

# **Crystal Technology, Inc.**

## **AOTF Utilities Release Notes**

**Revision 1.2**

**2010/08/11**

**Document #60-00100-01**



**Crystal Technology, Inc.**

**A Group Company of TDK-EPC Corporation**

Reproduction of the contents of this document without the permission of Crystal Technology, Inc. is expressly prohibited.

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
1.0	2008/12/05	Dale Gifford	Genesis.
1.1	2009/01/05	Dale Gifford	Release 2009-01-05
1.2	2010/08/11	Dale Gifford	Release 2010-08

**Table 1: Revision History**

## Table of Contents

<b>1. Introduction .....</b>	<b>8</b>
1.1. Purpose.....	8
1.2. Related Documents .....	8
1.3. Notation .....	8
<b>2. Release Notes 2010/08/11 .....</b>	<b>10</b>
<b>3. AotfUsbDriver and AotfLibrary .....</b>	<b>12</b>
<b>4. Cypress .....</b>	<b>14</b>
4.1. Microsoft .NET Framework (CyUSB.dll) .....	14
4.2. C++ (CyAPI.lib) .....	14
4.3. Device Driver .....	15
<b>5. Release Notes 2009/01/05 .....</b>	<b>16</b>
<b>6. AotfDriver and AotfLibrary.....</b>	<b>18</b>
<b>7. Cypress .....</b>	<b>21</b>
7.1. Microsoft .NET Framework (CyUSB.dll) .....	21
7.2. C++ (CyAPI.lib) .....	21
7.3. Device Driver (CyUSB.sys).....	21
<b>8. Firmware Releases .....</b>	<b>22</b>

**List of Tables**

Table 1: Revision History ..... iii

## List of Figures

Figure 1: AotfUsbDriver and AotfLibrary .....	13
Figure 2: AotfDriver and AotfLibrary .....	20

This Page Intentionally Left Blank

## 1. Introduction

### 1.1. Purpose

This document contains important information about the hardware, software, and firmware of Crystal Technology's *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](http://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](http://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](http://www.CrystalTechnology.com), Crystal Technology, Inc. 1040 East Meadow Circle, Palo Alto, CA 94303-4230.
- AOTF Controller Command Reference, Revision 1.3, [www.CrystalTechnology.com](http://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](http://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](http://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](http://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](http://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](http://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. Release Notes 2010/08/11

This release of the AOTF Utilities includes a new setup utility that will easily install and configure the AOTF Utilities software. To install the software execute the Setup.exe command and follow the instructions. These utilities and directories are installed by the Setup utility:

- **AotfManager**  
AotfManager is the replacement for AotfChat. AotfManager has many new features, such as the simultaneous communication with multiple AOTF Controllers, command history, drag and drop capability for script files, and PnP support. See the AotfManager Users Guide for additional information.
- **AotfCmd**  
AotfCmd is a utility for communicating with AOTF Controllers. AotfCmd is a console application that can be used in a CMD window and within batch (BAT) files. AotfCmd is registered with Microsoft Windows as the application to execute when files with the “.aotf” extension are executed. AotfCmd sends the commands in aotf files to the AOTF Controller for execution. See the AotfCmd Users Guide for additional information.
- **AODS Controller**  
AODS Controller is a LabView application that provides a Graphical User Interface (GUI) for the AOTF Controllers. This requires the installation of the National Instruments LabView Runtime Engine (included with the AOTF Utilities).
- **AotfChat**  
AotfChat is the legacy version of the utility for communicating with AOTF Controllers. AotfChat will soon be deprecated. It is included for customers that are not yet ready to migrate their applications and software over to the new Plug and Play device driver architecture. The replacement utility is AotfManager.
- **Documentation**  
The Documentation folder contains all of the latest documentation for the AOTF Controllers. Including a Quick Start Guide, User Guides, Integration Guides, and helpful usage guides for light intensity and temperature tracking. The documents are provided in PDF format and require the installation of the Adobe Acrobat PDF Reader utility. The Adobe Acrobat Reader can be downloaded from Adobe at this address: [Download Adobe Acrobat Reader](#)
- **AotfUsbDriver (32 bit+64 bit)**  
This release of the AOTF Utilities includes an updated device driver that supports these 32 bit and 64 bit platforms:
  - Windows 2000 (32 bit only, there was never a 64 bit release)
  - Windows XP (both 32 bit and 64 bit)
  - Windows Vista (both 32 bit and 64 bit)
  - Windows 7 (both 32 bit and 64 bit)

There have also been bug fixes and enhancements to the driver including improved support for Plug and Play (PnP) capability. See *Section 3, AotfUsbDriver, on page 12* for more details about the device driver.
- **LabView**  
The LabView directory contains the National Instrument’s Run Time Engine (RTE) for LabView applications. LabView RTE is required by the AODS Controller application.

- **Developer**

The Developer directory is where software developers will find all of the necessary tools and documentation for the libraries, DLLs, and device drivers that facilitate integration of the AOTF Controllers into custom applications. See *Section 4, Cypress, on page 14* for more details about the developer tools.

- **Developer/AotfLibrary**

AotfLibrary has changed only slightly with this release. Extensive testing has been done to insure compatibility with both the 32 bit and 64 bit device drivers. With this release the support for the legacy AotfDriver has been removed. See *Section 3, AotfUsbDriver, on page 12* for more details about AotfLibrary.

- **Developer/Cypress**

The Cypress directory contains the documentation for the libraries, DLLs, and device drivers that facilitate integration of the AOTF Controllers into custom applications. There is support for C, C++, and .NET applications. See *Section 4, Cypress, on page 14* for more details about the Cypress supplied utilities.

- **Developer/AotfLibraryTest**

The AotfLibraryTest directory contains a sample application that uses the AotfLibrary DLL to communicate with AOTF Controllers. This sample application provides a utility and source code example that developers can use as a guide for how to incorporate AOTF Controller communication into their own applications.

### 3. AotfUsbDriver and AotfLibrary

With this release of AotfUsbDriver and AotfLibrary the software architecture has gotten less complicated. Support for the Legacy driver has been deprecated and is no longer supported. This release includes a new device driver that supports both 32 bit and 64 bit platforms. All models of the AOTF Controllers (Octal, Quad, and Single) are supported by the AotfUsbDriver in this release, regardless of the PID configuration. The previous release of the AOTF Utilities used the PID to distinguish between using the Legacy device driver or the newer Plug and Play capable device driver. With this release of the AOTF Utilities all PIDs are supported by the same AotfUsbDriver, either 32 bit or 64 bit versions, depending on the host platform architecture.

AotfLibrary continues to be supported with this release. No new features or capabilities were added to AotfLibrary, which means that it is adequate for simple applications that do not need Plug and Play support. If applications need Plug and Play capability they should interface directly with the Cypress CyAPI and CyUSB.NET libraries and bypass the simple interface provided by AotfLibrary.

Improved Plug and Play support is only available by using the Cypress supplied libraries (CyAPI and CyUSB.NET). Environments that require Plug and Play support have no alternative except to migrate to the new architecture. The advantages, besides the Plug and Play capability, are:

- Improved documentation for all levels of developer effort, including Microsoft's' .NET Framework, C++, Visual Basic, LabView, and direct device driver interaction via IOCTLs.
- Improved developer support, and additional utilities for integrating AOTF communication into customer applications, such as AotfManager.

*Figure 1 on page 13* is divided into left and right halves. The left half shows the existing software architecture, with AotfChat, AotfLibrary, AotfDriver, and custom applications. The architecture diagram shows how the various components interact with each other and how they form a layered architecture for AOTF Controller integration. The right half shows the new architecture, with similar components, but without the support for the legacy device driver.

To take full advantage of the architecture applications need to migrate away from AotfLibrary and begin interfacing directly with one of the Cypress supplied techniques for communicating with USB devices. See *Section 4, Cypress, on page 14* for additional information on the Cypress components.

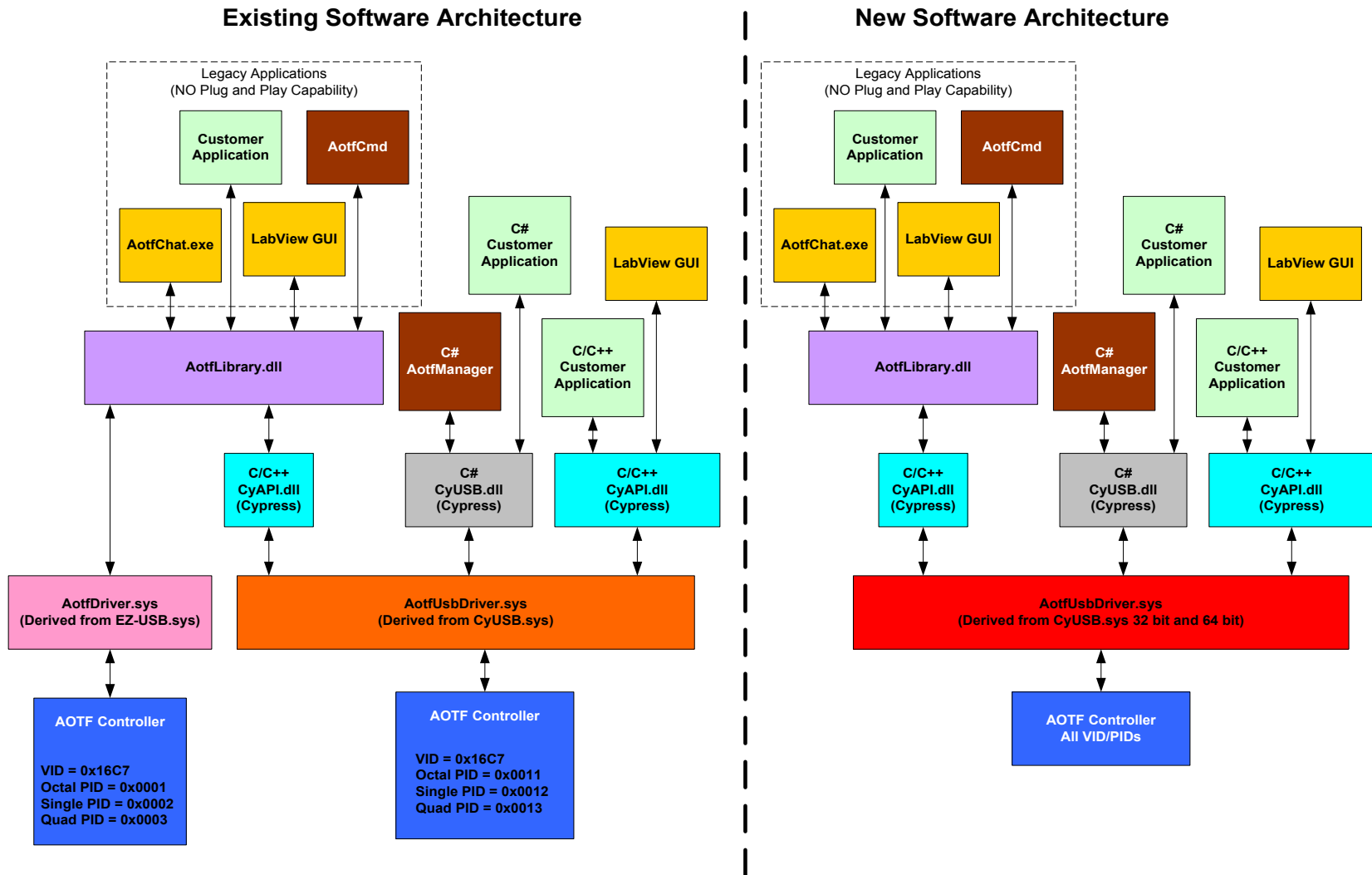


Figure 1: AotfUsbDriver and AotfLibrary

## **4. Cypress**

Cypress Semiconductor is the manufacturer of the microcontroller used in the AOTF Controllers. Cypress has spent considerable time and effort to provide full USB support for their microcontrollers and the products that utilize them. Developers can utilize the extensive amount of development that has been provided by Cypress by using the Cypress libraries to interface with AOTF Controllers. Cypress provides three mechanisms for developers to choose from:

- 1) Microsoft .NET Framework (CyUSB.dll)
- 2) C++ (CyAPI.lib)
- 3) Device Driver (CyUSB.sys)

Each of these interface mechanism is explained in more detail below.

### **4.1. Microsoft .NET Framework (CyUSB.dll)**

For Microsoft .NET Framework developers the CyUSB.dll provides managed code access to the USB interface of AOTF Controllers. This is the easiest way to get C# and VB applications functioning with AOTF Controllers. CyUSB.dll and accompanying documentation file, CyUSB.NET.chm, are installed with the AOTF Utilities Release.

### **4.2. C++ (CyAPI.lib)**

For C++ and other languages the CyAPI.lib and CyAPI.h provide C++ classes for access to the USB interface of AOTF Controllers. This is the easiest way to get applications written in C++, MFC and similar object oriented environments functioning with AOTF Controllers. CyAPI.lib, CyAPI.h, and the accompanying documentation file, CyAPI.chm, are installed with the AOTF Utilities Release.

### 4.3. *Device Driver*

For all other environments, direct access to the USB device driver is provided by one of the following device drivers, which have been optimized for each of the OS environments:

Device Drivers	
Name	OS Platform
<b>AotfUsbDriverW2K.sys</b>	<b>Windows 2000, 32 bit</b>
<b>AotfUsbDriverWXPx32.sys</b>	<b>Windows XP, 32 bit</b>
<b>AotfUsbDriverWXPx64.sys</b>	<b>Windows XP, 64 bit</b>
<b>AotfUsbDriverWVx32.sys</b>	<b>Windows Vista, 32 bit</b>
<b>AotfUsbDriverWVx64.sys</b>	<b>Windows Vista, 64 bit</b>
<b>AotfUsbDriverW7x32.sys</b>	<b>Windows 7, 32 bit</b>
<b>AotfUsbDriverW7x64.sys</b>	<b>Windows 7, 64 bit</b>

This mechanism provides developers with direct interface to the AOTF Controllers via IOCTLs that provide the most flexibility for custom interface requirements. CyUSB.sys and the accompanying documentation file, CyUSB.chm, are installed with the AOTF Utilities Release.

## 5. Release Notes 2009/01/05

This release of the AOTF Utilities includes a new setup utility that will easily install and configure the software. To install the software execute the Setup.exe command and follow the instructions. These utilities and directories are installed by the Setup utility:

- **AotfChat**

AotfChat is the legacy version of the utility for communicating with AOTF Controllers. AotfChat will soon be deprecated. It is included for customers that are not yet ready to migrate their applications and software over to the new Plug and Play device driver architecture. The replacement utility is AotfManager.

AotfChat was the first application developed for communicating with AOTF Controllers. It was sufficient at the time of development, but lacked more sophisticated capabilities, like Plug and Play, file drag and drop, and simultaneous communication to multiple AOTF Controllers. AotfChat relied on the legacy implementation of AotfLibrary, which in turn relied on a custom implementation of AotfDriver.sys, a device driver for USB communication. AotfChat is being deprecated, but it isn't completely dead yet. It will continue to be supported, but additional features and enhancements will be delivered via the AotfManager application. See the AotfManager Users Guide for additional information. Customers may continue to use AotfChat, but they should plan on migrating their applications to take advantage of the new Cypress supplied libraries and device driver. See *Section 6, AotfDriver and AotfLibrary, on page 18* for additional information about migrating applications.

- **AotfCmd**

AotfCmd is a new utility for communicating with AOTF Controllers. AotfCmd is a console application that can be used in a CMD window and within batch (BAT) files. AotfCmd is registered with Microsoft Windows as the application to execute when files with the “.aotf” extension are executed. AotfCmd sends the commands in aotf files to the AOTF Controller for execution. See the AotfCmd Users Guide for additional information.

- **AotfDriver**

AotfDriver is one of the major changes in this release. The major advantage introduced into this version of AotfDriver is the improved support for Plug and Play (PnP) capability. See *Section 6, AotfDriver and AotfLibrary, on page 18* for more details about AotfDriver.

- **AotfLibrary**

AotfLibrary has been enhanced to support Plug and Play capability. One of the goals of this release of AotfLibrary is to maintain backward compatibility with legacy applications like AotfChat, and existing customer applications that use AotfLibrary. But at the same time provide the ability for applications to migrate toward the new AotfDriver with support for an improved C++ and .NET Framework (C#, Visual Basic). See *Section 6, AotfDriver and AotfLibrary, on page 18* for more details about AotfLibrary.

- **AotfManager**

AotfManager is the replacement for the AotfChat. AotfManager has many new features, such as the simultaneous communication with multiple AOTF Controllers, command history, drag and drop capability for script files, and PnP support. See the AotfManager Users Guide for additional information.



- **Documentation**  
The Documentation folder contains all of the latest documentation for the AOTF Controllers.
- **Cypress**  
The Cypress directory is where software developers will find all of the necessary documentation for the libraries, DLLs, and device drivers that facilitate integration of the AOTF Controllers into custom applications. See *Section 7, Cypress, on page 21* for more details about the Cypress supplied utilities.
- **AODS Controller**  
AODS Controller is a LabView application that provides a Graphical User Interface (GUI) for the AOTF Controllers.
- **LabView**  
The LabView directory contains the Run Time Engine (RTE) for LabView applications. LabView RTE is required by the AODS Controller application.

## 6. AotfDriver and AotfLibrary

AotfDriver and AotfLibrary were the cornerstone of the initial AOTF Controller software. They created the foundation on which AotfChat, LabView applications and customer created custom applications were built. But in the high tech world of ever changing requirements and platforms, there comes a time when new capabilities and feature requests demand a change. In this case the need for a better Plug and Play architecture was required. The solution involves a change in the fundamental way that the AOTF Utilities software communicates with USB.

We recognize that all customers may not be ready to migrate to a new architecture, or that they may have systems already deployed that use the old architecture, and can not retrofit a new architecture into those already deployed systems. They may have service contracts and maintenance agreements in place that require them to continue supporting the current environment.

Our solution to the problem is to provide a way for customers to ease into the new architecture by continuing to support the existing applications, while at the same time providing a migration path for moving forward. We will continue to support AotfChat, AotfLibrary, and AotfDriver while providing new utilities like AotfCmd and AotfManager that provide superior capabilities and take advantage of the new architecture.

Improved Plug and Play support is only available with the new architecture. Environments that require Plug and Play support have no alternative except to migrate to the new architecture. The advantages, besides the Plug and Play capability, are:

- Improved documentation for all levels of developer effort, including Microsoft's' .NET Framework, C++, Visual Basic, Labview, and direct device driver interaction via IOCTLs.
- Improved developer support, and additional utilities for integrating AOTF communication into customer applications, such as AotfCmd and AotfManager.

*Figure 2 on page 20* is divided into left and right halves. The left half shows the existing (legacy) software architecture, with AotfChat, AotfLibrary, AotfDriver, and custom applications. The legacy architecture diagram shows how the various components interact with each other and how they form a layered architecture for AOTF Controller integration. The right half shows the new contemporary architecture, with similar components, but with many additional features and capabilities.

The fundamental concept that allows customers to decide how and when they choose to migrate to the new contemporary architecture is rooted at the bottom of the diagrams in the AOTF Controllers. All USB devices, including the AOTF Controllers contain a Vendor ID (VID) and Product ID (PID) that are used by the OS to uniquely identify the USB device. The Vendor ID is a unique number that belongs to the manufacturer of the USB device. In the case of AOTF Controllers the VID number is 5831 (0x16C7). The USB Special Interest Group (SIG) is responsible for administering VID numbers. The PID number is administered by the manufacturer. For AOTF Controllers, the PID numbers are:

- Octal Channel AOTF Controllers have a PID of 1 (0x0001).
- Single Channel AOTF Controllers have a PID of 2 (0x0002).
- Quad Channel AOTF Controllers have a PID of 3 (0x0003).

With this release of the firmware, AOTF Controllers have a new firmware command that allows them to change their PID (see *Section 8, Firmware Releases, on page 22* for additional information about the firmware commands). New PID numbers have been assigned to the AOTF Controllers so that customers

can change to the new PID numbers when they are ready to migrate to the new contemporary architecture. During the development phase, while customers are getting to launch new applications and/or systems that will use the new contemporary architecture, developers can toggle the PID between the old PID and the new PID to test and prepare for migration. The new PID numbers for AOTF Controllers are:

- Octal Channel AOTF Controllers have a PID of 17 (0x0011).
- Single Channel AOTF Controllers have a PID of 18 (0x0012).
- Quad Channel AOTF Controllers have a PID of 19 (0x0013).

Changing from the old PID to the new PID is very simple, just execute the command “`config pid new`”. The next time the AOTF Controller is rebooted it will begin using the new PID. If it becomes necessary to switch back to the old PID just issue the command “`config pid old`”.

Referencing *Figure 2* again, near the bottom of the diagram, the new PID allows the OS to associate the AOTF Controller with the new device driver architecture. To facilitate the migration to the new contemporary architecture, a new version of AotfLibrary (highlighted by the large **red arrow** in *Figure 2*) allows legacy applications like AotfChat and existing customer applications to begin working with the new contemporary architecture immediately, without change to the existing application. The new AotfLibrary allows existing applications to coexist with new contemporary applications. This greatly reduces the risk for developers, and speeds their integration and adoption of the new contemporary architecture.

Unfortunately there is no way for existing applications to take full advantage of all of the new features of the contemporary architecture. For example, existing applications that are not Plug and Play capable, will continue not being Plug and Play capable. They will be able to function, but they will not have any additional capabilities that didn't already have. Existing applications use the AotfLibrary, which is a shim that sits on top of the new contemporary architecture to provide backward compatibility to existing applications. To take full advantage of the new contemporary architecture applications need to migrate away from AotfLibrary and begin interfacing directly with one of the Cypress supplied techniques for communicating with USB devices. See *Section 7, Cypress, on page 21* for additional information on the Cypress components.

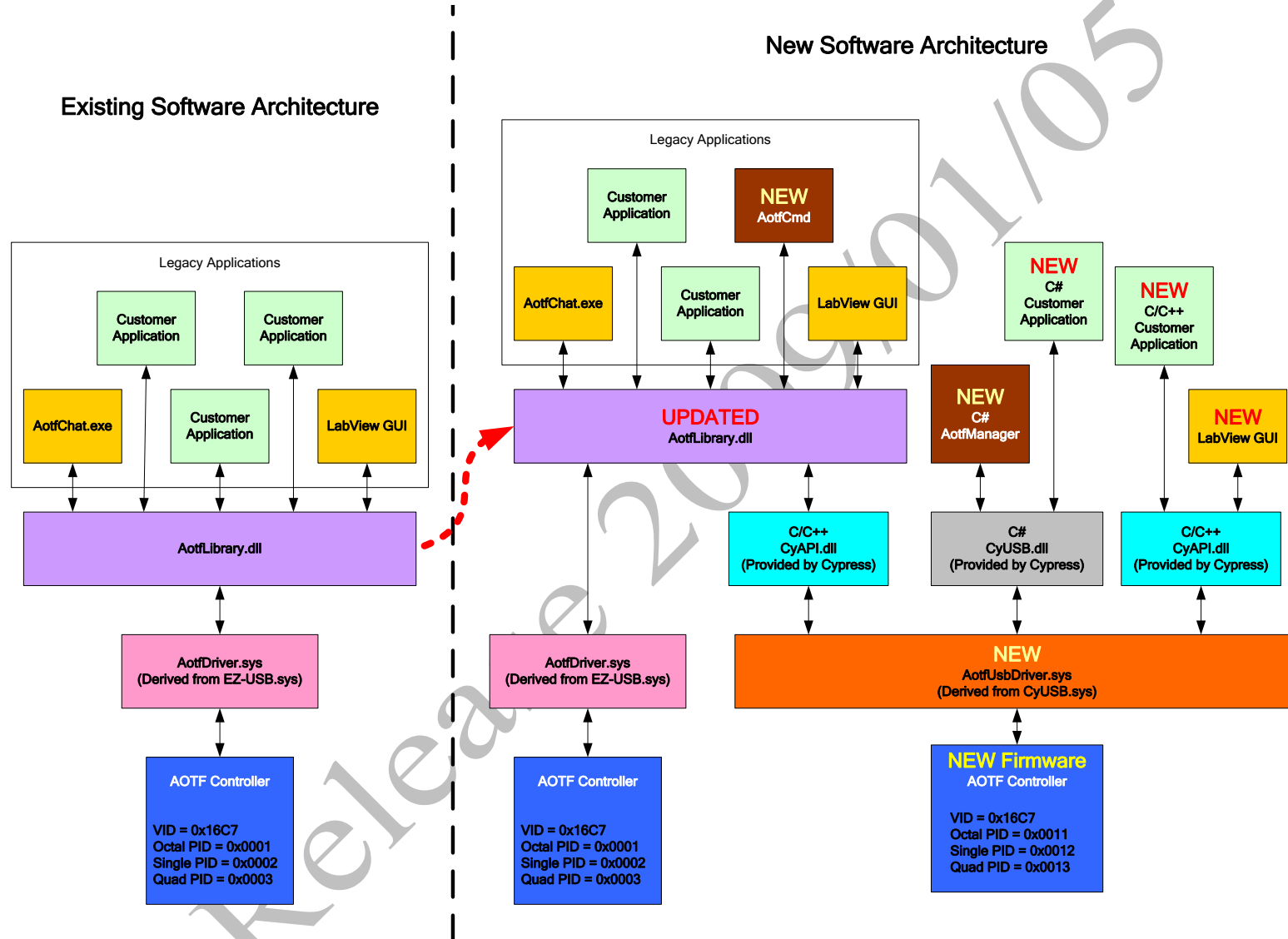


Figure 2: AotfDriver and AotfLibrary

## 7. Cypress

Cypress Semiconductor is the manufacturer of the microcontroller used in the AOTF Controllers. Cypress has spent considerable time and effort to provide full USB support for their microcontrollers and the products that utilize them. Developers can utilize the extensive amount of development that has been provided by Cypress by using the Cypress libraries to interface with AOTF Controllers. Cypress provides three mechanisms for developers to choose from:

- 4) Microsoft .NET Framework (CyUSB.dll)
- 5) C++ (CyAPI.lib)
- 6) Device Driver (CyUSB.sys)

Each of these interface mechanism is explained in more detail below.

### **7.1. Microsoft .NET Framework (CyUSB.dll)**

For Microsoft .NET Framework developers the CyUSB.dll provides managed code access to the USB interface of AOTF Controllers. This is the easiest way to get C# and VB applications functioning with AOTF Controllers. CyUSB.dll and accompanying documentation file, CyUSB.NET.chm, are installed with the AOTF Utilities Release.

### **7.2. C++ (CyAPI.lib)**

For C++ and other languages the CyAPI.lib and CyAPI.h provide C++ classes for access to the USB interface of AOTF Controllers. This is the easiest way to get applications written in C++, MFC and similar object oriented environments functioning with AOTF Controllers. CyAPI.lib, CyAPI.h, and the accompanying documentation file, CyAPI.chm, are installed with the AOTF Utilities Release.

### **7.3. Device Driver (CyUSB.sys)**

For all other environments, direct access to the USB device driver is provided by the CyUSB.sys driver. This mechanism provides developers with direct interface to the AOTF Controllers via IOCTLs that provide the most flexibility for custom interface requirements. CyUSB.sys and the accompanying documentation file, CyUSB.chm, are installed with the AOTF Utilities Release.

## 8. Firmware Releases

The following changes were introduced into the 2009/01 and later versions of the firmware:

- The “`config pid`” command was introduced. The “`config pid`” command allows the USB Product ID (PID) to be changed 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. To support advanced Plug and Play (PnP) capabilities a new driver architecture was introduced.
- 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.
- These commands have been deprecated. The functionality of these commands has been incorporated into the syntax used for specifying a frequency. These commands are still supported, but they will eventually be removed. Use the alternative command to accomplish the same function.
  - “`dds ftw`” was used to specify the frequency using a Frequency Tuning Word (FTW). Use the “`dds frequency`” command as a replacement. The “`dds frequency`” command now supports the more flexible frequency syntax common to all commands (See below).
  - “`dds wavelength`” was used to specify the frequency using the wavelength of laser light that should be diffracted. Use the “`dds frequency`” command as a replacement. The “`dds frequency`” command now supports the more flexible frequency syntax common to all commands (See below).
  - “`chirp ftw`” was used to specify the frequency using a Frequency Tuning Word (FTW). Use the “`chirp frequency`” command as a replacement. The “`chirp frequency`” command now supports the more flexible frequency syntax common to all commands (See below).
- These commands have been deprecated. The functionality of these commands has been incorporated into other commands. These commands are still supported, but they will eventually be removed. Use the alternative command to accomplish the same function.
  - “`daughter dac`” was used to manipulate the DAC chips on the digital daughter cards. The identical functionality is available with the “`mod dac`” command.
  - “`daughter fsk`” was used to manipulate the FSK signal. The identical functionality is available with the “`mod fsk`” command.
  - “`daughter blank`” was used to manipulate the BLANK signal. The identical functionality is available with the “`mod blank`” command.
  - “`daughter gain`” was used to manipulate the gain of the analog modulation. The identical functionality is available with the “`mod gain`” command.

- The command syntax for specifying frequencies has been enhanced. 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 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 # at the beginning of a number specifies the frequency in light wavelength. The wavelength is expressed in nano-meters (nm). 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.