ULN-2550 25MHz
Quick Start Instructions – rev. 1.1

The following is a short description on how to use the ULN-2550 GPSDO.

Table of Contents

Introduction

The ULN-2550 GPSDO includes an extremely high-performance GPS receiver that can acquire and track up to 50 GPS signals down to a state of the art –160dBm, a 32bit processor that runs a Real Time OS, one 100MHz CMOS output, one 10MHz CMOS output, four 25MHz LVDS outputs, 1PPS UTC synchronized LVDS and CMOS outputs, RS-232 5V-level control interface, and precision voltage references and DACs.

Powering Up the Unit

To operate the unit, simply follow these two steps:

1) Connect a 5V-compatible GPS antenna to connector J2.

2) Connect +12V DC Power to pins 5, and 7 of power connector U3, ground to pins 3 and 9 of U3.

The unit will now lock to GPS (Red LED is blinking when satellites are being received) and will indicate proper lock when the Green LED goes on. Once the green LED is on, the unit will output 10MHz, 25MHz, and 100MHz on the various outputs with significantly better than 1ppb frequency accuracy.

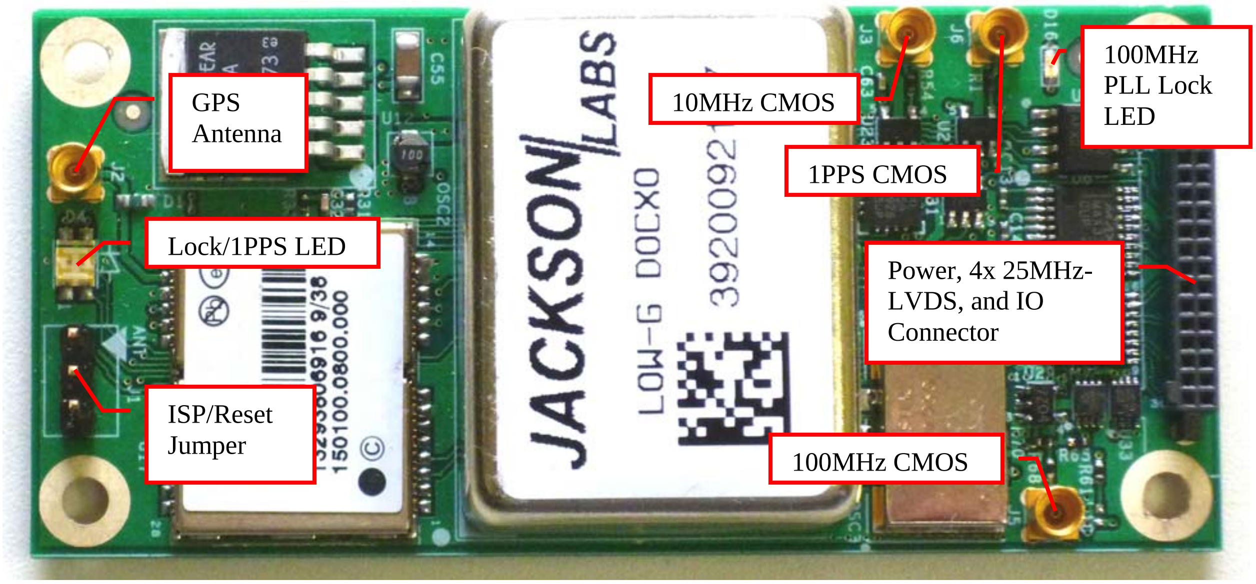
Please note that the GPS receiver establishes the internal antenna gain right after power-on, so for proper operation the GPS antenna should always be connected prior to turning on the +12V power.

PLEASE NOTE:

Please note that the RS-232 interface is a TTL level-only interface by default. An optional RS-232 level-translator may be stuffed onto the PCB by the factory. **Connecting the interface signals directly to an RS-232 level interface will damage the board, and must be avoided!**

Major connections

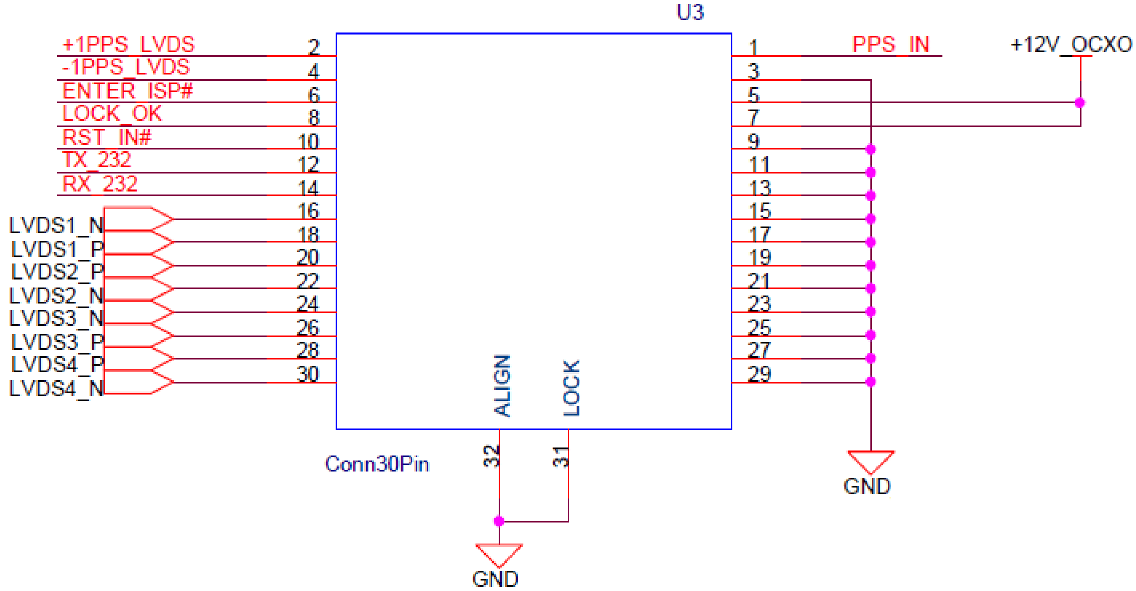
The major connections and features of the ULN-2550 PCB are shown below:



The following table shows the ULN-2550 revision 1.0 hardware connectors:

Ref	Name	Function	Specification	Pinning
J2	Antenna	GPS Antenna	5V Amplified Antenna MMCX connector	Center-RF Input, Shield-GND
J1	ISP/RESET	Reset and ISP	Pull pins 1 and 3 to GND to activate function	3-RESET-IN#, 2-GND, 1-ENTER_ISP#
J3	10MHz CMOS	10MHz CMOS Out	+5V CMOS, <2ns risetime	Center-RF, Shield-GND
J5	100MHz CMOS	100MHz CMOS Out	+3.3V CMOS, <2ns risetime	Center-RF, Shield-GND
J6	1PPS CMOS	UTC-Aligned 1PPS Out	5Vpp CMOS, Rising Edge Aligned, <2ns risetime	Center-RF, Shield-GND
U3A	ISP# and RESET#	External Reset and ISP select	Internally Pulled-Up, pull to ground to reset board, and to initiate Flash Update	U3.6-ENTER_ISP#, U3.10-RESET#
U3B	1PPS LVDS	1PPS LVDS Output	LVDS, +/-300mV differential	U3.5-1PPS_Pos, U3.4-1PPS_Neg
U3C	1PPS IN	External 1PPS input (optional)	3.3V CMOS, risetime <10ns	U3.1-1PPS-In, U3.3-GND
U3D	Green LED (OCXO LOCK)	Follows status of Green LED	3.3V CMOS	U3.8-LED+, U3.9-GND
U3E	RS-232 SCPI	RS-232 Serial Port	TTL Level, 115.2,8N1	U3.12-TX, U3.14-RX, U3.13-GND
U3F	+12V Power	Main Power Input	+11.0V to +14.0V	U3.5-+12V, U3.7-+12V, U3.3-GND, U3.9-GND
U3G	Ground	Ground	Ground	U3.3, U3.9, U3.11, U3.13, U3.15, U3.17, U3.19, U3.21, U3.23, U3.25, U3.27, U3.29

Typical Connections to Power Connector U3:



Power Harness Pinning

The following is a table of the power cable harness U3 pinout:

PIN	NAME
1	1PPS In CMOS level
2	+1PPS LVDS
3	GND
4	–1PPS LVDS
5	+12V
6	ENTER_ISP#
7	+12V
8	Green LED Signal
9	GND
10	RESET#
11	GND
12	TX TTL-Level RS-232
13	GND
14	RX TTL-Level RS-232
15	GND
16	25MHz LVDS-
17	GND
18	25MHz LVDS+
19	GND
20	25MHz LVDS+
21	GND
22	25MHz LVDS-
23	GND
24	25MHz LVDS-
25	GND
26	25MHz LVDS+
27	GND
28	25MHz LVDS+
29	GND
30	25MHz LVDS-

Harness Connectors

The manufacturer for connector U3 is Samtec. The part number of the connector soldered onto the ULN-2550 PCB is:

Samtec SFM-115-03-S-D-LC

NOTES:

- The LVDS signals (+/-25MHz, +/-1PPS) should be routed using 100Ohm differential wiring (twisted pair), or two 50Ohm single-ended coaxial cables with the shields being connected to ground. Terminate these signals with 100 Ohm resistors between the positive and the negative wires.

- The Green LED Lock signal is a 3.3V CMOS signals, and thus require a series resistor of typically 390 to 470 Ohms when used to drive LED's.

- The 10/100MHz 50 Ohm CMOS outputs do not require a termination if they remain unused.

Power

The unit is powered from a +11 .0V to +14.0V DC source, with +12.0V nominal voltage. The current is typically less than 0.35A at 12V. Connect a clean +12V power supply to pins 5 and 7 of the cable harness U3.

Do not reverse the polarity of the power pins, this will damage the unit.

Connecting the GPS Antenna

Connect the GPS antenna to the BNC to MMCX cable adapter. Caution: use a Lightning Arrestor on your Antenna setup. Use an amplified GPS antenna that is 5V LNA compatible. The ULN-2550 GPS receiver is a 50 channel high-sensitivity GPS receiver with very fast lock time. It does not require any self-survey or position-hold mode (auto survey), and thus can be used in mobile platforms.

Please note that the ULN-2550 unit supports 5V active, or passive antennae.

ULN-2550 is capable of generating standard navigation messages (see GPS:GPGGA, GPS:GPZDA, and GPS:GPRMC RS-232 commands) that are compatible with most GPS based navigation software.

The GPS receiver generates a 1PPS time signal that is phase synchronized to UTC. This 1PPS signal is used to frequency-lock the 10MHz Sine-Wave output of the ULN-2550 GPSDO to UTC, thus disciplining the units' 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).

Remote serial control

- **The serial interface is a TTL-level only interface - DO NOT CONNECT the RX, TX, and GND pins of the cable harness U3 to a standard RS-232 connector. A TTL to RS-232 level shifter is required if the unit is to be connected to a standard RS-232 interface.**

- The unit is controlled via the Serial port at 115200 baud, 8N1. Other Baud Rates can be set via SCPI commands.

Attach the ULN-2550 unit to your PC's Hyperterminal, the optional GPSCon software package, or a third-party freeware Windows-based application program called Z38xx.

The Z38xx program can be used to track the performance of the ULN-2550. Z38xx is available on the Jackson Labs Technologies, Inc. website under the following URL:

http://www.jackson-labs.com/docs/Z38xx.zip

- An optional RS-232 level shifter can be built into the ULN-2550 PCB by the factory upon special order.

"Help" and command overview

- A listing of the available RS-232 commands can be shown by typing "help?"

- "*IDN?" can be used to see if the connection works. Both commands need to be followed by pressing "Enter"

Loop parameter adjustment

- All loop parameters can be controlled via the RS-232 serial port.
- Loop parameters are optimized for the OCXO on the board, and changing the factory settings may result in the units' performance to deteriorate.

The commands to control the loop parameters are part of the servo? command. See also the **SERVO Subsystem** section below.

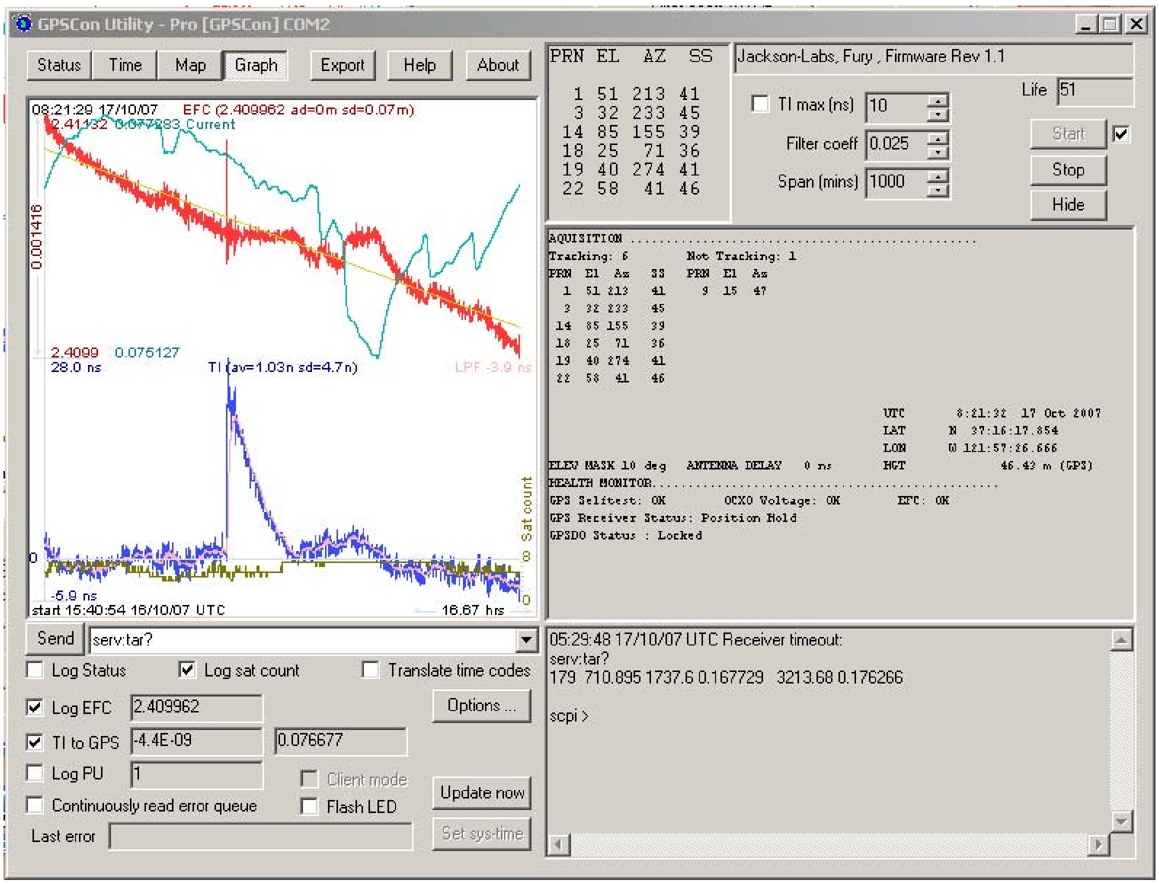
The individual commands are:

EFC Scale: this is the proportional gain of the PID loop. Higher values will give quicker convergence, and faster locking of the GPS time (lower loop time constant), lower values give less noise. Values between 0.7 (good double oven OCXO) and 6.0 (simple single-oven OCXO) are typical.

EFC Damping: overall IIR filter time constant. higher values increase loop time constant. Jackson Labs typically uses values between 10 to 50. 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 ULN-2550 1PPS signal and the UTC 1PPS signal as generated by the GPS receiver. Set higher values for tighter phase-following at the expense of frequency stability. Typical values range from 4 - 30, 25 being the default. Setting this value too high may cause loop instability.

A well-compensated unit will show performance similar to the following plot when experiencing small perturbations:



SCPI-Control Quick Start Instructions

The SCPI subsystem is accessed via the RS-232 interface and a terminal program. By default the terminal settings are 115200, 8N1.

There are a number of commands that can be used as listed below. Most of these are identical or similar to Symmetricom 58503A commands.

GPS Subsystem

Please note that FireFly-IIA displays antenna height in MSL Meters rather than in GPS Meters on all commands that return antenna height [the legacy Fury GPSDO uses GPS 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 GPS subsystem regroups all the commands related to the control and status of the GPS receiver. The list of the commands supported is the following :

GPS:SATellite:TRAcking:COUNt?

GPS:SATellite:VISible:COUNt?

GPS:GPGGA <int> [0,255]

GPS:GGASTat <int> [0,255]

GPS:GPRMC <int> [0,255]

GPS:GPZDA <int> [0,255]

GPS:PASHR <int> [0,255]

GPS:GYRO?

GPS:GYRO:CAL?

GPS:DYNAMic <int> [0,7]

GPS?

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

GPS:REFerence:PULse:SAWtooth?

GPS:RESET ONCE

GPS:SATellite

This group of commands describe the satellite constellation.

GPS:SATellite:TRAcking:COUNt?

This query returns the number of satellites being tracked.

GPS:SATellite:VISible:COUNt?

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

NMEA Support

The following two commands allow the FireFly-IIA GPSDO to be used as an industry standard navigation GPS receiver. The GPGGA and GPRMC 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 FireFly-IIA RS-232 interface.

Once enabled, FireFly-IIA will send out information on the RS-232 transmit pin automatically every N seconds. All incomming RS-232 commands are still recognized by FireFly-IIA since the RS-232 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 GPS receiver internally reported these to the FireFly-IIA Microprocessor, so the position is valid for the 1PPS pulse previous to the last 1PPS pulse at the time the data is sent (one second delay). The time and date are properly output with correct UTC synchronization to the 1PPS pulse immediately prior to the data being sent.

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

GPS:GPGGA

This command instructs the FireFly-IIA to send the NMEA standard string $GPGGA every N seconds, with N in the interval [0,255]. The command is disabled during the initial 7 minute OCXO warmup phase.
This command has the following format:

    GPS:GPGGA   <int> [0,255]

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

GPS:GGASTat

This command instructs the FireFly-IIA to send a modified version of the NMEA standard string $GPGGA every N seconds, with N in the interval [0,255]. The command is disabled during the initial 7 minute OCXO warmup phase.
This command has the following format:

    GPS:GGASTat   <int> [0,255]

This command replaces the regular NMEA GGA validity flag with a decimal number indicating the lock-state of the unit. Please see section **SERVo:TRACe** for a detailed description of the lock state variable. The command allows capture of the position and other information available in the GGA command, as well as tracking the lock state and health of the units' OCXO performance.

GGASTat shows height in MSL Meters, this is different from traditional GPS receivers that display height in GPS Meters. The difference between MSL and GPS height can be significant, 35m or more are common.

GPS:GPRMC

This command instructs the FireFly-IIA to send the NMEA standard string $GPRMC every N seconds, with N in the interval [0,255]. The command is disabled during the initial 7 minute OCXO warmup phase.
This command has the following format:

GPS:GPRMC <int> [0,255]

GPS:XYZSPeed

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

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

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

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

GPS:XYZSPeed <int> [0,255]

GPS:GPZDA

This command instructs the FireFly-IIA to send the NMEA standard string $GPZDA every N seconds, with N in the interval [0,255]. The command is disabled during the initial 7 minute OCXO warmup phase.
This command has the following format:

GPS:GPZDA <int> [0,255]

GPS:PASHR

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

$PASHR,POS,0,7,202939.00,3716.28369,N,12157.43457,W,00087.40,????,070.01,000.31,-000.10,05.6,03.5,04.3,00.0,DD00*32

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

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

Where:

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

This command instructs the FireFly-IIA to send the NMEA standard string $PASHR every N seconds, with N in the interval [0,255]. The command is disabled during the initial 7 minute OCXO warmup phase.
This command has the following format:

GPS:PASHR <int> [0,255]

GPS:GYRO

This command sets up the output period of the Accelerometer data (in g's per axis, and in tilt in Degrees per axis). The fastest period is 50ms (20 outputs per second) achieved when setting the command to GPS:GYRO 1. The resolution of this command is 1/20 of a second.

Setting the command to 0 disables the Accelerometer output.

This command has the following format:

GPS:GYRO <int> [0,255]

GPS:GYRO? returns the setting of the period of this command.

The output of the system is in six floating point numbers: the acceleration on the X, Y, and Z axis in g's, and the Tilt of the X, Y, and Z axis in Degrees (-90 to +90).

Please note that the Accelerometer has it's best resolution and accuracy when the earth's gravitational acceleration is perpendicular to the axis to be measured. Thus the sensors accuracy and resolution will be best when the tilt of any axis is around 0 Degrees. The worst performance (with an accuracy of only ~ +/-5 Degrees) is when any of the axis are at their +/- 90 Degree angles, this is due to the fact that a SIN(g-load) calculation is done to generate the output normalized to Degrees, and the vector is thus least sensitive at +/-90 Degree angles.

Acceleration can be measured with up to +/-3g range on all three axis. Please note that since the tilt angle is calculated from the acceleration vectors (using gravity) that the tilt angles will actually exceed +/-90 Degrees when acceleration greater than 1g is applied.

GPS:GYRO:CAL

This command calibrates the output period of the Accelerometer data (in g's per axis, and in tilt in Degrees per axis). Both offset and gain need to be calibrated on the unit to properly format the output of the three axis tilt to be -90 Degrees to +90 Degrees, and the 0 Degree reference.

By default, the unit is shipped with a normalized output of 0 Degrees Offset, and a gain of 1.0, and the user will need to calibrate their unit in their system.

This command has the following format:

GPS:GYRO:CAL <float>, <float>, <float>, <float>, <float>, <float>

Calibration is accomplished by slowly tilting the unit on all six axis and measuring the maximum and minimum tilt indications in all six axis, and writing these down on a piece of paper to calculate the calibration values. Later firmware releases may automate this process.

Example: if the result for say the X axis is -87 Degrees to +93 Degrees, then we need to remove an offset of +3 Degrees from the X axis while the gain of the X axis is properly set. We would thus send the following calibration command to the unit:

GPS:GYRO:CAL -3, 0, 0, 1, 1, 1

This will remove a 3 Degree offset from the X Axis, and keep the Y and Z axis without offset or gain adjustments.

After the Offset is removed, we may have to adjust the gain of the axis sensor. For example if the range of the X Axis output is -82 Degrees to +78 Degrees then the axis has both an offset of 2 degrees, and a gain error of 90/80 Degrees (a factor of 1.125).

Thus we need to add 2 Degrees offset, and a gain of 1.125. The calibration command would thus look as follows:

GPS:GYRO:CAL 2, 0, 0, 1.125, 1, 1

All three axis need to be calibrated simultaneously as shown above for the X Axis.

GPS:GYRO:CAL? returns the setting of the calibration factors of the accelerometer system.

Due to the nature of the accelerometer chip, the Z axis will have the largest error and lowest sensitivity of all three axis. The X, and Y axis will have the best resolution and highest sensitivity when the unit is sitting level (horizontally).

GPS:DYNAMic?

This command allows the user to select the dynamic motion model being applied to the Kalmann filters in the GPS receiver. This allows for larger amounts of filtering for lower velocity applications, effectively reducing noise. Applications with high acceleration can now be used with fast filter settings to allow for the most accurate GPS coordinates to be provided in high-dynamic applications such as Jet aircraft.

Firmware version 1.04 and later add an automated feature where the unit will select the GPS profile itself based on the unit's velocity. This alleviates the user from having to track the state of a mission and manually selecting the best dynamic setting.

The command has the following syntax:

GPS:DYNAMic <int> [0,8]

The following settings are available:

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

As an example, sending the following command to the FireFly-IIA will select a stationary GPS dynamic model:

gps:dynam 1

GPS:DYNAMic 8 (Automatic Dynamic Mode)

The following table shows the Dynamic mode the unit will program into the GPS receiver when Automatic Mode is selected (Dynamic Mode 8). In this Automatic mode, the unit will configure the GPS based on the actual vehicle-velocity:

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

Please note that in order to switch from the Automotive mode into the first Airborne (1g) mode, both a vehicle velocity greater than 60 knots as well as a climb/descent rate greater than 400 feet per minute are required. Without an appropriate climb/descent, the unit will remain in Automotive mode.

The following command returns the setting of the GPS dynamic model:

GPS:DYNAMic?

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

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

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

GPS:REF:ADEL 45ns

This command can be used to fine-tune different units to have co-incident 1PPS pulse outputs. Please note that during normal operation the 1PPS pulse may wander around the UTC 1PPS pulse while the unit is tracking the GPS signals. The present offset between the 1PPS output and the UTC 1PPS signal can be queried with the command SYNC:TINT? and this offset should be taken into account when calibrating two units’ 1PPS outputs to each other, since the lock algorithms will try to steer the OCXO for a 0.0ns offset to the UTC 1PPS time-pulse.

GPS:REFerence:PULse:SAWtooth?

This command returns the momentary sawtooth correction factory that the GPS receiver indicated.

GPS:RESET ONCE

Issues a reset to the internal GPS receiver. This can be helpful when changing the antenna for example, since the GPS receiver measures the antenna systems' C/No right after reset, and adjusts its internal antenna amplifier gains accordingly. It takes approximately 1 minute for locking to commence after a GPS reset, as indicated by the red blinking LED.

GPS?

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

GYRO SUBSYSTEM

Firmware revision 2.18 and later adds various Gyro Accelerometer specific commands that allow easy interfacing to the Gyro subsystem. The following Gyro commands are supported:

GYRO:MODE <ON | OFF>
GYRO:TRACE <int> [0,255]
GYRO:CALibrate <float,float,float,float,float,float>
GYRO:CALibrate:COMPute
GYRO:CALibrate:RESET
GYRO:SENSitivity <float,float,float>
GYRO:EFC <float>
GYRO:GLOAD?

GYRO:MODE <ON | OFF>

Enables or disables electronic compensation. This feature must be calibrated at the factory before it can be used.

GYRO:TRACE <int> [0,255]

Similar to the GPS:GYRO command. Please see the GPS:GYRO command for details

GYRO:CAL <float,float,float,float,float,float>

Similar to the GPS:GYRO:CAL command. Used to manually calibrate the gain and offset of the Accelerometer output. Please see the GPS:GYRO:CAL command for details. See also the GYRO:CAL:COMPUTE command for a semi-automated calibration method.

GYRO:CAL:COMPUTE

This command can be used to automatically compute the offset and gain compensation of the accelerometer output for units that are not factory-calibrated. The goal is to have the unit indicate a range of +/- 1.0g on all three axis when slowly rotated through the axis, as well as 0.0g for any axis that is perpendicular to the earths' gravity.

The user must establish the minimum and maximum g-loads that are displayed on all three axis by slowly tilting the unit over 180 degrees on all three axis. These values are written down on a piece of paper, and the system calculates the required gain and offset parameters to scale the accelerometer output to +/- 1g peak-to-peak scale, and 0g indication based on entering these values. The GYRO:CAL:COMPUTE command will query the user for the maximum and minimum indicated g-loads. The GYRO:TRACE 10 command can be used to help establish the g-indications on the three different axis to aid in this calibration.

GYRO:SENS, GYRO:EFC, and GPS:CAL:RESET

Used for factory calibration of Gyro subsystem.

GYRO:GLOAD?

This command will return the present acceleration on the X, Y, and Z axis.

PTIME Subsystem

The PTIME subsystem regroups all the commands related to the management of the time.The list of the commands supported is the following :

PTIMe:TZONe?

PTIMe:DATE?

PTIMe:TIME?

PTIMe:TIME:STRing?

PTIMe:TINTerval?

PTIME?

PTIMe:TZONe?

Returns the local time zone offset.

PTIMe:DATE?

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

PTIMe:TIME?

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

PTIMe:TIME:STRing?

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

PTIMe:TINTerval?

This query is equivalent to the command SYNChronisation:TINTerval

PTIME?

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

PTIME:DATE?

PTIME:TIME?

PTIME:TZONE?

PTIME:TINTerval?

SYNChronization Subsystem

This subsystem regroups the commands related to the synchronization of the FireFly-IIA with the GPS receiver. The list of the commands supported for this subsystem is the following:

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

SYNChronization:SOURce:STATE?

SYNChronization:HOLDover:DURation?

SYNChronization:HOLDover:STATe?

SYNChronization:HOLDover:INITiate

SYNChronization:HOLDover:RECovery:INITiate

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

SYNChronization:TINTerval?

SYNChronization:IMMEdiate

SYNChronization:FEEstimate?

SYNChronization:LOCKed?

SYNChronization?

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 GPS, 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:INITiate

The SYNC:HOLD:INIT and SYNC:HOLD:REC:INIT commands allow the user to manually enter and exit the holdover state, even while GPS signals are still being properly received. This forced-holdover allows the unit to effectively disable GPS locking, while still keeping track of the state of the 1PPS output in relation to the UTC 1PPS signal as generated by the GPS receiver. When the unit is placed into forced-holdover with this command, the unit will indicate the time interval difference between the 1PPS output and the GPS UTC 1PPS signal by using the SYNC:TINT? command. This allows the user to see the OCXO drift when not locked to

GPS for testing purposes, or to prevent the GPS receiver from being spoofed and affecting the OCXO frequency accuracy. All other frequency-disciplining functions of the unit will behave as if the GPS antenna was disconnected from the unit while in this forced-holdover state.

SYNChronization:HOLDover:RECovery:INITiate

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

SYNChronization:SOURce:MODE

The Source:Mode command allows an optional external 3.3V level 1PPS input to be connected to the FireFly-IIA board on connector harness U18. The unit can use this external 1PPS input instead of the internal, GPS generated 1PPS. Switching to the external 1PPS is either done manually with the EXT command option, or automatically with the AUTO command option in case the GPS receiver goes into holdover mode for any reason. The command has the following format:

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

SYNChronization:SOURce:STATE?

This query shows the state of the external 1PPS synchronization option.

SYNChronization:TINTerval?

This query returns the difference or timing shift between the FireFly-IIA 1 PPS and the GPS 1 PPS signals. The resolution is 1E-10 seconds.

SYNChronization:IMMEdiate

This command initiates a near-instantaneous alignment of the GPS 1 PPS and Receiver output 1 PPS. 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 1000s measurement interval and comparing the internal 1PPS to GPS 1PPS offset.

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

SYNChronization:LOCKed?

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

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

This command allows the generation of the 1PPS pulse upon power-on without an external GPS antenna being connected to the unit. By default the unit does not generate a 1PPS pulse until the GPS 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 a GPS antenna is not connected to the unit. Once the GPS receiver locks, the 1PPS pulse will align itself to UTC by stepping in

10 equally spaced steps toward UTC alignment. The default setting is OFF which means the 1PPS pulse is disabled until proper GPS lock is achieved.

SYNChronization?

This query returns the results of these four queries :

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

SYNChronization? Health Status Indicator

The last line in the sync? query is a hexadecimal number indicating the systems 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 the following system status information:

If the OCXO coarse-DAC is maxed-out at 255 — HEALTH STATUS |= 0x1;
If the OCXO coarse-DAC is mined-out at 0 — HEALTH STATUS |= 0x2;
If the phase offset to UTC is >250ns — HEALTH STATUS |= 0x4;
If the run-time is < 300 seconds — HEALTH STATUS |= 0x8;
If the GPS is in holdover > 60s — HEALTH STATUS |= 0x10;
If the Frequency Estimate is out of bounds — HEALTH STATUS |= 0x20;
If the OCXO voltage is too high — HEALTH STATUS |= 0x40;
If the OCXO voltage is too low — HEALTH STATUS |= 0x80;
If the short-term-drift (ADEV @ 100s) > 100ns — HEALTH STATUS |= 0x100;
For the first 7 minutes after a phase-reset, or a coarsedac change:
HEALTH STATUS |= 0x200;

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

1) UTC phase > 250ns: 0x4
2) OCXO voltage too high: 0x40
3) GPS in holdover: 0x10

‘Oring’ these values together results in:

0x40 | 0x10 | 0x4 = 0x54

The unit would thus indicate: **HEALTH STATUS: 0x54**

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

DIAGnostic Subsystem

This subsystem regroups the queries related to the diagnostic of the OCXO.The list of the commands supported for this subsystem is as follows:

DIAGnostic:ROSCillator:EFControl:RELative?

DIAGnostic:ROSCillator:EFControl:ABSolute?

DIAGnostic:ROSCillator:EFControl:RELative?

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

DIAGnostic:ROSCillator:EFControl:ABSolute?

This query returns the Electronic Frequency Control (EFC) output value of the internal reference oscillator. It returns a value in volts between 0 and 5 V

MEASURE Subsystem

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

MEASure:VOLTage?

MEASure:CURRent?

MEASure?

MEASure:VOLTage?

This query returns the power supply voltage applied to the OCXO (ca. 10.45 V)

MEASure:CURRent?

This query returns the current drawn by the OCXO. This current varies in order to keep a stable temperature inside the OCXO.

MEASure?

This query returns the result of the three following queries:

MEASure:VOLTage?

MEASure:CURRent?

SYSTEM Subsystem

This subsystem regroups the commands related to the general configuration of the FireFly-IIA. The list of the commands supported for this subsystem follows:

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

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

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

SYSTem:STATus?

SYSTem:FACToryReset ONCE

SYSTem:COMMunicate

SYSTem:COMMunicate:SERial:ECHO

This command enables/disables echo on RS-232. This command has the following format:

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

SYSTem:COMMunicate:SERial:PROmpt

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

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

SYSTem:COMMunicate:SERial:BAUD

This command sets the RS-232 serial speed. The serial configuration is always 8 bit, 1 stop bit, no parity, no HW flow control. Upon Factory reset, the speed is set at 115200 bauds. This command has the following format:

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

SYSTem:STATus?

This query returns a full page of GPS status in ASCII format. The output is compatible with GPSCon.

SYSTem:FACToryReset ONCE

This command applies the Factory Reset setting to the EEPROM. All aging, tempco, and user parameters are overwritten with factory default values.

SERVO Subsystem

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

SERVo:COARSeDac <int> [0,225]

SERVo:DACGain <int> [0.1,10000]

SERVo: EFCScale <float>[0.0 , 500.0]

SERVo:EFCDamping <float>[0.0 , 4000.0]

SERVo:SLOPe <NEG | POS >

SERVo:TEMPCOmpensation <float> [-4000.0, 4000.0]

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

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

SERVo:1PPSoffset <int> ns

SERVo:QUIet <ON | OFF>

SERVo:TRACe <int > [0,255]

SERVo:FASTlock <int> [1,20]

SERVo:FALEngth <int> [100,20000]

SERVo?

SERVo:FASTlock

The FASTlock command enables the FASTLOCK mode, and sets its gain parameter. Fastlock works by momentarily multiplying the EFCScale gain to a value determined by this SERVo:FASTlock parameter. Gain values of 1x to 20x can be set, with a gain of 1x effectively disabling the FASTLOCK feature.

By selecting gain values of >1, the PLL loop parameter Proportional gain (SERV:EFCscale) will be increased after power on, thus increasing loop aggressiveness and improving lock PLL time. It is not desirable to maintain a high loop gain for longer than necessary to lock the PLL since high loop gains come at the expense of increased phase noise (reduced short term stability). The FASTLOCK mechanism will automatically reduce the FASTLOCK gain over a period of time specified by the SERVo:FALEngth command, during which time the FASTLOCK gain is slowly decreased from it's initial value to 1.0x.

Setting the FASTLOCK gain to 2 for example will result in the Proportional gain value stored in the SERVo:EFCscale parameter to be multiplied by 2x initially after power on.

This dynamic gain is slowly reduced until the gain is back to 1.0x, the value stored in the SERVo:EFCScale parameter.

For example:

if we set SERVo:FASTlock to 2, and SERVo:FALEngth to 3600, and SERVo:EFCScale is set to 0.7

Then initially the unit will multiply the EFCscale by 2x, and an effective EFCscale value of 1.4 is applied to the PLL loop.

This increased gain value difference will be reduced every second by 1/3600, so that the gain after two seconds would be: 1.3998, until after 3600 seconds the gain has been reduced back to it's long term value of 0.70 as stored in the the SERVo:EFCscale parameter.

Disabling the FASTLOCK mode is accomplished by setting the SERVo:FASTlock to 1. This will set the dynamic gain to 1.0, effectively disabling the fastlock feature.

This command has the following format:

SERVo:FASTlock <int> [1,20]

SERVo:FALEngth

This command adjusts the length of time during which the FASTLOCK feature is active, please see the command SERVo:FASTlock above.

It can be set from 100 seconds to 20.000 seconds. The Dynamic FASTLOCK gain is slowly reduced until it reaches a gain of 1.0 after the FALEngth period of seconds. During this time the PLL loop gain is increased by the amount specified in the SERVo:FASTlock parameter, which will result in a faster initial phase lock to UTC after power-on, while giving the lowest possible noise floor (best short term stability) during normal operation.

This command has the following format:

SERVo:FALEngth <int> [100,20000]

SERVo:COARSeDac

This command sets the coarse Dac that controls the EFC. The FireFly-IIA control loop automatically adjusts this setting. The user should not have to change this value.

This command has the following format:

SERVo:COARSeDac <int> [0,225]

SERVo:DACGain

This command is used for factory setup.

SERVo: EFCScale

Controls the Proportional part of the PID loop. Typical values are 0.7 (double oven OCXO) to 6.0 (simple single oven OCXO). 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]

SERVo:EFCDamping

Set's the Low Pass filter effectiveness of the DAC. Values from 2.0 to 50 are typically used. Larger values result in less noise at the expense of phase delay.This command has the following format:

SERVo:EFCDamping <float>[0.0 , 4000.0]

SERVo:SLOPe

The parameter determines the sign of the slope between the EFC and the frequency variation of the OCXO. This parameter should be set to match your OCXO's EFC frequency slope. This command has the following format:

SERVo:SLOPe <NEG | POS >

SERVo:TEMPCOmpensation

This parameter is a coefficient that reflects the correlation between the Current provided to the OCXO and the EFC. This coefficient is automatically computed and adjusted over time by the Jackson-Labs firmware. This command has the following format:

SERVo:TEMPCOmpensation <float> [-4000.0, 4000.0]

SERVo:AGINGcompensation

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

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

SERVo:PHASECOrrection

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

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

SERVo:1PPSoffset

This command sets the FireFly-IIA 1PPS signal's offset to UTC in 16.7ns steps.

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

This command has the following format:

SERVo:1PPSoffset <int> ns

SERVo:TRACe

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

This command has the following format:

SERVo:TRACe <int > [0,255]

An example output is described here:

08-07-31 373815 60685 -32.08 -2.22E-11 14 10 6 0x54

[date][1PPS Count][Fine DAC][UTC offset ns][Frequency Error Estimate][Sats Visible][Sats Tracked][Lock State][Health Status]

Please see the **SYNChronization?** command for detailed information on how to decode the health status indicator values. The Lock State variable indicates one of the following states:

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

SERVo?

This command returns the result of the following queries:

SERVo:COARSeDac?

SERVo:DACGain?

SERVo: EFCScale?

SERVo:EFCDamping?

SERVo:SLOPe?

SERVo:TEMPCOmpensation?

SERVo:AGINGcompensation?

SERVo:PHASECOrrection?

SERVo:1PPSoffset?

SERVo:TRACe?


JACKSON LABS
SINGLE OVEN
3920092083