS to the [EXT/GNSS/STL] 1PPS offset.

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

> **NOTE**
> If the SERVo:LOOP? setting is OFF, the value returned by this query is unchanged and invalid since the oscillator disciplining is disabled with SERVo:LOOP settings.

SYNChronization:LOCKed?

This query returns the lock state (0=Not Locked, 1=Locked) of the PLL controlling the TCXO.

> **NOTE**
> If the SERVo:LOOP? setting is OFF, the value returned by this query is unchanged and invalid since the oscillator disciplining is disabled with SERVo:LOOP settings.

SYNChronization:LOCKed:LEDMODE <LOCK|BURST-LOCK>

This command selects the operation mode of Green LOCK LED D23 and Pin 11 of connector J3. The default settings is LOCK in which the LOCK LED will stay OFF until the unit is locked to the STL signals.

With BURST-LOCK settings, the Green LED D23 will behave the same as Green LED which will blink when STL packet has been received and has been successfully processed, until the unit is locked indicated by SYNChronization:LOCKed? query command. Once locked, the Green LED D23 will stay ON unless the unit becomes unlocked and start blinking for STL packet indication.

The format of this command is:

```
SYNChronization:LOCKed:LEDMODE <LOCK|BURST-LOCK>
```

This command will query the state of this command:

```
SYNChronization:LOCKed:LEDMODE?
```

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

This command allows the generation of the 1PPS output pulse upon power-on without an external L1 Band antenna being connected to the unit. By default the unit does not generate a 1PPS pulse until the receiver has locked onto the Satellites. With the command SYNC:OUT:1PPS:RESET ON the unit can now be configured to generate an asynchronous 1PPS output after power-on even if an L1 Band antenna is not connected to the unit. Once the receiver locks and generates a first 3D Fix, 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 first 3D Fix is achieved.

The format of this command is:

```
SYNChronization:OUTput:1PPS:RESET <ON|OFF>
```

This command will query the state of this command:

```
SYNChronization:OUTput:1PPS:RESET?
```

SYNChronization:OUTput:1PPS:HOLDover <int> <s> [0,604800]

This command allows the user to specify the period in seconds that the holdover 1PPS output (pin 7 of J3 connector) should remain active after losing the reference 1PPS source from the STL/GNSS receiver.

The default holdover period is 0 indicated by “ON” response where the special settings of 0 will instruct the STL-2600/STL-1400 to output 1PPS signal indefinitely after first fix. The holdover period can be configured up to one week of holdover 1PPS output from the internal or externally-connected holdover oscillator.

This command has the following format:

```
SYNChronization:OUTput:1PPS:HOLDover [0,604800]
```

The following command will query the settings of this command:

```
SYNChronization:OUTput:1PPS:HOLDover?
```

SYNChronization:OUTput:1PPS:DOMAin [STL|SERVo]

This command will configure the 1PPS output domain to be generated in either GPSDO or receiver (STL/GNSS) mode. In GPSDO mode the 1PPS output is generated by the 10MHz reference oscillator phase, and has extremely high stability (picoseconds jitter), whereas in receiver mode the phase of the 1PPS signal jumps nanoseconds every second just like a typical GNSS receiver so as to align the 1PPS as close as possible to the UTC(STL) phase. The advantage of the receiver mode is that on average the phase offset to UTC(STL) is minimized at the expense of massive second to second jitter.

In GPSDO mode the unit drives (disciplines/steers) the phase of the 10MHz oscillator very slowly to adjust the 1PPS output phase as closely as possible to UTC(STL) phase. This steering can take many hours, and the phase offset can be queried with the SYNC:TINT? command and can thus be externally compensated for. The advantage of the GPSDO mode is that the second to second 1PPS jitter is extremely small, on the order of picoseconds, while it can take significant time to align the 1PPS output phase to UTC(STL) due to the very slow oscillator steering. A secondary effect in GPSDO mode is that if the phase threshold is exceeded (see also the “SYNChronization:TINTerval:THReshold <int> [50,2000]” on page 72 command) then the phase will be “jam-synced” to the correct UTC(STL) phase, meaning there can be an instantaneous jump of 100’s of nanoseconds once the phase error threshold window is breached.

The STL 1PPS domain is used by default to minimize the overall 1PPS phase errors to UTC(STL) and thus the unit behaves like a typical GNSS receiver out of the box and when in STL mode.

The format for this command is:

```
SYNChronization:OUTput:1PPS:DOMAin [STL|SERVo]
```

The following command will query the settings of this command:

```
SYNChronization:OUTput:1PPS:DOMAin?
```

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.

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 system status information as described in Table 8.

Table 8 SYNChrinization:HEAlth? status information

If...	Status
The TCXO coarse-DAC is maxed-out at 255	HEALTH STATUS |= 0x1;
The TCXO coarse-DAC is mined-out at 0	HEALTH STATUS |= 0x2;
The phase offset to UTC is greater than settings in “SYNChronization:TINTerval:THReshold <int> [50,2000]” on page 72	HEALTH STATUS |= 0x4;
The run-time with 1PPS reference is < 300 seconds	HEALTH STATUS |= 0x8;
The STL-2600/STL-1400 is in holdover > 60s	HEALTH STATUS |= 0x10;
The TCXO coarse-DAC is maxed-out at 255	HEALTH STATUS |= 0x1;
The Frequency Error Estimate is out of bounds	HEALTH STATUS |= 0x20;
The power supply or TCXO voltage is too high	HEALTH STATUS |= 0x40;
The power supply or TCXO voltage is too low	HEALTH STATUS |= 0x80;
The short-term-drift (ADEV @ 100s) > 500ns	HEALTH STATUS |= 0x100;
For the first 3 minutes after a phase-reset, or a coarsedac change	HEALTH STATUS |= 0x200;

Table 8 SYNChrinization:HEAlth? status information

If...	Status
The on-board GNSS receiver indicates a strong jamming signal of >=50 (range is 0 to 255)	HEALTH STATUS |= 0x800;

As an example, if the unit is waiting for 1PPS reference, and the Frequency error estimate is too high, and the time since phase reset is < 3 minutes then the following errors would be indicated:

- 1PPS reference count < 300 seconds: 0x8
- Frequency error estimate > 5E-9 for STL 1PPS with TCXO: 0x20
- Time since phase normalization < 3 minutes: 0x200

'Oring' these values together results in:

0x200 | 0x20 | 0x8 = 0x228

The unit would thus indicate: HEALTH STATUS: 0x228

A health status of 0x0 indicates a properly locked, and warmed-up unit that is completely healthy.

NOTE

A non 0x0 health status and the corresponding lock_ok output going low does not necessarily mean the unit is unlocked.

NOTE

If the SERVo:LOOP? settings is OFF, the health status flags related to oscillator frequency stability are invalid since the oscillator disciplining is disabled with SERVo:LOOP OFF settings.

SYNChronization?

This query returns the results of following queries:

- `SYNChronization:SOURce:MODE?`
- `SYNChronization:SOURce:STATE?`
- `SYNChronization:OUTput:1PPS:DOMAin?`
- `SYNChronization:OUTput:1PPS:RESET?`
- `SYNChronization:OUTput:1PPS:HOLDover?`
- `SYNChronization:LOCKed?`
- `SYNChronization:HOLDover:STATe?`
- `SYNChronization:HOLDover:DURation?`
- `SYNChronization:FEEstimate?`
- `SYNChronization:TINTerval?`
- `SYNChronization:TINTerval:THReshold?`

- `SYNChronization:HEAlth?`

DIAGnostic Subsystem

This subsystem regroups the queries related to the diagnostic of the TCXO steering system and lifetime of the STL-2600/STL-1400 firmware.The list of the commands supported for this subsystem is the following:

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

DIAGnostic:ROSCillator:EFControl:RELative?

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

When an external Atomic oscillator is connected for disciplining, the steering value of the Atomic oscillator in units of parts-per-trillion (1E-12) is returned.

DIAGnostic:LIFetime:COUNt?

This command returns the number of hours the STL-2600/STL-1400 firmware has been running.

DIAGnostic:LIFetime:SECond?

This command returns the number of seconds the STL-2600/STL-1400 firmware has been running.

MEASURE Subsystem

This subsystem regroups the queries related of some parameters that are measured on-board on the STL-2600/1400. The list of the commands supported for this subsystem is the following:

- `MEASure:VOLTage?`
- `MEASure?`

MEASure:VOLTage?

This query is identical to the DIAGnostic:ROSCillator:EFControl:ABSolute? query.

MEASure?

This query returns the result of the following query:

```
MEASure:VOLTage?
```

SYSTEM Subsystem

This subsystem regroups the commands related to the general configuration of the STL-2600 board. The list of the commands supported for this subsystem is the following:

- `SYSTem:COMMunicate:SERial:ECHO <ON|OFF>`
- `SYSTem:COMMunicate:SERial:PROmpt <ON|OFF>`
- `SYSTem:COMMunicate:SERial:BAUD <9600|19200|38400|57600|115200|230400|460800|921600>`
- `SYSTem:COMMunicate:SERial:MSGMASK <ON|OFF>`
- `SYSTem:COMMunicate:SERial:OUTput <USBOFF|SCPIOFF|ALLOFF>`
- `SYSTem:COMMunicate:USB:ECHO <ON|OFF>`
- `SYSTem:COMMunicate:USB:PROmpt <ON|OFF>`
- `SYSTem:COMMunicate:USB:BAUD <9600|19200|38400|57600|115200|230400|460800|921600>`
- `SYSTem:COMMunicate:SCPI:PORT <ON|OFF>`
- `SYSTem:COMMunicate:SCPI:ECHO <ON|OFF>`
- `SYSTem:COMMunicate:SCPI:PROmpt <ON|OFF>`
- `SYSTem:COMMunicate:SCPI:BAUD <9600|19200|38400|57600|115200|230400|460800|921600>`
- `SYSTem:COMMunicate:CSAC:MODE <OFF|QUERY|TRANScoder>`
- `SYSTem:COMMunicate:CSAC:BAUD <9600|19200|38400|57600|115200|230400|460800|921600>`
- `SYSTem:SHOW?`
- `SYSTem:STATus?`
- `SYSTem:BUILD?`
- `SYSTem:OPERation:MODE <RECEiver|STL-3200>`

- `SYSTem:PWRSAVing:MODE [OFF|DUTY-CYCLE|RF-RECEIVER|SECOND-MCU|GNSS|ALL]`
- `SYSTem:PWRSAVing:DUTYCYCle <ON|OFF>`
- `SYSTem:PWRSAVing:DUTYCYCle:PERIOD [1,255]`
- `SYSTem:PWRSAVing:DUTYCYCle:RATIO [1,99]`
- `SYSTem:PWRSAVing:RFRECeiver <ON|OFF>`
- `SYSTem:PWRSAVing:SECONDMCU <ON|OFF>`
- `SYSTem:PWRSAVing:GNSS <ON|OFF>`
- `SYSTem:MASTERBRIDge`
- `SYSTem:MASTERBRIDge:TIMEOUT [5,300]`
- `SYSTem:SLAVECONfig <ON|OFF>`
- `SYSTem:SLAVESTATus?`
- `SYSTem:CPURESET`
- `SYSTem:ISP`
- `SYSTem:FACToryReset ONCE`
- `SYSTem:EEPROMUPDATE ONCE`

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

This command enables/disables echo on the currently selected serial port (USB/SCPI TTL) receiving this command. This command has the following format:

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

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

This command enables/disables the prompt “scpi >” on the SCPI command lines of the currently selected serial port (USB/SCPI TTL) receiving this command. The prompt must be enabled when used with the GPSCon Utility. This command has the following format:

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

SYSTem:COMMunicate:SERial:BAUD <9600|19200|38400|57600|115200|230400|460800|921600>

This command sets the serial speed for the currently selected serial port (USB/SCPI TTL) receiving this command. The serial configuration is always 8 bit, 1 stop bit, no parity, no HW flow control. The default speed is set at 921600 bauds for USB serial port and 115200 bauds for SCPI TTL serial port.

This command has the following format:

```
SYSTem:COMMunicate:SERial:BAUD <9600 | 19200 | 38400 | 57600
| 115200 | 230400 | 460800 | 921600>
```

The following command will configure serial speed of 115200 baud for the USB serial port if this command is received at the RX pin of USB serial port or serial speed for the SCPI TLL serial port if this command is received at the RX pin (pin 18 of J3 connector) of SCPI TTL serial port:

```
SYST:COMM:SER:BAUD 115200
```

SYSTem:COMMunicate:SERial:MSGMASK <ON|OFF>

Sending this command with ON settings will activate the message mask and temporarily disable output of all periodic and NMEA messages on both USB and SCPI serial ports.

> **NOTE**
> The setting changed in the command are NOT stored in EERPOM.

This command is useful for querying SCPI command settings and responses, without having the responses interrupted by other messages. Issuing SYST:COMM:SER:MSGMASK OFF command or power cycling the STL-2600/STL-1400 will both clear this mask and resume output of configured periodic and/or NMEA messages.

This command has the following format:

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

The following command will query the settings of this command:

```
SYSTem:COMMunicate:SERial:MSGMASK?
```

SYSTem:COMMunicate:SERial:OUTput <USBOFF|SCPIOFF|ALLOFF>

This command allows the user to turn off and disable output of all currently configured periodic and NMEA messages on USB port, SCPI port, or both serial ports. Note that the settings changed in this command is stored in EEPROM.

The USBOFF settings will check for all configured periodic and NMEA messages on USB serial port such as "RECEiver:GPGGA <int> [0,255]" and "RECEiver:PJLTSTAT <int> [0,255]" messages and disable all these messages. Similarly the SCPIOFF settings will disable all configured periodic and NMEA messages on SCPI serial port and the ALLOFF settings will disable all configured periodic and NMEA messages on both USB and SCPI serial ports.

This command has the following format:

```
SYSTem:COMMunicate:SERial:OUTput <USBOFF|SCPIOFF|ALLOFF>
```

SYSTem:COMMunicate:USB:ECHO <ON|OFF>

This command enables/disables echo on the USB serial port.

This command has the following format:

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

The following command will query the settings of this command:

```
SYSTem:COMMunicate:USB:ECHO?
```

SYSTem:COMMunicate:USB:PROmpt <ON|OFF>

This command enables/disables the prompt on the SCPI command lines of the USB serial port. The prompt must be enabled when used with the GPSCon Utility.

This command has the following format:

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

The following command will query the settings of this command:

```
SYSTem:COMMunicate:USB:PROmpt?
```

SYSTem:COMMunicate:USB:BAUD <9600|19200|38400|57600|115200|230400|460800|921600>

This command sets the serial speed for the USB serial port. The serial configuration is always 8 bit, 1 stop bit, no parity, no flow control. The default speed is set at 921600 bauds for USB serial port.

This command has the following format:

```
SYSTem:COMMunicate:USB:BAUD
<9600|19200|38400|57600|115200|230400|460800|921600>
```

The following command will query the settings of this command:

```
SYSTem:COMMunicate:USB:BAUD?
```

SYSTem:COMMunicate:SCPI:PORT <ON|OFF>

This command will enable or disable the SCPI TTL port (pins 16 and 18 of J3 connector) for serial communication. By default, the SCPI TTL port is enabled and should remain enabled for normal operation.

```
SYSTem:COMMunicate:SCPI:PORT <ON|OFF>
```

The following command will query the settings of this command:

```
SYSTem:COMMunicate:SCPI:PORT?
```

SYSTem:COMMunicate:SCPI:ECHO <ON|OFF>

This command enables/disables echo on the SCPI TTL serial port (pins 16 and 18 of J3 connector).

This command has the following format:

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

The following command will query the settings of this command:

```
SYSTem:COMMunicate:SCPI:ECHO?
```

SYSTem:COMMunicate:SCPI:PROmpt <ON|OFF>

This command enables/disables the prompt on the SCPI command lines of the SCPI TTL serial port (pins 16 and 18 of J3 connector). The prompt must be enabled when used with the GPSCon Utility.

This command has the following format:

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

The following command will query the settings of this command:

```
SYSTem:COMMunicate:SCPI:PROmpt?
```

SYSTem:COMMunicate:SCPI:BAUD <9600|19200|38400|57600|115200|230400|460800|921600>

This command sets the serial speed for the SCPI TTL serial port (pins 16 and 18 of J3 connector).

The serial configuration is always 8 bit, 1 stop bit, no parity, no flow control. The default speed is set at 115200 bauds for SCPI TTL serial port.

This command has the following format:

```
SYSTem:COMMunicate:SCPI:BAUD
<9600|19200|38400|57600|115200|230400|460800|921600>
```

The following command will query the settings of this command:

```
SYSTem:COMMunicate:SCPI:BAUD?
```

SYSTem:COMMunicate:CSAC:MODE <OFF|QUERY|TRANScoder>

This command selects the mode for the CSAC TTL serial port (pins 12 and 14 of J3 connector) to support different target devices or applications. Currently couple Jackson Labs products are supported including varies JLT CSAC-based and Rubidium-based GPSDO products and JLT Transcoder/CLAW Simulator products. By default, the CSAC mode is OFF where input data on CSAC port is ignored and STL-2600/STL-1400 will assume no device is connected to the CSAC TTL serial port. The SCPI command support on CSAC serial port is in progress and will allow the user to send and configure SCPI settings for STL-2600/1400.

The QUERY settings is designed for closed-loop application with STL-2600/STL-1400 providing STL 1PPS output to discipline external JLT CSAC/Rubidium GPSDO unit and CSAC/Rubidium 10MHz reference connected to 10MHz input (pin 6 of J3 connector) of STL-2600/1400. As described in “RECEiver:STL:STeer <double> <ppt> [-9000000.0,9000000.0]” on page 65, providing frequency steering info from external reference to the STL Kalman filter will mitigate the frequency feedback loop.

With QUERY settings and proper connection between STL-2600/STL-1400 and JLT CSAC/Rubidium GPSDO unit, the STL-2600/STL-1400 will periodically send the CSAC:STeer? query command and compute steering value for STL Kalman filter.

The TRANSCODER settings is designed for application with STL-2600/STL-1400 providing position and/or 1PPS output to JLT Transcoder/CLAW unit for transcoding. With TRANSCODER settings and proper connection between STL-2600/STL-1400 and JLT Transcoder/CLAW unit, the RF output from JLT Transcoder/CLAW unit can then be connected to target GPS/GNSS receiver with position and timing solution from STL-2600/1400. See “Operating with Jackson Labs Transcoder/CLAW Simulator” on page 27 for detailed descriptions of JLT Transcoder/CLAW operation with STL-2600/1400.

This command has the following format:

```
SYSTem:COMMunicate:CSAC:MODE <OFF|QUERY|TRANScoder>
```

The following command will query the settings of this command:

```
SYSTem:COMMunicate:CSAC:MODE?
```

SYSTem:COMMunicate:CSAC:BAUD <9600|19200|38400|57600|115200|230400|460800|921600>

This command sets the serial speed for the CSAC TTL serial port (pins 12 and 14 of J3 connector).

The serial configuration is always 8 bit, 1 stop bit, no parity, no flow control. The default speed is set at 115200 bauds for CSAC TTL serial port.

This command has the following format:

```
SYSTem:COMMunicate:CSAC:BAUD
<9600|19200|38400|57600|115200|230400|460800|921600>
```

The following command will query the settings of this command:

```
SYSTem:COMMunicate:CSAC:BAUD?
```

SYSTem:SHOW?

This query will return current configuration of all available SCPI commands in ASCII format and separated by line feeds. The response of this query may be copied over to another STL-2600/STL-1400 unit to load the same configuration, with minor modifications.

This command is useful for debugging.

SYSTem:STATus?

This query returns a full page of STL/GNSS receiver status in ASCII format. The output is compatible with the GPSCon Utility.

This command is useful for debugging.

SYSTem:BUILD?

This query will return the information on STL-2600/STL-1400 firmware revision, build date and time.

SYSTem:OPERation:MODE <RECEiver|STL-3200>

This command allows the user to configure the operation mode of STL-2600/STL-1400 with currently supported RECEIVER and STL-3200 modes. The default operation mode is RECEIVER which the STL-2600/STL-1400 is configured as receiver with position and timing output solution.

The STL-3200 settings will configure the BURST-LOCK settings in "SYNChronization:LOCKed:LEDMODE <LOCK|BURST-LOCK>" on page 73 and TRANSCODER settings in "SYSTem:COMMunicate:CSAC:MODE <OFF|QUERY|TRANScoder>" on page 83 for proper operation of STL-2600/STL-1400 and JLT Micro-Transcoder installed inside the JLT STL-3200 module. See "Operating with Jackson Labs Transcoder/CLAW Simulator" on page 27 for detailed descriptions of JLT Transcoder/CLAW operation with STL-2600/1400.

This command has the following format:

```
SYSTem:OPERation:MODE <RECEiver|STL-3200>
```

The following command will query the settings of this command:

```
SYSTem:OPERation:MODE?
```

SYSTem:PWRSAVing:MODE [OFF|DUTY-CYCLE|RF-RECEIVER|SECOND-MCU|GNSS|ALL]

This command allows the user to configure the power saving mode for the system which is recommended for battery-operated or other lower power applications. By enabling ALL power saving mode, power reduction of up to 550 mW may be achieved on STL-2600/STL-1400 with expense of reduced peak received and processed burst count down to ~100 bursts per minute.

The default settings is OFF for normal power mode. The ALL settings will enable (and OFF settings will disable) all following power saving options at the same time. Alternatively each individual option can be enabled/disabled separately.

- DUTY-CYCLE: The system periodically enters partial sleeping mode for the ON period of Duty Cycle and returns back to full power consumption for the OFF period of Duty Cycle. See “SYSTem:PWRSAVing:DUTYCYCle:PERIOD [1,255]” on page 86 for period and “SYSTem:PWRSAVing:DUTYCYCle:RATIO [1,99]” on page 86 for ratio of duty cycle.
- RF-RECEIVER: The system turns off the analog front end RF receiver at any moment when the system is busy processing the bursts.
- SECOND-MCU: The system disables second MCU on the STL-2600/STL-1400 by keeping the second MCU in reset state.
- GNSS: The system disables the on-board GNSS receiver by keeping the GNSS receiver in reset state.

This command has the following format:

```
SYSTem:PWRSAVing:MODE <OFF|DUTY-CYCLE|RF-RECEIVER|SECOND-
MCU|GNSS|ALL>
```

The following example command will enable power saving mode for Duty Cycle, RF Receiver, and Second MCU:

```
SYSTem:PWRSAVing:MODE DUTY-CYCLE RF-RECEIVER SECOND-MCU
```

The following command will query the settings of this command:

```
SYSTem:PWRSAVing:MODE?
```

SYSTem:PWRSAVing:DUTYCYCle <ON|OFF>

This command configures the Duty Cycle power saving mode for the system. With ON settings, the system will periodically enter partial sleeping mode for the ON period of Duty Cycle and return back to normal operation mode for the OFF period of Duty Cycle. The default settings is OFF for normal power mode. Refer to settings configured in “SYSTem:PWRSAVing:DUTYCYCle:PERIOD [1,255]” on page 86 for period and “SYSTem:PWRSAVing:DUTYCYCle:RATIO [1,99]” on page 86 for ratio of duty cycle.

For example, if period configured in “SYSTem:PWRSAVing:DUTYCYCle:PERIOD [1,255]” on page 86 is 10 seconds and ratio configured in “SYSTem:PWRSAVing:DUTYCYCle:RATIO [1,99]” on page 86 is 60 for 60%, then system will enter partial sleeping mode for 6 seconds and return back to normal operation mode for 4 seconds.

This command has the following format:

```
SYSTem:PWRSAVing:DUTYCYCle <ON|OFF>
```

The following command will query the settings of this command:

```
SYSTem:PWRSAVing:DUTYCYCle?
```

SYSTem:PWRSAVing:DUTYCYCle:PERIOD [1,255]

This command configures the duration in seconds for Duty Cycle power saving mode. The parameter is used only when Duty Cycle power saving mode is configured ON in “SYSTem:PWRSAVing:DUTYCYCle <ON|OFF>” on page 85. The power saving period can be configured between 1 seconds and 255 seconds with default settings of 10 seconds. See “SYSTem:PWRSAVing:DUTYCYCle <ON|OFF>” on page 85 for descriptions of Duty Cycle power saving mode and settings configured in “SYSTem:PWRSAVing:DUTYCYCle:RATIO [1,99]” on page 86 for ratio of duty cycle.

This command has the following format:

```
SYSTem:PWRSAVing:DUTYCYCle:PERIOD [1,255]
```

The following command will query the settings of this command:

```
SYSTem:PWRSAVing:DUTYCYCle:PERIOD?
```

SYSTem:PWRSAVing:DUTYCYCle:RATIO [1,99]

This command configures the ON / OFF ratio in percent of period for Duty Cycle power saving mode. The parameter is used only when Duty Cycle power saving mode is configured ON in “SYSTem:PWRSAVing:DUTYCYCle <ON|OFF>” on page 85. The system is in periodic power saving mode during ON time of Duty Cycle.

The power saving duty cycle can be configured between 1% and 99%. The default value for this system parameter is 50%. See “SYSTem:PWRSAVing:DUTYCYCle <ON|OFF>” on page 85 for descriptions of Duty Cycle power saving mode and settings configured in “SYSTem:PWRSAVing:DUTYCYCle:PERIOD [1,255]” on page 86 for period of duty cycle.

This command has the following format:

```
SYSTem:PWRSAVing:DUTYCYCle:RATIO [1,99]
```

The following command will query the settings of this command:

```
SYSTem:PWRSAVing:DUTYCYCle:RATIO?
```

SYSTem:PWRSAVing:RFRECeiver <ON|OFF>

This command configures the RF Receiver power saving mode for the system. With ON settings, the system will turn off the analog front end RF receiver at any moment when the system is busy processing the bursts. The default settings is OFF for normal power mode.

This command has the following format:

```
SYSTem:PWRSAVing:RFRECeiver <ON|OFF>
```

The following command will query the settings of this command:

```
SYSTem:PWRSAVing:RFRECeiver?
```

SYSTem:PWRSAVing:SECONDMCU <ON|OFF>

This command configures the Second MCU power saving mode for the system. With ON settings, the system will disable second MCU on the STL-2600/STL-1400 by keeping the second MCU in reset state.

The default settings is OFF for normal power mode.

This command has the following format:

```
SYSTem:PWRSAVing:SECONDMCU <ON|OFF>
```

The following command will query the settings of this command:

```
SYSTem:PWRSAVing:SECONDMCU?
```

SYSTem:PWRSAVing:GNSS <ON|OFF>

This command configures the GNSS Receiver power saving mode for the system. With ON settings, the system will disable the on-board GNSS receiver by keeping the GNSS receiver in reset state. The default settings is OFF for normal power mode.

This command has the following format:

```
SYSTem:PWRSAVing:GNSS <ON|OFF>
```

The following command will query the settings of this command:

```
SYSTem:PWRSAVing:GNSS?
```

SYSTem:MASTERBRIDge

This command will activate the pass-thru serial bridge between Master and Second MCUs of STL-2600/STL-1400 and configure In System Programming (ISP) mode for programing the Second MCU.

> **WARNING**
> This command will stop normal receiver operation and configured current serial port receiving this command for Second MCU programming mode. See Chapter 5 for firmware upgrade instructions. After time-out of inactivity configured in "SYSTem:MASTERBRIDge:TIMEOUT [5,300]" on page 88, the STL-2600 will reset the firmware automatically and resume normal operation mode.

This command has the following format:

```
SYSTem:PWRSAVing:MASTERBRIDge
```

SYSTem:MASTERBRIDge:TIMEOUT [5,300]

This command configures the time-out in seconds for active pass-thru serial bridge between Master and Second MPUs of STL-2600/1400. The default settings is 10 seconds.

If no activity is detected for time-out period configured in this command, then STL-2600/STL-1400 will reset the firmware automatically for normal operation mode. To apply changes, this command should be configured before activating the Master Bridge with SYSTem:MASTERBRIDge command.

This command has the following format:

```
SYSTem:PWRSAVing:MASTERBRIDge:TIMEOUT [5:300]
```

The following command will query the settings of this command:

```
SYSTem:PWRSAVing:MASTERBRIDge:TIMEOUT?
```

SYSTem:SLAVECONfig <ON|OFF>

This command allows the user to enable/disable Second (Slave) MCU of STL-2600/1400. The default settings is ON in which the Second MCU is enabled for burst processing. The Second MCU can be disabled anytime with OFF settings (or enabled with ON settings) and will take effect immediately and the settings is stored in EEPROM.

When the Second MCU is enabled and active, the STL-2600/STL-1400 will automatically off load the burst processing task from Master MCU to Slave MCU to expand burst processing capability and maximize burst count. When the Second MCU is disabled or inactive, the Master MCU is responsible for all operations including burst detection and processing. The status of Second MCU can be queried with SYSTem:SLAVESTATus? command.

> **NOTE**
>
> The Second MCU will consume additional power when active. The Second MCU may be temporarily disabled by SECONDMCU power saving settings configured in "SYSTem:PWRSAVing:MODE [OFF|DUTY-CYCLE|RF-RECEIVER|SECOND-MCU|GNSS|ALL]" on page 85 or "SYSTem:PWRSAVing:SECONDMCU <ON|OFF>" on page 87

This command has the following format:

```
SYSTem:SLAVECONFig <ON|OFF >
```

The following command will query the settings of this command:

```
SYSTem:SLAVECONFig?
```

SYSTem:SLAVESTATus?

This query will return the status of Second (Slave) MCU of STL-2600/1400.

Status of 1 indicates that the Second MCU is enabled and active for burst processing. Several factors may cause returned status of 0 including Second MCU being disabled in "SYSTem:SLAVECONfig <ON|OFF>" on page 88, Second MCU temporarily being disabled for power saving in "SYSTem:PWRSAVing:MODE [OFF|DUTY-CYCLE|RF-RECEIVER|SECOND-MCU|GNSS|ALL]" on page 85 or "SYSTem:PWRSAVing:SECONDMCU <ON|OFF>" on page 87, or communication/ hardware failure on the Second MCU.

SYSTem:CPURESET

This command causes the STL-2600/STL-1400 processor to reset.

SYSTem:ISP

This command causes the STL-2600/STL-1400 processor to reset into In System Programming (ISP) mode for firmware upgrades. Please see Chapter 5 for details on updating the firmware in ISP mode.

Initiating this command is equivalent to shorting the ISP# pin 13 of J3 connector to +3.3V. +3.3V can be found at the end of inductor D3 (between D3 and D4). Then, short pin 15 (RST_IN#) of J3 connector to Ground of the STL-2600/STL-1400 board during power-on.

Once the system is placed into ISP mode it will need to be power-cycled or an ISP-Reset command will need to be sent through the ISP programming interface commands to establish a normal operating mode.

SYSTem:FACToryReset ONCE

This command applies the Factory Reset setting to the EEPROM. WARNING: This operation resets all user parameters and clears factory calibration parameters. The STL-2600/STL-1400 will also initiate an AUTOMATIC FIRMWARE RESTART after the factory reset for proper re-initialization of the system.

Sending this command once results in a warning message being output. Sending the SYSTem:FACToryReset ONCE command a second time performs the factory reset.

> **ALERT**
> This command should not be used during normal field operation.

SYSTem:EEPROMUPDATE ONCE

This command applies the latest Factory default parameters embedded in the STL-2600/STL-1400 firmware to the EEPROM. WARNING: This operation overwrites all user parameters and clears factory calibration parameters. The STL-2600/STL-1400 will also initiate an AUTOMATIC FIRMWARE RESTART after the EEPROM update for proper re-initialization of the system.

Sending this command once results in a warning message being output. Sending the SYSTem:EEPROMUPDATE ONCE command a second time performs the EEPROM update.

In case of corrupted/non-functional EEPROM, this command is recommended over the SYSTem:FACToryReset ONCE command since this command allows the user to update the STL-2600/STL-1400 EEPROM with the latest default parameters.

> **ALERT**
> This command should not be used during normal field operation.

SERVO Subsystem

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

- SERVo:DACGain <float> [0.001,10000]
- SERVo:EFCScale <float> [0.0,500.0]
- SERVo:EFCDamping <int> [2,4000]
- SERVo:SLOPe <NEG|POS>
- SERVo:PHASECOrrection <float> [-500.0,500.0]
- SERVo:COARSedac <int> [0,255]
- SERVo:1PPSoffset <int> ns [-500000000,500000000]
- SERVo:TRACe <int> [0,255]
- SERVo:TRACe:SCPI <int> [0,255]

- `SERVo:LOOP <ON|OFF>`
- `SERVo:CLOCK:STeer <ON|OFF>`
- `SERVo:CLOCK:STeer:INVerse <ON|OFF>`
- `SERVo?`

SERVo:DACGain <float> [0.001,10000]

This command is used for factory setup ONLY.

This command has the following format:

```
SERVo:DACGain [0.001,10000]
```

The following command will query the settings of this command:

```
SERVo:DACGain?
```

SERVo:EFCScale <float> [0.0,500.0]

This command controls the Proportional part of the PID loop. Typical values are 0.2 to 3.0 for different 1PPS reference inputs. Larger values increase the loop control at the expense of increased noise while locked. Setting this value too high can cause loop instabilities.

This command has the following format:

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

The following command will query the settings of this command:

```
SERVo:EFCScale?
```

SERVo:EFCDamping <int> [2,4000]

This command adjusts the damping factor of the Low Pass 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 <int> [2,4000]
```

The following command will query the settings of this command:

```
SERVo:EFCDamping?
```

SERVo:SLOPe <NEG|POS>

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 which is typically set to POS. This command has the following format:

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

The following command will query the settings of this command:

```
SERVo:SLOPe?
```

SERVo:PHASECOrrection <float> [-500.0,500.0]

This parameter sets the Integral part of the PID loop. Loop instability will result if the parameter is set too high. Typical values are 4.0 to 40.0 for different 1PPS reference inputs. This command has the following format:

```
SERVo:PHASECOrrection <float> [-500.0,500.0]
```

The following command will query the settings of this command:

```
SERVo:PHASECOrrection?
```

SERVo:COARSedac <int> [0,255]

This command sets the coarse DAC that controls the TCXO EFC. The STL-2600/STL-1400 control loop automatically adjusts this setting during disciplining and when the SERVo:LOOP? settings is ON.

The user should not have to change this value.

This command has the following format:

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

The following command will query the settings of this command:

```
SERVo:COARSeDac?
```

SERVo:1PPSoffset <int> ns [-500000000,500000000]

This command sets the STL-2600/STL-1400 1PPS signal's offset to UTC in 4.17ns steps. The default offset is 0 ns and maximum configurable 1PPS offset is +/- 500ms.

Using the SERV:1PPS command results in immediate phase change of the 1PPS output signal on pin 7 of J3 connector.

This command has the following format:

```
SERVo:1PPSoffset <int> ns [-500000000,500000000]
```

The following command will query the settings of this command:

```
SERVo:1PPSoffset?
```

SERVo:TRACe <int> [0,255]

This command sets the period in seconds for the debug trace for the currently selected serial port receiving this command. Debug trace data can be used with Ulrich Bangert's "Plotter" utility to show UTC tracking versus time etc.

This command has the following format:

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

The following command will query the settings of this command:

```
SERVo:TRACe?
```

An example trace output is described here:

```
Trace: 0 109578 41835.32 127 372.92 334.78 7.47E-10 0.0
0.0000000 2950.31 0.00000 0.00 0 0 1 375.00 0x0
```

```
Trace: [Current Loop][1PPS Count][Fine DAC][Coarse DAC][Raw
UTC offset ns][Filtered UTC offset ns][Frequency Error
Estimate][Reserved][Reserved][Reserved][Reserved][Reserved]
[Reserved][Reserved][Lock Status][Raw 1PPS reference
offset][Health Status]
```

Please see "SYNChronization?" on page 76 for detailed information on how to decode the health status indicator values.

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

This command sets the period in seconds for the debug trace for the TTL SCPI serial port. Debug trace data can be used with Ulrich Bangert's "Plotter" utility to show UTC tracking versus time etc.

This command has the following format:

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

The following command will query the settings of this command:

```
SERVo:TRACe:SCPI?
```

An example trace output is described here:

```
Trace: 0 109578 41835.32 127 372.92 334.78 7.47E-10 0.0
0.0000000 2950.31 0.00000 0.00 0 0 1 375.00 0x0
```

```
Trace: [Current Loop][1PPS Count][Fine DAC][Coarse DAC][Raw
UTC offset ns][Filtered UTC offset ns][Frequency Error
Estimate][Reserved][Reserved][Reserved][Reserved][Reserved]
[Reserved][Reserved][Lock Status][Raw 1PPS reference
offset][Health Status]
```

Please see “SYNChronization?” on page 76 for detailed information on how to decode the health status indicator values.

SERVo:LOOP <ON|OFF>

This command enables or disables the servo loop. With the loop disabled, no changes are made to the oscillator frequency control.

Disabling the loop is recommended when an external reference oscillator is connected without the capability to discipline this external oscillator. If accessible, the user can connect the EFC output signal at J2 connector to the EFC input pin of the external TCXO or OCXO. By doing so, the STL-2600/STL-1400 can discipline this oscillator (see “Initial Operation” on page 25 for more details). Adjustment of the loop time constants is recommended and please refer to “Loop Parameter Adjustment” on page 27 for more details.

Enabling the servo loop will allow for 1PPS disciplining of the internal TCXO or external oscillator with EFC signal connected.

This command has the following format:

```
SERVo:LOOP <ON|OFF>
```

The following command will query the settings of this command:

```
SERVo:LOOP?
```

SERVo:CLOCK:STeer <ON|OFF>

This command will enable or disable the capability to notify the STL Kalman filter whenever the internal TCXO is being steered. This command aims to mitigate the frequency feedback loop avoiding the STL Kalman filter attempting to re-adjust the clock drift due to the internal TCXO disciplining.

The default settings is ON and the user should not have to change this settings.

This command has the following format:

```
SERVo:CLOCK:STeer <ON|OFF>
```

The following command will query the settings of this command:

```
SERVo:CLOCK:STeer?
```

SERVo:CLOCK:STeer:INVerse <ON|OFF>

This command will enable or disable the inverse sign of steering value on the internal TCXO being reported to the STL Kalman filter.

The default settings is ON and the user should not have to change this settings. Please refer to the SERVo:CLOCK:STeer <ON|OFF> command for the capability of notifying the STL Kalman filter.

`This command has the following format:`

SERVo:CLOCK:STeer:INVerse <ON|OFF>

The following command will query the settings of this command:

```
SERVo:CLOCK:STeer:INVerse?
```

SERVo?

This command returns the result of the following queries:

- `SERVo:LOOP?`
- `SERVo:COARSeDac?`
- `SERVo:DACGain?`
- `SERVo:EFCScale?`
- `SERVo:PHASECOrrection?`
- `SERVo:EFCDamping?`
- `SERVo:SLOPe?`
- `SERVo:1PPSoffset?`
- `SERVo:TRACe?`

4

Firmware Upgrade

The following topics are discussed in this chapter:

Introduction

The following is a short tutorial on how to upgrade the STL-2600/STL-1400 firmware. Please follow the instructions in order to prevent accidentally corrupting the STL-2600/ STL-1400 Flash, which may require reflashing at the factory.

With some practice, the entire Flash upgrade can be done in less than a couple of minutes, even though the following seems like a fairly long list of instructions.

ISP Flash Loader Utility installation

VIAVI recommends using the JLTerm application to upgrade the contents of Flash memory on the STL-2600/1400. The JLTerm application can be downloaded for free from the support page at the link provided below. Alternatively, you can also use the STM32ISP utility to perform the same upgrade. The STM32ISP utility is available from JLT upon request.

The JLTerm application is available for download at:

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

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

Two methods are supported for enabling the STL-2600/STL-1400 in System Programming (ISP) mode. Issuing the SCPI command SYST:ISP discussed in “SYSTem:ISP” on page 89 in JLTerm or another terminal program causes the board to reset into ISP mode from normal operation. This is automatically done by JLTerm if the “update device firmware...” button is pressed.

Although the JLTerm is recommended, the STM32ISP utility will be useful with hardware ISP entry if the STL-2600/STL-1400 has stopped communicating.

The board can also be put into ISP mode by first shorting-out the ISP# pin 13 of J3 to +3.3V. +3.3V can be found at the end of inductor D3 (between D3 and D4). Then, short pin 15 (RST_IN#) of J3 to Ground of the STL-2600/STL-1400 board during power-on..

Downloading the firmware

Request the latest version of STL-2600/STL-1400 firmware as well as downloading JLTerm from the VIAVI support page 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 STL-2600/STL-1400 is Micro-USB compatible. The firmware can also be upgraded via the SCPI TTL serial port when using JLTerm.

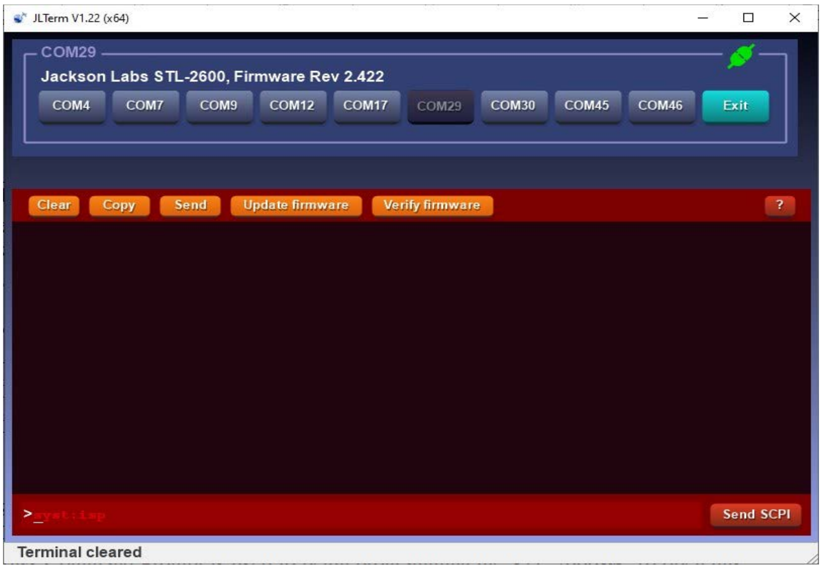
Using the JLTerm programming terminal

1. Download the JLTerm application from https://www.viavisolutions.com/en-us/software-download/gpscon-controller-software. Install and open the JLTerm application.
2. 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 10. The device must be in normal operation with working SCPI communication prior to JLTerm connection.

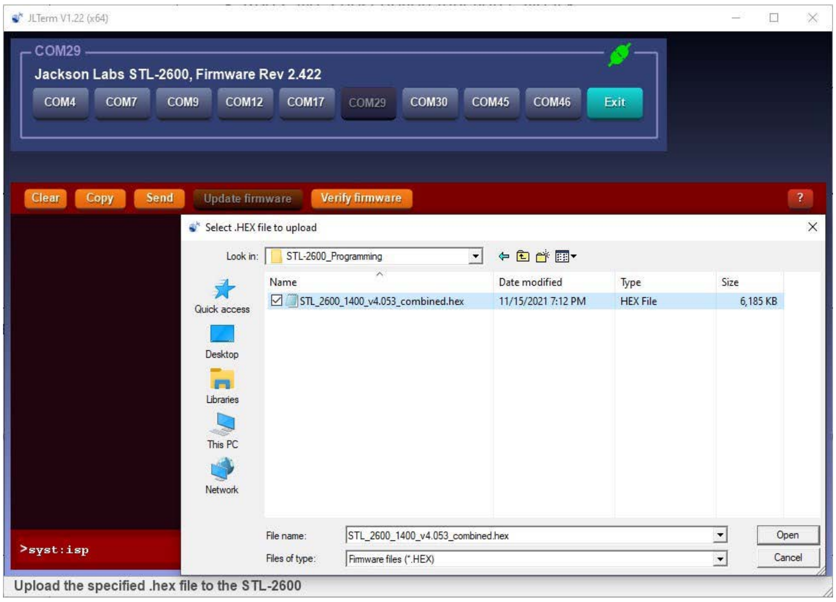
   > **NOTE**
   >
   > If there is no valid response from the COM port, check for valid driver and port number for the COM port in the Device Manager on your computer. The COM poer number may be conflicting with another COM port device. Each device should have a different COM port. Ensure GPSCon is not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

   Figure 10 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.
3. Once the device is connected in JLTerm, click the orange **Update device firmware...** button.
4. Choose the correct hex file to program the device and click **Open** in the pop-up, as shown in Figure 11. The firmware automatically downloads and the board is reset.

Figure 11 Open Hex file and download firmware



5 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 STM32ISP programming utility" on page 100 and attempt to finish the firmware upgrade using the STM32ISP programming utility.

Figure 12 shows an error in programming.

Figure 12 Error in programming

Using the STM32ISP programming utility

The following procedures describe how to use the STM32ISP programming utility.

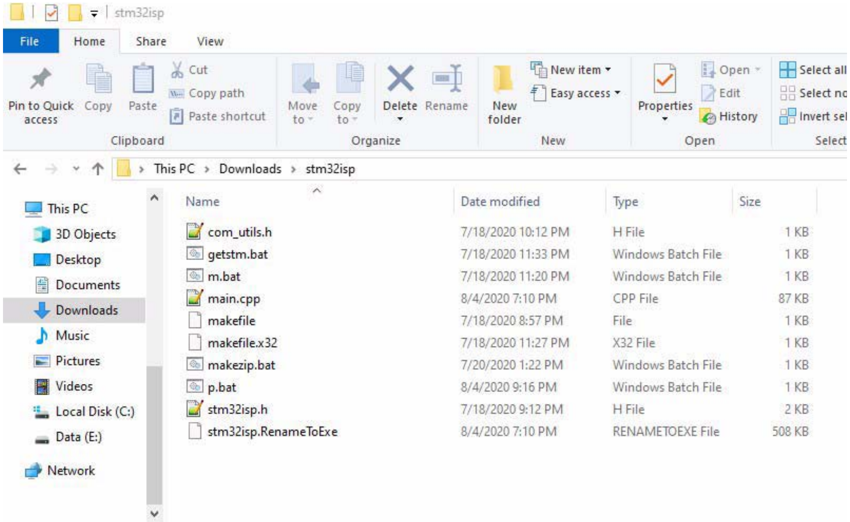
To prepare to use the STM32ISP programming utility

1 If:

- **a** The SCPI interface on STL-2600/STL-1400 board is not communicating properly, put the STL-2600/STL-1400 into ISP mode. See “Putting the PCB into In-Circuit Programming (ISP) mode” on page 98 for instructions to do so using the ISP and RESET# pins on connector J3.
- **b** The STL-2600 board can communicate and accept the SCPI command, send the SYST:ISP command on a terminal program with current baud rate

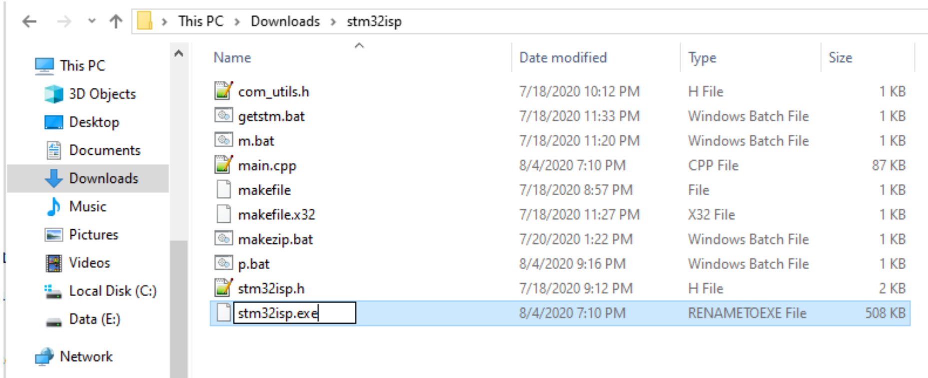
2 Request the latest STM32ISP utility and programming script from JLT. Install and open the STM32ISP utility folder as shown in Figure.

Figure 13 STM32ISP utility folder



3 Rename the file stm32isp.RenameToEXE to stm32isp.exe, as shown in Figure.

Figure 14 Rename stm32isp.RenameToExe file



4 Open the provided dual_mcu.bat file with a text editor.

> **NOTE**
> If the dual_mcu.bat file was not provided, create the file using a text editor to copy the contents of Figure.

5 Modify the file to point to correct path, file names, COM port, and baud rate to allow the utility to program the STL-2600. The command must have either of the following formats:

a *stm32isp, COM#, programming baud, write and/or verify, path of HEX file, and starting address after programming*

b *stm32isp, COM#, connection baud, send, and SCPI command to be sent*

The following is an example command:

```
stm32isp_v3.22 COM29 921600 WRITEREAD
C:\Downloads\STL_2600_1400_v4.053_combined.hex,master
0x08000000
```

Figure shows a command format for reference.

Figure 15 Example dual_mcu.bat file

C: > Users > JLT > STL-2600_Programming > dual_mcu.bat

```
@echo off

stm32isp_v3.22 COM29 921600 WRITE,NOPAUSE STL_2600_1400_v4.053_combined.hex,master 0x08000000 && ^
timeout 5 && ^
stm32isp_v3.22 COM29 921600 SEND,NOPAUSE "SYST:MASTERBRIDGE" && ^
timeout 3 && ^
stm32isp_v3.22 COM29 460800 WRITE,NOPAUSE STL_2600_1400_v4.053_combined.hex,slave 0x08000000
```

6 Save and close the dual_mcu.bat file.

7 Open the Windows Command Prompt :

a In the Windows search bar, type **`cmd`.**

b Select the Command Prompt application.

8 Navigate to the directory containing the stm32isp.exe utility by entering the cd command along with the file path for the stm32isp folder, as shown in figure.

Figure 16 Navigate the Windows command prompt

Administrator: Command Prompt

```
Microsoft Windows [Version 10.0.19041.1165]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd ..\..\Users\JLT\STL-2600_Programming

C:\Users\JLT\STL-2600_Programming>
```

9 Type **dual_mcu.bat** in the Command Prompt application and press the **Enter** key.

In the Command Prompt window, you will observe the firmware being downloaded to the processor with steps shown in Figure 4.8. Once the firmware has completed downloading press any key on your keyboard to continue.

Figure 17 Successful STN32ISP Dual_MCU download


```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19042.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\JLT\STL-2600_Programming>stm32isp_v3.22 COM29 921600 WRITE,NOPAUSE STL_2600_1400_v4.053_combined.hex,master 0x08000000

stm32isp version 3.22

   File: STL_2600_1400_v4.053_combined.hex (11/15/2021 7:12:12 PM)
Section: master
   Port: \\.\COM29
   Baud: 921600
 Bank 0: 819712 bytes at 0x08000000
 Bank 1: 751360 bytes at 0x08100000
  Start: 0x08000000

Synchronizing with device at COM29...
STM32 UART bootloader version = 3.1
STM32 device ID = 0x450
Preparing device memory for writing...
Erase in progress...
 Device memory successfully prepared for writing
Uploading bank 0 data...
  100% complete
Bank 0 data successfully uploaded
Uploading bank 1 data...
  100% complete
Bank 1 data successfully uploaded
Executing at 0x08000000...

C:\Users\JLT\STL-2600_Programming>stm32isp_v3.22 COM29 921600 SEND,NOPAUSE "SYST:MASTERBRIDGE"

stm32isp version 3.22
Transmitting: `SYST:MASTERBRIDGE`
Received:
scpi >
scpi > SYST:MASTERBRIDGE
Warning: This command stopped normal receiver operation and
configured current serial port for Slave CPU programming mode.
Follow instructions in user manual for firmware upgrade.
After timeout of inactivity, the system
Received:  will reset automatically
and resume normal operation mode.


C:\Users\JLT\STL-2600_Programming>stm32isp_v3.22 COM29 460800 WRITE,NOPAUSE STL_2600_1400_v4.053_combined.hex,slave

stm32isp version 3.22

   File: STL_2600_1400_v4.053_combined.hex (11/15/2021 7:12:12 PM)
Section: slave
   Port: \\.\COM29
   Baud: 460800
 Bank 0: 349696 bytes at 0x08000000
 Bank 1: 520192 bytes at 0x08100000

Synchronizing with device at COM29...
STM32 UART bootloader version = 3.1
STM32 device ID = 0x450
Preparing device memory for writing...
Erase in progress...
 Device memory successfully prepared for writing
Uploading bank 0 data...
  100% complete
Bank 0 data successfully uploaded
Uploading bank 1 data...
  100% complete
Bank 1 data successfully uploaded

C:\Users\JLT\STL-2600_Programming>
```

10 Verify the firmware update as described in "Verifying the firmware update" on page 103.

Verifying the firmware update

1 Power cycle the unit with the pin 13 of J3 (ISP) and pin 15 of J3 (RESET#) left floating.

2 During power on, the unit sends an ID string out of the USB serial port at 921600 Baud (115200 Baud for SCPI TTL serial port) by default. The firmware version can also be queried by sending the *IDN? command.

3 Verify that the firmware revision number that was downloaded.

NOTE

If errors occur repeatedly during the steps in “Using the JLTerm programming terminal” on page 99 and/or “Using the STM32ISP programming utility” on page 100, contact VIAVI support.

5

GPSCon Utility

The following topics are discussed in this chapter:

Description

GPSCon – VIAVI Edition is a free program for the monitoring and control of VIAVI GPSDO, Simulator, and Receiver products. It communicates with the receiver using the SCPI command set. This free version of the GPSCon utility is only compatible with VIAVI products, and is available for download from the following 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.

STL-2600 serial port config before using GPSCon

The STL-2600 baud rate of up to 921600 is now supported in latest version of GPSCon utility, however several settings are required for smooth connection with GPSCon including ON settings in `SYSTem:COMMunicate:SERial:ECHO <ON|OFF>` and `SYSTem:COMMunicate:SERial:PROmpt <ON|OFF>` commands, clean serial port with no periodic/NMEA output messages.

If current connected serial port is USB port and some periodic messages are still enabled, the `SYSTem:COMMunicate:SERial:OUTput <USBOFF|SCPIOFF|ALLOFF>` command with `USBOFF` settings can be used to disable all messages on USB port. Similarly, the `SCPIOFF` settings will disable all messages on SCPI port. Connect current selected serial port and correct COM port to a terminal program (TeraTerm is recommended), send necessary command(s) listed above to the STL-2600, then disconnect TeraTerm.

Start GPSCon using the correct COM port for STL-2600.

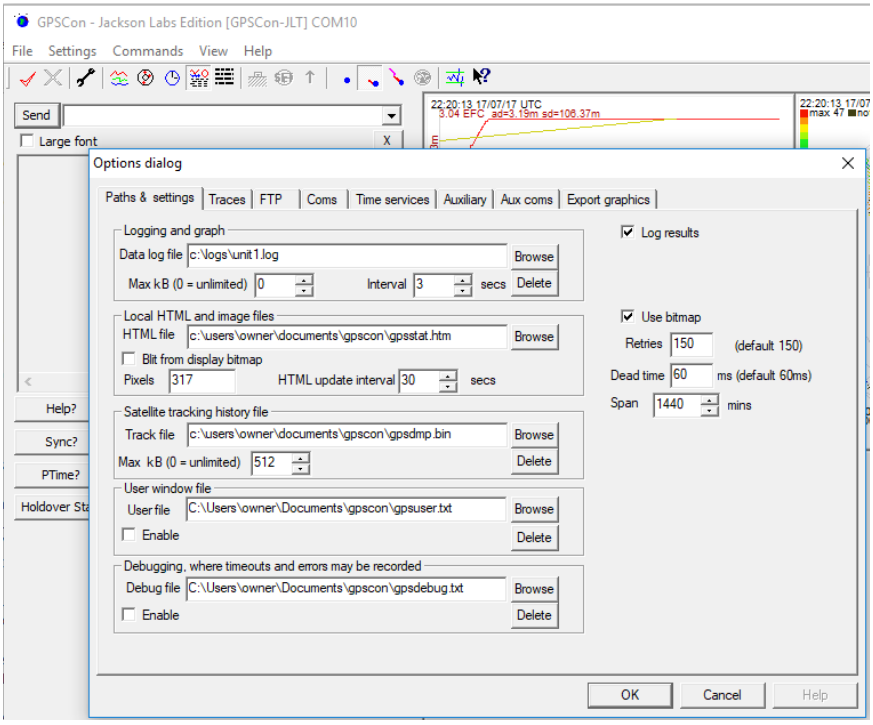
Using GSPCon

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 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 18 will appear. You can select from the tabs which options you wish to set.

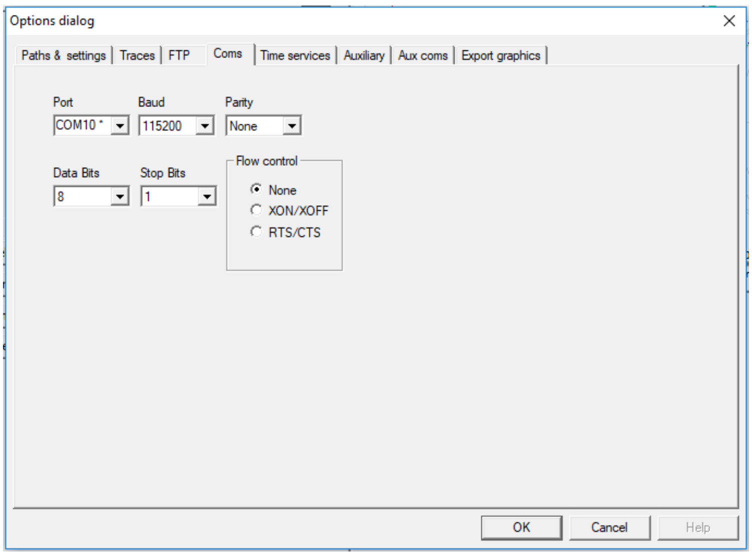
Figure 18 Options window



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 , then select the **Coms** tab. The Options window opens on the Coms tab, as shown in Figure 19.

Figure 19 Coms parameters tab

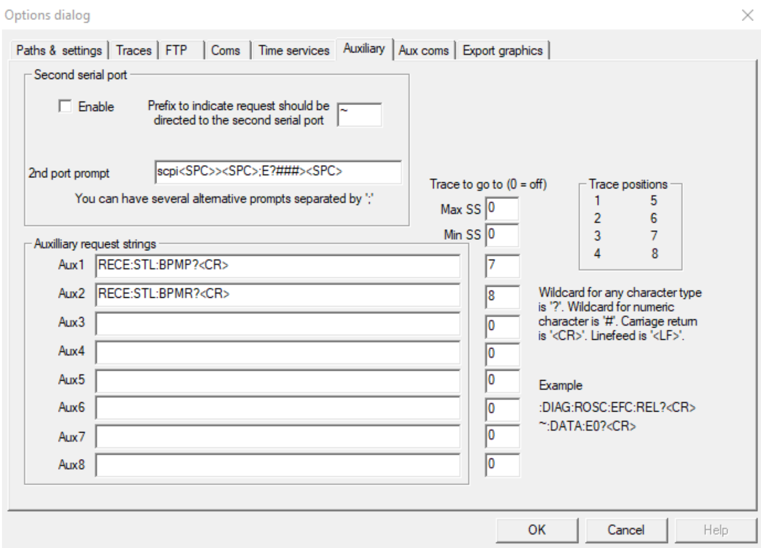


Available COM ports for your particular computer are indicated with a “*” symbol next to the COM port number in the drop-down menu. You can only select COM ports with this adjacent symbol. Select the correct COM port for your computer’s serial port connection and set the baud rate to 115200, parity to 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. Figure 20 shows an example of an auxiliary measurement.

Figure 20 Auxiliary parameters tab

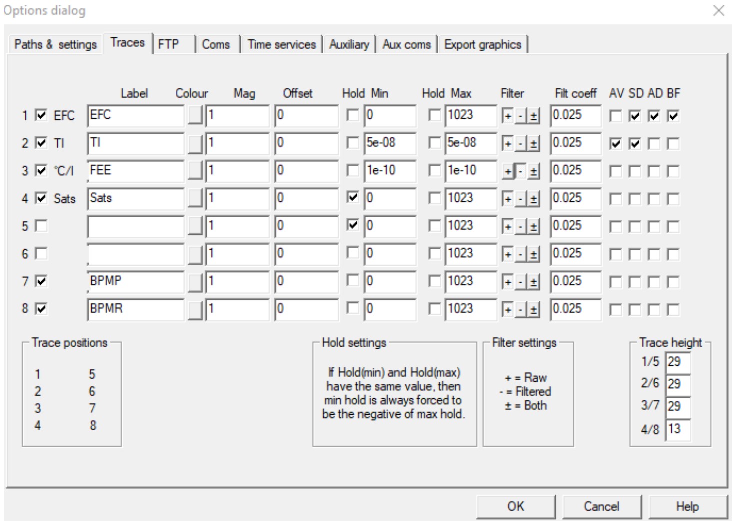


In the above example, the **Aux1** request string has been set to `GPS:STL:BPMP?<CR>` and the **Aux2** request string has been set to `GPS:STL:BPMR?<CR>`. The **Trace to go to** is set to trace positions 7 and 8 respectively. See the **Trace Position** diagram for the arrangement of the trace positions in the trace window. In this example, the bursts process and recevied obtained from the `GPS:STL:BPMP?` and `GPS:STL:BPMR?` queries will be plotted in trace position 7 and 8.

Traces parameters

After pressing the wrench icon , you can select the **Traces** tab and configure the trace labels and vertical plot ranges. Figure 21 shows an example of the trace parameter tab.

Figure 21 Traces parameters tab

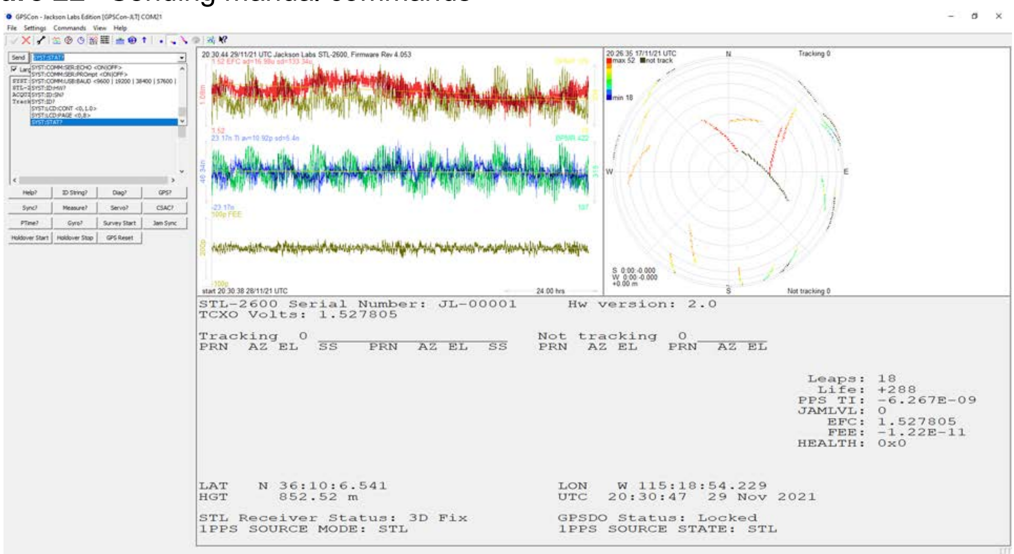


The labels and parameters are completed by default for traces 1 through 5. Any of the eight traces can be replaced by auxiliary traces as described in "Auxiliary parameters" on page 108. Press the **Help** button for a full description of each option in the Traces tab.

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 22.

Figure 22 Sending manual commands



Once a command is selected, press **Send** to send it to the STL-2600. You can also send common commands by clicking on the buttons below the message window. Hover over the buttons to see the exact command that is sent.

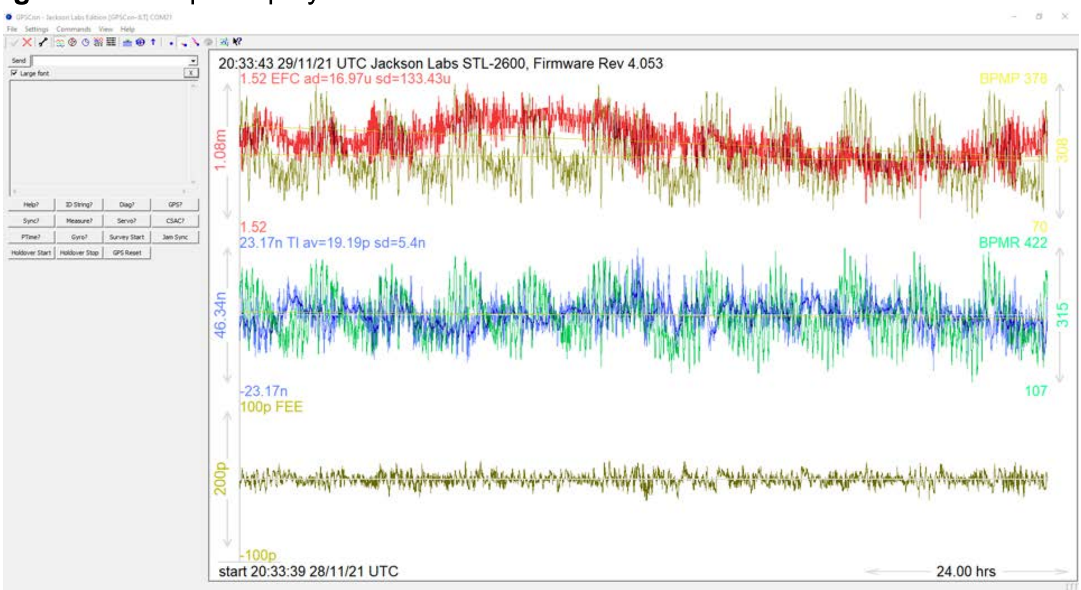
If the desired SCPI command is not listed in the drop-down box or command section, you can manually type the command in the text box and press the “Send” button.

> **NOTE**
> Ensure the selected command is supported by the STL-2600.

Using the mouse in the Graph display

Figure 23 shows the Graph display. 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.

Figure 23 Graph display



You can set the horizontal range of the graph using the mouse. Perform the following steps to set the horizontal range.

1. Set the start time by clicking on the point that marks the left side of the curve to be magnified.
2. Set the stop time by right-clicking on the point that marks the right most portion of the magnified curve.

> **NOTE**
> The Set and Stop times can be removed by double-clicking anywhere on the graph.

3. Return to the “Fit to window” view by double left-clicking 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

Exporting 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 be the default next time you open this dialog.

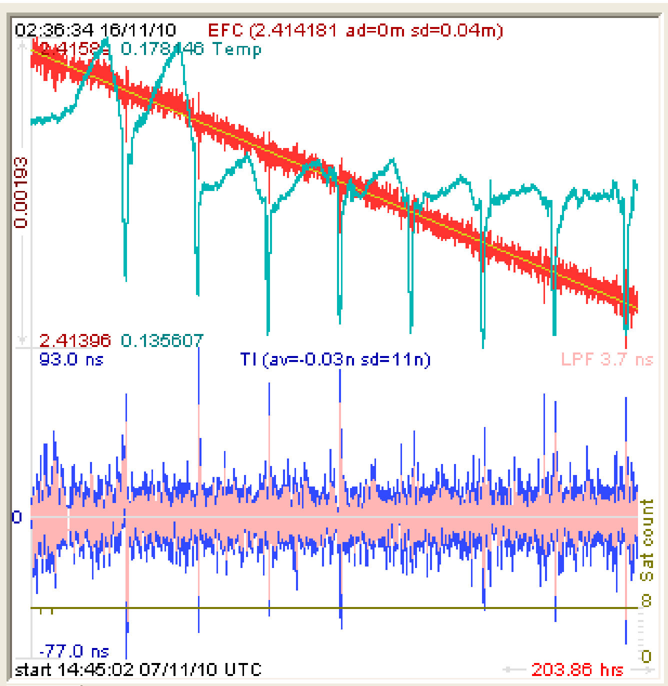
If you choose to export the graph, you can override the TI max setting in force on the screen display by entering a non-zero value into the 'Override TI' control. A value of zero causes the export to take the same setting if any as the screen display.

- To export automatically on a timed basis, 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.

Interpreting the Data

Figure shows an example of data acquired by a GPSDO unit over a period of time of more than 200 hours.

Figure 24 Captured data example



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 ~2 mV is visible over 200 hours. On the left side of teh screen, the EFC range over this 200 hour plot is displayed vertically as 0.00193 V. This means the drift of the EFC voltage due to aging is ~88 mV per year. The EFC sensitivity of the crystal is about 8 Hz per volt, so the crystal ages at:

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

At 10 MHz:

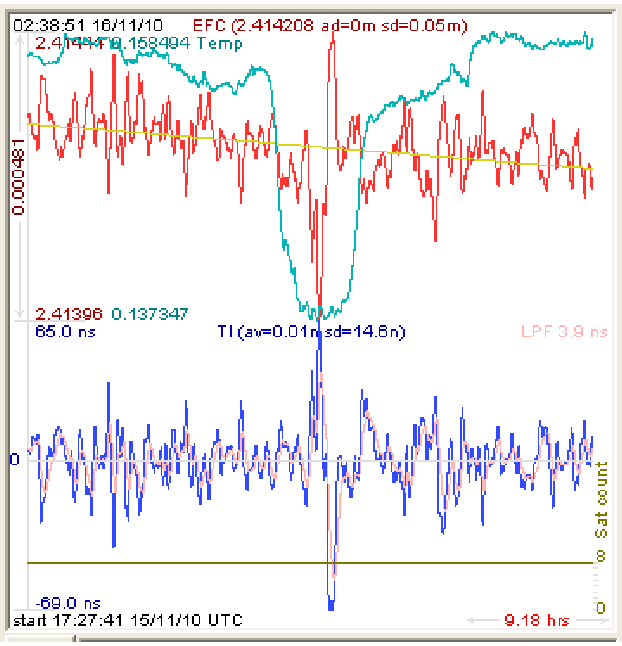
0.704Hz / 10MHz = 7.04E-08 aging rate per year

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 tortoise. 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.

Figure 25 shows a zoom of the captured data.

Figure 25 Zoomed captured data example



The image shows a phase offset error of the internal OCXO to the UTC GPS reference. The maximum drift is -77 ns to +93 ns. The average is (TI_{av} = -0.03 ns). The standard deviation over the 200 hour plot is sd = 11 ns. This means the average error of the 10 MHz phase of this unit over 200 hours is only +/- -11 ns. Or, in other words, the average jitter (wander) over 200 hours is:

11ns / 200Hrs = 1.528E-014

The unit performs as well as a high quality Cesium Atomic reference clock over long periods of time. The unit disciplines its internal 10MHz reference to within less than +/-

80ns peak to peak of UTC at all times, which is less than one complete clock cycle at 10MHz.

80200598
R001, May 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
Address	1445 South Spectrum Blvd., Suite 102, Chandler, AZ, 85286, USA