

FireFly-IIA User Manual

FireFly-IIA User Manual

Copyright © 2010, 2011, 2012 Jackson Labs Technologies, Inc.

1	Introduction	1
1.1	Overview	1
1.2	General Safety Precautions	1
1.2.1	Grounding	1
1.2.2	Power Connections	1
1.2.3	Environmental Conditions	1
2	Quick-Start Instructions	3
2.1	Powering Up the Unit	3
2.1.1	Major connections	3
2.1.2	Harness Pinning	4
2.1.3	Harness Connector	5
2.1.4	Coaxial Connectors	6
2.2	Power	6
2.3	Connecting the GPS Antenna	6
2.4	Remote serial control	6
2.5	Loop parameter adjustment	7
3	SCPI-Control Quick Start Instructions	9
3.1	Introduction	9
3.2	General SCPI Commands	9
3.2.1	*IDN?	9
3.2.2	HELP?	9
3.3	GPS Subsystem	9
3.3.1	GPS:SATellite	10
3.3.2	GPS:SATellite:TRAcking:COUNt?	10
3.3.3	GPS:SATellite:VISible:COUNt?	10
3.3.4	NMEA Support	10
3.3.5	GPS:GPGGA	11
3.3.6	GPS:GGASTat	11
3.3.7	GPS:GPRMC	12
3.3.8	GPS:XYZSPeed	12
3.3.9	GPS:GPZDA	12
3.3.10	GPS:PASHR	12
3.3.11	GPS:GYRO	13
3.3.12	GPS:GYRO:CAL	14
3.3.13	GPS:DYNAMic?	15
3.3.14	GPS:DYNAMic 8 (Automatic Dynamic Mode)	16
3.3.15	GPS:REFerence:ADELay <float> <s ns > [-32767ns,32767ns]	17
3.3.16	GPS:REFerence:PULse:SAWtooth?	17
3.3.17	GPS:RESET ONCE	17
3.3.18	GPS:INITial:DATE <yyyy,mm,dd>	17
3.3.19	GPS:INITial:TIME <hour,min,sec>	17
3.3.20	GPS:JAMlevel?	18
3.3.21	GPS?	18
3.4	GYRO SUBSYSTEM	18
3.4.1	GYRO:MODE <ON OFF>	18
3.4.2	GYRO:TRACE <int> [0,255]	18
3.4.3	GYRO:CAL <float,float,float,float,float,float>	18
3.4.4	GYRO:CAL:COMPUTE	19

3.4.5	GYRO:SENS, GYRO:EFC, and GPS:CAL:RESET	19
3.4.6	GYRO:GLOAD?	19
3.5	PTIME Subsystem	19
3.5.1	PTIME:TZONE?	19
3.5.2	PTIME:DATE?	19
3.5.3	PTIME:TIME?	20
3.5.4	PTIME:TIME:STRing?	20
3.5.5	PTIME:TINterval?	20
3.5.6	PTIME?	20
3.6	SYNChronization Subsystem	20
3.6.1	SYNChronization:HOLDoVer:DURation?	20
3.6.2	SYNChronization:HOLDoVer:INITiate	21
3.6.3	SYNChronization:HOLDoVer:RECOvery:INITiate	21
3.6.4	SYNChronization:SOURce:MODE	21
3.6.5	SYNChronization:SOURce:STATE?	21
3.6.6	SYNChronization:TINterval?	21
3.6.7	SYNChronization:IMMEdiate.	22
3.6.8	SYNChronization:FEEStimate?	22
3.6.9	SYNChronization:LOCKed?	22
3.6.10	SYNChronization:OUTput:1PPS:RESET [ON OFF]	22
3.6.11	SYNChronization:health?	22
3.6.12	SYNChronization?	23
3.7	DIAGnostic Subsystem	23
3.7.1	DIAGnostic:ROSCillator:EFControl:RELative?	23
3.7.2	DIAGnostic:ROSCillator:EFControl:ABSolute?	24
3.8	MEASURE Subsystem	24
3.8.1	MEASure:VOLTagE?.	24
3.8.2	MEASure:CURRent?.	24
3.8.3	MEASure?	24
3.9	SYSTEM Subsystem	24
3.9.1	SYSTem:COMMunicate	24
3.9.1.1	SYSTem:COMMunicate:SERial:ECHO	24
3.9.1.2	SYSTem:COMMunicate:SERial:PRoMpt	25
3.9.1.3	SYSTem:COMMunicate:SERial:BAUD	25
3.9.2	SYSTem:STATus?	25
3.9.3	SYSTem:FACToryReset ONCE	25
3.10	SERVO Subsystem	25
3.10.1	SERVo:FASTlock	26
3.10.2	SERVo:FALEngth	26
3.10.3	SERVo:COARSeDac	27
3.10.4	SERVo:DACGain	27
3.10.5	SERVo: EFCScale	27
3.10.6	SERVo:EFCDamping	27
3.10.7	SERVo:SLOPe	27
3.10.8	SERVo:TEMPCompensation	27
3.10.9	SERVo:AGINGcompensation	28
3.10.10	SERVo:PHASECOrrrection	28
3.10.11	SERVo:1PPSoffset	28
3.10.12	SERVo:TRACe	28
3.10.13	SERVo?	29

4	Firmware Upgrade Instructions	.31
4.1	Introduction	.31
4.2	ISP Flash Loader Utility installation	.31
4.2.1	Philips LPC2000 Flash Utility	.31
4.2.2	Flash Magic Flash Programming Utility	.31
4.3	Putting the PCB into In-Circuit Programming (ISP) mode	.32
4.4	Downloading the firmware	.32
4.4.1	Philips LPC2000 Flash Utility	.32
4.4.2	Flash Magic Flash Programming Utility	.34
4.5	Verifying Firmware Update	.37
5	GPSSCon Utility	.39
5.1	Description	.39
5.2	Installation	.39
5.3	Using GPSSCon	.39
5.3.1	Setting the options	.39
5.3.1.1	Communication Parameters	.40
5.3.1.2	Auxiliary parameters	.40
5.3.1.3	Other options	.42
5.3.2	Sending manual commands to the receiver	.42
5.3.3	Use of the mouse in graph mode	.43
5.3.4	Exporting the graphics	.45
5.4	Interpreting the Data	.46
6	Certification and Warranty	.47
6.1	Certification	.47
6.1.1	Warranty	.47
6.1.2	Limitation of Warranty	.47
6.1.3	Exclusive Remedies	.48

1 Introduction

1.1 Overview

The FireFly-IIA 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, a low-noise 10MHz distribution amplifier with three highly Isolated 10MHz Sine Wave outputs, two 10MHz LVDS outputs, 1PPS UTC synchronized LVDS and RS-232 outputs, RS-232 control interface, and precision voltage references and DACs.

1.2 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.

1.2.1 Grounding

To avoid damaging the sensitive electronic components in the FireFly-IIA GPSDO 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 GPS antenna.

1.2.2 Power Connections

Make sure to connect the DC power to the device following the polarity indicated in Section 2.1 . Do not reverse the power pins as this will cause serious damage to the circuit board.

1.2.3 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 ac mains voltage requirements and ambient operating temperature range.

2 Quick-Start Instructions

2.1 Powering Up the Unit

To operate the unit, simply do the following two steps:

- 1) Connect a 5V-compatible GPS antenna to connector J2.
- 2) Connect +12V DC Power to pins 15, and 16 of power connector U3, ground to pins 12 and 14 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 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.

2.1.1 Major connections

The major connections and features of the FireFly-IIA PCB are shown in [Figure 2.1](#).

Figure 2.1 FireFly-IIA major connections

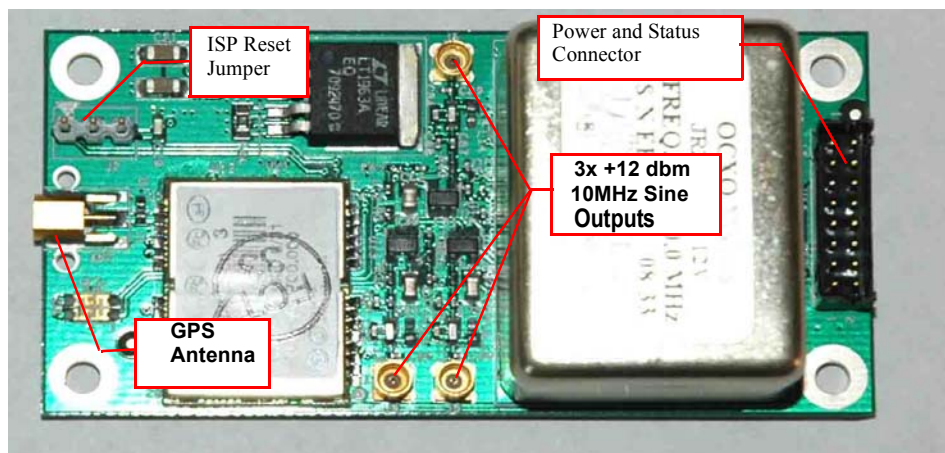
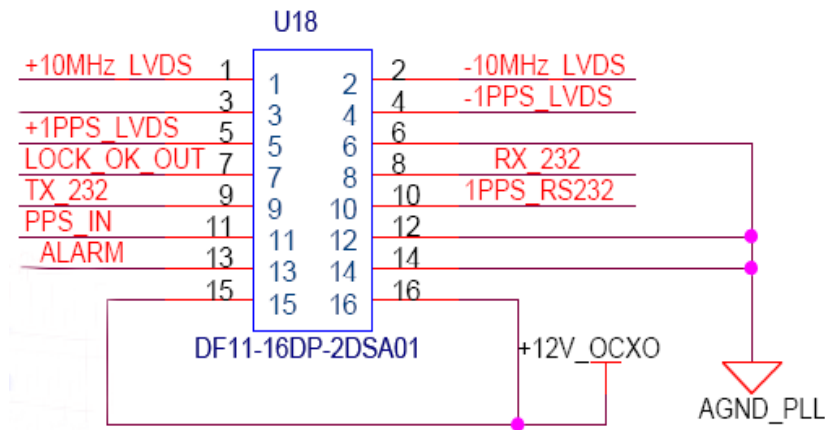


Table 2.1 shows the FireFly-IIA revision 1.0 hardware connectors

Table 2.1 FireFly-IIA hardware connectors

Ref	Name	Function	Specification	Pinning
J1	Antenna	GPS Antenna	5V Amplified Antenna MMCX connector	Center-RF Input, Shield-GND
J2	ISP/RESET	Reset and ISP	Pull pins 1 and 3 to GND to activate function	3-RESET-IN#, 2-GND, 1-ENTER_ISP#
U18A	10MHz LVDS	10MHz Output A	LVDS, +/-300mV differential	U18.1-10MHz_Pos, U18.2-10MHz_Neg
U18B	Red LED (GPS 1PPS)	Blinks if GPS is generating a 1PPS output, follows Red LED	3.3Vpp CMOS	U18.13-LED+, U18.14-GND
U18C	1PPS LVDS	1PPS Output A	LVDS, +/-300mV differential	U18.5-1PPS_Pos, U18.4-1PPS_Neg
U18D	1PPS RS-232	1PPS Output B	RS-232 level	U18.10-1PPS-RS232, U18.12-GND
U18E	1PPS IN	External 1PPS input (optional)	3.3V CMOS, risetime <10ns	U18.11-1PPS-In, U18.12-GND
U18F	LOCK_OK_OUT	Follows status of Green LED	3.3V CMOS	U18.7-LED+, U18.6-GND
U18G	RS-232 SCPI	RS-232 Serial Port	RS-232, 115.2,8N1	U18.9-TX, U18.8-RX, U18.6-GND
U18H	+12V Power	Main Power Input	+11.0V - +14.0V	U18.16-+12V, U18.15-+12V, U18.14-GND, U18.12-GND

Figure 2.2 Typical Connections to Power Connector U18



2.1.2 Harness Pinning

Table 2.2 shows the pinout of the cable harness U18.

Table 2.2 Cable harness U18 pinout

PIN	NAME
1	+10MHz LVDS
2	-10MHz LVDS
3	GND
4	-1PPS LVDS
5	+1PPS LVDS
6	GND
7	LOCK_OK_OUT
8	RX RS-232
9	TX RS-232
10	1PPS Out RS-232 Level
11	1PPS In CMOS level
12	GND
13	Red LED Signal
14	GND
15	+12V
16	+12V

2.1.3 Harness Connector

The manufacturer for connector U18 is Hirose. A mating housing part number for this connector is available from Digikey, and crimp pins are also available from Digikey for different wire sizes:

<http://search.digikey.com/scripts/DkSearch/dksus.dll?Detail&name=H2025-ND>

The part number of the connector soldered onto the FireFly-IIA PCB is:

Hirose DF11-16DP-2DSA01

NOTES:

The LVDS signals (+/-10MHz, +/-1PPS) should be routed using 100Ohm differential wiring (twisted pair), or two 500Ohm 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 LOCK_OK_OUT and Red LED signals are 3.3V CMOS signals, and thus require a series resistor of typically 390 to 470 Ohms when used to drive LED's.

The 10MHz 50 Ohm Sine Wave outputs (+12dBm) do not require a termination if they remain unused.

2.1.4 Coaxial Connectors

All four coaxial connectors on the FireFly-IIA board are generic MMCX types.

2.2 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 0.28A at 12V. Connect a clean +12V power supply to pins 15 and 16 of the cable harness U18.

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

2.3 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 FireFly-IIA 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 FireFly-IIA unit supports 5V antennae, whereas the FireFly-II units support 3.3V antennae.

FireFly-IIA is capable of generating standard navigation messages (see GPS:GPGGA 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 FireFly-IIA 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).

2.4 Remote serial control

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

Attach the FireFly-IIA unit to your PC's Hyperterminal, the optional GPSSCon software package (see Chapter 5 of this manual), or a third-party freeware Windows-based application program called Z38xx.

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

<http://www.jackson-labs.com/assets/uploads/main/Z38XX.zip>

2.5 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 unit's 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:

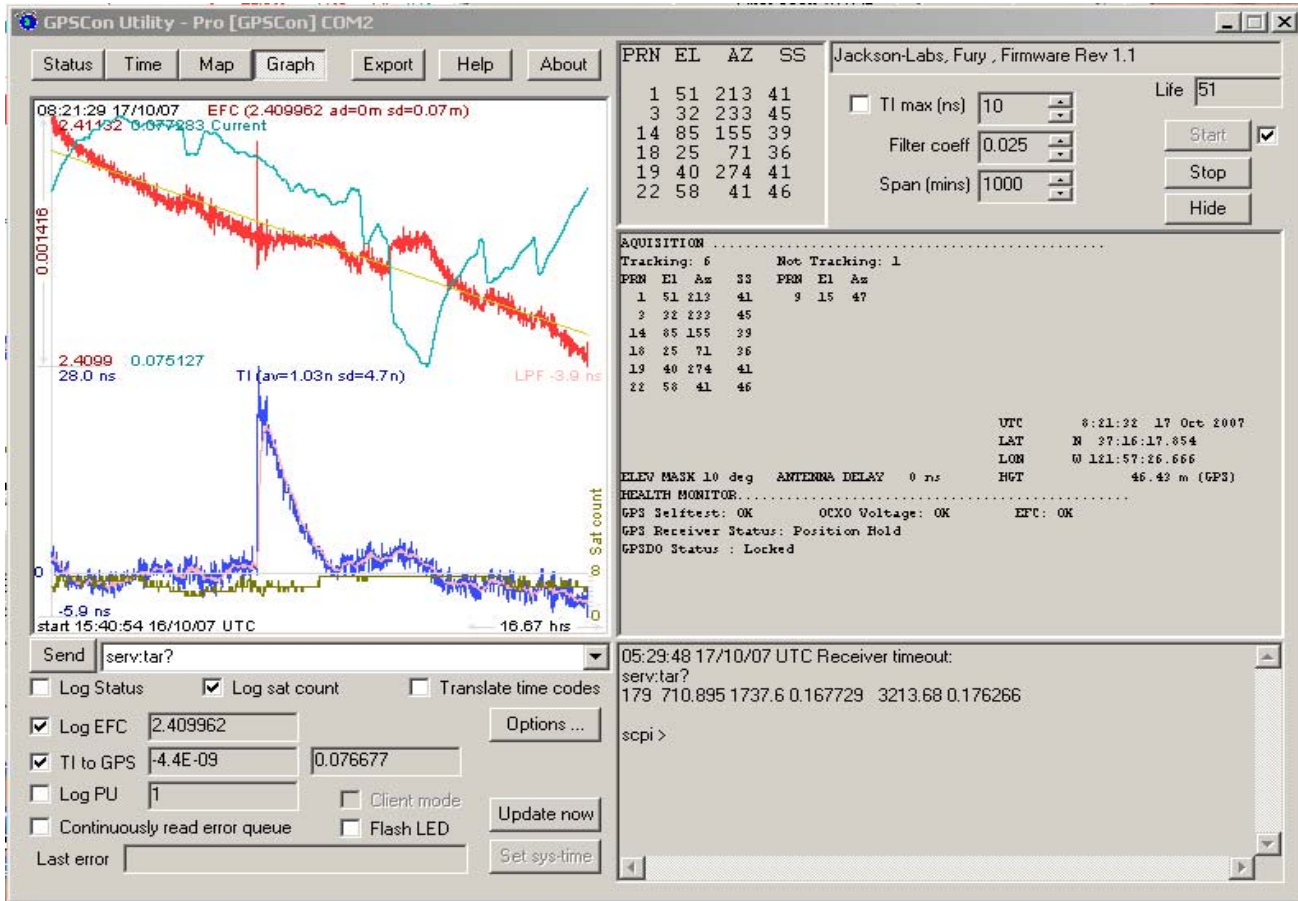
EFC Scale: this is the proportional gain of the PID loop. Higher values will give quicker convergence, and faster locking of the GPS time (lower loop time constant), lower values give less noise. Values between 0.7 (good double oven OCXO) and 6.0 (simple single-oven OCXO) are typical.

EFC Damping: overall IIR filter time constant. higher values increase loop time constant. Jackson Labs Technologies, Inc. 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 FireFly-IIA 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 plot shown in [Figure 2.3](#) when experiencing small perturbations:

Figure 2.3 FireFly-IIA phase compensation plot



3 SCPI-Control Quick Start Instructions

3.1 Introduction

The SCPI (Standard Commands for Programmable Instrumentation) 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. To get a listing of the available commands, send the HELP? query. This will return a list of all the available commands for the FireFly-IIA GPSDO.

Additional information regarding the SCPI protocol syntax can be found on the following web site:
<http://www.scpiconsortium.org>

Please refer to the document SCPI-99.pdf for details regarding individual SCPI command definitions. A basic familiarity with the SCPI protocol is recommended when reading this chapter.

3.2 General SCPI Commands

3.2.1 *IDN?

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

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

3.2.2 HELP?

This query returns a list of the commands available for the FireFly-IIA GPSDO.

3.3 GPS Subsystem

Note: 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:REference:ADELay <float> <s | ns > [-32767ns,32767ns]
GPS:REference:PULse:SAWtooth?
GPS:RESET ONCE
GPS:INITial:DATE <yyyy,mm,dd>
GPS:INITial:TIME <hour,min,sec>
GPS:JAMlevel?
GPS?

```

3.3.1 GPS:SATellite

This group of commands describe the satellite constellation.

3.3.2 GPS:SATellite:TRACKing:COUNT?

This query returns the number of satellites being tracked.

3.3.3 GPS:SATellite:VISible:COUNT?

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

3.3.4 NMEA Support

The following four commands allow the FireFly-IIA GPSDO to be used as an industry standard navigation GPS receiver. The GPGGA, GPRM, PASHR and GPZDA NMEA commands comprise all

necessary information about the antenna position, height, velocity, direction, satellite info, fix info, time, date and other information that can be used by standard navigation applications 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 incoming 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.

3.3.5 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 warm-up 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.

3.3.6 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 warm-up 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 unit's 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.

3.3.7 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 warm-up phase.

This command has the following format:

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

3.3.8 GPS:XYZSPeed

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

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

Additionally, the number of accrued 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]
```

3.3.9 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 warm-up phase.

This command has the following format:

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

3.3.10 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,eeee.eeeee,f,ggggg.gg,hhh,iii.ii,jjj.jj,kkkk.kk,ll.l,mm.m,nn.n,00.0,p.pp,*[checksum]

Where:

aa: Number of Sats
 bbbbb.00: Time of Day UTC
 cccc.ccccc,d: Latitude,S/N
 eeee.eeeee,f: Longitude,W/E
 ggggg.gg: Antenna Height in meters
 hhhh: Four fixed '?' symbols
 iii.ii: Course Over Ground
 jjj.jj: Speed in Knots
 kkkk.k: Vertical Velocity in meters/s
 ll.l: PDOP
 mm.mHDOP
 nn.nVDOP
 00.0 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 warm-up phase.

This command has the following format:

GPS:PASHR <int> [0,255]

3.3.11 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 its 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 $\sim \pm 5$ 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 $\pm 3g$ 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.

3.3.12 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).

3.3.13 GPS:DYNAMiC?

This command allows the user to select the dynamic motion model being applied to the Kalman 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
```

3.3.14 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.

3.3.15 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 unit's 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.

3.3.16 GPS:REFerence:PULse:SAWtooth?

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

3.3.17 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 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 GPS reset, as indicated by the red blinking LED.

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

Firmware revision 2.32 and later adds support for manually setting the time and date in the RTC in the event that no GPS signals are available. This command allows setting the internal RTC DATE manually when operating the unit in GPS denied environments. This command is compatible to the PTIME:OUT ON command described in section 3.5.5. to allow automatic time and date synchronization of two units to each other. The internal RTC is driven by the highly stable CSAC 10MHz signal, and thus has very high accuracy.

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

Firmware revision 2.32 and later adds support for manually setting the time and date in the RTC in the event that no GPS signals are available. This command allows setting the internal RTC TIME manually when operating the unit in GPS denied environments. This command is compatible to the PTIME:OUT ON command described in section 3.5.5. to allow automatic time and date synchronization of two units to each other. The internal RTC is driven by the highly stable CSAC 10MHz signal, and thus has very high accuracy.

3.3.20 GPS:JAMlevel?

Firmware revision 2.32 and later adds support for a GPS signal jamming-indicator. The GPS receiver will detect, and flag jamming interference with levels ranging from 0 (no jamming) to 255 (strong jamming). Any level exceeding 50 will cause a SYNC:HEALTH 0x800 event to be flagged, and the unit to disable the LOCK_OUT_OK signal on connector J1.

3.3.21 GPS?

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

3.4 GYRO SUBSYSTEM

The following Gyro commands are supported:

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

3.4.1 GYRO:MODE <ON | OFF>

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

3.4.2 GYRO:TRACE <int> [0,255]

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

3.4.3 GYRO:CAL <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.

3.4.4 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 earth's 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.

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

Used for factory calibration of Gyro subsystem.

3.4.6 GYRO:GLOAD?

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

3.5 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?

3.5.1 PTIME:TZONE?

Returns the local time zone offset.

3.5.2 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.

3.5.3 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.

3.5.4 PTIME:TIME:STRing?

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

3.5.5 PTIME:TINTerval?

This query is equivalent to the command SYNChronization:TINTerval

3.5.6 PTIME?

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

PTIME:DATE?

PTIME:TIME?

PTIME:TZONE?

PTIME:TINTerval?

3.6 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?

3.6.1 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.

3.6.2 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.

3.6.3 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.

3.6.4 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]
```

3.6.5 SYNChronization:SOURce:STATE?

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

3.6.6 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.

3.6.7 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.

3.6.8 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.

3.6.9 SYNChronization:LOCKed?

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

3.6.10 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.

3.6.11 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 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 OCXO coarse-DAC is maxed-out at 255	HEALTH STATUS = 0x1;
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;
If the GPS receiver indicates a strong jamming signal of >=50 (range is 0 to 255)	HEALTH STATUS = 0x800;

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.

3.6.12 SYNChronization?

This query returns the results of these four queries:

SYNChronization:SOURce:MODE?

SYNChronization:SOURce:STATE?

SYNChronization:LOCKed?

SYNChronization:HOLDover:DURation?

SYNChronization:health?

3.7 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?

3.7.1 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%.

3.7.2 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

3.8 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?

3.8.1 MEASure:VOLTage?

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

3.8.2 MEASure:CURRent?

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

3.8.3 MEASure?

This query returns the result of the two following queries:

MEASure:VOLTage?

MEASure:CURRent?

3.9 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

3.9.1 SYSTem:COMMunicate

3.9.1.1 SYSTem:COMMunicate:SERial:ECHO

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

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

3.9.1.2 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>

3.9.1.3 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>

3.9.2 SYSTem:STATus?

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

3.9.3 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.

3.10 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:TEMPCCompensation <float> [-4000.0, 4000.0]
 SERVo:AGINGcompensation <float> [-10.0, 10.0]
 SERVo:PHASECOrrrection <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?

3.10.1 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 its 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 its long term value of 0.70 as stored in 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:

```
SERV:FASTlock <int> [1,20]
```

3.10.2 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:FALEnith <int> [100,20000]

3.10.3 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]

3.10.4 SERVo:DACGain

This command is used for factory setup.

3.10.5 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]

3.10.6 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]

3.10.7 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 >

3.10.8 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 Technologies, Inc. firmware. This command has the following format:

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

3.10.9 SERV0: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 Technologies, Inc. firmware. This command has the following format:

```
SERV0:AGINGcompensation <float> [-10.0, 10.0]
```

3.10.10 SERV0:PHASECOrrrection

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:

```
SERV0:PHASECOrrrection <float> [-100.0, 100.0]
```

3.10.11 SERV0:1PPSoffset

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:

```
SERV0:1PPSoffset <int> ns
```

3.10.12 SERV0:TRACe

This command sets the period in seconds for the debug trace. 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:

```
SERV0: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

3.10.13 SERV0?

This command returns the result of the following queries:

- SERV0:COARSeDac?
- SERV0:DACGain?
- SERV0: EFCScale?
- SERV0:EFCDamping?
- SERV0:SLOPe?
- SERV0:TEMPCompensation?
- SERV0:AGINGcompensation?
- SERV0:PHASECOrrrection?
- SERV0:1PPSoffset?
- SERV0:TRACe?



4 Firmware Upgrade Instructions

4.1 Introduction

The following is a short tutorial on how to upgrade the FireFly-IIA GPSDO firmware. Please follow the instructions in-order to prevent corrupting the FireFly-IIA Flash, which may require reflashing at the factory.

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

4.2 ISP Flash Loader Utility installation

There are two Flash loader utilities available to upgrade the FireFly-IIA firmware. You can download the Philips LPC2000 utility from the Jackson Labs Technologies, Inc. website under the Support tab:

<http://www.jackson-labs.com/support.html>

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

<http://www.flashmagictool.com/>

4.2.1 Philips LPC2000 Flash Utility

The first is the Philips LPC2000 utility version 2.2.3. Please note that some computers are known to be incompatible with the LPC2000 flash utility. Preliminary investigations show Windows Media Center and/or Centrino vPro processor systems to create download difficulties. Please use a different computer if you experience problems such as the download breaking up in the middle of the transfer. Or, alternatively, you may use the Flash Magic programming tool.

Please ensure that you have at least version 2.2.3 of the LPC2100 flash utility installed. Earlier versions may not recognize the LPC2138 processor used on the FireFly-IIA boards.

4.2.2 Flash Magic Flash Programming Utility

The second utility is the Flash Magic tool available on the Flash Magic website:

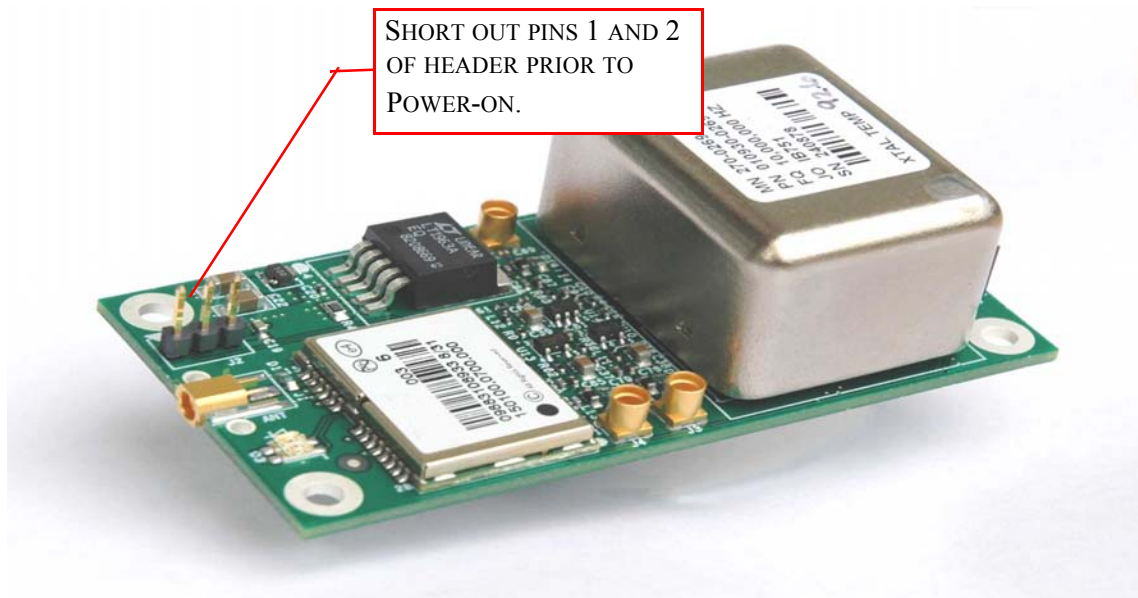
<http://www.flashmagictool.com/>

If the Philips LPC2000 tool doesn't work, please use this one.

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

Momentarily short-out pins 1 and 2 of header J2 using a jumper or other conductive material during power-on. Both LED's should remain off, indicating the unit is properly placed into ISP mode. If the LED's light up after power-on, the unit is not in ISP mode.

Figure 4.1 Location of header J2



4.4 Downloading the firmware

Download the latest version of FireFly-IIA firmware from the Jackson Labs Technologies, Inc. support website and store it in a place that will be remembered. The file is in .hex format.

The unit needs to be connected to the computer's RS-232 serial port prior to firmware download. Connect a DB-9 serial connector to the FireFly-IIA as indicated in Section 2.1.1.

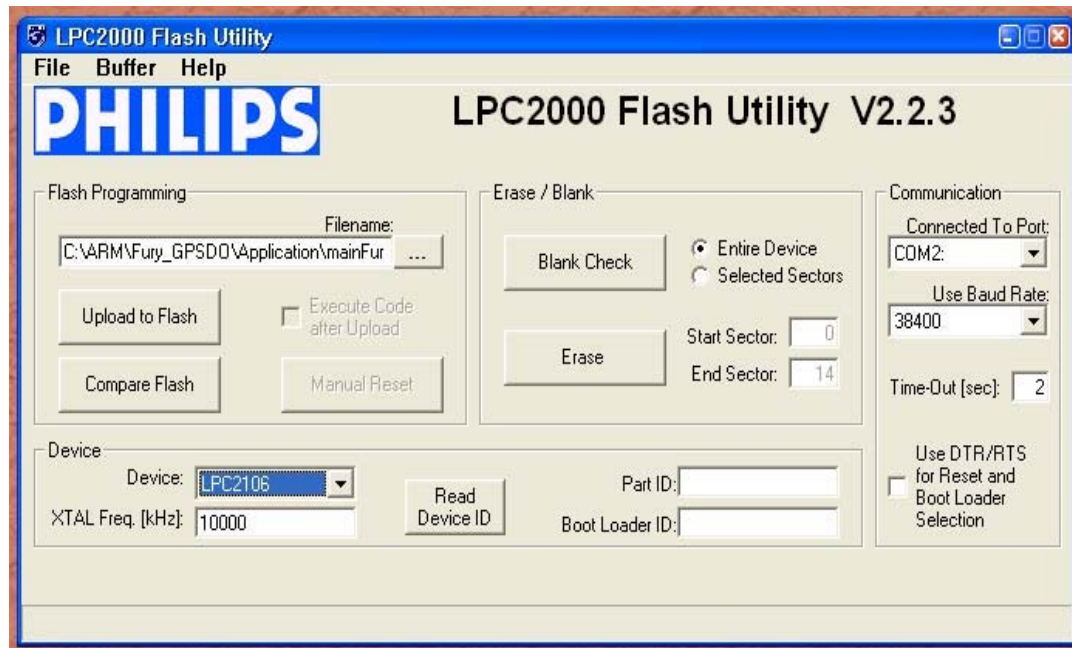
4.4.1 Philips LPC2000 Flash Utility

- A) Open the LPC2000 utility. Set the COM port in the LPC2000 application as needed on your PC.
- B) Select the Baud Rate of the LPC2000 utility to be 38400 or slower. Faster Baud rates will not work properly.

C) Press the “READ DEVICE ID” button, this should then show “LPC2138” in the DEVICE window if the unit is communicating correctly to the application.

Warning: Make sure NOT(!) to press the “erase” button under any circumstances, this may erase factory calibration data, and the unit will not operate and will have to be returned to the factory. Pressing the “erase” button on the ISP utility will thus void the warranty.

Figure 4.2 LPC2000 flash utility



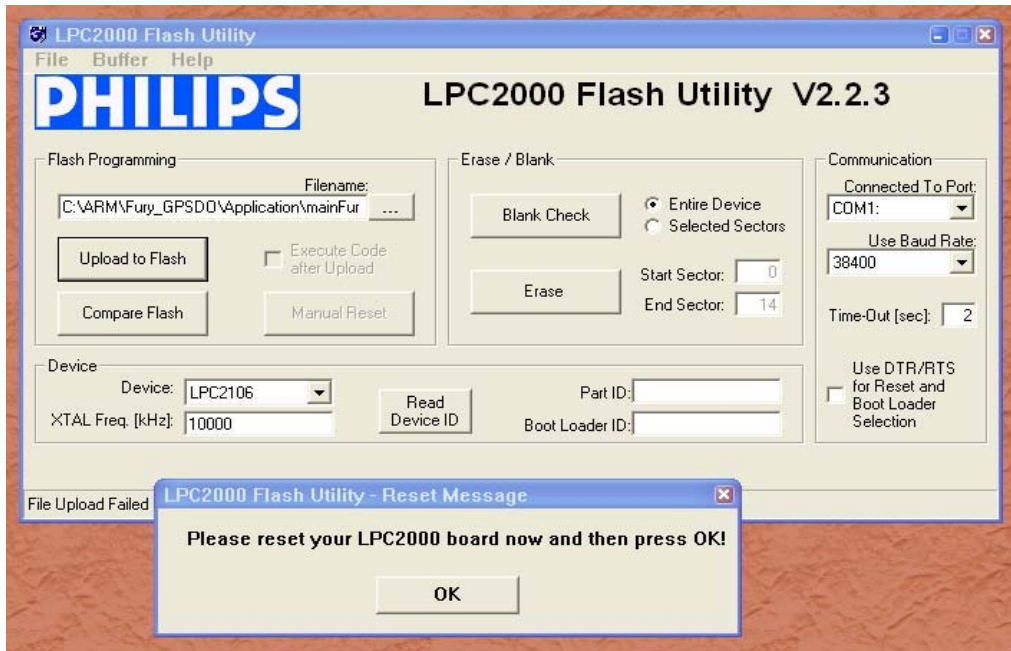
Please note that the “Use Baud Rate” setting needs to be set to 38400 Baud or less, it will not work faster than 38400 Baud.

The “DEVICE” should show up as “LPC2138” after pressing “READ DEVICE ID”

Also, please point the “Filename” to the directory where you have stored the latest firmware hex file that is to be downloaded.

D) Start the download by pressing “Upload to Flash” button. The following window should appear if the correct COM port has been chosen etc:

Figure 4.3 LPC2000 flash download



Press the “OK” button, and the download should start. Sometimes the utility gets confused and this process (from item 4.4 B) has to be tried several times.

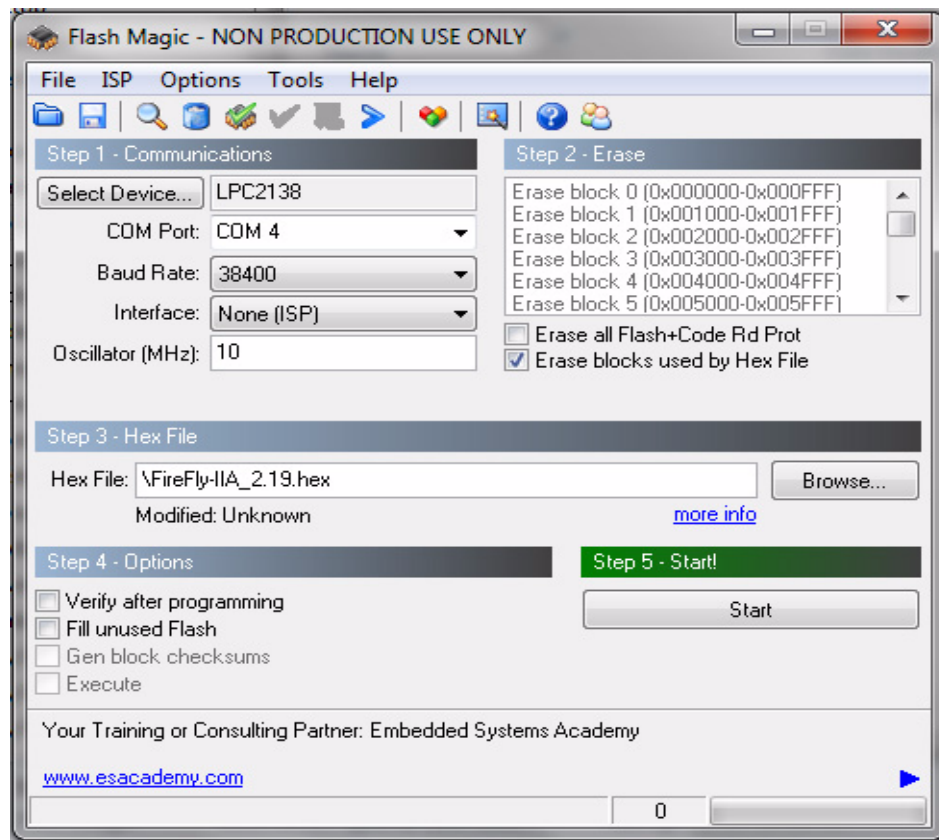
It is not necessary to press “reset” as the utility is asking. Just press “OK” on the utility window’s button.

Warning: DO NOT PRESS THE “ERASE” BUTTON AT ANY TIME! THIS WILL RENDER THE PCB USELESS AND CAN ONLY BE RECOVERED AT THE FACTORY!

4.4.2 Flash Magic Flash Programming Utility

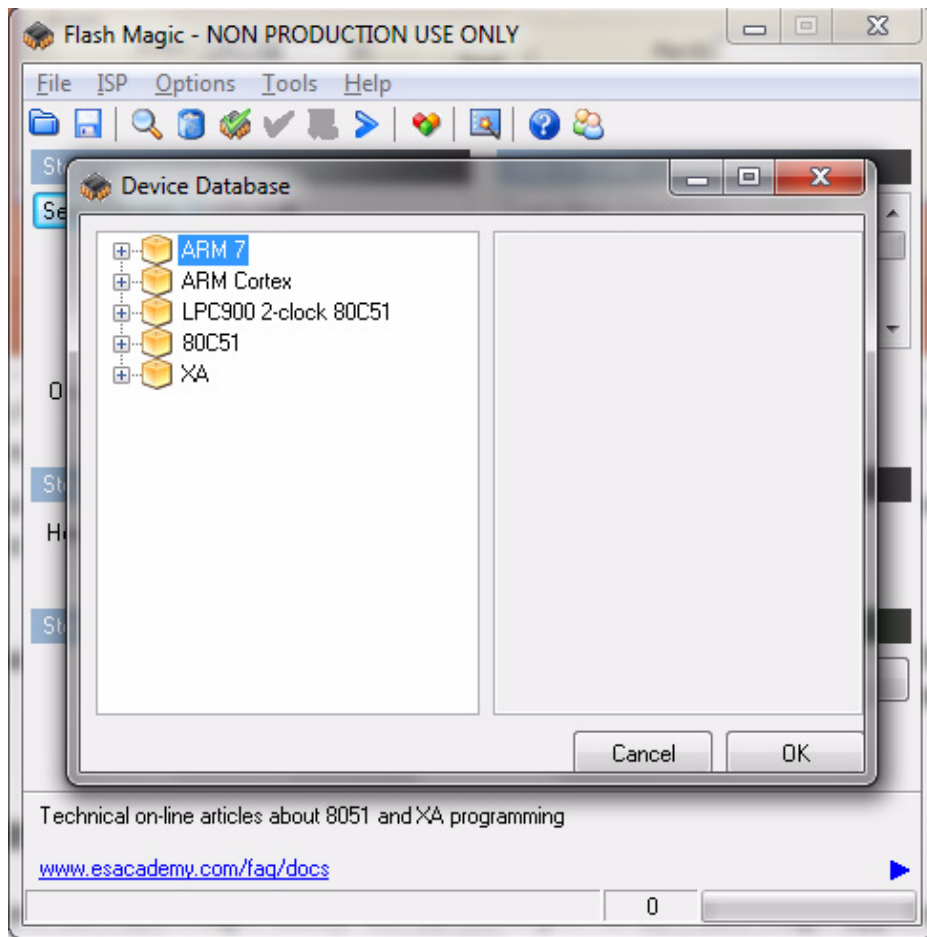
A) Open the Flash Magic utility. Set the COM port in the Flash Magic application as needed on your PC. Set “Interface” to “None (ISP)”.

Figure 4.4 Flash Magic utility

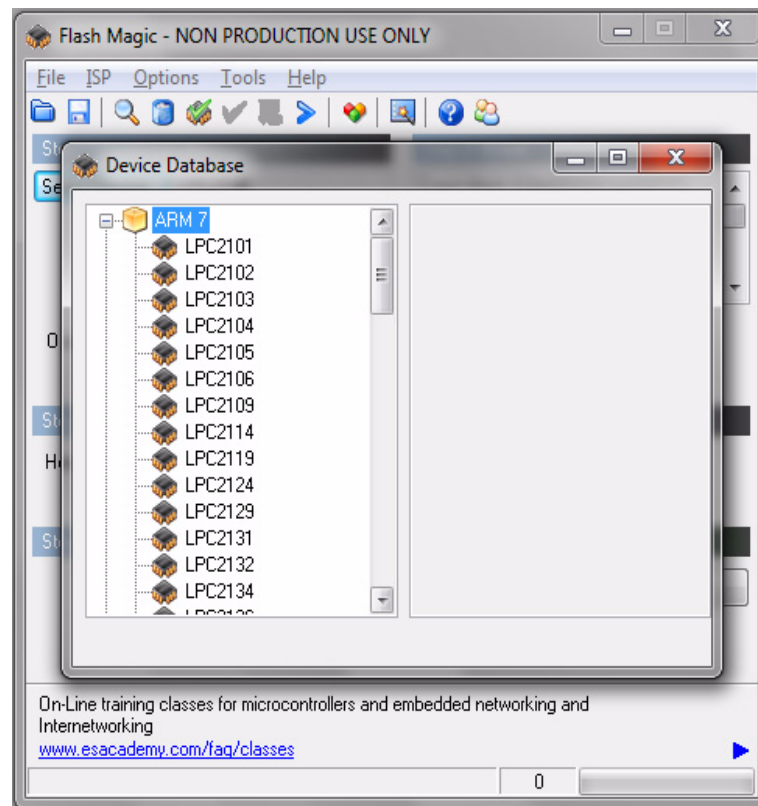


B) Press the “Select Device” button and the window shown in [Figure 4.5](#) will appear:

Figure 4.5 Device selection window



C) Expand the ARM7 folder and select the appropriate processor, in this case the LPC2138.

Figure 4.6 Expanded device selection window

D) Select the Baud Rate of the Flash Magic utility to be 38400 or slower. Faster Baud rates will not work properly.

E) Set the Oscillator (MHz) to “10”.

F) Check the box marked “Erase blocks used by Hex File”.

Warning: Make sure NOT(!) to check the box marked “Erase all Flash+Code Rd Prot” under any circumstances, this may erase factory calibration data, and the unit will not operate and will have to be returned to the factory. Checking this box on the ISP utility will thus void the warranty.

G) Under “Step 3 - Hex File” browse for the hex file that you downloaded in step 4.4 .

H) Go to Step 5 and press “Start”. You will observe the firmware being downloaded to the FireFly-IIA.

4.5 Verifying Firmware Update

Remove the jumper from header J2, and power cycle the unit. Both LED’s should blink.

During power on, the unit sends an ID string out of the serial port at 115200 Baud by default. The firmware version can also be queried by sending the *IDN? command. Verify that the firmware version is the version that was downloaded.

5 GPSCon Utility

5.1 Description

GPSCon is a program for the monitoring and control of a variety of GPS time and frequency standard receivers. It communicates with the receiver using the SCPI command set. This utility can be obtained directly from Real Ham Radio.com at the following URL:

<http://www.realhamradio.com/gpscon-buy-now.htm>

Important note: On newer, faster computers running Windows 7, GPSCon may not acquire data correctly. If you encounter this problem, it is recommended that you install GPSCon on a slower computer using Windows XP.

5.2 Installation

Follow the directions that come with GPSCon for installing the utility on your computer.

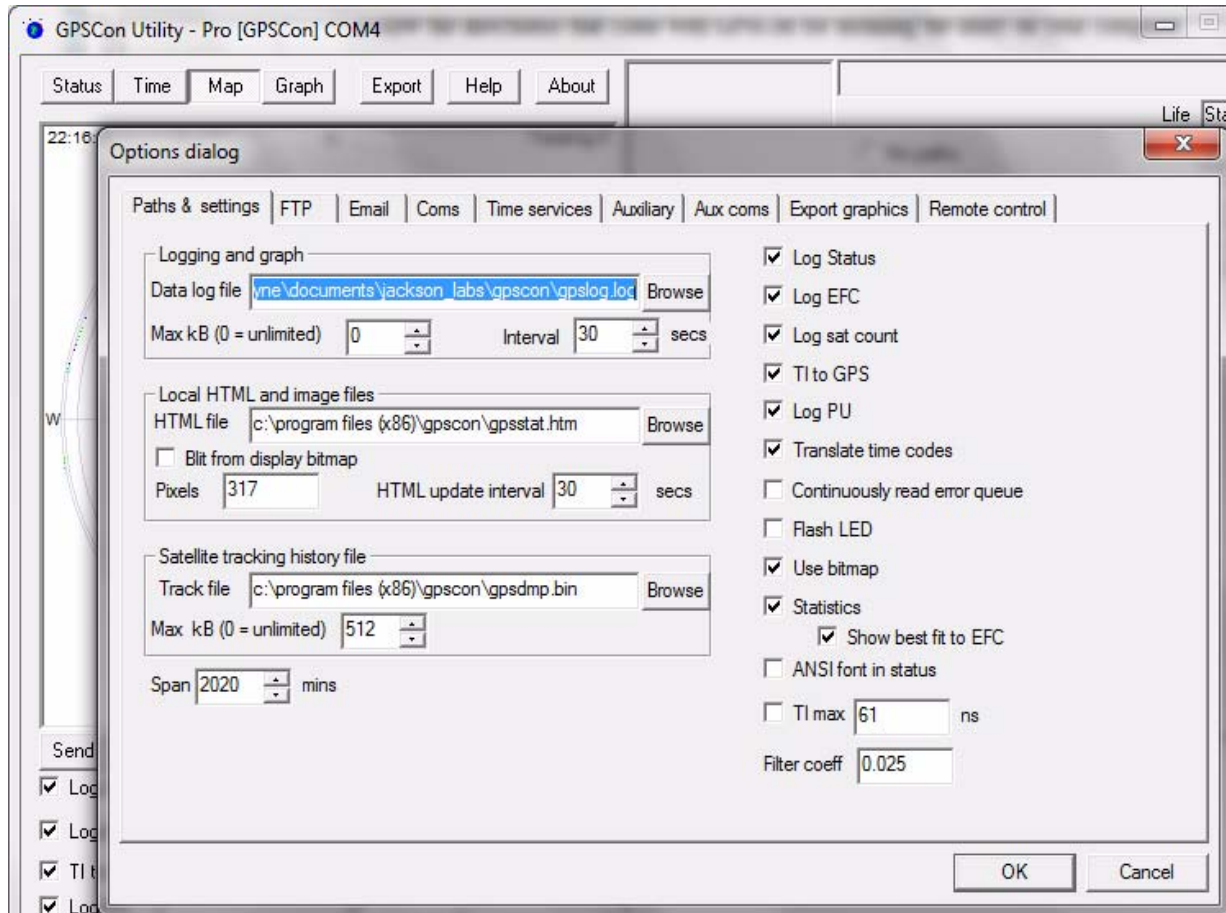
5.3 Using GPSCon

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

5.3.1 Setting the options

To set up the options for your GPSCon session, press the “Options” button below the display area. The window shown in [Figure 5.1](#) will appear. You can select from the tabs which options you wish to set.

Figure 5.1 Options window



5.3.1.1 Communication Parameters

Before you can use GPSCon you must set the communication parameters for your system. Open the dialog box by pressing the “Options” button. Then select the “Coms” tab. You will see the window shown in [Figure 5.2](#). Select the correct COM port for your computer and set the baud rate to 115200, parity to None, Data Bits to 8 and Stop Bits to 1. Set Flow Control to “None”. Once you have configured the communication parameters, press the “OK” button to close the window.

5.3.1.2 Auxiliary parameters

After pressing the “Options” button, you can select “Auxiliary” and set other options or measurements. See [Figure 5.3](#) for an example of an auxiliary measurement. You will notice that the “Aux 1 request string” has been set to `meas : current ? <CD>` and the “Log Aux1” box is checked. In the area below labeled “Traces to be visible on the graph”, the box “Aux 1” is checked and the label “OCXO curr” has been added.

Figure 5.2 Setting the communications parameters

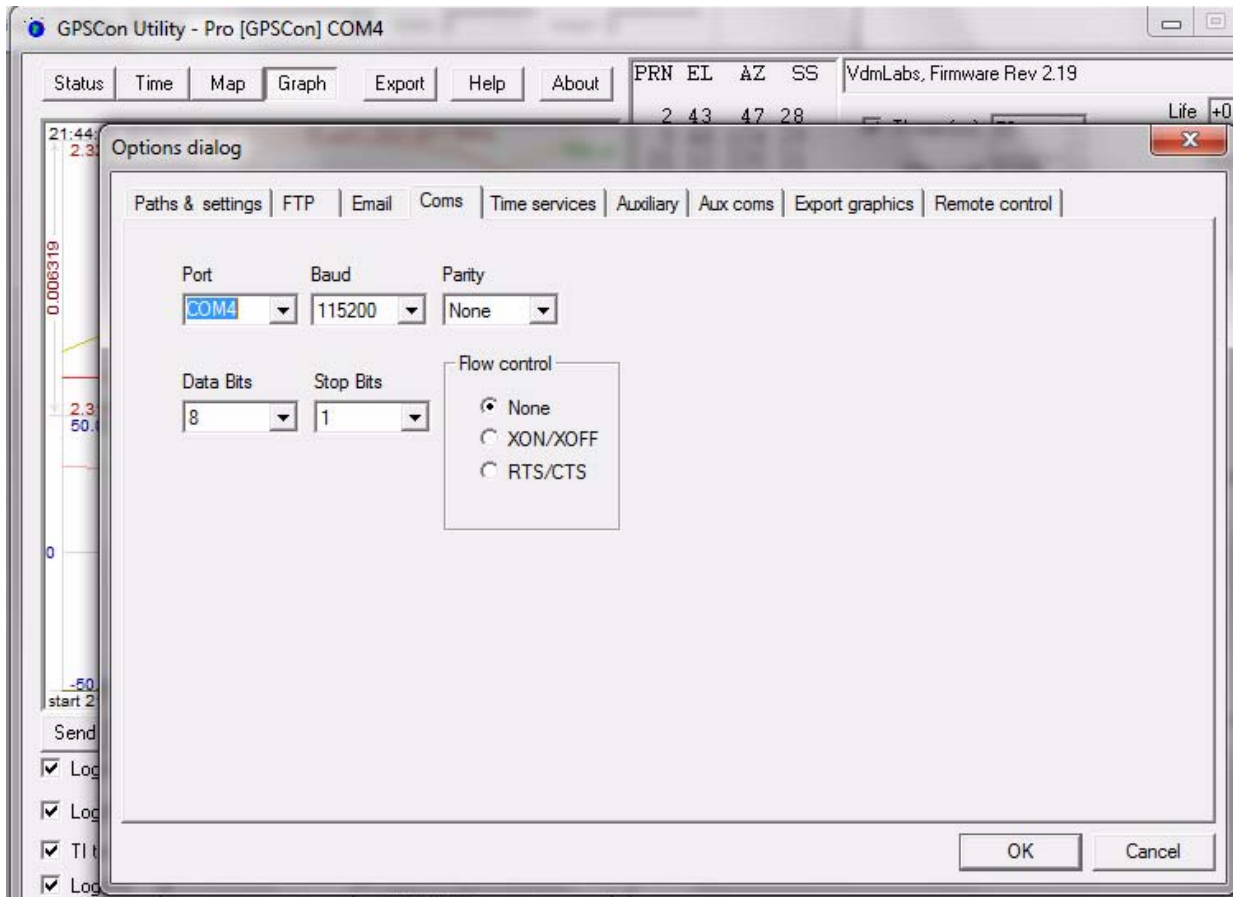
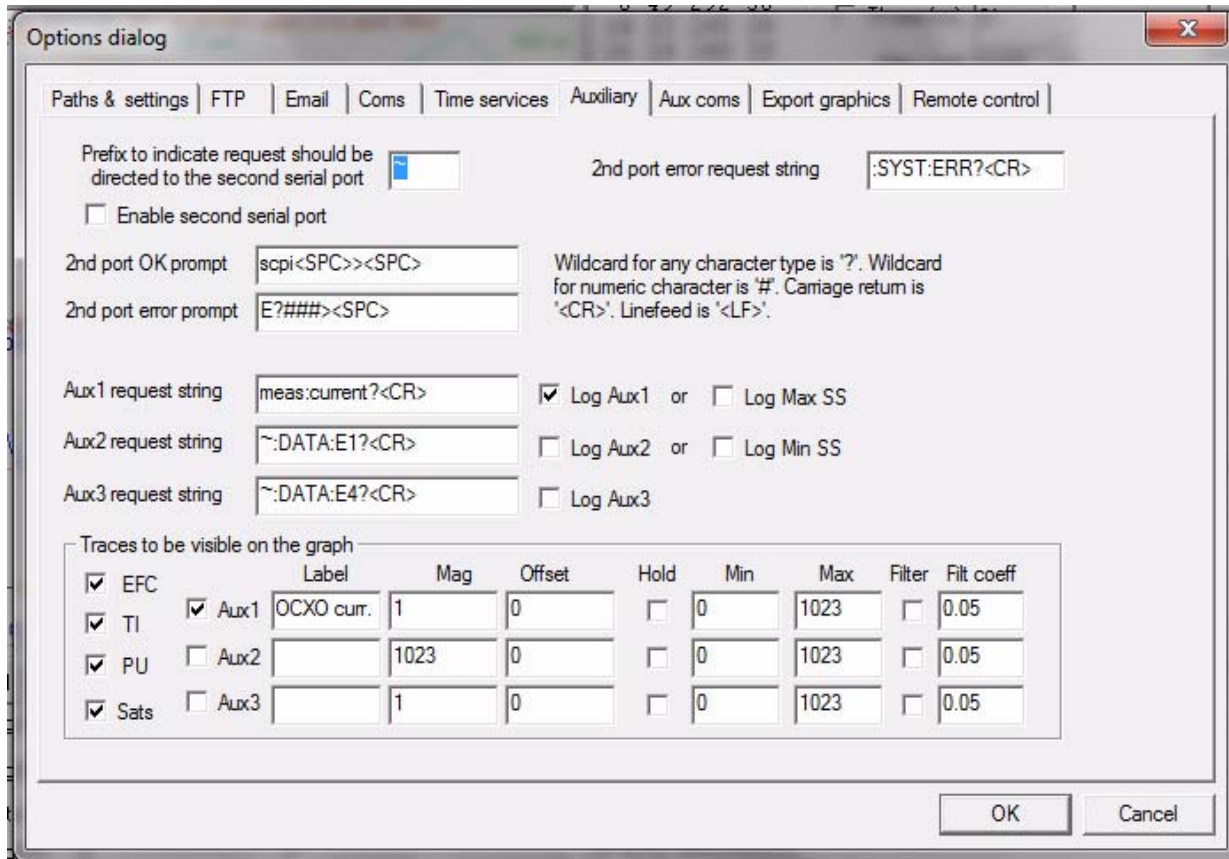


Figure 5.3 Auxiliary Parameters window



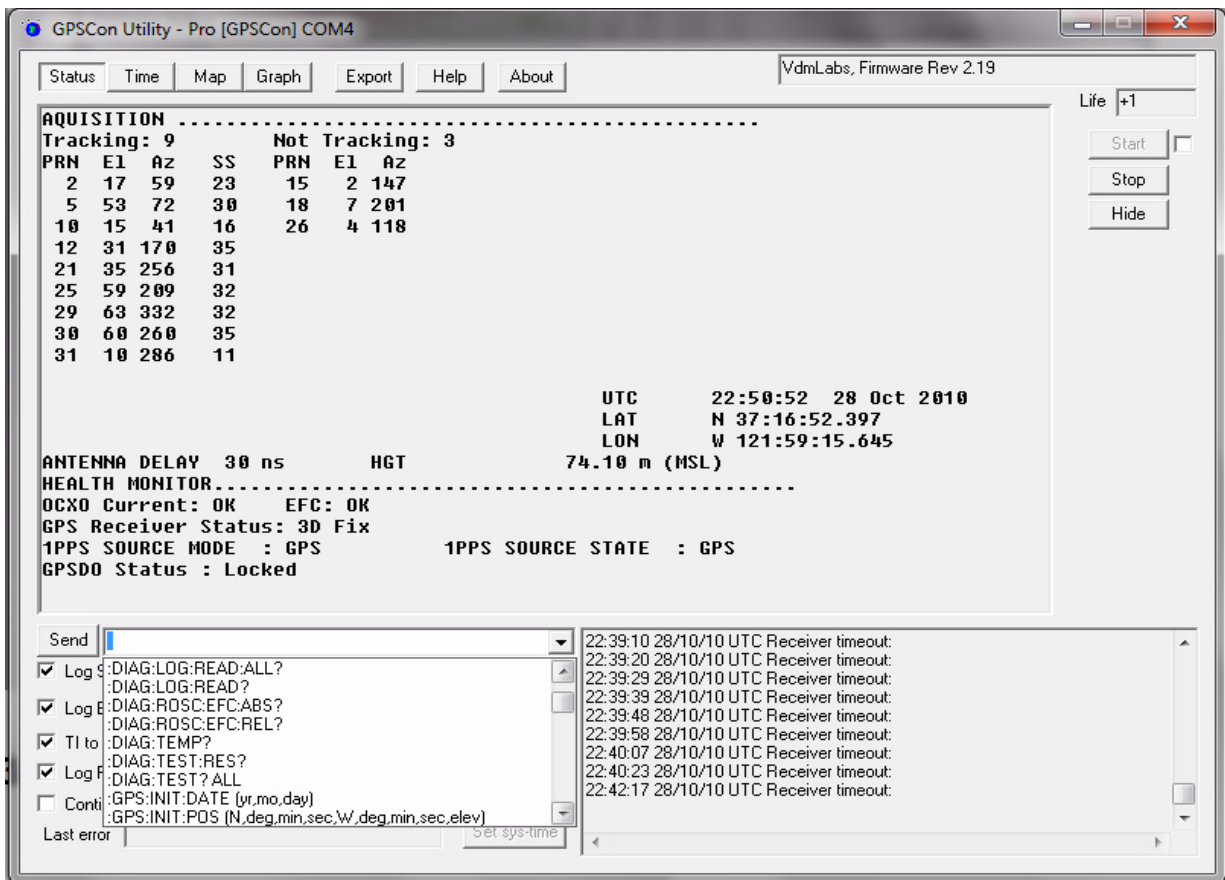
5.3.1.3 Other options

The other tabs in the options window can be selected and you can set whatever options you need, such as paths for logging or exporting graphics.

5.3.2 Sending manual commands to the receiver

You can send SCPI commands manually by using the drop-down box under the display window as shown in [Figure 5.4](#). Care must be taken when sending these commands so be sure that the command that you select is supported by the FireFly-IIA GPSDO. Once you’ve selected the command, you can press “Send” to send it to the FireFly-IIA.

Figure 5.4 Sending manual commands



5.3.3 Use of the mouse in graph mode

Refer to [Figure 5.5](#) for the following description. In graph mode the span of the graph may be set using the span setting. Alternatively, the start and or stop time of the graph may be locked using the mouse. Set the start time by left clicking on the desired start point. If you wish, the stop time may also be locked by right clicking the desired stop point. This can all be unlocked by left double-clicking anywhere on the graph. Double-click always causes all of the selected span data to be displayed. At the next update cycle, the selected span, if not set to zero, will be enforced. However, the left click, and if chosen the right click, always overrides the span setting.

To display all of the data in the file without manually setting the span to zero, you should right double-click in the graph. This has the effect of setting the start time to zero, the stop time to infinity, and asserting the mouse override condition. To release, left double-click.

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

"To zoom in: The mouse is used to set the left extent and the right extent of the portion of the curve that the user wants to fill the screen. Click once with the left mouse button on the point that marks the left side of what you want to be the magnified curve. Immediately that point becomes the left end of the curve. Then similarly click the right mouse button on the curve at the time you wish to be the right most portion of the magnified curve and it immediately becomes the end point on the right side. And, finally to return to the zoomed out ("fit to window") view, left double-click on the curve."

Remember, in order to see all the data in the log file, you must either set the span control to zero, or right double-click in the graph.

When you have locked the start and stop time using the mouse, you can scroll left or right through the data without changing the span. To scroll to a later time, use Shift + Left click. To scroll to an earlier time, use Shift + Right click. Double left click to release everything.

The time span indication at the lower right of the graph will turn red to signify that mouse override is in effect.

Figure 5.5 Graph display

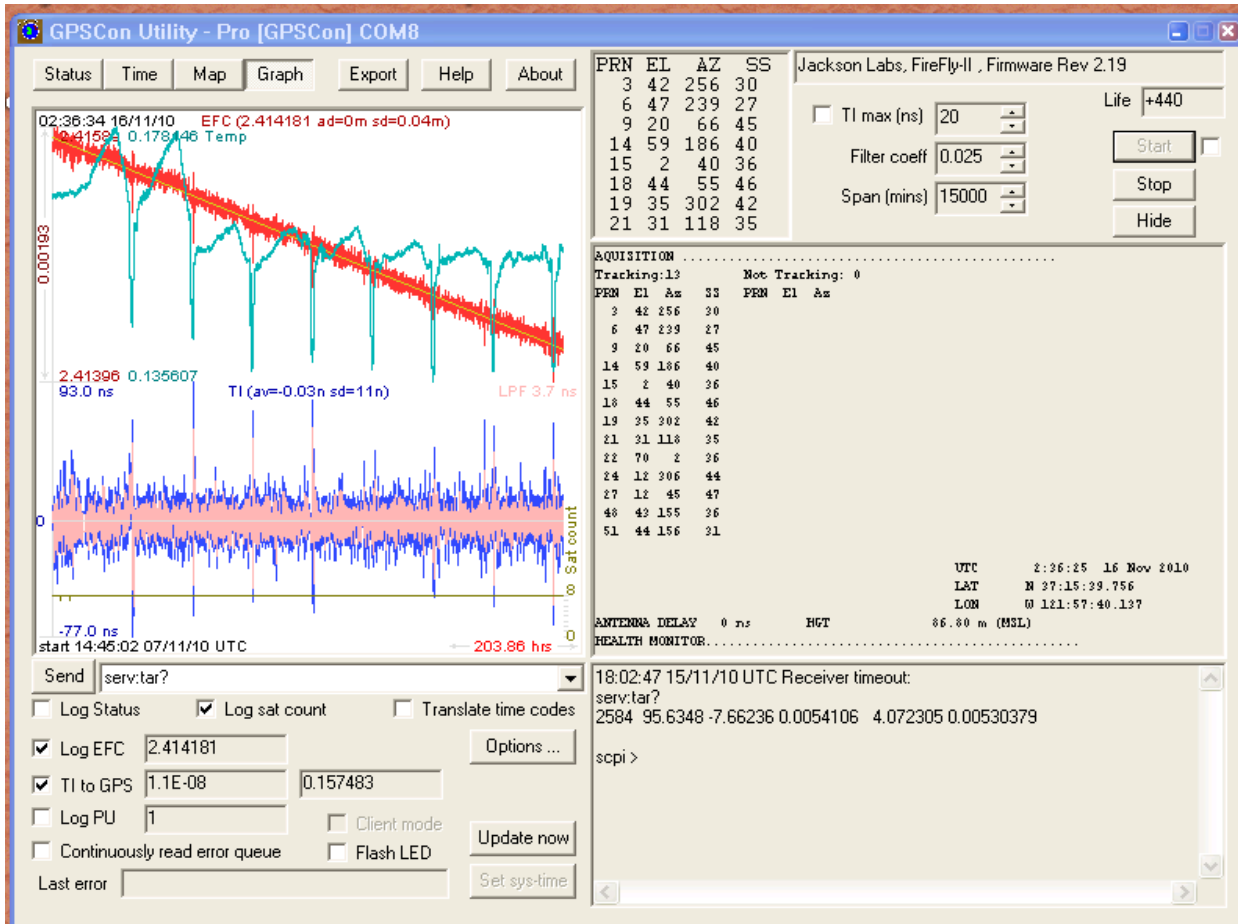
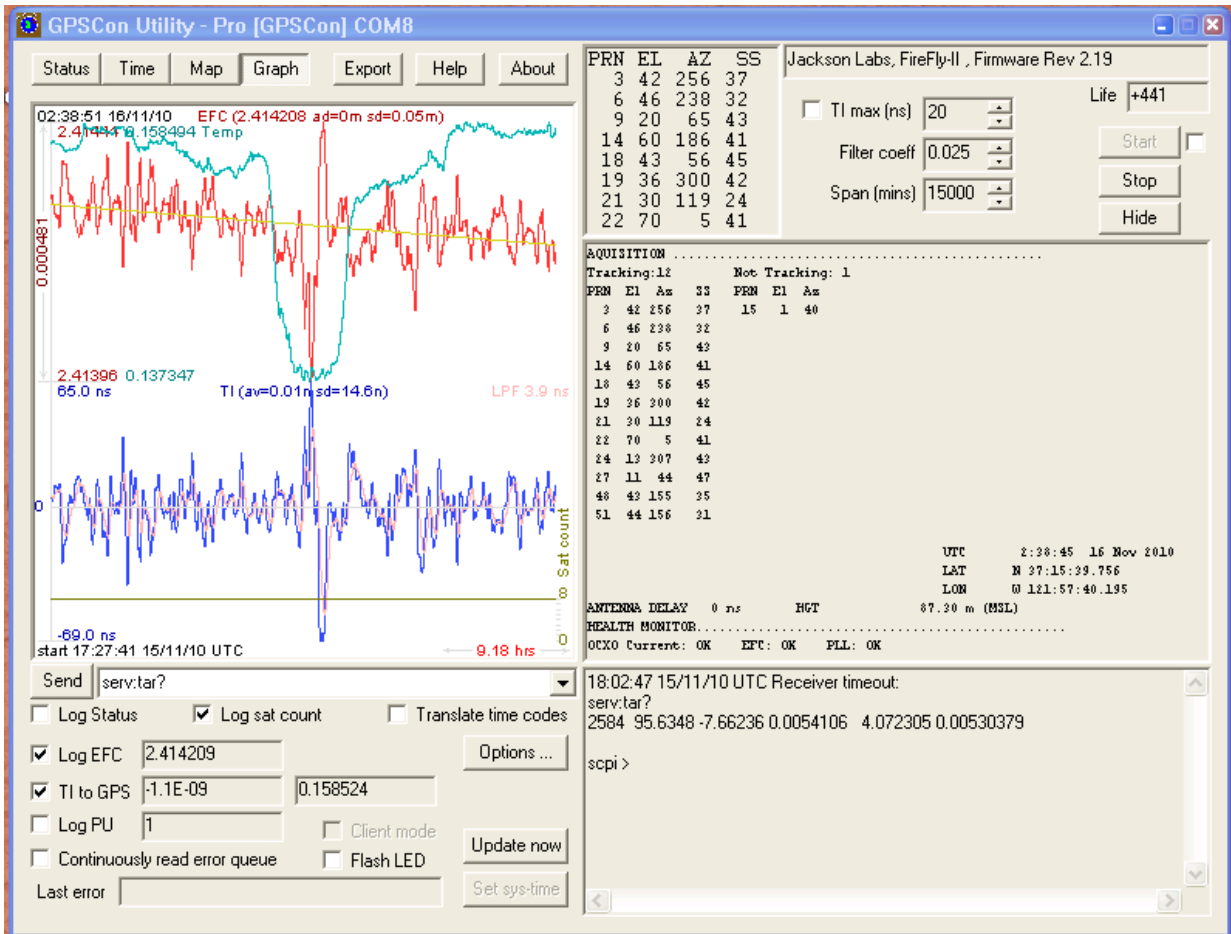


Figure 5.6 Expanded Graph Display



5.3.4 Exporting the graphics

The settings which control the mode of the Export function are contained in the Options dialog.

Export allows you to create an image file of either the graph or the satellite map. You select which you want using the radio buttons. If you select '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 "All" to make this choice.

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

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

5.4 Interpreting the Data

Figure 5.5 shows the data acquired by the FireFly-IIA unit over a period of more than 200 hours. The red trace is EFC (crystal frequency control voltage). The crystal is aging (becoming faster in frequency over time). This requires the control voltage to be lowered to maintain 10.0MHz exactly. A drift of ~2mV is visible over 200 hours. On the left side of the screen the EFC range over this 200 hour plot is displayed vertically as 0.00193V. This means the drift of the EFC voltage due to aging is ~88mV per year. The EFC sensitivity of the crystal is about 8Hz per volt, so the crystal ages at:

$$8\text{Hz/V} * 0.088\text{V/Year} = 0.704\text{Hz/Year drift.}$$

At 10MHz:

$$0.704\text{Hz} / 10\text{MHz} = 7.04\text{E-}08 \text{ aging rate per year.}$$

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

In Figure 5.6, which is a zoom of Figure 5.5, we can see the phase offset error of the internal OCXO to the UTC GPS reference. We can see the maximum drift is -77ns to +93ns. The average is (TI av=-0.03ns). The standard deviation over the 200 hour plot is sd=11ns. This means the average error of the 10MHz phase of this unit over 200 hours is only +/-11ns rms. Or, in other words the average jitter (wander) over 200 hours of operation is:

$$11\text{ns} / 200\text{Hrs} = 1.528\text{E-}014$$

or in other words the unit performs as well as a high quality Cesium Atomic reference clock over long periods of time. The unit disciplines its internal 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.

6 Certification and Warranty

6.1 Certification

Jackson Labs Technologies, Inc. certifies that this product met its published specifications at time of shipment.

6.1.1 Warranty

This Jackson Labs Technologies, Inc. hardware product is warranted against defects in material and workmanship for a period of 1 (one) year from date of delivery. During the warranty period Jackson Labs Technologies, Inc. will, at its discretion, either repair or replace products that prove to be defective. Jackson Labs Technologies, Inc. does not warrant that the operation for the software, firmware, or hardware shall be uninterrupted or error free even if the product is operated within its specifications.

For warranty service, this product must be returned to Jackson Labs Technologies, Inc. or a service facility designated by Jackson Labs Technologies, Inc. Customer shall prepay shipping charges (and shall pay all duties and taxes) for products returned to Jackson Labs Technologies, Inc. for warranty service. Except for products returned to Customer from another country, Jackson Labs Technologies, Inc. shall pay for return of products to Customer. If Jackson Labs Technologies, Inc. is unable, within a reasonable time, to repair or replace any product to condition as warranted, the Customer shall be entitled to a refund of the purchase price upon return of the product to Jackson Labs Technologies, Inc.

6.1.2 Limitation of Warranty

The foregoing warranty shall not apply to defects resulting from improper or inadequate maintenance by the Customer, Customer-supplied software or interfacing, unauthorized modification or misuse, opening of the instruments enclosure or removal of the instruments panels, operation outside of the environmental or electrical specifications for the product, or improper site preparation and maintenance. **JACKSON LABS TECHNOLOGIES, INC. SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR**

PURPOSE. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Jackson Labs Technologies, Inc. products are not intended for use in medical, life saving, or life sustaining applications.

6.1.3 Exclusive Remedies

THE REMEDIES PROVIDED HEREIN ARE THE CUSTOMER'S SOLE AND EXCLUSIVE REMEDIES. JACKSON LABS TECHNOLOGIES, INC. SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER BASED ON CONTRACT, TORT, OR ANY OTHER LEGAL THEORY.