

In order to multiple readout mode , like the HDR mode, and the hardware Binning Mode. From the Version 2019.2.15 and later, The QHYCCD add the ReadMode APIs. Because the mode change will change the whole define of the basic image X, Y. And the memory allocate. To compatible with all exist software which use the QHYCCD SDK. We add this APIs.

Examples:

QHY42PRO, it support both HDR and STD mode. In HDR mode , it outputs 4096*2048 resolution image. Which consist two images. Left is low gain channel image and right side is high gain channel image. In STD mode, it outputs the 2048*2048 resolution images.

The suggest call sequence is :

InitQHYCCDResource()

ScanQHYCCD()

GetQHYCCDId

OpenQHYCCD

GetQHYCCDNumberOfReadModes

GetQHYCCDReadModeName (check each name of the mode)

In host software. list all the model and allow user to select one

SetQHYCCDReadMode

SetQHYCCDStreamingMode

InitQHYCCD

SetQHYCCDBinMode

SetQHYCCDResolution

BeginExposure (Single/Live)

GetFrame(single/live)

...

...

...

Note *1: Since there is the roi , binning etc. The actual image size is still return from the GetQHYCCDSingleFrame function or GetQHYCCDLiveFrame function. Other function let getQHYCCDChipInfo, GetQHYCCDReadoutResolution is just the basic x,y , not the final X,Y

Note *2: If the host software plan to have compatible with the old QHYCCD APIs. You can use the error catch for calling the following APIs. Since the old QHYCCD SDK do not have these APIs. If you call it without the error catch, it will cause the software crash.

Note *3: If does not call these APIs. The SDK will works with default mode . So if the host software do not call these function. It is no matter. So that the old software still compatible with this new QHYCCD SDK.

Note *4: if you want to get to know the readout mode first and do not want to do next steps, you can use this method.

InitQHYCCDResource()

ScanQHYCCD()

GetQHYCCDId

OpenQHYCCD

GetQHYCCDNumberOfReadModes

GetQHYCCDReadModeName (check each name of the mode)

In host software. list all the model and allow user to select one

CloseQHYCCD

And then later

InitQHYCCDResource()

ScanQHYCCD()

GetQHYCCDId

OpenQHYCCD

SetQHYCCDReadMode

CloseQHYCCD

The ReadMode APIs

```
EXPORTC uint32_t STDCALL GetQHYCCDNumberOfReadModes(qhyccd_handle
*h,uint32_t *numModes);// Get the maximum resolution for a read modeEXPORTC
uint32_t STDCALL GetQHYCCDReadModeResolution(qhyccd_handle *h,uint32_t
modeNumber, uint32_t* width, uint32_t* height);// Get the name of a read
modeEXPORTC uint32_t STDCALL GetQHYCCDReadModeName(qhyccd_handle
```

```

*h,uint32_t modeNumber, char* name);// Set the read modeEXPORTC uint32_t
STD_CALL SetQHYCCDReadMode(qhyccd_handle *h,uint32_t modeNumber);// Get
the read modeEXPORTC uint32_t STD_CALL GetQHYCCDReadMode(qhyccd_handle
*h,uint32_t* modeNumber);

```

```

The C# Wrapper of these APIs[DllImport( "qhyccd.dll", EntryPoint =
"GetQHYCCDNumberOfReadModes", CharSet = CharSet.Ansi,
CallingConvention = CallingConvention.StdCall )] public unsafe static extern
UInt32 GetQHYCCDNumberOfReadModes( IntPtr handle, ref UInt32 numModes );
[DllImport( "qhyccd.dll", EntryPoint = "GetQHYCCDReadModeResolution",
CharSet = CharSet.Ansi, CallingConvention = CallingConvention.StdCall )]
public unsafe static extern UInt32 GetQHYCCDReadModeResolution( IntPtr handle,
UInt32 modeNumber, ref UInt32 width,ref UInt32 height );
[DllImport( "qhyccd.dll", EntryPoint = "GetQHYCCDReadModeName", CharSet =
CharSet.Ansi, CallingConvention = CallingConvention.StdCall )] public unsafe
static extern UInt32 GetQHYCCDReadModeName( IntPtr handle, UInt32
modeNumber,byte [] name ); [DllImport( "qhyccd.dll", EntryPoint =
"SetQHYCCDReadMode", CharSet = CharSet.Ansi, CallingConvention =
CallingConvention.StdCall )] public unsafe static extern UInt32
SetQHYCCDReadMode( IntPtr handle, UInt32 modeNumber );
[DllImport( "qhyccd.dll", EntryPoint = "GetQHYCCDReadMode", CharSet =
CharSet.Ansi, CallingConvention = CallingConvention.StdCall )] public unsafe
static extern UInt32 GetQHYCCDReadMode( IntPtr handle, ref UInt32 modeNumber );

```