MaxIm CCD Camera/Filter Wheel Plug-In Driver

This document is copyright © 1999-2004 Diffraction Limited.

Use of The Plug-In Software

This source code is licensed for use in developing a plug-in DLL for controlling CCD cameras under MaxIm CCD only. This plug-in source code file shall not be distributed in whole or in part without the express written permission of Diffraction Limited. Executable DLL files built using this source code may be freely distributed. Please see the full terms of the license agreement included below.

Technical Notes

This file is designed for Visual C++ 5.0 or higher; modifications may be required for other compilers. MaxIm CCD will try to link to the DLL on startup if its name is CCDPlug*.dll (where * is any valid file name string) and the DLL has the correct two entry points.

This is an object-oriented interface, so the camera is defined as an object. If you are a "C" programmer, you do not require a detailed understanding of C++ classes to implement a driver; you can simply follow the example code.

This MaxImCCDPlugIn.h file defines an abstract class. All member functions must be overridden, even if they are not being used. The two DLL entry points are not part of the class; however they use C++ calling conventions, including name mangling.

The DLL is responsible for allocating the image buffer and all other variables when OpenCamera is called, and deallocating all variables when CloseCamera is called. Failure to properly allocate/deallocate memory may cause memory corruption and crash the application.

Source code for a sample "DummyCCD" DLL is included. This DLL will be detected by MaxIm CCD and will imitate a camera with filter wheel. DummyCCD is an excellent starting point for your driver; just modify the code to add your camera interface.

The DLL does not have to be linked to MFC, but it may be. If a function from MFC is required, please use static linking.

Description of Methods

The Methods are simply functions that are contained in the CMaxImCCDPlugIn class. All methods must be defined even if they are not being used. The interface can be used to write drivers for cameras or filter wheels. In some cases filter wheels are controlled through a camera, in other cases they use a standalone interface.

GetParameters

The GetParameters method controls the Setup dialog box and determines the capabilities of the driver. Set HasCamera to true if the driver can control a camera, HasFilterWheel to true if the driver can control a filter wheel, and HasGuiderRelays to true if the driver can control a camera and the camera has autoguider output relays. MaxIm CCD calls this method when the program is first started to determine the capabilities of the driver. This determines whether the driver appears in the CCD, Filter, and Guider fields on the Setup tab.

The DialogContents structure is used to initialize the camera and filter wheel setup dialog boxes. Do not exceed the defined maximum string lengths. The fields are defined as follows:

- CameraName should contain a short name for the camera which will appear in the Setup tab.
- FilterName should contain a short name for the filter wheel which will appear in the Setup tab.
- Copyright should contain up to three lines of text. Normally, the copyright field includes your copyright notice, version number, and information on who the customer contact for technical support when using the driver (that should be you, not us!).
- UseFilePath should be set to true if your driver uses some kind of initialization (.ini) file. The dialog box will prompt to supply a path for this file.

- NumParameters tells how many dialog box parameters should appear on the camera setup dialog. The maximum number is five.
- Parameters contains setup information for each dialog box field. Each field has a ParamVals record which contains:
  - ParamName, the name of the parameter to appear above the field on the dialog box
  - NumOptions, the number of items in the drop-down combo box
  - Options, which contains a set of OptionVal records. Each record contains a Display value which appears in the drop-down combo, and Value which is sent to the driver when OpenCamera is called. This is how the user specifies options such as port addresses, etc.
- NumFilters, the number of valid filter positions for the filter wheel
- NumFilterParameters, the number of parameters used in the Setup Filter Wheel dialog box. Up to three parameters are available. These can be used to set port addresses, etc.
- FilterParameters, a structure similar to Parameters used for the filter wheel.

Note that if the driver does not support a filter wheel, none of the filter wheel settings are used. If the driver does not support a camera (filter wheel only), none of the camera settings are used. Also note that all methods (functions) must be present even if they are not used (they can be empty).

OpenCamera

This function is called when OK is clicked in the camera setup dialog, or when the Restart button is clicked. If UseFileName was turned on, then FilePath contains the path to the initilization file. The Param structure contains the other selections made in the dialog box.

OpenCamera should allocate a memory buffer for data coming from the camera. Normally this is an array of unsigned short [XSize * YSize]. Any other required memory should be allocated at this time.

CloseCamera

CloseCamera is called when the user clicks the Shutdown button or changes the camera selection. Any memory allocated previously must be deleted when this function is called.

GetArraySize

This function is called when MaxIm CCD wants to know how big the CCD chip is. Return the full unbinned dimensions of the array.

GetImageSize

This function is called when MaxIm CCD wants to know how big the previously-exposed CCD image is. Return the X and Y dimensions of the image, plus the X offset and Y offset from the edge of the chip. All dimensions must *include* binning.

GetPixelAspect

This function returns the pixel dimensions in micron units. If IncludeBinning is true, return the size in accordance with the binning mode selected for the last exposure. If IncludeBinning is false, return the size of pixel when binning is set to 1.

GetImageBuffer

This function returns a pointer to the image buffer containing the exposed image. If subframes or binning are used, the data will be shorter than the length of the buffer. The buffer should only be allocated in OpenCamera and deallocated in CloseCamera.

StartExposure

This starts an exposure. The following parameters are provided:

- Exposure, the exposure duration in 1/100ths of a second.
- XStart, the subframe X offset (from the left, includes binning)

- YStart, the subframe Y offset (from the top, includes binning)
- NumX, the number of pixels across the subframe (includes binning)
- NumY, the number of pixels down the subframe (includes binning)
- BinX, the X binning factor
- BinY, the Y binning factor (ignore if the camera can only bin in 1:1 ratios)
- Light, true if the shutter should be open during the exposure, false if the shutter should be closed
- FastReadout is true when the exposure was taken from the Focus tab; optionally use this to select a faster readout mode (e.g. turn off double-correlated sampler or reduce integration time).
- HoldShutter is true when the exposure is being taken for guiding; optionally use this to hold the shutter open during readout to avoid wearing out the shutter.

AbortExposure

When called, the driver should stop the current exposure and reset the camera. Note: if the camera has a latency time after an abort before it can take another exposure, GetCameraState should return CS_FLUSHING until the camera is ready for the next exposure.

TransferImage

TransferImage is used to download an image from the camera into the driver's buffer. The operation can all occur in one step, in which case this routine only needs to exit when readout is done. However, if the readout takes some time, then it is recommended to return from this routine periodically. Return a value in PercentDone (0 to 100) to indicate progress; this is displayed to the user and then the routine is immediately called again. Set TransferDone to true once the download is complete.

We recommend that, for best performance, this routine *not* return after every readout line. Returning more than 10 times per readout is probably excessive.

SetTemperature

This routine is used to set the temperature of the CCD Peltier cooler. ControlOn turns on the cooler regulation; if false the cooler should be turned off immediately (a warning message is displayed to the user before allowing this to occur). GoToAmbient is set true when the user has requested a gradual cool-down. Temp indicates the desired temperature setpoint when ControlOn is true and GoToAmbient is false.

GetTemperature

This method is called periodically to get the temperature of the CCD chip. State can be set as follows:

- COOL_OFF – cooler is currently switched off
- COOL_ON – cooler is on
- COOL_ATAMBIENT – cooler is holding the CCD at ambient temperature; warm-up cycle complete
- COOL_GOTOAMBIENT – cooler is driving the CCD towards ambient, but has not finished the warm-up cycle
- COOL_NOCONTROL – indicates that this camera does not have temperature control capability or the temperature is set externally
- COOL_INITIALIZING – return when information on the cooler is not available yet or the functions are still being initialized. Displays "Please Wait" to the user.

GetCameraStats

This method returns various information on the camera capabilities, as follows:

- int &XSize, number of rows in physical CCD array (no binning)
- int &YSize, number of columns in physical CCD array (no binning)

- int &MaxBinFactorX, maximum binning factor allowed on X axis
- int &MaxBinFactorY, maximum binning factor allowed on Y axis
- bool &DifferentAxes, return true if X and Y binning factors can be different
- bool &HasShutter, return true if camera has a shutter
- bool &HasCoolerControl, return true if camera has a controllable cooler setpoint
- bool &PowerOfTwoBinning, return true if user can select 1/2/4/8/16 binning only
- bool &HasGuiderRelays, return true if camera has built-in autoguider output relays

GetCameraState

This routine returns the current status of the camera in State, and the current shutter position in ShutterOpen. The possible camera states are as follows:

- CCD_ERROR, camera is not available because of an error condition
- CCD_FILTERWHEELMOVING, camera is waiting for the filter wheel to finish moving
- CCD_FLUSHING, flushing CCD chip or camera otherwise busy; can't start an exposure yet
- CCD_WAITTRIGGER, waiting for an external trigger event
- CCD_DOWNLOADING, currently downloading the image from camera hardware to the PC
- CCD_READING, the camera hardware is currently reading out the CCD chip (prior to download)
- CCD_EXPOSING, exposing dark or light frame
- CCD_IDLE, camera is idle; available for an exposure

ActivateRelay

This routine is only used if HasGuiderRelays was true and the camera is initialized as an autoguider.

If the camera is autoguider-capable (HasRelays was set), then this method is called whenever the autoguider state machine wants to send a correction to the camera. The directions are arbitrary; the standard autoguider calibration procedure is used to determine the correct commands to be sent. However, it is recommended that the X axis be Right Ascension and the Y axis be Declination.

Only one direction (X or Y) will be commanded at a time. The number may be positive or negative, depending on which direction the telescope should be moved. The absolute value of the number indicates how long to close the relay in 1/100ths of a second.

IsRelayDone

This routine is only used if HasGuiderRelays was true and the camera is initialized as an autoguider.

This routine is called periodically while the software is waiting for the relay command to be completed. Return true when the relay has been released and the guiding cycle may continue.

OpenFilterWheel

This routine is used to open a filter wheel, and is called when OK is clicked in the Setup Filter Wheel dialog or Restart is clicked. The filter wheel may be associated with a camera operated by the same driver, or may be entirely stand-alone. When Restart is clicked the camera is always set up first. Param contains the selections made by the user in the associated dialog box.

CloseFilterWheel

This routine is called when the Shutdown button is clicked.

SetFilterWheel

This routine is called when the filter wheel is to be rotated to a new position. The position is passed in as an integer.

IsFilterWheelMoving

This routine should return true if the filter wheel is moving; false if not. This delays the start of the exposure until the filter wheel cycle has completed. This may be in response to a signal from the hardware or just by checking the PC's real-time clock.

Debugging in Visual C++

The driver can be debugged using the source-level debugger built in to Microsoft Developer Studio. Compile the DLL in Debug mode, and click the Run button. You will be prompted for an executable program, browse to the MaxIm DL program directory and select MaxIm_DL.exe. MaxIm DL will start, and you can open the MaxIm CCD window and initialize the camera driver. You can then set breakpoints inside the driver, single step, view variables, etc.

Plug-In Driver License Agreement

IMPORTANT: This agreement (the Agreement) is a legal agreement between you and Diffraction Limited for this software product, including all source code and documentation (the Product). By installing, copying, or otherwise using the Product, you agree to be bound by the terms of this Agreement. If you do not agree to be bound by the terms of this Agreement, immediately return the unused Product to the vendor for a full refund.

PRODUCT LICENSE

This Product is licensed, not sold. The Product is the property of Diffraction Limited or its licensers and is protected by copyright law. This Agreement grants you the following rights:

1. You may install and use one copy of the Product on a single computer;
2. You may make one copy of the Product for archival purposes;
3. You may use the Product on a network provided that you have a licensed copy of the Product for each computer that can access the Product over that network;
4. You may transfer the Product on a permanent basis to another person or entity, provided that you retain no copies of the Product and the recipient agrees in writing to all terms of this Agreement.
5. You may distribute copies of "DLL" executable code created using this product for use only with licensed copies of MaxIm DL and MaxIm CCD.

You may not:

1. Sublicense, share, rent or lease all or any portion of the Product;
2. Reverse engineer, decompile, disassemble, modify, translate, or create derivative works from the Product;
3. Use a previous version of the Product if you have received a replacement disk set or upgrade version as a replacement for the previous version.
4. Transmit Diffraction Limited's proprietary information to a third party in any form, including but not limited to the contents of the file MaxImCCDPlugIn.h, the Dummy CCD driver example code, and all associated documentation.

TERMINATION

If you fail to comply with the terms of this Agreement, Diffraction Limited may terminate the Agreement at its discretion. Upon termination, you must return or destroy all copies of the Product including all of its component parts. This does not restrict the rights of Diffraction Limited to other remedies.

JURISDICTION

This Agreement is governed by the laws of the Province of Ontario, Canada. Each of the parties hereto irrevocably agrees to the jurisdiction of the courts of the Province of Ontario.

LIMITED WARRANTY

Diffraction Limited warranties that the media upon which the Product is distributed will be free of defects for a period of thirty (30) days from the date the Product is delivered to you. In the event of a breach of this warranty Diffraction Limited will, at the discretion of Diffraction Limited, either replace the defective media or accept the return of the Product accompanied by proof of purchase for a full refund. Diffraction Limited does not warrant that the software in this Product will be defect-free or meet your requirements.

THIS WARRANTY IS EXCLUSIVE AND IN LIEU OF ALL OTHER WARRANTIES, WHETHER EXPRESS OR IMPLIED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR USE, AND NON-INFRINGEMENT. This warranty gives you specific rights. You may have other rights which vary from jurisdiction to jurisdiction.

DISCLAIMER

IN NO EVENT SHALL DIFFRACTION LIMITED BE LIABLE TO YOU FOR ANY SPECIAL, CONSEQUENTIAL, INDIRECT, OR SIMILAR DAMAGES, INCLUDING ANY LOST PROFITS OR LOST DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS PRODUCT EVEN IF DIFFRACTION LIMITED HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME JURISDICTIONS DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES SO THE ABOVE LIMITATION MAY NOT APPLY TO YOU. IN NO CASE SHALL DIFFRACTION LIMITED'S LIABILITY EXCEED THE PURCHASE PRICE FOR THE PRODUCT.

MISCELLANEOUS

Should you wish to contact Diffraction Limited, you may do so at 25 Conover Street, Unit 106, Ottawa, Ontario, Canada, K2G 4C3, telephone (613) 225-2732, fax (613) 225-9688.