Crystal Technology, Inc.

AOTF Controller Command Reference

Revision 1.3

2010/08/10

Document #60-00105-01

Crystal Technology, Inc.
A Group Company of TDK-EPC Corporation

Crystal Technology, Inc. reserves the right to make changes to its products or to discontinue any product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current and complete.

Crystal Technology, Inc. warrants the performance of its products and related software to the specifications applicable at the time of sale in accordance with Crystal Technology's standard warranty. Testing and other quality control techniques are utilized to the extent Crystal Technology, Inc. deems necessary to support this warranty. Specific testing of all parameters is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

Crystal Technology, Inc. products are not designed, intended, authorized, or warranted to be suitable for use in life support applications, devices, systems, or other critical applications.

Inclusion of Crystal Technology, Inc. products in such applications is understood to be fully at the risk of the customer. Use of Crystal Technology, Inc. products in such applications requires the written approval of an appropriate Crystal Technology, Inc. officer. Questions concerning potential risk applications should be directed to Crystal Technology, Inc. at the address listed below.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

Crystal Technology, Inc. assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does Crystal Technology, Inc. warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of Crystal Technology, Inc. covering or relating to any combination, machine, or process in which such products or services might be or are used.

This document has been prepared with the greatest care regarding its contents. However, in the event that it contains omissions, errors or any other misinformation, please feel free to direct comments to Crystal Technology, Inc. at the address listed below.

You are also welcome to call Crystal Technology, Inc. at (650) 856-7911, or you may FAX inquiries to (650) 354-0173.

Crystal Technology, Inc.
1040 East Meadow Circle
Palo Alto, CA 94303-4230

Revision History			
Revision	**Date**	**Who**	**Comments**
0.8	2005/12/15	Dale Gifford	Genesis, integrated into the Octal Channel AOTF Controller Technical Specifications.
1.0	2008/12/01	Calvin Hwung	Re-organized and updated from Octal Channel AOTF Controller Technical Specifications.
1.1	2008/12/05	Dale Gifford	Consolidated commands from Octal and Single AOTF Controller documentation.
1.2	2009/01/13	Dale Gifford	Added the "`config pid`" command.
1.3	2010/08/10	Dale Gifford	Updated the "`Track Adjust`" command, updated logo. Release 2010-08.

Table 1: Revision History

Table of Contents

List of Tables

List of Figures

This Page Intentionally Left Blank

1. Introduction

1.1. Purpose

This document describes the firmware commands available for Crystal Technology's Single, Quad, and Octal Channel *Acousto-Optic Tunable Filter (AOTF) Controllers*.

1.2. Related Documents

The following references may be useful in fully understanding and utilizing the *AOTF Controller*:

- Octal Channel AOTF Controller Integration Guide, Revision 1.1, 2010/08/10, www.CrystalTechnology.com, Crystal Technology, Inc. 1040 East Meadow Circle, Palo Alto, CA 94303-4230.

- Quad Channel AOTF Controller Integration Guide, Revision 1.1, 2010/08/10, www.CrystalTechnology.com, Crystal Technology, Inc. 1040 East Meadow Circle, Palo Alto, CA 94303-4230.

- Single Channel AOTF Controller Integration Guide, Revision 1.1, 2010/08/10, www.CrystalTechnology.com, Crystal Technology, Inc. 1040 East Meadow Circle, Palo Alto, CA 94303-4230.

- AOTF Controllers and FSK Operation, Revision 1.3, www.CrystalTechnology.com, Crystal Technology, Inc. 1040 East Meadow Circle, Palo Alto, CA 94303-4230.

- AOTF Controllers and Temperature Compensation, Revision 1.2, www.CrystalTechnology.com, Crystal Technology, Inc. 1040 East Meadow Circle, Palo Alto, CA 94303-4230.

- AOTF Controllers and Light Intensity Tracking, Revision 1.2, www.CrystalTechnology.com, Crystal Technology, Inc. 1040 East Meadow Circle, Palo Alto, CA 94303-4230.

- AotfManager User's Guide, Revision 1.1, www.CrystalTechnology.com, Crystal Technology, Inc. 1040 East Meadow Circle, Palo Alto, CA 94303-4230.

- AotfCmd User's Guide, Revision 1.1, www.CrystalTechnology.com, Crystal Technology, Inc. 1040 East Meadow Circle, Palo Alto, CA 94303-4230.

1.3. Notation

- Numbers with an "h" suffix or "0x" prefix are hexadecimal. All other numbers are decimal.
- Register and bit names ending in "**[#]**" and "**[#:#]**" signify selection of a subset of the register (e.g. **I2CS[0]** represents bit 0 of the **I2CS** register, and **I2CS[5:3]** represents bit 5 through 3 of the **I2CS** register).
- Signal names ending with '#' (e.g. **INT0#**) indicates an active low signal.

- N/A is an abbreviation for Not Applicable.
- Register bits are either set (1) or cleared (0).

2. AOTF Controllers

At the time of this publication there are three models of *AOTF Controllers*:

- Octal AOTF Controller, standard and high power.
- Quad AOTF Controller.
- Single AOTF Controller.

The commands available for each model of *AOTF Controller* are mostly identical. There are only minor variations to support functions and capabilities specific to particular models. Since the commands are mostly identical, this document serves as a reference for all models.

3. Command Reference

All *AOTF Controllers* accepts commands from the RS232 interface, the USB interface, or the Bluetooth interface. The commands are sequences of ASCII characters. Command lines are delimited by Carriage Return (ASCII character 0x0D) or Line Feed (ASCII character 0x0A). Multiple commands can be entered on a single command line by delimiting them with semicolons (“;”).

Commands are composed of Verbs and Arguments, where each command begins with a Verb followed by any required Arguments. Verbs and Arguments are matched against keywords by comparing all of the ASCII characters in the Verb or Argument with valid keywords. A match is obtained if the Verb or Argument matches the beginning of the keyword. A Verb or Argument can be truncated to the minimum number of characters necessary to uniquely match a keyword.

The commands are divided into multiple categories that classify related commands. The command categories are:

- Help – Display helpful text about command Verbs and Arguments.
- I2C – Perform I2C bus related operations.
- EEPROM – Perform EEPROM related operations (i.e. boot EEPROM operations).
- ADC – Perform Analog to Digital conversions (i.e. read sensor inputs).
- FLASH – Perform FLASH related operations (i.e. firmware upgrades).
- DDS – Perform DDS related operations (i.e. DDS frequency operations).
- TRACK – Perform tracking related operations (i.e. temperature or light intensity compensation)
- CONFIG – Perform Configuration related operations, such as save and restore the current setup.
- CALIBRATION – Perform Temperature Calibration operations (i.e. create, save, restore, and assign calibration tables).
- TEMPERATURE – Perform temperature conversions (i.e. read the sensor inputs).
- ONEWIRE – Perform One Wire EEPROM operations (i.e. serial number and Daughter Board ID operations).
- MODULATION – Perform Modulation operations (i.e. Enable either Analog or Digital modulation).

- USB – Perform USB operations
- DAUGHTER – Perform Daughter Board operations (i.e. DAC, FSK, and Blank Control).
- CHIRP – Perform Chirp related operation (i.e. frequency chirps).
- BOARDID – Perform BOARDID related functions (i.e: Displays unique characteristic of the board such as serial number, firmware version, etc)

Each of the command categories is detailed in the sections that follow.

4. Definition of Syntax Elements

The command descriptions that follow include the following syntactical elements.

- Square Brackets "[ ]" indicate optional Arguments.
- Ellipsis "…" indicates repeated Arguments.
- Vertical bar "|" indicates logical OR

5. Syntax for Frequencies, Wavelengths, and Frequency Tuning Words (FTW)

Beginning with the firmware release of December 2008, the command syntax for specifying frequencies has been enhanced. Ultimately the DDS chip uses a Frequency Tuning Word (FTW) to specify the frequency. But FTWs are not the most convenient or common way of expressing frequencies. The firmware of the AOTF Controller converts the common syntax of frequencies (MHz and Hz) and wavelengths (nm) into FTWs for the DDS chips. All commands that expect frequency arguments now accept the following syntax:

- A number alone is the frequency in Mega Hertz (MHz).
  Example: "`dds frequency 0 123.456`" would set the frequency for channel 0 to 123.456MHz.
- An exclamation point (!) at the beginning of a number specifies that the frequency is in Hertz (Hz).
  Example: "`dds frequency 0 !123456000`" would set the frequency for channel 0 to 123.456MHz.
- An at sign (@) at the beginning of a number specifies the frequency in FTW. An FTW is the actual 32 bit value that is used by the DDS chip. This is the most precise way of expressing a frequency. FTW values range from 0 to 2147483647 (which is $2^{31} - 1$) and span the frequency range linearly from 0Hz to 200MHz.
  Example: "`dds frequency @1325598706`" would set the frequency for channel 0 to 123.456MHz.
- A pound sign (#) at the beginning of a number specifies the frequency in light wavelength. The wavelength is expressed in nano-meters (nm = 10^{-12} meters). The calibration tuning polynomial coefficients are used to calculate the frequency corresponding to the wavelength. The proper tuning polynomial coefficients must be entered into the calibration settings for this syntax to operate properly.
  Example: "`dds frequency #650`" would set the frequency for channel 0 to ~89.296MHz. The exact frequency is dependent on the crystal and the tuning polynomial coefficients.

6. Historical Side Note about FSK Nomenclature

When first designing and architecting the *AOTF Controllers* it was thought that the primary usage of the FSK capability would be for two capabilities:

- Output Blanking – The RF output can be turned ON/OFF.
- Frequency Shift Keying (i.e. Frequency Modulation) – Selecting between two frequencies.

With these two concepts in mind the control signals on the schematics and connectors were named FSK and BLANK. These names seemed appropriate at that time.

As engineers, developers, and customers began using the *AOTF Controllers* it became necessary to develop new nomenclature to describe and discuss the new ways that *AOTF Controllers* were being utilized. The DDS chip at the heart of the *AOTF Controllers* utilized two profile select pins. The hardware implementation of the FSK and BLANK capability was implemented via the profile select pins of the DDS chip.

What eventually evolved was the concept of "Profiles". A "Profile" is a collection of configuration settings that are grouped together, and selected by the profile select pins of the DDS chip. Using the Profile concept it's possible to implement not only the original concepts of BLANK and FSK, but much more flexibility.

The end result is nomenclature that might seem confusing to new users, but has historical roots that are difficult to eradicate.

7. Historical Side Note about Firmware Commands

As the family of *AOTF Controllers* grew from the original Octal Channel *AOTF Controller* to the Single Channel and Quad Channel *AOTF Controllers* the desire to maintain a common command syntax became mandatory for compatibility with various host environments. A common command syntax for the entire family of *AOTF Controllers* allows engineers and developers to leverage the investment in software and command scripts. As a result, there are some commands that seem to be redundant, and command arguments that appear to be unnecessary for a particular *AOTF Controller*. For example, specifying the channel number when using a Single Channel *AOTF Controller*. There is only one channel, but for compatibility with the multi channel *AOTF Controllers*, the channel number remains a part of the command syntax, even though the only valid channel number is channel 0.

Some of the areas where *AOTF Controllers* differ the most are:

- **Host Modulation Interface**
  The Host Modulation Interface for Octal and Quad *AOTF Controllers* is implemented as separate Daughter Cards. At the time of this writing there are Analog and Digital daughter cards.

  The Single Channel *AOTF Controller* does not have a daughter card architecture. The Host Modulation Interface is integrated into the main circuitry of the *AOTF Controller* and has the capability of doing either Analog or Digital modulation.

- **RF Gain**
  The granularity of control over the RF Gain has been an area of feature expansion in the *AOTF Controllers*. The Octal and Quad *AOTF Controllers* have RF gain control for each channel, but not for each profile.

  The original design of the Single Channel *AOTF Controller* (FX0) was identical to the Octal and Quad and had RF gain control for each channel, but not for each profile. The newer architecture for the Single Channel *AOTF Controller* (FX1) incorporated finer granularity and per profile control of the RF gain.

The result of these variations has resulted in slight (undesirable but unavoidable) differences in the command syntax, depending on the model of *AOTF Controller*. In order to keep the command syntax as similar as possible, for AOTF Controllers that lack the underlying hardware, some commands are implemented as No Operations (NOOPS). The command is accepted, but since the hardware does not support the operation, the command is quietly ignored.

8. Help Commands

The Help Commands provide a short description of each of the commands, or a detailed description of specific commands.

Syntax:

```
Help [command] [...]
? [command] [...]
```

Arguments:

[command] One of the valid *AOTF Controller* commands.

Examples:

The following is a list of descriptions of all valid *AOTF Controller* commands.

Help	Display command help
?	Display command help
I2c	Perform I2C Bus functions
EEProm	Perform EEPROM functions
Adc	Perform ADC functions
Flash	Perform FLASH functions
Dds	Perform DDS functions
Track	Perform Tracking functions
Config	Perform Configuration functions
Calibration	Perform Calibration functions
Temperature	Perform Temperature functions
OneWire	Perform One Wire EEPROM functions
Modulation	Perform Modulation functions
Usb	Perform USB functions
Daughter	Perform Daughter Board functions
Chirp	Perform Chirp functions
BoardId	Perform Board Identification functions
Remark	Remark Text

The following example will display a detailed description of the EEPROM and ADC commands:

```
help ee adc
```

`"EEProm [arguments]"` The EEProm command performs various EEPROM related functions.

`"Adc [arguments]"` The ADC command performs various ADC related functions.

9. I2C Commands

The I2C commands provide the foundation for performing I2C related operations. The I2C commands consist of the following operations:

Help	Displays a list of I2C commands and provides detailed syntax for specific commands.
Scan	Scan the I2C bus for devices. The I2C bus is scanned from device address 0x00 through 0xFE. The address of each responding device is displayed. Reference the Octal, Quad, and Single Channel documentation for additional details about the devices connected to the I2C Bus.
Read	Read from a device. The data is read and displayed. The `–s` argument causes the I2C command to terminate without the final STOP phase of the I2C transaction.
Write	Write to a device. The data is written to the device. The `–s` argument causes the I2C command to terminate without the final STOP phase of the I2C transaction.
Stop	Create a Stop Condition. The stop command is useful for generating a stop condition on the I2C bus at the end of a complex sequence of interconnected I2C transactions.
Poll	Poll a device. The poll command continuously attempts to read from a device address until a response is received.
Speed	Set the speed of the I2C bus. The I2C bus can operate at either 100KHz or 400KHz.

Syntax:

```
I2C Help [Command…]
I2C ? [Command…]
I2C Scan
I2C Read [-s] Device Count
I2C Write [-s] Device Data [Data] [...]
I2C Stop
I2C Poll Device
I2C Speed [100 | 400]
```

Arguments:

Command	Any of the I2C commands.
-s	If the –s option is present the I2C command will be terminated without the final STOP phase of the I2C transaction.
Device	The address of the device on the I2C bus. I2C device addresses are represented as a single byte in the range of 0x00 through 0xFE, and must be even. Reference the

	Octal, Quad, and Single Channel documentation for additional details about the devices connected to the I2C Bus.
Count	The number of bytes to read. Count must be between 0 and 64.
Data	The data to write. A maximum of 64 bytes can be written in a single command.

Examples:

The following example will scan the I2C bus and display the available devices:

```
i2c scan
Device located at address 0x52
Device located at address 0x54
Device located at address 0x58
Device located at address 0x5A
Device located at address 0x90
Device located at address 0xA2
Device located at address 0xE0
```

The following command will display the contents of the EEPROM device located at address 0xA2: The first command establishes the address within the EEPROM, and the second command reads the data.

```
i2c write 0xa2 0 0
i2c read 0xa2 10
0xB6 0x34 0x12 0x01 0x00 0x00 0x00 0x05 0x00 0x00
```

10. EEPROM Commands

The EEPROM commands provide the ability to manipulate the Boot Loader EEPROM, and the Calibration EEPROM. The EEPROM commands consist of the following operations:

Help	Displays a list of EEPROM commands and provides detailed syntax for specific commands.
Read	Read data from an EEPROM.
Write	Write data to an EEPROM.
Download	Download data from an EEPROM. The data is formatted into a Intel HEX format and displayed. See *Section 27, Intel Hex File Format, on page 70*, for a description of the Intel Hex File format.
Upload	Upload data to an EEPROM. The data is expected to be in Intel HEX format. See *Section 27, Intel Hex File Format, on page 70*, for a description of the Intel Hex File format.

Syntax:

```
EEProm Help [Command…]
EEProm ? [Command…]
EEProm Read [-Device] Address Count
EEProm Write [-Device] Address Data [Data] [...]
EEProm Download [-Device] Size
EEProm Upload [-Device]
```

Arguments:

Command	Any of the EEPROM commands.
Device	The address of the device on the I2C bus. I2C device addresses are represented as a single byte in the range of 0x00 through 0xFE, and must be even. Reference the Octal, Quad, and Single Channel documentation for additional details about the devices connected to the I2C Bus.
Address	The address within the EEPROM. I2C EEPROM devices use either 1 or 2 byte addresses. The address must be either 1 or 2 bytes, depending on the type of I2C EEPROM devices being accessed.
Count	The number of bytes to read. Count must be between 0 and 64.
Data	The data to write. A maximum of 64 bytes can be written in a single command.
Size	The number of bytes to download.

Examples:

The following command reads 32 bytes from location 0x100 of the boot loader EEPROM:

```
ee read 0x100 0x20
```



```
0x35 0x08 0xF5 0x83 0xE0 0xFE 0xA3 0xE0 0xFF 0xAC 0x06 0x8C 0x0E
0xF5 0x0F 0xE5
0x12 0x70 0x3D 0x8E 0x10 0x8F 0x11 0x85 0x09 0x82 0x85 0x08 0x83
0xE0 0xFF 0x64
```

The following command will download the first 20 bytes of the calibration EEPROM:

```
ee down -0xae 20
:10000000AA9903FFFFFFFFFFFFFFFFFFFFFFFFFFB7
:04001000FFFFFFFFF0
:00000001FF
```

11. ADC Commands

The ADC commands provide the ability to read the Analog to Digital (AtoD) sensor inputs. The ADC commands consist of the following operations:

Help	Displays a list of ADC commands and provides detailed syntax for specific commands.
Read	Read data from an AtoD channel. The AtoD channel is read and the result is displayed. The `-f` argument causes the command to read and display the AtoD conversion continuously, until the user presses a key.
Gain	Set the gain for an AtoD channel. Without any gain argument, the current setting of the AtoD channel's gain is displayed. The gain is a number between 0 and 255, which represents the setting of the Xicor digitally controlled potentiometers. A setting of 128 is unity gain.
Peak	Adjust the gain for an AtoD channel so that the current reading is 85% of the maximum. The gain is a number between 0 and 255, which represents the setting of the Xicor digitally controlled potentiometers. A setting of 128 is unity gain. The gain is automatically adjusted so that the reading from the ADC channel is 85% of the maximum. This leaves some margin for readings that are slightly greater than the current setting.
Average	Set the number of samples for averaging of an ADC channel. The argument will determine how many samples are taken from the ADC channel to be established into an average. Count is the number of samples to be read from the source before all the readings are averaged. The average implements a low pass filter that attempts to smooth out the readings.

Syntax:

```
Adc Help [Command…]
Adc ? [Command…]
Adc Read [-f] Channel*
Adc Gain Channel* [Gain]
Adc Peak Channel*
Adc Average Channel* [Count]
```

Arguments:

Command	Any of the ADC commands.
-f	The FOREVER argument causes the command to be executed until the user presses a key.
Channel*	The AtoD channel to access. The AtoD channels are numbered 0 through 7. An asterisk ('*') can be used to select all channels.
Gain	The gain for the AtoD channel. The gain is a number between 0 and 255.
`Count`	The number of samples to be taken for an average.

Examples:

The following command will display the AtoD sensor input from channel 2:

```
Adc read 2
18
```

The following command will display the AtoD sensor input from all channels:

```
adc r *
3689 1704 14 0 0 0 1854 1838
```

The following command will display the gain for all channels:

```
adc gain *
127 183 131 0 0 0 183 183
```

The following command will set the peak gain for all channels:

```
adc peak *
ADC Channel 0 Gain set to 127
ADC Channel 1 Gain set to 183
ADC Channel 2 Gain set to 215
ADC Channel 3 Gain set to 255
ADC Channel 4 Gain set to 255
ADC Channel 5 Gain set to 255
ADC Channel 6 Gain set to 183
ADC Channel 7 Gain set to 183
```

The following command will set the number of samples taken to obtain an average for all channels to 15.

```
adc average * 15
adc average *
15 15 15 15 15 15 15 15
```


12. FLASH Commands

The FLASH commands provide the ability to manipulate the FLASH memory that contains the firmware of the microcontroller. The FLASH commands consist of the following operations:

Help	Displays a list of FLASH commands and provides detailed syntax for specific commands.
Read	Read data from FLASH.
Write	Write data to FLASH.
Download	Download data from FLASH. The data is formatted into Intel HEX format and displayed. See *Section 27, Intel Hex File Format, on page 70*, for a description of the Intel Hex File format.
Upload	Upload data to FLASH. The FLASH upload is a two phase process. The first phase is a test of the communication channel, and the second phase is the actual upload of the data into the FLASH. The test insures that the flow control signals (RTS and CTS) are functioning properly between the host and the *AOTF Controller*. After issuing the Upload command, the new AOTF.HEX file should be sent to the *AOTF Controller* twice, once for each phase of the command. The `-r` argument reboots the *AOTF controller* after the upload has been completed. The `-i` argument initializes the configuration table to the default state. The data is expected to be in Intel HEX format. See *Section 27, Intel Hex File Format, on page 70*, for a description of the Intel Hex File format.

Syntax:

```
Flash Help [Command…]
Flash ? [Command…]
Flash Read Bank Address Count
Flash Write Bank Address Data [Data] [...]
Flash Download Bank Size
Flash Upload [-ri]
```

Arguments:

Command	Any of the FLASH commands.
Bank	The bank number within the FLASH. The FLASH is divided into 16 banks of 32K bytes each. Bank must be a number between 0 and 15.
Address	The address within the FLASH. The FLASH is divided into 16 banks of 32K bytes each. Address must be a number between 0x0000 and 0x7FFF.
Count	The number of bytes to read. Count must be between 0 and 64.
Data	The data to write. A maximum of 64 bytes can be written in a single command.

Size	The number of bytes to download.
-r	Causes the *AOTF Controller* to reboot after the upload completes.
-i	Initializes the configuration table to the default state.

Examples:

The following command reads 5 bytes from location 0x0100 of bank 3 of the FLASH:

```
flash read 3 0x0100 5
0xA3 0xEE 0xF0 0xA3 0xEF
```

The following command uploads a new firmware image into the FLASH:

```
f u -t
Performing actual upload.
Upload the data now:
```

<At this point the AOTF.HEX file is sent to the *AOTF Controller*>

```
Uploading to bank 0
Uploading to bank 1
Uploading to bank 2
Uploading to bank 3
Uploading to bank 4
Uploading to bank 5
Uploading to bank 6
Uploading to bank 7
Uploading to bank 8
Uploading to bank 9
Uploading to bank 10
Uploading to bank 11
Uploading to bank 12
Uploading to bank 13
Uploading to bank 14
Uploading to bank 15
Flash upload was successful.
```


13. DDS Commands

The DDS commands provide the ability to manipulate the DDS chips that generate the frequency and RF output of the *AOTF Controller*. The DDS commands consist of the following operations:

Help — Displays a list of DDS commands and provides detailed syntax for specific commands.

Reset — Reset all DDS chips to their default state. The default state of each DDS chip is:

- The frequency is set to 0.
- The amplitude scale factor is set to 0.
- The phase is set to 0 degrees.
- Temperature Tracking is disabled.
- Each Profile is allocated a single slot in the internal SRAM (Chirp disabled).
- FSK mode is disabled (single tone frequency mode).

Frequency — Set the frequencies of a DDS channel to Freq0, Freq1, Freq2, and Freq3. Freq0 is output when the DDS is operating in single tone mode. When operating in FSK mode the frequency output from the DDS is Freq0, Freq1, Freq2, or Freq3 depending on the FSK and BLANK control signals.

The syntax where the frequency for multiple profiles is specified by a list of frequencies on the command line has been deprecated (i.e. “dds Frequency Freq0 Freq1 Freq2 Freq3”). It is still supported, but will soon be removed. The preferred syntax for specifying the frequency of a particular profile is to use the –p argument.

See *Section 5,Syntax for Frequencies, Wavelengths, and Frequency Tuning Words (FTW), on page 11*, for details about the syntax for frequencies.

The `-i` argument causes the command to enter an interactive adjustment mode, where the following interactive commands are available:

+ Increment the frequency by the delta.

- Decrement the frequency by the delta.

= Set the Frequency. After entering the = command, enter decimal digits followed by a carriage return to change the frequency.

Change the Frequency delta. The delta is the adjustment to the frequency that is applied whenever the + or – commands are issued. The initial delta is 1 KHz. After entering the # command, enter decimal digits followed by a carriage return to change the delta.

@ Displays the selected channel and current frequency.

? Display a list of adjustment commands.

. (period) Exit the interactive adjustment mode.

The `-p` argument specifies the profile. Each DDS channel has 4 profiles, numbered 0 to 3. When the DDS is operating in single tone mode, only profile 0 is utilized. When operating in FSK mode the profile selection is controlled with the FSK and BLANK signals. The -p argument allows the user to individually set or display the frequency of each profile. If not specified, the profile defaults to 0. Using an (‘*’) for the profile specifies all 4 profiles.

Wavelength

This function has been deprecated. Beginning with the firmware release of December 2008 the functionality of the Wavelength command has been incorporated into the syntax for specifying a frequency. The Wavelength command is still supported, but will eventually be removed.

Set the wavelengths of a DDS channel to Wave0, Wave1, Wave2, and Wave3. This function is identical to the Frequency command except the frequency is specified by using the wavelength of the light to be diffracted.

The syntax where the wavelength for multiple profiles is specified by a list of wavelengths on the command line has been deprecated (i.e. “dds Wave Wave0 Wave1 Wave2 Wave3”). It is still supported, but will soon be removed. The preferred syntax for specifying the wavelength of a particular profile is to use the –p argument.

See *Section 5,Syntax for Frequencies, Wavelengths, and Frequency Tuning Words (FTW), on page 11*, for details about the syntax for wavelengths.

The `-i` argument causes the command to enter an interactive adjustment mode, where the following interactive commands are available:

+ Increment the frequency by the delta.

- Decrement the frequency by the delta.

= Set the Frequency. After entering the = command, enter decimal digits followed by a carriage return to change the frequency.

Change the Frequency delta. The delta is the adjustment to the frequency that is applied whenever the + or – commands are issued. The initial delta is 1 KHz. After entering the # command, enter decimal digits followed by a carriage return to change the delta.

@ Displays the selected channel and current frequency.

? Display a list of adjustment commands.

. (period) Exit the interactive adjustment mode.

The `-p` argument specifies the profile. Each DDS channel has 4 profiles, numbered 0 to 3. When the DDS is operating in single tone mode, only profile 0 is utilized. When operating in FSK mode the profile selection is controlled with the FSK and BLANK signals. The -p argument allows the user to individually set or display the frequency of each profile. If not specified, the profile defaults to 0. Using an ‘(‘*’) for the profile specifies all 4 profiles.

Track

The Track command serves the same purpose as the Frequency command. Both commands set or retrieve the frequency. They differ with respect to how they interact with the temperature and light intensity tracking capability:

- The Frequency command always reports the current frequency, including any adjustments that may have been made by the temperature and light intensity tracking adjustments.
- The Track command always reports the frequency that was originally used to set the frequency, regardless of what adjustments may have been made to the frequency by the temperature and light intensity tracking adjustments.

The Syntax for the Track command is identical to the Frequency command.

Fsk

Enables FSK mode, also known as Multi Profile Mode, which is the ability to rapidly select profiles by utilizing the BLANK and FSK pins of the front panel connector.

See *Section 6, Historical Side Note about FSK Nomenclature, on page 12* for details about the nomenclature.

The subject of FSK mode, Single Profile Mode, and Multi Profile mode is covered in details in the document titled "*AOTF Controllers and FSK Operation*", available from Crystal Technology.

The mode parameter selects the Virtual Profiles Mode as shown in *Table 2*. The FSK and BLANK pins signals on the front panel connector can be used to control the selected profile, which controls the frequency of the RF output.

Virtual Profiles Mode				
Virtual Profiles Mode	**BLANK (Input from Host)**	**FSK (Input from Host)**	**DDS Profile**	**RF Output Frequency**
0 (Single Profile Mode)	Not Used	Not Used	Profile 0	Frequency 0
	Not Used	Not Used	Profile 0	Frequency 0
	Not Used	Not Used	Profile 0	Frequency 0
	Not Used	Not Used	Profile 0	Frequency 0
1 (Invert BLANK)	0	0	Profile 2	Frequency 2
	0	1	Profile 3	Frequency 3
	1	0	Profile 0	Frequency 0
	1	0	Profile 1	Frequency 1
3 (Identity Mapping)	0	0	Profile 0	Frequency 0
	0	1	Profile 1	Frequency 1
	1	0	Profile 2	Frequency 2
	1	1	Profile 3	Frequency 3
5 (Inverse Mapping)	0	0	Profile 3	Frequency 3
	0	1	Profile 2	Frequency 2
	1	0	Profile 1	Frequency 1
	1	1	Profile 0	Frequency 0
7 (Invert FSK)	0	0	Profile 1	Frequency 1
	0	1	Profile 0	Frequency 0
	1	0	Profile 3	Frequency 3
	1	1	Profile 2	Frequency 2

Table 2: Virtual Profiles Mode

Ftw

This function has been deprecated. Beginning with the firmware release of December 2008 the functionality of the Ftw command has been incorporated into the syntax for specifying a frequency. The Ftw command is still supported, but will eventually be removed.

Adjust the Frequency Tuning Word (FTW) for a DDS channel. With an argument, the Ftw command sets the DDS output frequency. The Ftw argument represents a frequency between 0 and 200Mhz. The power on reset value is 0.

The syntax where the FTWs for multiple profiles is specified by a list of FTWs on the command line has been deprecated (i.e. “dds Ftw Ftw0 Ftw1 Ftw2 Ftw3”). It is still supported, but will soon be removed. The preferred syntax for specifying the FTW of a particular profile is to use the –p argument.

See *Section 5,Syntax for Frequencies, Wavelengths, and Frequency Tuning Words (FTW), on page 11*, for details about the syntax for wavelengths.

Without an argument, the Ftw command displays the current Ftw.

The `-i` argument causes the command to enter an interactive adjustment mode, where the following interactive commands are available:

+	Increment the FTW by delta amount.
-	Decrement the FTW by delta amount.
=	Set the FTW. After entering the = command, enter decimal digits followed by a carriage return to change the FTW.
#	Change the FTW delta. The delta is the adjustment to the FTW that is applied whenever the + or – commands are issued. The initial delta is 1. After entering the # command, enter decimal digits followed by a carriage return to change the delta.
@	Display the selected channel and current FTW.
?	Display a list of the adjustment commands.
.	(period) Exit the interactive adjustment mode.

The `-p` argument specifies the profile. Each DDS channel has 4 profiles, numbered 0 to 3. When the DDS is operating in single tone mode, only profile 0 is utilized. When operating in FSK mode the profile selection is controlled with the FSK and BLANK signals. The -p argument allows the user to individually set or display the FTW of each profile. If not specified, the profile defaults to 0. Using an asterisk (‘*’) for the profile specifies all 4 profiles.

Peak	Scan the spectrum searching for peak light intensity. The Peak command is a multi step algorithm designed to optimize the diffracted light intensity. See *Section 24,* Peak Light Intensity, on *page 65*, for a more detailed description of the peak detection algorithm. The `-f` argument causes the command to loop until a key is pressed. The `Start` argument specifies the lowest frequency for the spectrum range. `Start` is specified in megahertz. The `End` argument specifies the highest frequency for the spectrum range. `End` is specified in megahertz. The `Inc` argument specifies the maximum delta frequency for the spectrum scan. `Inc` is specified in Hertz and must be a value from 1Hz to 1 MHz.
Sweep	Sweep through the spectrum. The Sweep command sweeps the DDS channel frequency through the spectrum. The –f argument causes the command to sweep through the spectrum continuously, until the user presses a key. The `Start` argument specifies the lowest frequency for the spectrum range. `Start` is specified in megahertz. The `End` argument specifies the highest frequency for the spectrum range. `End` is specified in megahertz. The `Inc` argument specifies the maximum delta frequency for the spectrum scan. `Inc` is specified in Hertz and must be a value from 1 Hz and 15 kHz.
Amplitude	Set the Amplitude Scale Factor (ASF) for a DDS chip. The ASF is the maximum modulation amplitude that the DDS will output. The ASF ranges from 0 to 16383, with 0 being no output, and 16383 being the maximum. The power on reset ASF is 0. With an argument, the Amplitude command sets the DDS output maximum amplitude. See *Section 24,* Peak Light Intensity, on *page 65*, for a more information about using the ASF. Without an argument, the Amplitude command displays the current Amplitude Scale Factor. The `-i` argument causes the command to enter an interactive adjustment mode, where the following interactive commands are available: ? Display a list of the adjustment commands. @ Display the selected channel and current Amplitude. # Change the Amplitude delta. The delta is the adjustment to the Amplitude that is applied whenever the + or – commands are issued. The initial delta is

1. After entering the # command, enter decimal digits followed by a carriage return to change the delta.

+ Increment the Amplitude by delta amount.

- Decrement the Amplitude by delta amount.

= Set the Amplitude. After entering the = command, enter decimal digits followed by a carriage return to change the Amplitude.

. (period) Exit the interactive adjustment mode.

AmpPeak Sweep the Amplitude Scale Factor (ASF) range from 0 to 16383 and locate the ASF setting that corresponds to the peak light intensity. See *Section 24,* Peak Light Intensity*,* on *page 65*, for a more information about the AmpPeak command.

Gain Adjust the Gain of the RF Output for a specific DDS channel. The Gain Circuitry of the *AOTF Controllers* has had different implementations for the various *AOTF Controllers* as shown in *Table 3*.

AOTF Controller Gain Capabilities		
Model	**Per Profile Gain Control**	**Range**
Octal (Standard Power)	NO	0 to 31
Octal (High Power)	NO	0 to 31
Quad	NO	0 to 31
Single FX0	NO	0 to 31
Single FX1	YES	0 to 255

Table 3: AOTF Controller Gain Capabilities

With an argument, the Gain command sets the Gain. The Gain is a number between 0 and Maximum Gain. The Maximum Gain is AOTF Controller model dependant, *See Table 3, AOTF Controller Gain Capabilities, on page 29.*

Without an argument, the Gain command displays the current Gain.

The `-p` argument specifies the profile. Each DDS channel has 4 profiles, numbered 0 to 3. When the DDS is operating in single tone mode, only profile 0 is utilized. When operating in FSK mode the profile selection is controlled with the FSK and BLANK signals. The -p argument allows the user to individually set or display the Gain of each profile. If not specified, the profile defaults to 0. Using an asterisk (‘*’) for the profile specifies all 4 profiles.

The `-i` argument causes the command to enter an interactive adjustment mode, where the following interactive commands are available:

+	Increment the Gain by delta amount.
-	Decrement the Gain by delta amount.
=	Set the Gain. After entering the = command, enter decimal digits followed by a carriage return to change the Gain setting.
#	Change the Gain delta. The delta is the adjustment to the Gain that is applied whenever the + or – commands are issued. The initial delta is 1. After entering the # command, enter decimal digits followed by a carriage return to change the delta.
@	Display the selected channel and current Gain setting.
?	Display a list of the adjustment commands.
.	(period) Exit the interactive adjustment mode.

Phase

Set the phase of a channel ranging from 0 to 16383, 0 representing 0 phase and 16383 representing 360º.

The `-i` argument causes the command to enter an interactive adjustment mode, where the following interactive commands are available:

+	Increment the Phase by delta amount.
-	Decrement the Phase by delta amount.
=	Set the Phase. After entering the = command, enter decimal digits followed by a carriage return to change the Phase setting.
#	Change the Phase delta. The delta is the adjustment to the Gain that is applied whenever the + or – commands are issued. The initial delta is 1. After entering the # command, enter decimal digits followed by a carriage return to change the delta.
@	Display the selected channel and current Phase setting.
?	Display a list of the adjustment commands.
.	(period) Exit the interactive adjustment mode.

Syntax:

```
Dds Help [Command…]
Dds ? [Command…]
Dds Reset
Dds Frequency[-ip profile*] Channel* [Freq]
Dds Track[-ip profile*] Channel* [Freq]
Dds Fsk Channel* [Fsk]
Dds Peak [-f] Channel [Start [End [Inc]]]
Dds Sweep [-f] Channel [Start [End [Inc]]]
```



```
Dds Amplitude [-i] Channel* [Asf]
Dds AmpPeak Channel
Dds Gain [-ip profile*] Channel* [Gain]
Dds Phase [-i] Channel* [Phase]
```

NOTE: The following commands have been deprecated:

```
Dds Frequency[-ip profile*] Channel* [Freq0][Freq1][Freq2][Freq3]
Dds Wavelength[-p profile*] Channel* [Wave]
Dds Wavelength[-p profile*] Channel* [Wave0][Wave1][Wave2][Wave3]
Dds Ftw [-ip profile*] Channel* [Ftw]
```

Arguments:

Command	Any of the DDS commands.
-i	The INTERACTIVE argument causes the command to enter into the interactive adjustment mode.
-p	The PROFILE argument is used to select the desired profile.
Profile*	The profile to access. Profiles are numbered 0 through 3. An asterisk (‘*’) can be used to select all profiles.
Freq	The frequency is a floating point number between 0MHz and 200MHz. See *Section 5, Syntax for Frequencies, Wavelengths, and Frequency Tuning Words (FTW), on page 11*, for detailed information on the syntax used for specifying frequencies.
Freq0 Freq1 Freq2 Freq3	These arguments, and the command syntax that uses them are being deprecated. These arguments and the associated command syntax should not be used. They are currently still supported, but will eventually be removed. The preferred syntax is to use the –p argument to specify the profile.
Wave	The wavelength is a floating point number between ~1950nm (way into the infrared) and 290nm (way into the ultraviolet). See *Section 5, Syntax for Frequencies, Wavelengths, and Frequency Tuning Words (FTW), on page 11*, for detailed information on the syntax used for specifying wavelengths.
Wave0 Wave1 Wave2 Wave3	These arguments, and the command syntax that uses them are being deprecated. These arguments and the associated command syntax should not be used. They are currently still supported, but will eventually be removed. The preferred syntax is to use the –p argument to specify the profile.

Channel*	The DDS channel to access. The DDS channels are numbered 0 through 7. An asterisk (‘*’) can be used to select all channels.
Ftw	The Frequency Tuning Word (FTW) is an integer between 0 and 2147483647 (2^{31}-1). See *Section 5, Syntax for Frequencies, Wavelengths, and Frequency Tuning Words (FTW), on page 11*, for detailed information on the syntax used for specifying Frequency Tuning Words.
Fsk	The FSK mode.
Asf	The Amplitude Scale Factor (ASF).
Start	The lowest frequency for the spectrum range, a floating point number between 0.0MHz and 200.0MHz. The value represents the frequency in megahertz.
End	The highest frequency for the spectrum range, a floating point number between 0.0MHz and 200.0MHz. The value represents the frequency in megahertz.
Inc	The delta frequency for the spectrum range, a floating point number between 1 and 1000000. The value represents the frequency in hertz.
-f	The FOREVER argument causes the command to be executed until the user presses a key.
Gain	The Gain value. The gain can be a value from 0 to the Maximum Gain, which is AOTF Controller model dependant, *See Table 3, AOTF Controller Gain Capabilities, on page 29.*
Phase	The phase offset for the given channel. The phase offset can be a value from 0 (no offset) to 16383 (360º offset).

Examples:

The following command will set the frequency of profile 0 of channel 5 to 50Mhz:

```
Dds freq 5 50
```

The following command will set the FSK mode for all channels to mode 5:

```
Dds fsk * 5
```

The following command will allow the DDS frequency command to enter interactive mode for channel 0. Then “?” is used to display the interactive commands available. Then “#1000000” is used to set the delta as 1Mhz. Then “++++++++” is used to increment the frequency by 8 Mhz. “.” Is used to exit the interactive mode.

```
Dds f -i 0
Channel 0 profile 0 frequency 8.000000e+07Hz(Ftw 858993472)
?+      Increment
-       Decrement
=       Set
```



```
#       Delta
@       Display
?       Help
.       Exit
=1000000
++++++++@Channel 0 profile 0 frequency 8.800000e+07Hz (Ftw
944892816)
.Channel 0 profile 0 frequency 8.800000e+07Hz (Ftw 944892816)
*
```

The following command will locate the peak light intensity between 88Mhz and 95Mhz, using channel 3:

Command	Frequency	Tuning Word
Dds peak 3 88 95		
DdsPeak (within 7000Hz):	8.9211e+07	957895653
DdsPeak (within 3500Hz):	8.92214e+07	958007892
DdsPeak (within 1750Hz):	8.92127e+07	957913722
DdsPeak (within 875Hz):	8.92197e+07	957988788
DdsPeak (within 437.504Hz):	8.92017e+07	957795859
DdsPeak (within 218.752Hz):	8.91992e+07	957769612
DdsPeak (within 109.376Hz):	8.92056e+07	957837492
DdsPeak (within 54.688Hz):	8.92e+07	957778021
DdsPeak (within 27.344Hz):	8.92108e+07	957893989
DdsPeak (within 13.672Hz):	8.92113e+07	957898946
DdsPeak (within 6.832Hz):	8.92106e+07	957891262
DdsPeak (within 3.416Hz):	8.92121e+07	957907708
DdsPeak (within 1.712Hz	8.92116e+07	957902762
DdsPeak (within 1Hz):	8.92118e+07	957905022
DdsPeak (within 1Hz):	8.92117e+07	957902952
DdsPeak (within 1Hz):	8.92117e+07	957902924
DdsPeak (within 1Hz):	8.92116e+07	957902786
DdsPeak (within 1Hz):	8.92117e+07	957902898
DdsPeak (within 1Hz):	8.92116e+07	957902836
DdsPeak (within 1Hz):	8.92116e+07	957902736
DdsPeak (within 1Hz):	8.92116e+07	957902676
DdsPeak (within 1Hz):	8.92116e+07	957902656

The following command will set the Amplitude Scale Factor for all channels to 16383:

```
Dds a * 16383
```


14. Track Commands

The Track commands provide the ability to utilize the Temperature Compensation and Light Intensity Tracking algorithms. Reference these companion documents for more detailed information about Temperature Compensation and Light Intensity Tracking:

- AOTF Controllers and Temperature Compensation, Revision 1.1, www.CrystalTechnology.com, Crystal Technology, Inc. 1040 East Meadow Circle, Palo Alto, CA 94303-4230.

- AOTF Controllers and Light Intensity Trackng, Revision 1.1, www.CrystalTechnology.com, Crystal Technology, Inc. 1040 East Meadow Circle, Palo Alto, CA 94303-4230.

The Track commands consist of the following operations:

Help	Displays a list of Track commands and provides detailed syntax for specific commands.
Stop	Stops automatic tracking on the specified channel.
Reference	Sets the Reference Frequency and Reference Temperature that are used when temperature compensation is enabled (either auto or manual tracking). The `-p` argument specifies the profile. Each DDS channel has 4 profiles, numbered 0 to 3. When the DDS is operating in single tone mode, only profile 0 is utilized. When operating in FSK mode the profile selection is controlled with the FSK and BLANK signals. The -p argument allows the user to individually set or display the Reference Frequency and Reference Temperature of each profile. If not specified, the profile defaults to 0. Using an asterisk ('*') for the profile specifies all 4 profiles.
Frequency	Sets the Reference Frequency that is used when temperature compensation is enabled (either auto or manual tracking). The `-p` argument specifies the profile. Each DDS channel has 4 profiles, numbered 0 to 3. When the DDS is operating in single tone mode, only profile 0 is utilized. When operating in FSK mode the profile selection is controlled with the FSK and BLANK signals. The -p argument allows the user to individually set or display the Reference Frequency of each profile. If not specified, the profile defaults to 0. Using an asterisk ('*') for the profile specifies all 4 profiles.
Temperature	Sets the Reference Temperature that is used when temperature compensation is enabled (either auto or manual tracking). The `-p` argument specifies the profile. Each DDS channel has 4 profiles, numbered 0 to 3. When the DDS is operating in single tone mode, only profile 0 is utilized. When operating in FSK mode the profile selection is controlled with the FSK and BLANK signals. The -p argument allows the user to individually set or display the Reference Temperature of each profile. If not specified, the profile defaults to 0. Using an asterisk ('*') for the profile specifies all 4 profiles.

Auto — Enables auto tracking mode for the specified channel. Auto tracking uses an elapsed time period to check for necessary adjustments. If not specified the default period is 5 seconds. Auto tracking can utilize either Temperature Compensation or Light Intensity. For Temperature Compensation the Temperature Delta defaults to 0ºC if not specified, which results in no minimum Temperature Delta. In this case any change in temperature will cause an adjustment to be made.

The `-i` argument is used to specify Light Intensity Tracking, which is the default.

The `-t` argument is used to specify Temperature Compensation Tracking.

The `-v` argument makes the command display additional output information. The default is silence.

The `-a` argument forces the frequency compensation adjustments to be performed only on the currently active profile. When making a frequency adjustment the microcontroller must change profiles to program the new frequency. In some environments this causes an undesired change in the RF output. The –a argument causes the microcontroller to discard the update unless the currently active profile matches the profile of the update, thus preventing an undesired change in the profile selection.

Manual — Enables manual tracking mode for the specified channel. Manual tracking can utilize either Temperature Compensation or Light Intensity. For Temperature Compensation the Temperature Delta defaults to 0ºC if not specified, which results in no minimum Temperature Delta. In this case any change in temperature will cause an adjustment to be made. Manual Tracking works in conjunction with the Adjust command.

The `-i` argument is used to specify Light Intensity Tracking, which is the default.

The `-t` argument is used to specify Temperature Compensation Tracking.

The `-v` argument makes the command display additional output information. The default is silence.

Adjust — The Adjust command is used in conjunction with the Manual command to make Manual adjustments to the frequency.

The –f argument causes the command to make adjustments continuously, until the user presses a key.

The `-o` argument overrides the delta temperature threshold and makes an adjustment regardless of the delta temperature.

Syntax:

```
Track Stop Channel*

Track Reference [-p profile*] Channel* [RefFrequency*
RefTemperature*]

Track Frequency [-p profile*] Channel* [RefFrequency*]
```



```
Track temperature [-p profile*] Channel* [RefTemperature*]
Track Auto [-itv] Channel* [TemperatureDelta][Period]
Track Manual [-itv] Channel* [TemperatueDelta]
Track Adjust [-fo]
```

Arguments:

Channel*	The DDS channel to access. The DDS channels are numbered 0 through 7. An asterisk ('*') can be used to select all channels.
-p	The PROFILE argument is used to select the profile.
Profile*	The profile to access. The profiles are numbered 0 through 3. An asterisk ('*') can be used to select all profiles.
RefFrequency	Reference frequency in MHz.
RefTemperature	Reference Temperature in ºC.
TemperatureDelta	Temperature threshold used for temperature compensation. If the difference between the previous temperature compensation adjustment and the current temperature of the AOTF Crystal is below the TemperatureDelta then no adjustment will be performed.
Period	The elapsed time in seconds between Auto Tracking adjustments.
-i	The INTENSITY argument uses the photo intensity sensor to determine the peak frequency. This is the default.
-t	The TEMPERATURE argument uses temperature to estimate the peak frequency.
-v	The VERBOSE argument makes the command display additional output information. The default is silence.
-f	The FOREVER argument causes the command to execute until a character is pressed.
-o	The OVERRIDE argument overrides the delta temperature threshold and makes an adjustment regardless of the delta temperature.

Examples:

The following command will set the Reference Frequency to 80.303MHz and Reference Temperature to 27.5ºC for channel 0 profile 0:

```
Track reference –p 0 0 80.303 27.5
```

The following command will automatically perform a compensation check every 20 seconds and compensate if temperature variation is greater than 2.0º for channel 0:

```
Track Auto –tva 0 2.0 20
```


15. Config Commands

The Config commands provide the ability to change, save, and restore the current configuration of the *AOTF Controller*. The Config commands consist of the following operations:

Help	Displays a list of Config commands and provides detailed syntax for specific commands.
Name	Configure *the AOTF Controller's* name. The name can be used to distinguish multiple *AOTF Controllers* apart from each other. In environments where multiple *AOTF Controllers* are being used, the name can be used to identify the different *AOTF Controllers*.
User	Configure the *AOTF Controller's* user name. The user name is used for security when accessing the *AOTF Controller* with networking protocols.
Password	Configure the *AOTF Controller's* password. The password is used for security when accessing the *AOTF Controller* with networking protocols.
Pid	Configure the *AOTF Controller's* PID. The PID is used by the USB interface to identify the *AOTF Controller* to the OS. The PID can be toggled between the old legacy PID and the new contemporary PID. The PID is used by the OS to configure USB devices. The OS uses the PID to locate and associate a device driver with the attached hardware.
IP	Configure the *AOTF Controller's* IP Address. The IP Address is used when accessing the *AOTF Controller* with networking protocols. The IP Address is represented in standard dotted decimal IP address notation. The default IP Address is 192.168.10.1.
Save	Saves the current configuration to FLASH. The -a argument allows the configuration to automatically be restored on power up. The -i argument initializes a saved configuration to the default state. The -r argument reboots the *AOTF Controller's* firmware after saving the configuration.
Restore	Restore the configuration of the *AOTF Controller* from FLASH.
Reboot	Reboots the *AOTF Controller*.

Syntax:

```
Config Help [Command…]
Config ? [Command…]
Config Name
Config User
Config Password
Config [old | new]
Config IP
Config Save [-air]
```



```
Config Restore
Config Reboot
```

Arguments:

Command	Any of the Config commands.
-a	The AUTO argument causes the configuration to be restored each time the *AOTF Controller* powers up or resets.
-i	The INITIALIZE argument initialize the saved configuration to a default state.
-r	The REBOOT argument reboots the *AOTF Controller* after the configuration has been saved.
Name	The name can be any sequence of alpha-numeric letters, no white spaces (space, tab, or carriage return).
User	The user can be any sequence of alpha-numeric letters, no white spaces (space, tab, or carriage return).
Password	The password can be any sequence of alpha-numeric letters, no white spaces (space, tab, or carriage return).
IpAddress	The IP Address is represented in standard dotted decimal IP address notation. The default IP Address is 192.168.10.1.

Examples:

The following command will set the name to “Kosh”:

```
Config name kosh
```

The following command will set the IP Address to 192.168.77.129:

```
Config ip 192.168.77.129
```

The following command will set the PID to the new PID:

```
Config pid new
```


16. Calibration Commands

The Calibration commands provide the ability to create, save, restore, and assign calibration tables to DDS channels. The calibration data is maintained in the EEPROM on the Temperature Sensor Board (TSB). The TSB is a small circuit board that is mounted in the enclosure that houses the AOTF Crystal. The Calibration commands consist of the following operations:

Help	Displays a list of Calibration commands and provides detailed syntax for specific commands.
Identify	The Identify command can be used to displays and modify the identification information for the AOTF Crystal. During manufacturing and test the Identify command can be used to initialize the identification information.
Save	Save the calibration information in the EEPROM on the TSB.
Restore	Restore the calibration information from the EEPROM on the TSB.
Polynomial	Set the calibration coefficients used for the temperature compensation algorithm. There are 5 coefficients for the temperature compensation algorithm, numbered 0 to 4. With an argument the coefficient is set to the value of the argument. Without an argument the command displays the coefficient. The Index argument selects the particular coefficient. An asterisk (‘*’) specifies all of the coefficients.
Tuning	Set the calibration coefficients used for the wavelength to frequency conversion algorithm. There are 5 coefficients for the wavelength to frequency conversion algorithm, numbered 0 to 4. With an argument the coefficient is set to the value of the argument. Without an argument the command displays the coefficient. The Index argument selects the particular coefficient. An asterisk (‘*’) specifies all of the coefficients.
Tune	Calculate the tuned frequency for a wavelength. The Tune command uses the wavelength to frequency coefficients to estimate the frequency required to provide optimum diffracted light intensity for the specified wavelength.
State	The State command is used by OEMs to hold additional information about the TSB. The value can range from 0 to 255.
Test1	Manufacturing and production testing. This command is only useful in the production environment at Crystal Technology.
Test2	Manufacturing and production testing. This command is only useful in the production environment at Crystal Technology.
Test3	Manufacturing and production testing. This command is only useful in the production environment at Crystal Technology.

Syntax:

```
Calibration Help [Command…]
Calibration ? [Command…]
Calibration Identify [Part# Serial# CellID Date]
Calibration Save
Calibration Restore
Calibration Polynomial Index* [Coefficient]
Calibration Tuning Index* [Coefficient]
Calibration Tune Wavelength
Calibration State [Value]
Calibration Test1
Calibration Test2
Calibration Test3
```

Arguments:

Index*	The coefficient to access. Coefficients are numbered 0 through 4. An asterisk ('*') can be used to select all coefficients.
Command	Any of the Calibration commands.
Part#	The part number. Part numbers have the syntax ##-#####-##, where # represents a numeric digit.
Serial#	The serial number. Serial numbers are 4 digits.
CellID	The Cell ID. Cell Ids are 5 digits.
Date	The Date. The date syntax is YYYY-MM-DD, where YYY is a 4 digit year, MM is a 2 digit month, and DD is a 2 digit day.
Coefficient	Coefficients are floating point numbers.
Value	The state value to be saved.

Examples:

The following command will set the identification information:

```
Calibration identify 55-55555-55 #### ##### yyyy-mm-dd
```

The following commands will set and display polynomial coefficients values for all 5 indexes:

```
Calibration Polynomial 0 -6.600000e-02
Calibration Polynomial 1 2.541000E-3
Calibration Polynomial 2 -3.086000E-5
Calibration Polynomial 3 1.639000E-7
```



```
Calibration Polynomial 4 -3.050000E-10

Calibration polynomial *
Calibration Polynomial Coefficient 0 is -6.600000e-02
Calibration Polynomial Coefficient 1 is 2.541000e-03
Calibration Polynomial Coefficient 2 is -3.086000e-05
Calibration Polynomial Coefficient 3 is 1.639000e-07
Calibration Polynomial Coefficient 4 is -3.050000e-10
```


17. Temperature Commands

The Temperature commands provide the conversion from AtoD sensor inputs to degrees Celsius. The Temperature command also provides the ability to configure the parameters used in averaging the readings from the temperature sensors for the Temperature Compensation algorithms. Reference this companion document for more detailed information about Temperature Compensation:

- AOTF Controllers and Temperature Compensation, Revision 1.2, www.CrystalTechnology.com, Crystal Technology, Inc. 1040 East Meadow Circle, Palo Alto, CA 94303-4230.

The three temperature sensors are:

Temperature Sensors		
Name	**AtoD Channel**	**Comment**
Crystal	1	This temperature sensor is an LM60 mounted on the Temperature Sensor Board, which is mounted in the enclosure of the AOTF Crystal.
Oscillator	6	This temperature sensor is an LM60 mounted on the *AOTF Controller* near the Oscillator that generates the base frequency for the DDS chips.
Amplifier	7	This temperature sensor is an LM60 mounted on the *AOTF Controller* near the RF Power Amplifier.

Table 4: Temperature Sensors

The Temperature commands consist of the following operations:

Help — Displays a list of Temperature commands and provides detailed syntax for specific commands.

Read — Read the temperature from one of the temperature sensors. The temperature sensor can be Crystal, Oscillator, or Amplifier. An asterisk (‘*’) can be used to specify all sensors.

The –f argument causes the command to read the sensor continuously, until the user presses a key.

Override — Overrides the temperature sensor with a user defined value. The override command provides the ability for the host to supply the temperature from an external temperature sensor. The temperature sensor can be Crystal, Oscillator, or Amplifier. An asterisk (‘*’) can be used to specify all sensors.

With an argument the override temperature is set.

Without an argument the current override temperature is displayed.

To terminate temperature overriding use an asterisk (‘*’) as the argument.

Average	Reads the average temperature. The average temperature is calculated over time, using the Interval and Count parameters. The temperature sensor can be Crystal, Oscillator, or Amplifier. An asterisk (‘*’) can be used to specify all sensors. The –f argument causes the command to read the average continuously, until the user presses a key.
Interval	The time interval for averaging the temperature. Every Interval number of milli-seconds the temperature will be sampled and added into the running average. The temperature sensor can be Crystal, Oscillator, or Amplifier. An asterisk (‘*’) can be used to specify all sensors.
Count	The number of averages taken to create the running average. The running average is created over time. The Count is the number of Intervals that will be used in the running average. A larger value results in slower moving average, creating a low pass filter, and prevents the average from reacting to minor fluctuations in the temperature sensor inputs. The temperature sensor can be Crystal, Oscillator, or Amplifier. An asterisk (‘*’) can be used to specify all sensors.

Syntax:

```
Temperature Help [Command…]
Temperature ? [Command…]
Temperature Read [-f] Oscillator | Crystal | Amplifier | *
Temperature Override Sensor* [Temperature]
Temperature Average [-f] Sensor*
Temperature Interval Sensor* [Ms]
Temperature Count Sensor* [Count]
```

Arguments:

Command	Any of the Temperature commands.
-f	The FOREVER argument causes the command to be executed until the user presses a key.
Oscillator	The Oscillator temperature sensor.
Crystal	The Crystal temperature sensor.
Amplifier	The Amplifier temperature sensor.
Sensor*	The Sensor to read. The sensor can be either the oscillator, crystal, or amplifier. An asterisk (‘*’) can be used to select all sensors.
Temperature	Temperature measured in degrees Celsius.
Ms	Time in milliseconds
Count	Specifies how many averages are taken to create a running average.

Examples:

The following command will display the temperature of the Crystal:

```
Temp r c
Crystal Temperature 29.1437 degrees C
```

The following command will display the temperature of all of the temperature sensors:

```
temp r *
Crystal Temperature 29.2469 degrees C
Oscillator Temperature 30.5875 degrees C
RF Amplifier Temperature 27.9063 degrees C
```

The following command will set the averaging interval to 100ms for the Crystal:

```
temp interval crystal 100
```

The following command will set the number of Intervals used in the averaging algorithm to 5 for the Crystal:

```
temp count crystal 5
```


18. One Wire EEPROM Commands

The One Wire EEPROM commands provide the ability to manipulate the One Wire EEPROM. The One Wire EEPROM resides on the *Daughter Board*, and provides the ability for the *AOTF Controller* to identify the type of *Daughter Board*. The One Wire EEPROM device consists of three independent regions:

ROM Serial Number (8 bytes)

One time programmable Application Register (8 bytes)

EEPROM (32 bytes)

The One Wire EEPROM commands have optional arguments for directing the activity: The default is to apply the operation to the EEPROM region of the One Wire EEPROM. The EEPROM commands consist of the following operations:

Help	Displays a list of One Wire EEPROM commands and provides detailed syntax for specific commands.
Read	Read data from One Wire EEPROM.
Write	Write data to One Wire EEPROM.
Download	Download data from the One Wire EEPROM. The data is formatted into Intel HEX format and displayed. See *Section 27, Intel Hex File Format, on page 70*, for a description of the Intel Hex File format.
Upload	Upload data to the One Wire EEPROM. After issuing the Upload command, the data file should be sent to the *AOTF Controller*. The data is formatted into Intel HEX format and displayed. See *Section 27, Intel Hex File Format, on page 70*, for a description of the Intel Hex File format.
Status	Display the status of the One Wire EEPROM. The status of the One Wire EEPROM is used to determine if the one time programmable Application Register has been programmed.

Syntax:

```
OneWire Help [Command…]
OneWire ? [Command…]
OneWire Read [-cars] Address Count
OneWire Write [-ca] Address Data [Data] [...]
OneWire Download [-car] Count
OneWire Upload [-ca]
OneWire Status
```

Arguments:

Command	Any of the One Wire EEPROM commands.

-c	Append the checksum on write commands, verify the checksum on read commands.
-a	Apply the operation to the Application Register.
-r	Apply the operation to the ROM.
-s	Show the checksum.
Address	The address within the One Wire EEPROM.
Count	The number of bytes to read. Count must be between 0 and 64.
Data	The data to write. A maximum of 64 bytes can be written in a single command.

Examples:

The following command will write the Application Register of the One Wire EEPROM, and append the proper checksum:

```
one write -ac 0 0 1 2 255 255 255 255
```

The following command will display the contents of the ROM of the One Wire EEPROM, and verify the checksum:

```
one read -rc 0 8
0x14 0xD6 0x42 0x78 0x01 0x00 0x00 0x0C
```


19. Modulation Commands

The Modulation commands provide the ability to configure and manipulate the interface signals of the Host Modulation Interface. The Host Modulation Interface is accessed by the connector on the front panel of the *AOTF Controller*. The features and capabilities of the Host Modulation Interface include:

- **Analog Modulation**
  The ability to control the RF power output of the *AOTF Controller* based on an analog input signal.

- **Digital Modulation**
  The ability to control the RF power output of the *AOTF Controller* based on a digital input signal.

- **Profile Selection**
  The FSK and BLANK signals of the Host Modulation Interface allow the host to select the active profile. Profiles can be configured with different frequencies and RF power output levels.

- **Modulation Gain**
  The ability to control the gain of the analog modulation input amplifiers allows the *AOTF Controller* to accommodate different host interface requirements. Some host environments supply an analog voltage from 0V to +10V, while others supply a differential ±5V, and still others supply 0 to +3.3V.

The Host Modulation Interface capabilities are *AOTF Controller* model dependent. Different models of *AOTF Controllers* have different capabilities as shown in *Table 5* and *Table 6*.

AOTF Controller Host Modulation Type		
Model	**Analog Modulation**	**Digital Modulation**
Octal (Standard Power)	With Analog Daughter Card	With Digital Daughter Card
Octal (High Power)	With Analog Daughter Card	With Digital Daughter Card
Quad	With Analog Daughter Card	With Digital Daughter Card
Single FX0	Yes	Yes
Single FX1	Yes	Yes

Table 5: Host Modulation Type

AOTF Controller Host Modulation Gain	
Model	**Modulation Gain**
Octal (Standard Power)	With Analog Daughter Card 1.0V to 10.0V
Octal (High Power)	With Analog Daughter Card 1.0V to 10.0V
Quad	With Analog Daughter Card 1.0V to 10.0V
Single FX0	No The modulation gain is controlled by fixed resistors. The default values configure the host modulation interface for a 10V input range.
Single FX1	No The modulation gain is controlled by fixed resistors. The default values configure the host modulation interface for a 10V input range.

Table 6: Host Modulation Gain

The modulation commands consist of the following operations:

Help — Displays a list of Modulation commands and provides detailed syntax for specific commands.

? — Displays a list of Modulation commands and provides detailed syntax for specific commands.

Digital — For Single Channel *AOTF Controllers* this command enables the digital modulation capability of the Host Modulation Interface. The host provides a clock and data signal that carries the modulation information for the RF output.

For the Octal and Quad *AOTF Controllers* this command is a NOOP. Digital or Analog Modulation is determined by the type of Daughter Board installed. Refer to *Table 5* for a summary of the Modulation Type.

Analog — For the Single Channel *AOTF Controller* this command enables the analog modulation capability of the Host Modulation Interface. The host provides an analog signal that carries the modulation information for the RF output.

The –m argument provides a mechanism for the microcontroller to force the analog modulation to its maximum value, regardless of the value being supplied on the analog input pins of the Host Modulation Interface connector on the front panel of the *AOTF Controller*.

For the Octal and Quad *AOTF Controllers* this command is a NOOP. Digital or Analog Modulation is determined by the type of Daughter Board installed. Refer to *Table 5* for a summary of the Modulation Type.

Gain For Octal and Quad Channel *AOTF Controllers* with Analog Daughter Boards the Gain command adjusts the Gain of the analog amplitude modulation circuit for a specific channel. With an argument, the Gain command sets the Gain of the analog amplitude modulation circuit on the daughter board. The Gain is a number between 0 and 255, with 0 being the minimum, and 255 being the maximum. The Gain circuitry uses non volatile storage for the Gain setting, so the power on reset value is the previous Gain setting before power was removed.

Without an argument, the Gain command displays the current Gain.

The –i argument causes the command to enter an interactive adjustment mode, where the following interactive commands are available:

? Display a list of the adjustment commands.

@ Display the selected channel and current Gain setting.

Change the Gain delta. The delta is the adjustment to the Gain that is applied whenever the + or – commands are issued. The initial delta is 1. After entering the # command, enter decimal digits followed by a carriage return to change the delta.

+ Increment the Gain by delta amount.

- Decrement the Gain by delta amount.

= Set the Gain. After entering the = command, enter decimal digits followed by a carriage return to change the Gain setting.

. (period) Exit the interactive adjustment mode.

For Octal and Quad Channel *AOTF Controllers* with Digital Daughter Boards and Single Channel *AOTF Controllers* this command is a NOOP. Refer to *Table 6* for a summary of the Modulation Gain.

Dac For Octal and Quad Channel *AOTF Controllers* with Digital Daughter Cards, and Single Channel *AOTF Controllers* the DAC command adjusts the DAC output level for a specific channel. With an argument, the DAC command sets the DAC output voltage. The DAC argument is a number between 0 and 16383, with 0 being no amplitude, and 16383 being the maximum amplitude. The power on reset value is 0. The DAC output is translated into a voltage between -1 and +1 Volts to interface to the circuitry on the *AOTF Controller*.

Without an argument, the DAC command displays the current DAC value.

The `-i` argument causes the command to enter an interactive adjustment mode, where the following interactive commands are available:

- \+ Increment the Dac by the delta.
- \- Decrement the Dac by the delta.
- = Set the Dac. After entering the = command, enter decimal digits followed by a carriage return to change the Dac value.
- # Change the Dac delta. The delta is the adjustment to the frequency that is applied whenever the + or – commands are issued. The initial delta is 1. After entering the # command, enter decimal digits followed by a carriage return to change the delta.
- @ Displays the selected channel and Dac value.
- ? Display a list of adjustment commands
- . (period) Exit the interactive adjustment mode.

For Octal and Quad Channel *AOTF Controllers* with Analog Daughter Cards this command is a NOOP. Refer to *Table 5* for a summary of the Modulation Type.

For Octal and Quad Channel *AOTF Controllers* with Digital Daughter Cards the ability of the Microcontroller to affect the DAC is determined by the state of the Daughter Card Enable. See *Section 21, Daughter Board Commands, on page 54,* for additional information.

Fsk

The Fsk command allows the microcontroller to override the FSK pin from the Host Modulation Interface. The FSK pin is used in conjunction with the BLANK pin to select the active profile. Reference this companion document for more detailed information about FSK Operation:

- AOTF Controllers and FSK Operation, Revision 1.3, www.CrystalTechnology.com, Crystal Technology, Inc. 1040 East Meadow Circle, Palo Alto, CA 94303-4230.

With an argument the Fsk command selects the appropriate profile depending on the FSK Mode. See *Section 13, DDS Commands, on page 23*, for additional information about the FSK Mode.

Without an argument, the Fsk command displays the current Fsk state.

For Octal and Quad Channel *AOTF Controllers* the ability of the Microcontroller to affect the FSK is determined by the state of the Daughter Card Enable. See *Section 21, Daughter Board Commands, on page 54,* for additional information.

Blank

The Blank command allows the microcontroller to override the BLANK pin from the Host Modulation Interface. The BLANK pin is used in conjunction with the FSK pin to select the active profile. Reference this companion document for more detailed information about FSK Operation:

- AOTF Controllers and FSK Operation, Revision 1.1, www.CrystalTechnology.com, Crystal Technology, Inc. 1040 East Meadow Circle, Palo Alto, CA 94303-4230.

With an argument the Blank command selects the appropriate profile depending on the FSK Mode. See *Section 13, DDS Commands, on page 23*, for additional information about the FSK Mode.

Without an argument, the Blank command displays the current Blank state.

For Octal and Quad Channel *AOTF Controllers* the ability of the Microcontroller to affect the BLANK is determined by the state of the Daughter Card Enable. See *Section 21, Daughter Board Commands, on page 54,* for additional information.

Syntax:

```
Modulation Help [Command…]
Modulation ? [Command…]
Modulation Digital Channel*
Modulation Analog [-m] Channel*
Modulation DAC [-i] Channel* [Dac]
Modulation FSK Channel* [Fsk]
Modulation Blank Channel* [Blank]
Modulation Gain [-i] Channel* [Gain]
```

Arguments:

Command	Any of the Modulation commands.
Channel*	The DDS channel to access. The DDS channels are numbered 0 through 7. An asterisk (‘*’) can be used to select all channels.
-m	The MAXIMUM argument sets the modulation input to its maximum value
-i	The INTERACTIVE argument causes the command to enter into the interactive adjustment mode.
Dac	The DAC value (0 to 16383).
Fsk	The value of the FSK pin. (0 or 1).
Blank	The value of the Blank pin. (0 or 1).
Gain	The Gain value (0 to 255).

Examples:

The following command will set the FSK pin to 1 for channel 0

```
Modulation FSK 0 1
```

The following command will set the Blank pin to 0 for all channels

```
Modulation Blank * 0
```


20. USB Commands

The USB commands provide the ability to manipulate the USB interface. The USB commands consist of the following operations:

Help	Displays a list of USB commands and provides detailed syntax for specific commands.
Disconnect	Disconnect and then re-connect to the USB bus.
Send	Send text over the USB bus.

Syntax:

```
Usb Help [Command…]
Usb ? [Command…]
Usb Disconnect
Usb Send Text
```

Arguments:

`Command`	Any of the USB commands.
`Text`	The text to send.

Examples:

The following command will disconnect and then re-connect to the USB bus.

```
Usb disconnect
```


21. Daughter Board Commands

The Daughter Board commands provide the ability to manipulate features of the *Daughter Boards*. The Daughter Board commands consist of the following operations:

Help	Displays a list of Daughter Board commands and provides detailed syntax for specific commands.
Enable	Enable the microcontroller to manipulate the Daughter Board controls. When enabled, the external interfaces of the Daughter Board are non operational, and the microcontroller can control the Amplitude Modulation, FSK, and Blank signals.
Disable	Disable microcontroller manipulation of the Daughter Board controls. When disabled, the external interfaces of the Daughter Board are operational, and the external inputs to the Daughter Board control the Amplitude Modulation, FSK, and Blank signals.
Dac	This function has been deprecated. The identical functionality is available with the "`mod dac`" command. The Dac command is still supported, but will eventually be removed. See *Section 19, Modulation Commands, on page 47*, for additional information.
Fsk	This function has been deprecated. The identical functionality is available with the "`mod fsk`" command. The Fsk command is still supported, but will eventually be removed. See *Section 19, Modulation Commands, on page 47*, for additional information.
Blank	This function has been deprecated. The identical functionality is available with the "`mod blank`" command. The Blank command is still supported, but will eventually be removed. See *Section 19, Modulation Commands, on page 47*, for additional information.
Gain	This function has been deprecated. The identical functionality is available with the "`mod gain`" command. The Gain command is still supported, but will eventually be removed. See *Section 19, Modulation Commands, on page 47*, for additional information.

Syntax:

```
Daughter Help [Command…]
Daughter ? [Command…]
Daughter Enable
Daughter Disable
```

NOTE: The following commands have been deprecated:

```
Daughter Dac [-i] Channel* [Dac]
Daughter Fsk Channel* [Fsk]
Daughter Blank Channel* [Blank]
```

```
Daughter Gain [-i] Channel* [Gain]
```

Arguments:

Command	Any of the Daughter Board commands.
-i	The INTERACTIVE argument causes the command to enter into the interactive adjustment mode.
Channel*	The DDS channel to access. The DDS channels are numbered 0 through 7. An asterisk (‘*’) can be used to select all channels.
Dac	The DAC value (0 to 16383).
Fsk	The FSK value, 0 or 1.
Blank	The Blank value, 0 or 1.
Gain	The Gain setting. A value between 0 and 255.

Examples:

The following example will set the DAC output for all channels:

```
Dau dac * 16383
```

The following example will display the Gain setting for all channels:

```
Dau gain *
Channel 0 @ 127
Channel 1 @ 128
Channel 2 @ 200
Channel 3 @ 200
Channel 4 @ 100
Channel 5 @ 100
Channel 6 @ 100
Channel 7 @ 100
```

The following example will set the FSK to 0 for all channels and BLANK to 1 for all channels:

```
Dau Fsk * 0
Dau Blank 0 1
```

22. Chirp Commands

The *Chirp* commands provide the *AOTF Controller* the ability to create sequences of frequencies. The sequences consist of different frequencies that are output from the *AOTF Controller*, creating a spectral chirp pattern in the frequency domain. The sequences are usually a frequency ramp, either up or down, but could consist of any arbitrary sequence of frequencies. The chirp sequence can be output when commanded from the host or the microcontroller, they can automatically repeat, and there is also control over what happens at the end of the sequence.

Only the Single Channel *AOTF Controllers* implement the *Chirp* commands.

A *Chirp* is a single sweep through the sequence of frequencies. Each profile can have a different sequence. Each profile can also have a different number of *Slots* in the sequence, a different *Rate*, and a different *Mode***.**

A *Slot* is one of the 1024 locations in the internal RAM of the *AOTF Controller*. The *Chirp* commands manipulate the *Slots* allocated to each of the profiles. Each profile is initially allocated a single *Slot*. *Slots* are numbered from 0 to 1020. Each profile has its own set of *Slots*, so *Slot* 0 of profile 0 is different than *Slot* 0 of profile 1.

The *Rate* is the number of synclocks that the *AOTF Controller* will dwell on each of the *Slots* during a *Chirp* sequence. Each profile has its own *Rate*. The *Rate* is a number from 1 to 65535. 0 is invalid. A single synclock is 4 of the *AOTF Controller's* internal clock cycles. The internal clock of the *AOTF Controller* is 400Mhz, so a single synclock is 100Mhz, or 10ns.

The *Allocation* is the number of slots *Allocated* to a profile. Each profile has at least one *Slot*. The largest possible *Allocation* is 1021, which gives 1021 *Slots* to a single profile, and then 1 *Slot* to each of the remaining profiles.

The *Mode* is one of the following:

- **Mode 0 – Single Tone**
  This Mode is a single output frequency. *Slot* 0 of the chirp sequence is the source for the frequency that generates the output.

  Any change on the FSK or BLANK signals will change profiles and begin the *Chirp* sequence of the associated profile.

- **Mode 1 – Ramp up**
  The most common use of this Mode is to use the Manual Sweep input from the Host Modulation Interface connector on the front panel of the *AOTF Controller* to trigger a *Chirp* sequence. The "`Chirp Chirp`" command can also be used to initiate a software generated *Chirp* sequence. This Mode will sweep through the *Chirp* sequence using the *Slots* of the RAM, the *Rate*, and the *NoDwell* setting. The *Chirp* sequence will start from *Slot* 0 and precede through to the last *Slot*. When the last *Slot* is reached the *AOTF Controller's* output will either remain outputting the final *Slot* contents or the output will turn off, depending on the *NoDwell* setting.

  Any change on the FSK or BLANK signals will change profiles and begin the *Chirp* sequence of the associated profile.

The Manual Sweep input from the Host Modulation Interface connector on the front panel f the *AOTF Controller* can be used to restart a *Chirp* sequence

- **Mode 2 – Bidirectional Ramp**
  This Mode provides a ramp-up and a ramp-down capability. If the FSK signal on the Host Modulation Interface connector on the front panel of the *AOTF Controller* is 0 then the *AOTF Controller* will ramp-up. If the FSK signal is 1 the *AOTF Controller* will ramp-down.

  Since Mode 2 uses the FSK signal, it is only available for profile 0. While Mode 2 is in use, profiles 1, 2, and 3 can not be used.

  The Manual Sweep input from the Host Modulation Interface connector on the front panel f the *AOTF Controller* can be used to restart a *Chirp* sequence

- **Mode 3 – Continuous Bidirectional Ramp**
  This Mode provides a continuous bidirectional ramp-up/ramp-down capability. The *Chirp* sequence begins at *Slot* 0 and precedes to the last *Slot*. Then the sequence begins in reverse, from the last *Slot* down to *Slot* 0. The entire *Chirp* sequence then begins again.

  Any change on the FSK or BLANK signals will change profiles and begin the *Chirp* sequence of the associated profile.

  The Manual Sweep input from the Host Modulation Interface connector on the front panel of the *AOTF Controller* can be used to restart a *Chirp* sequence

- **Mode 4 – Recirculate Ramp-Up**
  This Mode provides a continuous ramp-up capability. The *Chirp* sequence begins at *Slot* 0 and precedes to the last *Slot*. The entire *Chirp* sequence then begins again.

  Any change on the FSK or BLANK signals will change profiles and begin the *Chirp* sequence of the associated profile.

  The Manual Sweep input from the Host Modulation Interface connector on the front panel f the *AOTF Controller* can be used to restart a *Chirp* sequence

The *NoDwell* is a term for what happens at the end of a Chirp sequence. The choices are:

- **Dwell**
  Stop at the last frequency and wait. The last frequency will continue to be output from the *AOTF Controller*.

- **NoDwell**
  The *AOTF Controller* will stop outputting a signal (i.e. 0MHz).

The Chirp commands consist of the following operations:

Help	Displays a list of Chirp commands and provides detailed syntax for specific commands.
?	Displays a list of Chirp commands and provides detailed syntax for specific commands.

Fill	The *Fill* command is the quickest and easiest way to fill the *Slots* of a *Chirp* sequence. The *Slots* allocated to the specified profile are initialized to contain FTWs that correspond to a ramp from *Start* frequency to *End* frequency. The command calculates how many *Slots* are available, and then fills each *Slot* with an incrementing FTW, evenly spaced between *Start* frequency and *End* frequency. If the *Start* and *End* frequencies are absent from the command line the current contents of each *Slot* is displayed. If *Start* frequency is present, but *End* frequency is absent, then the entire sequence will be filled with the *Start* frequency. See *Section 5,Syntax for Frequencies, Wavelengths, and Frequency Tuning Words (FTW), on page 11*, for details about the syntax for frequencies. The `-p` argument specifies the profile. Each DDS channel has 4 profiles, numbered 0 to 3. When the DDS is operating in single tone mode, only profile 0 is utilized. When operating in FSK mode the profile selection is controlled with the FSK and BLANK signals. The -p argument allows the user to select the profile. If not specified, the profile defaults to 0. Using an asterisk ('*') for the profile specifies all 4 profiles.
Frequency	The Frequency command provides the ability to create custom sequences. With this command a user can build up a *Chirp* sequence one *Slot* at a time, and edit a *Chirp* sequence by updating individual *Slots*. If the *Frequency* is absent from the command line, the current contents of the *Slot* are displayed. If *Slot* is an asterisk ('*') then all *Slots* will be affected. See *Section 5,Syntax for Frequencies, Wavelengths, and Frequency Tuning Words (FTW), on page 11*, for details about the syntax for frequencies. The `-p` argument specifies the profile. Each DDS channel has 4 profiles, numbered 0 to 3. When the DDS is operating in single tone mode, only profile 0 is utilized. When operating in FSK mode the profile selection is controlled with the FSK and BLANK signals. The -p argument allows the user to select the profile. If not specified, the profile defaults to 0. Using an asterisk ('*') for the profile specifies all 4 profiles.
FTW	This function has been deprecated. Beginning with the firmware release of December 2008 the functionality of the *FTW* command has been incorporated into the syntax for specifying a frequency. The *FTW* command is still supported, but will eventually be removed. The *Frequency* command supports the same functionality.
Rate	The *Rate* command provides the ability to configure the amount of time that the *AOTF Controller* will dwell on each *Slot* of the *Chirp* sequence. The *Rate* is the number of synclocks that the *AOTF Controller* will dwell on each of the *Slots* during a *Chirp* sequence. Each profile has its own *Rate*. The *Rate* is a number from 1 to 65535. 0 is invalid. A single synclock is 4 of the *AOTF Controller's* internal clock cycles. The internal clock of the *AOTF Controller* is 400Mhz, so a single synclock is 100Mhz, or 10ns.

If the *Rate* is absent from the command line, then the current *Rate* is displayed.

The duration of the entire *Chirp* sequence is the number of *Slots* * *Rate* * 10ns.

The `-p` argument specifies the profile. Each DDS channel has 4 profiles, numbered 0 to 3. When the DDS is operating in single tone mode, only profile 0 is utilized. When operating in FSK mode the profile selection is controlled with the FSK and BLANK signals. The -p argument allows the user to select the profile. If not specified, the profile defaults to 0. Using an asterisk (‘*’) for the profile specifies all 4 profiles.

Mode

The *Mode* command determines how the *AOTF Controller* outputs the *Chirp* sequence. Mode is a value from 0-4 and is described above.

The –m argument configures the Manual Sweep feature of the Host Modulation Interface on the front panel connector of the *AOTF Controller*. The Manual Sweep uses the front panel connector and the differential CHIRP_P/N input signals.

When the Manual Sweep feature is enabled a rising edge on the CHIRP_P/N signals will generate a sweep. If the rising edges of the CHIRP_P/N signals are faster than the duration of a *Chirp* sequence then the *Chirp* sequence will be interrupted and the sequence will start again from the beginning.

If the –m argument is present then the Manual Sweep feature is enabled. If the –m argument is absent then the Manual Sweep feature is disabled.

The –d argument configures the *NoDwell* setting for the profile. If the –d argument is present then the *AOTF Controller* will stop outputting an RF signal (i.e. 0MHz) at the end of the *Chirp* sequence. This effectively turns off the RF output.

Without the –d argument, when the *Chirp* sequence is over, the *AOTF Controller* will continue to output the last frequency of the *Chirp* sequence.

The `-p` argument specifies the profile. Each DDS channel has 4 profiles, numbered 0 to 3. When the DDS is operating in single tone mode, only profile 0 is utilized. When operating in FSK mode the profile selection is controlled with the FSK and BLANK signals. The -p argument allows the user to select the profile. If not specified, the profile defaults to 0. Using an asterisk (‘*’) for the profile specifies all 4 profiles.

Allocate

The *Allocate* command is used to allocate *Slots* to each of the profiles. Each channel has 1024 *Slots* that can be allocated among the profiles. The minimum number of *Slots* allocated to each profile is 1. The maximum number of *Slots* that can be allocated to a single profile is 1021. Initially each profile is allocated 1 *Slot* and there are 1020 *Slots* free (unallocated). Slots are numbered starting at 0.

The allocation should be done prior to other commands because re-allocating the *Slots* assigned to each of the profiles destroys the contents of the sequence.

If the *Allocation* is absent on the command line the current *Allocation* is displayed.

The `-p` argument specifies the profile. Each DDS channel has 4 profiles, numbered 0 to 3. When the DDS is operating in single tone mode, only profile 0 is utilized. When operating in FSK mode the profile selection is controlled with the FSK and BLANK signals. The -p argument allows the user to select the profile. If not specified, the profile defaults to 0. Using an asterisk (‘*’) for the profile specifies all 4 profiles.

Chirp

The *Chirp* command provides the ability for the microcontroller to perform the same action as the Manual Sweep capability of the Host Modulation Interface. If the configured *Mode* supports the Manual Sweep capability, then the *Chirp* command can also be used for the same purposes.

The `-i` argument causes the command to enter an interactive adjustment mode, where the following interactive commands are available:

- \+ Will initiate a *Chirp*.
- 0 Set the FSK signal to 0.
- 1 Set the FSK signal to 1.
- ? Display a list of adjustment commands.
- . (period) Exit the interactive adjustment mode.

Syntax:

```
Chirp Help [Command…]
Chirp ? [Command…]
Chirp Fill [-p*] Channel* [Start [End]]
Chirp Frequency [-p*] Channel* Slot* [Frequency]
Chirp Rate [-p*] Channel* [Rate]
Chirp Mode [-mdp*] Channel* [Mode]
Chirp Allocate [-p*] Channel* [Allocation]
Chirp Chirp [-i]
```

NOTE: The following commands have been deprecated:

```
Chirp FTW [-p*] Channel* Slot* [FTW]
```

Arguments:

Command — Any of the *Chirp* commands.

-p* — The PROFILE argument is used to display or modify the value of a profile. The profiles are numbered 0 through 3. An asterisk (‘*’) can be used to select all profiles.

Channel* — The DDS channel to access. The DDS channels are numbered 0 through 7. An asterisk (‘*’) can be used to select all channels.

Start — The starting frequency for the *Chirp* sequence. See *Section 5,Syntax for Frequencies, Wavelengths, and Frequency Tuning Words (FTW), on page 11*, for details about the syntax for frequencies.

End	The ending frequency for the *Chirp* sequence. See *Section 5,Syntax for Frequencies, Wavelengths, and Frequency Tuning Words (FTW), on page 11*, for details about the syntax for frequencies.
Slot*	Slot is a value from 0-1023.
Frequency*	A frequency to be placed into a Slot. See *Section 5,Syntax for Frequencies, Wavelengths, and Frequency Tuning Words (FTW), on page 11*, for details about the syntax for frequencies.
Rate	Rate is a value from 0 to 65535.
-m	The MANUAL argument configures the Manual Sweep feature of the *AOTF Controller*.
-d	The NODWELL argument determines what the *AOTF Controller* will output after th Chirp sequence terminates.
Mode	Mode is a value from 0-4. The Mode determines how the *AOTF Controller* will interpret the *Chirp* sequence.
Allocation	Allocation is a value from 1-1021.
-i	The INTERACTIVE argument allows the user to enter an interactive adjustment mode.

Examples:

The following commands will allocate 1021 *Slots* to profile 0 for channel 0, and 1 *Slot* to profiles 1 through 3.

```
Chirp Allocate –p* 0 1
Chirp Allocate –p0 0 1021
```

The following commands will allocate 1021 *Slots* to profile 0 of channel 0, and then fill the 1021 *Slots* with a sequence of frequencies from 100Mhz to 120Mhz.

```
Chirp Allocate –p* 0 1
Chirp Allocate –p0 0 1021
Chirp Fill -p0 0 100 120
```

The following commands will allocate different sized sequences to each of the profiles of channel 0, and then fill each of the sequences with frequencies corresponding to frequencies from 100Mhz to 120Mhz. But each of the profiles will have a different granularity in the frequency steps because each of them has a different number of *Slots* in the sequence. Profile 0 with have the roughest granularity and profile 3 will have the finest granularity.

```
Chirp Allocate –p* 0 1
Chirp Allocate –p0 0 10
Chirp Allocate –p1 0 50
```



```
Chirp Allocate –p2 0 100
Chirp Allocate –p3 0 500
Chirp Fill –p* 0 100 120
```

The following command will write 125Mhz to Slot 5 of profile 1 of channel 0.

```
Chirp Frequency –p1 0 5 125
```

The following command will set the *Rate* for all profiles of channel 0 to 1000 synclocks, which is 10 microseconds (1000 x 10ns) for each *Slot* of the sequence.

```
Chirp Rate -p* 0 1000
```

The following command will set the mode for profile 0 of channel 0 to mode 4.

```
chirp mode -p0 0 4
```

The following command will initiate a chirp.

```
chirp chirp
```


23. BoardID Commands

The BoardID commands provide the capability to identify the AOTF Controller and access the unique identification information.

Help	Displays a list of BoardID commands and provides detailed syntax for specific commands.
Version	Displays the date of the firmware as well as the microcontroller platform (FX0 or FX1)
Serial	Displays the serial number of the internal microcontroller.
Identify	Daughter Card Type
Architecture	Displays the microcontroller platform (w/o the date of the firmware)
PartNumber	Displays the part number of the AOTF Controller
ModelNumber	Displays the model number of the AOTF Controller
CTISerialNumber	Displays the serial number of the AOTF Controller
Date	Displays the date the *AOTF Controller* was manufactured
Options	Displays any user described string of information
Save	Saves the Board information into the bootloader EEPROM

Syntax:

```
BoardId Help [Command…]
BoardId ? [Command…]
BoardId Version
BoardId Serial
BoardId Identify
BoardId Architecture
BoardId PartNumber
BoardId ModelNumber
BoardId Date
BoardId Options
BoardId Save
```

Arguments:

Command	Any of the BoardID commands.

Examples:

The following example will display the date of the firmware as well as the microcontroller platform:

```
Boardid revision
Revision Aug 03 2008 19:19:53 Architecture FX0-8
```

The following example will display the serial number of the internal microcontroller:

```
Boardid serial
 Serial Number: 142E035103000057
```

The following example will display the serial number of the AOTF Controller

```
Boardid ctiserialnumber
SerialNumber 1262
```

The following example will display the part number of the AOTF Controller

```
Boardid partnumber
PartNumber 97-02861-10
```


24. Peak Light Intensity

The Peak Light Intensity algorithms require a Light Intensity meter connected to the Light Intensity sensor input of the *AOTF Controller*. Configure the host environment as shown in *Figure 1*:

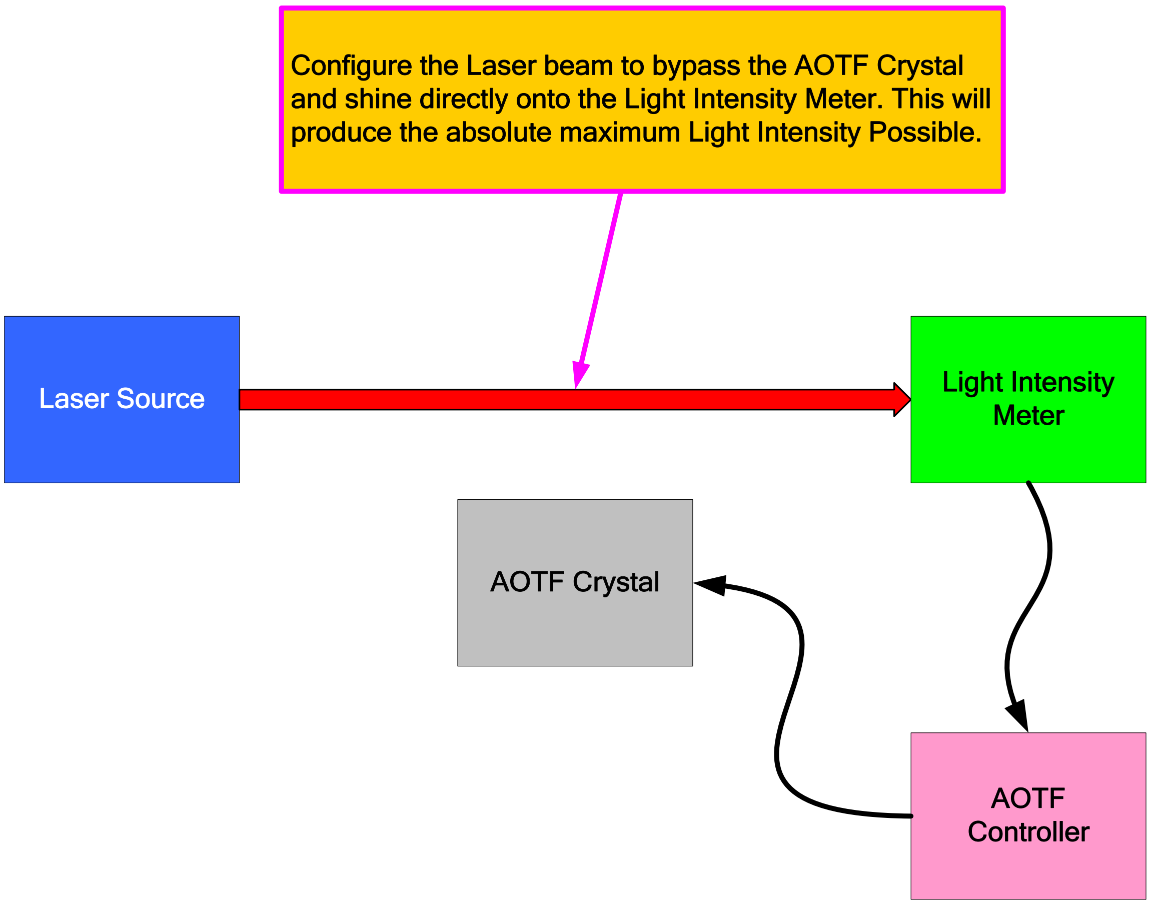

Figure 1: Peak Light Intensity Calibration

This phase of the Peak Light Intensity calibration should be performed with the modulation amplitude configured for maximum intensity. The modulation amplitude can be either Analog Modulation or Digital Modulation, depending on the configuration of the *AOTF Controller*. NOTE: It's possible to overdrive the AOTF Crystal, and the final phase of the Peak Light Intensity calibration will attempt to compensate for the overdrive. For now adjust the modulation for maximum intensity.

This configuration will allow the *AOTF Controller's* sensor input to be calibrated for the correct amount of gain to produce the optimum range for the sensor input. With the host environment correctly configured, issue this command:

```
Adc Peak 2
```

The "`Adc Peak`" command will adjust the gain of sensor input 2, which is the Light Intensity sensor input, for a full scale reading. This will make the full range f the ADC available for the next phase of the Peak Light Intensity algorithm.

Alternatively, the AODS Controller application can accomplish the same command as shown in *Figure 2*:

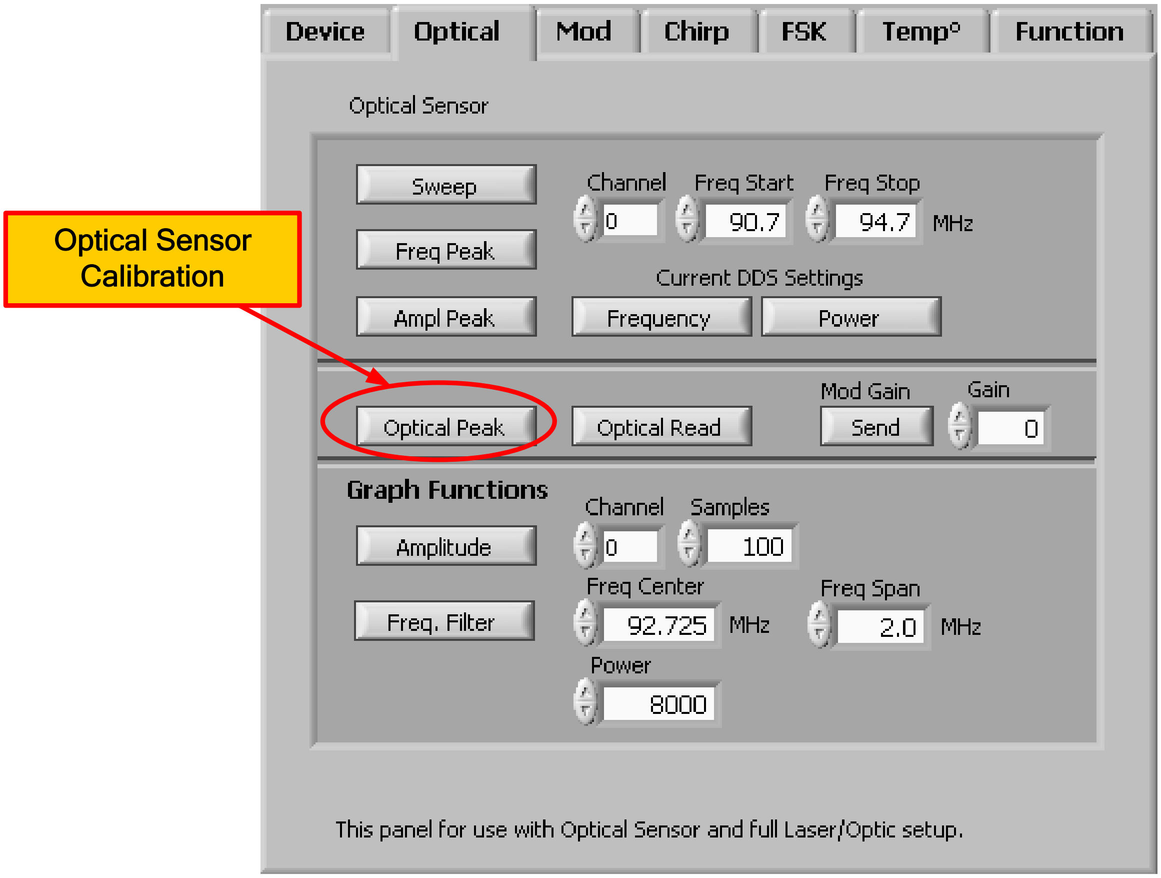

Figure 2: ADC Optical Sensor Calibration

Reconfigure the host environment as shown in *Figure 3*:

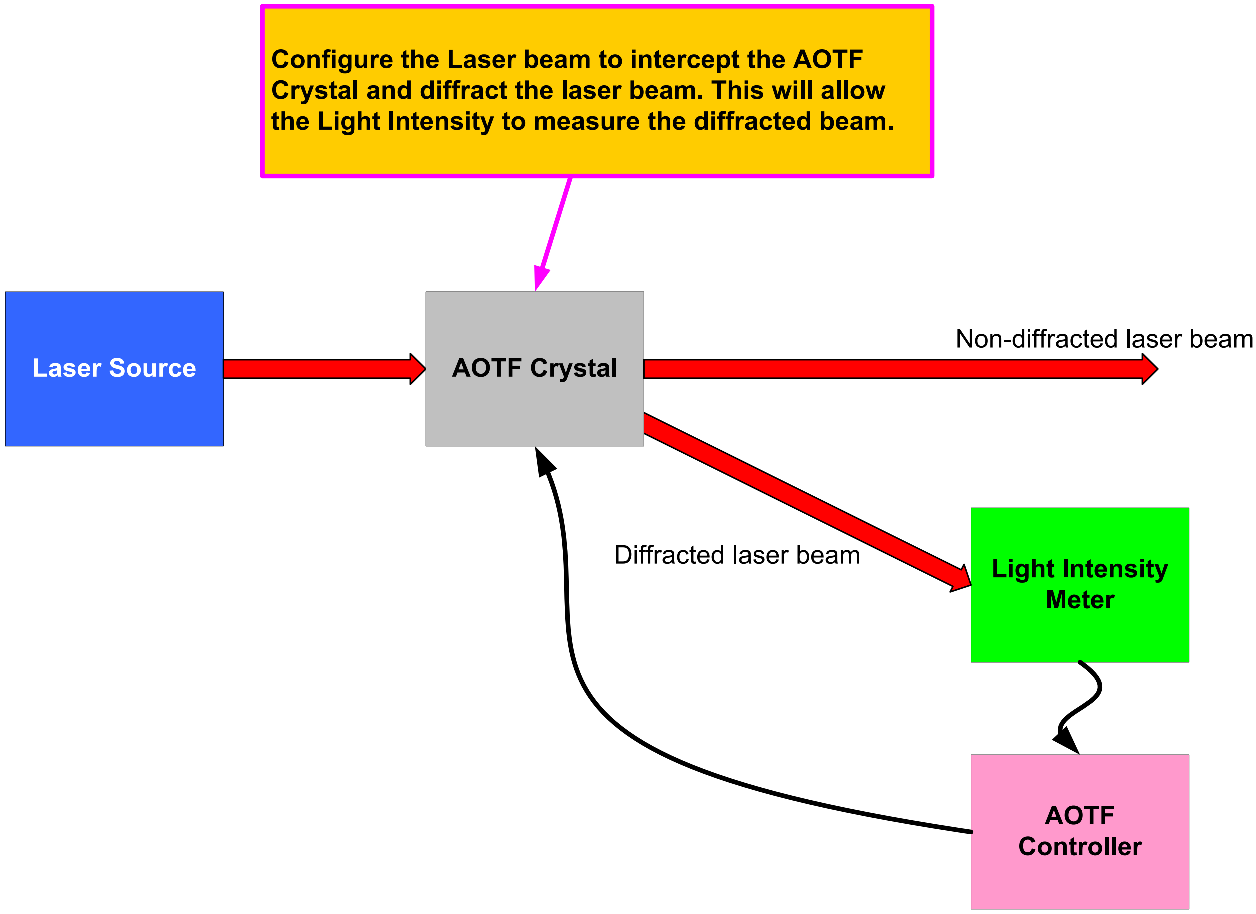

Figure 3: Peak Light Intensity

With the host environment configured like *Figure 3* issue this command:

```
Dds Peak 0 50 150
```

This will use channel 0 of the *AOTF Controller* to sweep through the RF spectrum from 50MHz to 150MHz and search for the Peak Light Intensity of the diffracted beam. When the algorithm completes the frequency of channel 0 will be positioned on the frequency that provided the Peak Light Intensity.

Alternatively, the AODS Controller application can accomplish the same command as shown in *Figure 2*:

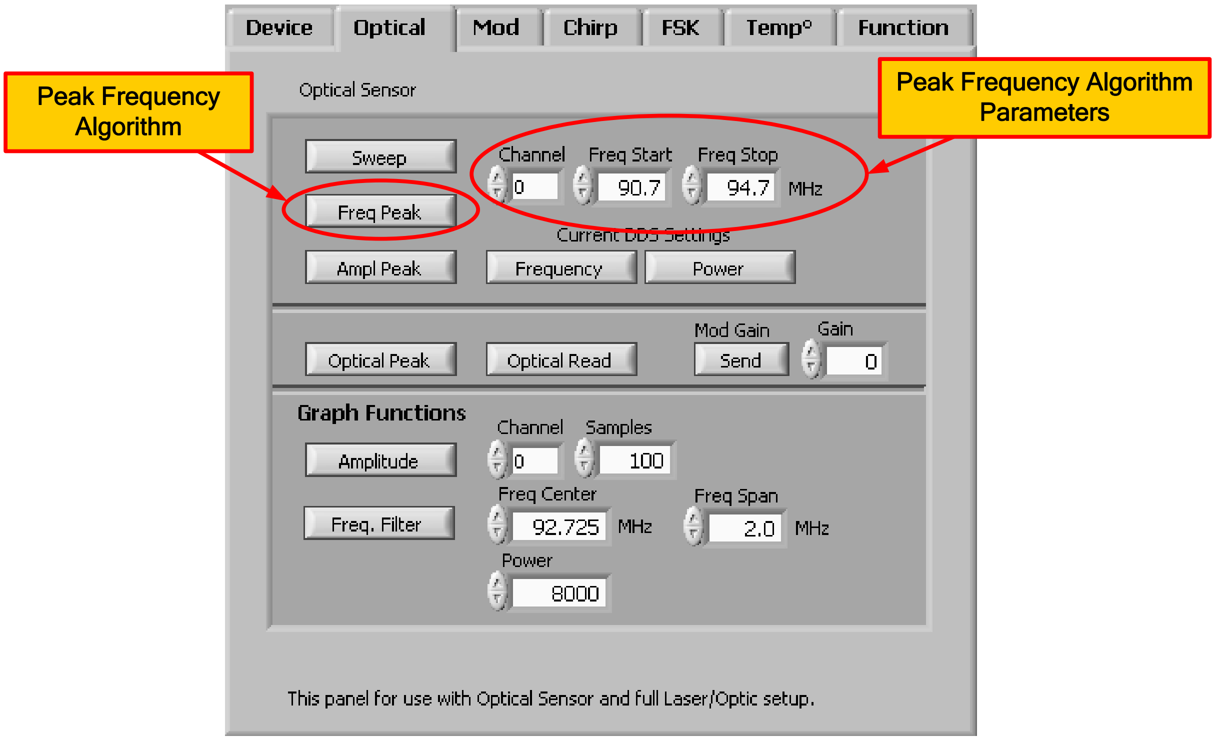

Figure 4: Peak Frequency Algorithm

Now that the Peak Frequency has been determined, the optimum modulation can be configured. It's possible to overdrive the AOTF Crystal, which results in less diffracted light as the modulation voltage increases beyond the optimum. There are two ways to optimize the modulation voltage:

- **<u>Adjust the Modulation Voltage</u>**
  This is accomplished by having the host environment reduce the modulation voltage to 0 and then slowly raising the modulation voltage to maximum while tracking the Peak Light Intensity on the light intensity meter. This calibration step requires that the host perform the modulation voltage adjustments while using the *AOTF Controller's* "`Adc Read`" command to track the light intensity meter sensor input and record the modulation voltage setting that results in the maximum light intensity. This is the preferred technique because it preserves the maximum range for the SIN wave generator, and allows the DDS chip to produce the best possible SIN wave with the most number of points along the SIN wave curve.

- **<u>Adjust the Amplitude Scale Factor</u>**
  The Amplitude Scale Factor (ASF) of the DDS chip can be adjusted to reduce the amplitude output, and thereby reducing the RF power output from the DDS chip. The ASF can be adjusted manually with the "`dds amplitude`" command. Alternatively the "`dds amppeak`" command can perform an automatic adjustment of the ASF to optimize the Peak Light Intensity. The "`dds amppeak`" command increments the ASF from 0 to the maximum and observes the light intensity sensor input. When the command completes the ASF will be set to the value that produced the maximum light intensity.

25. Bluetooth Interface

The Bluetooth interface provides a wireless remote access capability to the *AOTF Controller*. The Bluetooth capability is implemented with a Bluetooth module connected to a serial port interface on the *AOTF Controller*. The Bluetooth module is configured as an *Acceptor* device, and is continuously listening for incoming Bluetooth connections.

The Bluetooth module supports the Serial Port Profile (SPP), which allows two Bluetooth devices to communicate wirelessly, as if they were connected via an RS232 serial cable. The Bluetooth connection wirelessly transports the serial port communication data between the two devices.

On top of the Bluetooth interface, two separate and independent mechanisms have been provided for communication:

- No Protocol communication (RAW) using utilities such as ZTerm, Procomm, and Hyperterm. The RAW communication is the simplest way to begin using the Bluetooth interface. The host computer (either a PC or PDA) uses one of the many serial communication programs to issue commands to the *AOTF Controller*.

- Network Protocol communication (Networking). The Networking communication provides a more robust and feature rich environment for communicating with the *AOTF Controller*. Two standard networking protocol interfaces are provided:

    - Telnet is a terminal emulation protocol that provides an interface identical to the command line interface of the *AOTF Controller*.

    - Hypertext Transfer Protocol is a web browser interface that uses the web browser of the host computer to provide a graphical interface for issueing commands to the *AOTF Controller*.

To use the Bluetooth interface, first make a Bluetooth connection to the Bluetooth module, and then use either the RAW or Networking protocol utilities to communicate with the *AOTF Controller*.

26. Network Protocols

The Network Protocols implemented in the *AOTF Controller* provide a robust communication mechanism that is built on standard networking protocols. The Network Protocols implemented in the *AOTF Controller* consist of the following:

- Point To Point Protocol (PPP)
- Link Control Protocol (LCP)
- Password Authentication Protocol (PAP)
- Internet Protocol Control Protocol (IPCP)
- Internet Control Message Protocol (ICMP)
- Internet Protocol (IP)
- User Datagram Protocol (UDP)
- Transmission Control Protocol (TCP)
- Telnet Protocol (Telnet)
- Hypertext Transfer Protocol (HTTP)

27. Intel Hex File Format

The Intel HEX file is an ASCII text file with lines of text that follow the Intel HEX file format. Each line in an Intel HEX file contains one HEX record. These records are made up of hexadecimal numbers that represent machine-language code and/or constant data.

Intel HEX files are often used to transfer the program and data that would be stored in a ROM or EPROM.

An Intel HEX file is composed of any number of HEX records. Each record is made up of five fields arranged in the following format:

```
:llaaaatt[dd...]cc
```

Each group of letters corresponds to a different field, and each letter represents a single hexadecimal digit. Each field is composed of at least two hexadecimal digits (which make up a byte) as described below:

:	The colon that starts every Intel HEX record.
ll	The record-length field that represents the number of data bytes (dd) in the record.
aaaa	The address field that represents the starting address for subsequent data in the record.
tt	The type field that represents the HEX record type, which can be one of the following:
	00 data record

01 end-of-file record

02 8086 segment address record

04 extended linear address record

dd The data field that represents one byte of data. A record can have several data bytes. The number of data bytes in the record must match the number specified by the ll field.

cc The checksum field that represents the checksum of the record. The checksum is calculated by adding the values of all hexadecimal digit pairs in the record modulo 256 and taking the two's complement.

27.1. Data Record

The Intel HEX file is made up of any number of data records terminated with a carriage return and a linefeed. Data records appear as follows:

```
:10246200464C5549442050524F46494C4500464C33
```

Where

10	is the number of data bytes in the record.
2462	is the address where the data are to be located in memory.
00	is the record type 00 (a data record).
464C...464C	is the data.
33	is the checksum of the record.

27.2. Segment Address Record

The Intel HEX file contains 8086 segment address records to specify a paragraph number (one paragraph is 16 bytes). This record type is replaced by an extended linear-address record if you are using the H167 directive. The paragraph number is used as the offset for all subsequent data records in the HEX file. 8086 segment address records appear as follows:

```
:020000021000EC
```

Where

02	is the number of data bytes in the record.
0000	is always 0 in an extended 8086 segment record.
02	is the record type 02 (an extended linear-address record).
1000	is the paragraph number (address: 0x10000).
EC	is the checksum of the record.

27.3. Extended Address Record

The Intel HEX file contains extended linear-address records when the H167 directive is used. This record specifies the two most significant bytes (bits 16 and 31) of the absolute address. This address offset is used for all subsequent data records in the HEX file. Extended linear-address records appear as follows:

```
:0200000400FFFB
```

Where

02	is the number of data bytes in the record.
0000	is always 0 in a extended 8086 segment record.
04	is the record type 04 (an extended linear-address record).
00FF	is the high word of the address offset (0x00FF0000).
FB	is the checksum of the record.

27.4. End-of-File (EOF) Record

An Intel HEX file must end with an end-of-file (EOF) record. This record must have the value 01 in the record type field. An EOF record always appears as follows:

```
:00000001FF
```

Where

00	is the number of data bytes in the record.
0000	is the address where the data are to be located in memory. The address in end-of-file records is meaningless and is ignored. An address of 0000h is typical.
01	is the record type 01 (an end-of-file record).
FF	is the checksum of the record and is calculated as 01h + NOT(00h + 00h + 00h + 01h).

