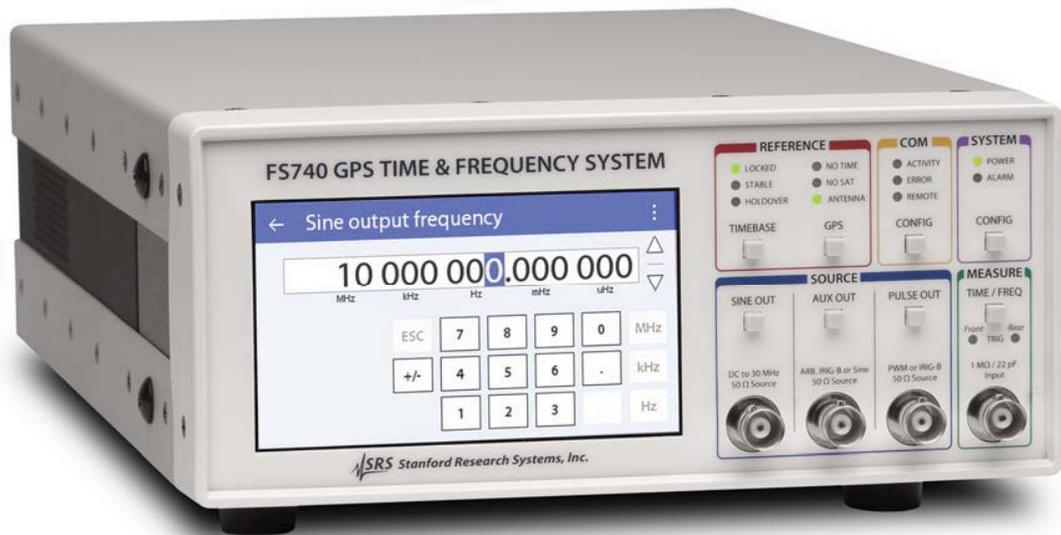




www.lambdaphoto.co.uk  
T: +44 (0)1582 764334

# FS740 GPS Time and Frequency System

## User Manual



---

## Certification

Stanford Research Systems certifies that this product met its published specifications at the time of shipment.

---

## Warranty

This Stanford Research Systems product is warranted against defects in materials and workmanship for a period of one (1) year from the date of shipment.

---

## Service

For warranty service or repair, this product must be returned to a Stanford Research Systems authorized service facility. Contact Stanford Research Systems or an authorized representative before returning this product for repair.

Information in this document is subject to change without notice.

Copyright © Stanford Research Systems, Inc., 2013. All rights reserved.

Stanford Research Systems, Inc.  
1290-C Reamwood Avenue  
Sunnyvale, California 94089  
Phone: (408) 744-9040  
Fax: (408) 744-9049  
[www.thinkSRS.com](http://www.thinkSRS.com)

Printed in the U.S

# Contents

<b>Contents</b>	<b>i</b>
<b>Safety and Preparation for Use</b>	<b>vii</b>
<b>Symbols You May Find on SRS Products</b>	<b>viii</b>
<b>Specifications</b>	<b>ix</b>
<b>Remote Interface Commands</b>	<b>xi</b>
<b>Quick Start Instructions</b>	<b>1</b>
Installing the FS740	1
Correcting UTC for the Antenna Delay	1
Installing the Antenna	2
Selecting a Location for the Antenna	2
Active vs Passive Antennas	2
Antenna Warning LED	2
Antenna Delay Correction	3
<b>Introduction</b>	<b>5</b>
<b>Feature Overview</b>	<b>5</b>
<b>Front-Panel Overview</b>	<b>7</b>
Front Panel Display	7
Display Navigation	7
Navigation Sequences	8
Parameter Modification	9
Numeric Parameters	9
Enter a New Value	9
Modify an Existing Value	9
Navigating Back	9
Enumerations	10
Boolean Check Boxes	10
Actions	10
Top Level Displays and Buttons	11
Timebase	11
GPS	12
Communications	12
System	13
Sine output	14
Aux output	14
Pulse output	14
Measurements	15
Front Panel Connectors	16
Sine Out	16
Aux Out	16
Pulse Out	16

TIME / FREQ	16
<b>Rear-Panel Overview</b>	<b>17</b>
AC Power	17
Remote Interfaces	17
RS-232	17
Ethernet	18
Alarm Relay	18
GPS Antenna	18
Rear Measurement Input	18
Source Outputs	18
Sine Out	18
Aux Out	18
Pulse OUT	18
10 MHz OUT	18
Option Boards	18
<b>Operation</b>	<b>21</b>
<b>Introduction</b>	<b>21</b>
GPS	21
Timebase Stability	22
Wander of the FS40 vs UTC	23
Loop Time Constants	23
<b>Timebase</b>	<b>23</b>
State	24
Related Commands	25
Duration	25
Related Commands	25
Phase Locking to GPS	25
Optimum Time Constant	27
Current Time Constant and Timing Error	27
Related Commands	27
Installed Timebase	27
Related Commands	27
Timebase Events	27
Related Commands	28
Frequency Control	28
Related Commands	28
Configuration	28
Lock to GPS	29
Related Commands	29
Loop Bandwidth	29
Related Commands	29
Loop Time Constant	29
Related Commands	29
Criteria to Enter Holdover	30
Related Commands	30
Criteria to Leave Holdover	30
Related Commands	31

<b>GPS</b>	<b>31</b>
Date and Time	32
Related Commands	32
Position	32
Related Commands	33
Satellites	33
Related Commands	33
Receiver Status	33
Related Commands	34
Configuration	34
Timing Alignment	34
Related Commands	35
Timing Quality	35
Related Commands	35
Local Time Offset	36
Related Commands	36
Survey	36
Related Commands	36
Position Fixes in Survey	36
Related Commands	36
Antenna Corrections	36
Related Commands	37
<b>Communications</b>	<b>37</b>
Remote Mode	38
Activity	38
Related Commands	39
RS232 Configuration	39
Related Commands	40
Ethernet	40
MAC Address	40
Status	40
Speed	40
Related Commands	41
TCP/IP Configuration	41
Configuration Method	41
Static Configuration	42
Reset	42
Current Configuration	43
Related Commands	43
<b>System</b>	<b>43</b>
Save and Recall User Settings	44
Save Settings	44
Recall Settings	44
Related Commands	45
Display	45
Related Commands	45
Device Information	45
Related Commands	46
Alarm	46

Related Commands	46
Alarm Mode	47
Related Commands	47
Alarm Conditions	47
Related Commands	48
Rear Panel Options	48
Related Commands	48
<b>Source Outputs</b>	<b>48</b>
Amplitude	49
Units	49
Related Commands	50
Frequency	50
Related Commands	50
Phase	50
Units	51
Relative Phase	51
Synchronize Phase to UTC	51
Auto Sync	52
Output Delay	52
Related Commands	53
Sine Output	53
Aux Output	53
Related Commands	54
Pulse Output	54
Related Commands	55
<b>Measure</b>	<b>55</b>
Making Measurements	56
Measurement Configuration	57
Related Commands	57
Input Configuration	57
Related Commands	58
Frequency Measurements	58
Frequency Offset	58
Gate	59
Samples	59
Auto Triggering	59
Fast Averaging	60
Timeout	60
Related Commands	60
Frequency Statistics	60
Related Commands	61
Frequency Stability	61
Making Frequency Stability Measurements	62
Related Commands	62
FS740 Baseline Measurement Stability	62
FS740 Timebase Stability	64
Effects of Averaging on Frequency Stability	64
Timing Measurements	64

---

<b>Factory Default Settings</b>	<b>66</b>
<b>Remote Programming</b>	<b>69</b>
<b>Introduction</b>	<b>69</b>
RS-232	69
Ethernet	69
Telnet	69
Bare TCP	69
VXI-11	70
Network Security	70
Front-Panel Indicators	70
<b>SCPI Command Language</b>	<b>71</b>
SubSystems	71
Understanding Command Syntax	71
Keyword Case	72
Punctuation Used in Definitions	72
Numeric Suffix	72
Examples	72
Queries	73
Separators	73
IEEE 488.2 Common Commands	73
Parameter Types	73
Numeric Values	73
Units	74
Special Numbers	75
Discrete Parameters	75
String Parameters	75
Command Termination	76
<b>Status Reporting</b>	<b>76</b>
Architecture	76
Condition Register	76
Event Register	77
Enable Register	77
<b>FS740 Status</b>	<b>79</b>
Serial Poll Status Byte	79
Standard Event Status Register	81
Questionable Status	82
Operation Status	83
GPS Receiver Status	84
<b>Making Measurements</b>	<b>85</b>
Simple Measurements	85
Configure and Read	85
Initiate and Fetch	86
<b>Common IEEE-488.2 Commands</b>	<b>88</b>
<b>Measurement Commands</b>	<b>93</b>
<b>Calculate Subsystem</b>	<b>97</b>
<b>Data Subsystem</b>	<b>99</b>

---

<b>GPS Subsystem</b>	<b>101</b>
<b>Input Subsystem</b>	<b>106</b>
<b>Route Subsystem</b>	<b>107</b>
<b>Sample Subsystem</b>	<b>108</b>
<b>Sense Subsystem</b>	<b>108</b>
<b>Source Subsystem</b>	<b>110</b>
<b>Status Subsystem</b>	<b>116</b>
<b>System Subsystem</b>	<b>118</b>
<b>Timebase Subsystem</b>	<b>129</b>
<b>Trigger Subsystem</b>	<b>134</b>
<b>Error Codes</b>	<b>135</b>
Command Errors	135
Execution Errors	136
Device Specific Errors	136
Query Errors	136
Instrument Errors	137
<b>Example Programming Code</b>	<b>138</b>
<b>FS740 Operation Verification</b>	<b>143</b>
<b>Overview</b>	<b>143</b>
Equipment Required	143
<b>Output Power Tests</b>	<b>143</b>
BNC Output Power Test	143
<b>Conclusions</b>	<b>143</b>
<b>Circuit Description</b>	<b>149</b>
<b>Parts List</b>	<b>151</b>
<b>Appendix A : Schematic Diagrams</b>	<b>153</b>
<b>Revisions</b>	<b>157</b>

## Safety and Preparation for Use

### Line Voltage

The instruments operate from a 90 to 132 V<sub>AC</sub> or 175 to 264 V<sub>AC</sub> power source having a line frequency between 47 and 63 Hz. Power consumption is less than 80 VA total. This instrument is intended to be powered at all times. Therefore, there is no power switch. Power is applied to the instrument as soon as the line cord is plugged in.

### Power Entry Module

A power entry module, labeled AC POWER on the back panel of the instrument, provides connection to the power source and to a protective ground.

### Power Cord

The unit is shipped with a detachable, three-wire power cord for connection to the power source and protective ground.

The exposed metal parts of the box are connected to the power ground to protect against electrical shock. Always use an outlet which has a properly connected protective ground. Consult with an electrician if necessary.

### Grounding

BNC shields are connected to the chassis ground and the AC power source ground via the power cord. Do not apply any voltage to the shield.

### Line Fuse

The line fuse is internal to the instrument and may not be serviced by the user.

### Operate Only with Covers in Place

To avoid personal injury, do not remove the product covers or panels. Do not operate the product without all covers and panels in place.

### Serviceable Parts

There are no user serviceable parts. Refer service to a qualified technician.

## Symbols You May Find on SRS Products

Symbol	Description
	Alternating Current
	Caution – risk of electrical shock
	Frame or Chassis terminal
	Caution – refer to accompanying document
	Earth (ground) terminal
	Battery
	Fuse
	Power On
	Power Off
	Power Standby

## Specifications

### Standard TCXO Timebase

Oscillator type	Oven controlled, 3 <sup>rd</sup> OT, AT-cut crystal
Stability	$<2 \times 10^{-6}$ (20 to 30°C)
Aging (undisciplined to GPS)	$<5$ ppm/year

### OCXO Timebase

Oscillator type	Oven controlled, 3 <sup>rd</sup> OT, SC-cut crystal
Stability	$<1 \times 10^{-9}$ (20 to 30°C)
Aging (undisciplined to GPS)	$<0.2$ ppm/year

### Rubidium Timebase

Oscillator type	Oven controlled, 3 <sup>rd</sup> OT, SC-cut crystal
Physics package	Rubidium vapor frequency discriminator
Stability	$<1 \times 10^{-10}$ (20 to 30°C)
Aging (undisciplined to GPS)	$<0.0005$ ppm/year

### GPS Receiver

Power on to satellite acquisition	$<1$ minute (typical)
Time to acquire almanac	$\sim 15$ minutes when continuously tracking satellites
Optimized for static applications	Over determined clock mode enables receiver to use all satellites for timing.
Accuracy of UTC	$< 100$ ns
Timing wander	$< 15$ ns rms (in over determined clock mode)
Antenna delay correction range	$\pm 0.1$ s

### Sine Output (50 $\Omega$ load)

Direct DDS generation	
Frequency range	1 MHz to 30.1 MHz
Frequency resolution	1 $\mu$ Hz
Frequency error	$<10$ pHz + timebase error $\times f_c$
Phase settability	1 mDeg
Phase accuracy to internal reference	$<1$ ns
Amplitude	10 mV <sub>PP</sub> to 1.414 V <sub>PP</sub>
Amplitude resolution	$<1$ %
Amplitude accuracy	$\pm 5$ %
Harmonics	$<-40$ dBc
Spurious	$<-70$ dBc
Output coupling	DC, 50 $\Omega$ $\pm 2$ %
User load	50 $\Omega$
Reverse protection	$\pm 5$ V <sub>DC</sub>

### Aux Output (50 $\Omega$ load)

Output options:	Sine, Triangle, Square, 100MHz, or AM IRIG-B
Frequency range	
Sine	1 MHz to 10 MHz
Triangle or square	1 MHz to 1 MHz
100 MHz sine	100 MHz

AM IRIG-B	1 kHz
Frequency resolution	1 $\mu$ Hz
Frequency error	$<10 \text{ pHz} + \text{timebase error} \times f_C$
Phase settability	1 mDeg (cannot adjust phase of 100MHz sine output)
Amplitude (sine, triangle, square)	10 mV <sub>PP</sub> to 1.414 V <sub>PP</sub>
Amplitude (100 MHz)	2.75 dBm $\pm$ 0.5 dBm
Amplitude resolution	$<1 \%$
Amplitude accuracy	$\pm 5 \%$
Harmonics	$<-40 \text{ dBc}$
Spurious	$<-70 \text{ dBc}$
Output coupling	DC, 50 $\Omega$ $\pm 2 \%$
User load	50 $\Omega$
Reverse protection	$\pm 5 V_{DC}$

### Pulse Output

Output options:	Period/width, Freq/duty, Pulse IRIG-B
Period	40 ns to 1000 s
Width	5 ns to (Period – 5 ns)
Period/width resolution	1 ps
Frequency range	1 mHz to 25 MHz
Frequency resolution	1 $\mu$ Hz
Frequency error	$<10 \text{ pHz} + \text{timebase error} \times f_C$
Jitter	$<50 \text{ ps (rms)}$
Level	+5 V CMOS logic
Transition time	$<2 \text{ ns}$
Source impedance	50 $\Omega$

### Time and Frequency Input

Time tag resolution:	1 ps
Time tag jitter (rms)	$<50 \text{ ps}$
Frequency resolution	1 $\mu$ Hz
Measurement stability (1 s gate)	$<5 \times 10^{-12}$ (synchronous with fast averaging enabled) $<5 \times 10^{-11}$ (otherwise)

### Computer Interfaces (standard)

Ethernet (LAN)	10/100 Base-T. TCP/IP & DHCP default.
RS-232	4.8k-115.2k baud, RTS/CTS flow

### General

Line power	$<90 \text{ W}$ , 90 to 264 V <sub>AC</sub> , 47 to 63 Hz with PFC
EMI Compliance	FCC Part 15 (Class B), CISPR-22 (Class B)
Dimensions	8.5" $\times$ 3.5" $\times$ 13" (W $\times$ H $\times$ D)
Weight	$<10 \text{ lbs}$
Warranty	One year on parts and labor

## Remote Interface Commands

<b>Common Commands</b>	<b>Page</b>
*CLS	88
*ESE	88
*ESR?	88
*IDN?	89
*OPC	89
*OPT?	89
*PSC	89
*RCL	90
*RST	90
*SAV	90
*SRE	91
*STB?	91
*WAI	91

<b>Measurement Commands</b>	<b>Page</b>
MEASure:FREQuency?	93
MEASure:TIME?	93
CONFigure?	94
CONFigure:FREQuency	94
CONFigure:TIME	94
INITiate	95
FETCh?	96
ABORT	96
STOP	96

<b>Calculate Subsystem</b>	<b>Page</b>
CALCulate:FILTer	97
CALCulate:REFErence	97
CALCulate:STABility	98
CALCulate:STATistics	98

<b>Data Subsystem</b>	<b>Page</b>
DATA:COUNT?	99
DATA:POINTs?	99
DATA:READ?	100
DATA:REMOve?	100

<b>GPS Subsystem</b>	<b>Page</b>
GPS:CONFig:MODe	101
GPS:CONFig:SAVe	101
GPS:CONFig:SURVey:Mode	101
GPS:CONFig:SURVey:FIXes	102
GPS:CONFig:ALIGnment	102

GPS:CONFIg:QUALity	103
GPS:CONFIg:ADELay	103
GPS:POSition?	103
GPS:POSition:HOLD:STATe?	104
GPS:POSition:SURVey:DELeTe	104
GPS:POSition:SURVey:PROGress?	104
GPS:POSition:SURVey:SAVe	104
GPS:POSition:SURVey:STARt	105
GPS:POSition:SURVey:STATe?	105
GPS:SATellite:TRACking?	105
GPS:SATellite:TRACking:STATus?	105
GPS:UTC:OFFSet?	106

<b>Input Subsystem</b>	<b>Page</b>
INPut:LEVeL	106
INPut:SLOPe	107

<b>Route Subsystem</b>	<b>Page</b>
ROUte:OPTion	107

<b>Sample Subsystem</b>	<b>Page</b>
SAMPlE:COUNt	108

<b>Sense Subsystem</b>	<b>Page</b>
SENSe:FREQuency:GATE	108
SENSe:FREQuency:TImeout	109
SENSe:TIme:BMODE	109

<b>Source Subsystem</b>	<b>Page</b>
SOURce:FREQuency	110
SOURce2:FUNction	110
SOURce3:FUNction	111
SOURce:PHASe	111
SOURce:PHASe:REFerence	112
SOURce:PHASe:SYNChronize	112
SOURce:PHASe:SYNChronize:AUTO	113
SOURce:PHASe:SYNChronize:TDELay	113
SOURce3:PULSe:DCYCLe	113
SOURce3:PULSe:PERiod	114
SOURce3:PULSe:VIEW	114
SOURce3:PULSe:WIDTh	115
SOURce:VOLTagE	115
SOURce:VOLTagE:UNITs	115

<b>Status Subsystem</b>	<b>Page</b>
-------------------------	-------------

STATus:GPS:CONDition?	116
STATus:GPS:ENABle	116
STATus:GPS:EVENT?	116
STATus:OPERation:CONDition?	117
STATus:OPERation:ENABle	117
STATus:OPERation:EVENT?	117
STATus:QUEStionable:CONDition?	117
STATus:QUEStionable:ENABle	118
STATus:QUEStionable:EVENT?	118

<b>System Subsystem</b>	<b>Page</b>
SYSTem:ALARm?	119
SYSTem:ALARm:CLEAr	119
SYSTem:ALARm:CONDition?	119
SYSTem:ALARm:ENABle	119
SYSTem:ALARm:EVENT?	120
SYSTem:ALARm:FORCe:STATe	120
SYSTem:ALARm:MODE	120
SYSTem:ALARm:GPS:TINTerval	121
SYSTem:ALARm:HOLDoVer:Duration	121
SYSTem:COMMunicate:LAN:DHCP	122
SYSTem:COMMunicate:LAN:EPHY?	121
SYSTem:COMMunicate:LAN:EPHY:SPEEd	122
SYSTem:COMMunicate:LAN:GATeway	122
SYSTem:COMMunicate:LAN:IPADress	123
SYSTem:COMMunicate:LAN:MAC?	123
SYSTem:COMMunicate:LAN:SMASK	124
SYSTem:COMMunicate:SERial:BAUD	124
SYSTem:COMMunicate:SERial:RESet	124
SYSTem:COMMunicate:LOCK?	125
SYSTem:COMMunicate:UNLock?	125
SYSTem:DATE	125
SYSTem:DISPlay:POWer	126
SYSTem:DISPlay:SCReen	126
SYSTem:ERRor?	127
SYSTem:SECurity:IMMediate	127
SYSTem:TIME	127
SYSTem:TIME:LOFFset	128
SYSTem:TIME:POWeron	128

<b>Timebase Subsystem</b>	<b>Page</b>
TBAsE:CONFig:BWIDth	129
TBAsE:CONFig:HMODE	129
TBAsE:CONFig:LOCK	129
TBAsE:CONFig:TINTerval:LIMit	130
TBAsE:EVENT:CLEAr	130
TBAsE:EVENT:COUNT	130
TBAsE:EVENT:NEXT?	131
TBAsE:FCONtrol	131

TBASE:FCONtrol:SAVe	131
TBASE:STATe?	132
TBASE:STATe:HOLdover:DURation?	132
TBASE:STATe:LOCK:DURation?	133
TBASE:STATe:WARMup:DURation?	133
TBASE:TCONstant	133
TBASE:TINterval	134

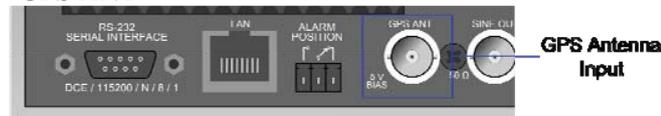
<b>Trigger Subsystem</b>	<b>Page</b>
TRIGger:CONtinuous	134

# Quick Start Instructions

## Installing the FS740

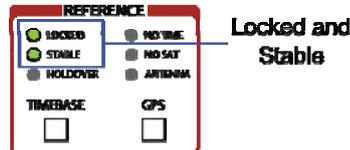
To setup and install the FS740 follow these steps:

1. Install the GPS antenna in a location that has a clear view of the sky, such as the roof of a building. See below for details.
2. Connect the cable from the antenna to the rear panel input of the FS740 labeled “GPS ANT”



3. Power on the FS740 by plugging in the AC power cord.

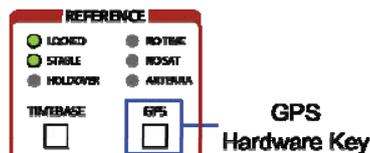
If you don't care about the absolute accuracy of the FS740's phase relative to UTC then setup is complete (otherwise, see below). The FS740 will automatically search for GPS satellites and lock its internal timebase to them. Once locked, the FS740 may take up to 1 hour to fully stabilize. When the FS740 is fully stabilized the front panel locked and stable LEDs should be on.



## Correcting UTC for the Antenna Delay

The user can improve the absolute calibration of the FS740's estimate of UTC by computing and entering an antenna delay, the amount of time it takes for signals to travel from the antenna to the FS740. To do this, follow these additional steps:

4. Measure the length of cable in feet that was used to connect the antenna to the FS740.
5. With the FS740 powered on, press the hardware key labeled GPS to activate the top level GPS display.



- Navigate to GPS > Config > Antenna corrections



- To calculate the cable delay for the antenna, multiply 1.5417 ns/ft by the length of the cable in feet as measured in step 4, and enter the result as a NEGATIVE number in ns using the virtual numeric key pad.

## Installing the Antenna

### Selecting a Location for the Antenna

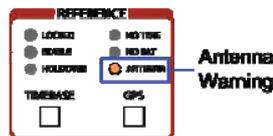
The signals broadcast by the GPS satellites are extremely weak and difficult to detect. Generally speaking, you will get best results if the antenna has a clear unobstructed view of the sky. This is commonly on the roof of the building within which the FS740 is located. If this is not possible, the user can try locating the antenna at a window. Doing so, however, may degrade the quality and reliability of the GPS signal as fewer satellites will typically be visible and with less SNR. This can degrade the long term stability of the FS740 by a factor of three.

### Active vs Passive Antennas

The antenna provided by SRS is an active antenna. We recommend using active antennas with the FS740 to ensure sufficient signal is available at the receiver input. 5 VDC is applied to the antenna cable to power the active antenna. Active antennas typically have more gain and improved signal detection capability over passive antennas. They also help to overcome the signal loss introduced by long antenna cables and, therefore, are necessary in most cases.

### Antenna Warning LED

The orange LED on the front panel labeled ANTENNA warns if the FS740 detects that the antenna input is either an open or short circuit. It typically goes off when the antenna is properly connected to the FS740.



The state of the LED, however, is based solely on the DC current draw on the antenna cable, not on measured signal levels. If the current draw is less than 5 mA, the FS740 will warn of an open circuit. If the current draw is greater than 75 mA, the FS740 will warn of a short circuit. The total current draw for the antenna is limited to 300 mA. Please note that the performance of the FS740 is not impacted at all if the power draw of your antenna falls outside this range, The FS740's GPS receiver will continuously search for GPS satellites regardless of the current draw on the antenna. The warning LED is merely meant to be a helpful diagnostic to consider when troubleshooting a lack of signal for the most common installations.

## Antenna Delay Correction

The FS740’s estimate of UTC is based on when the receiver detects the signal, not when the signal arrives at the antenna. If a long cable is needed to connect the antenna to the FS740, this delay can be significant. The typical delay for most BNC cables is 1.5417 ns per foot. Thus, for a 30 ft cable, the delay would be  $1.5417 \times 30 = 46.25$  ns. If uncorrected, the FS740’s estimate of UTC would be 46.25 ns later than a properly calibrated unit.

The user can correct for this delay by entering a negative correction:  $-46.25$  ns in this case. To do this navigate to GPS > Config > Antenna corrections and enter a correction of  $-46.25$  ns. The correction will be stored in nonvolatile memory and need not be re-entered when the unit is power cycled. When entered correctly, the display will look like that shown in Figure 1.

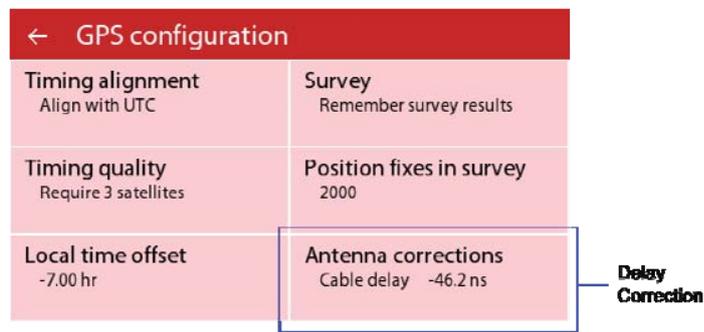


Figure 1: Antenna delay correction



---

# Introduction

---

## Feature Overview

The FS740 GPS Time and Frequency System is primarily designed to provide continuously calibrated time and frequency distribution to a local laboratory. Calibration is maintained by locking its internal reference to the time of day signals broadcast by the GPS satellites. The satellites which make up the GPS network are equipped with rubidium and cesium clocks which are monitored and controlled by the U.S. Air Force which steers GPS time to maintain synchronicity with UTC as maintained by the U.S. Naval Observatory (USNO). The FS740 includes a GPS receiver that is specifically designed to generate precise timing. When locked to the GPS constellation of satellites, the FS740 can align its outputs and time tag signals on its inputs to UTC with a relative precision of 5 ps, an RMS deviation of less than 15 ns and an absolute accuracy of 100 ns.

On its front panel, the FS740 provides three user configurable outputs: a sine output, an aux output, and a pulse output. The sine output supports the generation of frequencies from 1 mHz to 30.1 MHz and amplitudes ranging from 10 mV<sub>PP</sub> to 1.414 V<sub>PP</sub>. Its phase can be adjusted with 1 millidegree of relative precision. Furthermore, if the FS740 is locked to GPS, the zero crossings of the output can be synchronized to UTC if desired.

The aux output supports the generation of several different signals: sine waves, triangle waves, zero-centered square waves, a fixed 100 MHz sine wave, and an AM modulated IRIG-B timing code. Frequencies up to 10 MHz are supported for sine waves, and up to 1 MHz for triangle and square waves. Like the sine output, amplitudes for these three signals may range from 10 mV<sub>PP</sub> to 1.414 V<sub>PP</sub>. Its phase can be adjusted with 1 millidegree of relative precision, and synchronized to UTC if desired. The 100 MHz sine output has a fixed amplitude (2.75 dBm) and frequency and is intended for synchronization of high frequency equipment. Finally, the IRIG-B signal may be used for time of day distribution

The pulse output generates 5 V CMOS waveforms with frequencies ranging from 1 mHz to 25 MHz, and pulse widths ranging from 5 ns to the period – 5 ns. The user may specify a period and pulse width, or a frequency and duty cycle. Periods and pulse width may be specified with 1 ps of resolution. Frequencies may be specified with 1 uHz resolution. The phase of the pulses can be adjusted with 5 ps of resolution. If the FS740 is locked to GPS, the rising edge of the pulse output can be synchronized to UTC if desired. Finally, the pulse output can also be configured to generate pulse-width-modulated IRIG-B timing codes for time of day distribution.

For expanded distribution, the FS740 comes standard with copies of the three front panel outputs as well as a 10 MHz output on its rear panel. Any of these outputs, including the 10 MHz output, can be expanded with up to 15 more buffered outputs with the installation of up to three option boards on the rear panel, thereby providing ample distribution capability for a local laboratory.

In addition to distributing time and frequency signals to other instrumentation, the FS740 also includes the ability to measure time and frequency of external signals. It can time tag an event relative to UTC with less than 50 ps of jitter. Alternatively, it can measure frequencies of up to 200 MHz with gates ranging from 10 ms to 1000 s, providing up to 12 digits of resolution in a 1 second gate interval. Two measurement inputs are available to the user: one on the front panel and one on the rear panel. Both inputs may be operated independently, and simultaneously.

Finally, the FS740 is ideally suited to making frequency stability measurements. It generates averaged time tags at 10 ms intervals with no dead time enabling it to compute Allan deviation stability results for intervals from 10 ms to 50 million seconds automatically. Measurements can be easily configured from the front panel and results viewed in real time without an external computer. Long term stability measurements can be initiated and left to run autonomously for months at a time if desired. Results may be periodically downloaded while the measurement continues to run.

## Front-Panel Overview

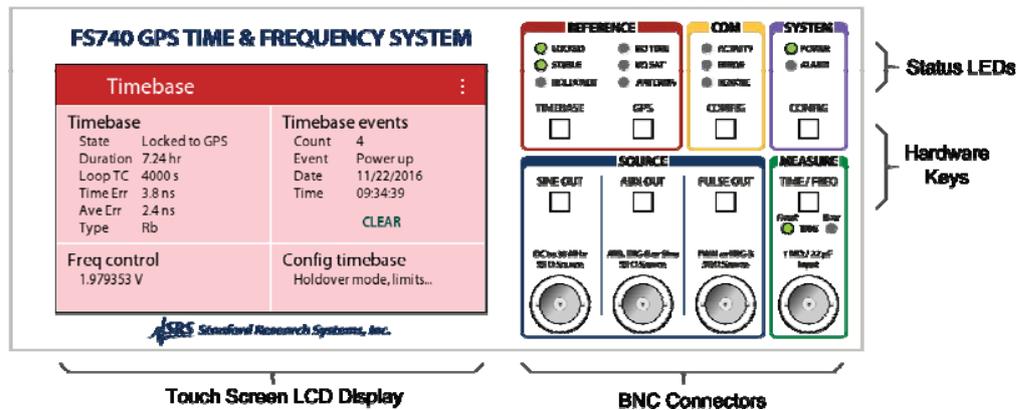


Figure 2: The FS740 front panel

The FS740 front panel consists of a touch screen, color, LCD display with 4 BNC connectors at the bottom right and 8 hardware keys divided into 5 sections: reference, communications, system, source outputs, and measurement inputs. Each section is color coded to assist the user in identifying common functionality. The hardware keys in each section provide direct access to status and settings related to that section or function. Dedicated LEDs highlight important status related to the given section. A red box surrounds the reference section which provides information on the internal timebase and GPS. The timebase is directly coupled to the GPS receiver which provides absolute timing pulses to which the timebase locks. A yellow box surrounds the communications section which provides access to remote interface configuration and status. A purple box surrounds the system section which provides access to overall system configuration. A blue box surrounds the 3 source outputs: sine out, aux out, and pulse out. Dedicated keys provide direct access to the configuration of each output. Finally, a green box surrounds the measurement input. This section enables the user to configure time and frequency measurements of signals applied to either the front or rear input.

## Front Panel Display

A touch screen, color LCD display provides easy access to status and configuration information for the FS740. All settings of the FS740 may be easily updated through interaction with the display.

## Display Navigation

Dedicated hardware keys on the right hand side of the front panel enable the user to immediately navigate to status and configuration information related to that section or item. Virtual cards and buttons on the touch screen, itself, allow the user to further navigate through all the configuration parameters associated with a given section. The front panel display is organized as follows. At the top is a title bar which displays the overall title for the screen. It is color coded to match the section color on the right hand side of the front panel to which it applies. Below the title, the screen is divided into virtual cards that display information on a specific function. Virtual cards are separated from each other by white space to clearly delineate them. Each card contains a subtitle followed by abbreviated configuration and status information related to it. Finally, at the

far right of the title bar is a virtual button consisting of 3 white dots aligned vertically. This button provides access to a menu of settings for lesser used configuration parameters and options. All of these features are highlighted in Figure 3, which shows the main display when the hardware GPS key in the REFERENCE section of the front panel is pressed.

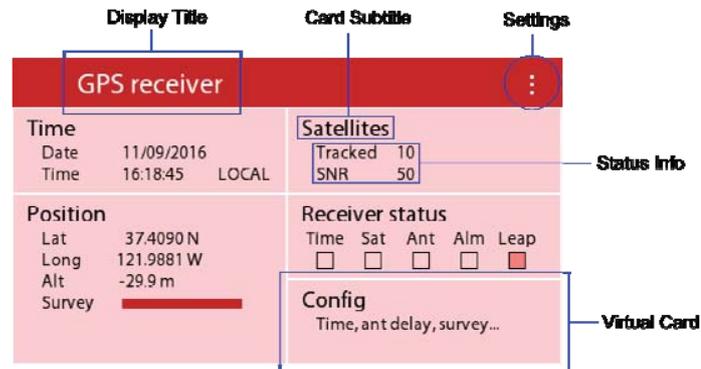


Figure 3: Display organization with various components highlighted.

Pressing a virtual card will normally enable the user to alter a configuration parameter, or display more detailed information about it. Pressing the Satellites card shown in Figure 3 will cause the FS740 to display more detailed information on the GPS Satellites as shown in Figure 4.

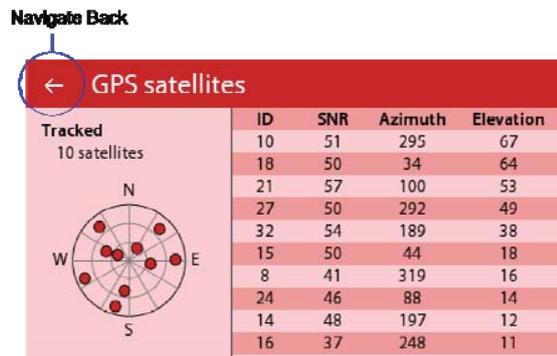


Figure 4: Detailed display with navigate back highlighted.

Detailed information screens may or may not have additional cards that refine the information display. Nevertheless, all of them will have a virtual button with a left facing arrow located at the top left of the title bar. Use this button to navigate back to the previous display. In this case it leads back to the GPS Receiver display.

## Navigation Sequences

In this manual we will often need to refer to navigation sequences which describe how to modify or configure a specific parameter. The navigation sequence is the sequence of virtual key presses that take you to a specific display. The sequence starts with the title of the main display followed by the title of each virtual card pressed to get to the display of interest. Each key press is separated by a greater than sign (>). For example, the detailed satellite information shown in Figure 4 has the following navigation sequence:

GPS receiver > Satellites

The sequence starts with the GPS receiver display. This is one of eight top level displays associated with the eight hardware keys located on the right hand side of the front panel above the BNCs. Press the appropriate hardware key to arrive at desired top level display. After that, the navigation sequence identifies the subtitles of cards pressed to arrive at the display of interest. In this case the user should press the virtual card, Satellites, to navigate to the GPS satellites display.

## Parameter Modification

### Numeric Parameters

Many parameters take numerical values. In this case the user has a number of options for entering or modifying the parameter. To enter the sine output frequency, for example, navigate to Sine Output > Frequency. A sample screen is shown in Figure 5. The user has two options: enter a new value directly, or modify the existing value.

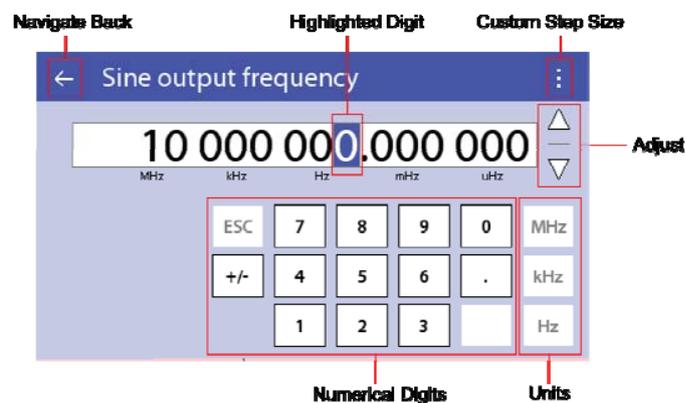


Figure 5: Parameter modification.

### Enter a New Value

To enter a new value use the numerical digit keys to enter the number and complete the entry by selecting the desired units for the entry. To enter 5 MHz, for example, press the keys '5' and 'MHz' in sequence.

### Modify an Existing Value

To modify the existing value, use the up and down arrow keys to adjust the value up and down, respectively. The highlighted digit identifies the step size for the adjustment. For the display shown in Figure 5, the value will be adjusted in 1 Hz steps. The user can easily change the step size to any desired resolution by touching the digit he wants to be highlighted.

The user can also enter a custom step size using the Settings button (three vertical dots) at the far right of the title bar. Select 'Custom step size' and enter the desired step size using the virtual numeric keypad. Then use the up and down arrows to step the value at the custom step size. The highlighted digit will reflect the most significant digit of the custom step size.

### Navigating Back

When entering values directly, the screen will typically navigate back to the previous screen, automatically, once the numeric entry is completed. This is not the case when

modifying values, however. Here, the user must press the left facing arrow at the left of the title bar to navigate back.

## Enumerations

Some parameters are not numeric in nature. Rather, they take on one of a small set of enumerated values. The Aux output configuration is an example of this. Navigate to Aux output > Config, and the display will be similar to that shown in Figure 6.

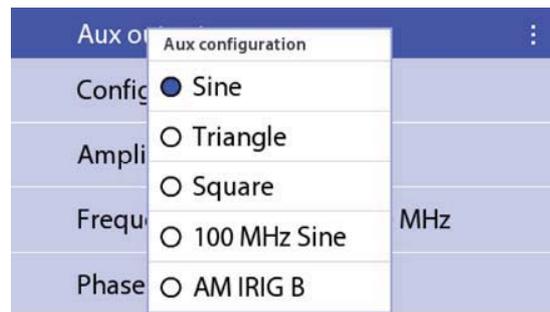


Figure 6: Parameter enumeration.

The display shows that Sine is the currently configured waveform. Simply press one of the other options to change the selection. Press anywhere else to cancel the selection process.

## Boolean Check Boxes

Some functions are either on or off. These are often represented with check boxes. When checked, the function is enabled. When unchecked, the function is disabled or off. An example of this is shown in Figure 7.

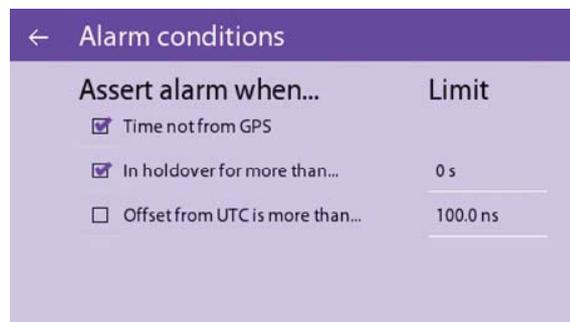


Figure 7: Boolean check boxes

Here the alarm is asserted if the time of day has not been set by GPS or if the timebase goes into holdover. It is not affected by the measured offset from UTC. The user can easily toggle a selection by touching the desired check box.

## Actions

Sometimes, rather than setting a value, the user may need to initiate an action. This is usually accomplished with an action button. Action buttons are usually represented as a single word or short phrase written in dark-green, capital letters. An example of this is shown in Figure 8.

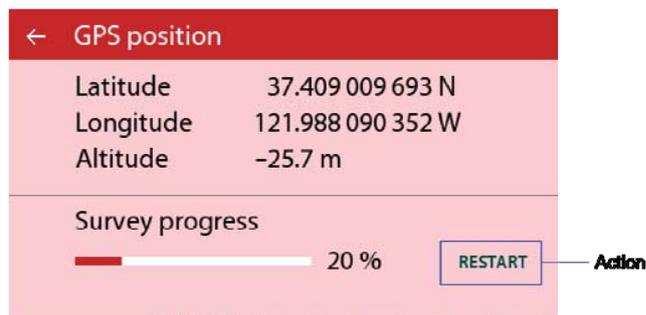


Figure 8: Action button

Here, pressing the RESTART button would restart the position survey, which is currently 20% complete.

## Top Level Displays and Buttons

On the right hand side of the front panel, above the 4 BNCs, are 8 physical buttons identified as hardware keys in Figure 2. Each button is associated with a top level display that enables the user to view status and configuration information for a given area of functionality.

### Timebase

The top level timebase display provides configuration and status information for the internal timebase. It is composed of four virtual cards: timebase status, timebase events, frequency control, and configuration. A sample display is shown in Figure 9.

Timebase			
<b>Timebase</b>	Status Locked to GPS	<b>Timebase events</b>	Count 4
	Duration 21.02 hr		Event Power up
	Loop TC 4000 s		Date 11/10/2016
	Time Err 4.7 ns		Time 15:28:52
	Ave Err -2.6 ns		<b>CLEAR</b>
	Type Rb		
<b>Freq control</b>	1.983794 V	<b>Config timebase</b>	Holdover mode, limits...

Figure 9: Top level timebase display

The timebase card shows the current state of the internal timebase, how long it has been in that state, detailed information on the phase lock loop that locks the internal timebase to GPS, and the type of timebase installed.

The phase lock loop information includes the current loop time constant as well as the most recent timing error and average timing error of the timebase relative to GPS. Positive errors indicate the timebase lags GPS. Negative errors indicate the timebase leads GPS.

The FS740 automatically captures each time the state of the timebase changes as an event. Every time the timebase either locks or unlocks from GPS, for instance, the event will be logged. These events may be viewed one by one on the Timebase events card.

The frequency control voltage is the voltage applied to the frequency input of the timebase to maintain lock to GPS. It is updated every second when the receiver is generating timing pulses. When the timebase is locked to GPS, the frequency control value cannot be changed by the user. If the timebase is not locked to GPS, the user may change the frequency of the timebase by updating this voltage.

## GPS

The top level GPS receiver display provides configuration and status information for the GPS receiver. It is composed of five virtual cards: time of day, position, satellites, receiver status, and configuration. A sample display is shown in Figure 10.



**Figure 10: Top level GPS receiver display**

The time of day card shows the current date and time of day as discerned by the GPS receiver. If the time of day is unknown, it will warn that the time is unset.

The position card shows the latitude, longitude and altitude of the antenna for the receiver. Latitude and longitude are presented in degrees. The altitude is in meters above average sea levels. Also shown is a bar graph indicating how much of the position survey has been completed. In Figure 10, we see that it is 100 % complete.

The satellites card shows the number of satellites being tracked and the average signal to noise ratio of the detected GPS signal for each of the satellites.

The receiver status card indicates items that might affect the quality of the information collected. In the example, we see that a leap second is pending.

Finally, the configuration card allows navigation to screens which configure the operation of the receiver and the alignment of its timing pulses.

## Communications

The top level communications display provides configuration and status information for the RS232 and Ethernet LAN remote interfaces. It is composed of four virtual cards: activity, RS232 configuration, Ethernet LAN configuration, and TCP/IP configuration. A sample display is shown in Figure 11.

Communications	
<b>Activity</b> Interface rs232_0 Errors No errors	<b>RS232 Config</b> Baud 115200 Frame 8/N/1 Flow RTS/CTS
<b>Ethernet</b> MAC 00:19:b3:0b:00:06 Status Disconnected Speed 100 Base T	<b>TCP/IP Config</b> Method Static IP IP Addr 0.0.0.0

**Figure 11: Top level communications display**

The activity card shows which interface was most recently active, and the number of errors detected on that interface.

The RS232 configuration card shows the current configuration for the port, including the baud rate for sending and receiving data.

The Ethernet card shows the MAC address for the FS740 as well as whether the FS740 is currently connected to the LAN. Finally, it shows the configured speed of the connection.

The TCP/IP configuration card shows the current IP address of the FS740 and the method used to obtain that IP address.

## System

The top level system display provides configuration and status information for the FS740 as a whole. It is composed of five virtual cards: user settings, display, device info, alarm status, and rear options. A sample display is shown in Figure 12.

System configuration	
<b>User settings</b> SAVE RECALL	<b>Alarm status</b> Inactive
<b>Display</b> Pwr down... Never	<b>Rear options</b> None installed
<b>Device info</b> SN_000006	

**Figure 12: Top level system display**

The user settings card enables the user to save and recall instrument settings from nonvolatile memory. Up to nine different user settings may be saved.

The display card controls the duration of front panel inactivity after which the display is powered down. This period of inactivity may range from 10 minutes to never.

The device info card navigates to detailed information on serial number, installed options, the Ethernet MAC address, and firmware versions installed on this device.

The alarm status card shows the current status of the rear panel alarm and navigates to options for configuring the alarm.

Finally, the rear options card navigates to information and configuration of any installed rear option boards.

## Sine output

The top level sine output display provides configuration information for the sine output. It enables the user to set the amplitude, frequency and phase of the output. A sample display is shown in Figure 13.

Sine output	
Amplitude	1.000 V <sub>pp</sub>
Frequency	10.000 000 000 MHz
Phase	0.000 Deg

Figure 13: Top level sine output display

## Aux output

The top level aux output display provides configuration information for the aux output. It enables the user to set the waveform, amplitude, frequency and phase of the output. A sample display is shown in Figure 14.

Aux output	
Config	Sine
Amplitude	1.000 V <sub>pp</sub>
Frequency	1.000 000 000 MHz
Phase	0.000 Deg

Figure 14: Top level aux output display

## Pulse output

The top level pulse output display provides configuration information for the pulse output. It enables the user to set the frequency and duty cycle, or period and width of the output. It also allows the user to adjust the phase of the output. A sample display is shown in Figure 15.

Pulse output	
Config	Period / Width
Period	1.000 000 000 000 s
Width	500.000 000 000 ms
Phase	0.000 Deg

Figure 15: Top level pulse output display

## Measurements

The top level measurement display provides configuration information for the front and rear measurement inputs. It enables the user to make time and frequency measurements on the front and rear inputs and view the results. A sample display is shown in Figure 16.

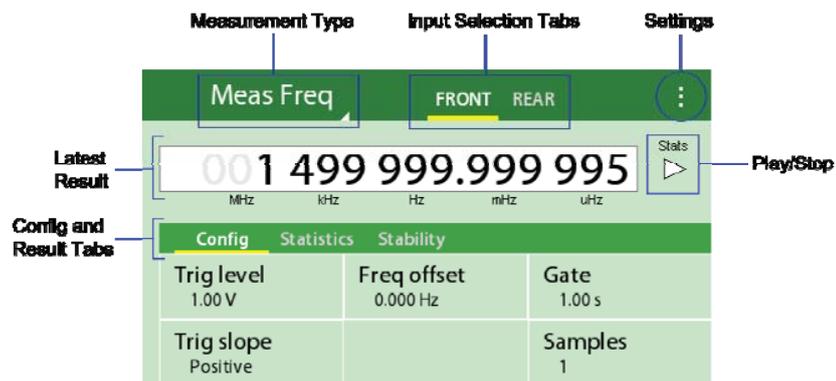


Figure 16: Top level measurement display

The title bar identifies the current measurement. In this case it is a frequency measurement. The measurement type can be easily changed by pressing the current measurement type and then selecting a new one. Next to the title are two tabs: front and rear. In the figure, the yellow bar indicates that the front input is selected. The user may easily switch tabs by pressing the desired input. Immediately below the title bar is the most recent frequency measurement. Below this is another bar of tabs which display configuration and results. In the figure, the yellow bar indicates that configuration data is being displayed. The user may press the Statistics or Stability tabs to display relevant measurement results in those areas.

Finally, in local mode, frequency measurements are automatically triggered and displayed as they become available. However, these automatically triggered results are only visible from the front panel and cannot be retrieved over the remote interfaces. Nor do they contribute to statistics or stability results. To do that the user must explicitly initiate a measurement by pressing the Play/Stop button or by requesting it over the remote interface with an INITiate command or something similar. When a measurement is explicitly initiated, the FS740 will make the requested number of measurements, and calculate statistics and stability based on the results. Up to 250,000 results can be stored in internal memory for retrieval over the remote interfaces.

The Settings button (three vertical dots) at the far right of the title bar can be used to enable or disable automatic triggering of measurements and adjust the measurement timeout duration. The latter may need adjustment if the signal frequency is below 1 Hz.

## Front Panel Connectors

The front panel contains four BNC connectors labeled: Sine out, Aux out, Pulse out, and Time/Freq. The first three are source outputs and the fourth is a measurement input.

### Sine Out

The sine output provides the user with a Sine wave output whose frequency can vary from 1 mHz to 30.1 MHz. Its amplitude can vary from 10 mV<sub>PP</sub> to 1.414 V<sub>PP</sub>. The user may control the phase of this output and align its zero crossing to coincide with the FS740's best estimate of UTC.

### Aux Out

The aux output provides the user with several possible waveforms with user selectable frequencies. These include, a sine wave, a triangle wave, a square wave centered about zero volts, a 100 MHz sine wave, and an AM modulated IRIG-B output. The sine wave supports frequencies from 1 mHz to 10

### Pulse Out

The pulse output provides the user with a CMOS output at a configurable frequency and duty cycle from 1 mHz to 25 MHz. The phase of the pulse output can be adjusted by the user and synchronized to UTC if desired. The output may also be configured to generate pulse modulated IRIG-B for use in time distribution.

### TIME / FREQ

The time and frequency input enables the user to make time or frequency measurements of an applied signal. The user can set the trigger level and slope of the input. Frequencies up to 200 MHz can be measured with gate times from 10 ms to 1000 s supported. Timing measurements enable the user to time stamp a signal relative to the FS740's best estimate of UTC with <50 ps jitter.

## Rear-Panel Overview

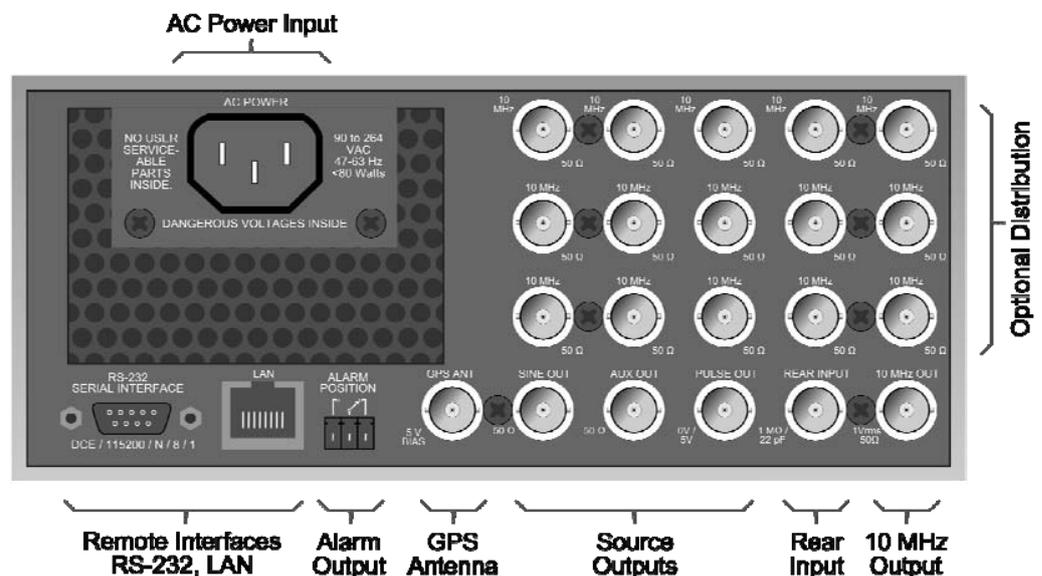


Figure 17: The FS740 Rear Panel

The rear panel provides connectors for AC power, remote computer interfaces, a switch tied to the system alarm, a GPS antenna input, replicas of the front panel source outputs, an independent rear panel measurement input, and a 10 MHz output. Space is available for the installation of 3 optional distribution boards, each with 5 connectors a piece. The optional boards may be used to expand the distribution capability of the instrument. Distribution is available for 10 MHz, Sine/Aux outputs, and Pulse outputs.

## AC Power

Connect the unit to a power source through the power cord provided with the instrument. The center pin is connected to the chassis so that the entire box is earth grounded. The unit will operate with an AC input from 90 to 264 V, and with a frequency of 47 to 63 Hz. The instrument requires 80W and implements power factor correction. Connect only to a properly grounded outlet. Consult an electrician if necessary.

## Remote Interfaces

The instruments support remote control via RS-232, or an Ethernet local area network (LAN). A computer can perform any operation that is accessible from the front panel. Programming the instrument is discussed in the Remote Programming chapter.

### RS-232

The RS-232 port uses a standard 9 pin, female, subminiature-D connector. It is configured as a DCE and supports baud rates from 4.8 kb/s to 115 kb/s. The remaining communication parameters are fixed at 8 Data bits, 1 Stop bit, No Parity, with RTS/CTS hardware flow control.

## Ethernet

The Ethernet uses a standard RJ-45 connector to connect to a local area network (LAN) using standard Category-5 or Category-6 cable. It supports both 10 and 100 Base-T Ethernet connections and a variety of TCP/IP configuration methods.

## Alarm Relay

This connector is tied to a 3A, SPDT switch, with normally open and normally closed connections. The alarm may be manually controlled, or configured to assert when in holdover or the timing error is large.

## GPS Antenna

In order to lock to GPS, the FS740 must be connected to a GPS antenna. The FS740 provides 5V power on the antenna input to support active antennas with more gain. The FS740 tries to detect fault conditions related to the antenna to alert the user of potential problems. If the current draw is too large, an antenna short fault is reported. Alternately, if the current draw is too small, an antenna open fault is reported.

## Rear Measurement Input

The FS740 provides a rear input for making time and frequency measurements that has completely independent measurement circuitry, which can function in parallel with the front panel input.

## Source Outputs

### Sine Out

This is a duplicate of the front panel sine output. Its phase cannot be independently modified relative to the front panel output.

### Aux Out

This is a duplicate of the front panel aux output. Its phase cannot be independently modified relative to the front panel output.

### Pulse OUT

This is a duplicate of the front panel pulse output. Its phase cannot be independently modified relative to the front panel output.

### 10 MHz OUT

The instrument also provides a 10 MHz output for referencing other instrumentation to the FS740's internal reference. Its phase cannot be synchronized to UTC.

## Option Boards

The rear panel has space for the installation of up to 3 option boards for expanded distribution. Three different types of option boards are offered: Sine/Aux distribution, Pulse distribution, and 10 MHz distribution. The user may mix and match the installed options to meet application needs. Each board provides 5 buffered outputs of the desired signal. The Sine/Aux distribution options may be dynamically configured to provide 5 buffered duplicates of the Sine output, or 5 buffered duplicates of the Aux output. If

multiple Sine/Aux distribution option boards are installed, each one may be configured independently of the others. The other option boards have fixed configurations.



# Operation

---

## Introduction

The previous chapter provided an overview of the instrument's features and layout. This chapter describes the operation of the FS740 in detail and provides step by step instructions for setting up and configuring the FS740.

### GPS

One of the central features of the FS740 is its ability to lock its internal reference to GPS. The Global Positioning System, or GPS, is a radio-navigation system which allows users with a clear view of the sky to identify their current position and time of day from any location around the globe. The system was originally designed for the military, but has been used for a wide variety of civil applications since its deployment, particularly in automobile and marine navigation. The system is managed and controlled by the United States Air Force (USAF). It consists of three parts: a space segment, a ground-based control segment, and the user segment.

The space segment consists of 24 satellites orbiting earth in 12 hour orbits at an altitude of approximately 20,200 km. The satellites are arranged into 6 equally spaced orbital planes with 4 satellites allocated to each plane. The organization is designed so that a user will always be able to view at least 4 satellites from virtually any location around the world. Atomic clocks on board the satellites maintain precise time of day information. The satellites then broadcast ephemeris and time of day information to the user segment. A user with a view of at least 4 satellites can use this information to triangulate their position and time.

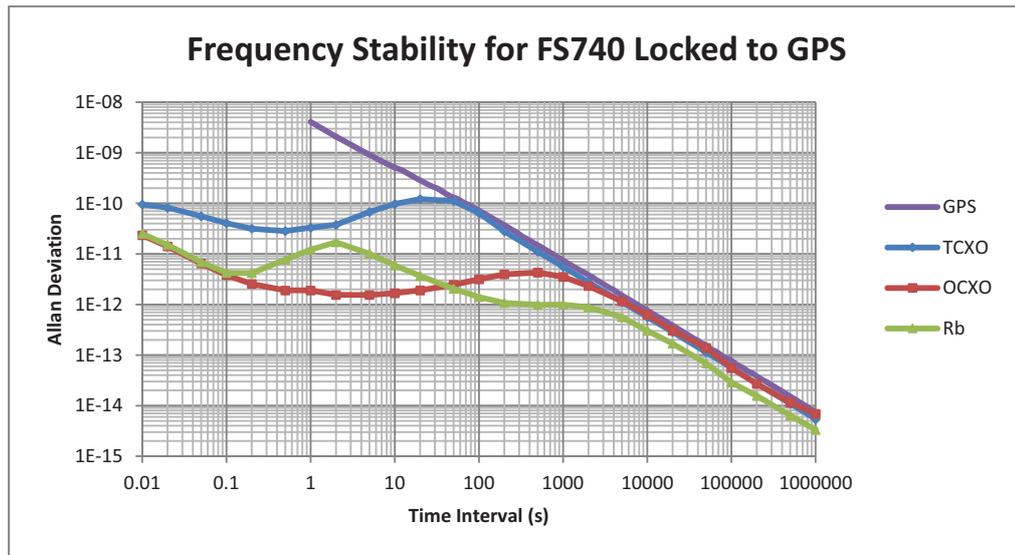
The control segment monitors the health of the satellites and uploads corrections and updates to the navigational information stored in the satellites. It is also in charge of decommissioning satellites when they have reached end of life, and the commissioning of new satellites to replace them. It consists of a master control station and a number of antennas and monitoring sites spread throughout the world.

The user segment consists of an antenna and a receiver which extracts the signals broadcast by the satellites. If the user can track at least 4 satellites, it has enough information to triangulate its position (latitude, longitude, height), and time of day.

Position can be resolved to a few meters. Time of day can be resolved to a few nanoseconds. For the FS740, it is this latter metric which is of significant importance. GPS time is monitored and maintained by a network of atomic clocks on the ground and aboard the satellites which are collectively steered to follow universal coordinated time (UTC) as maintained by the United States Naval Observatory (USNO). Thus, by locking its internal timebase to GPS, the FS740 can provide long term frequency stability on par with the best time keeping instrumentation in world, but at a fraction of the cost.

## Timebase Stability

While the long term stability of GPS is excellent, its short term stability is rather poor in comparison to modern oscillators. The FS740 may be configured at the time of purchase with one of three different oscillators: a TCXO, an OCXO, or a Rb atomic frequency standard. Each oscillator has a different short term stability profile and offers a trade off in performance and holdover capability. Figure 18 shows the typical measured stability of the FS740 locked to GPS for each of the different timebases. The stability of GPS is also shown for comparison purposes.



**Figure 18: Frequency stability of FS740 with different timebases installed.**

Notice that all three timebases offer superior stability to that of GPS at a 1 second time interval. Here the OCXO offers the best performance. For time intervals beyond about 50 s the Rb oscillator outperforms. And for time intervals beyond about 5,000 s, all three timebases follow the performance of GPS, with the Rb oscillator slightly outperforming the other timebases due to its ability to effectively average the GPS signal over a longer time interval.

The differences between the three oscillators are more striking when you look at their stability when not locked to GPS. This indicates the performance of the FS740 if the signal to GPS is lost. Here, the superiority of the Rb oscillator over the other timebases is clearly evident in that its stability surpasses GPS out to 10,000 seconds where its stability exceeds that of the OCXO by more than a factor of 10.

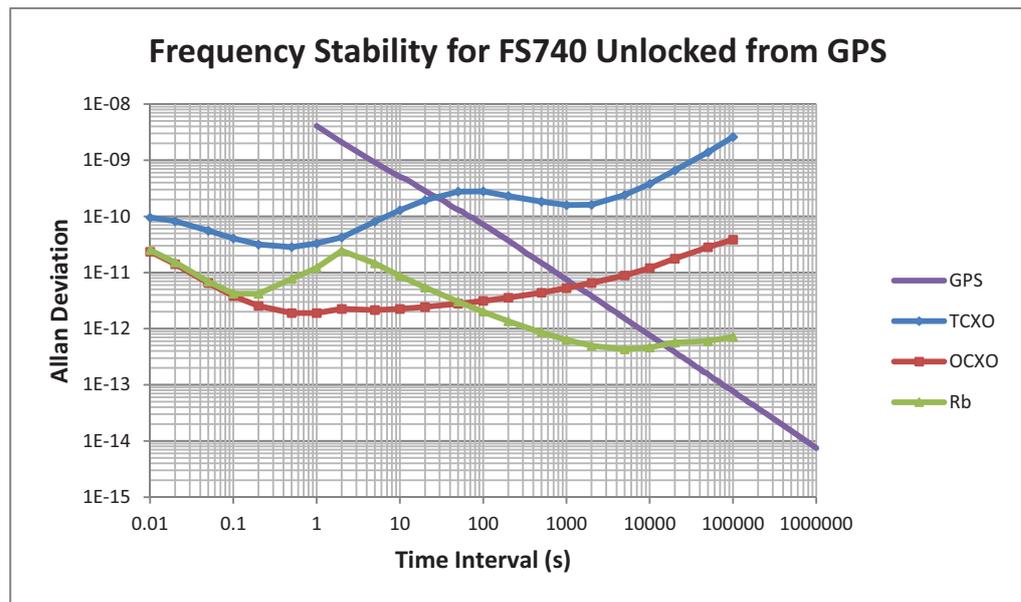


Figure 19: Frequency stability of FS740 when unlocked from GPS.

## Phase Wander vs UTC

Frequency stability, of course, is not the only performance criteria to consider. Another metric to consider is the wander of the phase of the FS740 when it is locked to GPS relative to UTC. The wander in phase relative to UTC is a function of the quality of the GPS receiver and how tightly the oscillator is locked to GPS. At short loop time constants the FS740 faithfully follows the performance of the receiver. At long loop time constants the FS740 will wander further away from UTC, but display much better short term stability.

## Loop Time Constants

The differences in stability of the three timebases affect the optimum time constant with which it is locked to GPS. If the time constant is too small, the short term stability is degraded by GPS. If the time constant is too large, the long term stability will underperform raw GPS. We want to choose a time constant that provides the best stability over all time scales. Stability is not the only concern, however. The accuracy of UTC must also be considered. Long time constants may improve the short term stability of Rb, for example, but they significantly degrade the accuracy of the FS740's estimate of UTC. Thus, the selection of the loop time constant represents a tradeoff between several metrics of performance.

## Timebase

The top level timebase display provides configuration and status information for the internal timebase. It is composed of four virtual cards: timebase status, timebase events, frequency control, and configuration. A sample display is shown in Figure 20.



Figure 20: Top level timebase display

## State

For assessing timebase stability, the most important information to consider is the current state of the timebase, which is shown at the top of the timebase card. In the figure, the timebase is “Locked to GPS,” meaning that the timebase is tracking and following the timing provided by GPS. The state of the timebase may take one of the following values detailed in Table 1.

Table 1: Enumeration of timebase states

State	Meaning
Power up	Device was recently powered on. Timebase is still warming up.
Searching for GPS	Device has warmed up, but no GPS satellites have been found, yet.
Stabilizing	GPS satellites have been found. Making sure timebase is stable enough to lock to GPS.
Validating time	Verifying that the time of day extracted from GPS is stable and consistent. The FS740 instrument time is set at exit of this state.
Locked to GPS	Timebase is locked to GPS.
Holdover, manual	Timebase is not locked to GPS at the user’s request.
Holdover, no GPS	Timebase is not locked to GPS because the receiver is not generating timing pulses. This is usually because no satellites are being tracked.
Holdover, timing	Timebase is not locked to GPS because the timing error between the timebase and GPS is too large.
Rb unlocked	Timebase is not locked to GPS because the installed Rb timebase is unlocked.

A simplified state diagram for the timebase showing the most common transitions is presented in Figure 21. At power on, the timebase always starts in “Power up.” Then, as conditions merit, the timebase moves to succeeding states one by one until the time of day is received from the satellites and lock is achieved. Once locked, timing and status are monitored and if lock is no longer possible the time base will jump to one of the holdover states. When conditions improve, the timebase will jump back to the locked state.

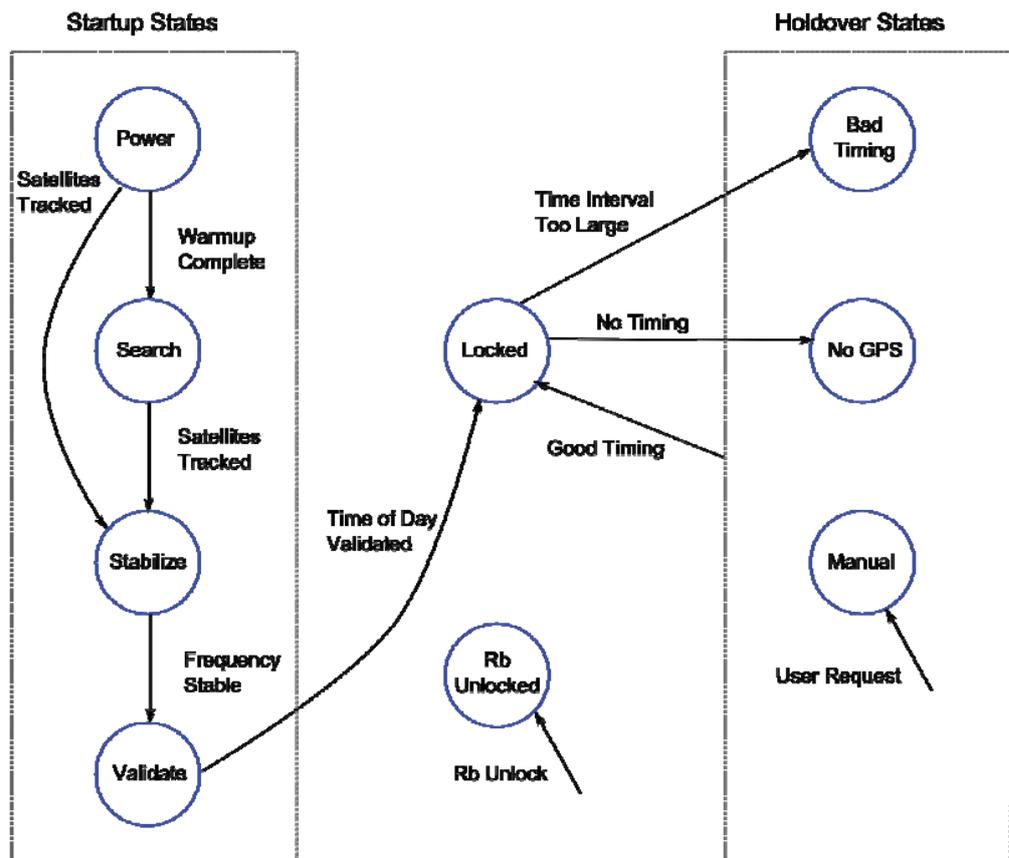


Figure 21: A simplified timebase state diagram showing the most common transitions.

### Related Commands

TBAsE:STATe?

### Duration

In Figure 20, listed directly below the timebase state is the duration. This informs the user the amount of time the FS740 has been in one of three broad categories: startup, lock, or holdover. In Figure 21, boxes are drawn around the start up and holdover states. Startup refers to all states the timebase might traverse before the time of day of the instrument is set. Holdover refers to all states where the timebase has unlocked from GPS after the time of day of the instrument has been set. Lock, of course, refers to when the instrument is locked to GPS, identified by the single state, Locked.

### Related Commands

TBAsE:STATe:HOLDOVer:DURation?  
 TBAsE:STATe:LOCK:DURation?  
 TBAsE:STATe:WARMUp:DURation?

### Phase Locking to GPS

The phase lock loop used by the FS740 to lock its timebase to GPS can be described by the diagram in Figure 22.

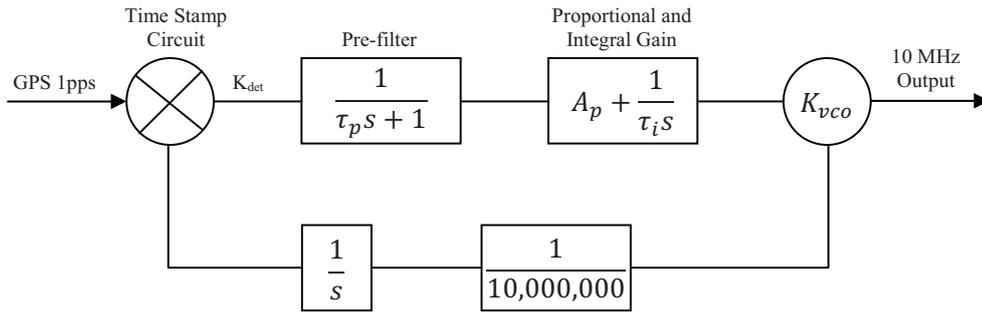


Figure 22: GPS phase lock loop

The symbols in the diagram have the following definitions:

- s Laplace frequency
- $A_p$  Proportional gain
- $\tau_i$  Integral time constant
- $\tau_p$  Pre-filter time constant
- $K_{det}$  Phase detector gain
- $K_{vco}$  VCO gain

The basic architecture is that of a 2<sup>nd</sup> order phase lock loop with proportional and integral gain. This basic loop is then augmented with the insertion of a pre-filter designed to reduce the loop’s sensitivity to the broadband noise of GPS. Given the definitions above, we define the natural loop time constant,  $\tau_n$ , and use it to compute the following variables in the phase lock loop:

$$A_p = \frac{2}{K_{det}K_{vco}\tau_n}$$

$$\tau_i = \tau_n^2 K_{det}K_{vco}$$

$$\tau_p = \frac{\tau_n}{6}$$

Given these definitions and ignoring the pre-filter for the moment, the loop would have the following response to a step in frequency and phase:

$$\Delta T(t) = t[F_0 - \Delta T(0)/\tau_n]e^{-\frac{t}{\tau_n}} + \Delta T(0)e^{-\frac{t}{\tau_n}}$$

with the following symbol definitions:

- t Time
- $\Delta T(t)$  The phase deviation as a function of time
- $\Delta T(0)$  The initial phase deviation at time  $t = 0$
- $F_0$  The initial frequency offset at time  $t = 0$
- $\tau_n$  The natural loop time constant for the phase lock loop

The insertion of the pre-filter perturbs this solution slightly, but the overall response is very similar. The speed with which the FS740 follows GPS is controlled by the natural loop time constant,  $\tau_n$ . By shortening the time constant, the FS740 will follow GPS more faithfully, including the broadband noise of GPS. By extending the time constant

to longer time scales, the FS740 will continue to follow GPS over the long term, but rely on its own stability for short term performance. The goal is to select a time constant that reflects the best balance between short and long term performance for the loop.

## Optimum Time Constant

The FS740 may be configured at the time of purchase with one of three different timebases: a TCXO, an OCXO, and a Rb atomic frequency standard. The optimum time constant for locking each of these timebases to GPS is different, of course, and detailed in Table 2.

**Table 2: Optimum loop time constants**

Timebase	Optimum loop time constant
TCXO	30 s
OCXO	500 s
Rb	4000 s

Although these are the optimum time constants when everything has stabilized, they are not ideal during warm-up or when the FS740 first locks to GPS. During these times it is preferable to have a short time constant so that the FS740 may quickly align its phase and frequency with GPS. As such FS740 will automatically reduce the time constant to as little as 3 s so that it may quickly align its phase to that of GPS. Once adequate phase alignment has been achieved, the time constant will be gradually increased to the optimum time constant detailed in Table 2. Optimum frequency stability is not achieved, however, until the time constant reaches the optimum time constant.

## Current Time Constant and Timing Error

In Figure 20, listed directly below the duration, is the current loop time constant for the phase lock loop, the most recent timing error, and the average timing error of the timebase relative to GPS. Positive errors indicate the timebase lags GPS. Negative errors indicate the timebase leads GPS. The most recent error represents the output of the time stamp circuit in Figure 22. The average error represents the output of the pre-filter.

## Related Commands

TBAsE:TcONstant  
TBAsE:TiNteRval

## Installed Timebase

In Figure 20, listed at the bottom of the Timebase card is the installed timebase. In the figure the timebase is a Rb atomic frequency standard.

## Related Commands

\*OPT?

## Timebase Events

The FS740 automatically captures each time the state of the timebase changes as an event. Every time the timebase either locks or unlocks from GPS, for instance, the event will be logged. These events may be viewed one by one on the Timebase events card.

The FS740 has room to store up to 10 events. If more than 10 events occur, the oldest event is discarded to make room.

The event queue provides the user with automatic self monitoring capability. The user need not continually query the state of the timebase. Rather, he or she can view the event log to see if anything happened over the time period of interest. By comparing time stamps, one can easily deduce how long the timebase spent in each state.

### Related Commands

```
TBASE:EVENT:CLEAr
TBASE:EVENT:COUNT
TBASE:EVENT:NEXT?
```

## Frequency Control

The frequency control voltage is the voltage applied to the frequency input of the timebase to maintain lock to GPS. It is updated every second by the digital phase lock loop when the receiver is generating timing pulses. Every 24 hours of phase lock to GPS the FS740 will automatically save the current frequency control value so that the timebase will power up at a value close to optimum if GPS is not present.

When the timebase is locked to GPS, the frequency control value cannot be changed by the user. If the timebase is not locked to GPS, however, the user may change the frequency of the timebase by updating this voltage. The user may force the FS740 to unlock from GPS, if necessary, to make this possible. See section Lock to GPS on page 29 for details.

### Related Commands

```
TBASE:FCONTROL
TBASE:FCONTROL:SAVe
```

## Configuration

By default the FS740 will initially lock to GPS with a short time constant and then gradually extend the time constant to the optimum time constant for the given timebase when phase alignment is achieved. The user may modify this default configuration by pressing the Config timebase card located at the lower right portion of the Timebase screen as shown in Figure 20. This should lead to the Timebase configuration display as shown in Figure 23.

The user may enable or disable lock to GPS, adjust the bandwidth or loop time constant of the phase lock loop and stipulate the conditions under which the phase lock loop will enter and leave holdover.

← Timebase configuration	
Lock to GPS Enabled	Enter holdover if Time error > 1.000 us
Loop bandwidth Auto	Leave holdover by Waiting for good 1 pps
Loop time constant 30 s	

Figure 23: Timebase configuration display

## Lock to GPS

By default the FS740 will lock to GPS when satellites are being tracked. If the user prefers manual control of the frequency, he should choose to disable lock to GPS by pressing the “Lock to GPS” card and choosing “Disabled.”

When lock to GPS is disabled, the FS740 will still attempt to retrieve the time of day from the GPS satellites. However, after the instrument time of day has been set, the timebase will enter manual holdover and it will not attempt to maintain phase alignment thereafter. Rather, its frequency will be maintained at the last known saved good value.

The user may also adjust the frequency calibration of the FS740 by modifying frequency control voltage. See section Frequency Control on page 28 for information on how to modify the frequency control voltage.

### Related Commands

TBAsE:CONFig:LOCK

## Loop Bandwidth

By default, the FS740 will automatically throttle the bandwidth with which it locks to GPS in order to maintain good phase alignment with GPS. When good phase alignment has been achieved and the timebase frequency is stable, the FS740 will automatically reduce the bandwidth to improve its overall frequency stability.

Advanced users may want to specify the bandwidth or loop time constant with which the FS740 locks to GPS. To do this press the “Loop Bandwidth” card and select “Manual.” This will enable the “Loop time constant” card so that a manual time constant can be selected.

### Related Commands

TBAsE:CONFig:BWIDth

## Loop Time Constant

The “Loop time constant” card is only enabled when manual loop bandwidth control is selected. See section Loop Bandwidth above for details. This setting enables the user to specify the phase lock loop time constant with which the FS740 locks to GPS. To change the time constant, press the “Loop time constant” card and enter a new time constant in seconds. See section Phase Locking to GPS on page 25 for details on the phase lock loop used by the FS740 to lock to GPS and the optimum time constants for each timebase. Short time constants force the FS740 to faithfully follow the phase of GPS including its noise. Longer time constants improve the short term stability of the FS740 but also may increase the wander of its estimate of UTC.

### Related Commands

TBAsE:TCONstant

## Criteria to Enter Holdover

When locked to GPS the FS740 will try to adjust its frequency to maintain good phase alignment with GPS. Under normal conditions this is preferred. Under abnormal conditions, however, this may no longer be desirable, particularly if the measured timing error is large.

There are two scenarios worth considering. In the more common scenario the FS740 may lose the GPS signal for an extended period of time, during which the phase of the FS740 will drift away from alignment. At some point the FS740 may reacquire the GPS signal and see that its phase has drifted significantly away from alignment. How should the FS740 recover? Should it slew the phase back by adjusting its frequency, or should it jump phase instantaneously to the new location?

The second scenario to consider is when the GPS receiver generates rogue timing pulses due to processing incorrect timing information from a failing satellite, having timing data corrupted by noise, or possibly a firmware bug within the GPS receiver itself. In this scenario the GPS timing pulses may suddenly jump in phase. Again the user must decide how the FS740 should recover.

The decision can be broken down into two parts. The first, to be decided here, is the maximum measured timing error allowed before the FS740 will abandon lock and enter holdover. The default is 1  $\mu$ s. To modify this limit, press the card “Enter holdover if” and enter a new limit.

The second part, to be decided in the next section, is how to recover from holdover.

### Related Commands

TBAsE:CONFig:TINTerval:LIMit

## Criteria to Leave Holdover

The FS740 may enter holdover for a number of reasons, the most common of which is that it lost the GPS signal for a time. When it regains the GPS signal, how should it respond? If the timing error is less than the threshold set above, then the FS740 will proceed as if it never lost lock. If the timing error is greater than the threshold set above, then the FS740 has three options:

1. Wait for the timing error to fall below the threshold and then proceed as if it never lost lock.
2. Jump phase instantaneously to the new location and resume lock.
3. Momentarily ignore the threshold and slew the phase to the new location by adjusting the frequency of the timebase.

Option 1 may be preferred for the case where the receiver outputs rogue pulses for a time and then later recovers. Option 2 may be preferred if the user is primarily concerned about frequency calibration and doesn't want the frequency to be disturbed, merely because the phase is out of alignment. Option 3 may be preferred if the user needs phase alignment but instrumentation connected to the FS740 cannot handle a sudden hiccup in phase.

To make a selection, press the “Leave holdover by” card and select the desired behavior. The display will look similar to that shown in Figure 24.



Figure 24: Selecting the criteria for leaving holdover.

## Related Commands

TBAsE:CONFig:HMODe

## Timebase Status LEDs

Status LEDs in the reference section of the front panel provide continuous information on the current state of the timebase. The three LEDs above the timebase button are specific to the timebase, as highlighted in Figure 25.

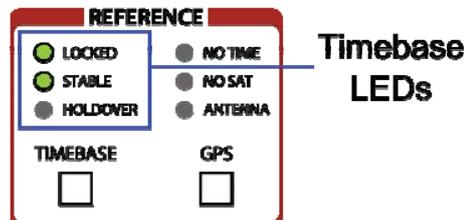


Figure 25: Timebase status LEDs.

The first LED labeled “LOCKED” indicates whether the internal timebase is locked to GPS. The second LED labeled “STABLE” indicates whether the internal timebase has reached optimum stability. Optimum stability is only reached when the timing error is small and the loop time constant has reached the optimum time constant for the installed timebase. The last LED labeled “HOLDOVER” indicates when the FS740 has lost lock from GPS and is in holdover.

## GPS

The top level GPS display provides configuration and status information for the FS740’s GPS receiver. It is composed of five virtual cards: time of day, position, satellites, receiver status, and configuration. A sample display is shown in Figure 26.



Figure 26: Top level GPS display

## Date and Time

The “Time” card shows the current date and time. At power on the time is set to January 6, 1980 at midnight, which is the start of GPS time. The displayed time will be marked as “UNSET” until the time of day has been set by GPS. Once the time has been set by GPS, the displayed time can be configured to be UTC, GPS, or local time. In Figure 26, local time is being displayed. Pressing the “Time” card presents the user with a full screen display of the current date and time. A sample display is shown in Figure 27.



Figure 27: GPS time display

## Related Commands

SYSTem:DATE  
SYSTem:TIME

## Position

The “Position” card shows the current surveyed position of the GPS antenna. Latitude is shown in degrees north or south. Longitude is shown in degrees east or west. Altitude is shown in meters above mean sea level. Finally the current progress of the survey of position is shown as a bar graph. Pressing the “Position” card results in the display of more detailed numbers regarding the position as shown in Figure 28

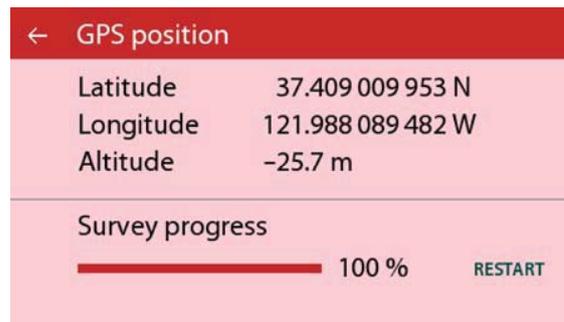


Figure 28: GPS position display

Here the position is displayed with greater precision. The user is also given the opportunity to restart the survey, if desired. This may be desirable if the antenna is ever moved or replaced.

## Related Commands

```
GPS:POSition?
GPS:POSition:SURVey:PROGress?
GPS:POSition:SURVey:STARt
GPS:POSition:SURVey:STATe?
```

## Satellites

The “Satellites” card shows the number of GPS satellites being tracked and the average SNR with which they are being tracked. The receiver has the capability of tracking 32 satellites simultaneously. It is quite rare to have that many satellites visible, however. Pressing the “Satellites” card results in the display of detailed information about the satellites being tracked, including their position in the sky and the signal to noise ratio with which they are being followed. A sample display is shown in Figure 29.

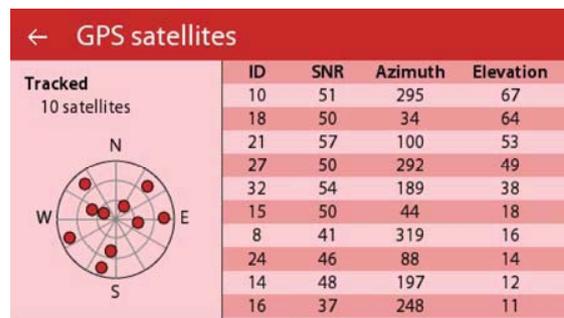


Figure 29: GPS satellites display

## Related Commands

```
GPS:SATEllite:TRACKing?
GPS:SATEllite:TRACKing:STATus?
```

## Receiver Status

The “Receiver status” card shows summary status for the GPS receiver. It indicates if time has been set by GPS, whether satellites are being tracked, whether the antenna is properly connected, whether an almanac has been downloaded, and whether a leap

second is pending. In Figure 26, we see that a leap second is pending. Pressing the “Receiver status” card results in the more detailed display as shown in Figure 30

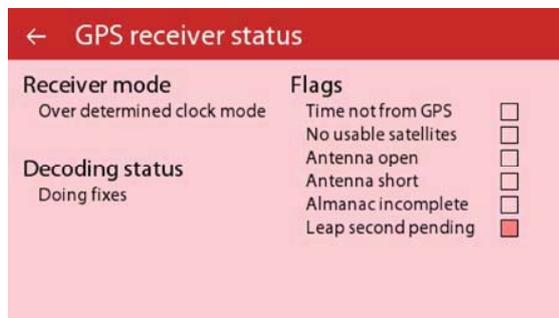


Figure 30: GPS receiver status display

In this display, the meaning of each of the flags is written out for easier comprehension. Additional information about the mode and decoding status of the receiver is also provided. The receiver mode indicates whether position fixes are being computed and if a survey is in progress. In over determined clock mode, position fixes are not being computed. Instead, the surveyed position is being substituted so that all satellites can be dedicated to timing. The decoding status indicates whether a timing fix is being computed or if the receiver is searching for usable satellites.

## Related Commands

STATus:GPS:CONDition?  
STATus:GPS:EVENT?

## Configuration

The “Config” card enables the user to configure the position survey, timing alignment, timing quality, local time offset, and antenna cable delay corrections. Pressing the “Config” card results in the display shown in Figure 31.

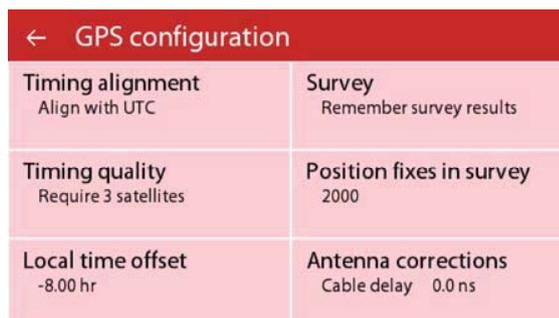


Figure 31: GPS configuration display

Six virtual cards show the current configuration and enable the user to modify the configuration to match their application.

## Timing Alignment

The user can choose to have its timing aligned to GPS time or UTC. Universal coordinated time (UTC) is the official time of day accepted by the international community. It is based on atomic time, but it is occasionally modified with the insertion

of a leap second in order to keep it synchronized to mean solar time. These corrections ensure that the time of sunrise and sunset do not shift over time. GPS time, on the other hand, is strictly atomic time. It is not modified by the insertion of leap seconds. Therefore, as of January 2017, GPS time is 18 seconds ahead of UTC. This difference is called the UTC offset.

Although GPS time is atomic time and is based on a network of cesium atomic standards, just like UTC, it is an independent clock and therefore its phase can deviate from that of UTC. This phase is continuously monitored by the ground control segment of GPS, however, and they can and do steer GPS time to ensure that the phase of GPS does not deviate by more than 1  $\mu$ s from UTC (USNO). In practice, the deviation is usually less than 10 ns. Both the current UTC offset and the current UTC to GPS phase deviation are broadcast by the GPS satellites as part of the almanac.

If the user chooses to align FS740 time to GPS, then both the displayed time of day and the physical phase of the FS740 will be aligned to GPS rather than UTC, and vice versa. The default alignment is to UTC.

## Related Commands

```
GPS:CONFig:ALIGnment
GPS:UTC:OFFSet?
```

## Timing Quality

Normally, GPS receivers must track at least 4 satellites before they can generate a position and timing fix. This stems from the fact that there are 4 unknowns to be solved: latitude, longitude, altitude, and time. 4 equations are required to solve for 4 unknowns. However, if the receiver is not moving, then it need not re-compute its position each time. Rather, it can use the position computed from before, since it is not changing. In this case only 1 satellite is needed to compute time.

In normal operation, the FS740 will automatically survey its position for 2000 seconds to accurately determine its position. After that, the receiver will enter over determined clock mode and all satellites will be dedicated to generating the best possible timing. Only one satellite is required to generate a solution. However, the quality of such a solution increases as more satellites are incorporated into it.

The user has a tradeoff to consider. Should the receiver generate potentially noisy timing pulses when tracking just one satellite? Or should it wait until it is tracking at least three satellites before generating timing pulses. The quality of the timing is significantly better for three satellites, but it also increases the likelihood that the FS740 will be forced into holdover. Noisier timing may be preferred to no timing at all.

When the antenna has a clear view of the sky, emphasizing quality makes sense. When this is not the case, having noisy timing may be preferred over no timing at all. The factory default is 3 satellites.

## Related Commands

```
GPS:CONFig:QUALity
```

## Local Time Offset

The user may prefer to have the local time of day displayed, rather than UTC. To display local time, the user must enter the offset in hours between UTC and the local time. If the offset is zero, UTC or GPS time will be displayed according to the selected timing alignment. When the offset is nonzero, the local version of UTC or GPS time will be displayed.

### Related Commands

SYSTEM:TIME:LOFFset

## Survey

The FS740's GPS receiver provides the best timing performance when the antenna is stationary and not moving. It gets enhanced performance by surveying its position for a time and then using that position to improve the timing. The FS740 support three modes of operation:

1. Disable survey
2. Redo survey at power on
3. Remember previous survey at power on and reuse it

The factory default is to remember the previous survey results at power on. It is most appropriate when the FS740 is installed in a static location and never moved. If the FS740 is static when powered, but may move from one location to another between power cycles, then the survey should be repeated at power on. Finally, the survey should be disabled if the FS740 is installed in a mobile environment where its position might change during operation.

### Related Commands

GPS:CONFig:SURVey:Mode

## Position Fixes in Survey

The FS740 position survey is designed to accurately identify the receiver's current position by averaging together the position fixes computed when the receiver is not moving. By default, the survey will average together 2,000 position fixes to obtain its surveyed position. If desired, the user may change the number of fixes included in the survey.

### Related Commands

GPS:CONFig:SURVey:FIXes

## Antenna Corrections

The FS740's GPS receiver computes position and time based on the position of the antenna. However, the antenna signal must travel through a potentially long cable before it reaches the GPS receiver input on the FS740. Thus, the signal reaching the receiver is delayed compared to the case where the antenna is attached directly to the receiver with no intervening cable. The typical delay for most BNC cables is 1.5417 ns per foot. For a

30 ft cable, the delay would be  $1.5417 \times 30 = 46.25$  ns. If uncorrected, the FS740's estimate of UTC would be 46.25 ns later than a properly calibrated unit.

The user may correct for this delay, however, by entering a negative antenna cable delay correction. The user would correct for a 30 ft cable by entering a correction of  $-46.25$  ns. The correction will be stored in nonvolatile memory and need not be re-entered when the unit is power cycled. When entered correctly, the display will look like that shown in Figure 32.

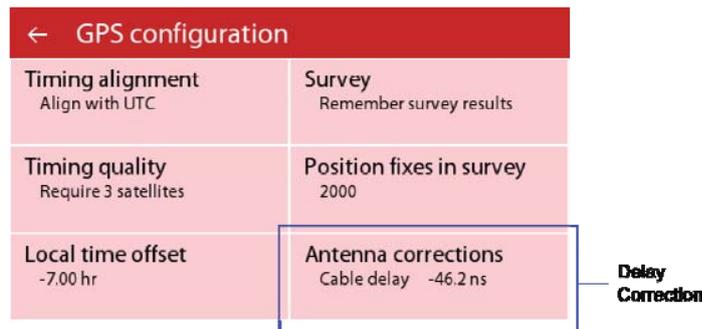


Figure 32: Antenna delay correction

### Related Commands

GPS:CONFig:ADELay

### GPS Status LEDs

Status LEDs in the reference section of the front panel provide continuous information on the current state of the GPS receiver. The three LEDs above the GPS button are specific to the receiver, as highlighted in Figure 33.

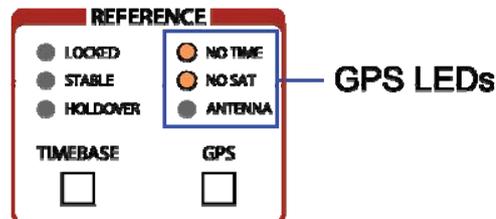


Figure 33: GPS status LEDs.

The first LED labeled “NO TIME” indicates whether the receiver has obtained the time of day from the GPS satellites. The second LED labeled “NO SAT” indicates whether the receiver is successfully tracking GPS satellites. The last LED labeled “ANTENNA” indicates when the FS740 thinks there might be a problem with the antenna connected to the FS740. The LED turns on when the connection appears to be an open or a short circuit. See section Antenna Warning LED on page 2 for more details.

## Communications

The FS740 comes with support to communicate with remote hosts via RS232 and Ethernet. The communications display enables the monitoring and configuration of remote interfaces. The top level communications display provides information on the

most recent remote interface activity and the current configuration of the communication interfaces. A sample display is shown in Figure 34.

Communications	
<b>Activity</b> Interface rs232_0 Errors No errors	<b>RS232 Config</b> Baud 115200 Frame 8/N/1 Flow RTS/CTS
<b>Ethernet</b> MAC 00:19:b3:0b:00:06 Status Disconnected Speed 100 Base T	<b>TCP/IP Config</b> Method Static IP IP Addr 0.0.0.0

Figure 34: Top level communications display

### Remote Mode

When the FS740 receives commands over a remote interface, it will typically enter remote mode. In this mode the front panel interface will be locked out and the orange REMOTE LED in the COMM section of the front panel will be highlighted. Any attempt to interact with the front panel will result in a pop up window notifying the user about remote mode and giving the user the opportunity to return to local mode if desired.

Communications	
<b>Activity</b> Interface rs232_0 Errors No errors	<b>RS232 Config</b> Baud 115200 Frame 8/N/1 Flow RTS/CTS
<b>Ethernet</b> MAC 00:19:b3:0b:00:06 Status Disconnected Speed 100 Base T	<b>TCP/IP Config</b> Method Static IP IP Addr 0.0.0.0

In remote mode. Go to local mode?

GO TO LOCAL

Figure 35: Remote mode pop up window

Press the virtual button GO TO LOCAL to regain access to the front panel user interface.

### Activity

The “Activity” card shows which interface has most recently communicated with instrument. In Figure 34 we see that the RS232 interface was most recently active and that no errors have been detected on that interface. If an error is detected, the red LED in the COMM section of the front panel will be highlighted and the number of errors.

Navigating into the “Activity” card leads to more detailed information about the active interface as shown in Figure 36

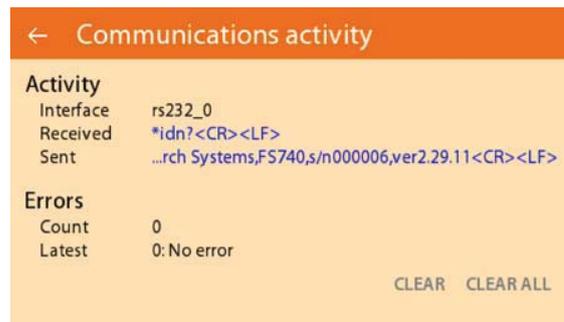


Figure 36: Communications activity display

The active interface and the latest characters received and sent over the interface are displayed. This can be useful when troubleshooting communications and command execution. The user can verify that the FS740 is actually receiving what was sent and that the remote interface is receiving the response. In the figure we see that the FS740 received the \*IDN? command and that it responded with its identification string. The green ACTIVITY LED in the COMM section of the front panel will flash when characters are sent and received. This may also serve as a helpful diagnostic when troubleshooting communication problems.

The number of errors and the latest error are also displayed. Errors may be dismissed one by one by pressing CLEAR. All errors may be flushed by pressing CLEAR ALL.

## Related Commands

SYSTem:ERRor?

## RS-232 Configuration

The FS740 supports communication over an RS-232 port. The rear panel contains a standard 9-pin, female, subminiature-D connector. It is configured as a DCE and supports baud rates from 4,800 to 115,200 baud. The remaining communication parameters are fixed at 8 Data bits, 1 Stop bit, No Parity, with RTS/CTS hardware flow control. The factory default baud rate is 115,200 baud.

The "RS232 Config" card shows the current configuration for the RS-232 port. Navigating into the "RS232 Config" card provides more detailed information and the opportunity to select the desired baud rate for the port. A sample display is shown in Figure 37.



Figure 37: RS232 settings display

Both the FS740 and the remote host must agree on all configuration parameters for communication to succeed. Press “Baud rate” to modify the baud rate as shown in Figure 38.

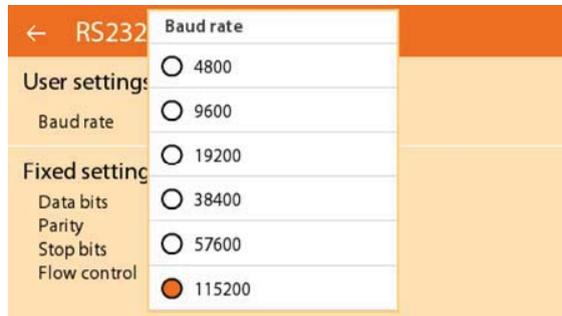


Figure 38: Selecting an RS232 baud rate.

In the figure, the current baud rate is shown as 115200. Select a different baud rate to change it, if desired. If a new baud rate is selected, the interface will be reset and brought back up with the new baud rate configured.

## Related Commands

SYSTem:COMMunicate:SERial:BAUD  
SYSTem:COMMunicate:SERial:RESet

## Ethernet

The FS740 may be connected to an Ethernet base local area network (LAN). The rear panel contains a standard RJ-45 connector which accepts Category-5 or Category-6 cable. It supports both 10 and 100 Base-T Ethernet connection speeds.

### MAC Address

Every device connected to an Ethernet network must have a unique media access address, or MAC address. This address cannot be changed by the user, but it is sometimes useful to know because some networks use the MAC addresses to identify and regulate which devices can connect to the network. The “Ethernet” card displays the MAC address.

### Status

Below the MAC address is the status of the Ethernet connection. This shows whether the FS740 is currently connected to the Ethernet or not. Status will show “Connected” or “Disconnected”

### Speed

The FS740 supports both 10 and 100 Base-T Ethernet connection speeds. The “Ethernet” card shows the current speed. The factory default is 100 Base-T. The user may change this by pressing the “Ethernet” card as shown in Figure 39.

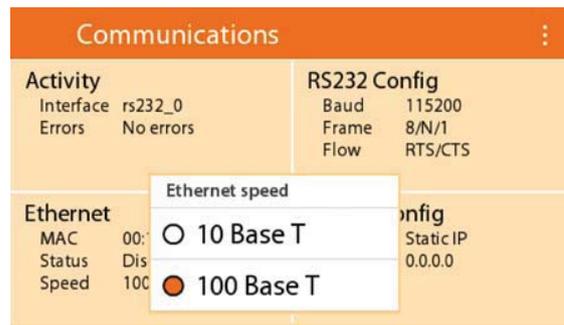


Figure 39: Selecting the Ethernet speed.

In the figure, the Ethernet speed is currently 100 Base-T. Press “10 Base T” to change it.

### Related Commands

SYSTem:COMMunicate:LAN:MAC?  
 SYSTem:COMMunicate:LAN:EPHY?  
 SYSTem:COMMunicate:LAN:EPHY:SPEEd

## TCP/IP Configuration

In order to function properly on a TCP/IP network, the FS740 must be configured with a proper IP address, subnet mask, and gateway or router address. The FS740 supports 3 different methods of obtaining its network configuration: DHCP, Auto IP, and Static IP. The “TCP/IP Config” card shows the method used to obtain the FS740’s current network configuration and its IP address. Full configuration details may be obtained by navigating into the “TCP/IP Config” card. The display will be similar to that of Figure 40.



Figure 40: TCP/IP settings display.

### Configuration Method

The FS740 supports three methods of obtaining its network configuration: DHCP, Auto IP, and Static IP. The user should check with his network administrator to ascertain the preferred method before connecting the FS740 to the network.

Dynamic Host Configuration Protocol (DHCP) is an automated method of obtaining network configuration parameters. In this method the FS740 asks for its configuration from a DHCP server which then responds with its proper network configuration. Many routers support this configuration method. This is the factory default method of configuration.

Auto IP refers to the automated selection of a link local private address in the 169.254.x.x address space. When the FS740 has an IP address in this address space, it cannot communicate with hosts beyond the local network.

Static IP refers to manual configuration. In this method, the user must manually enter the proper network configuration parameters for the FS740 to use.

To select a configuration method, press the current configuration method. The display will be similar to that of Figure 41.

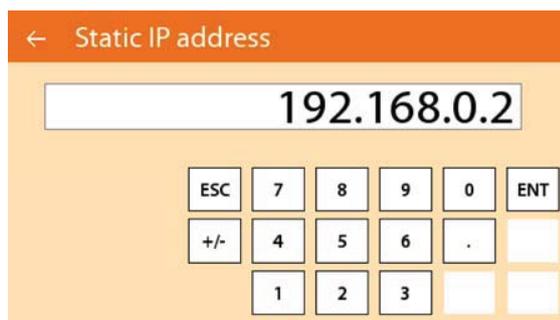


**Figure 41: TCP/IP configuration method.**

In the figure, DHCP is selected. For Auto IP or static configuration, choose “Static IP.” Auto IP is used when “Static IP” is selected and the static IP address is 0.0.0.0. Static IP configuration is used for all other IP addresses. Static IP is also used as a fall back configuration option when DHCP fails.

## Static Configuration

When static IP configuration is selected, the user must explicitly enter the IP address, subnet mask and gateway address. Navigate to the given parameter and enter it using the virtual keypad as shown in Figure 42.



**Figure 42: Entering an IP address.**

IP addresses are composed of 4 numbers in the range from 0 to 255 separated by a decimal point (.). In the figure the IP address “192.168.0.2” is shown. The same procedure is used to enter the subnet mask and gateway address for static configurations.

## Reset

When the IP configuration is changed, the interface should be reset to force the new configuration to take effect. The user can initiate a reset of the interface by pressing the virtual RESET button.

## Current Configuration

The current IP configuration is shown at the top right of the TCP/IP settings display. See Figure 40. Note that the current configuration may differ from the manual configuration parameters entered in the static configuration section, especially if DHCP is used to obtain its address. The current configuration is what the user should refer to when trying to communicate with the FS740 over a TCP/IP connection.

## Related Commands

```

SYSTEM:COMMunicate:LAN:DHCP
SYSTEM:COMMunicate:LAN:IPADress
SYSTEM:COMMunicate:LAN:SMASK
SYSTEM:COMMunicate:LAN:GATeway

```

## Communication Status LEDs

Status LEDs in the communication section of the front panel provide information on the activity of the remote interfaces, as highlighted in Figure 43.

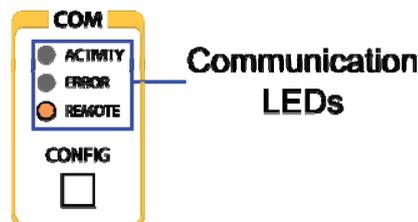


Figure 43: Communication status LEDs.

The first LED labeled “ACTIVITY” flashes when a character is received or sent over one of the remote interfaces. This can be useful when diagnosing communication problems. The second LED labeled “ERROR” indicates that the FS740 did not execute a command due to an error. Details about the error or errors can be found by navigating to Communications > Activity. The last LED indicates if the FS740 is in remote mode. When in remote mode the front panel of the instrument is locked out. All key presses will result in the display of a message indicating that remote control is active and the option to go to local mode. The user must click the “GO TO LOCAL” virtual button to regain front panel control of the instrument.

## System

The top level systems display provides information about alarms, installed options, and the overall configuration of the FS740. A sample display is shown in Figure 44. It enables the user to save and recall user settings, configure the power down of the display, configure the rear panel alarm, configure installed options, and view device information.

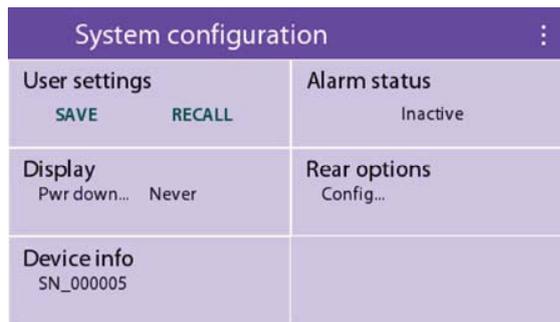


Figure 44: Top level system configuration display.

## Save and Recall User Settings

The user may save and recall up to nine different user settings. The settings are stored in nonvolatile memory, so that they may be accessed even if the unit is power cycled. User settings include the configuration of the source outputs and measurement inputs. System settings related to the Timebase, GPS, and communication configuration are not stored or recalled.

### Save Settings

To save user settings press the virtual button SAVE. The display will be similar to that of Figure 45. Then select a location to which settings should be saved.

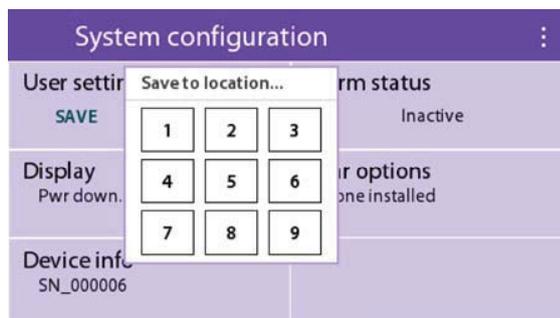


Figure 45: Saving user settings.

### Recall Settings

To recall previously saved settings press the virtual button RECALL. The display will be similar to that of Figure 46. Then select the location from which settings should be recalled.

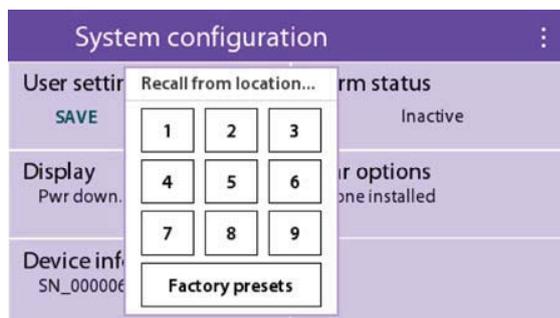


Figure 46: Recalling user settings.

In addition to recalling previously saved settings, the user may alternatively recall factory presets. When this is chosen, user settings are restored to factory default settings. This is similar to executing the command \*RST.

## Related Commands

\*SAV  
\*RCL  
\*RST

## Display

The user can optionally power down the display after a period of inactivity. The factory default is to never power down the display. If the display is powered down, the user need only touch the display to reactivate it. To change the current configuration, press the “Display” card. The display will be similar to that shown in Figure 47.



Figure 47: LCD display settings.

In the figure, the current selection is “Never.” Select a different time period to change the selection. If “Now” is selected, the display will power down immediately, but the previous selection will still be active. For example, if “Never” is the current selection and the user selects to power down the display “Now,” the display will power down immediately. However, “Never” will remain the current selection, so once the display is awakened, it won’t power down again.

## Related Commands

SYSTem:DISPlay:POWer

## Device Information

Navigating into the “Device info” card presents detailed information about the overall system configuration of the FS740. It includes information such as the serial number, the firmware versions instrument and front panel, installed options and timebases, and information about the GPS receiver. The display is similar to that of Figure 48.

← Device info	
Serial Number	SN_000005
Instrument firmware	2.29.11
Front panel firmware	1.30
Ethernet mac	00:19:b3:0b:00:05
Installed timebase	Rb
Option 1	10MHz distribution
Option 2	Pulse distribution
Option 3	Sine/Aux distribution
GPS receiver	Resolution SMTx
GPS firmware	2.2.2.7

Figure 48: Device information display.

## Related Commands

\*IDN?  
\*OPT?

## Alarm

The rear panel of the FS740 contains screw terminals to a SPDT switch actuated by the system alarm. The system alarm may be controlled manually, or it may be configured to assert in response to anomalous conditions, such as detecting the timebase is in holdover, or has drifted more than 100 ns from UTC. The “Alarm status” card shows the current state of the alarm. Navigating into the “Alarm status” card brings up a display similar to that of Figure 52 when the alarm is asserted.

← Alarm	
Status	Causes
<b>ALARM</b>	Time not from GPS
	CLEAR
Mode	Assert alarm when...
Track current condition	Time not from GPS

Figure 49: System alarm display.

The top portion of the display shows the current status of the alarm. When the alarm is asserted, the display also shows what condition caused the alarm to assert. In the figure we see that the alarm is being asserted because time of day has not been extracted from the GPS receiver, yet. When the alarm is in manual mode, the user is also given the option to SET or CLEAR the alarm.

## Related Commands

SYSTem:ALARm?  
SYSTem:ALARm:CONDition?  
SYSTem:ALARm:EVENTt?  
SYSTem:ALARm:CLEAr  
SYSTem:ALARm:FORCe:STATe

## Alarm Mode

The system alarm can operate in one of three modes: tracking, latching, or manual. The user may choose the mode by navigating into the “Mode” card and making the desired selection as shown in Figure 50.

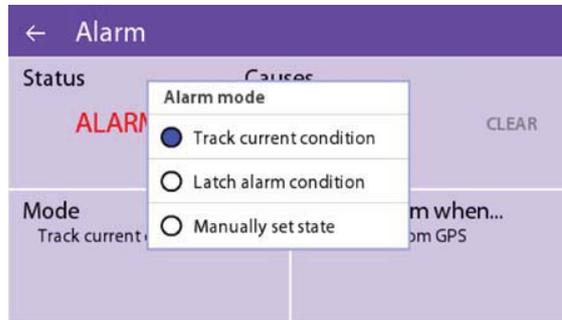


Figure 50: System alarm mode display.

In tracking mode, the alarm is asserted when a configured condition is true, and de-asserted when the configured condition is false. The user cannot forcibly clear the alarm except by changing the mode to manual or removing the condition.

In latching mode, the alarm is asserted when a configured condition is true. Unlike tracking mode, however, it is not de-asserted when the configured condition becomes false. Rather, it remains asserted until the user explicitly clears the alarm. The user will not succeed in clearing the alarm if the configured condition is still true, however, because it will merely be reasserted. In this case the alarm can only be cleared by removing the condition or changing the mode to manual.

In manual mode, the user explicitly sets the state of the alarm. The state can be toggled by pressing the virtual SET and CLEAR buttons.

## Related Commands

SYSTem:ALARm:MODE  
 SYSTem:ALARm:FORCe:STATE

## Alarm Conditions

When the system alarm is in tracking or latching mode, the user must specify the conditions which will trigger the alarm. The “Assert alarm when...” card shows which conditions will trigger an alarm. The user modifies the conditions by navigating into the card and modifying the selections. The display will look similar to that of Figure 51

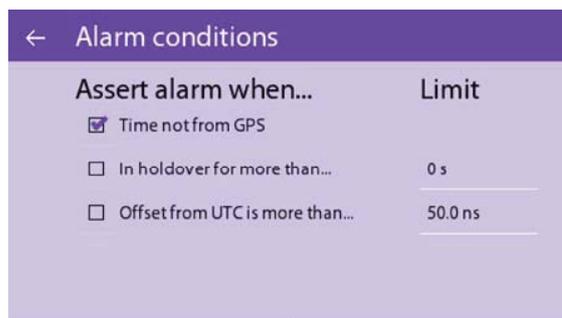


Figure 51: System alarm conditions display.

In the figure, the alarm is asserted when the time of day has not been set by GPS. The user may also set limits for how long the FS740 must be in holdover, and for how large the measured deviation from UTC can be before the alarm is asserted.

### Related Commands

```
SYSTem:ALARm:ENABle
SYSTem:ALARm:GPS:TINTerval
SYSTem:ALARm:HOLDOVer:Duration
```

## Rear Panel Options

The rear panel options display clearly shows which options have been installed and enables the user to configure the Sine/Aux distribution options to output a copy of either the Sine output or the Aux output. The display is similar to that of Figure 52. It clearly indicates what has been installed and gives the user the option to configure the Sine/Aux distribution boards if installed.

← Rear panel options		
Option 1	10MHz distribution	CONFIG
Option 2	Pulse distribution	CONFIG
Option 3	Sine distribution	CONFIG

Figure 52: Rear options configuration.

In the figure, option 3 is currently configured for Sine distribution. This may be changed to Aux distribution by pressing the virtual button CONFIG, and selecting Aux distribution.

### Related Commands

```
ROUTE:OPTion
*OPT?
```

## System Status LEDs

Status LEDs in the system section of the front panel provide information on the state of the alarm and overall system power as highlighted in Figure 53.

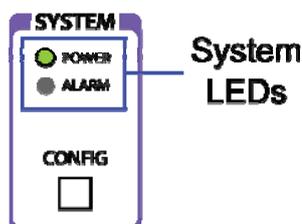


Figure 53: System status LEDs.

The first LED labeled “POWER” should be highlighted whenever power is applied to the instrument. The second LED labeled “ALARM” is highlighted whenever the system alarm is asserted. When lit, navigate to System > Alarm for details on why the alarm has been asserted.

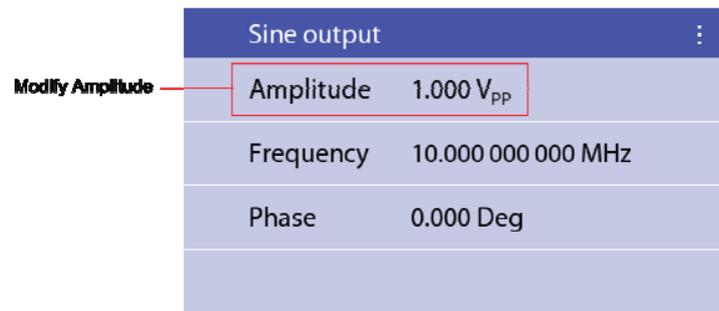
## Source Outputs

The front panel has three source outputs: sine, aux, and pulse. The rear panel provides copies of these outputs and a dedicated 10 MHz output. The user may purchase additional distribution capability for any of these outputs by ordering optional boards installed on the rear panel of the FS740.

The configuration of the individual outputs is similar. The hardware key on the right hand side of the front panel, above the output’s BNC connector, takes the user to the top level display for that output. The user can then modify the amplitude, frequency, and phase of that output.

## Amplitude

The user can adjust the amplitude of the sine and aux outputs. The top level display for the output shows the current amplitude as shown in Figure 54 for the sine output.



Sine output	
Amplitude	1.000 V <sub>pp</sub>
Frequency	10.000 000 000 MHz
Phase	0.000 Deg

Figure 54: Modifying amplitude.

Press the current value to modify it. The amplitude can be adjusted or explicitly changed using the techniques discussed in the section Numeric Parameters on page 9.

## Units

Sinusoidal waveforms may be specified in three different units: V<sub>pp</sub>, V<sub>RMS</sub>, and dBm. Other waveforms may only be specified in V<sub>pp</sub>. The current units may be changed by pressing the UNITS button and selecting the desired units as shown in Figure 55. When entering an amplitude explicitly, use the numeric keypad and complete the entry with the desired units.

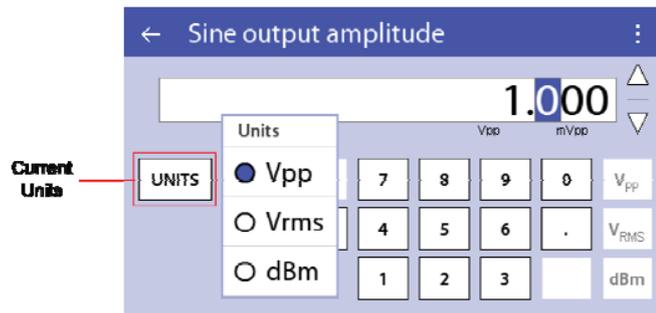


Figure 55: Changing amplitude units.

### Related Commands

SOURce:VOLTage  
 SOURce:VOLTage:UNITs

## Frequency

The top level display for the output shows the current frequency as shown in Figure 56 for the sine output.

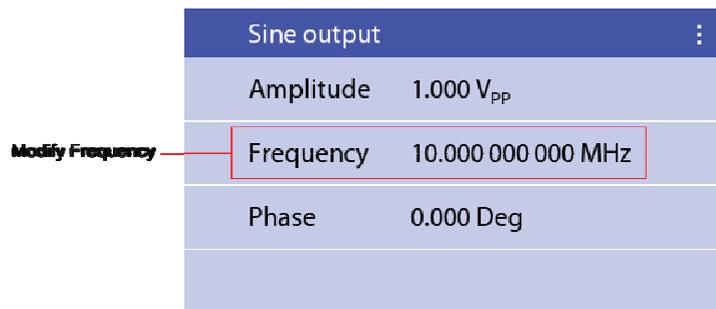


Figure 56: Modifying frequency.

Press the current value to modify it. The frequency can be adjusted or explicitly changed using the techniques discussed in the section Numeric Parameters on page 9.

### Related Commands

SOURce:FREQuency

## Phase

The top level display for the output shows the current phase for the output as shown in Figure 57 for the sine output.

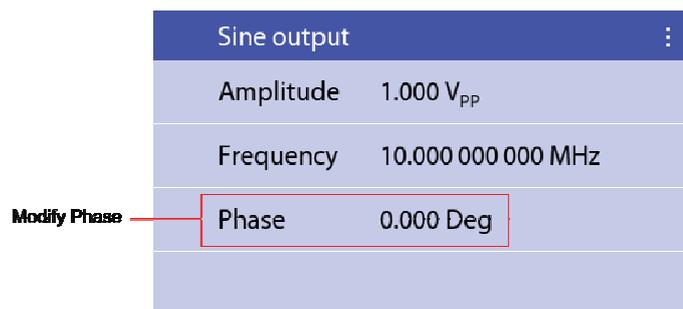


Figure 57: Modifying phase.

Press the current value to modify it. The phase can be adjusted or explicitly changed using the techniques discussed in the section Numeric Parameters on page 9. Phase can be adjusted by up to  $\pm 360$  degrees at a time. After moving the phase the display is renormalized modulo 360 degrees. For example, if the current phase is 350 degrees, the user may legally enter 370 degrees because this is only a phase shift of 20 degrees. After the phase shift, the display will show 10 degrees, because 370 modulo 360 is 10.

## Units

Phase may be specified in degrees, radians, or time. The current units may be changed by pressing the UNITS button and selecting the desired units as shown in Figure 58.

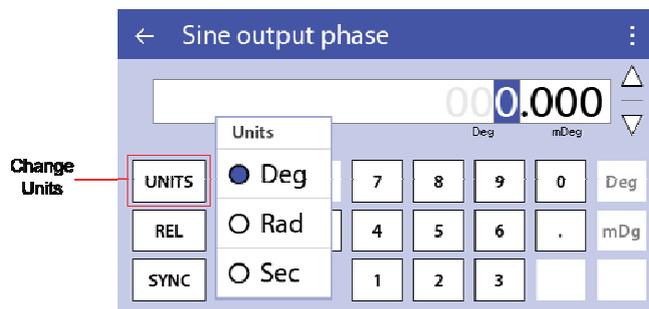


Figure 58: Changing units of phase.

## Relative Phase

Sometimes it is convenient to define the current phase to be zero, without physically changing the phase. This may be accomplished by pressing the REL button as shown in Figure 59.

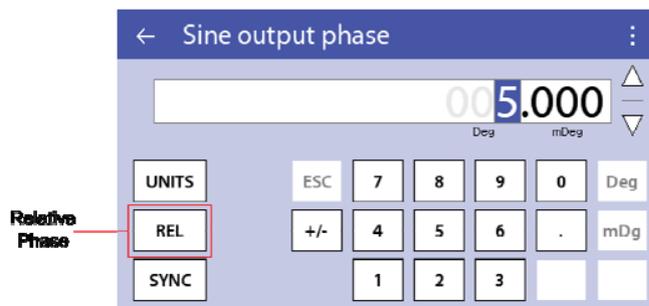


Figure 59: Relative phase.

This is useful when the user wants to align the phase of the FS740 with some external reference phase and then define that as zero degrees of phase. Once the phase is aligned and REL'd, the user could put the FS740 in quadrature with the reference signal by entering 90 degrees.

## Synchronize Phase to UTC

If the FS740 has received the time of day from GPS, then the phase of the various source outputs may be synchronized to the FS740's best estimate of UTC. This may be accomplished by pressing the SYNC button as shown in Figure 60.

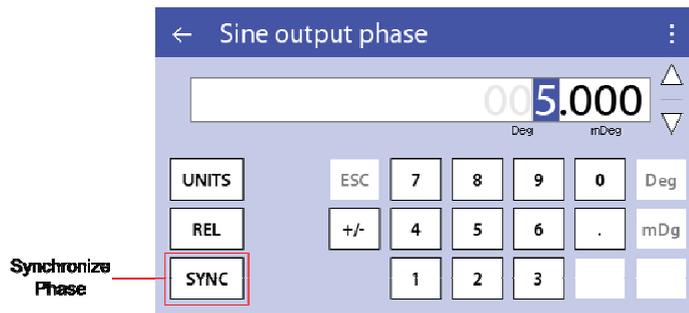


Figure 60: Synchronizing phase to UTC.

When the phase is synchronized the output will be momentarily disabled and then re-enabled such that the zero crossing or rising edge of the output will coincide with UTC. The displayed phase will be reset to zero as well.

## Auto Sync

Each of the three source outputs may be configured to automatically synchronize to UTC whenever the frequency is changed, or if the FS740 must jump phase to align itself with UTC. The selection is made by pressing the “Settings” button, which is the 3 vertically aligned dots at the right of the title bar, and toggling the “Auto sync” setting on or off as desired. The factory default is off.

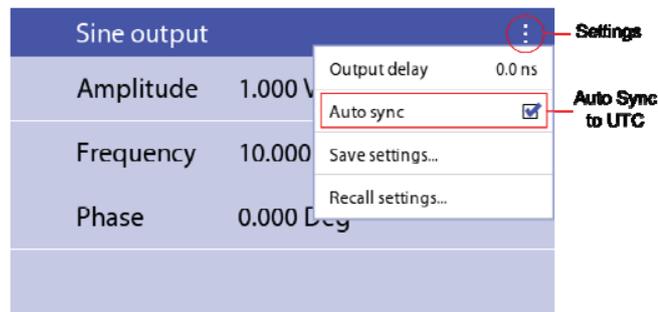


Figure 61: Auto synchronize output to UTC.

As noted before, when the FS740 synchronizes an output with UTC, it will disable the output and then re-enable it such that the zero crossing or rising edge of the output will coincide with UTC. Therefore the output may be disabled for up to a second when synchronizing to UTC.

## Output Delay

The FS740 is calibrated so that the zero crossing or rising edge of the output at the BNC corresponds to UTC. However, the user’s equipment may be located some distance away from the FS740. Thus, it may be desirable to advance or delay the signal so that rising edge of the output coincides with UTC at the input to the user’s equipment. This is achieved by pressing the “Settings” button, which is the 3 vertically aligned dots at the right of the title bar, and pressing the “Output delay” menu item. Enter a negative delay to advance the signal. Enter a positive delay to retard it. The factory default is 0.0 ns.

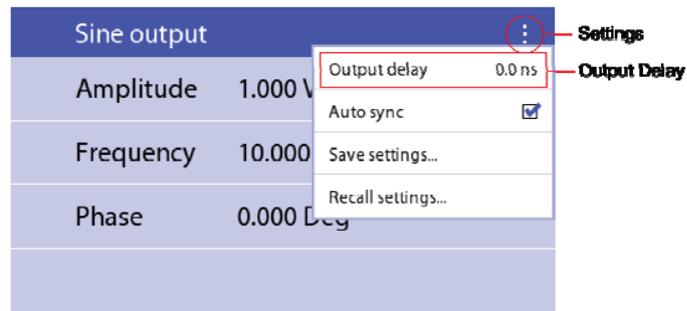


Figure 62: Adjust timing of source output.

For example, to compensate for 4 feet of RG-58 cable, which has a delay of 1.5417 ns/ft, the user would enter an output delay of  $-6.17$  ns, to advance the signal so that it will arrive at the end of the cable when the UTC second transition occurs.

## Related Commands

```
SOURce:PHASe
SOURce:PHASe:REFerence
SOURce:PHASe:SYNChronize
SOURce:PHASe:SYNChronize:AUTO
SOURce:PHASe:SYNChronize:TDELay
```

## Sine Output

The top level sine output display is shown in Figure 63. The user can easily navigate to this display by pressing the hardware key on the front panel above the sine output BNC. From here the amplitude, frequency and phase of the sine output can be easily configured using the techniques discussed above.

Sine output	
Amplitude	1.000 V <sub>pp</sub>
Frequency	10.000 000 000 MHz
Phase	0.000 Deg

Figure 63: Top level sine output display.

The sine output supports amplitudes ranging from 0.010 to 1.414 V<sub>pp</sub>. It supports frequencies from 1 mHz to 30.1 MHz. Its phase can be synchronized to UTC and stepped by up to 360 degrees in a single step. The phase can be adjusted beyond 360 degrees if it is done in multiple steps.

## Aux Output

The top level aux output display is shown in Figure 64. The user can easily navigate to this display by pressing the hardware key on the front panel above the aux output BNC. The aux output can generate several different types of waveforms including sine, triangle and square waves. The amplitude, frequency and phase of these waveforms can be easily configured using the techniques discussed above. The aux output can also

generate two special purpose outputs, a fixed 100 MHz sine output and an AM modulated IRIG B signal.

Aux output	
Config	Sine
Amplitude	1.000 V <sub>pp</sub>
Frequency	1.000 000 000 MHz
Phase	0.000 Deg

Figure 64: Top level aux output display.

The user selects the waveform output on the aux output by pressing the current configuration, which is “Sine” in Figure 64. This enables the user to select the desired configuration as shown in Figure 65.

Aux output	
Config	<input checked="" type="radio"/> Sine <input type="radio"/> Triangle <input type="radio"/> Square <input type="radio"/> 100 MHz Sine <input type="radio"/> AM IRIG B
Amplitude	
Frequency	MHz
Phase	

Figure 65: Aux output waveform configuration.

The aux output supports amplitudes ranging from 0.010 to 1.414 V<sub>pp</sub>. It supports frequencies from 1 mHz to 10 MHz for sine wave outputs, and 1 mHz to 1 MHz for the triangle and square wave outputs. The phase of these waveforms can be synchronized to UTC and stepped by up to 360 degrees in a single step. The phase can be adjusted beyond 360 degrees if it is done in multiple steps. The phase of the IRIG B is specified by the standard, but it can be shifted by modifying its output delay. The 100 MHz waveform is of a fixed amplitude and phase and cannot be synchronized to UTC.

## Related Commands

SOURce2:FUNction

## Pulse Output

The top level pulse output display is shown in Figure 66Figure 63. The user can easily navigate to this display by pressing the hardware key on the front panel above the pulse output BNC. The pulse output generates digital 5V CMOS levels. The user can specify a period and pulse width, or alternatively, a frequency and duty cycle. Like the other source outputs, the rising edge of the pulse output can be synchronized to UTC, and its phase adjusted by up 360 degrees in a single step using the same techniques discussed above.

Pulse output	
Config	Period / Width
Period	1.000 000 000 000 s
Width	500.000 000 000 ms
Phase	0.000 Deg

Figure 66: Top level pulse output display.

The pulse output can also generate a pulse coded IRIG B waveform. Like the AM modulated IRIG B waveform, the phase of the pulse coded IRIG B waveform is specified by the standard, but it can be shifted by modifying the output delay for the pulse output as discussed above.

The user can change the pulse output configuration by pressing the current configuration, which is “Period/Width” in Figure 66. This enables the user to select the desired configuration as shown in Figure 67.

Pulse output	
Config	<div style="border: 1px solid black; padding: 2px;">                     Pulse configuration  <input checked="" type="radio"/> Period / Width  <input type="radio"/> Freq / Duty  <input type="radio"/> Pulse IRIG B                 </div>
Period	000 s
Width	00 ms
Phase	0.000 Deg

Figure 67: Pulse output configuration.

The pulse period can range from 40 ns to 1000 s. and can be specified with 1 ps resolution. The frequency can range from 1 mHz to 25 MHz and can be specified with 1 μHz resolution. The pulse width may range from 5 ns to the period – 5 ns. These limits also constrain the minimum and maximum duty cycle.

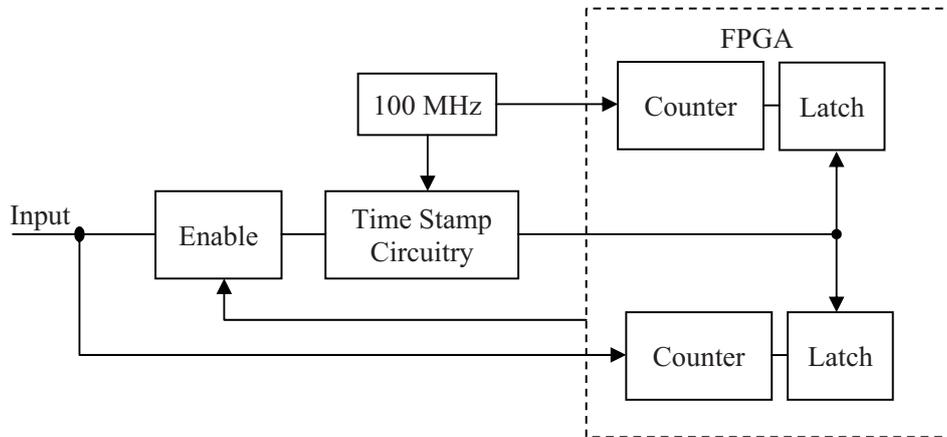
### Related Commands

```
SOURce3:FUNCTION
SOURce3:PULSe:DCYCLe
SOURce3:PULSe:PERiod
SOURce3:PULSe:WIDTh
SOURce3:PULSe:VIEW
```

## Measure

The FS740 can make independent time and frequency measurements on its front and rear inputs. It can time stamp an event with 50 ps of resolution relative to its estimate of UTC. The FS740 also has a built in counter which, when combined with the time stamp circuitry, enables successive frequency measurements with no dead time. As such, the FS740 is particularly well suited to making frequency stability measurements over time intervals ranging from 0.01 s to 50 million seconds. Furthermore, by averaging up to 625

time stamps in a single 10 ms gate, the FS740 can reduce the inherent noise in the time stamp circuitry to about 1 ps. Thus, with an OCXO installed, the FS740 can measure stabilities approaching  $2 \times 10^{-12}$  in a 1 second interval.



**Figure 68: Frequency measurement circuitry.**

A simplified diagram of the frequency measurement circuitry used by the FS740 is shown in Figure 68. A 100 MHz clock drives a free running counter in the FPGA which keeps track of system time with 10 ns of resolution. The input drives a second free running counter which tracks the number of edges generated by the input signal. When enabled, the time stamp circuitry latches the two counters and then measures the time between the 100 MHz clock edge and the input signal edge with a time to amplitude converter. Once the time tag has been recorded the FPGA resets the circuitry and re-enables the input. Since the counters are free running, no events are lost. Therefore, successive frequency measurements can be made with zero dead time as are needed when making frequency stability measurements. The time stamp circuitry can make a measurement once every 16  $\mu$ s for a total of 625 measurements in a 10 ms gate. By averaging these time stamps together, the baseline noise of the measurements can be reduced to about 1 ps. At this performance level, the noise of the measurement is completely determined by the stability of the timebase for time intervals larger than 1 second.

For time stamping of events, the same circuitry is used, except that the event counter is ignored and no averaging of time stamps takes place. Therefore, the noise floor is set by the single shot jitter of the time stamp circuitry, which is about 50 ps. Since no aggregating of data occurs, the maximum rate at which time tags can be processed is I/O limited to about 1 kHz and nondeterministic.

The FS740 provides two measurement input channels: one on the front panel and another on the rear panel. Each channel has separate, dedicated time stamp circuitry that can operate in parallel making it possible to time stamp two simultaneous events relative to a common clock.

## Making Measurements

The hardware key on the right hand side of the front panel, above the BNC connector located in the MEASURE section takes the user to the top level display for measurements. From here the user can configure and view time and frequency measurement results. A sample display is shown in Figure 69.

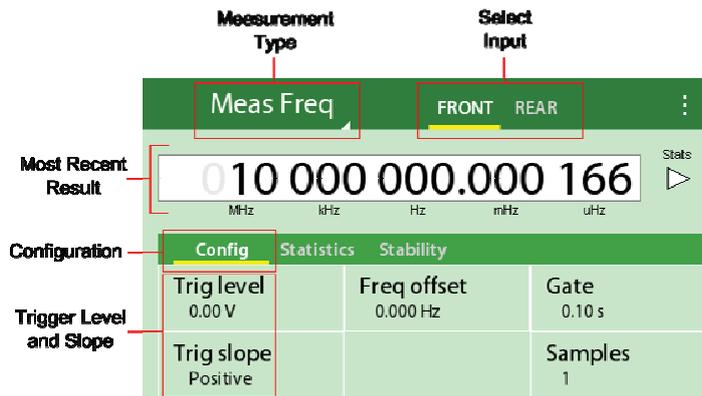


Figure 69: Top level measurement display.

At the top of the display in the title bar, the user makes two selections: the input channel being configured, and the measurement type for that channel. In the figure, the front input is selected as indicated by the yellow bar, and frequency is being measured on that input. Directly below the title bar is the result of the most recent measurement for the front input. When the user is operating the unit from the front panel, results are normally continuously updated and shown here. Finally, below the results are virtual tabs and cards which enable the user to configure and view more detailed information about the selected measurement. For frequency measurements, extra tabs are available for viewing measurement statistics and stability results. In the figure, configuration details are being shown.

### Measurement Configuration

When configuring measurements, typically the first two items to select are the input channel and the measurement type. Input selection is made by pressing the desired input: “FRONT” or “REAR.” The measurement type is selected by pressing the current measurement and selecting the desired measurement as shown in Figure 70.



Figure 70: Measurement type selection.

### Related Commands

```
MEASure:FREQuency?
CONFigure:FREQuency
MEASure:TIME?
CONFigure:TIME
```

### Input Configuration

For both frequency and timing measurements the input trigger level and slope must be configured. The trigger level defines the voltage level that must be crossed for an event

to be time stamped or counted. The trigger slope defines the direction or slope the input voltage must have to trigger an event. Positive slope indicates that the voltage should be rising when the trigger level is crossed. Negative slope indicates that the voltage should be falling when the trigger level is crossed. Both parameters may be easily modified using the techniques discussed in the section Parameter Modification on page 9.

## Related Commands

INPut:LEVel  
INPut:SLOPe

## Frequency Measurements

The configuration process for frequency measurements is usually quite simple:

1. Select the input channel.
2. Select frequency for the measurement type.
3. Adjust trigger level and slope as appropriate for your input signal.

The location for each of these parameters is highlighted in Figure 69. The LED above the front panel BNC input will flash or light when the corresponding input is successfully triggered. If the FS740 is not being controlled by a remote interface, the unit will immediately commence making frequency measurements, continuously displaying the latest result. If not, see the section Auto Triggering below.

## Frequency Offset

Sometime it is desirable to view measured frequency results relative to some specified offset, so that small deviations in the results are more easily comprehended. This is easily accomplished by setting the frequency offset. Pressing the “Freq offset” card presents the user with a window similar to that shown in Figure 71.



Figure 71: Frequency offset selection.

In the figure, we see that the current frequency offset is 10 MHz, and that the latest measurement was 136  $\mu$ Hz below 10 MHz. The user is presented with three options: set, clear, or specify. Pressing “SET” causes the frequency offset to be set to the latest measurement result, which would be 136  $\mu$ Hz below 10 MHz or 9,999,999.999,864 Hz for the example shown in the figure. Pressing “CLEAR” forces the offset to zero, which is the default. Finally, pressing “SPECIFY” enables the user to enter a specific offset if desired.

## Gate

To make a frequency measurement, the FS740 must make two measurements. It must time stamp the edge of the signal at the start of the measurement, and again at the end of the measurement. The frequency is computed as the total counts divided by the difference in the time stamps. The gate for a frequency measurement defines the minimum allowed time between the two measurements. Since the jitter of each time stamp is typically constant, the resolution of the measurement generally increases or decreases in proportion to the gate interval. For the default gate interval of 0.1 s, the FS740 can generally provide 11 digits of resolution. This increases to 12 digits for a 1 s gate, and decreases to 10 digits for a 0.01 s gate. The FS740 accepts gate intervals from 0.01 s to 1000 s.

## Samples

The FS740 has enough internal storage to hold 500,000 frequency results. It can also compute some common statistical measures for a group of measurements, such as the min, max, mean, and Allan deviation. The “Samples” card enables the user to set the number of samples in a group of measurements from 1 to 1 billion. Although the latest measurement is often auto-triggered and displayed immediately, groups of measurements must be explicitly initiated by the user by pressing the play button as shown in Figure 72

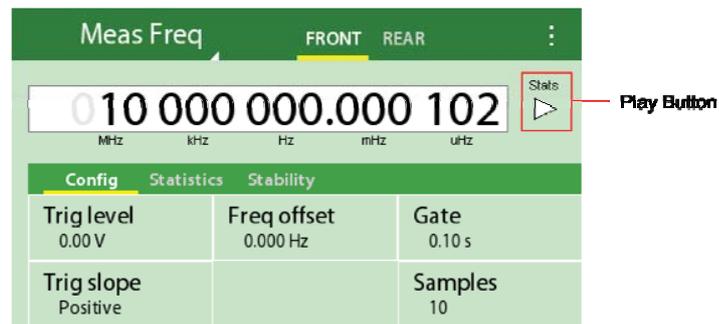


Figure 72: Initiating a group of measurements.

Once a group of measurements is initiated, statistics and frequency stability calculations for those measurements will be automatically computed. Although, the latest result is always displayed on the front panel, stored results can generally only be downloaded via one of the remote interfaces.

## Auto Triggering

The user explicitly initiates a measurement or a group of measurements by pressing the play button as shown in Figure 72. Measurement results which are explicitly initiated are automatically stored in internal memory and may be downloaded by the user via one of the remote interfaces. However, when operated from the front panel, the FS740 will normally automatically trigger measurements and display the most recent result on the front panel. Automatic triggering of measurements is convenient during setup and initial characterization of a signal as it provides the user with immediate feedback. However, automatically triggered results are not stored in internal memory and are generally not available over the remote interfaces. If desired, automatic triggering may be turned off via the settings button, the three vertically aligned dots at the top right corner of the display, as shown in Figure 73.

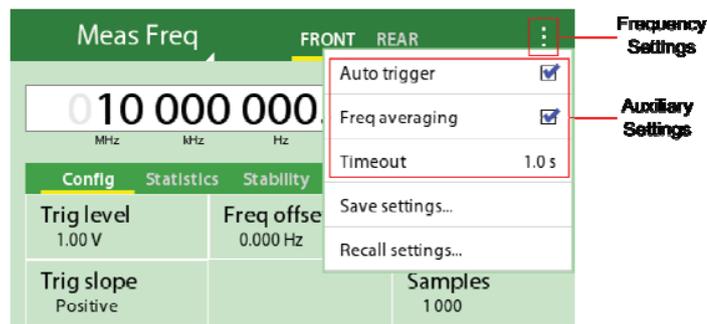


Figure 73: Auxiliary frequency settings.

Uncheck auto trigger to turn off auto triggering. When auto triggering is off, all measurements must be explicitly requested by pressing the play button.

## Fast Averaging

As mentioned above, the FS740 has the ability to make 625 time tag measurements in a 10 ms gate. By averaging these measurements together, the jitter in the results can be reduced from 50 ps to 1 ps. When making frequency measurements, this is normally desirable. Thus, this averaging is enabled by default. The user may disable it, however, if desired. Uncheck freq averaging to turn it off as shown in Figure 73.

## Timeout

The time required to make a frequency measurement is normally determined by the gate for the measurement. However, if a signal is removed during a frequency measurement, the measurement cannot complete normally, because it must wait for a final trigger which will never occur. In this case, the measurement will timeout and the result will be undefined. Undefined results return 9.91e37 when queried over a remote interface. The default timeout period of 1.0 s should work well for most signals. If the signal period is longer than 1.0 s, then the timeout period should be extended to be longer than the period of the signal. Press “Timeout” in frequency settings to modify the timeout period as shown in Figure 73.

## Related Commands

```
MEASure:FREQuency?
CONFigure:FREQuency
INPut:LEVel
INPut:SLOPe
CALCulate:REFerence
SENSe:FREQuency:GATE
SAMPle:COUNT
```

## Frequency Statistics

When a group of measurements is initiated by pressing the play button as shown in Figure 72, the FS740 automatically computes some summary statistics on the measured results including the mean, Allan deviation, min, max, and count. These may be easily viewed from the front panel by pressing the “Statistics” virtual tab as shown in Figure 74.

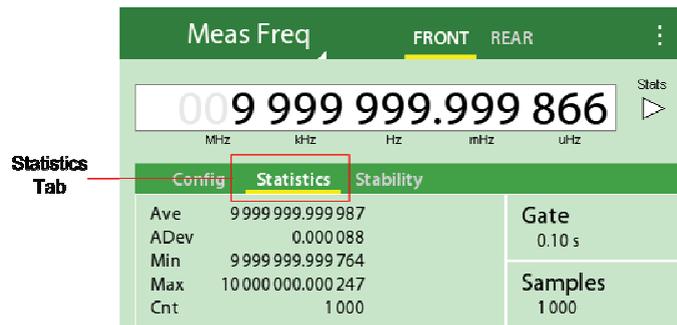


Figure 74: Viewing frequency statistics.

Frequency statistics are updated on the fly as measurements are completed.

## Related Commands

CALCulate:STATistics

## Frequency Stability

Frequency stability is commonly measured by computing the Allan variance of a series of measurements. The Allan variance is defined as follows:

where  $\tau$  is the observation or gate interval for the measurement while  $y_n$  is the fractional frequency, and  $x_n$ , the time error, of the  $n^{\text{th}}$  interval. The brackets denote the ensemble average of the sequence. The Allan deviation is merely the square root of the Allan variance:

To make Allan deviation measurements, the user must collect a series of time tags of the appropriate interval with no dead time and compute the average frequency deviation from one interval to the next in the series. Furthermore, users are often interested in computing the Allan deviation over a wide range of observational intervals,  $\tau$ , not just one. Making these measurements correctly is often difficult and time consuming to setup and compute.

The FS740, however, makes Allan deviation measurements easy. The FS740 is continuously making time tag measurements with no dead time, regardless of the configured gate. Instead of discarding these measurements, they are used to compute the Allan deviation for all time intervals in a 1, 2, 5, sequence from 0.01 s to 50 million seconds. The user need only configure the total time interval for the measurement which is the gate interval times the number of samples. For example, to get Allan deviation results for a 1 million second time record, one could configure a measurement with a 1 second gate and a million samples.

For quicker convergence when computing the ensemble averages, the FS740 uses an algorithm similar to that used in the overlapped Allan variance calculation. In the overlapped Allan variance calculation all time tags are included when computing the ensemble average for a given interval. This is not feasible for the FS740 due to memory

constraints. Instead, the FS740 retains a maximum of 50 time tags at each time interval. The time tags are spread evenly over the interval, however, to maximize the independent degrees of freedom of the measurement.

## Making Frequency Stability Measurements

When a group of measurements is initiated by pressing the play button as shown in Figure 72, the FS740 automatically computes frequency stability results, i.e. Allan deviations, for time intervals from 0.01 s to 50 million seconds. The results may be easily viewed by pressing the “Stability” virtual tab as shown in Figure 75.



Figure 75: Viewing frequency stability.

Results are shown in tabular format for intervals from 0.01 s to 50 million seconds in a 1, 2, 5 sequence. In Figure 75, the first column of results represents the Allan deviations for time intervals 0.1, 0.2, and 0.5 s. the second column represents the Allan deviations for time intervals 1.0, 2.0, and 5.0 s. The third and fourth columns are interpreted similarly with the interval increasing by a factor of 10 with each column. Longer and shorter intervals in the table may be viewed by pressing the display interval arrows, as noted in the figure. The FS740 returns zero for stability measurements that have not completed, yet.

### Related Commands

CALCulate:STABility

### FS740 Baseline Measurement Stability

The time tagging circuitry of the FS740 has a typical jitter of approximately 30 ps. By enabling averaging, this performance improves by more than 10 fold. However, the time to amplitude converters also contain systematic nonlinearities that contribute jitter to measurements which depend on the frequency of the signal in question. The nonlinearities are partially attributable to curvature in the slope of the time to amplitude converters. However, they are also attributable to interference with other signals within the FS740. If the signal being measured is nearly synchronous with one of these interfering noise sources, the jitter performance will be degraded.

Signals with a period that is an integer multiple of 10 ns are synchronous with the FS740’s timebase and measurement stability will be superior at these frequencies because nonlinearities and interference in the time to amplitude converters are eliminated. When the frequency is no longer synchronous, jitter performance will be degraded depending on the frequency offset and whether frequency averaging is enabled.

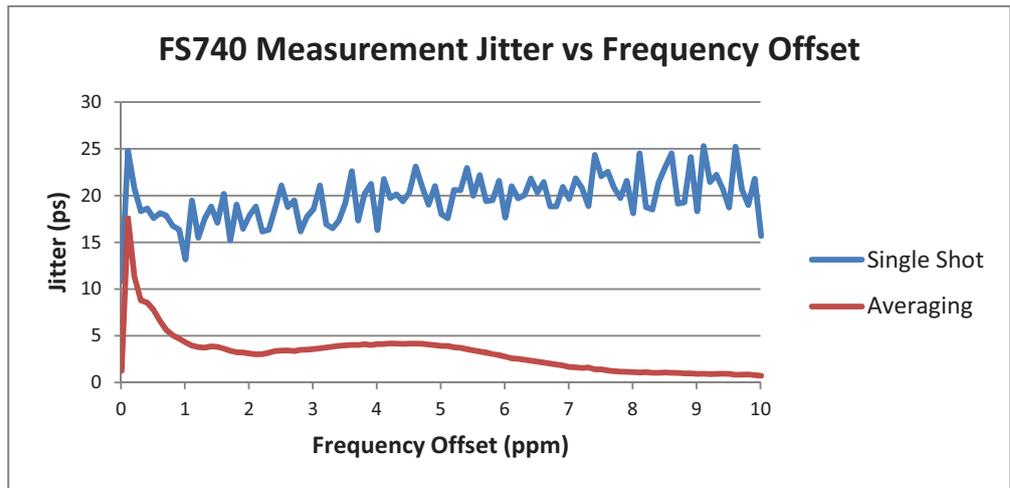


Figure 76: FS740 measurement noise vs frequency offset from a synchronous frequency

Figure 76 details typical measurement jitter of the FS740 time tags as the signal frequency deviates from a synchronous frequency. If frequency averaging is turned off the measurement jitter is roughly constant at about 20 ps, regardless of offset frequency. When frequency averaging is turned on, however, random noise is effectively averaged away and the noise drops to about 1 ps at synchronous frequencies. At asynchronous frequencies the noise increases to about 17 ps at a frequency offset of 0.1 ppm, and then gradually drops below 5 ps at a frequency offset of 1 ppm and then gradually falls to 1 ps at 10 ppm. The increased noise at 0.1 ppm is due to the additional jitter from interference and nonlinearities in the time to amplitude converters that are systematic on a time scale of 10 ms and cannot be averaged away. At large frequency offsets, these additional noise sources start to appear random again, and averaging effectively removes it.

For white noise the frequency stability is just the jitter divided by the time interval. Thus, 17 ps of jitter corresponds to a baseline frequency stability of  $1.7 \times 10^{-9}$  for a 10 ms interval and  $1.7 \times 10^{-11}$  for a 1 s interval. This is graphically depicted in Figure 77.

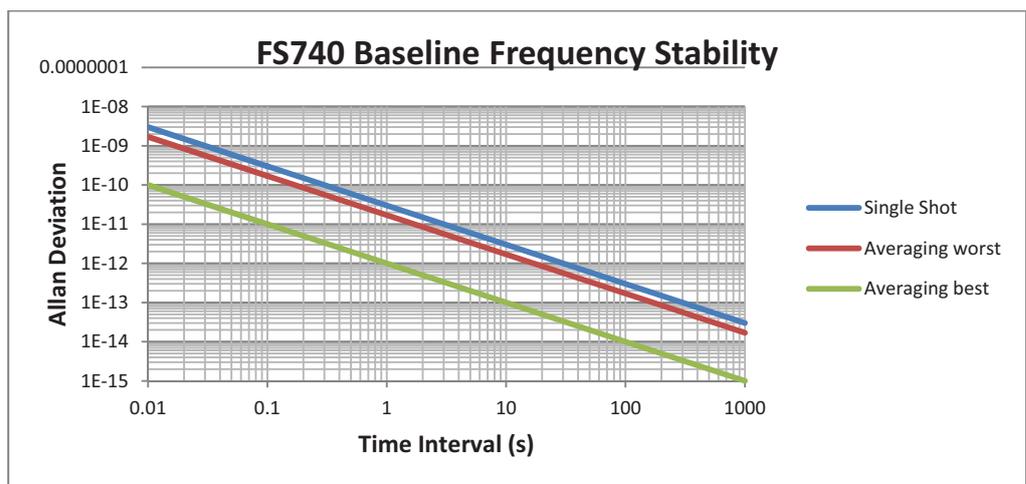


Figure 77: FS740 baseline noise floor for frequency stability measurements.

## FS740 Timebase Stability

When making stability measurements, one must also consider the stability of the FS740's installed timebase which is reproduced in Figure 78 for the various available timebases.

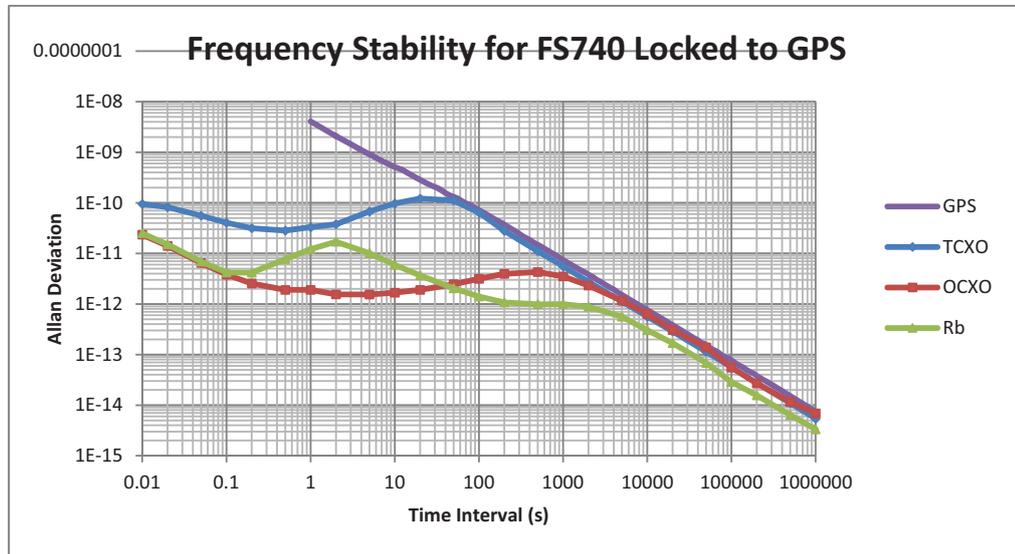


Figure 78: Frequency stability of FS740 with different timebases installed.

Notice that the timebase stability for the OCXO and Rb actually exceeds the baseline measurement stability shown in Figure 77 for time intervals shorter than 1 second. Thus for time intervals less than 1 second the measured stability will be limited by the baseline noise of the measurement as depicted in Figure 77. Conversely, for time intervals greater than 1 second, the measured stability will be limited to stability of the installed timebase as depicted in Figure 78.

## Effects of Averaging on Frequency Stability

As noted above, the baseline frequency performance of the FS740 is improved by up to a factor of 25 by averaging together up to 625 time tags in a 10 ms gate. The improvement is achieved because the jitter in the time to amplitude converters is averaged away. However, it should also be noted that broadband noise in the signal under analysis will also be averaged away for the same reason. The averaging effectively inserts into the measurement a low pass filter with a 3 dB point at 44 Hz. As a result, for clocks with significant broadband noise, the FS740 will report short term stabilities that are better than reality. Users who wish to include such broadband noise in their stability analysis should turn off frequency averaging (see Figure 73). Doing so, however, will necessarily degrade the baseline noise of the FS740 (see Figure 77), and limit the maximum performance that can be measured.

## Timing Measurements

The FS740 can time tag external events with a resolution of better than 50 ps. The event is time tagged relative to the FS740's best estimate of UTC. The absolute accuracy is 100 ns, but the variation is typically better than 10 ns. Finally, the relative precision of two events is better than 50 ps plus the timebase error. Time tagging has multiple applications in astronomy, physics, and the power industry.

The configuration process for timing measurements is quite simple:

1. Select the input channel.
2. Select time for the measurement type.
3. Adjust trigger level and slope as appropriate for your input signal.
4. Set the desired number of samples.
5. Press play button to initiate the capture of time tags.

The location for each of these parameters is highlighted in Figure 79. The LED above the front panel BNC input will flash or light when the corresponding input is successfully triggered. If the FS740 is not being controlled by a remote interface, the unit will immediately commence making timing measurements, continuously displaying the latest result. If not, see the section Auto Triggering on page 59.

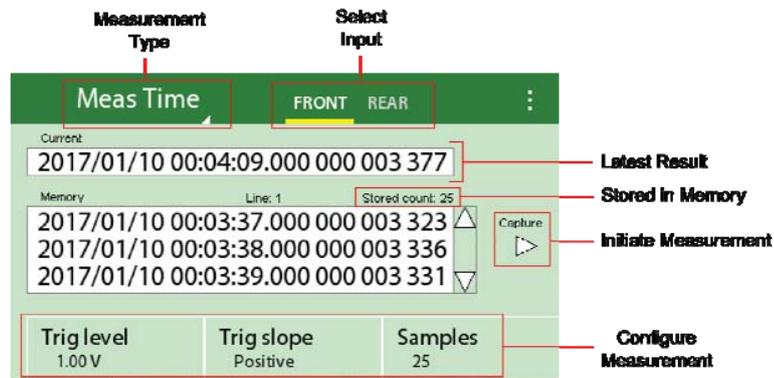


Figure 79: Timing measurement configuration.

The FS740 has enough internal memory to store up to 250,000 measurements. By default, when the memory fills up, the oldest result will be discarded to make room for the newest result. Alternatively, the user may choose to keep the oldest results and discard the newest results. To do this press the settings button and check the box labeled “Keep first tag,” as shown in Figure 80.

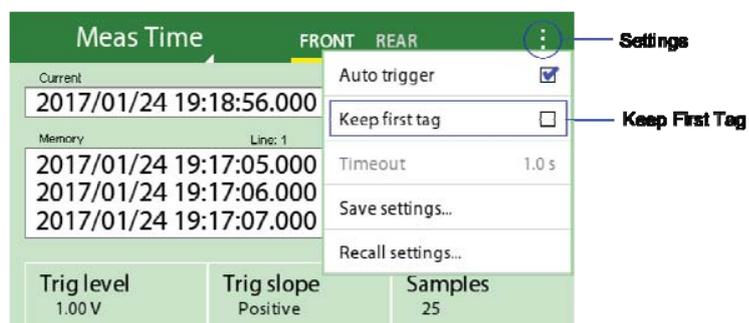


Figure 80: Setting time tag buffering behavior.

The user may view the first few results via the front panel user interface. For large data sets, however, it will only make sense to download the results via one of the remote interfaces.

## Factory Default Settings

The factory default settings for the FS740 are shown in Table 3. The settings are divided into two categories: system settings and user settings. User settings largely apply to the configuration of the source outputs and the measurement inputs. The user may force these parameters to their default settings by executing the \*RST command or recalling factory presets via the front panel interface. These parameters are identified by a check mark at the far right of Table 3.

System settings, on the other hand, largely apply to the configuration of the timebase, GPS, communications parameters, and system alarms. They are unaffected by the execution of the \*RST command or a recall of factory presets from the front panel. They are stored in nonvolatile memory and retain their configuration even when power is cycled. These parameters are generally intended to be set once and then left alone.

**Table 3: Factory default settings**

Parameter	Value	Set by *RST
Display	Timebase Display	✓
Display power down	Never	
Sine frequency	10 MHz	✓
Sine amplitude	1 Vpp	✓
Sine amplitude units	Vpp	✓
Sine phase	0 Degrees	✓
Sine phase units	Degrees	✓
Sine phase auto sync	Off	
Sine phase time delay	0 ps	
Aux waveform	Sine wave	✓
Aux frequency	1 MHz	✓
Aux amplitude	1 Vpp	✓
Aux amplitude units	Vpp	✓
Aux phase	0 Degrees	✓
Aux phase units	Degrees	✓
Aux phase auto sync	Off	
Aux phase time delay	0 ps	
Pulse waveform	Pulse	✓
Pulse view	Period/Width	✓
Pulse frequency	1 Hz	✓
Pulse duty cycle	50 %	✓
Pulse period	1 s	✓
Pulse width	500 ms	✓
Pulse phase	0 Degrees	✓
Pulse phase units	Degrees	✓
Pulse phase auto sync	Off	
Pulse phase time delay	0 ps	
Input level	0 V	✓
Input slope	Positive	✓
Input measurement	Frequency	✓
Input frequency offset	0 Hz	✓
Input frequency gate	0.1 s	✓
Input sample size	1	✓

Input auto trigger	On	✓
Input frequency averaging	On	✓
Input time out	1 s	✓
Input time tag buffer mode	Keep last	✓
Timebase GPS lock	Enabled	
Timebase holdover mode	Wait for good 1pps timing from GPS	
Timebase bandwidth mode	Auto bandwidth control	
Timebase time interval limit	1 $\mu$ s	
Timebase manual time constant	30 s	
GPS timing alignment	UTC	
GPS timing quality	Require 3 satellites	
GPS survey mode	Remember survey results	
GPS position fixes in survey	2000	
GPS antenna cable delay	0 ns	
Local time offset	0 hr	
RS-232 baud rate	115200 baud	
Ethernet speed	100 Base T	
TCP/IP configuration method	DHCP	
TCP/IP static IP address	0.0.0.0	
TCP/IP static subnet address	0.0.0.0	
TCP/IP static gateway address	0.0.0.0	
System alarm mode	Manually set state	
System alarm manual state	Off	
System alarm holdover duration limit	0 s	
System alarm timing error limit	100 ns	
Option sin/aux routing	Sine out	

## Forcing User Settings to Factory Defaults

The user may easily force user settings to their factory default values from the front panel by pressing Settings > Recall Settings, and then choosing “Factory Presets”

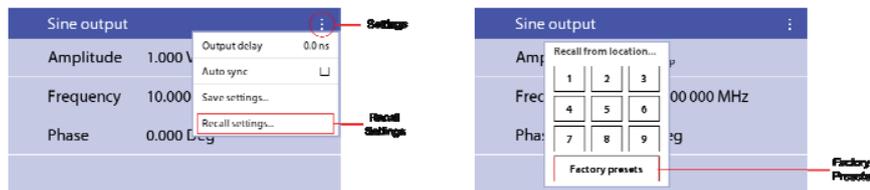


Figure 81: Setting user settings to factory presets.

This is equivalent to executing the remote interface command \*RST. It forces parameters in Table 3 with a check mark at the right to the default setting.

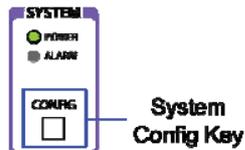
## Related Commands

\*RST

## Forcing All Settings to Factory Defaults

Occasionally it may be useful to force ALL instrument settings to their factory default state. This may be necessary, for example, when transferring a unit from a secure location. Perform the following procedure to wipe the instrument of all user settings and force all system settings to their factory default values:

1. Unplug the power cord to the FS740.
2. Press the front panel System Config key.



3. While pressing the System Config key, re-plug in the power cord to the FS740.

The FS740 will display a message that all settings are being restored to their default values.

### Related Commands

`SYSTEM:SECURITY:IMMEDIATE`

---

# Remote Programming

---

## Introduction

The FS740 may be programmed via remote interfaces included with the instrument. Any host computer interfaced to the instrument can easily control and monitor its operation. The front panel can even be locked out when necessary for complete computer control. This might be desirable as part of an automated test environment. The FS740 supports two standard remote interfaces: RS-232 and an Ethernet based local area network (LAN) interface which supports TCP/IP communication.

### RS-232

An RS-232 communications port is included on the rear panel of the unit. The RS-232 interface connector is a standard 9 pin, type D, female connector configured as a DCE (transmit on pin 2, receive on pin 3). In order to communicate properly over RS-232, the instrument and the host computer both must be configured to use the same settings. The following baud rates are supported: 115200 (default), 57600, 38400, 19200, 9600, and 4800. The rest of the communication parameters are fixed at 8 data bits, 1 stop bit, no parity, and RTS/CTS hardware flow control. See section RS-232 Configuration on page 39 for details on how to configure the RS-232 port from the front panel.

### Ethernet

A rear panel RJ-45 connector may be used to connect the instrument to a 10/100 Base-T Ethernet LAN. In order to function properly on an Ethernet LAN, the unit needs to obtain a valid TCP/IP configuration, which includes an IP address, a subnet mask, and a default gateway or router address. The FS740 supports three methods for obtaining these parameters: DHCP for automatic IP configuration, Auto-IP for link local addressing, and Static IP for manual configuration. Check with your network administrator for the proper method of configuration of instruments on your network before attaching the FS740 to your network. See section TCP/IP Configuration on page 41 for details on how to select the proper method of TCP/IP configuration from the front panel.

Once connected to the network, the user can connect to the FS740 using one of the following protocols: telnet, bare TCP stream connections, and VXI-11.

### Telnet

The FS740 accepts telnet connections on port 5024. Telnet is useful for interactive sessions because it supports line based command execution with automatic echoing of typed characters. This enables users to easily type (and correct) commands manually from the keyboard. Telnet, however, should not be used for automated computer control, however, because the echoing of characters and processing of telnet commands will significantly complicate the endeavor.

### Bare TCP

The FS740 accepts bare TCP stream connections on port 5025. Bare TCP connections are ideal for automated computer control. All bytes are passed directly to the input queue

of the FS740 and all responses sent directly back to the user application. This connection is very similar in style to RS-232 connections.

## VXI-11

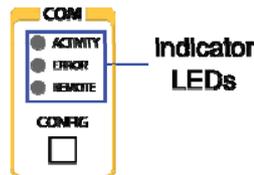
VXI-11 is a TCP based network connection that mimics a GPIB interface. Like the other connections, the user can write to the input queue of the FS740 and read back responses from its output buffer, but it also enables the user to issue GPIB commands such as a device clear, or a serial poll and it supports request for service (RQS) notifications. Interfacing to this server, however, requires the installation of a VISA library on the host computer. The user makes calls into the VISA library to connect and communicate with the FS740. The VISA library then issues the low level calls to the VXI-11 server on the FS740. Unfortunately, SRS does not supply an implementation of this library. However, it can be purchased from several third party vendors, such as National Instruments and Keysight. In fact, the library is often bundled for free with the purchase of hardware instrumentation from either of these companies.

## Network Security

Network security is an important consideration for all TCP/IP networks. Please bear in mind that the FS740 does NOT provide security controls, such as passwords or encryption, for controlling access. If such controls are needed, you must provide it at a higher level on your network. This might be achieved, for example, by setting up a firewall and operating the instrument behind it.

## Front-Panel Indicators

To assist in programming, there are three front panel indicators located under the COM section of the front panel: ACTIVITY, ERROR, and REMOTE. The location of the LEDs is shown in Figure 82.



**Figure 82: Communications indicator LEDs.**

The REMOTE LED is on when the instrument is in remote lock out. In this mode, the front panel interface is locked out and the instrument can only be controlled via the remote interface. To go back to local mode, the user must touch the LCD display and then press the virtual button GO TO LOCAL.

The ACTIVITY LED serves as an activity indicator that flashes every time a character is received or transmitted over one of the remote interfaces. This is useful when troubleshooting connections because it clearly indicates when the FS740 successfully received and responded to a command.

The ERROR LED will be highlighted when a remote command fails to execute due to illegal syntax or invalid parameters. Once highlighted, the LED will remain lit until the error queue is cleared. Errors may be viewed and cleared by navigating to Communications > Activity. The most recent error is displayed at the bottom. Press the virtual button CLEAR to dismiss the error. Press CLEAR ALL to dismiss all errors.

## SCPI Command Language

The FS740 uses the SCPI (Standard Commands for Programmable Instruments) language for controlling the instrument over a remote interface. The SCPI language is an ASCII based command language that organizes functions into a hierarchical tree of commands with branches of the tree separated by colons.

### SubSystems

The base or root of the tree represents a subsystem of the instrument. Each succeeding branch of the tree subdivides the subsystem into related categories of functionality. The final branch of the tree identifies a command related to the subsystem that can be executed by the FS740. This structure facilitates understanding of the functions carried out by commands. As an example, consider the subset of the STATus subsystem shown below.

```

STATus:
  GPS:
    CONDition?
    ENABle
    ENABle?
    [:EVENT]?
  OPERation:
    CONDition?
    ENABle
    ENABle?
    [:EVENT]?

```

STATus is a subsystem of the FS740 and it is at the root of the tree. At the next level down, the STATus subsystem is divided into two branches: GPS and OPERation. Each of these categories is then further subdivided into four virtually identical commands: CONDition, ENABle, ENABle?, and [:EVENT]?. However, because of the hierarchical structure of the language, we can infer that the commands listed under the GPS branch refer to GPS receiver status, while those listed under OPERation carry out the same functions but refer to OPERation status rather than GPS status. Thus, the hierarchical structure of the commands aids in the understanding of the operations carried out by the individual commands.

### Understanding Command Syntax

SCPI commands often take one or more parameters which modify or identify the numerical value a variable should take. Some parameters are required. Others may be optional. Furthermore, the data types for each parameter may differ. Thus, for brevity, we need a set of conventions for defining commands which clearly identifies all the valid variations of the command without having the list each possibility separately. These conventions are set forth here.

An example command is illustrated below:

```
SOURce[{:1|2}]:VOLTage[:LEVel] {<voltage>|MINimum|MAXimum|DEFault}
```

## Keyword Case

Keywords are defined with a mixture of upper-case and lower-case letters. The upper-case letters indicate the short or abbreviated version of the keyword. This is usually the first 3 or 4 letters of the keyword. The user may send either the short version or the entire long version of the keyword in their programs. The case of the letters sent to the FS740 does not matter. It is only used here to succinctly identify the two versions of the keyword. Thus, SOUR, source, and Sour are all acceptable forms of the keyword. Other forms, such as SOU, or SOURC, are not. Given the definition above, the following commands are all identical:

```
SOUR:VOLT MIN
SOURCE:VOLTAGE MINIMUM
SOUR:VOLTAGE MIN
```

## Punctuation Used in Definitions

The following punctuation is used to identify variations and options for the command:

- Braces ( { } ) enclose different parameter choices. The braces, themselves, are not sent with the command
- A vertical bar ( | ) separates alternative parameter choices for the command. In the example above, the choices are a <voltage> or one of the keywords: MINimum, MAXimum, or DEFault. The vertical bar is not sent with the command.
- Triangle brackets ( < > ) indicate that you must specify a numerical value. In the example above, <voltage> would be specified as a number. Thus, one could set the voltage using the following command: SOUR:VOLT 1.253. The triangle brackets are not sent with the command.
- Square brackets identify optional keywords or parameters in the command. Optional items may be omitted if desired. In such case a default value is normally substituted for the parameter. In the example above, the keyword LEVel is optional and may be omitted. Thus, the command SOUR:VOLT:LEVEL 1.253 is identical to the command that omits the keyword: SOUR:VOLT 1.253.

## Numeric Suffix

Some keywords can have a numeric suffix appended to the keyword. In the example SOURce may be followed with a 1 or 2. These suffixes identify multiple channels of the same subsystem. Thus, SOUR1 identifies channel 1, and SOUR2 identifies channel 2. The numeric suffix is optional and if omitted, channel 1 is assumed.

## Examples

Putting it all together, all of the following commands are valid given the example definition presented above.

```
SOUR:VOLT MIN
SOUR2:VOLT DEFAULT
SOURCE1:VOLTAGE 1.234
SOURCE:VOLT MAXIMUM
```

## Queries

Command queries are usually formed by appending a question mark ( ? ) to the command. To query the output voltage of channel 1, we use the following command:  
SOUR:VOLT?

## Separators

As mentioned above, a colon ( : ) separates the different keywords that make up a command. If a command takes a parameter, a space MUST separate the last keyword of a command and the first parameter of that command. If a command takes multiple parameters, they are separated from each other with a comma ( , ). Finally, a semicolon ( ; ) is used to separate multiple commands on the same line. If the following command is in the same subsystem as the preceding command then the subsystem is not repeated for the second command. Otherwise the command must be fully specified and preceded by a colon. Given the example command tree presented in section SubSystems above, the following commands are valid:

```
STAT:GPS:ENAB 1; EVENT?
STAT:GPS:COND?; :STAT:OPER:COND?
```

In the first line, both ENABLE and EVENT are in the same subsystem, STAT:GPS, so that portion of the command is omitted for the EVENT? query. However, the two condition queries are in different subsystems, so the operational condition query is preceded by a colon and fully specified on the command line.

When multiple queries, separated by semicolons, are made in a single command line, the responses from the individual queries are separated by a semicolon as well.

## IEEE 488.2 Common Commands

The IEEE 488.2 standard defines several common commands that nearly all instruments support. Common commands start with an asterisk ( \* ) followed by three letters. Like with SCPI commands, a space MUST separate the command and any parameters which follow. Multiple common commands may be executed in a single line by separating the commands with a semicolon ( ; ). An example is given below.

```
*RST; *OPC?
```

## Parameter Types

The SCPI language supports several different data types for use with command parameters.

## Numeric Values

Parameters that take numeric values accept all common decimal representations of numbers, including optional signs, decimal points, or scientific notation. Hexadecimal values specified with a prefix of 0x as used in the C language are also accepted. If only certain values are allowed, numeric entries will be rounded to the nearest allowed value. The following examples are all valid numeric entries:

```
100
-123.456
+1.23456e2
-.456
0x64
```

The last example, 0x64, is the hexadecimal representation for the decimal number 100.

Many commands that take numeric parameters will also accept the keywords MINimum, MAXimum, or DEFault to set the parameter to the requested value for that parameter.

## Units

Many numeric values may be followed with a unit designation. The most common engineering prefixes are also accepted. For example, source frequency may be specified in Hz as SOUR:FREQ 10 KHZ. The FS740 accepts the units specified in Table 4:

Table 4: Units accepted by the FS740

Freq.	VDC	VAC			Time	Phase		Percent
		V <sub>PP</sub>	V <sub>RMS</sub>	dB		Deg	Rad	
Hz	nV	nV <sub>pp</sub>	nV <sub>rms</sub>	dBnW	ps	nDeg	pRad	%
kHz	uV	uV <sub>pp</sub>	uV <sub>rms</sub>	dBuW	ns	uDeg	nRad	
MHz MAHz	mV	mV <sub>pp</sub>	mV <sub>rms</sub>	dBm dBmW	us	mDeg	uRad	
GHz	V	V <sub>pp</sub>	V <sub>rms</sub>	dBW	ms	Deg	mRad	
	kV	kV <sub>pp</sub>	kV <sub>rms</sub>		s		Rad	
	MAV	MAV <sub>pp</sub>	MAV <sub>rms</sub>					

Note that since the SCPI language is case insensitive, it cannot use the case of the letter “m” to distinguish between mV (millivolts) and MV (megavolts). Thus, the prefix MA is used to designate Mega in SCPI. The one exception to this is for frequency where convention prevails and the unit designations MHz and MAHz both represent megahertz.

### Special Numbers

The SCPI language defines 3 special numbers: infinity, negative infinity, and not a number. These are defined to take the values shown in Table 5

Table 5: SCPI Special Numbers

Abbreviation	Meaning	Value
INF	Infinity	+9.90e37
NINF	Negative infinity	-9.90e37
NAN	Not a number	+9.91e37

These special numbers may be returned by commands when data is invalid. A measurement of frequency, for example, may time out. In this case, NAN will be returned as its value. The physical values were chosen to be so large as to be not confused with normally expected physical values, but small enough to be easily representable by a single precision floating point number format.

### Discrete Parameters

Some parameters take one of a small list of allowed keywords. They often have a short form and a long form, just like command keywords. In the command definition, the upper case letters indicate the short form. Either case may be used when sending the short or long form of the value to the instrument. Queries will always return the short form. Consider the following command definitions:

```
TBASe:CONFig:BWIDth [{ AUTO | MANual }]
TBASe:CONFig:BWIDth?
```

The user may specify auto bandwidth control by sending the command TBAS:CONF:BWID AUTO. The query will return AUT, which is the short form of the value.

### String Parameters

Quoted string parameters allow one to send almost any sequence of characters, including characters that are normally reserved as separator characters, such as a comma,

semicolon, or colon. The string must begin and end with the same quote character: either a single quote, or a double quote. The quote delimiter may itself be included in the string if it is typed twice without any characters in between.

### Command Termination

Commands should be terminated with a line feed <LF>. They may optionally be terminated with a carriage return <CR> followed by a line feed <LF>. As previously noted, multiple commands may be sent in a single line if they separated by a semicolon (;). Commands are executed in the order received and execution commences once the command separator or terminator is received.

## Status Reporting

### Architecture

The FS740 reports on its status via a hierarchy of status registers. Instrument status is stored in three 16-bit registers: the questionable status register, the operational status register, and the GPS receiver status register. Parsing and command execution status is reported via the standard event register (\*ESR?). Summaries of all these registers are reported as bits in the serial poll status byte for the instrument (\*STB?). The serial poll status byte may be configured to generate a request for service, RQS, from a remote interface when a given bit is set. This scheme enables the user to be notified when events of interest occur, and to ignore events that are not of interest. Detailed status is always available in the source registers. However, with proper configuration, these registers need not be queried until an event of interest actually occurs.

Each instrument status register has three associated status words: a condition register, an event register and an enable register. The relationship of these three registers is depicted in Figure 83.

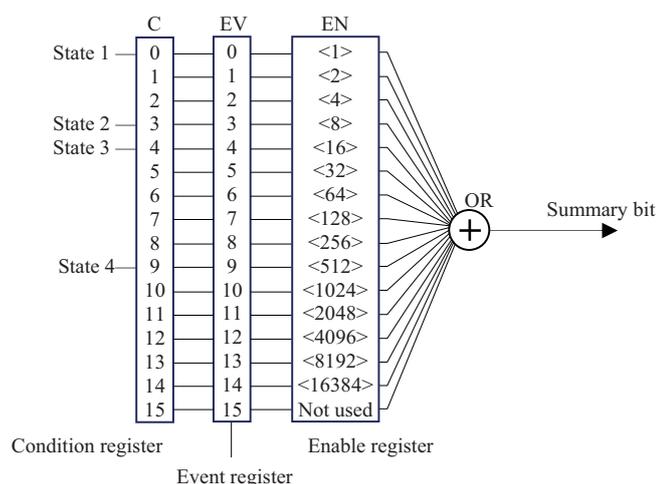


Figure 83: Organization of status registers

### Condition Register

The condition register reports on the current state of the instrument. At the far left of Figure 83 are listed four items of state which feed into four different bit locations of the condition register. When the instrument is in the given state, then the corresponding bit

in the condition register is set. When the instrument leaves the given state the corresponding bit is cleared. Bits in the condition register which are set indicate items of state that are true at the time of the query. Bits in the condition register which are clear indicate items of state that are false at the time of the query. Querying the condition register does not alter the bits in the register. Only changes in the actual instrument state alter the bits of the condition register.

When the condition register is queried, only one number is returned which is the binary sum of all bits in the register which are set. The binary weight of each bit in the register is shown in the enable register at the far right of Figure 83. The binary weight increases by a factor of two for each bit. Bit 0 has a weight of 1. Bit 1 has a weight of 2. Bit 2 has a weight of 4, and so on up to bit 14 which has a weight of 16384. The number returned by a query of the condition register is the sum of weights of the bits which are set.

For example, in Figure 83, State 1 feeds into bit 0 and State 2 feeds into bit 3. If both State 1 and State 2 are true, then both bits 0 and 3 will be set. If all other states are false, then the number returned by a query of the condition register will be the binary sum of bit 0 and bit 3, which is  $1 + 8 = 9$ . If only State 2 is true, then only bit 3 will be set and a query of the condition register will return 8. Similarly, if only State 1 is true, then only bit 0 will be set and the query will return 1. Finally, if none of the states is true, the query will return 0.

## Event Register

Bits from the condition register feed into corresponding bits in the event register. The event register differs from the condition register, however, in that the bits are sticky. Once a bit is set, it remains set until explicitly cleared by a query of the event register. The event register, therefore, enables the user to capture all events that have occurred since the previous query of the register, even if the states in question are short lived and not true at the moment of the query.

Like the condition register, a query of the event register returns a single number which is the binary sum of all bits in the register which are set. (See the discussion of the condition register above for a detailed explanation of this process.)

Unlike the condition register, a query of the event register clears any bits which were previously set. Executing the \*CLS command will also clear this register. Of course if corresponding bits in the condition register are still set, these bits will immediately be set again after the clear from the query.

## Enable Register

The enable register is a mask register that controls which bits from the event register will set the overall summary bit. If a bit in the enable register is set, then the corresponding bit in the event register will be combined with other enabled bits of the event register via a logical OR operation to create an overall summary bit.

The user sets the enable register with a single number which is the binary sum of all bits in the register which should be set. (See the discussion of the condition register above for a detailed explanation of this process.) Continuing with the example from Figure 83, If the enable register is set to  $9 = 1 + 8$ , then the summary bit will be set if either bit 0 or bit 3 of the event register is set. Therefore, if the user detects that the summary bit is set,

he can infer that either State 1 or State 2 or both were at least momentarily true since the last query of the event register.

Enable bits are set via a command. They are not cleared by a query or the execution of the \*CLS command. To clear enabled bits, the user must send another set command with those bits set to zero. Sending the number zero will clear all bits of the enable register and prevent the summary bit from ever being set.

## FS740 Status

The FS740 status is reported through the standard event register and three instrument status registers. The organization and hierarchy of these registers is depicted in Figure 84. There are three instrument status registers: GPS receiver status, questionable status, and operational status. These are all 16-bit registers which report on the status of the instrument and its operation. The 8-bit standard event register reports on the status of command parsing and execution. The summary bits from each of these registers feed into a single, 8-bit condition register called the serial poll status byte. Summary bits for the error queue and the output buffer also feed into this status byte. The serial poll status byte, therefore, provides summary status for the entire instrument.

The serial poll status byte (\*STB) may be configured to generate a request for service interrupt to the remote interface, if desired. This is achieved with the help of an enable register (\*SRE) associated with it that allows bits in the serial poll status byte to be combined into a single summary bit, also in the serial poll status byte located at bit 6. When this bit is set, the instrument will generate a request for service interrupt to the remote interface if supported.

### Serial Poll Status Byte

The serial poll status byte provides summary status for the instrument as a whole. The interpretation for bits in the serial poll status byte is shown in **Error! Reference source not found.**

**Table 6: Interpretation of serial poll status bits**

Bit	Name	Meaning
0		
1	GPS	An unmasked bit in the GPS receiver status has been set.
2	ERR	There is at least one error in the error queue. Query the error with the command SYST:ERR?
3	QUES	An unmasked bit in the QUES status register has been set.
4	MAV	The interface output buffer has at least one character in it. Perform a read of instrument to retrieve it.
5	ESR	An unmasked bit in the standard event status register (*ESR) has been set.
6	MSS	Master summary bit. Indicates that the instrument is requesting service because an unmasked bit in this register has been set.
7	OPER	An unmasked bit in the OPER status register has been set.

The serial poll status byte may be queried with the \*STB? command. The service request enable register (\*SRE) may be used to control when the instrument asserts a request-for-service on interfaces where that is supported.

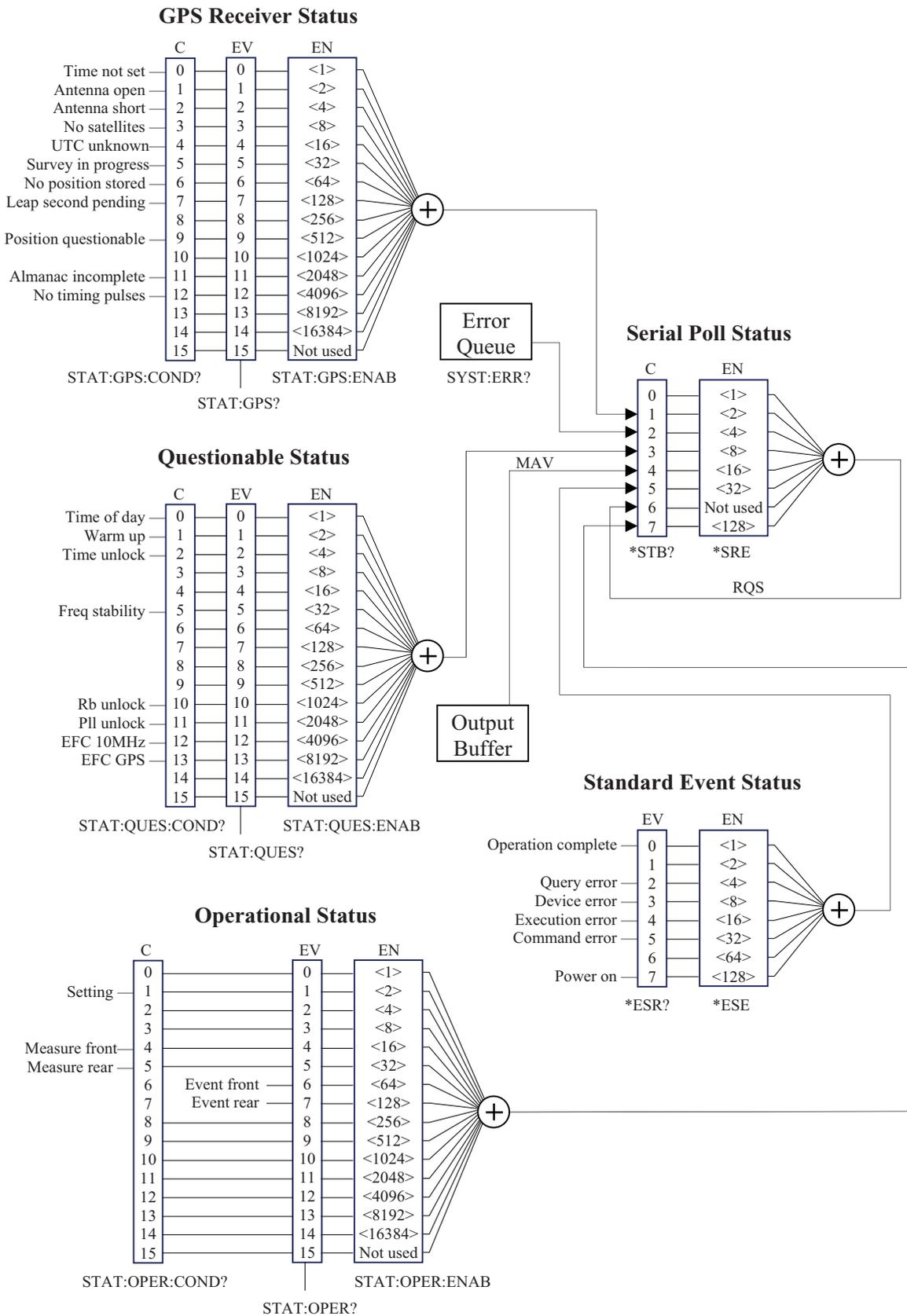


Figure 84: FS740 Status reporting

## Standard Event Status Register

The standard event register provides status information on command parsing and execution. The interpretation for bits in the standard event status register is shown in Table 7.

**Table 7: Interpretation of standard event status bits**

Bit	Name	Meaning
0	OPC	Operation complete. All previous commands have completed. See command *OPC.
1		
2	QYE	Query error occurred.
3	DDE	Device dependent error occurred.
4	EXE	Execution error. A command failed to execute correctly because a parameter was invalid.
5	CME	Command error. The parser detected a syntax error.
6		
7	PON	Power on. The unit has been power cycled.

The standard event status register may be queried with the \*ESR? command. The standard event status enable register (\*ESE) may be used to control the setting of the ESR summary bit in the serial poll status byte (\*STB).

## Questionable Status

Bits in the questionable status register warn the user when the validity of some instrument function is in question. For example, bit 0 of the questionable status register will be set until the GPS receiver has successfully decoded the time of day from the GPS satellites. Until this happens, all time of day information reported by the FS740 is invalid. Other bits in the questionable status register provide information on the frequency stability of the timebase.

The interpretation for bits in the questionable status register is shown in Table 8

**Table 8: Interpretation of questionable status bits**

Bit	Name	Meaning
0	Time of day	Instrument time of day has not been set by the GPS receiver. Absolute time measurements are invalid.
1	Warm up	The timebase is still warming up. Frequency drift will be much larger than normal.
2	Time unlock	The timebase is not locked to GPS. Time and frequency measurements may be degraded.
3		
4		
5	Freq stability	The timebase has not been locked to GPS long enough to reach optimum frequency stability.
6		
7		
8		
9		
10	Rb unlock	The installed Rb timebase is unlocked. Its frequency is not stable
11	PLL unlock	One of the internal PLL circuits in the FS740 has come unlocked. This may signal a need for instrument repair.
12	EFC 10MHz	Indicates that the frequency control of the internal TCXO is near the rail.
13	EFC GPS	Indicates that the frequency control for the installed timebase is saturated. This might indicate a large timing error.
14		
15		

The condition, event, and enable registers for questionable status are queried using the following commands, respectively.

```
STAT:QUES:COND?
STAT:QUES?
STAT:QUES:ENAB?
```

## Operation Status

Bits in the operational status register provide information on the operation of the instrument. For example, bit 1 of the operational status register indicates that a hardware setting in the FS740 has changed. Most settings change quickly, so this bit will normally only be detected via the event register.

The interpretation for bits in the operational status register is shown in Table 9

**Table 9: Interpretation of operation status bits**

Bit	Name	Meaning
0		
1	Setting	Hardware instrument settings are changing
2		
3		
4	Measure front	Measurement on front input in progress
5	Measure rear	Measurement on rear input in progress
6	Event front	Timing event detected on front input.
7	Event rear	Timing event detected on rear input.
8		
9		
10		
11		
12		
13		
14		
15		

Bits 4 and 5 indicate that a measurement is in progress on the given input. Both bits 4 and 5 must be false before the command \*OPC? returns 1.

Bits 6 and 7 will only be set if the input is configured for a timing measurement, CONF:TIM, and a successful time tag is recorded on the configured input.

The condition, event, and enable registers for operational status are queried using the following commands, respectively.

```
STAT:OPER:COND?
STAT:OPER?
STAT:OPER:ENAB?
```

## GPS Receiver Status

Bits in the GPS receiver status register provide information on the status of the GPS receiver and its ability to track the GPS satellites.

The interpretation for bits in the GPS receiver status register is shown in Table 10

**Table 10: Interpretation of GPS receiver status bits**

Bit	Name	Meaning
0	Time not set	Time of day information has not been received from the satellites, yet.
1	Antenna open	The GPS antenna does not appear to be connected to the receiver input.
2	Antenna short	The receiver input appears to be shorted.
3	No satellites	No GPS satellites have been found
4	UTC unknown	The UTC offset from GPS is unknown. The offset is recorded in the almanac which can take up to 15 minutes to download from the satellites.
5	Survey in progress	A position survey is in progress.
6	No position stored	No surveyed position has been stored in nonvolatile memory.
7	Leap second pending	A leap second is pending. When pending, they are normally scheduled for the end of the day on June 30 <sup>th</sup> or December 31 <sup>st</sup> .
8		
9	Position questionable	The stored position does not appear to be correct according to data now being collected. A new survey may need to be collected.
10		
11	Almanac incomplete	A complete almanac has not been downloaded from the GPS satellites, yet. It can take up to 15 minutes of continuous tracking of satellites to download the almanac.
12	No timing pulses	Timing pulses are not being generated by the receiver. Timing pulses must be generated in order for the FS740 to lock its timebase to GPS.
13		
14		
15		

The condition, event, and enable registers for GPS receiver status are queried using the following commands, respectively.

```
STAT:GPS:COND?
STAT:GPS?
STAT:GPS:ENAB?
```

## Making Measurements

The FS740 can make frequency and timing measurements of an external signal source. The front and rear inputs have independent circuitry enabling simultaneous measurements. All the measurement commands take an optional numerical suffix of 1 or 2. The value 1 refers to the front input. The value 2 refers to the rear input. If omitted, the front input is assumed. The SCPI language provides several ways to configure and initiate measurements, which vary in complexity and control.

### Simple Measurements

The MEASure commands provide the simplest method of quickly configuring a measurement and reading the result in one command.

#### Syntax

```
MEASure[{1|2}]:FREQuency? [{<expected>|MINimum|MAXimum|DEFAULT}
    [, {<resolution>|MINimum|MAXimum|DEFAULT}]]
MEASure[{1|2}]:TIMe?
```

#### Description

The first version configures a frequency measurement with default parameters. The second version configures a measurement of time. In both cases the measurement is triggered immediately and the result sent directly to the output buffer. The frequency measurement takes two optional parameters. The first is an expected frequency for the measurement. If omitted, 10 MHz is assumed. The second is the desired resolution of the measurement. If omitted, a resolution is selected such that the gate for the measurement will be 0.1 s resulting in about 11 digits of precision.

#### Examples

To make single measurements of frequency on the front and rear inputs, use the following commands:

```
MEAS:FREQ?
MEAS2:FREQ?
```

Similarly, to make single measurements of time on the front and rear inputs, use the following commands:

```
MEAS:TIM?
MEAS2:TIM?
```

## Configure and Read

The MEASure commands configure a measurement and trigger it immediately. This does not allow the user to refine the measurement configuration. To enable finer control of the configuration, use the CONFIgure and READ commands. The CONFIgure commands have the same syntax and meaning as the MEASure commands discussed above, except that they are set commands rather than queries, reflecting the fact that the measurement is not triggered, and so no results will be forthcoming to read. The hardware for a given measurement is configured with default settings. The user should send additional commands to customize the configuration before the measurement is triggered. Once the configuration is complete, the user can trigger a measurement and read the result with the READ command.

**Syntax**

```

CONFigure[{1|2}]:FREQ [ {<expected>|MINimum|MAXimum|DEFault}
    [ , {<resolution>|MINimum|MAXimum|DEFault} ] ]
CONFigure[{1|2}]:TIME
READ[{1|2}]?

```

**Examples**

To configure a frequency measurement on the front input with a 1 second gate, use the following commands:

```

CONF:FREQ
SENS:FREQ:GATE 1.0
READ?

```

The first line configures the front input for a frequency measurement with default settings. The second line changes the gate from the default to 1.0 seconds. The last line initiates the measurement and sends it to the output buffer for reading.

The next example configures the rear input for 10 timing measurements and triggers a measurement:

```

CONF2:TIM
SAMP2:COUN 10
READ2?

```

The first line configures the rear input for an absolute time measurement. The second line changes the sample size of the measurement to 10. The last line initiates the measurement and sends the results to the output buffer for reading.

**Initiate and Fetch**

The READ command discussed above triggers a measurement and sends the results immediately to the output. This is the desired result in most cases. However, sometimes it is useful to separate the initiation of the measurement and the reading of the results. This is achieved with the INITiate and FETCh commands. The initiate triggers a measurement but does not send the results to the output buffer. Instead they are sent to an internal memory buffer, which has enough space to hold the last 250,000 results. The FETCh command causes the results to be copied from the internal memory to the output buffer. A READ command is equivalent to an INITiate followed immediately by a FETCh.

**Syntax**

```

INITiate[{1|2}][:IMMediate]
FETCh[{1|2}]?

```

Separating measurement initiation from the reading of the results is particularly useful when the measurement takes a long time or the sample size is large. In the former case, separating the INITiate from the FETCh enables the user to query the status registers to monitor the progress of the measurement without having to depend on long timeouts and an unresponsive unit. In the latter case when the sample size is large, it may be that the user is uninterested in the individual results and only wants to query the statistics for the measurement. In this case the FETCh command need not be sent at all. Instead, the CALC:STAT command is sent after all measurements complete.

Like the READ command, the FETCh command won't complete until all samples in the measurement have completed. For large data sets this may be undesirable. In this case the user can use the DATA subsystem to query how much data is available and read only a portion of the data at a time.

## Common IEEE-488.2 Commands

---

**\*CLS** **Clear Status**

### Syntax

\*CLS

### Description

This command immediately clears all status registers as well as the SYST:ERR queue.

---

**\*ESE** **Standard Event Status Enable**

### Syntax

\*ESE <value>  
\*ESE?

### Description

Set the Standard Event Status Enable register to <value>. The value may range from 0 to 255. Bits set in this register cause ESR (in \*STB) to be set when the corresponding bit is set in the \*ESR register. The query returns the current value of the enable register. Definitions for the bits in the standard event register are given on page 81.

### Example

\*ESE 1

Enable bit 0 so that an operation complete event will set the ESR bit in the serial poll status byte.

---

**\*ESR?** **Standard Event Status Register**

### Syntax

\*ESR?

### Description

Query the Standard Event Status Register. After the query, the returned bits of the \*ESR register are cleared. The bits in the ESR register have the following meaning:

<u>Bit</u>	<u>Meaning</u>
0	OPC – operation complete
1	Reserved
2	QYE – query error
3	DDE – device dependent error
4	EXE – execution error
5	CME – command error
6	Reserved
7	PON – power-on

See page 81 for more detailed information on the event status register.

### Example

\*ESR?

A return of '176' would indicate that PON, CME, and EXE are set.

**\*IDN?** **Identification String**

**Syntax**

\*IDN?

**Description**

Query the instrument identification string.

**Example**

\*IDN?

Returns a string similar to ‘Stanford Research Systems,FS740,s/n004025,ver2.26.11’

**\*OPC** **Operation Complete**

**Syntax**

\*OPC  
\*OPC?

**Description**

The set form sets the OPC flag in the \*ESR register when all prior commands have completed. The query form returns ‘1’ when all prior commands have completed, but does not affect the \*ESR register.

**\*OPT?** **Operation Complete**

**Syntax**

\*OPT?

**Description**

The query returns a comma separated list of the four possible installed options in the following order: top rear panel board, middle rear panel board, bottom rear panel board, and the installed timebase. They may take on the following values:

Type	Option	Value
Board	Not installed	0
	10 MHz distribution	1
	Sine/Aux output distribution	2
	Pulse output distribution	3
Timebase	TCXO	4
	OCXO	5
	Rb	6

**Example**

\*OPT?

The query returns the current installed options. A return of “1,0,0,5” would indicate that a 10 MHz distribution board is installed in the top slot, but that the middle and bottom slots are empty, and that the installed timebase is an OCXO.

**\*PSC** **Power-on Status Clear**

**Syntax**

```
*PSC <value>
 *PSC?
```

**Description**

Set the Power-on Status Clear flag to <value>. The Power-on Status Clear flag is stored in nonvolatile memory in the unit, and thus, maintains its value through power-cycle events.

If the value of the flag is 0, then the Service Request Enable and Standard Event Status Enable Registers (\*SRE, \*ESE) are stored in non-volatile memory, and retain their values through power-cycle events. If the value of the flag is 1, then these two registers are cleared upon power-cycle.

**Example**

Use the following commands to set power on status clear to 1 and then query the setting.

```
*PSC 1
 *PSC?
```

**\*RCL****Recall Instrument Settings****Syntax**

```
*RCL <location>
```

**Description**

Recall instrument settings from <location>. The <location> may range from 0 to 9. Locations 1 to 9 are for arbitrary use. Location 0 is reserved for the recall of default instrument settings.

**Example**

```
*RCL 3
```

Recall instruments settings from location 3.

**\*RST****Reset Instrument****Syntax**

```
*RST
```

**Description**

Reset the instrument to default settings. This is equivalent to \*RCL 0. See Factory Default Settings on page 66 for a list of default settings.

**\*SAV****Save Instrument Settings****Syntax**

```
*SAV <location>
```

**Description**

Save instrument settings to <location>. The <location> may range from 0 to 9. However, location 0 is reserved for current instrument settings. It will be overwritten after each front panel key press.

**Example**

```
*SAV 3
```

Save current instrument settings to location 3.

**\*SRE** **Service Request Enable**

**Syntax**

\*SRE <value>  
\*SRE?

**Description**

Set the Service Request Enable register to <value>. Bits set in this register cause the FS740 to generate a service request when the corresponding bit is set in the serial poll status register, \*STB.

**Example**

\*SRE 16

Set bit 4 of the enable register. This will generate a service request when the FS740 has bytes in its output buffer ready to be read. Definitions for the bits in the serial poll status byte are given on page 79.

**\*STB?** **Status Byte**

**Syntax**

\*STB?

**Description**

Query the standard IEEE 488.2 serial poll status byte. The bits in the STB register have the following meaning:

<u>Bit</u>	<u>Meaning</u>
0	Reserved
1	GPS status summary bit
2	Error queue is not empty
3	Questionable status summary bit
4	Message available, MAV.
5	ESR status summary bit
6	MSS – master summary bit
7	Operational status summary bit

See page 79 for more detailed information on the serial poll status byte.

**Example**

\*STB?

A return of '114' would indicate that GPS, MAV, ESR, and MSS are set. GPS indicates that an enabled bit in STAT:GPS is set. MAV indicates that a message is available in the output queue. ESR indicates that an enabled bit in the \*ESR is set. MSS reflects the fact that at least one of the summary enable bits is set and the instrument is requesting service.

**\*WAI** **Wait for Command Execution**

**Syntax**

\*WAI

**Description**

The instrument will not process further commands until all prior commands including this one have completed.

**Example**

\*WAI

Wait for all prior commands to execute before continuing.

## Measurement Commands

Measurement commands can apply to either the front or rear input. The user selects the input by optionally appending a 1 or 2 to the root keyword of the command. When the suffix is 1 or omitted, the front input is selected. When the suffix is 2, the rear input is selected. For example, the following commands all apply to the front input:

```
MEAS:FREQ?
CONF1:TIM
READ1?
CONF:FREQ
INIT1
FETC?
```

Notice that the suffix is optional when referring to the front input. Modify the commands as follows to make them apply to the rear input:

```
MEAS2:FREQ?
CONF2:TIM
READ2?
CONF2:FREQ
INIT2
FETC2?
```

Here, the suffix is required.

---

### MEASure:FREQuency?

### Measure Frequency

#### Syntax

```
MEASure[{1|2}]:FREQuency? [{<expected>|MINimum|MAXimum|DEFAULT}
[, {<resolution>|MINimum|MAXimum|DEFAULT}]]
```

#### Description

Configures hardware for a frequency measurement and immediately triggers a measurement and sends the result to the output buffer. The first parameter is optional and informs the instrument of the expected frequency of the signal. The second parameter is also optional. It sets the requested resolution of the measurement. Neither parameter is used directly by the FS740. Instead the ratio of the resolution to the expected frequency is used to set the gate time for the measurement. If the parameters are omitted, a gate time of 0.1 second is used which corresponds to approximately 11 digits of precision. Measurements are returned as floating point values in units of Hz. If a measurement times out NAN is returned.

#### Example

```
MEAS:FREQ?
```

Configure a frequency measurement on the front panel input, using the default 0.1 s gate. Trigger the measurement immediately and send the result to the output buffer for reading.

---

### MEASure:TIme?

### Measure Time

#### Syntax

```
MEASure[{1|2}]:TIme?
```

#### Description

Configures hardware for a measurement of time and immediately triggers a measurement and sends the result to the output buffer. A single result consists of 11 comma separated integer values in the

following order: timing metric, year, month, day, hour, minute, seconds, milliseconds, microseconds, nanoseconds, picoseconds. The timing metric is a copy of the questionable status register at the time of the measurement.

### Example

```
MEAS2:TIM?
```

Configure a measurement of time on the rear panel input. Trigger the measurement immediately and send the result to the output buffer for reading.

---

## CONFigure?

## Configure

### Syntax

```
CONFigure[{{1|2}}?
```

### Description

Query the current measurement configuration. The command returns a quoted string with the measurement type and parameters used to configure the measurement.

### Example

```
CONF:FREQ
CONF?
```

Configure a measurement of frequency on the front panel input. The query will return:  
FREQ 1.0e+07,1.0e-04.

---

## CONFigure:FREQuency

## Configure Frequency

### Syntax

```
CONFigure[{{1|2}}:FREQuency [ {<expected>|MINimum|MAXimum|DEFault}
[, {<resolution>|MINimum|MAXimum|DEFault} ] ]
```

### Description

Configure the hardware for a frequency measurement. This is identical to the MEASure command except that the measurement isn't triggered and sent to the output buffer. Like the MEASure command, the first parameter is optional and informs the instrument of the expected frequency of the signal. The second parameter is also optional. It sets the requested resolution of the measurement. Neither parameter is used directly by the FS740. Instead the ratio of the resolution to the expected frequency is used to set the gate time for the measurement. If the parameters are omitted, a gate time of 0.1 second is used which corresponds to approximately 11 digits of precision. Measurement results are returned with a READ command as floating point values in units of Hz. If a measurement times out, NAN is returned.

### Example

```
CONF:FREQ
SENS:FREQ:GATE 1.0
READ?
```

Configure a frequency measurement on the front panel input, using the default 0.1 s gate. Then change the gate to 1.0 s. Finally trigger a measurement and read the result.

---

## CONFigure:TIME

## Configure Time

**Syntax**

```
CONFigure[{1|2}]:TIME
```

**Description**

Configures hardware for a measurement of time and immediately triggers a measurement and sends the result to the output buffer. Results are returned with a READ command. A single result consists of 11 comma separated integer values in the following order: timing metric, year, month, day, hour, minute, seconds, milliseconds, microseconds, nanoseconds, picoseconds. The timing metric is a copy of the questionable status register at the time of the measurement.

**Example**

```
CONF2:TIM
SAMP2:COUN 10
READ2?
```

Configure a measurement of time on the rear panel input. Change the sample count from 1 to 10. Finally trigger the measurement and read the results.

**READ?****Read Results****Syntax**

```
READ[{1|2}]?
```

**Description**

Trigger a measurement using the current configuration and read the result. See commands MEASure:FREQuency and MEASure:TIME for details on the format of results returned. When the sample size is greater than one, results are separated from each other by commas ( , ).

**Example**

```
CONF:FREQ
SENS:FREQ:GATE 1.0
READ?
```

Configure a measurement of frequency on the front panel input. Then change the gate time for the measurement to 1.0 s. Finally trigger the measurement and read the result.

**INITiate****Initiate Measurement****Syntax**

```
INITiate[{1|2}][:IMMediate]
```

**Description**

Trigger a measurement using the current configuration but leave the results in internal memory. The internal memory has enough space to store the last 250,000 measurements. The user should send the FETCh command to retrieve the results from internal memory. Alternatively, the user may use commands in the DATA subsystem to read only a portion of the results.

**Example**

```
CONF:FREQ
SAMP:COUN 1000
INIT
FETC?
```

Configure a measurement of frequency on the front panel input using default settings. Then change the sample count from 1 to 1000. Then initiate a measurement. Finally read out the results.

**FETCh?****Fetch Results****Syntax**

```
FETCh[{1|2}]?
```

**Description**

Copy results stored in the internal buffer to the output buffer for reading. Like the READ command, this command will not complete until all measurements have completed. See commands MEASure:FREQuency and MEASure:TIME for details on the format of results returned. When the sample size is greater than one, results are separated from each other by commas ( , ).

**Example**

```
CONF:FREQ
SAMP:COUN 1000
INIT
FETC?
```

Configure a measurement of frequency on the front panel input using default settings. Then change the sample count from 1 to 1000. Next initiate a measurement. Finally read out the results, which should contain 1000 frequency measurements.

**ABORt****Abort Measurement****Syntax**

```
ABORt[{1|2}]
```

**Description**

Abort any measurement in progress and discard any results produced by it.

**Example**

```
ABOR
```

Abort any measurement initiated on the front input.

**STOP****Stop Measurement****Syntax**

```
STOP[{1|2}]
```

**Description**

Stop any measurement in progress, but do not discard any results produced by it.

**Example**

```
CONF:FREQ
SAMP:COUN 1000
INIT
STOP
FETC?
```

Configure the front input for a frequency measurement. Change the sample count to 1000. Initiate the measurement. Stop the measurement but do not discard any results already produced. Fetch those results

## Calculate Subsystem

Commands in the Calculate Subsystem can apply to either the front or rear input. The user selects the input by optionally appending a 1 or 2 to the CALCulate keyword. When the suffix is 1 or omitted, the front input is selected. When the suffix is 2, the rear input is selected.

---

### CALCulate:FILTer

### Calculation Filter

#### Syntax

```
CALCulate[{{1|2}}]:FILTer {NONE|FAST}
CALCulate[{{1|2}}]:FILTer?
```

#### Description

The first definition changes input filter for frequency measurements. The second definition queries the current input filter. There are two filter options: NONE and FAST. If NONE is selected, then just two time tags are used to generate a frequency measurement, one at the beginning of the gate interval and one at the end. When the FAST filter is selected, up to 625 time tags are averaged together at the beginning of the gate interval to produce an average starting time tag. Another 625 time tags are averaged together at the end of the gate interval to produce an average ending time tag. These two averaged time tags are used to compute a frequency measurement. The benefit of this filter is that it is effective in removing broadband noise inherent in the measurement. Noise can be reduced by more than a factor of 10 with the use of this filter. The default filter is FAST and it is the recommended setting for most measurements. The following example shows how to turn off the filter, if desired.

#### Example

```
CONF:FREQ
CALC:FILT NONE
READ?
```

Configure front input for a frequency measurement. Turn off the FAST filter. Trigger a measurement and read the results.

---

### CALCulate:REFErency

### Calculation Reference Frequency

#### Syntax

```
CALCulate[{{1|2}}]:REFErency [{{<frequency>|MINimum|MAXimum|DEFault}}]
CALCulate[{{1|2}}]:REFErency?
```

#### Description

For each frequency measurement, the reference frequency is subtracted from the measured frequency to produce the final result. This enables one to monitor the deviation of the frequency from a specified target value rather than the absolute frequency itself. The first definition sets the reference frequency to <frequency>. If <frequency> is omitted, it is set to the value of the most recent measurement. The second definition queries the current reference frequency. The default reference frequency is 0 Hz.

**Example**

```
CONF:FREQ
CALC:REF 10MHz
READ?
```

Configure front input for a frequency measurement. Set the reference frequency to 10 MHz. Measure the deviation of the input from 10MHz.

**CALCulate:STABility****Calculate Frequency Stability****Syntax**

```
CALCulate[{{1|2}}]:STABility[:ALL]?
```

**Description**

This command computes the frequency stability, or Allan deviation, of all frequency measurements for time intervals from 10 ms to 50 million seconds in a 1, 2, 5 sequence and returns them as a comma delimited list of relative Allan deviations. The first is the computed stability for a 10 ms interval. The second is for a 20 ms interval. The third is for a 50 ms interval and so forth. Each succeeding stability measurement is for an interval that increases in a 1, 2, 5 sequence with the 30<sup>th</sup> value corresponding to an interval of 50 million seconds. If the data set is not large enough to compute a valid stability for a given interval, zero is returned for that entry. Note that the FS740 is computing averaged time tags every 10 ms with no dead time, regardless of the configured gate. This enables the FS740 to compute the frequency stability for all time intervals, not just the configured gate interval. Furthermore, the user need not wait for the measurement to complete before sending this command. It will return the computed stability of all measurements completed at the time of the query.

**Example**

```
CONF:FREQ
SENS:FREQ:GATE 1.0
SAMP:COUN 1000000
INIT
...
CALC:STAB?
```

Configure front input for a frequency measurement. Set the gate to 1.0 s and the sample size to 1 million. Start the measurement. Wait. Then query the computed frequency stability of the measurements. Note that one need not wait for the measurement to complete. The FS740 will return the computed stability for all measurements completed so far.

**CALCulate:STATistics****Calculate Measurement Statistics****Syntax**

```
CALCulate[{{1|2}}]:STATistics[:ALL]?
```

**Description**

This command computes some basic statistics of the current measurement and returns the following values in a comma separated list: the mean, the Allan deviation, the minimum, the maximum, and the number of measurements made. The statistical values returned are in Hz, including the Allan deviation. To get a relative stability, divide the returned Allan deviation by the mean. The Allan deviation returned is only for the configured gate interval, and is computed only from the measurements stored in internal memory. This is in contrast to the results returned by the CALC:STAB command which incorporates many more time tags not included in the frequency measurements stored in internal memory.

**Example**

```

CONF:FREQ
SENS:FREQ:GATE 1.0
SAMP:COUN 1000
INIT
...
CALC:STAT?

```

Configure front input for a frequency measurement. Set the gate to 1.0s and the sample size to 1000. Start the measurement. Wait. Then query the statistics for the measurements. Note that one need not wait for the measurement to complete. The FS740 will return the computed statistics for all measurements completed so far.

## Data Subsystem

FS740 has enough internal memory to store 250,000 results for both the front and rear input. The Data Subsystem enables the user to query and process these results. Commands in the Data Subsystem can apply to either the front or rear input. The user selects the input by optionally appending a 1 or 2 to the DATA keyword. When the suffix is 1 or omitted, the front input is selected. When the suffix is 2, the rear input is selected.

---

**DATA:COUNT?****Data Count****Syntax**

```
DATA[{{1|2}}]:COUNT?
```

**Description**

This command returns the total number of measurements completed so far.

**Example**

```

CONF:FREQ
SAMP:COUN 100
INIT
...
DATA:COUN?

```

Configure front input for a frequency measurement. Set the sample size to 100. Start the measurement. Wait. Then query the number of measurements completed for the front input. When all measurements are complete, the returned value should agree with the configured sample count.

---

**DATA:POINTs?****Data Points****Syntax**

```
DATA[{{1|2}}]:POINTs?
```

**Description**

This command returns the total number of measurements stored in internal memory. This number may differ from the DATA:COUNT if some data has been removed with a DATA:REM command or if the total number of measurements causes the internal memory to overflow. In the latter case, only the most recent measurements are kept. The internal memory has space to store up to 250,000 measurements.

**Example**

```
CONF2:FREQ
```

```
SAMP2:COUN 100
INIT2
...
DATA2:POIN?
```

Configure rear input for a frequency measurement. Set the sample size to 100. Start the measurement. Wait. Then query the number of measurements stored in internal memory for the rear input. These values may be read immediately using a DATA:READ or DATA:REM command.

---

**DATA:READ?**
**Data Read****Syntax**

```
DATA[{1|2}]:READ? [<index>][,<count>]
```

**Description**

This command returns <count> measurements stored in internal memory, starting with the one located at <index>. The first measurement in memory has an <index> of 0. The second has an <index> of 1, and so on. The user should send the DATA:POIN command to determine how many measurements are stored in memory. If omitted, <index> is assumed to be 0. If omitted, <count> is assumed to be 1. If <index> or <count> is out of range, error -222, "Data out of range," is generated and no data returned. See commands MEASure:FREQuency and MEASure:TIME for details on the format of results returned. When the sample size is greater than one, results are separated from each other by commas ( , ).

**Example**

```
CONF2:FREQ
SAMP2:COUN 100
INIT2
...
DATA2:POIN?
DATA2:READ? 10, 5
```

Configure rear input for a frequency measurement. Set the sample size to 100. Start the measurement. Wait. Then query the number of measurements stored in internal memory for the rear input. Retrieve 5 measurements starting with the measurement at index 10.

---

**DATA:REMOve?**
**Data Remove****Syntax**

```
DATA[{1|2}]:REMOve? <count>
```

**Description**

This command returns the first <count> measurements stored in internal memory and removes them from memory. If the requested <count> exceeds the amount of data available, error -222, "Data out of range," is generated and no data is returned. This command is useful if the configured sample count is larger than can be stored in internal memory. It enables the user to retrieve measurements as they become available to free up space in internal memory so that no measurement results are lost. See commands MEASure:FREQuency and MEASure:TIME for details on the format of results returned. When the sample size is greater than one, results are separated from each other by commas ( , ).

**Example**

```
CONF:FREQ
SAMP:COUN 1000000
INIT
```

```
...
DATA:POIN?
DATA:REM? 100
```

Configure front input for a frequency measurement. Set the sample size to 1 million. Start the measurement. Wait. Then query the number of measurements stored in internal memory for the front input. Retrieve the first 100 measurements and remove them from memory.

## GPS Subsystem

Commands in the GPS Subsystem enable configuration of the GPS receiver and report on its operation.

---

### GPS:CONFig:MODe

### GPS Configure Mode

#### Syntax

```
GPS:CONFig:MODe <anti-jamming>, <elevation mask>, <signal mask>
GPS:CONFig:MODe?
```

#### Description

The first definition enables the user to set anti-jamming mode, the elevation mask and the signal mask. The second definition queries the current values for these parameters. <Anti-jamming> is a Boolean value which enables or disables anti-jamming in the receiver. The factory default is enabled. <Elevation mask> is the elevation angle in radians, below which satellites are ignored in over determined clock mode. It can range from 0 to  $\pi/2$ . <Signal mask> is the minimum signal level in dBHz, below which satellites are ignored in over determined clock mode. It can range from 0 to 55 dBHz. The default value for both masks is 0. The user must execute the command GPS:CONF:SAVE to save the current values to nonvolatile memory.

#### Example

```
GPS:CONF:MODE 1,0.2618,10
```

Enable anti-jamming. Set elevation mask to 15 degrees = 0.2618 radians. Set the minimum signal level to 10 dBHz. These limits only apply in over determined clock mode.

---

### GPS:CONFig:SAVe

### GPS Configure Save

#### Syntax

```
GPS:CONFig:SAVe
```

#### Description

The user uses commands in the GPS:CONFIG subsystem to configure the operation of the receiver. The configuration is lost, however, if the power is cycled unless this command is executed. It saves the current GPS receiver configuration to nonvolatile memory, so that it may be automatically recalled when the power is cycled.

#### Example

```
GPS:CONF:SAV
```

Save the current GPS receiver configuration to nonvolatile memory.

---

### GPS:CONFig:SURVey:Mode

### GPS Configure Survey Mode

**Syntax**

```
GPS:CONFig:SURVey[:MODE] [{DISabled|REDo|REMember}]
GPS:CONFig:SURVey[:MODE]?
```

**Description**

The first definition sets the GPS survey mode to the given configuration. If mode is omitted, the default mode is REMember. The second definition queries the current GPS survey mode. When DISabled is selected no survey is carried out. This mode is appropriate in mobile environments. REDo causes the survey to be repeated each time the instrument is power cycled. REMember causes the results of the survey to be saved in nonvolatile memory. When the instrument is power cycled, the surveyed position is recalled from memory and the survey is not repeated. The user must execute the command GPS:CONF:SAVE to save the current value to nonvolatile memory.

**Example**

```
GPS:CONF:SURV REM
```

Configure the GPS receiver to do a survey at startup the first time, but to remember the results of the first survey from then on rather than repeating the survey.

**GPS:CONFig:SURVey:FIXes****GPS Configure Survey Fixes****Syntax**

```
GPS:CONFig:SURVey:FIXes [<fixes>]
GPS:CONFig:SURVey:FIXes?
```

**Description**

The first definition sets the number of position fixes in the survey. If <fixes> is omitted, the default number is 2000. The second definition queries the current number of position fixes in the position survey. This is the number of position fixes that get averaged together to define the GPS antennas location. Once the position is well known, the receiver can be put into over determined clock mode and the receiver can provide improved timing by dedicating the signal from all satellites to timing. The user must execute the command GPS:CONF:SAVE to save the current value to nonvolatile memory.

**Example**

```
GPS:CONF:SURV:FIX 3000
```

Configure the GPS receiver to include 3000 position fixes in the position survey.

**GPS:CONFig:ALIGnment****GPS Configure Timing Alignment****Syntax**

```
GPS:CONFig[:TIMing]:ALIGnment [{GPS|UTC}]
GPS:CONFig[:TIMing]:ALIGnment?
```

**Description**

The first definition sets the GPS 1pps alignment. If alignment is omitted, the default alignment is UTC. The second definition queries the current GPS 1pps alignment. When GPS is selected, all timing is aligned to GPS. When UTC is selected, all timing is aligned to UTC. The user must execute the command GPS:CONF:SAVE to save the current value to nonvolatile memory.

**Example**

```
GPS:CONF:ALIG UTC
```

Alignment of the GPS 1pps is to UTC.

---

**GPS:CONFig:QUALity**
**GPS Configure Timing Quality**
**Syntax**

```
GPS:CONFig[:TIMing]:QUALity [{1SAT|3SAT}]
GPS:CONFig[:TIMing]:QUALity?
```

**Description**

The first definition sets the minimum number of satellites the receiver must track before outputting a hardware 1pps pulse. If omitted, the default quality is 3 satellites. The second definition queries the current timing quality. Timing quality generally increases as the number of satellites increases. However, the user must also consider reliability and holdover performance. Degraded performance may be preferred over no timing whatsoever. The user must execute the command GPS:CONF:SAVE to save the current value to nonvolatile memory.

**Example**

```
GPS:CONF:QUAL 3SAT
```

Require the GPS receiver to track 3 satellites before outputting a hardware 1pps pulse for the timebase to track.

---

**GPS:CONFig:ADELay**
**GPS Configure Timing Antenna Delay**
**Syntax**

```
GPS:CONFig[:TIMing]:ADELay <delay>
GPS:CONFig[:TIMing]:ADELay?
```

**Description**

The first definition sets the antenna delay to <delay> in seconds. The second definition queries the current antenna delay in seconds. The <delay> may range from  $-0.1$  s to  $+0.1$  s. Note that the user should enter a negative number here to compensate for the length of their antenna cable. It affects the timing of all inputs and outputs. The user must execute the command GPS:CONF:SAVE to save the current value to nonvolatile memory.

**Example**

```
GPS:CONF:ADEL -100ns
```

Set delay to  $-100$  ns. This advances the timing generated by the GPS receiver by 100 ns.

---

**GPS:POSition?**
**GPS Position**
**Syntax**

```
GPS:POSition?
```

**Description**

Query the GPS position. This routine returns the surveyed average position of the receiver in latitude, longitude, and altitude. Latitude is specified in radians, with positive values indicating north, and negative values indicating south. Longitude is specified in radians, with positive values indicating east, and negative values indicating west. Altitude is specified in meters above average sea levels.

**Example**

```
GPS:POS?
```

Query the current surveyed position of the GPS receiver.

---

**GPS:POSition:HOLD:STATe?****GPS Position Hold State****Syntax**

```
GPS:POSition:HOLD:STATe?
```

**Description**

Query whether the GPS receiver is in position hold mode where all satellites are being used for maximum timing performance. The query will return 1 if the receiver is in position hold mode, otherwise 0. The receiver typically enters position hold mode once the position survey is complete.

**Example**

```
GPS:POS:HOLD:STAT?
```

Query whether the GPS receiver has entered position hold mode.

---

**GPS:POSition:SURVey:DELeTe****GPS Position Survey Delete****Syntax**

```
GPS:POSition:SURVey:DELeTe
```

**Description**

Delete the surveyed position stored in nonvolatile memory, if any.

**Example**

```
GPS:POS:SURV:DEL
```

Delete the current stored position in nonvolatile memory.

---

**GPS:POSition:SURVey:PROGress?****GPS Position Survey Progress****Syntax**

```
GPS:POSition:SURVey:PROGress?
```

**Description**

Query how much of the position survey has completed. The FS740 will return an integer between 0 and 100 %.

**Example**

```
GPS:POS:SURV:PROG?
```

Query the progress of the position survey by the GPS receiver.

---

**GPS:POSition:SURVey:SAVe****GPS Position Survey Save****Syntax**

```
GPS:POSition:SURVey:SAVe
```

**Description**

Save the current position in nonvolatile memory. The receiver will subsequently use it to enter position hold mode where all satellites are used for maximum timing performance.

**Example**

```
GPS:POS:SURV:SAV
```

Save the current position in nonvolatile memory.

**GPS:POSition:SURVey:START****GPS Position Survey Start****Syntax**

```
GPS:POSition:SURVey:START
```

**Description**

Restart the GPS receiver's position survey. Note that previously saved results are not deleted by this command.

**Example**

```
GPS:POS:SURV:STAR
```

Restart the GPS receiver's position survey.

**GPS:POSition:SURVey:STATe?****GPS Position Survey State****Syntax**

```
GPS:POSition:SURVey:STATe?
```

**Description**

Query whether a position survey is in progress or not. The query returns 1 if the survey is in progress, otherwise 0.

**Example**

```
GPS:POS:SURV:STAT
```

Query whether a position survey is in progress.

**GPS:SATellite:TRACking?****GPS Satellite Tracking****Syntax**

```
GPS:SATellite:TRACking?
```

**Description**

Query which GPS satellites are being tracked by the receiver. The query returns the number of satellites being tracked, followed by the IDs of the satellites as a comma ( , ) separated list.

**Example**

```
GPS:SAT:TRAC?
```

Query the number and IDs of the satellites being tracked by the receiver.

**GPS:SATellite:TRACking:STATus?****GPS Satellite Tracking Status**

**Syntax**

```
GPS:SATellite:TRACking:STATus?
```

**Description**

The receiver has 20 channels for tracking satellites. This command returns the information shown in Table 11 for each channel, successively:

**Table 11: Satellite tracking information**

Index	Parameter
0	Satellite ID number
1	Acquired
2	Ephemeris
3	Is old
4	Signal level in dBHz
5	Elevation in degrees
6	Azimuth in degrees
7	Space vehicle type

If a channel is not tracking a satellite, it will return zero for all parameters. In all,  $20 \times 8 = 160$  parameters are returned as a comma ( , ) separated list.

**Example**

```
GPS:SAT:TRAC:STAT?
```

Return tracking information on all satellites being followed.

**GPS:UTC:OFFSet?****GPS UTC Offset****Syntax**

```
GPS:UTC:OFFSet?
```

**Description**

Query the current offset between UTC and GPS in seconds. As of January 1, 2017, UTC, which has leap seconds inserted intermittently, is 18 seconds behind GPS which does not have leap seconds inserted. Note that this command will return 0 until the time of day has been set properly.

**Example**

```
GPS:UTC:OFFS?
```

Query the current offset between UTC and GPS in seconds.

## Input Subsystem

Commands in the Input Subsystem can apply to either the front or rear input. The user selects the input by optionally appending a 1 or 2 to the DATA keyword. When the suffix is 1 or omitted, the front input is selected. When the suffix is 2, the rear input is selected.

**INPut:LEVel****Input Trigger Level****Syntax**

```
INPut[{1|2}]:LEVel {<level>|MINimum|MAXimum|DEFault}
INPut[{1|2}]:LEVel?
```

**Description**

The first definition sets the input trigger level to <level> in volts. The second definition queries the current input trigger level. The <level> may vary from -3.0 to +3.0 V. The default trigger level is 0.0 V.

**Example**

```
INP:LEV 1.0
INP2:LEV 2.0
INP:LEV?
```

Set the trigger level from the front input to 1.0 V. Set the trigger level for the rear input to 2.0 V. Then query the current trigger level for the front input.

**INPut:SLOPe****Input Trigger Slope****Syntax**

```
INPut[{1|2}]:SLOPe {NEGative|POSitive|DEFault}
INPut[{1|2}]:LEVe1?
```

**Description**

The first definition sets the input trigger slope to the desired polarity. The second definition queries the current input trigger slope. The default trigger slope is positive.

**Example**

```
INP:SLOP POS
INP2:SLOP NEG
INP:SLOP?
```

Set the front input trigger slope to positive. Set the rear input trigger slope to negative. Then query the current trigger slope for the front input.

## Route Subsystem

Commands in the Route Subsystem enable the users to configure the routing of signals to the Sine/Aux option boards. The rear panel of the FS740 accepts the installation of 1 to 3 option boards for the distribution of 10 MHz, the Sine/Aux outputs, or the Pulse outputs. The Sine/Aux option boards can distribute either the Sine outputs or the Aux outputs. Commands in the Route Subsystem enable the user to select which signal to distribute on these boards.

**ROUTe:OPTion****Route Option****Syntax**

```
ROUTe:OPTion[{1|2|3}] [{SINe|AUX}]
ROUTe:OPTion[{1|2|3}]?
```

**Description**

The first definition sets the desired distribution signal for the given option. The second definition returns the current routing for the given option. These values only have significance if a Sine/Aux option is actually installed in the appropriate slot. This value is not affected by a \*RST and is stored in nonvolatile memory so that it is automatically recalled when the instrument is power cycled.

**Example**

```
ROUT:OPT1 AUX
ROUT:OPT2 SIN
```

Route the Aux signal to be distributed on option 1. Route the Sine signal to be distributed on option 2.

## Sample Subsystem

Commands in the Sample Subsystem can apply to either the front or rear input. The user selects the input by optionally appending a 1 or 2 to the SAMPLE keyword. When the suffix is 1 or omitted, the front input is selected. When the suffix is 2, the rear input is selected.

---

### SAMPLE:COUNT

### Sample Count

#### Syntax

```
SAMPLE[{1|2}]:COUNT {<count>|MINimum|MAXimum|DEFAULT}
SAMPLE[{1|2}]:COUNT?
```

#### Description

The first definition changes the number of samples for a configured measurement to <count>. The second definition queries the current sample count. The <count> may vary from 1 to 1,000,000,000. The default count is 1.

#### Example

```
CONF:FREQ
SAMP:COUN 1000
READ?
```

Configure front input for a frequency measurement. Change the sample count to 1000. Trigger a measurement and read the results. The following command queries the current sample count for the rear input.

```
SAMP2:COUN?
```

## Sense Subsystem

The Sense Subsystem enables the user to refine the selected measurement configuration. Commands in the Sense Subsystem can apply to either the front or rear input. The user selects the input by optionally appending a 1 or 2 to the SENSE keyword. When the suffix is 1 or omitted, the front input is selected. When the suffix is 2, the rear input is selected.

---

### SENSE:FREQUENCY:GATE

### Frequency Gate Interval

#### Syntax

```
SENSE[{1|2}]:FREQUENCY:GATE[:TIME] {<duration>|MINimum|MAXimum|DEFAULT}
SENSE[{1|2}]:FREQUENCY:GATE[:TIME]? [{MINimum|MAXimum|DEFAULT}]
```

#### Description

The first definition sets gate interval for frequency measurements to <duration> in seconds. The gate interval may range from 10 ms to 1000 s. The default is 0.1 s. The user may alternatively use the keywords to set the duration to the desired limit. The second definition queries the current gate interval when the optional keywords are omitted. Otherwise, the specified limit is returned.

#### Example

```
CONF:FREQ
```

```

SENS:FREQ:GATE 1.0
CONF2:FREQ
SENS2:FREQ:GATE MIN
SENS2:FREQ:GATE?
READ?
READ2?

```

Configure the front input for frequency measurements. Set the gate interval to 1.0 for the front input. Configure the rear input for frequency measurements. Set the gate interval to the minimum allowed value for the rear input. Then query the current gate interval for the rear input. Trigger a frequency measurement and read the result for the front input. Trigger a frequency measurement and read the result for the rear input.

---

## SENSe:FREQuency:TIMEout

## Frequency Timeout

### Syntax

```

SENSe[{1|2}]:FREQuency:TIMEout <timeout> {MINimum|MAXimum|DEFault|INFinity}
SENSe[{1|2}]:FREQuency:TIMEout? [{MINimum|MAXimum|DEFault}]

```

### Description

The first definition sets time out period for frequency measurements to <timeout> in seconds. The timeout may range from 10 ms to 2000 s. The default is 1.0 s. The user may alternatively use the keywords to set the duration to the desired limit. The second definition queries the current frequency time out period when the optional keywords are omitted. Otherwise, the specified limit is returned.

Note that measurement time outs are tied to the arrival of triggers on the input. A measurement will time out if the arrival between two successive edges on the input exceeds the time out interval. With this scheme, one need not worry about configuring time outs for most signals. For the default time out of 1 s, a time out will not occur if the signal frequency is greater than 1 Hz, regardless of the configured gate interval. Total measurement time, however, will closely follow the gate interval in most cases.

When a time out does occur during a frequency measurement, the FS740 returns NAN for that sample, and moves onto the next sample in the set of measurements. NAN is defined by SCPI to be 9.91e37.

### Example

```

CONF:FREQ
SENS:FREQ:GATE 20.0
SENS:FREQ:TIM 10.0
READ?

```

Configure the front input for frequency measurements. Set the gate interval to 20 s for the front input. Set the measurement time out to 10 s. Trigger a frequency measurement and read the result for the rear input. Changing the time out to 10 s is appropriate for signals whose frequency is between 0.1 and 1.0 Hz. **SENSe:TIME:BMODE**      **Time Buffer Mode**

### Syntax

```

SENSe[{1|2}]:TIME:BMODE {KFIRst|KLASt}
SENSe[{1|2}]:TIME:BMODE?

```

### Description

The first definition sets buffer mode for time measurements. The second definition queries the current buffer mode for time measurements. The internal buffer has space to hold 250,000 time measurements. This command controls the behavior of the FS740 when more than 250,000 measurements are made. If KFIRst (Keep First) is specified then older measurements are preserved

and new measurements are dropped. If KLASt (Keep Last) is specified then older measurements are dropped and newer measurements are preserved. The default value is KLASt.

**Example**

```
CONF:TIM
SENS:TIM:BMOD KFIR
SAMP:COUN 1000000
INIT
```

Configure the front input for time measurements. Set the buffer mode to keep first. Set the sample size to 1 million. Start the measurement. After 250,000 time tags are recorded, newer time tags will be dropped unless space is created by removing measurements with the DATA:REMOve command.

## Source Subsystem

The Source Subsystem enables users to configure the frequency and amplitude of the FS740's various source outputs. Commands can apply to one of three possible outputs: the sine output, the aux output, or the pulse output. The user selects the output by optionally appending a 1, 2 or 3 to the SOURCe keyword. When the suffix is 1 or omitted, the sine output is selected. When the suffix is 2, the aux output is selected. When the suffix is 3 the pulse output is selected. Note that some commands may not apply to all outputs. In those cases the allowed suffixes are restricted to those outputs for which the command is supported.

### SOURce:FREQuency

### Source Frequency

**Syntax**

```
SOURce[ {1|2|3} ]:FREQuency[ { :CW|:FIXed} ] {<freq>|MINimum|MAXimum|DEFault}
SOURce[ {1|2|3} ]:FREQuency[ { :CW|:FIXed} ]? [ {MINimum|MAXimum} ]
```

**Description**

The first definition sets the output frequency for the selected output to <freq> in Hz. The second definition queries the current frequency for the selected output, or the given limit when the optional parameter is included. The range of allowed frequencies for each output is shown in Table 12.

**Table 12: Allowed frequency ranges for each output**

Output	Frequency Range
Sine	1 mHz to 30.1 MHz
Aux	1 mHz to 1 MHz
Aux (sinusoid)	1 mHz to 10 MHz
Pulse	1 mHz to 25 MHz

**Example**

```
SOUR:FREQ 10 MHz
SOUR2:FREQ 50 kHz
SOUR3:FREQ 2.5 MHz
```

Set the frequency of the sine output to 10 MHz. Set the frequency of the aux output to 50 kHz. Set the frequency of the pulse output to 2.5 MHz.

### SOURce2:FUNction

### Aux Output Function

**Syntax**

```
SOURce2:FUNction[ :SHApe] {SINusoid|TRIangle|SQUare|HMHZ|IRIG}
SOURce2:FUNction[ :SHApe]?
```

**Description**

The first definition sets the function for the Aux output. The second definition queries the current function for the Aux output. The allowed keywords correspond to the waveforms detailed in Table 13

**Table 13: Allowed waveforms for the Aux output**

Keyword	Waveform
SINusoid	Sinusoidal waveform
TRIangle	Triangle waveform
SQUare	Square waveform
HMHZ	100 MHz sine wave
IRIG	AM modulated IRIG-B

**Example**

```
SOUR2:FUNC SIN
SOUR2:FREQ 5 MHz
```

Select a sinusoidal waveform for the Aux output. Set its frequency to 5 MHz.

**SOURce3:FUNCTION**

**Pulse Output Function**

**Syntax**

```
SOURce3:FUNCTION[:SHAPE] {PULSe|IRIG}
SOURce3:FUNCTION[:SHAPE]?
```

**Description**

The first definition sets the function for the Pulse output. The second definition queries the current function for the Pulse output. The allowed keywords correspond to the waveforms detailed in Table 14.

**Table 14: Allowed waveforms for the Pulse output**

Keyword	Waveform
PULSe	Pulse waveform
IRIG	Pulse modulated IRIG-B

When pulse waveform is selected, the user can specify its pulse width and period or its frequency and duty cycle.

**Example**

```
SOUR3:FUNC PULS
SOUR3:FREQ 2 MHz
SOUR3:PULS:DCYC 50%
```

Select the pulse waveform for the Pulse output. Set its frequency to 2 MHz. Set its duty cycle to 50%.

**SOURce:PHASe**

**Source Phase Adjust**

**Syntax**

```
SOURce[{1|2|3}]:PHASe[:ADJust] {<phase>|MINimum|MAXimum|DEFault}
SOURce[{1|2|3}]:PHASe[:ADJust]? [{MINimum|MAXimum}]
```

**Description**

The first definition adjusts the phase of the selected output to <phase> or one of the keyword values. The second definition queries the current phase for the selected output. <Phase> may be specified in degrees, radians, or seconds. See section Units on page 74 for more information on specifying units. Command queries are returned in units specified by the SOUR:PHAS:UNITS command. These units are also assumed for <phase> if units are omitted in the set command. At \*RST the default units are degrees. Positive values of <phase> cause the phase to lead the reference. Negative values of <phase> cause the phase to lag the reference. The phase may be adjusted by up to 360 degrees at a time. After the phase is adjusted, the new value is normalized modulo 360 degrees. If the current phase is 350 degrees, for example, the user may legally set the phase to 370 degrees. This will advance the phase by 20 degrees. Afterward the phase will be renormalized and return 10 degrees when queried.

**Example**

```
SOUR:PHAS:SYNC
SOUR:PHAS 45 Deg
SOUR2:PHAS:SYNC
SOUR2:PHAS -90 Deg
SOUR3:PHAS:SYNC
SOUR3:PHAS:UNITS sec
SOUR3:PHAS 10e-9
```

Synchronize the phase for the Sine output to UTC. Set the phase of the Sine output to 45 Deg. Synchronize the phase for the Aux output to UTC. Set the phase of the Aux output to -90 Deg. Synchronize the phase for the Pulse output to UTC Set the default units of phase for the Pulse output to seconds. Set the phase of the pulse output to 10 ns.

**SOURce:PHASe:REfERENCE****Source Phase Reference****Syntax**

```
SOURce[{1|2|3}]:PHASe:REfERENCE
```

**Description**

Make the current phase for the selected output the reference for future phase adjustments. The current phase will be zero after this command but the physical output is not changed in any way.

**Example**

```
SOUR:PHAS:SYNC
SOUR:PHAS 45 Deg
SOUR:PHAS:REF
SOUR:PHAS?
```

Synchronize the phase for the Sine output to UTC. Set the phase of the Sine output to 45 degrees. Make it so that 45 degrees from UTC is now 0 degrees. The query will return 0.

**SOURce:PHASe:SYNChronize****Source Phase Synchronize****Syntax**

```
SOURce[{1|2|3}]:PHASe:SYNChronize
```

**Description**

Adjust the phase for the selected output to align with UTC, if possible. If UTC is not yet known, then phase is aligned to a common internal reference. For the Sine and Aux outputs, the zero crossing of

the rising edge of the waveform is aligned. Outputs are disabled and re-enabled on the next second boundary of UTC. Thus, this command may take up to 1 second to complete.

### Example

```
SOUR:PHAS:SYNC
```

Synchronize the phase for the Sine output to UTC.

---

## SOURce:PHASe:SYNChronize:AUTO

## Source Phase Synchronize Auto

### Syntax

```
SOURce[{1|2|3}]:PHASe:SYNChronize:AUTO {ON|1|OFF|0}
SOURce[{1|2|3}]:PHASe:SYNChronize:AUTO?
```

### Description

When auto synchronization is on, the phase for the given output will be re-aligned to UTC any time its frequency is changed. This will ensure the phase will stay in alignment with UTC, but it also means the output will be disabled for up to 1 second each time a frequency change is made. When auto synchronization is off, no such re-alignment occurs and frequency changes are seamless. The first definition above allows the user to turn auto synchronization on or off. The second definition queries the current state of auto synchronization. The factory default is off. This is a system setting which is unaffected by a \*RST command or a recall of default settings. The setting is stored in nonvolatile memory and automatically restored at power on.

### Example

```
SOUR:PHAS:SYNC
```

Synchronize the phase for the Sine output to UTC.

---

## SOURce:PHASe:SYNChronize:TDElay

## Source Phase Synchronize Time Delay

### Syntax

```
SOURce[{1|2|3}]:PHASe:SYNChronize:TDElay {<delay>|MINimum|MAXimum|DEFault}
SOURce[{1|2|3}]:PHASe:SYNChronize:TDElay?
```

### Description

This command enables the user to control alignment of the signal of an output relative to UTC. The first definition sets the time delay to <delay> or the given limit. The second definition queries the current value of the time delay. The <delay> can range from -1.0 to +1.0 seconds. Negative delays advance the phase of the signal. Positive delays retard the phase of the signal. This command is useful for correcting insertion delays of cables used to get signals from the FS740 to application equipment. The factory default is 0.0 seconds. This is a system setting which is unaffected by a \*RST command or a recall of default settings. The setting is stored in nonvolatile memory and automatically restored at power on.

### Example

```
SOUR:PHAS:SYNC:TDEL -100 ns
```

Advance the phase of the Sine output by 100 ns. This will correct for several feet of cable delay in getting the signal to application equipment.

---

## SOURce3:PULSe:DCYCLE

## Source Pulse Duty Cycle

**Syntax**

```
SOURce3:PULSe:DCYClE {<duty cycle>|MINimum|MAXimum|DEFault}
SOURce3:PULSe:DCYClE? [{MINimum|MAXimum}]
```

**Description**

The first definition sets the duty cycle of the Pulse output to <duty cycle> or the selected limit. The second definition queries the current duty cycle of the Pulse output. Duty cycle is specified in percent, and can range from 0 to 100 %. The actual limits vary with frequency because the pulse width must obey the following constraint:  $5 \text{ ns} \leq \text{pulse width} \leq \text{pulse period} - 5 \text{ ns}$ . Once the duty cycle is set, the instrument will try to maintain that duty cycle for future changes in frequency or period of the Pulse output.

**Example**

```
SOUR3:FREQ 1 kHz
SOUR3:PULSe:DCYC 25%
```

Set the frequency of the pulse output to 1 kHz. Set the duty cycle of the waveform to 25 %.

**SOURce3:PULSe:PERiod****Source Pulse Period****Syntax**

```
SOURce3:PULSe:PERiod {<period>|MINimum|MAXimum|DEFault}
SOURce3:PULSe:PERiod? [{MINimum|MAXimum}]
```

**Description**

The first definition sets the period of the Pulse output to <period> in seconds or the selected limit. The second definition queries the current period of the Pulse output. The <period> can range from 40 ns to 1000 s. The default period is 1 s. Note that the user can alternatively set the frequency of the Pulse output with the SOUR3:FREQ command. When the pulse period is modified, the pulse width may also change. If a duty cycle was specified, then the pulse width will be modified to maintain the specified duty cycle. Alternatively, if a pulse width was explicitly set, then its setting will be preserved, unless doing so would violate the constraint that it be less the period – 5 ns, in which case it will be clipped to that limit.

**Example**

```
SOUR3:PULSe:PER 1 us
SOUR3:PULSe:WIDT 10 ns
```

Set the pulse period to 1  $\mu\text{s}$  and the pulse width to 10 ns.

**SOURce3:PULSe:VIEW****Source Pulse Period****Syntax**

```
SOURce3:PULSe:VIEW {PERiod|FREQuency}
SOURce3:PULSe:VIEW?
```

**Description**

The first definition sets the front panel view for the Pulse output to Period/Width or Freq/Duty. The second definition queries the current view.

**Example**

```
SOUR3:PULSe:VIEW FREQ
```

Set the front panel view for pulse output to Freq/Duty.

**SOURce3:PULSe:WIDTh****Source Pulse Width****Syntax**

```
SOURce3:PULSe:WIDTh {<width>|MINimum|MAXimum|DEFault}
SOURce3:PULSe:WIDTh? [{MINimum|MAXimum}]
```

**Description**

The first definition sets the pulse width of the Pulse output to <width> in seconds or the selected limit. The second definition queries the current pulse width of the Pulse output. The <width> can range from 5 ns to the pulse period – 5 ns. The default pulse width is that which gives a 50 % duty cycle. Once the pulse width is set with this command, the instrument will try to maintain that pulse width for future changes in frequency or period of the Pulse output, if possible.

**Example**

```
SOUR3:PULSe:PER 1 us
SOUR3:PULSe:WIDTh 10 ns
```

Set the pulse period to 1  $\mu$ s and the pulse width to 10 ns.

**SOURce:VOLTage****Source Voltage****Syntax**

```
SOURce[{1|2}]:VOLTage[:LEVel] {<voltage>|MINimum|MAXimum|DEFault}
SOURce[{1|2}]:VOLTage[:LEVel]? [{MINimum|MAXimum}]
```

**Description**

The first definition sets the AC voltage level for the output to <voltage> or the selected limit. The second definition queries the current AC voltage level for the output. The <voltage> may be specified in  $V_{PP}$ ,  $V_{RMS}$ , or dBm. Note that only  $V_{PP}$  is accepted for the Aux output when non sinusoidal waveforms are selected. See section Units on page 74 for more details on specifying units. Queries are returned in units specified by the SOUR:VOLT:UNIT command. These units are also assumed for <voltage> if units are omitted in the set command. At \*RST the default units are  $V_{PP}$ .

**Example**

```
SOUR:VOLT 0.25 Vrms
SOUR2:VOLT 1.0 Vpp
```

Set the amplitude of the Sine output to 0.25  $V_{RMS}$ . Set the amplitude of the Aux output to 1.0  $V_{PP}$ .

**SOURce:VOLTage:UNITs****Source Voltage Units****Syntax**

```
SOURce[{1|2}]:VOLTage:UNITs {VPP|VRMS|DBM}
SOURce[{1|2}]:VOLTage:UNITs?
```

**Description**

The first definition selects the default units when specifying or querying AC voltage levels with the command SOUR:VOLT. At \*RST the default units will be  $V_{PP}$ .

**Example**

```
SOUR:VOLT 0.25 Vrms
SOUR2:VOLT 1.0 Vpp
```

Set the amplitude of the Sine output to 0.25  $V_{RMS}$ . Set the amplitude of the Aux output to 1.0  $V_{PP}$ .

## Status Subsystem

Commands in the Status Subsystem report on instrument status. Each element of status has 3 registers associated with it: a condition register, an event register, and an enable register. For a detailed discussion of these registers and how they are related see Status Reporting on page 76.

---

### STATus:GPS:CONDition?

### Status GPS Condition

#### Syntax

```
STATus:GPS:CONDition?
```

#### Description

Query the current condition of the GPS receiver. See section GPS Receiver Status on page 84 for detailed information on the interpretation of GPS receiver status. See Status Reporting on page 76 for more information on condition registers.

#### Example

```
STAT:GPS:COND?
```

Query the current condition of the GPS receiver status register.

---

### STATus:GPS:ENABLE

### Status GPS Enable

#### Syntax

```
STATus:GPS:ENABLE <mask>
STATus:GPS:ENABLE?
```

#### Description

The first definition sets the mask for combining GPS receiver status bits into the summary bit located in the serial poll status byte. The second definition queries the current mask. See section GPS Receiver Status on page 84 for detailed information on the interpretation of GPS receiver status. See section Status Reporting on page 76 for more information about enable registers.

#### Example

```
STAT:GPS:ENAB 1
```

Set the summary bit in the serial poll status byte if the time of day of the instrument has not been set by the GPS receiver.

---

### STATus:GPS:EVENT?

### Status GPS Event

#### Syntax

```
STATus:GPS[:EVENT]?
```

#### Description

Query the GPS receiver status event register. See section GPS Receiver Status on page 84 for detailed information on the interpretation of GPS receiver status. See Status Reporting on page 76 for more information on event registers.

#### Example

```
STAT:GPS?
```

Query the event register for GPS receiver status. This returns all bits that have been set since the previous query. The query then clears all bits.

**STATus:OPERation:CONDition?****Status Operation Condition****Syntax**

```
STATus:OPERation:CONDition?
```

**Description**

Query the current condition of operational status for the FS740. See section Operation Status on page 83 for detailed information on the interpretation of the operation status bits. See Status Reporting on page 76 for more information on condition registers.

**Example**

```
STAT:OPER:COND?
```

Query the current condition of operational status for the FS740.

**STATus:OPERation:ENABLE****Status Operation Enable****Syntax**

```
STATus:OPERation:ENABLE <mask>
STATus:OPERation:ENABLE?
```

**Description**

The first definition sets the mask for combining operational status bits into the summary bit located in the serial poll status byte. The second definition returns the current mask. See section Operation Status on page 83 for detailed information on the interpretation of operational status bits. See section Status Reporting on page 76 for more information about enable registers.

**Example**

```
STAT:OPER:ENAB 64
```

Set the summary bit in the serial poll status byte if an event was detected on the front input since the previous query of the event register.

**STATus:OPERation:EVENT?****Status Operation Event****Syntax**

```
STATus:OPERation[:EVENT]?
```

**Description**

Query the event register of operational status for the FS740. See section Operation Status on page 83 for detailed information on the interpretation of the operation status bits. See Status Reporting on page 76 for more information on event registers.

**Example**

```
STAT:OPER?
```

Query the event register for operational status. This returns all bits that have been set since the previous query. The query then clears all bits.

**STATus:QUESTionable:CONDition?****Status Questionable Condition**

**Syntax**

```
STATus:QUEStionable:CONDition?
```

**Description**

Query the current condition of questionable status for the FS740. See section Questionable Status on page 82 for detailed information on the interpretation of the questionable status bits. See Status Reporting on page 76 for more information on condition registers.

**Example**

```
STAT:QUES:COND?
```

Query the current condition of questionable status for the FS740.

**STATus:QUEStionable:ENABLE****Status Questionable Enable****Syntax**

```
STATus:QUEStionable:ENABLE <mask>
STATus:QUEStionable:ENABLE?
```

**Description**

The first definition sets the mask for combining questionable status bits into the summary bit located in the serial poll status byte. The second definition returns the current mask. See section Questionable Status on page 82 for detailed information on the interpretation of questionable status bits. See section Status Reporting on page 76 for more information about enable registers.

**Example**

```
STAT:QUES:ENAB 32
```

Set the summary bit in the serial poll status byte if the timebase was not at optimum frequency stability since the previous query of the event register.

**STATus:QUEStionable:EVENT?****Status Questionable Event****Syntax**

```
STATus:QUEStionable[:EVENT]?
```

**Description**

Query the event register of questionable status for the FS740. See section Questionable Status on page 82 for detailed information on the interpretation of the questionable status bits. See Status Reporting on page 76 for more information on event registers.

**Example**

```
STAT:QUES?
```

Query the event register for questionable status. This returns all bits that have been set since the previous query. The query then clears all bits.

## System Subsystem

Commands in the System Subsystem control overall system behavior. System settings are typically not affected by a \*RST and are stored in nonvolatile memory so that they may be recalled automatically when the power is cycled.

---

**SYSTem:ALARm?****System Alarm****Syntax**

```
SYSTem:ALARm?
```

**Description**

Query the current state of the system alarm. The FS740 will return 1 if the alarm is asserted, otherwise 0.

**Example**

```
SYST:ALAR?
```

Query the current state of the system alarm.

---

**SYSTem:ALARm:CLEAr****System Alarm Clear****Syntax**

```
SYSTem:ALARm:CLEAr
```

**Description**

Clear the event register for the system alarm. When the current mode for command SYST:ALARm:MODE is LATCH, this will clear the alarm assuming current limits are not being exceeded.

**Example**

```
SYST:ALAR:CLE
```

Clear the event register for the system alarm.

---

**SYSTem:ALARm:CONDition?****System Alarm Condition****Syntax**

```
SYSTem:ALARm:CONDition?
```

**Description**

Query the condition register for the system alarm. This register indicates which of the possible alarm conditions are currently true. The system alarm will only be asserted if a condition is true and it is enabled in the enable register.

**Example**

```
SYST:ALAR:COND?
```

Query the condition register for the system alarm.

---

**SYSTem:ALARm:ENABle****System Alarm Enable****Syntax**

```
SYSTem:ALARm:ENABle <mask>  
SYSTem:ALARm:ENABle?
```

**Description**

Mask possible alarm conditions so that only those that are enabled here can cause the system alarm to be asserted. When the current mode for command SYST:ALARm:MODE is TRACK, this register

masks the condition register for the system alarm, SYST:ALARm:CONDition. When the current mode for command SYST:ALARm:MODE is LATCh, this register masks the event register for the system alarm, SYST:ALARm:EVENT.

### Example

```
SYST:ALAR:ENAB 1
```

Enable alarm if the time of day has not been set by GPS.

---

## SYSTem:ALARm:EVENT?

## System Alarm Event

### Syntax

```
SYSTem:ALARm:EVENT?
```

### Description

Query the event register for the system alarm. This register indicates which of the possible alarm conditions that have been latched since the last time the event register was cleared. When the current mode for command SYST:ALARm:MODE is LATCh the system alarm will be asserted if an event condition is true AND it is enabled in the enable register. Note that unlike the event registers in the Status Subsystem, reading this register does not clear it. It must be explicitly cleared with the SYSTem:ALARm:CLEAr command.

### Example

```
SYST:ALAR:EVENT?
```

Query the condition register for the system alarm.

---

## SYSTem:ALARm:FORCe:STATe

## System Alarm Force State

### Syntax

```
SYSTem:ALARm:FORCe[:STATe] {1|ON|0|OFF}
SYSTem:ALARm:FORCe[:STATe]?
```

### Description

The first definition sets the forced state of the alarm. The second definition queries the current forced state of the alarm. This value only has significance if the current mode for command SYST:ALARm:MODE is FORCe.

### Example

```
SYST:ALAR:FORC ON
```

Assert the alarm if the alarm mode is FORCe.

---

## SYSTem:ALARm:MODE

## System Alarm Mode

### Syntax

```
SYSTem:ALARm:MODE {TRACK|LATCh|FORCe}
SYSTem:ALARm:MODE?
```

### Description

The first definition sets the alarm mode to one of three options: track, latch, or force. The second definition queries the current alarm mode.

Tracking mode causes the alarm to follow current conditions. The alarm is asserted when current limits are exceeded. The alarm is de-asserted when current limits are no longer exceeded.

Latching mode causes the alarm to be asserted when current limits are exceeded. However, the alarm will not be de-asserted until explicitly requested to do so and the limit is no longer exceeded.

In force mode, the user manually sets the state of the alarm.

### Example

```
SYST:ALAR:MODE
```

Query the current mode for the system alarm.

---

## SYSTem:ALARm:GPS:TINTerval

## System Alarm GPS Time Interval

### Syntax

```
SYSTem:ALARm[:GPS]:TINTerval {<time error>|MINimum|MAXimum|DEFault}
SYSTem:ALARm[:GPS]:TINTerval?
```

### Description

The first definition sets the time interval between GPS and the internal timebase that must be exceeded before the alarm condition for a timing error is asserted. The <time error> may range from 50 ns to 1 s. The default is 100 ns. The second definition queries the current value for the time interval.

### Example

```
SYST:ALAR:TINT 1 us
```

Set the timing error limit to 1  $\mu$ s. This means that the alarm condition for timing errors will not be set unless the measured time interval between GPS and the internal timebase exceeds 1  $\mu$ s. Note that the system alarm will not actually be asserted unless this condition is also enabled with the SYST:ALARm:ENABLE command.

---

## SYSTem:ALARm:HOLDover:Duration

## System Alarm Holdover Duration

### Syntax

```
SYSTem:ALARm[:HOLDover]:DURation {<duration>|MINimum|MAXimum|DEFault}
SYSTem:ALARm[:HOLDover]:DURation?
```

### Description

The first definition sets the amount of time in seconds that the FS740 must be in holdover before the alarm condition for holdover is asserted. The <duration> may be any 32 bit unsigned integer. The default is 0. The second definition queries the current value for the duration.

### Example

```
SYST:ALAR:DUR 100
```

Set the holdover duration to 100 seconds. This means that the alarm condition for holdover will not be set unless the FS740 is in holdover for more than 100 seconds. Note that the alarm will not actually be asserted unless this condition is also enabled with the SYST:ALARm:ENABLE command.

---

## SYSTem:COMMunicate:LAN:EPHY?

## System Communicate LAN EPHY

**Syntax**

```
SYSTem:COMMunicate:LAN[:EPHY]?
```

**Description**

Query whether the FS740 is connected to the Ethernet LAN. The query returns 1 if connected, otherwise 0.

**Example**

```
SYST:COMM:LAN?
```

Query whether the FS740 is connected to the Ethernet LAN.

**SYSTem:COMMunicate:LAN:EPHY:SPEEd****System Communicate LAN EPHY Speed****Syntax**

```
SYSTem:COMMunicate:LAN[:EPHY]:SPEEd [{10BASET|100BASET}]
SYSTem:COMMunicate:LAN[:EPHY]:SPEEd?
```

**Description**

The first definition configures the speed of the Ethernet network to which the FS740 is connected, 10BaseT or 100BaseT. If omitted the command defaults to 100BaseT. The second definition queries the current configured speed. Note that changes to this configuration do not take effect until the LAN is reset via a SYSTem:COMMunicate:LAN:RESet command or the power is cycled.

**Example**

```
SYST:COMM:LAN:SPE 10BaseT
SYST:COMM:LAN:RESET
```

Configure the FS740 to connect to a 10BaseT Ethernet network. Then reset the LAN so that the new configuration takes effect.

**SYSTem:COMMunicate:LAN:DHCP****System Communicate LAN DHCP****Syntax**

```
SYSTem:COMMunicate:LAN:DHCP [{1|ON|0|OFF}]
SYSTem:COMMunicate:LAN:DHCP?
```

**Description**

The first definition enables or disables DHCP for configuring the TCP/IP address of the FS740. Note that the new configuration does not take effect until the LAN is reset via a SYSTem:COMMunicate:LAN:RESet command or the power is cycled.

**Example**

```
SYST:COMM:LAN:DHCP ON
SYST:COMM:LAN:RESET
```

Configure the FS740 to use DHCP to configure the TCP/IP address of the instrument. Then reset the LAN so that the new method of configuration becomes active.

**SYSTem:COMMunicate:LAN:GATeway****System Communicate LAN Gateway****Syntax**

```
SYSTem:COMMunicate:LAN:GATeway <ip address>
SYSTem:COMMunicate:LAN:GATeway? [{CURRent|STATic}]
```

**Description**

The first definition sets the IP address of the gateway or router on the users TCP/IP network to be used in static configurations. The second definition queries either the current gateway address or the gateway address to be used in static configurations. If the parameter is omitted, the current gateway address is returned. The <ip address> should be specified as a string in the form “xxx.xxx.xxx.xxx” where each xxx is replaced by an integer in the range from 0 to 255. Each integer should be separated by a decimal point ( . ) with no spaces.

**Example**

```
SYST:COMM:LAN:DHCP OFF
SYST:COMM:LAN:IPAD "192.168.0.10"
SYST:COMM:LAN:SMAS "255.255.255.0"
SYST:COMM:LAN:GAT "192.168.0.1"
SYST:COMM:LAN:RESET
```

Disable DHCP. Set the IP address to 192.168.0.10. Set the subnet mask to 255.255.255.0. Set the gateway to 192.168.0.1. Then reset the LAN so that the new configuration will take effect.

**SYSTem:COMMunicate:LAN:IPADress****System Communicate LAN IP Address****Syntax**

```
SYSTem:COMMunicate:LAN:IPADress <ip address>
SYSTem:COMMunicate:LAN:IPADress? [{CURRent|STATic}]
```

**Description**

The first definition sets the IP address of the FS740 on the users TCP/IP network to be used in static configurations. The second definition queries either the current IP address or the IP address to be used in static configurations. If the parameter is omitted, the current IP address is returned. These may differ, for example, if DHCP is enabled. The <ip address> should be specified as a string in the form “xxx.xxx.xxx.xxx” where each xxx is replaced by an integer in the range from 0 to 255. Each integer should be separated by a decimal point ( . ) with no spaces.

**Example**

```
SYST:COMM:LAN:DHCP OFF
SYST:COMM:LAN:IPAD "192.168.0.10"
SYST:COMM:LAN:SMAS "255.255.255.0"
SYST:COMM:LAN:GAT "192.168.0.1"
SYST:COMM:LAN:RESET
```

Disable DHCP. Set the IP address to 192.168.0.10. Set the subnet mask to 255.255.255.0. Set the gateway to 192.168.0.1. Then reset the LAN so that the new configuration will take effect.

**SYSTem:COMMunicate:LAN:MAC?****System Communicate LAN MAC address****Syntax**

```
SYSTem:COMMunicate:LAN:MAC?
```

**Description**

Query the Ethernet MAC address for the FS740. It returns a string of the form “00:19:b3:0b:00:03” identifying the MAC address for the device.

**Example**

```
SYST:COMM:LAN:MAC?
```

Query the Ethernet MAC address for the FS740.

---

**SYSTem:COMMunicate:LAN:SMASK**
**System Communicate LAN Subnet Mask**
**Syntax**

```
SYSTem:COMMunicate:LAN:SMASK <ip address>
SYSTem:COMMunicate:LAN:SMASK? [{CURRent|STATic}]
```

**Description**

The first definition sets the subnet mask for the users TCP/IP network to be used in static configurations. The second definition queries either the current subnet mask or the subnet mask to be used in static configurations. If the parameter is omitted, the current subnet mask is returned. These may differ, for example, if DHCP is enabled. The <ip address> should be specified as a string in the form "xxx.xxx.xxx.xxx" where each xxx is replaced by an integer in the range from 0 to 255. Each integer should be separated by a decimal point ( . ) with no spaces.

**Example**

```
SYST:COMM:LAN:DHCP OFF
SYST:COMM:LAN:IPAD "192.168.0.10"
SYST:COMM:LAN:SMAS "255.255.255.0"
SYST:COMM:LAN:GAT "192.168.0.1"
SYST:COMM:LAN:RESET
```

Disable DHCP. Set the IP address to 192.168.0.10. Set the subnet mask to 255.255.255.0. Set the gateway to 192.168.0.1. Then reset the LAN so that the new configuration will take effect.

---

**SYSTem:COMMunicate:SERial:BAUD**
**System Communicate Serial Baudrate**
**Syntax**

```
SYSTem:COMMunicate:SERial:BAUD {4800|9600|19200|38400|57600|115200}
SYSTem:COMMunicate:SERial:BAUD?
```

**Description**

The first definition configures the RS-232 port to operate at the selected baud rate. The second definition queries the current baud rate. Note that the new configuration does not take effect until the port is reset via a SYSTem:COMMunicate:SERial:RESet command or the power is cycled.

**Example**

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

Set the serial port baud rate to 115200 baud. Then reset the port so that the new baud rate becomes active.

---

**SYSTem:COMMunicate:SERial:RESet**
**System Communicate Serial Reset**
**Syntax**

```
SYSTem:COMMunicate:SERial:RESet
```

**Description**

Reset the serial port and activate it using the current configured baud rate

**Example**

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

---

```
SYST:COMM:SER:RESET
```

Set the serial port baud rate to 115200 baud. Then reset the port so that the new baud rate becomes active.

---

## SYSTem:COMMunicate:LOCK?

## System Communicate Lock

### Syntax

```
SYSTem:COMMunicate:LOCK?
```

### Description

Request an exclusive lock on communication with the FS740. The FS740 will return 1 if the request is granted, otherwise 0. When an interface has an exclusive lock on communication with the FS740, other remote interfaces as well as the front panel are prevented from changing the instrument state. The user should call the command SYSTem:COMMunicate:UNLock command to release the exclusive lock when it is no longer needed.

### Example

```
SYST:COMM:LOCK?
SOUR:FREQ 5 MHz
SOUR2:FREQ 1 MHz
SYST:COMM:UNL?
```

Request a communications lock. If all is well, the FS740 will return 1, indicating it has granted the lock. Set the frequency for the Sine output to 5 MHz. Set the frequency for the Aux output to 1 MHz. Finally, release the communications lock.

---

## SYSTem:COMMunicate:UNLock?

## System Communicate Unlock

### Syntax

```
SYSTem:COMMunicate:UNLock?
```

### Description

Release an exclusive lock on communication with the FS740 that was previously granted with the SYSTem:COMMunicate:LOCK command. The FS740 will return 1 if the lock was released, otherwise 0.

### Example

```
SYST:COMM:LOCK?
SOUR:FREQ 5 MHz
SOUR2:FREQ 1 MHz
SYST:COMM:UNL?
```

Request a communications lock. If all is well, the FS740 will return 1, indicating it has granted the lock. Set the frequency for the Sine output to 5 MHz. Set the frequency for the Aux output to 1 MHz. Finally, release the communications lock.

---

## SYSTem:DATE

## System Date

### Syntax

```
SYSTem:DATE <year>,<month>,<day>
SYSTem:DATE?
```

**Description**

The first definition sets the FS740 date if it has not been set by GPS. If the date has already been set by GPS, error –221, “Settings conflict,” will be generated and the requested date ignored. The second definition queries the current date. It returns the year, month, and day as a comma ( , ) separated list.

**Example**

```
SYST:DATE 2016,12,15
```

Set the current date for the FS740 to December 15, 2016, if the date has not already been set by GPS.

**SYSTem:DISPlay:POWer****System Display Power****Syntax**

```
SYSTem:DISPlay:POWer {NOW|TMIN|OHR|THR|ODAY|NEVer}
SYSTem:DISPlay:POWer?
```

**Description**

The first definition sets the period of inactivity, after which the display is powered down. It may be one of the values detailed in Table 15. The second definition queries the current setting.

**Table 15: Allowed display power down settings**

<b>Keyword</b>	<b>If no activity power down display...</b>
NOW	Immediately
TMIN	After 10 minutes
OHR	After 1 hour
THR	After 10 hours
ODAY	After 1 day
NEVer	Never power down display

Note that the keyword NOW is special in that it forces the display to power down immediately, but it does not change the previous setting. Thus, if the previous setting was NEVER, then executing the command SYST:DISP:POW NOW will cause the display to power down immediately, but leave the setting at NEVER.

**Example**

```
SYST:DISP:POW TMIN
```

Set the display to power down after 10 minutes of inactivity.

**SYSTem:DISPlay:SCReen****System Display Screen****Syntax**

```
SYSTem:DISPlay[:SCReen] [{TBASe|GPS|COMMunicate|SYSTem|SINe|AUX|PULSe|MEASure}]
SYSTem:DISPlay[:SCReen]?
```

**Description**

The first definition sets the display. This is equivalent to pressing one of the 8 keys located in the right hand portion of the front panel display. Allowed values are detailed in Table 16. If omitted the display defaults to timebase status and configuration. The second definition queries the current display setting.

Table 16: Allowed display settings

Keyword	Display setting
TBASE	Timebase status and configuration
GPS	GPS receiver status and configuration
COMMunicate	Communication settings
SYSTem	System settings
SINe	Sine output configuration
AUX	Aux output configuration
PULSe	Pulse output configuration
MEASure	Measurement display

**Example**

```
SYST:DISP:SYST
```

Set the display to system settings.

---

**SYSTem:ERRor?**

**System Error**

**Syntax**

```
SYSTem:ERRor[:NEXT]?
```

**Description**

Query the next error at the front of the error queue and then remove it. The error queue can store up to 10 errors. If the error queue overflows, the last error in the queue will be replaced with the error: -350, "Error queue overflow."

**Example**

```
SYST:ERR?
```

Query the next error in the error queue. If no error is in the queue, then 0, "No error," is returned.

---

**SYSTem:SECurity:IMMediate**

**System Security Immediate**

**Syntax**

```
SYSTem:SECurity:IMMediate
```

**Description**

This command wipes the instrument of user settings and restores the unit to factory default settings. All settings stored in nonvolatile memory will be erased except for firmware and calibration data. This includes configuration settings for the GPS receiver and all the remote interfaces. All system configuration settings will be replaced with factory defaults. Execution of this command is not recommended in normal use, as system configuration settings will be lost. However, it may be required when removing the instrument from a secure area. This command may take up to 10 seconds to complete. It should not generally be executed frequently. Use \*RST for general use.

**Example**

```
SYST:SEC:IMM
```

Wipe the instrument of user settings and restore instrument to factory default settings.

---

**SYSTem:TIME**

**System Time**

**Syntax**

```
SYSTem:TIME <hour>,<minute>,<second>
SYSTem:TIME?
```

**Description**

The first definition sets the FS740 time of day if it has not been set by GPS. If the time has already been set by GPS, error -221, "Settings conflict," will be generated and the requested time ignored. The second definition queries the current time of day. It returns the hour, minute, and second as a comma ( , ) separated list of integers. The second field will be returned as a decimal fraction with 10 ns of resolution representing the precise time that the query was executed.

**Example**

```
SYST:TIM 16,23,43
```

Set the current time to 16:23:43, if the time has not already been set by GPS.

**SYSTem:TIME:LOFFset****System Time Local Offset****Syntax**

```
SYSTem:TIME:LOFFset <offset>
SYSTem:TIME:LOFFset?
```

**Description**

The first definition sets the local offset from GPS or UTC to local time in seconds. The second definition queries the current local offset in seconds. This setting is not affected by a \*RST command. Its value is stored in nonvolatile memory and will be automatically restored when the power is cycled.

**Example**

```
SYST:TIM:LOFF -3600
```

Set the local time offset to -1 hour. With this offset in effect, times reported by the FS740 will be 1 hour earlier than UTC.

**SYSTem:TIME:POWeron****System Time Power On****Syntax**

```
SYSTem:TIME:POWeron?
```

**Description**

Query the date and time at which the FS740 was powered on. The FS740 returns the year, month, day, hour, minute, and second as a comma ( , ) separated list. Until the date and time are set by GPS this command will report power on to be midnight, January 6, 1980, which is the start date of GPS. Once time has been set by GPS, the true date and time of power on will be returned.

**Example**

```
SYST:TIM:POW?
```

Query the date and time at which the FS740 was powered on.

## Timebase Subsystem

Commands in the Timebase Subsystem control the operation of the internal timebase, including whether it locks to GPS, when it comes unlocked from GPS, and how it recovers from the unlocked state. Note that commands in this subsystem are not affected by a \*RST. They control system configuration and are automatically saved to nonvolatile memory.

---

### TBAsE:CONFig:BWIDth

### Timebase Configure Bandwidth

#### Syntax

```
TBAsE:CONFig:BWIDth [{AUTO|MANual}]
TBAsE:CONFig:BWIDth?
```

#### Description

The first definition sets the desired bandwidth control. The second definition queries the current value for bandwidth control. When AUTO is selected, the bandwidth with which the timebase follows GPS is automatically adjusted based on the measured timing errors. When the timing error is large bandwidth is increased. Conversely, when timing errors are small bandwidth is decreased. When MANUAL is selected, the bandwidth is fixed and the time constant of the phase lock loop is governed by the value set with the TBAsE:TCONstant command. When the parameter is omitted, the value is assumed to be AUTO.

#### Example

```
TBAS:CONF:BWID AUT
```

Configure the timebase to automatically increase bandwidth when timing errors are large and then gradually narrow the bandwidth when lock is stable.

---

### TBAsE:CONFig:HMODe

### Timebase Configure Holdover Mode

#### Syntax

```
TBAsE:CONFig:HMODe [{WAIT|JUMP|SLEW}]
TBAsE:CONFig:HMODe?
```

#### Description

The first definition controls how the timebase leaves holdover mode when timing offsets are larger than allowed. The second definition queries the current behavior for leaving holdover mode. When WAIT is selected, the timebase will wait for the timing offsets to improve before leaving holdover mode. If JUMP is selected, the timebase will leave holdover by jumping from its current phase to that of GPS to correct the offset immediately. If SLEW is selected the timebase will leave holdover by slewing its phase from its current value to that of GPS to correct the offset.

#### Example

```
TBAS:CONF:HMOD JUMP
```

Configure the timebase to leave holdover by suddenly jumping its phase to that of GPS if the current timing error exceeds the configured limit.

---

### TBAsE:CONFig:LOCK

### Timebase Configure Lock

#### Syntax

```
TBAsE:CONFig:LOCK [{1|ON|0|OFF}]
TBAsE:CONFig:LOCK?
```

**Description**

The first definition controls whether the timebase locks to GPS or not. When set to 1 or ON, the timebase will lock to GPS if it is generating timing pulses. When set to 0 or OFF, the timebase will not lock to GPS. If the parameter is omitted, it is assumed to be ON. The second definition queries the current setting.

**Example**

```
TBAS:CONF:LOCK 1
```

Configure the timebase to lock to GPS when possible.

**TBASe:CONFig:TINTerval:LIMit****Timebase Configure Time Interval Limit****Syntax**

```
TBASe:CONFig[:TINTerval]:LIMit <time error>
TBASe:CONFig[:TINTerval]:LIMit?
```

**Description**

The first definition sets the limit for timing errors to <time error> in seconds. The second definition queries the current limit. The <time error> may range from 50 ns to 1.0 s. The factory default value is 1  $\mu$ s. When the measured timing error of the timebase relative to GPS exceeds this limit, the timebase will unlock from GPS and enter holdover.

**Example**

```
TBAS:CONF:LIM 100 ns
```

Configure the timebase to unlock from GPS if the time interval error exceeds 100 ns.

**TBASe:EVENT:CLEar****Timebase Event Clear****Syntax**

```
TBASe:EVENT:CLEar
```

**Description**

Discard all events in the timebase event queue

**Example**

```
TBAS:EVENT:CLE
```

Discard all events in the timebase event queue.

**TBASe:EVENT:COUNT****Timebase Event Count****Syntax**

```
TBASe:EVENT:COUNT?
```

**Description**

Query the number of events in the timebase event queue.

**Example**

```
TBAS:EVENT:COUN?
```

Query the number of events in the timebase event queue.

**TBAsE:EvEnt:NEXt?**

**Timebase Event Next**

**Syntax**

TBAsE:EvEnt [ :NEXt ] ?

**Description**

Query the queue of timebase events for the next event. The command returns the name of the event, followed by the year, month, day, hour, minute, and second that the event occurred as a comma ( , ) separated list. Then the event is removed from the queue and discarded. If no event has occurred, then NONE is returned with the current time. Allowed events are detailed in Table 17.

**Table 17: Possible timebase events**

Event	Name	Meaning
NONE	None	No events in the queue
POWerup	Power up	Time at which FS740 powered up
UNLock	Unlock	Rb timebase came unlocked
SEARch	Searching for GPS	Searching for GPS satellites
STABILize	Stabilizing	Waiting for timebase frequency to stabilize
VTIME	Validate time	Validating time of day before setting it.
LOCK	Lock	Locked to GPS
MANual	Manual holdover	Entered holdover at user request
NGPS	No GPS	Entered holdover because timing pulses are not being generated by the GPS receiver.
BGPS	Bad GPS	Entered holdover because the timing error of the timebase relative to GPS exceeds the limit specified in the command TBAS:CONF:LIM.

**Example**

TBAS:EVEN?

Query the queue of timebase events for the next event.

**TBAsE:FCONtrol**

**Timebase Frequency Control**

**Syntax**

TBAsE:FCONtrol <fc>  
TBAsE:FCONtrol?

**Description**

The first definition sets the frequency control value for the timebase to <fc>. The second definition returns the current frequency control value. Valid values may range from 0 to 4.096. Error -221, "Settings conflict," is generated if the user tries to manually set the frequency control value when the timebase is locked to GPS. This setting is not automatically saved to nonvolatile memory. It must be explicitly saved with the TBAsE:FCONtrol:SAVe command if desired.

**Example**

TBAS:FCON 2.0

Set the frequency control for the timebase to 2.0.

**TBAsE:FCONtrol:SAVe**

**Timebase Frequency Control Save**

**Syntax**

TBASE:FCOnTrol:SAVe

**Description**

Save the current frequency control value to nonvolatile memory. This value will be restored when the FS740 is power cycled. It will control the frequency of the timebase until it locks to GPS again.

**Example**

TBAS:FCOn:SAV

Save the current frequency control value to nonvolatile memory.

**TBASE:STATe?**

**Timebase State**

**Syntax**

TBASE[:STATe]?

**Description**

Query the current state of the timebase. Allowed states for the timebase are detailed in Table 18.

**Table 18: Allowed states for the timebase**

State	Meaning
POWerup	Instrument has been recently powered up
SEARCh	GPS receiver is searching for satellites
STABilize	Waiting for frequency of the timebase to stabilize
VTIME	Validating time of day reported by GPS
LOCK	Locked to GPS
MANual	Manual: in holdover at the request of the user
NGPS	No GPS: in holdover because timing pulses are not being generated by the GPS receiver.
BGPS	Bad GPS: in holdover because timing error of the timebase relative to GPS exceeds the limit specified in the command TBAS:CONF:LIM.
UNLock	Installed Rb oscillator is unlocked

**Example**

TBAS?

Query the current state of the timebase.

**TBASE:STATe:HOLDoVer:DURation?**

**Timebase Holdover Duration**

**Syntax**

TBASE[:STATe]:HOLDoVer[:DURation]?

**Description**

Query the length of time in seconds the FS740 has been in holdover. If the FS740 is not currently in holdover, it returns 0.

**Example**

TBAS:HOLD?

Query the holdover duration.

---

**TBASE:STATe:LOCK:DURation?**
**Timebase Lock Duration**
**Syntax**

```
TBASE[:STATe]:LOCK[:DURation]?
```

**Description**

Query the length of time the FS740 has been locked to GPS. If the FS740 is not currently locked to GPS, it will return 0.

**Example**

```
TBAS:LOCK?
```

Query the timebase lock duration.

---

**TBASE:STATe:WARMup:DURation?**
**Timebase Warm Up Duration**
**Syntax**

```
TBASE[:STATe]:WARMup[:DURation]?
```

**Description**

Query the time in seconds that passed between when the FS740 was powered on and it first locked to GPS. If it has not successfully locked to GPS, then it returns the time in seconds since power on.

**Example**

```
TBAS:WARM?
```

Query the warm up duration.

---

**TBASE:TCONstant**
**Timebase Time Constant**
**Syntax**

```
TBASE:TCONstant <time constant>
TBASE:TCONstant? [{CURRent|TARGet|MANual}]
```

**Description**

The first definition sets the time constant for the phase lock loop that locks the timebase to GPS when MANual is selected for the command TBASe:CONFIg:BWIDth. The second definition queries one of three different time constants: the current time constant, the target time constant, and the manual time constant set with the first definition above. If the parameter is omitted, the current time constant is returned. If the timebase is configured for automatic bandwidth control (the default), the current time constant may vary from 3 s up to the target time constant for the installed timebase. The target time constant is a factory setting which identifies the optimum time constant for the installed timebase that should be used when the timebase has fully stabilized and timing errors are small.

**Example**

```
TBAS:TCON 40
TBAS:TCON?
TBAS:TCON? MAN
```

Set the manual time constant to 40 s. Query the current time constant. Query the manual time constant set above. It should be 40.

**TBASe:TINterval****Timebase Time Interval****Syntax**

```
TBASe:TINterval? [{CURRent|AVERAge}]
```

**Description**

Query the current or average measured time interval in seconds between the internal timebase and GPS. If the parameter is omitted, the current time interval is returned. The current time interval is the measured time interval between the internal timebase and the latest GPS timing pulse. The average time interval reported is an exponential average of the current time interval with a time constant equal to 1/6 of the time constant of the phase lock loop. See the TBASe:TCONstant command. Positive values indicate the internal timebase is lagging GPS. Negative values indicate the internal timebase is leading GPS. If time of day has not yet been set by the GPS receiver, this command generates error -230, "Data corrupt or stale." Alternatively, if GPS is lost after time has been set, this command returns the last known time interval as the current time interval and zero for the average time interval.

**Example**

```
TBAS:TINT?
```

Query the current time interval between the internal timebase and GPS.

## Trigger Subsystem

Commands in the Trigger Subsystem control the triggering of measurements. Normally when under local control, measurements are triggered continuously in order to provide continuous feedback while interacting with the instrument. Conversely, under remote control, measurements are only triggered when requested so that the data displayed corresponds closely with data retrieved. Commands in this subsystem enable the user to alter this behavior.

**TRIGger:CONTInuous****Trigger Continuous****Syntax**

```
TRIGger:CONTInuous [{1|ON|0|OFF}]
TBASe:CONFig:BWIDth?
```

**Description**

The first definition sets the desired trigger mode. The second definition queries the current trigger mode. When the parameter is omitted,

**Example**

```
TBAS:CONF:BWID AUT
```

Configure the timebase to automatically increase bandwidth when timing errors are large and then gradually narrow the bandwidth when lock is stable.

## Error Codes

The instrument contains an error buffer that may store up to 10 error codes associated with errors encountered during power-on self tests, command parsing, or command execution. The ERR LED will be highlighted when a remote command fails for any reason. The errors in the buffer may be read one by one by executing successive SYST:ERR commands. The user may also view the errors from the front panel by navigating to Communication > Activity. Press CLEAR to discard the error. Errors are displayed in the order in which they occurred. The ERR LED will go off when all errors have been discarded.

The meaning of each of the error codes is described below.

### Command Errors

**-100 Command Error**

A nonspecific command error occurred.

**-104 Data type error**

A parameter has the wrong data type.

**-108 Parameter not allowed**

A parameter was supplied that is not allowed.

**-109 Missing parameter**

A required parameter was missing.

**-113 Undefined header**

The supplied command is undefined.

**-114 Header suffix out of range**

A numeric suffix for one of the command keywords was out of range.

**-120 Numeric data error**

A numeric value was supplied that could not be represented internally. This can happen if a number with an exponent larger than  $\pm 43$  is supplied.

**-131 Invalid suffix**

A supplied parameter contained invalid units.

**-141 Invalid character data**

A supplied character data keyword was invalid or undefined.

**-148 Character data not allowed**

The character data keyword supplied is defined but not allowed.

**-151 Invalid string data**

The supplied string did not contain matched quotes.

**-190 Command buffer overflow**

The supplied command exceeded 256 characters and the command buffer overflowed.

**Execution Errors****-200 Execution error**

A nonspecific execution error occurred.

**-213 Init ignored**

An INITiate command was received but a measurement was already in progress, so the command was ignored.

**-221 Settings conflict**

A valid command was received, but it cannot be executed because current settings of the instrument are incompatible with the command. This may occur, for example, if one tries to manually set the frequency of the timebase when it is locked to GPS.

**-222 Data out of range**

A supplied parameter is out of range.

**-230 Data corrupt or stale**

An attempt to read data failed because it is already been discarded, corrupted, or lost.

**Device Specific Errors****-300 Device specific error**

A device specific error occurred.

**-314 Save/recall memory lost**

A \*SAV or \*RCL command failed because the settings stored in nonvolatile memory were corrupted or lost.

**-350 Error queue overflow**

More have errors occurred, but they were discarded because the error queue overflowed.

**-360 Communications error**

A framing or parity error occurred on the communications interface.

**-363 Input buffer overrun**

The input buffer for the communications interface overflowed. All data was flushed and the communications interface reset.

**-365 Time out error**

The instrument timed out waiting for data over the remote interface.

**Query Errors****-400 Query error**

A nonspecific query error occurred.

**-410 Query INTERRUPTED**

A new command was received before the previous query was read by the remote interface. The old query will be discarded.

**-450 Query lost data**

The output buffer for the communications interface overflowed. The query result was discarded.

**-451 Query no data**

The query resulted in no data.

## Instrument Errors

**500 GPS time out error**

A command was sent to the GPS receiver, but the instrument time out waiting for a response.

**501 GPS failed**

A command was sent to the GPS receiver, but the receiver rejected it as invalid.

**514 Not allowed**

A command was received, but rejected not allowed because another remote interface has an exclusive lock on the instrument.

**800 EEPROM read/write failed**

A read or write to nonvolatile EEPROM failed.

**801 FPGA failed**

The FPGA failed to initialize.

**900 Self test failed**

The instrument self test failed.

## Example Programming Code

The following program can be used as sample code for communicating with the instrument over TCP/IP. The program is written in the C++ language and should compile correctly on a Windows based computer. It could be made to work on other platforms with minor modifications. In order to use the program, you will need to connect the unit to your LAN and configure it with an appropriate IP address. Contact your network administrator for details on how to do this. To identify the unit's current IP address from the front panel, navigate to Communication > TCP/IP Config. The current IP address will be displayed under Current Config.

Copy the program into a file named "fs740\_ctrl.cpp". To avoid typing in the program manually, download the electronic version of this manual from the SRS website ([www.thinksrs.com](http://www.thinksrs.com)). Select the program text and copy/paste it into the text editor of your choice. Compile the program into the executable "fs740\_ctrl.exe". At the command line type something like the following:

```
fs740_ctrl 192.168.0.5
```

where you should replace "192.168.0.5" with the IP address of the unit. You should see something like the following:

```
Connection Succeeded
Stanford Research Systems, FS740, s/n001013, ver2.26.11
Closed connection
```

The program connects to the unit at the supplied IP address sets several parameters and then closes. If successful, the sine output frequency should be set to 1 MHz and the sine output voltage level should be set to 0.5 V<sub>pp</sub>.

```

/* fs740_ctrl.c : Sample program for controlling the FS740 via TCP/IP */
#include "Winsock2.h"
#include <stdio.h>

/* prototypes */
void init_tcpip(void);
int fs740_connect(unsigned long ip);
int fs740_close(void);
int fs740_write(char *str);
int fs740_write_bytes(const void *data, unsigned num);
int fs740_read(char *buffer, unsigned num);

SOCKET sFS740;          /* FS740 tcpip socket */
unsigned fs740_timeout = 6000; /* Read timeout in milliseconds */

int main(int argc, char * argv[])
{
    char buffer[1024];

    /* Make sure ip address is supplied on the command line */
    if ( argc < 2 ) {
        printf("Usage: fs740_ctrl IP_ADDRESS\n");
        exit(1);
    }

    /* Initialize the sockets library */
    init_tcpip();

    /* Connect to the FS740 */
    if ( fs740_connect( inet_addr(argv[1]) ) ) {
        printf("Connection Succeeded\n");

        /* Get identification string */
        fs740_write("*idn?\n");
        if ( fs740_read(buffer, sizeof(buffer)) )
            printf(buffer);
        else
            printf("Timeout\n");
        /* Reset instrument */
        fs740_write("*rst\n");
        /* Set sine output frequency to 1 MHz */
        fs740_write("sour:freq 1e6\n");
        /* Set sine output voltage level to 0.5 Vpp */
        fs740_write("sour:volt 0.5\n");
        /* Make sure all commands have executed before closing connection */
        fs740_write("*opc?\n");
        if ( !fs740_read(buffer, sizeof(buffer)) )
            printf("Timeout\n");
        /* Close the connection */
        if (fs740_close())
            printf("Closed connection\n");
        else
            printf("Unable to close connection");
    }
    else
        printf("Connection Failed\n");

    return 0;
}

```

```

void init_tcpip(void)
{
    WSADATA wsadata;
    if ( WSASStartup(2, &wsadata) != 0 ) {
        printf("Unable to load windows socket library\n");
        exit(1);
    }
}

int fs740_connect(unsigned long ip)
{
    /* Connect to the FS740 */
    struct sockaddr_in intrAddr;
    int status;

    sFS740 = socket(AF_INET,SOCK_STREAM,0);
    if ( sFS740 == INVALID_SOCKET )
        return 0;

    /* Bind to a local port */
    memset(&intrAddr,0,sizeof(intrAddr));
    intrAddr.sin_family = AF_INET;
    intrAddr.sin_port = htons(0);
    intrAddr.sin_addr.S_un.S_addr = htonl(INADDR_ANY);
    if ( SOCKET_ERROR == bind(sFS740,(const struct sockaddr *)&intrAddr,sizeof(intrAddr)) ) {
        closesocket(sFS740);
        sFS740 = INVALID_SOCKET;
        return 0;
    }

    /* Setup address for the connection to fs740 on port 5025 */
    memset(&intrAddr,0,sizeof(intrAddr));
    intrAddr.sin_family = AF_INET;
    intrAddr.sin_port = htons(5025);
    intrAddr.sin_addr.S_un.S_addr = ip;
    status = connect(sFS740,(const struct sockaddr *)&intrAddr,sizeof(intrAddr));
    if ( status ) {
        closesocket(sFS740);
        sFS740 = INVALID_SOCKET;
        return 0;
    }
    return 1;
}

int fs740_close(void)
{
    if ( closesocket(sFS740) != SOCKET_ERROR )
        return 1;
    else
        return 0;
}

int fs740_write(char *str)
{
    /* Write string to connection */
    int result;

    result = send(sFS740,str,(int)strlen(str),0);
    if ( SOCKET_ERROR == result )
        result = 0;
    return result;
}

```

```
int fs740_write_bytes(const void *data, unsigned num)
{
    /* Write string to connection */
    int result;

    result = send(sFS740, (const char *)data, (int)num, 0);
    if ( SOCKET_ERROR == result )
        result = 0;
    return result;
}

int fs740_read(char *buffer, unsigned num)
{
    /* Read up to num bytes from connection */
    int count;
    fd_set setRead, setWrite, setExcept;
    TIMEVAL tm;

    /* Use select() so we can timeout gracefully */
    tm.tv_sec = fs740_timeout/1000;
    tm.tv_usec = (fs740_timeout % 1000) * 1000;

    FD_ZERO(&setRead);
    FD_ZERO(&setWrite);
    FD_ZERO(&setExcept);
    FD_SET(sFS740, &setRead);
    count = select(0, &setRead, &setWrite, &setExcept, &tm);
    if ( count == SOCKET_ERROR ) {
        printf("select failed: connection aborted\n");
        closesocket(sFS740);
        exit(1);
    }
    count = 0;
    if ( FD_ISSET(sFS740, &setRead) ) {
        /* We've received something */
        count = (int)recv(sFS740, buffer, num-1, 0);
        if ( SOCKET_ERROR == count ) {
            printf("Receive failed: connection aborted\n");
            closesocket(sFS740);
            exit(1);
        }
        else if ( count ) {
            buffer[count] = '\0';
        }
        else {
            printf("Connection closed by remote host\n");
            closesocket(sFS740);
            exit(1);
        }
    }
    return count;
}
```



# FS740 Operation Verification

## Overview

The operation of an FS740 may be evaluated by running a series of tests designed to measure the accuracy of its inputs and outputs and comparing the results with their associated specifications. While the verification tests presented here are not as extensive as the tests performed at the factory, one can nevertheless have confidence that a unit that passes these tests is functioning properly and within specification.

## Equipment Required

In addition to the FS740 under test, the following equipment will be required to carry out the performance tests:

- Agilent U2004A power meter: 9 kHz to 6 GHz
- Agilent 34410A DVM
- SRS SR620 time interval counter
- SRS CG635 clock generator

Equivalent equipment may be substituted as desired as long as they have similar or superior specifications. Standard BNC cables will be required to connect the test equipment to the FS740. Additionally accessories required include 50 Ω terminators and various adapters.

## Source Output Tests

The source output tests are intended to test the integrity of the source outputs: sine out, aux out, and pulse out.

### Sine Output Amplitude Test

The sine output test requires the setup shown in Figure 85. The 50 Ω terminator should be applied at the Sine output so that cable losses do not affect the measurements.

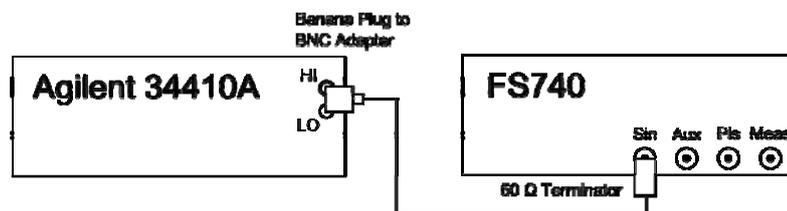


Figure 85: Sine output test setup

Configure the DVM for AC voltage measurements. Enable the math function for dBm calculations with a reference resistor of 50 Ω. Set the FS740 sine output frequency to 10 kHz. Then set the sine output amplitude to the values shown in the first column of Table 19 and record the value measured by the DVM in the second column.

Table 19: Sine output amplitude test values

Amplitude (dBm)	Recorded Value (dBm)	Error
5.00		
0.00		
-5.00		
-10.00		
-15.00		
-20.00		
-25.00		
-30.00		
-35.00		

Compute the error by subtracting the set amplitude from the recorded value and enter it in the third column. The absolute value of the error should be less than 0.4 dBm

Repeat the test for the rear sine output.

## Aux Output Amplitude Test

The aux output test requires the setup shown in Figure 86. The 50 Ω terminator should be applied at the Aux output so that cable losses do not affect the measurements.

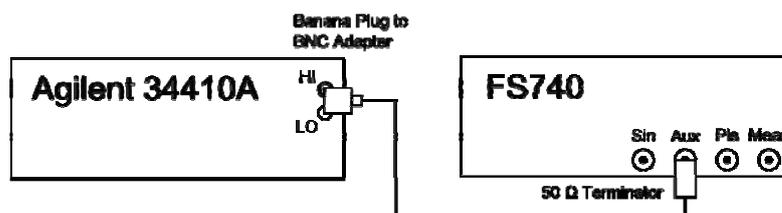


Figure 86: Aux output test setup

Configure the DVM for AC voltage measurements. Enable the math function for dBm calculations with a reference resistor of 50 Ω. Set the FS740 aux waveform to sine and output frequency to 10 kHz. Then set the aux output amplitude to the values shown in the first column of Table 20 and record the value measured by the DVM in the second column.

Table 20: Aux output amplitude test values

Amplitude (dBm)	Recorded Value (dBm)	Error
5.00		
0.00		
-5.00		
-10.00		
-15.00		
-20.00		
-25.00		
-30.00		
-35.00		

Compute the error by subtracting the set amplitude from the recorded value and enter it in the third column. The absolute value of the error should be less than 0.4 dBm

Repeat the test for the rear aux output.

### Aux 100 MHz Output Amplitude Test

The aux 100 MHz output amplitude test requires the setup shown in Figure 87.

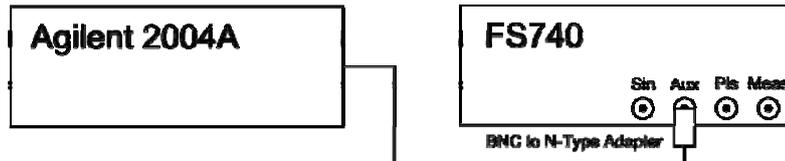


Figure 87: Aux 100 MHz output test setup

Ideally the Agilent power meter should be connected directly to the output with no intervening cables. Set the FS740 aux output to 100 MHz. Measure the power with the Agilent power meter. It should be  $2.75 \pm 0.5$  dBm.

### Sine Output Frequency Test

The sine output frequency test requires the setup shown in Figure 88.

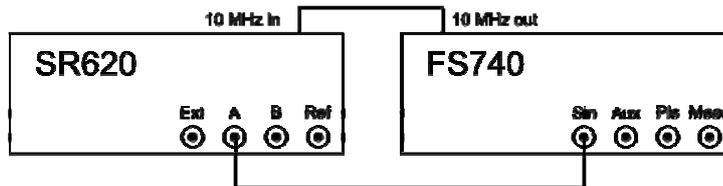


Figure 88: Sine output frequency test setup

Configure the SR620 to measure frequency on channel A, with a gate of 1.0 s, a sample size of 1, and a trigger level of 0.0 V, terminated into 50 Ω. Set the FS740 sine output amplitude to 1.0 V<sub>pp</sub>. Set the sine output frequency to each test value shown in Table 21 and record the frequency measured by the SR620 next to it in column 2.

Table 21: Sine output frequency test values

Set Frequency (MHz)	Recorded Frequency (Hz)	Relative Error
5.0		
10.0		
15.0		
20.0		
25.0		
30.0		

Compute the relative error for each measurement using the following equation:

---

The absolute value of the relative error should be less than 1e-9.

Repeat the test for the rear sine output.

## Aux Output Frequency Test

The aux output frequency test requires the setup shown in Figure 89.

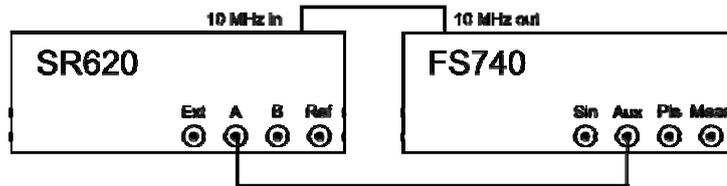


Figure 89: Aux output frequency test setup

Configure the SR620 to measure frequency on channel A, with a gate of 1.0 s, a sample size of 1, and a trigger level of 0.0 V, terminated into 50 Ω. Set the FS740 aux output to sine waves with amplitude 1.0 V<sub>pp</sub>. Set the aux output frequency to each test value shown in Table 22 and record the frequency measured by the SR620 next to it in column 2. For the final measurement, change the output waveform from sine waves to 100 MHz.

Table 22: Aux output frequency test values

Set Frequency (MHz)	Recorded Frequency (Hz)	Relative Error
2.0		
4.0		
6.0		
8.0		
10.0		
100 MHz waveform		

Compute the relative error for each measurement using the following equation:

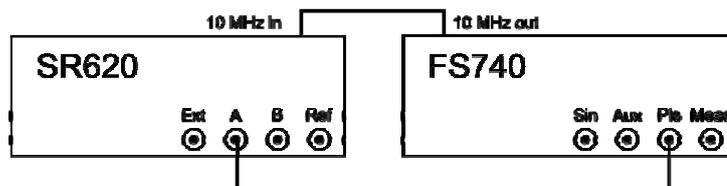
$$\frac{\text{Recorded Frequency} - \text{Set Frequency}}{\text{Set Frequency}}$$

The absolute value of the relative error should be less than 1e-9.

Repeat the test for the rear aux output.

## Pulse Width Test

The pulse width test requires the setup shown in Figure 90.



**Figure 90: Pulse width test setup.**

Configure the SR620 to measure pulse width on channel A with a sample size of 1000 and the trigger level set to 2.0 V. Configure the FS740 with a pulse period of 1.01  $\mu$ s and a pulse width of 100 ns. Take a measurement with the SR620 and then set the REL. New measurements should be close to zero. Now increase the pulse width of the FS740 in 1 ns steps and record the results in Table 23.

**Table 23: Pulse width test values**

Set Width (ns)	Recorded Width (ns)	Expected Width (ns)	Error
100.0		0.0	
101.0		1.0	
102.0		2.0	
103.0		3.0	
104.0		4.0	
105.0		5.0	
106.0		6.0	
107.0		7.0	
108.0		8.0	
109.0		9.0	
110.0		10.0	

Compute the error by subtracting the expected value from the recorded value. The absolute value of the error should be less than 0.1 ns.

Repeat the test for the rear pulse output.

## Pulse Frequency Test

The pulse frequency test uses the same setup as the pulse width test shown in Figure 90. Configure the SR620 for frequency measurements on channel A, a gate of 1 second, a sample size of 1, and a trigger level of 2.0 V. Configure the FS740 pulse output to source 1 MHz with a 50 % duty cycle. Record the frequency measured by the SR620 at the top of the second column in Table 24. Next, set the FS740 pulse output frequency to each frequency listed in Table 24, and record the frequency measured by the SR620 next to it in the table.

**Table 24: Pulse frequency test values**

Set Frequency (Hz)	Recorded Frequency (Hz)	Error
1,000,000.0		
1,000,001.0		
1,000,002.0		
1,000,003.0		
1,000,004.0		
1,000,005.0		
1,000,006.0		
1,000,007.0		
1,000,008.0		
1,000,009.0		
1,000,010.0		

Compute the error for each measurement by subtracting the recorded measurement from the set frequency. The absolute value of the error should be less than 0.001 Hz.

Repeat the test for the rear pulse output.

---

## Measurement Input Tests

The measurement input tests are designed to test the functionality of the front and rear measurement inputs.

---

## Conclusions

The tests described in this document are designed to test the basic functionality of the unit. They are not intended to be a substitute for the complete performance test which is performed at the factory. Nevertheless, one can have reasonable confidence that instruments that pass the tests described in this document are operating correctly. As always, if an instrument fails to pass a test, verify that the setup has been duplicated correctly, and that the individual procedures have been followed as specified. Instruments that have failed to meet specifications may be returned to SRS for repair.

## Circuit Description

This is the circuit description.



# Parts List

## Front Display (Assemblies 320 & 321)

Z42 SG396, SPACER BZL Fabricated component 7-02433

## Front Display EMI Filter (Assembly 324)

Ref	Value	Description	SRS P/N
C1	1000P	Capacitor, 0603, NPO	5-00740
C2	1000P	Capacitor, 0603, NPO	5-00740
Z0	SIM-PCB S/N	Label	9-01570

## Motherboard (Assemblies 322 & 323)

Ref	Value	Description	SRS P/N
C100	1000P	Capacitor, 0603, NPO	5-00740
C101	100000P	Capacitor, 0603, X7R	5-00764

## FS740 (Assembly 327)

Ref	Value	Description	SRS P/N
C100	10000P	Capacitor, 0603, X7R	5-00752
C101	4.7U - 16V X5R	Ceramic, 16V, 1206, X5R	5-00611
Z0	SIM-PCB S/N	Label	9-01570

## RF Synthesizer for SG396 (Assembly 333)

Ref	Value	Description	SRS P/N
C100	10000P	Capacitor, 0603, X7R	5-00752
C101	4.7U - 16V X5R	Ceramic, 16V, 1206, X5R	5-00611
Z39	SG396, BEZEL	Fabricated component	7-02432
Z40	SG394 Lexan	Fabricated component	7-02443
Z39	SG396, BEZEL	Fabricated component	7-02432
Z40	SG394 Lexan	Fabricated component	7-02443
Z39	SG396, BEZEL	Fabricated component	7-02432
Z40	SG394 Lexan	Fabricated component	7-02443
Z39	SG396, BEZEL	Fabricated component	7-02432
Z40	SG394 Lexan	Fabricated component	7-02443
Z39	SG396, BEZEL	Fabricated component	7-02432
Z40	SG394 Lexan	Fabricated component	7-02443
Z39	SG396, BEZEL	Fabricated component	7-02432
Z40	SG394 Lexan	Fabricated component	7-02443
Z41	SG392 Lexan	Fabricated component	7-02444



## Appendix A : Schematic Diagrams

Schematic 1: Block Diagram







---

# Revisions

Rev	Date	Changes
1.00	1/25/17	First release

Distribution in the UK & Ireland



**Characterisation,  
Measurement &  
Analysis**

**Lambda Photometrics Limited**  
Lambda House Batford Mill  
Harpenden Herts AL5 5BZ  
United Kingdom  
E: [info@lambdaphoto.co.uk](mailto:info@lambdaphoto.co.uk)  
W: [www.lambdaphoto.co.uk](http://www.lambdaphoto.co.uk)  
T: +44 (0)1582 764334  
F: +44 (0)1582 712084