

Dell SDK for Monitors

Application Programming Interface Guide

for SDK version 1.3



Information in this document is subject to change without notice.

© 2016 Dell Inc. All rights reserved.

Reproduction of these materials in any manner whatsoever without the written permission of Dell Inc. is strictly forbidden.

Trademarks used in this text: Dell™, the DELL logo, and UltraShrap™ are trademarks of Dell Inc.; Microsoft®, Windows®, and the Windows start button logo are either trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries;

Other trademarks and trade names may be used in this document to refer to either the entities claiming the marks and names or their products. Dell Inc. disclaims any proprietary interest in trademarks and trade names other than its own.

Contents

Introduction	6
Monitor Management	7
GetAvailableMonitors	7
ConnectMonitor	7
DisconnectMonitor	8
SetAssetTag	8
GetAssetTag	9
GetMonitorName	9
GetMonitorSerialNumber	10
GetBacklightHours	10
Power Management	11
GetPowerState	11
SetPowerState	12
GetPowerLED	12
SetPowerLED	13
GetPowerUSB	13
SetPowerUSB	14
Reset Power	14
Image Management	15
GetBrightness	15
SetBrightness	16
GetContrast	16
SetContrast	17
GetDynamicContrast	17
SetDynamicContrast	18
GetAspectRatio	18
SetAspectRatio	19
GetSharpness	20
SetSharpness	20
GetResponseTime	21
SetResponseTime	21

Color Management	22
GetSaturation	22
SetSaturation	23
GetHue	23
SetHue	24
GetColorTempCaps	25
GetColorTemp	25
SetColorTemp	26
GetColorSpaceCaps	27
GetColorSpaceState	27
SetColorSpaceState	28
GetInputColorFormat	29
SetInputColorFormat	29
GetColorPresetCaps	30
GetColorPreset	31
SetColorPreset	32
GetCustomColor	33
SetCustomColor	34
GetGammaMode	35
SetGammaMode	35
GetUniformityCompensation	36
SetUniformityCompensation	36
ResetColor	37
LUT Management	38
GetCalibrationHours	38
GetCalBrightness	38
SetCalBrightness	39
SetColorControl	40
SetLUT	40
Video Input Management	42
GetAutoSelect	42
SetAutoSelect	42
GetVideoInputCaps	43
GetVideoInput	44
SetVideoInput	45

PIP/PBP Management	46
GetPxPMode	46
SetPxPMode	47
GetPxPSubInput	47
SetPxPSubInput	48
GetPxPLocation	49
SetPxPLocation	50
OSD	51
GetOSDTransparency	.51
SetOSDTransparency	.51
GetOSDLanguage	52
SetOSDLanguage	53
GetOSDRotation	54
SetOSDRotation	54
GetOSDTimer	55
SetOSDTimer	55
GetOSDButtonLock	56
SetOSDButtonLock	56
ResetOSD	57
System Management	58
GetVersionFirmware	58
GetVersionSDK	58
GetMST	59
SetMST	59
GetLCDConditioning	60
SetLCDConditioning	60
FactoryReset	61
SetDebugLevel	61

Introduction

This document describes the APIs for supported Dell UltraSharp monitors on Linux(x86), OSX and Windows platforms. These APIs are to be used for remote display management and control from a Host PC to supported Dell UltraSharp monitors via a USB connection. A USB 3.0 A to B cable should be used for the connection between the host and the display.

The following monitors are supported:

1. UP3017

The API described in this document corresponds to SDK version 1.3.

Monitor Management

GetAvailableMonitors

Returns the number of supported monitor(s)

API	
MONITOR_CODE GetAvailableMonitors(BYTE *pbCount)	
Params	
byMonitors[]	Array for return values
Return	
MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure
pbCount	Number of supported monitors connected

ConnectMonitor

Connect to monitor and start session

API	
MONITOR_CODE ConnectMonitor(BYTE iID)	
Params	
iID	Index of monitor as returned by GetMonitorCaps to connect to. Index starts at 0 for the first monitor.
Return	
MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

DisconnectMonitor

Disconnect to monitor and end session

API

MONITOR_CODE DisconnectMonitor(void)

Params

-

Return

MONITOR_CODE Code describing the result of the API call

0 Success

1 Timeout

2 Parameters Error

3 Not connected

-1 Other Failure

SetAssetTag

Set the asset tag of the monitor.

API

MONITOR_CODE SetAssetTag(BYTE *pbyAssetTag)

Params

*pbyAssetTag Pointer to asset tag ID string (max 10 chars)

Return

MONITOR_CODE Code describing the result of the API call

0 Success

1 Timeout

2 Parameters Error

3 Not connected

-1 Other Failure

GetAssetTag

Returns the monitor asset tag. Asset Tag will be empty until set by SetAssetTag.

API	
MONITOR_CODE GetAssetTag(BYTE *pbyAssetTag)	
Params	
*pbyAssetTag	Pointer to return asset tag ID string
Return	
MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure
pbyAssetTag	Asset tag ID string (max 10 chars)

GetMonitorName

Returns the monitor name

API	
MONITOR_CODE GetMonitorName(BYTE *pbyMonitorName)	
Params	
*pbyMonitorName	Pointer to return monitor name
Return	
MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure
pbyMonitorName	Monitor name string (max 10 chars)

GetMonitorSerialNumber

Returns the monitor serial number

API	
MONITOR_CODE GetMonitorSerialNumber(BYTE *pbySerialNumber)	
Params	
*pbySerialNumber	Pointer to return monitor serial number
Return	
MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure
pbySerialNumber	Monitor serial number string (max 12 chars)

GetBacklightHours

Returns the monitor backlight hours

API	
MONITOR_CODE GetBacklightHours(SWORD16 *ps16Val)	
Params	
*ps16Val	Pointer to return monitor backlight hours
Return	
MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure
ps16Val	Monitor backlight hours

Power Management

GetPowerState

Returns the current power state of the monitor

API

MONITOR_CODE GetPowerState(UBYTE *pu8Val)

Params

*pu8Val Pointer to return power state

Return

MONITOR_CODE Code describing the result of the API call

0 Success

1 Timeout

2 Parameters Error

3 Not connected

-1 Other Failure

pu8Val Power State

0 Off

1 On

2 Standby

SetPowerState

Set the monitor on or standby

API	
MONITOR_CODE SetPowerState(UBYTE u8Val)	
Params	
u8Val	Power state to set
0	Off
1	On
2	Standby
Return	
MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

GetPowerLED

Returns the power LED setting of the monitor

API	
MONITOR_CODE GetPowerLED(UBYTE *pu8Val)	
Params	
*pu8Val	Pointer to return power LED setting
Return	
MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure
pu8Val	Power LED Setting
0	Off during Active
1	On during Active

SetPowerLED

Set the power LED setting

API	
MONITOR_CODE SetPowerLED(UBYTE u8Val)	
Params	
u8Val	Power LED Setting
0	Off during Active
1	On during Active
Return	
MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

GetPowerUSB

Returns the power USB setting of the monitor

API	
MONITOR_CODE GetPowerUSB(UBYTE *pu8Val)	
Params	
*pu8Val	Pointer to return power USB setting
Return	
MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure
pu8Val	Power USB Setting
0	Off during Standby
1	Off during Standby

SetPowerUSB

Set the power USB setting

API	
MONITOR_CODE SetPowerUSB(UBYTE u8Val)	
Params	
u8Val	Power USB Setting
0	Off during Standby
1	Off during Standby
Return	
MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

Reset Power

Reset power to the monitor

API	
MONITOR_CODE ResetPower(void)	
Params	
-	
Return	
MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

Image Management

GetBrightness

Returns the brightness level of the monitor

API

MONITOR_CODE GetBrightness(UBYTE *pu8Val)

Params

*pu8Val Pointer to return brightness value

Return

MONITOR_CODE Code describing the result of the API call

0 Success

1 Timeout

2 Parameters Error

3 Not connected

-1 Other Failure

pu8Val Brightness value

Integer value 0 (dark) to 100 (bright)

Default 75

Values in increments of 1

SetBrightness

Set the brightness level of the monitor

API	
MONITOR_CODE SetBrightness(UBYTE u8Val)	
Params	
u8Val	Brightness value Integer value 0 (dark) to 100 (bright) Default 75 Values in increments of 1
Return	
MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

GetContrast

Returns the contrast level of the monitor

API	
MONITOR_CODE GetContrast(UBYTE *pu8Val)	
Params	
*pu8Val	Pointer to return contrast value
Return	
MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure
pu8Val	Contrast value Integer value 0 (minimal) to 100 (maximum) Default 75 Values in increments of 1

SetContrast

Set the contrast level of the monitor.

NOTE: Uniformity Compensation must be turned off for this to work.

API	
MONITOR_CODE SetContrast(UBYTE u8Val)	
Params	
u8Val	Contrast value
	Integer value 0 (minimal) to 100 (maximum)
	Default 75
	Values in increments of 1
Return	
MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

GetDynamicContrast

Returns the dynamic contrast setting. Applicable for Movies and Gaming.

NOTE: Only works in Color Preset Game or Movie.

API	
MONITOR_CODE GetDynamicContrast(UBYTE *pu8Val)	
Params	
*pu8Val	Pointer to return dynamic contrast value
Return	
MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

pu8Val	Dynamic Contrast
0	On
1	Off

SetDynamicContrast

Turns on/off the dynamic contrast setting. Applicable for Movies and Gaming.

NOTE: Only works in Color Preset Game or Movie.

API

MONITOR_CODE SetDynamicContrast(UBYTE u8Val)

Params

u8Val	Dynamic Contrast
0	Off
1	On

Return

MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

GetAspectRatio

Returns the aspect ratio

API

MONITOR_CODE GetAspectRatio(UBYTE *pu8Val)

Params

*pu8Val	Pointer to return aspect ratio
---------	--------------------------------

Return

MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

pu8Val	Aspect Ratio	
	0	Wide 16:10
	1	Auto Resize
	2	4:3
	3	1:1

SetAspectRatio

Sets the aspect ratio

API

MONITOR_CODE SetAspectRatio(UBYTE u8Val)

Params

u8Val	Aspect Ratio	
	0	Wide 16:10
	1	Auto Resize
	2	4:3
	3	1:1

Return

MONITOR_CODE	Code describing the result of the API call	
	0	Success
	1	Timeout
	2	Parameters Error
	3	Not connected
	-1	Other Failure

GetSharpness

Returns the sharpness level

API	
MONITOR_CODE GetSharpness(UBYTE *pu8Val)	
Params	
*pu8Val	Pointer to return sharpness value
Return	
MONITOR_CODE	Code describing the result of the API call
	0 Success
	1 Timeout
	2 Parameters Error
	3 Not connected
	-1 Other Failure
pu8Val	Sharpness value
	Integer value 0 to 100
	Default 50
	Values in increments of 10

SetSharpness

Sets the sharpness level

API	
MONITOR_CODE SetSharpness(UBYTE u8Val)	
Params	
u8Val	Sharpness value
	Integer value 0 to 100
	Default 50
	Values in increments of 10
Return	
MONITOR_CODE	Code describing the result of the API call
	0 Success
	1 Timeout
	2 Parameters Error
	3 Not connected
	-1 Other Failure

GetResponseTime

Returns the response time

API

MONITOR_CODE GetResponseTime(UBYTE *pu8Val)

Params

*pu8Val	Pointer to return response time value
---------	---------------------------------------

Return

MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

pu8Val	Response Time
0	Normal
1	Fast

SetResponseTime

Sets the response time

API

MONITOR_CODE SetResponseTime(UBYTE u8Val)

Params

u8Val	Response Time
0	Normal
1	Fast

Return

MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

Color Management

GetSaturation

Returns the color saturation level

NOTE: Only works in Color Preset Game or Movie.

API

MONITOR_CODE GetSaturation(UBYTE *pu8Val)

Params

*pu8Val	Pointer to return color saturation level
---------	--

Return

MONITOR_CODE	Code describing the result of the API call
--------------	--

0	Success
----------	---------

1	Timeout
----------	---------

2	Parameters Error
----------	------------------

3	Not connected
----------	---------------

-1	Other Failure
-----------	---------------

pu8Val	Color Saturation level
--------	------------------------

	Integer value 0 to 100
--	------------------------

	Default 50
--	------------

	Values in increments of 1
--	---------------------------

SetSaturation

Sets the color saturation level

NOTE: Only works in Color Preset Game or Movie.

API	
MONITOR_CODE SetSaturation(UBYTE u8Val)	
Params	
u8Val	Color Saturation level
	Integer value 0 to 100
	Default 50
	Values in increments of 1
Return	
MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

GetHue

Returns the hue level

NOTE: Only works in Color Preset Game or Movie.

API	
MONITOR_CODE GetHue(UBYTE *pu8Val)	
Params	
*pu8Val	Pointer to return hue level
Return	
MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

pu8Val	Color Saturation level
	Integer value 0 to 100
	Default 50
	Values in increments of 1

SetHue

Sets the hue level

NOTE: Only works in Color Preset Game or Movie.

API

MONITOR_CODE SetHue(UBYTE u8Val)

Params

u8Val	Hue level
	Integer value 0 to 100
	Default 50
	Values in increments of 1

Return

MONITOR_CODE	Code describing the result of the API call.
	0 Success
	1 Timeout
	2 Parameters Error
	3 Not connected
	-1 Other Failure

GetColorTempCaps

Returns the supported color temperatures of the monitor

NOTE: Only works in Color Preset Color Temp.

API	
MONITOR_CODE GetColorTempCaps(UWORD32 *pu32Val)	
Params	
*pu32Val	Pointer to return color temperature capabilities.
Return	
MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure
pu32Val	Bitwise representation of color temperature capabilities
0x00000001	5000K
0x00000002	5700K
0x00000004	6500K
0x00000008	7500K
0x00000010	9300K
0x00000020	10000K
For example: 0x00000013 would indicate 5000K, 5700K and 9300K supported	

GetColorTemp

Returns the current color temperature

NOTE: Only works in Color Preset Color Temp.

API	
MONITOR_CODE GetColorTemp(UWORD32 *pu32Val)	
Params	
*pu32Val	Pointer to return color temperature
Return	

MONITOR_CODE	Code describing the result of the API call	
	0	Success
	1	Timeout
	2	Parameters Error
	3	Not connected
	-1	Other Failure
pu32Val	Color temperature	
	0x00000001	5000K
	0x00000002	5700K
	0x00000004	6500K
	0x00000008	7500K
	0x00000010	9300K
	0x00000020	10000K

SetColorTemp

Sets the color temperature

NOTE: Only works in Color Preset Color Temp.

API

MONITOR_CODE SetColorTemp(UWORD32 u32Val)

Params

u32Val	Color temperature	
	0x00000001	5000K
	0x00000002	5700K
	0x00000004	6500K
	0x00000008	7500K
	0x00000010	9300K
	0x00000020	10000K

Return

MONITOR_CODE	Code describing the result of the API call	
	0	Success
	1	Timeout
	2	Parameters Error
	3	Not connected
	-1	Other Failure

GetColorSpaceCaps

Returns the supported color spaces of the monitor

NOTE: *Only works in Color Preset Color Space.*

API

MONITOR_CODE GetColorSpaceCaps(UWORD32 *pu32Val)

Params

*pu32Val Pointer to return color space capabilities

Return

MONITOR_CODE Code describing the result of the API call

0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

pu32Val Bitwise representation of supported color spaces

0x00000001	Adobe RGB
0x00000002	sRGB
0x00000004	Rec 709
0x00000008	DCI-P3
0x00000010	CAL 1
0x00000020	CAL 2

For example:
0x00000013 would indicate Adobe RGB, sRGB and CAL 1 supported

GetColorSpaceState

Returns the current color space state

NOTE: *Only works in Color Preset Color Space.*

API

MONITOR_CODE GetColorSpaceState(UWORD32 *pu32Val)

Params

*pu32Val Pointer to return color space state

Return

MONITOR_CODE	Code describing the result of the API call	
	0	Success
	1	Timeout
	2	Parameters Error
	3	Not connected
	-1	Other Failure
pu32Val	Color space	
	0x00000001	Adobe RGB
	0x00000002	sRGB
	0x00000004	Rec 709
	0x00000008	DCI-P3
	0x00000010	CAL 1
	0x00000020	CAL 2

SetColorSpaceState

Sets the color space state

NOTE: Only works in Color Preset Color Space.

API

MONITOR_CODE SetColorSpaceState(UWORD32 u32Val)

Params

u32Val	Color space	
	0x00000001	Adobe RGB
	0x00000002	sRGB
	0x00000004	Rec 709
	0x00000008	DCI-P3
	0x00000010	CAL 1
	0x00000020	CAL 2

Return

MONITOR_CODE	Code describing the result of the API call	
	0	Success
	1	Timeout
	2	Parameters Error
	3	Not connected
	-1	Other Failure

GetInputColorFormat

Returns the input color format

API	
MONITOR_CODE GetInputColorFormat(UBYTE *pu8Val)	
Params	
*pu8Val	Pointer to return response time value
Return	
MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure
pu8Val	Input Color Format
0	RGB
1	YPbPr

SetInputColorFormat

Sets the input color format

API	
MONITOR_CODE SetInputColorFormat(UBYTE u8Val)	
Params	
u8Val	Input Color Format
0	RGB
1	YPbPr
Return	
MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

GetColorPresetCaps

Returns the available color presets

API	
MONITOR_CODE GetColorPresetCaps(UWORD32 *pu32Val)	
Params	
*pu32Val	Pointer to return color space capabilities
Return	
MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure
pu32Val	Bitwise representation of supported color presets
0x00000001	Standard
0x00000002	Multimedia
0x00000004	Movie
0x00000008	Game
0x00000010	Paper
0x00000020	Color Temp.
0x00000040	Color Space
0x00000080	Custom Color
0x00000100	Dicom
0x00000200	ComfortView
0x00000400	Warm
0x00000800	Cool
0x00001000	SRGB
0x00002000	Game FPS
0x00004000	Game RTS
0x00008000	Game RPG
For example: 0x00000203 would indicate Standard, Multimedia and ComfortView presets available.	

GetColorPreset

Returns the current color preset

API	
MONITOR_CODE GetColorPreset(UWORD32 *pu32Val)	
Params	
*pu32Val	Pointer to return color preset
Return	
MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure
pu32Val	Color preset
0x00000001	Standard
0x00000002	Multimedia
0x00000004	Movie
0x00000008	Game
0x00000010	Paper
0x00000020	Color Temp.
0x00000040	Color Space
0x00000080	Custom Color
0x00000100	Dicom
0x00000200	ComfortView
0x00000400	Warm
0x00000800	Cool
0x00001000	SRGB
0x00002000	Game FPS
0x00004000	Game RTS
0x00008000	Game RPG

SetColorPreset

Sets the color preset

API

MONITOR_CODE SetColorPreset(UWORD32 u32Val)

Params

u32Val	Color preset
0x00000001	Standard
0x00000002	Multimedia
0x00000004	Movie
0x00000008	Game
0x00000010	Paper
0x00000020	Color Temp.
0x00000040	Color Space
0x00000080	Custom Color
0x00000100	Dicom
0x00000200	ComfortView
0x00000400	Warm
0x00000800	Cool
0x00001000	SRGB
0x00002000	Game FPS
0x00004000	Game RTS
0x00008000	Game RPG

Return

MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

GetCustomColor

Returns the current custom color

NOTE: Only works in Color Preset Custom Color.

API

MONITOR_CODE GetCustomColor(UBYTE u8Val,
UBYTE *pu8ValR, UBYTE *pu8ValG, UBYTE *pu8ValB,
UBYTE *pu8ValC, UBYTE *pu8ValM, UBYTE *pu8ValY)

Params

u8Val	Custom color type to return values
*pu8ValR	Pointer to return R value
*pu8ValG	Pointer to return G value
*pu8ValB	Pointer to return B value
*pu8ValC	Pointer to return C value
*pu8ValM	Pointer to return M value
*pu8ValY	Pointer to return Y value

Return

MONITOR_CODE Code describing the result of the API call

0 Success

1 Timeout

2 Parameters Error

3 Not connected

-1 Other Failure

u8Val Custom color type

0 Gain

1 Offset

2 Hue

3 Saturation

pu8ValR R value, 0 to 100

pu8ValG G value, 0 to 100

pu8ValB B value, 0 to 100

pu8ValC C value, 0 to 100 (Only valid for custom color types Hue and Saturation)

pu8ValM M value, 0 to 100 (Only valid for custom color types Hue and Saturation)

pu8ValY Y value, 0 to 100 (Only valid for custom color types Hue and Saturation)

SetCustomColor

Sets the custom color

NOTE: Only works in Color Preset Custom Color.

API	
MONITOR_CODE SetCustomColor (UBYTE u8Val, UBYTE u8ValR, UBYTE u8ValG, UBYTE u8ValB, UBYTE u8ValC, UBYTE u8ValM, UBYTE u8ValY)	
Params	
u8Val	Custom color type
	0 Gain
	1 Offset
	2 Hue
	3 Saturation
u8ValR	R value, 0 to 100
u8ValG	G value, 0 to 100
u8ValB	B value, 0 to 100
u8ValC	C value, 0 to 100 (Only valid for custom color types Hue and Saturation)
u8ValM	M value, 0 to 100 (Only valid for custom color types Hue and Saturation)
u8ValY	Y value, 0 to 100 (Only valid for custom color types Hue and Saturation)
Return	
MONITOR_CODE	Code describing the result of the API call
	0 Success
	1 Timeout
	2 Parameters Error
	3 Not connected
	-1 Other Failure

GetGammaMode

Returns the gamma mode

API	
MONITOR_CODE GetGammaMode(UBYTE *pu8Val)	
Params	
*pu8Val	Pointer to return gamma mode
Return	
MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure
u8Val	Gamma Mode
0	2.2
1	1.8

SetGammaMode

Sets the gamma mode

API	
MONITOR_CODE SetGammaMode(UBYTE u8Val)	
Params	
u8Val	Gamma Mode
0	2.2
1	1.8
Return	
MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

GetUniformityCompensation

Returns the uniformity compensation setting

API	
MONITOR_CODE GetUniformityCompensation(UBYTE *pu8Val)	
Params	
*pu8Val	Pointer to return uniformity compensation setting
Return	
MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure
pu8Val	Uniformity Compensation
0	Off
2	On

SetUniformityCompensation

Sets the uniformity compensation

API	
MONITOR_CODE SetUniformityCompensation(UBYTE u8Val)	
Params	
u8Val	Uniformity Compensation
0	Off
2	On
Return	
MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

ResetColor

Reset all color settings to default

API

MONITOR_CODE ResetColor(void)

Params

-

Return

MONITOR_CODE	Code describing the result of the API call
--------------	--

0	Success
----------	---------

1	Timeout
----------	---------

2	Parameters Error
----------	------------------

3	Not connected
----------	---------------

-1	Other Failure
-----------	---------------

LUT Management

GetCalibrationHours

Returns the calibration hours of CAL1 or CAL2 since last SetLUT.

NOTE: Only works in Color Preset Color Space CAL1 or CAL2.

API

MONITOR_CODE GetCalibrationHours(SWORD16 *ps16Val)

Params

*ps16Val Pointer to return monitor calibration hours

Return

MONITOR_CODE Code describing the result of the API call

0 Success

1 Timeout

2 Parameters Error

3 Not connected

-1 Other Failure

ps16Val Monitor calibration hours

GetCalBrightness

Get the brightness level of CAL1 or CAL2.

NOTE: Only works in Color Preset Color Space CAL1 or CAL2.

API

MONITOR_CODE GetCalBrightness(UBYTE *pu8Val)

Params

*pu8Val Pointer to return CAL brightness value

Return

MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure
pu8Val	CAL brightness value Integer value 0 (dark) to 100 (bright) Values in increments of 1

SetCalBrightness

Set the brightness level of CAL1 or CAL2.

NOTE: Only works in Color Preset Color Space CAL1 or CAL2.

API

MONITOR_CODE SetCalBrightness(UBYTE u8Val)

Params

u8Val	CAL brightness value Integer value 0 (dark) to 100 (bright) Values in increments of 1
-------	---

Return

MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

SetColorControl

Enables or disables color engine for native color.

NOTE: Only works in Color Preset Color Space CAL1 or CAL2.

API

MONITOR_CODE SetColorControl(UBYTE u8Val)

Params

u8Val	0	Disable
	1	Enable

Return

MONITOR_CODE	Code describing the result of the API call	
	0	Success
	1	Timeout
	2	Parameters Error
	3	Not connected
	-1	Other Failure

SetLUT

Setup the LUT (Look Up Tables) for CAL1 or CAL2.

NOTE: Only works in Color Preset Color Space CAL1 or CAL2.

API

MONITOR_CODE SetLUT(UWORD16 arrPreGamma[3][1025],
UWORD16 arrPostGamma[3][1025],
UWORD16 arrColorMatrix[9])

Params

arrPreGamma	Array with 1025 16-bit color values for RGB channels, where [0][1025] for Red channel, [1][1025] for Green channel and [2][1025] for Blue channel
arrPostGamma	Array with 1025 16-bit color values for RGB channels, where [0][1025] for Red channel, [1][1025] for Green channel and [2][1025] for Blue channel

arrColorMatrix	<p>2's complement values of the 3x3 16-bit color matrix, in the order from a₁ to a₉</p> $R_{out} = (1+a_1)R_{in} + a_2G_{in} + a_3B_{in}$ $G_{out} = a_4R_{in} + (1+a_5)G_{in} + a_6B_{in}$ $B_{out} = a_7R_{in} + a_8G_{in} + (1+a_9)B_{in}$
----------------	---

Return

MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

Video Input Management

GetAutoSelect

Returns the input source auto select setting

API	
MONITOR_CODE GetAutoSelect(UBYTE *pu8Val)	
Params	
*pu8Val	Pointer to return auto select setting
Return	
MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure
pu8Val	Auto Select
0	Off
1	On

SetAutoSelect

Turns on/off input source auto select

API	
MONITOR_CODE SetAutoSelect(UBYTE u8Val)	
Params	
u8Val	Auto Select
0	Off
1	On
Return	

MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

GetVideoInputCaps

Returns the available video inputs

API	
MONITOR_CODE	GetVideoInputCaps(UWORD32 *pu32Val)

Params	
*pu32Val	Pointer to return available video inputs

Return	
MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

pu32Val	Bitwise representation of available video inputs
0x00000001	HDMI1
0x00000002	HDMI2
0x00000004	HDMI3
0x00000008	DP1
0x00000010	DP2
0x00000020	DP3
0x00000040	VGA1
0x00000080	VGA2
0x00000100	DVI1
0x00000200	DVI2

For example:
 0x00000149 would indicate HDMI1, DP1, VGA1 and DVI1 available

GetVideoInput

Returns the current video input source

API	
MONITOR_CODE GetVideoInput(UWORD32 *pu32Val)	
Params	
*pu32Val	Pointer to return video input source
Return	
MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure
pu32Val	Video Input Source
0x00000001	HDMI1
0x00000002	HDMI2
0x00000004	HDMI3
0x00000008	DP1
0x00000010	DP2
0x00000020	DP3
0x00000040	VGA1
0x00000080	VGA2
0x00000100	DVI1
0x00000200	DVI2

SetVideoInput

Sets the video input source

API

MONITOR_CODE SetVideoInput(UWORD32 u32Val)

Params

u32Val	Video Input Source
0x00000001	HDMI1
0x00000002	HDMI2
0x00000004	HDMI3
0x00000008	DP1
0x00000010	DP2
0x00000020	DP3
0x00000040	VGA1
0x00000080	VGA2
0x00000100	DVI1
0x00000200	DVI2

Return

MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

PIP/PBP Management

GetPxPMode

Returns the current PIP/PBP mode

API

MONITOR_CODE GetPxPMode(UBYTE *pu8Val)

Params

*pu8Val Pointer to return PIP/PBP mode

Return

MONITOR_CODE Code describing the result of the API call

0 Success

1 Timeout

2 Parameters Error

3 Not connected

-1 Other Failure

pu8Val PIP/PBP Mode

0 Off

1 PIP Small

2 PIP Large

3 PBP Aspect Ratio

4 PBP Fill

SetPxPMode

Sets the PIP/PBP mode

API

MONITOR_CODE SetPxPMode(UBYTE u8Val)

Params

u8Val	PIP/PBP Mode
0	Off
1	PIP Small
2	PIP Large
3	PBP Aspect Ratio
4	PBP Fill

Return

MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

GetPxPSubInput

Returns the current PxP sub video input source

API

MONITOR_CODE GetPxPSubInput(UWORD32 *pu32Val)

Params

*pu32Val	Pointer to return PxP sub video input source
----------	--

Return

MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

pu32Val	Video Input Source	
	0x00000001	HDMI1
	0x00000002	HDMI2
	0x00000004	HDMI3
	0x00000008	DP1
	0x00000010	DP2
	0x00000020	DP3
	0x00000040	VGA1
	0x00000080	VGA2
	0x00000100	DVI1
0x00000200	DVI2	

SetPxPSubInput

Sets the PxP sub video input source

API

MONITOR_CODE SetPxPSubInput(UWORD32 u32Val)

Params

u32Val	Video Input Source	
	0x00000001	HDMI1
	0x00000002	HDMI2
	0x00000004	HDMI3
	0x00000008	DP1
	0x00000010	DP2
	0x00000020	DP3
	0x00000040	VGA1
	0x00000080	VGA2
	0x00000100	DVI1
0x00000200	DVI2	

Return

MONITOR_CODE	Code describing the result of the API call	
0	Success	
1	Timeout	
2	Parameters Error	
3	Not connected	
-1	Other Failure	

GetPxPLocation

Returns the current PIP/PBP location

API

MONITOR_CODE GetPxPLocation(UBYTE *pu8Val)

Params

*pu8Val Pointer to return PIP/PBP location

Return

MONITOR_CODE Code describing the result of the API call

0 Success

1 Timeout

2 Parameters Error

3 Not connected

-1 Other Failure

pu8Val PIP/PBP Location

0 Top-Right

1 Top-Left

2 Bottom-Right

3 Bottom-Left

SetPxPLocation

Sets the PIP/PBP location

API

MONITOR_CODE SetPxPLocation(UBYTE u8Val)

Params

u8Val	PIP/PBP Location
0	Top-Right
1	Top-Left
2	Bottom-Right
3	Bottom-Left

Return

MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

OSD

GetOSDTransparency

Get the OSD Transparency

API

MONITOR_CODE GetOSDTransparency(UBYTE *pu8Val)

Params

*pu8Val Pointer to return OSD Transparency value

Return

MONITOR_CODE Code describing the result of the API call

0 Success

1 Timeout

2 Parameters Error

3 Not connected

-1 Other Failure

pu8Val OSD Transparency
Integer value 0 (opaque) to 100 (transparent)
Default 20
Values in increments of 20

SetOSDTransparency

Set the OSD Transparency

API

MONITOR_CODE SetOSDTransparency(UBYTE u8Val)

Params

u8Val OSD Transparency
Integer value 0 (opaque) to 100 (transparent)
Default 20
Values in increments of 20

Return

MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

GetOSDLanguage

Get the OSD Language

API

MONITOR_CODE GetOSDLanguage(UBYTE *pu8Val)

Params

*pu8Val Pointer to return OSD Language value

Return

MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

pu8Val	OSD Language
0	English
1	Espanol
2	Francais
3	Deutsch
4	Português (Brasil)
5	Русский
6	简体中文
7	日本語

SetOSDLanguage

Set the OSD Language

API

MONITOR_CODE SetOSDLanguage(UBYTE u8Val)

Params

u8Val	OSD Language
0	English
1	Espanol
2	Francais
3	Deutsch
4	Português (Brasil)
5	Русский
6	简体中文
7	日本語

Return

MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

GetOSDRotation

Get the OSD Rotation

API

MONITOR_CODE GetOSDRotation(UBYTE *pu8Val)

Params

*pu8Val Pointer to return OSD Rotation value

Return

MONITOR_CODE Code describing the result of the API call

0 Success

1 Timeout

2 Parameters Error

3 Not connected

-1 Other Failure

pu8Val OSD Rotation

0 0 degree

1 90 degrees

2 270 degrees

3 180 degrees

SetOSDRotation

Set the OSD Rotations

API

MONITOR_CODE SetOSDRotation(UBYTE u8Val)

Params

u8Val OSD Rotation

0 0 degree

1 90 degrees

2 270 degrees

3 180 degrees

Return

MONITOR_CODE Code describing the result of the API call

0 Success

1 Timeout

2 Parameters Error

3 Not connected

-1 Other Failure

GetOSDTimer

Get the OSD Timer

API

MONITOR_CODE GetOSDTimer(UBYTE *pu8Val)

Params

*pu8Val Pointer to return OSD Timer value

Return

MONITOR_CODE Code describing the result of the API call

0 Success

1 Timeout

2 Parameters Error

3 Not connected

-1 Other Failure

pu8Val OSD Timer

Integer value 5 to 60 seconds

Default 20 seconds

Values in increments of 1

SetOSDTimer

Set the OSD Timer

API

MONITOR_CODE SetOSDTimer(UBYTE u8Val)

Params

pu8Val OSD Timer

Integer value 5 to 60 seconds

Default 20 seconds

Values in increments of 1

Return

MONITOR_CODE Code describing the result of the API call

0 Success

1 Timeout

2 Parameters Error

3 Not connected

-1 Other Failure

GetOSDButtonLock

Get the OSD Timer

API

MONITOR_CODE GetOSDButtonLock(UBYTE *pu8Val)

Params

*pu8Val	Pointer to return OSD Timer value
---------	-----------------------------------

Return

MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

pu8Val	OSD Button Lock
0	Unlock
1	Lock

SetOSDButtonLock

Set the OSD Timer

API

MONITOR_CODE SetOSDButtonLock(UBYTE u8Val)

Params

u8Val	OSD Button Lock
0	Unlock
1	Lock

Return

MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

ResetOSD

Reset OSD to default settings

API

MONITOR_CODE ResetOSD(void)

Params

-

Return

MONITOR_CODE	Code describing the result of the API call
--------------	--

0	Success
----------	---------

1	Timeout
----------	---------

2	Parameters Error
----------	------------------

3	Not connected
----------	---------------

-1	Other Failure
-----------	---------------

System Management

GetVersionFirmware

Returns the firmware version of the monitor

API	
MONITOR_CODE GetVersionFirmware(BYTE *ps16Val)	
Params	
*pbyFirmwareVersion	Pointer to return OSD Transparency value
Return	
MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure
pbyFirmwareVersion	Version string (max 10 chars)

GetVersionSDK

Returns the SDK version

API	
MONITOR_CODE GetVersionSDK(SWORD16 *ps16Val)	
Params	
*pbyVersion	Pointer to firmware version for return
Return	
MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure
ps16Val	Version value where MSB = major version and LSB = minor version. Eg) 0x0100 will mean Version 1.0

GetMST

Returns if the MST is on or off

API

MONITOR_CODE GetMST(UBYTE *pu8Val)

Params

*pu8Val	Pointer to return MST value
---------	-----------------------------

Return

MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

pu8Val	MST value
0	Off
1	On

SetMST

Turns on / off the MST

API

MONITOR_CODE SetMST(UBYTE u8Val)

Params

u8Val	MST value to set
0	Off
1	On

Return

MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

GetLCDConditioning

Returns if the LCD Conditioning is enabled or disabled

API

MONITOR_CODE GetLCDConditioning(UBYTE *pu8Val)

Params

*pu8Val	Pointer to return LCD Conditioning value
---------	--

Return

MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

pu8Val	LCD Conditioning
0	Disabled
1	Enabled

SetLCDConditioning

Enable / Disable the LCD Conditioning

API

MONITOR_CODE SetLCDConditioning (UBYTE u8Val)

Params

u8Val	LCD Conditioning value to set
0	Disabled
1	Enabled

Return

MONITOR_CODE	Code describing the result of the API call
0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

FactoryReset

Reset to factory settings

API

MONITOR_CODE FactoryReset(void)

Params

-

Return

MONITOR_CODE Code describing the result of the API call

0	Success
1	Timeout
2	Parameters Error
3	Not connected
-1	Other Failure

SetDebugLevel

Set the level of debug for the SDK

API

MONITOR_CODE SetDebugLevel(UBYTE u8Val)

Params

u8Val	Debug Level
0	Turn off debug
1	Errors
2	Warnings
3	Debug
4	Trace

Return

MONITOR_CODE Code describing the result of the API call

0	Success
1	Errors
2	Parameters Error
3	Not connected
-1	Other Failure
