



**SPECTRADYNAMICS, INC.**



**9.192 GHZ CESIUM FREQUENCY SYNTHESIZER  
CS-1  
PROGRAMMING MANUAL**

**SPECTRADYNAMICS, INC** • 1849 Cherry St. Unit 2 • Louisville, CO 80027  
Phone: (303) 665-1852 • Fax: (303) 604-6088  
[www.spectradynamics.com](http://www.spectradynamics.com)

# Table of Contents



Description.....	2
RS-232 Port.....	3
Communication Screen.....	4
ASCII Command Set.....	5
State Machine ASCII Command Set.....	32
Troubleshooting.....	43



## Description



The CS-1 is a versatile low noise synthesizer with an output frequency range of 9.18963177 GHz to 9.19563177 GHz. The CS-1 is designed to be used in the implementation of a Cesium atomic clock. The unit is provided in a 2U, 19 inch rack-mount enclosure. All synthesizer functions are accessed from the front panel or a RS-232 interface. An external trigger input may be used to synchronize programmable events such as frequency sweeps, phase modulation and amplitude modulation with external events.

The synthesizer has 48-bit frequency resolution, 14-bit phase resolution and 12-bit amplitude control. All instrument functions and settings are displayed and controlled via the front panel LCD display and keypad. Remote control of the instrument is possible through RS-232 communications. This manual describes the RS-232 ASCII command set of the CS-1 synthesizer.

## RS-232 Port



### RS-232 Communication Port

The CS-1 Synthesizer Module functions are accessed through the RS-232 port located on the front panel. A standard serial cable with a DB-9 connector can be used to interface to the unit. The user can input commands using a simple dumb terminal program on a remote computer or more sophisticated control can be used with software such as Labview.

On the front panel above the RS-232 connector there are 3 LEDs. The power LED labeled ON should be lit when power is applied. The second LED labeled DATA will light up only when data is being received or sent on the RS-232 port. This LED can be used to verify that the unit is communicating. The third LED is labeled status and is a hardware representation of the internal status flag. The status LED is on whenever an error has occurred. The user must query the unit to determine the source of error and then clear the error flag. When the error flag is cleared the LED will turn off.

### Port Settings

On power-up the RS-232 port settings are:

Baud rate 9600 8 Bits 1 Stop Bit No Parity.

Hardware handshaking is not used. The DB-9 connector pin-out is described below.

<u>Pin</u>	<u>Function</u>
1	NC
2	Data out
3	Data in
4	NC
5	GND
6	NC
7	NC
8	NC
9	NC

## Communications Screen



**Communications Screen** The communications screen displays the current RS-232 serial port settings. The soft keys at the bottom of the screen are used to set new RS-232 settings, initiate RS-232 control of the instrument or test the serial port connection. The RS-232 port is setup to be controlled by a dumb terminal. A null modem adapter is not needed and should not be used. Hardware handshaking is not used. For additional pin-out information please refer to the RS-232 port section on page 3 of this manual.

### DISPLAYS

Current baud rate setting.

### MENU

- |              |   |
|--------------|---|
| <b>REM</b>   | Enter remote RS-232 control mode.   |
| <b>BAUD</b>  | Toggle through available baud rates.<br><b>9600</b> , 19200, 38400, 57600, 115200, 14400, 28800 |
| <b>LOCAL</b> | Return to local control and terminate remote RS-232 control session.                            |
| <b>EXIT</b>  | Exit to previous screen.  |

# ASCII Command Set



The CS-1 command set is used to control most synthesizer functions. The characters sent to the CS-1 must be upper case ASCII characters.

<u>Command Summary</u>	<u>Description</u>	<u>Page</u>
AMASK [rt]	Set amplitude ramp time	7
AMASK?	Query amplitude ramp time setting	7
AMPL [ampl] [aunits]	Set amplitude in dBm, Vrms or Vpp	8
AMPL?	Query amplitude	8
ASK[askcmd]	Start/stop amplitude shift keying	9
BAUD [baud]	Change baud rate	10
BAUD?	Query baud setting	10
COFF [df]	Set frequency offset from Cs frequency	11
COFF?	Query frequency offset from Cs frequency	11
DATE [mo/day/yr]	Change date	12
DATE?	Query date	12
FMFSK [f1][f2]	Set frequency shift parameters	13
FMFSK?	Query frequency shift parameters	13
FMSWP [f1][f2][df][rate]	Change frequency sweep parameters	14
FMSWP?	Query frequency sweep parameters	14
FOFF [df]	Change frequency setting	15
FOFF?	Query frequency setting	15
FSK [fskcmd]	Start/stop frequency shift keying	16
FREQ [freq]	Change frequency setting	17
FREQ?	Query frequency setting	17
HELP	Basic help on ASCII command set	18
LOCL	Return to local control	19
PHAS [phase]	Set synthesizer phase	20
PHAS?	Query synthesizer phase	20
PLL [pllcmd]	Set Int/Ext frequency reference	21
PLL?	Query phase lock loop status	22
PMPSK [p1][p2]	Set phase shift keying parameters	23
PMPSK?	Query phase shift keying parameters	23
PSK [pskcmd]	Enable/disable phase shift keying	24
RFPWR [rf]	Turn on/off RF power	25
RFPWR?	Query RF power setting	25
SWP [swpcmd]	Start/stop frequency sweep	26
TEMP?	Query instrument temperature	27

# ASCII Command Set



TIME [hr:min:sec]	Change instrument time of day	28
TIME?	Query time of day	28
TRIG	Software trigger signal	29
*RST	Reset synthesizer	30
*SRE	Get status byte	31
*CLS	Clear status byte	31

## Command Arguments

rt	Amplitude ramp time from on/off or off/on. {164uS – 10485uS }
ampl	Amplitude entered in units indicated by aunit –10 dBm to +15 dBm}
aunits	Amplitude units {1 – dBm, 2 – Vrms, 3 – Vpp}
askcmd	Amplitude shift command {0 – off, 1 – on/trigger, 2 – signal on, 3 – signal off}
auxcmd	Auxiliary output command {0 – off, 1 – on}
baud	Serial port baud rate. The default is 9600.
mo	Month {1 – 12}
day	Day {1 – 31}
year	Year {1900 – 2100}
f1	Start frequency in Hz {9189631770 Hz – 9195631770 Hz}
f2	Stop frequency in Hz {f1 – 9195631770 Hz}
df	Step frequency in Hz {1uHz – 6MHz}
rate	Step rate in Hz {48 Hz – 25 MHz}
fskcmd	Frequency shift keying command {0 – off, 1 – on/trigger, 2 – set f1, 3 – set f2}
freq	Frequency in Hz {9189631770 Hz – 9195631770 Hz }
phase	Phase in degrees {+/- 360}
pllcmd	PLL command {0 – DC Tuning, 1 – Lock to Ext 5 MHz}
p1	First phase setting in degrees {+/-360}
p2	Second phase setting in degrees {+/-360}
rf	Parameter to turn on/off RF power {0 – off, 1 – on}
pskcmd	Phase shift command {0 – off, 1 – on/trigger, 2 – set p1, 3 – set p2}
swpcmd	Sweep command {0 - off, 1 – on/trigger, 2 – autosweep}
hr	Hour {0 – 23}
min	Minute {0 – 59}
sec	Second {0 – 59}



## AMASK, AMASK?



**AMASK** sets the ramp time of the output amplitude from off to on or on to off.

**AMASK [rt]<cr>**

AMASK has 1 parameter and is executed following the carriage return <cr>

**rt** valid values are in micro-seconds: **164 – 10485**

**Example: AMASK 165<cr>**

Sets the ramp time to 165 microseconds. <cr> is a carriage return.

**AMASK?** Queries the synthesizer ramp time setting.

**AMASK?<cr>**

AMASK? is executed following the carriage return <cr>

The results of the query are in the following format.

**AMASK? [rt]<cr>**

**Example: AMASK?<cr>**

Queries the ramp time setting of the synthesizer. <cr> is a carriage return.  
The function will return:

**AMASK? 165<cr>**

# AMPL, AMPL?



**AMPL** sets the synthesizer amplitude.

**AMPL [ampl][aunits]<cr>**

AMPL has 2 parameters and is executed following the carriage return <cr>

**ampl** amplitude in desired unit specified by aunits  
**aunits** amplitude units desired

**valid values:**

<b>ampl</b>	<b>aunits</b>	<b>units</b>
-10 to +15	1	dBm
0.071 to 1.26	2	Vrms
0.2 to 3.56	3	Vpp

**Example: AMPL 13.0 1<cr>**

Sets the amplitude to 13.0 dBm. <cr> is a carriage return.

**AMPL?** Queries the synthesizer amplitude setting.

**AMPL?<cr>**

AMPL? is executed following the carriage return <cr>

The results of the query are in the following format.

**AMPL? [ampl][units]<cr>**

**Example: AMPL?<cr>**

Queries the amplitude setting of the synthesizer. <cr> is a carriage return.  
The function will return:

**AMPL? 13.0 dBm<cr>**

# ASK



**ASK** enables or disables the amplitude shift keying. When amplitude shift keying is enabled a rising edge on the external trigger will toggle the amplitude on/off. A software trigger will do the same. Use the TRIG command to issue a software trigger.

**ASK [askcmd]<cr>**

ASK has 1 parameter and is executed following the carriage return <cr>

**askcmd amplitude shift command:**

- 0 – disabled**
- 1 – enabled**
- 2 – set amplitude on**
- 3 – set amplitude off**

**Example: ASK 0<cr>**

Disables amplitude shift keying.<cr> is a carriage return.

## BAUD, BAUD?



**BAUD** changes the baudrate of the synthesizer.

**BAUD [baud]<cr>**

BAUD has 1 parameter and is executed following the carriage return <cr>

**baud baudrate valid values are:**

<b>9600</b>	19200	38400	57600	115200
14400	28800			

**Example: BAUD 9600<cr>**

Sets the baudrate of the synthesizer to 9600. <cr> is a carriage return.

**BAUD?** Queries the synthesizer baudrate.

**BAUD?<cr>**

BAUD? is executed following the carriage return <cr>

The results of the query are in the following format.

**BAUD? [baud]<cr>**

**Example: BAUD?<cr>**

Queries the baudrate setting of the synthesizer. <cr> is a carriage return.  
The function will return:

**BAUD? 9600<cr>**

## COFF, COFF?



**COFF** sets the synthesizer offset from Cesium frequency.

**COFF [df]<cr>**

COFF has 1 parameter and is executed following the carriage return <cr>

**df** is the frequency offset in Hz from the Cesium frequency of 9192631770 Hz.

**df** range is +/- 3 MHz, resolution is 1.0 E-6 Hz.

**Example: COFF 1.0<cr>**

Sets the frequency of the synthesizer to 9192631771 Hz. <cr> is a carriage return.

**COFF?** Queries the synthesizer offset from the Cesium frequency.

**COFF?<cr>**

COFF? is executed following the carriage return <cr>

The results of the query are in the following format.

**COFF? [df]<cr>**

**Example: COFF?<cr>**

Queries the frequency offset from 9192631770 Hz. <cr> is a carriage return.  
The function will return:

**COFF? 1Hz<cr>**

## DATE, DATE?



**DATE** changes the date setting of the instrument.

**DATE [month/day/year]<cr>**

DATE has 3 parameters and is executed following the carriage return <cr>

<b>mo</b>	<b>month valid values are 1 - 12.</b>
<b>day</b>	<b>day valid values are 1 - 31.</b>
<b>year</b>	<b>year valid values are 1900 – 2100.</b>

**Example: DATE 02/02/2001<cr>**

Sets the date of the instrument to February 2, 2001. <cr> is a carriage return.

**DATE?** queries the instrument date setting.

**DATE?<cr>**

DATE? is executed following a carriage return.

The results of the query are in the following format.

**DATE? [mo/day/year]<cr>**

**Example: DATE?<cr>**

Queries the date setting of the instrument. <cr> is a carriage return.  
The function will return:

**DATE? 02/02/2001<cr>**

## FMFSK, FMFSK?



**FMFSK** is used set the frequency shift keying parameters.

**FMFSK [f1] [f2]<cr>**

FMFSK has two parameters and is executed following the carriage return <cr>

**f1** start or first frequency in Hz, valid range is 9189631770 Hz – 9195631770 Hz.

**f2** stop or second frequency in Hz, valid range is f1 to 9195631770 Hz.

**Example: FMFSK 9190000000 9191000000<cr>**

This command will set the first frequency to be 9.19 GHz and the second frequency to be 9.191 GHz. <cr> is a carriage return.

**FMFSK?** queries the instrument frequency shift keying settings.

**FMFSK?<cr>**

FMFSK? is executed following a carriage return.

The results of the query are in the following format.

**FMFSK? [f1] [f2]<cr>**

**Example: FMFSK?<cr>**

Queries the instrument frequency shift keying settings. <cr> is a carriage return.

The function will return:

**FMFSK? 9190000000 9191000000<cr>**

## FMSWP, FMSWP?



**FMSWP** is used set the frequency sweep parameters.

**FMSWP [f1] [f2][df][rate]<cr>**

FMSWP has four parameters and is executed following the carriage return <cr>

- f1** start or first frequency in Hz, valid range is 9189631770 Hz – 9195631770 Hz.
- f2** stop or second frequency in Hz, valid range is f1 to 9195631770 Hz.
- df** step frequency in Hz, valid range is 1uHz to 6 MHz.
- rate** the frequency step rate, valid range is 48 Hz to 25 MHz.

**Example: FMSWP 9189631770 9195631770 1 100000<cr>**

This command will set the start frequency to be 9189631770 Hz and the stop frequency to be 9195631770 Hz. The frequency will increment with a step size of 1 Hz at a rate of 100000 steps per second. <cr> is a carriage return.

**FMSWP?** queries the instrument frequency sweep settings.

**FMSWP?<cr>**

FMSWP? is executed following a carriage return.

The results of the query are in the following format.

**FMSWP? [f1] [f2] [df] [rate]<cr>**

**Example: FMSWP?<cr>**

Queries the instrument frequency sweep settings. <cr> is a carriage return.  
The function will return:

**FMSWP? 9189631770 Hz 9195631770 Hz 1 Hz 100000 Hz<cr>**



## FOFF, FOFF?



**FOFF** is used to add a frequency offset to the synthesizer frequency.

**FOFF [df] <cr>**

FOFF has one parameter and is executed following the carriage return <cr>

**df**                **range of the final frequency must be less than +/- 3MHz from 9195631770 Hz, resolution is 1.0 E-6Hz.**

**Example:    FOFF 0.001<cr>**

The frequency of the CS-1 will be set to 9189631770.001 Hz. <cr> is a carriage return.

**FOFF?** queries the last frequency offset entered.

**FOFF?<cr>**

FOFF? is executed following a carriage return.

The results of the query are in the following format.

**FOFF? [df]<cr>**

**Example:    FOFF?<cr>**

Queries the last frequency offset entered. <cr> is a carriage return.  
The function will return:

**FREQ? 0.001Hz<cr>>**

# FSK



**FSK** enables or disables the frequency shift keying. When frequency shift keying is enabled a rising edge on the external modulation input will toggle the frequency from f1 to f2 or vice-versa. A software trigger will do the same. Use the TRIG command to issue a software trigger.

## **FSK [fskcmd]<cr>**

FSK has 1 parameter and is executed following the carriage return <cr>

### **fskcmd frequency shift command:**

**0 – disabled**

**1 – enabled**

**2 – set frequency to f1**

**3 – set frequency to f2**

**Example: FSK 0<cr>**

Disables frequency shift keying.<cr> is a carriage return.

## FREQ, FREQ?



**FREQ** is used to set the synthesizer frequency.

**FREQ [freq] <cr>**

FREQ has one parameter and is executed following the carriage return <cr>

**freq** range is 9189631770 Hz – 9195631770 Hz, resolution is 1.0 E-6Hz.

**Example: FREQ 9189631770.001<cr>**

The frequency of the CS-1 will be set to 9189631770.001 Hz. <cr> is a carriage return.

**FREQ?** queries the instrument frequency setting.

**FREQ?<cr>**

FREQ? is executed following a carriage return.

The results of the query are in the following format.

**FREQ? [freq]<cr>**

**Example: FREQ?<cr>**

Queries the frequency setting of the instrument. <cr> is a carriage return.  
The function will return:

**FREQ? 9189631770.001 Hz<cr>>**

# HELP



**HELP** is used to display basic help on the ASCII command set.

**HELP**<cr>

HELP is executed following the carriage return <cr>

**Example:**   **HELP**<cr>

# LOCL



---

**LOCL** turns off the RS-232 communications and returns control to the LCD screen and keypad on the instrument.

**LOCL<cr>**

LOCL is executed following the carriage return <cr>

## PHAS, PHAS?



**PHAS** changes the phase of the synthesizer.

**PHAS [phase]<cr>**

PHAS has one parameter and is executed following the carriage return <cr>

**phase** value is in degrees, the range is 0 to +/- 360, and the resolution is 0.022 deg.

**Example: PHAS 36<cr>**

Sets the output phase of the CS-1 to +36 degrees from the phase = 0 condition. Note that if **phase** = 360 deg and we issue a **PHAS 360<cr>** command the output phase will not change because the synthesizer output is already at 360 deg.

**PHAS?** queries the instrument phase.

**PHAS?<cr>**

PHAS? is executed following a carriage return.

The results of the query are in the following format.

**PHAS? [phase]<cr>**

**Example: PHAS?<cr>**

Queries the phase of the instrument. <cr> is a carriage return.  
The function will return:

**PHAS? 36 deg<cr>**



**PLL** is used to select the frequency reference for the synthesizer. The phase locked loop can be enabled to allow locking of the synthesizer to an externally provided 5 MHz reference. DC tuning of the internal 5 MHz reference is selected when the PLL is turned off.

**PLL [pllcmd] <cr>**

PLL has one parameter and is executed following the carriage return <cr>

**pllcmd**      **0 – pll disabled. DC tuning of internal 5 MHz reference is selected. The tuning voltage is applied to the external reference SMA input. The tuning voltage should not exceed +/-5 VDC. The tuning bandwidth is approximately 300 Hz and the tuning port impedance is 10 kOhm.**

**1 – pll enabled. Locking to an externally provided 5 MHz signal is enabled. The 5 MHz signal should be applied to the external reference SMA input. The level of the 5 MHz signal should be between +7 dBm to +15 dBm for optimal operation. Locking to signals as low as +3 dBm is possible, however there may be a degradation in the close-in phase noise.**

**Example: PLL 1<cr>**

PLL is enabled and the synthesizer will lock to an external 5 MHz reference. <cr> is a carriage return.

## PLL?



**PLL?** queries the critical levels and voltages in the main instrument control loop.

### **PLL?<cr>**

PLL? is executed following the carriage return <cr>

**5 MHz** power level of the internal 5 MHz signal. It should be about +12 dBm.

**Ref** power level of the 5 MHz external reference signal. It should be between +7 and +15 dBm.

**100 MHz** power level of the 100 MHz signal. It should be at least +11 dBm.

**LVn:** indicates the locking voltage for the three PLLs of the CS-1.  $n=1$  is the 5 MHz PLL, while  $n=2$  is the 100 MHz PLL. They are locked when the displayed value is between 0.2 and 0.3.  $n=3$  is the PLL for the DRO and it is locked when the displayed value is 2.5.

**TVn:** indicates the tuning voltage for the internal 5 MHz and 100 MHz oscillators. It should be within  $\pm 5$ .

**DC tuning enabled:** indicates that the Internal Reference configuration is enabled.  
*This is displayed in place of Ref, LV1 and TV1 because the 5 MHz PLL is disabled.*

**Note** that if the tuning port voltage exceeds  $\pm 5$  V the instrument will lose phase-lock to the external reference. This voltage can also be monitored and used to decide when the internal oscillator must be mechanically tuned to adjust for aging.

### **Example: PLL?<cr>**

Queries the critical levels and voltages in the CS-1 control loop. <cr> is a carriage return.



## PMPSK, PMPSK?



**PMPSK** is used to set the phase shift keying parameters.

**PMPSK [p1] [p2]<cr>**

PMPSK has two parameters and is executed following the carriage return <cr>

**p1** first phase in degrees, valid range is +/- 360 degrees.

**p2** second phase, valid range is +/- 360 degrees.

**Example: PMPSK 0.0 180.0<cr>**

The output of the synthesizer will change in phase from 0 degrees to 180 degrees.  
<cr> is a carriage return.

**PMPSK?** queries the phase shift keying parameters.

**PMPSK? <cr>**

PMPSK? executed following a carriage return.

The query returns the phase shift keying parameters.

The results of the query are in the following format.

**PMPSK? [p1] [p2]<cr>**

**Example: PMPSK?<cr>**

Queries the phase shift keying parameters. <cr> is a carriage return.  
The function will return:

**PMPSK? 0.0Deg 180.0Deg<cr>**

# PSK



**PSK** enables or disables phase shift keying. When phase shift keying is enabled a rising edge on the external modulation input will toggle the phase from p1 to p2 or vice-versa. A software trigger will do the same. Use the TRIG command to issue a software trigger.

**PSK [pskcmd]<cr>**

PSK has 1 parameter and is executed following the carriage return <cr>

**pskcmd phase shift command:**

- 0 – disabled**
- 1 – enabled**
- 2 – set phase to p1**
- 3 – set phase to p2**

**Example: PSK 0<cr>**

Disables phase shift keying.<cr> is a carriage return.

## RFPWR, RFPWR?



**RFPWR** enables or disables the 9.1XX GHz RF output.

**RFPWR [rf]<cr>**

RFPWR has 1 parameter and is executed following the carriage return <cr>

**rf command:**

**0 – disabled**

**1 – enabled**

**Example: RFPWR 0<cr>**

Disables RF output.<cr> is a carriage return.

**RFPWR?** queries the state of the RF output port.

RFPWR? executed following a carriage return.

The query returns the status of the RF port. It returns 0 if disabled and 1 if enabled.

The results of the query are in the following format.

**RFPWR? [rf]<cr>**

**Example: RFPWR?<cr>**

Queries the status of the RF port. <cr> is a carriage return.

The function will return:

**RFPWR? 0<cr>**



**SWP** enables or disables frequency sweep function. When frequency sweeping is enabled a rising edge on the external modulation input will toggle the frequency sweep from f1 to f2 or vice-versa. A software trigger will do the same. Use the TRIG command to issue a software trigger.

## **SWP [swpcmd]<cr>**

SWP has 1 parameter and is executed following the carriage return <cr>

### **swpcmd frequency sweep command:**

- 0 – disabled**
- 1 – enabled**
- 2 – auto sweep**

**Example: SWP 1<cr>**

Enables the frequency sweep function.<cr> is a carriage return.

**TEMP?**



**TEMP?** Queries the system temperature.

**TEMP?<cr>**

TEMP? is executed following a carriage return,

**Example: TEMP?<cr>**

Returns

**TEMP? 40.1C<cr>**

## TIME, TIME?



**TIME** changes the time of day of the instrument.

**TIME [hr:min:sec] <cr>**

TIME has 3 parameters and is executed following the carriage return <cr>

**hr** Hour valid range 0 – 23

**min** Minutes valid range 0 – 59

**sec** Seconds valid range 0 – 59

**Example: TIME 12:01:00<cr>**

Sets the time of day to 12 hours, 1 minute and 0 seconds. <cr> is a carriage return.

**TIME?** Queries the system time of day.

**TIME?<cr>**

TIME? is executed following a carriage return,

**Example: TIME?<cr>**

Returns

**TIME? 12:01:31<cr>**

# TRIG



**TRIG** issues a software trigger.

**TRIG** <cr>

TRIG has no parameters and is executed following the carriage return <cr>

**Example:** TRIG<cr>

Issues a software trigger. This is the equivalent of a rising edge arriving on the external modulation input connector. <cr> is a carriage return.

## **\*RST**



**\*RST** resets the synthesizer to default power on settings.

**\*RST<cr>**

\*RST is executed following the carriage return <cr>

**Example: \*RST<cr>**

Resets the synthesizer. <cr> is a carriage return.



## \*SRE, \*CLS



**\*SRE** queries the synthesizers for the value of the status register.

**\*SRE<cr>**

\*SRE is executed following a carriage return.

The results of the query are in the following format. SRE [status]<cr>

**status** is a 16-bit value that contains the sum of the error conditions. status is 0 when there are no errors.

External reference error	0x0001
5MHz oscillator error	0x0002
External PLL Lock error	0x0004
5MHz Tuning voltage error	0x0008
100MHz oscillator error	0x0010
100MHz PLL lock error	0x0020
100MHz Tuning voltage error	0x0040
DRO PLL error	0x0080
Temperature error	0x0100
Time error	0x0200
Command not recognized	0x0400
Invalid parameter	0x0800
Reserved	0x1000
Reserved	0x2000
Reserved	0x4000
Reserved	0x8000

**Example:**   **\*SRE<cr>**   query status  
                  **SRE 4096**   the return value indicates that an invalid parameter was sent to the synthesizer.

**\*CLS** clears the status register and turns off the status LED.

**\*CLS<cr>**

\*CLS is executed following a carriage return.

# State Machine ASCII Command Set



The State Machine command set is used to program the synthesizer behavior in state machine mode. All the state machine commands expect the frequency parameters to be the IF frequency of the synthesizer before the single sideband translation to 9 GHz. The frequency that is set with these commands is the frequency of the signal that appears on the IF monitor port on the front panel of the synthesizer. The amplitude units are expected to be in dBm and the phase parameters are expected to be in degrees. The characters sent to the CS-1 must be upper case ASCII characters.

Command Summary	Description	Page
CSTATE [state]	Manually execute a selected state	33
GETST [state]	Query the synthesizer parameters of a state	34
IFFREQ [freq]	Set the IF frequency	35
IFFREQ?	Query IF frequency	35
NUMST [nstates]	Sets the total number of states	36
NUMST?	Query the number of states	36
SETST [state][ampl][phase][freq][fstep][rate]	Setup the synthesizer parameters of a state	37
SME	Display message corresponding to error code	38
STATE [state]	Set the current state to state	39
STATE?	Query current state	39
STMOFF	Turn off the state machine	40
STMON [state]	Turn on the state machine at state	41
*SME	Get state machine error code	42

# CSTATE



**CSTATE** is used to execute the selected state without turning on the state machine and waiting for the appropriate hardware trigger signal. This command allows the user to try out the selected state.

**CSTATE [state]<cr>**

**CSTATE** has one parameter and is executed following the carriage return <cr>

**state – the selected state to execute, this number should be between 1 and nstates which is the number of states in the state machine.**

**Example: CSTATE 1<cr>**

This command programs the synthesizer with the parameters that are define by state number one. Once the synthesizer is programmed, an external trigger signal is simulated and the synthesizer executes state one. <cr> is a carriage return.

# GETST



**GETST** is used to retrieve the parameters that define the state given by the **state** parameter.

**GETST [state]<cr>**

**GETST** has one parameter and is executed following the carriage return <cr>

**state** – the selected state to query, this number should be between 1 and n states which is the number of states in the state machine.

**Example:**

**GETST 1<cr>**

This command retrieves the parameters that are defined by state number one.

Returns

**GETST 0 0 1000000.00000 0 0<cr>**

The parameters that are returned are:

**GETST [ampl] [phas] [freq] [fstep] [rate]<cr>**

ampl – the amplitude in dBm

phas – the phase in degrees

freq – the frequency in Hz

fstep – the frequency step size in Hz

rate – the rate at which the synthesizer changes frequency

<cr> is a carriage return.

## IFFREQ, IFFREQ?



**IFFREQ** sets the synthesizer IF frequency.

**IFFREQ [freq]<cr>**

IFFREQ has 1 parameter and is executed following the carriage return <cr>

**freq** – is the IF frequency in Hz. The range is 4368230 Hz to 10368230 Hz.

**Example: IFFREQ 10000000<cr>**

Sets the IF frequency of the synthesizer to 10 MHz. <cr> is a carriage return.

**IFFREQ?** Queries the synthesizer IF frequency.

**IFFREQ?<cr>**

IFFREQ? is executed following the carriage return <cr>

The results of the query are in the following format.

**IFFREQ? [freq]<cr>**

**Example: IFFREQ?<cr>**

Queries the synthesizer IF frequency. <cr> is a carriage return.

The function will return:

**IFFREQ? 10000000.00000Hz<cr>**

## NUMST, NUMST?



**NUMST** sets the number of states that will be in the state machine.

**NUMST [nstate]<cr>**

NUMST has 1 parameter and is executed following the carriage return <cr>

**nstate – is the number of states in the state machine. The range is 1 to 10.**

**Example: NUMST 3<cr>**

Sets the number of states in the state machine to 3. <cr> is a carriage return.

**NUMST?** Queries the number of states in the state machine of the synthesizer.

**NUMST?<cr>**

INUMST? is executed following the carriage return <cr>

The results of the query are in the following format.

**NUMST? [nstate]<cr>**

**Example: NUMST?<cr>**

Queries the number of states in the state machine of the synthesizer. <cr> is a carriage return. The function will return:

**NUMST? 3<cr>**

# SETST



**SETST** is used to set the value for all the parameters that define the selected state.

**SETST [state] [ampl] [phase] [freq] [fstep] [rate]<cr>**

**SETST** has six parameters and is executed following the carriage return <cr>

**state** – the selected state to query, this number should be between 1 and nstates which is the number of states in the state machine.

**ampl** – the amplitude for the selected state in dBm. The range is 15 to -10.

**phas** – the phase for the selected state in degrees. The range is -360 to 360.

**freq** – the frequency for the selected state in Hz. The range is 4368230 Hz to 10368230 Hz.

**fstep** – the frequency step size for the selected state. The range is 0.00001 Hz to 10368230 Hz.

**rate** – the step rate for frequency sweeps. The range is 48 Hz to 25 MHz.

**Example:**

**SETST 1 0 0 10000000 1000 1000<cr>**

programs state one to amplitude of 0 dBm, phase of 0 degrees, frequency of 10MHz, the frequency step size is 1kHz and the frequency step rate is 1000 steps/second.

<cr> is a carriage return.



**SME** is used to retrieve a text message of the state machine status register.

**SME <cr>**

**SME** has one parameter and is executed following the carriage return <cr>

**Example:**

**SME<cr>**

This command retrieves text message describing the state machine status register.

Returns

**NO ERROR<cr>**



## STATE, STATE?



**STATE** sets the active state of the state machine..

**STATE [state]<cr>**

STATE has 1 parameter and is executed following the carriage return <cr>

**state – is the state to be loaded into the synthesizer on the next trigger.**

**Example: STATE 3<cr>**

state three will be executed by the synthesizer state machine on the next trigger signal.

<cr> is a carriage return.

**STATE?** Returns the active state of the synthesizer state machine.

**STATE?<cr>**

STATE? is executed following the carriage return <cr>

The results of the query are in the following format.

**STATE? [state]<cr>**

**Example: STATE?<cr>**

Returns the active state of the synthesizer state machine. <cr> is a carriage return.

The function will return:

**STATE? 3<cr>**

# STMOFF



**STMOFF** is used to turn off the state machine mode. The state machine function is turned off and the synthesizer reverts to the state it was in prior to the state machine mode selection. The external trigger will no longer be active.

**STMOFF <cr>**

**STMOFF** is executed following the carriage return <cr>

**Example:**

STMOFF<cr>



**STMON** sets up the initial state of the state machine and activates the external trigger. After this command executes, the next external trigger initiates the state machine and executes the selected initial state.

**STMON [state] <cr>**

**STMON** has one parameter and is executed following the carriage return <cr>

**state – the initial state of the state machine. Range 1 to nstate.**

**Example:**

**STMON 2<cr>**

This command sets the initial state of the state machine to state number 2 and initiates the state machine mode.

## **\*SME**



**\*SME** is used to retrieve the state machine status register.

**\*SME <cr>**

**SME** is executed following the carriage return <cr>

**Example:**

**\*SME<cr>**

This command retrieves the state machine status register.

Returns

**\*SME 0<cr>**

**An status code of 0 indicates there are no errors. Text for non zero status codes can be obtained with the SME command. The status register is reset after every state machine related command except \*SME and SME.**



## **DC Power Check**

Check the LED indicators on the DC power module. If all LEDs are on then check the CS-1 Synthesizer back panel LED indicators. If any of the indicators on the CS-1 synthesizer are not ON then check the DC cables for continuity.

## **Synthesizer will not lock to an external reference.**

Make sure the synthesizer has warmed up for at least 30 minutes. Make sure that the external reference frequency is 5 MHz +/- 0.5 Hz and that the level is in the +10 to +15 dBm range.

## **STATUS LED is on**

Activate the PLL Menu and check the PLL status and make sure that the external reference is adequate. If the RS-232 control mode is being used use the \*SRE command to read the status register and the \*CLS command to clear the status register.

## **RS-232 communications failed**

Check that the correct RS-232 cable is being used.

Do not use a null modem adapter.

Check baud rate of the CS-1 and set the controller to the same baud rate.

Use the TEST function to test the serial connection.

The DATA led should flash when data is being received by the CS-1.

Make sure that the CS-1 is in remote control mode and not in local control mode.

## **If any error condition persists please contact technical support.**

SpectraDynamics, Inc.  
1849 Cherry Street Unit 2.  
Louisville, CO 80027  
USA

Tel: (303) 665-1852  
Fax: (303) 604-6088  
support@spectradynamics.com  
www.spectradynamics.com

