

# HP DreamColor Z31x/Z27xG2 SDK/API Documentation



Version 1.1, February 2018

## Table of contents

Introduction .....	7
Connect APIs.....	7
Scan for Displays.....	7
Connect to a Display.....	7
Disconnect from Display.....	8
Color Space APIs.....	8
Adjust Luminance.....	8
Read Luminance.....	8
Switch Color Space .....	9
Read Color Space .....	9
Read Color Space name.....	10
Read Red Primaries in xy format.....	10
Read Red Primaries in uv format.....	10
Read Green Primaries in xy format .....	10
Read Green Primaries in uv format.....	11
Read Blue Primaries in xy format.....	11
Read Blue Primaries in uv format.....	11
Read White Primaries in xy format.....	11
Read White Primaries in uv format .....	12
Read Gamma.....	12
Read all Color Space names .....	12
Set Blue Light mode .....	13
Read Blue Light mode.....	13
Adjust RGB Downstream .....	13
Read RGB Downstream .....	13
Video Input APIs .....	14
Change Video Input.....	14
Read Video Input .....	14
Change Auto Switch status .....	14
Read Auto Switch status.....	15
Set Video Input Name .....	15

Read Video Input Name .....	16
Change USB Port Binding Status .....	16
Read USB Port Binding Status.....	16
Change USB Port Binding Configuration .....	17
Read USB Port Binding Configuration.....	17
Image Adjustment APIs .....	17
Change Image Scaling.....	17
Read Image Scaling .....	18
Change DCI Region settings.....	18
Read DCI Region settings .....	19
Change Overscan status.....	19
Read Overscan status .....	20
Change Overdrive status .....	20
Read Overdrive status.....	20
Change Blue Channel Only status .....	20
Read Blue Channel Only status.....	21
Change Deinterlacer setting .....	21
Read Deinterlacer status.....	21
Change Cadence Detection setting .....	21
Read Cadence Detection status .....	22
Change Show Cropped Region status .....	22
Read Show Cropped Region status.....	22
Change Cropped Region opacity.....	22
Read Cropped Region Opacity .....	23
Change Video Legal status.....	23
Read Video Legal status .....	23
Change Marker Details .....	23
Read Marker Details .....	24
Change Marker Color .....	24
Read Marker Color .....	24
Clear Markers .....	25
PIP Control APIs .....	25
Change PIP Position.....	25
Read PIP Position .....	25
Change PIP Status.....	26
Read PIP Status .....	26
Change PIP Mode .....	26
Read PIP Mode .....	27
Change PIP Size .....	27
Read PIP Size .....	27
Change PIP Primary Input Source.....	28

Read PIP Primary Input Source .....	28
Change PIP Secondary Input Source.....	28
Read PIP Secondary Input Source .....	29
Swap PIP and Inputs.....	29
Read Swap PIP Input Status .....	29
Change PIP Video Legal Status .....	30
Read PIP Video Legal Status.....	30
Change PIP Overscan Status .....	30
Read PIP Overscan Status.....	30
Change PIP DCI Status.....	31
Read PIP DCI Status .....	31
Change PIP Marker Details.....	31
Read PIP Marker Details .....	32
Change PIP Marker Color.....	32
Read PIP Marker Color .....	32
Clear PIP Markers .....	33
Management APIs .....	33
Change Auto sleep mode .....	33
Read Auto sleep mode.....	33
Change Power on recall status .....	33
Read Power on recall status.....	34
Change DDC/CI Communication status.....	34
Read DDC/CI Communication status.....	34
Change Auto EDID update status .....	34
Read Auto EDID update status.....	35
Change Hot Plug Initiate status .....	35
Read Hot Plug Initiate status.....	35
Change DP HPD Status .....	35
Read DP HPD Status.....	36
Change Firmware update status .....	36
Read Firmware update status.....	36
Change Calibration status .....	37
Read Calibration status.....	37
Change WS-Management status.....	37
Read WS-Management status .....	37
Change WS Identify Support status .....	38
Read WS Identify Support status.....	38
Change Web Server status.....	38
Read Web Server status .....	38
Change Dashboard Security .....	39
Read Web Dashboard Security.....	39

Change IP configuration .....	39
Read IP configuration .....	39
Switch OFF Manage Internal processor .....	40
Change Date and Time .....	40
Read Date and Time .....	40
Change Auto Adjust Date and Time status .....	40
Read Auto Adjust Date and Time Status .....	41
Change Auto Adjust Daylight Saving status .....	41
Read Auto Adjust Daylight Saving Status .....	41
Change Device Time Format .....	41
Read Device Time Format .....	42
Change External Calibration Device Support .....	42
Read External Calibration Device Support .....	42
Change User Preset Support Status .....	42
Read User Preset Support Status .....	43
Change USB Type-C Input Configuration .....	43
Read USB Type-C Input Configuration .....	43
Change Time Zone .....	43
Read Time Zone .....	46
Lock Management Menu .....	46
Align Internal Calibration Device .....	46
Change USB Functionality during sleep .....	46
Read USB Functionality during sleep .....	47
Change DisplayPort EDID Configuration .....	47
Read Display Port EDID Configuration .....	47
Change Luminance Uniformity .....	47
Read Luminance Uniformity .....	48
Factory Reset .....	48
Menu and Message Control APIs .....	48
Change OSD Position .....	48
Read OSD Position .....	49
Change OSD Opacity .....	49
Read OSD Opacity .....	49
Change OSD Timeout .....	50
Read OSD Timeout .....	50
Change Power on Information OSD Settings .....	50
Read Power on Information OSD Settings .....	50
Change Input source Information OSD Settings .....	51
Read Input Source Information OSD Settings .....	51
Change Colorspace Information OSD Settings .....	51
Read Colorspace Information OSD Settings .....	51

Change Warmup Time Reminder Info OSD Settings .....	52
Read WarmupTime Reminder Info OSD Settings.....	52
Change Calibration Notification Info OSD Settings .....	52
Read Calibration Notification Info OSD Settings.....	53
Change Function Button Settings.....	53
Read Function Button Settings .....	54
Change Function Button Mode.....	55
Read Function Button Mode .....	55
Change Bezel Button Color .....	55
Read Bezel Button Color .....	56
Change Bezel Button Brightness.....	56
Read Bezel Button Brightness .....	56
Change Bezel Button Auto-Fade .....	56
Read Bezel Button Auto-Fade.....	57
Change Menu Language .....	57
Read Menu Language.....	57
Information APIs.....	58
Read Firmware revision .....	58
Read Display Serial Number .....	58
Read Display Mode .....	58
Read Calibration hours.....	58
Read Backlight hours .....	59
Read Remote Customer ID Asset Tag .....	59
Read Display Name.....	59
Read Deployment Location .....	59
Read Deployment Date.....	60
Set Log.....	60
Test Pattern Generator APIs .....	60
Enable Test Pattern.....	60
Disable Test Pattern.....	60
Read Test Pattern Status .....	61
Select Test Pattern: Service Mode.....	61
Select Test Pattern: User Mode .....	61
Remote Configuration APIs.....	62
Read Colormanagement Status .....	62
Read Calibration Alert Limit .....	62
Read Dithering Mode Status .....	62
SDK Version API .....	63
Release Version.....	63
Calibration APIs .....	63
Change Calibration schedule .....	63

Read Calibration schedule..... 63

Change Warmup Settings..... 63

Read Calibration schedule..... 64

Recalibrate Active Preset ..... 64

Configure and Calibrate Preset ..... 64

Send File to Display ..... 65

Validate Studio XML File ..... 65

Perform Studio Calibration ..... 66

Get File from Display ..... 66

Variable Definitions..... 67

## Introduction

This document provides the supported APIs for the HP Z31x/Z27x-G2 USB SDK on the Linux(x86), Mac and Windows platforms. These APIs are to be used for remote display management and control from a Host PC to the Z31x/Z27x-G2 via a USB connection. A USB 3.0 A to B cable should be used for the connection between the host and the display.

## Connect APIs

### Scan for Displays

#### Description

Scans all the USB devices connected to Host PC, and return the list of active displays. The serial number of the displays is used for identification.

#### API

```
STATUS Scan(BYTE byDisplayName[MAX_DISPLAYS][SERIALNUMBER_SIZE]);
```

#### Example

```
Status = Scan(byDisplayName);
```

#### Params

- MAX\_DISPLAYS = 3
- SERIALNUMBER\_SIZE = 12
- STATUS : returns the number of connected displays or error reasons
  - STATE\_ER\_OPEN = -12,
  - STATE\_ER\_ENUM,
  - STATE\_ER\_NO\_DEV,
  - STATE\_ER\_TIMEOUT,
  - STATE\_ER\_CLAIM\_INTERFACE,
  - STATE\_ER\_INVALID\_ARG,
  - STATE\_ER\_NOT\_CONNECTED,
  - STATE\_ER\_COMM,
  - STATE\_ER\_API,
  - STATE\_ER\_API\_NOT\_SUPPORTED,
  - STATE\_ER\_MONITOR\_MODE,
  - FAILURE = -1,
  - SUCCESS = 0
- byDisplayName : The serial numbers of all the active connected displays.

### Connect to a Display

#### Description

Connect to an active display. Connect to one of the display, selected from the list detected by the Scan API.

#### API

```
STATUS Connect(BYTE iID);
```

#### Example

```
Status = Connect(iID);
```

#### Params

- iID : The index of the active display scanned. To connect to the first display in the list, use iID = 0.
- STATUS: returns the connection status
  - STATE\_ER\_OPEN = -12,

```
STATE_ER_ENUM,
STATE_ER_NO_DEV,
STATE_ER_TIMEOUT,
STATE_ER_CLAIM_INTERFACE,
STATE_ER_INVALID_ARG,
STATE_ER_NOT_CONNECTED,
STATE_ER_COMM,
STATE_ER_API,
STATE_ER_API_NOT_SUPPORTED,
STATE_ER_MONITOR_MODE,
FAILURE = -1,
SUCCESS = 0
```

## Disconnect from Display

### Description

Disconnect from a display.

### API

```
STATUS Disconnect(void);
```

### Example

```
Status = Disconnect();
```

### Params

- STATUS : returns the connection status
- FAILURE = -1
- SUCCESS = 0

## Color Space APIs

### Adjust Luminance

#### Description

Adjust backlight luminance value. The minimum value the luminance can be set to is 48 cd/m<sup>2</sup> and the maximum value is 250 cd/m<sup>2</sup>. The accuracy target for luminance adjustment is  $\pm 1$  cd/m<sup>2</sup>.

#### API

```
STATUS SetDisplayLuminance(SWORD16 s16Val);
```

#### Example

```
Status = SetDisplayLuminance(s16Val);
```

#### Params

- s16Val : Luminance value to be set in cd/m<sup>2</sup>.  
Value ranges from 48 to 250 cd/m<sup>2</sup>

### Read Luminance

#### Description

Read backlight luminance value

#### API

```
STATUS GetDisplayLuminance(SWORD16 *ps16Val);
```



### Example

```
Status = GetDisplayLuminance(ps16Val);
```

### Params

- ps16Val : Current Luminance value

## Switch Color Space

### Description

Switch between the available color spaces. The preset number of the Color Spaces range from 1 to 8. The default factory Color Space assignments are:

- 1 : DCI P3 D65
- 2 : BT.709
- 3 : BT.2020
- 4 : sRGB D65
- 5 : sRGB D50
- 6 : AdobeRGB D65
- 7 : Adobe RGB D50
- 8 : Native

### API

```
STATUS SetActiveColorSpace(UBYTE u8Val);
```

### Example

```
Status = SetActiveColorSpace(u8Val);
```

### Params

- u8Val : Color Space preset number to be set.  
Value ranges from 1 to 8. Default values are:
- 1 : DCI P3 D65
- 2 : BT.709
- 3 : BT.2020
- 4 : sRGB D65
- 5 : sRGB D50
- 6 : AdobeRGB D65
- 7 : Adobe RGB D50
- 8 : Native

These default values can be changed via Studio calibration.

## Read Color Space

### Description

Read the current Color Space preset number.

### API

```
STATUS GetActiveColorSpace(UBYTE *pu8Val)
```

### Example

```
Status = GetActiveColorSpace(pu8Val);
```

### Params

- u8Val : Current Color Space preset number
- If Blue Light mode is set, the Color Space value returned will be 0.

## Read Color Space name

### Description

Read the current Color Space name.

### API

STATUS GetCurrentColorSpaceName(BYTE byVal[COLOR\_PRESET\_NAME\_LENGTH])

### Example

```
Status = GetCurrentColorSpaceName(byVal);
```

### Params

- pbyVal : Current Color Space name

COLOR\_PRESET\_NAME\_LENGTH -15

## Read Red Primaries in xy format

### Description

Read the red primaries of current Color Space in xy format.

### API

STATUS GetRedPrimaryXY(float \*x, float \*y);

### Example

```
Status = GetRedPrimaryXY(x, y);
```

### Params

- x : red primaries x
- y : red primaries y

## Read Red Primaries in uv format

### Description

Read the red primaries of current Color Space in uv format.

### API

STATUS GetRedPrimaryUV(float \*U, float \*V);

### Example

```
Status = GetRedPrimaryUV(U, V);
```

### Params

- U : red primaries u' value
- V : red primaries v' value

## Read Green Primaries in xy format

### Description

Read the green primaries of current Color Space in xy format.

### API

STATUS GetGreenPrimaryXY(float \*x, float \*y);

**Example**

```
Status = GetGreenPrimaryXY(x, y);
```

**Params**

- x : green primaries x
- y : green primaries y

**Read Green Primaries in uv format****Description**

Read the green primaries of current Color Space in uv format.

**API**

```
STATUS GetGreenPrimaryUV(float *U, float *V);
```

**Example**

```
Status = GetGreenPrimaryUV(U, V);
```

**Params**

- U : green primaries u' value
- V : green primaries v' value

**Read Blue Primaries in xy format****Description**

Read the blue primaries of current Color Space in xy format.

**API**

```
STATUS GetBluePrimaryXY(float *x, float *y);
```

**Example**

```
Status = GetBluePrimaryXY(x, y);
```

**Params**

- x : blue primaries x
- y : blue primaries y

**Read Blue Primaries in uv format****Description**

Read the blue primaries of current Color Space in uv format.

**API**

```
STATUS GetBluePrimaryUV(float *U, float *V);
```

**Example**

```
Status = GetBluePrimaryUV(U, V);
```

**Params**

- U : blue primaries u' value
- V : blue primaries v' value Shiny

**Read White Primaries in xy format****Description**

Read the white primaries of current Color Space in xy format.

#### **API**

STATUS GetWhitePrimaryXY(float \*x, float \*y);

#### **Example**

```
Status = GetWhitePrimaryXY(x, y);
```

#### **Params**

- x : White primaries x
- y : White primaries y

### **Read White Primaries in uv format**

#### **Description**

Read the white primaries of current Color Space in uv format.

#### **API**

STATUS GetWhitePrimaryUV(float \*U, float \*V);

#### **Example**

```
Status = GetWhitePrimaryUV(U, V);
```

#### **Params**

- U : White primaries u' value
- V : White primaries v' value

### **Read Gamma**

#### **Description**

Read the Gamma Name of current Color Space.

#### **API**

STATUS GetGamma(BYTE \*pbyVal);

#### **Example**

```
Status = GetGamma(pbyVal);
```

#### **Params**

- pbyVal : Gamma Name of current Color Space

### **Read all Color Space names**

#### **Description**

Read all the color space name of the monitor.

#### **API**

STATUS GetPresetNameList(BYTE pbyVal[COLOR\_PRESET\_NUM][COLOR\_PRESET\_NAME\_LENGTH]);

#### **Example**

```
STATUS GetPresetNameList(BYTE pbyVal[COLOR_PRESET_NUM][COLOR_PRESET_NAME_LENGTH]);
```

#### **Params**

- COLOR\_PRESET\_NUM = 8
- COLOR\_PRESET\_NAME\_LENGTH = 15
- pbyVal : list all the 8 colorspace name

## **Set Blue Light mode**

#### **Description**

Set Blue Light mode status of the monitor.

#### **API**

```
STATUS SetBlueLightModeStatus(UBYTE u8Val);
```

#### **Example**

```
STATUS SetBlueLightModeStatus(u8Val);
```

#### **Params**

- u8Val 1 : Enable ; 0 : Disable

## **Read Blue Light mode**

#### **Description**

Read Blue Light mode status of the monitor.

#### **API**

```
STATUS GetBlueLightModeStatus(UBYTE *pu8Val);
```

#### **Example**

```
STATUS GetBlueLightModeStatus(pu8Val);
```

#### **Params**

- u8Val 1 : Enable ; 0 : Disable

## **Adjust RGB Downstream**

#### **Description**

Adjust RGB Downstream Gain and Offset.

#### **API**

```
STATUS SetDownstreamRGBAdjust(SWORD16 u8RGBGain[3], SWORD16 u8RGBOffset[3]);
```

#### **Example**

```
STATUS SetDownstreamRGBAdjust(u8RGBGain[3], u8RGBOffset[3]);
```

#### **Params**

- u8RGBGain : Gain for RGB respectively ranges from 0-1023
- u8RGBOffset : Offset for RGB respectively ranges from 0-1023

## **Read RGB Downstream**

#### **Description**

Read RGB Downstream Gain and Offset.

#### **API**

```
STATUS GetDownstreamRGBAdjust(SWORD16 u8RGBGain[3], SWORD16 u8RGBOffset[3]);
```

### Example

```
STATUS GetDownstreamRGBAdjust(u8RGBGain[3],u8RGBOffset[3]);
```

### Params

- u8RGBGain : Gain for RGB respectively
- u8RGBOffset : Offset for RGB respectively

## Video Input APIs

### Change Video Input

#### Description

Switch between video inputs. The video input types supported are DisplayPort, HDMI and USB Type-C.

#### API

```
STATUS SetActiveVideoInput(UBYTE u8Val)
```

### Example

```
Status = SetActiveVideoInput(u8Val);
```

### Params

- u8Val : Video input value to be set.  
Value ranges from 0 to 4. Default values are:
  - 0 : DisplayPort 1
  - 1 : DisplayPort 2
  - 2 : HDMI 1
  - 3 : HDMI 2
  - 4 : USB Type-C

### Read Video Input

#### Description

Read the current video input.

#### API

```
STATUS GetActiveVideoInput(UBYTE *u8Val)
```

### Example

```
Status = GetActiveVideoInput(u8Val);
```

### Params

- u8Val : Current video input

### Change Auto Switch status

#### Description

Enable/Disable the auto switch status.

#### API

```
STATUS SetAutoSwitchStatus(UBYTE u8Val)
```

### Example

```
Status = SetAutoSwitchStatus(u8Val);
```

### Params

- u8Val : Enable/Disable Auto switch status.  
0 or 1 can be set.

## Read Auto Switch status

### Description

Read the auto switch status.

### API

STATUS GetAutoSwitchStatus(UBYTE \*u8Val)

### Example

```
Status = GetAutoSwitchStatus(u8Val);
```

### Params

- u8Val : Current auto switch status

## Set Video Input Name

### Description

Set the Video Input Name.

### API

STATUS SetVideoInputName(UBYTE u8videoInputNo, UBYTE u8videoInputNameType, BYTE byCustomName[VIDEO\_INPUT\_NAME\_LENGTH]);

### Example

```
Status = SetVideoInputName(u8videoInputNo, u8videoInputNameType, byCustomName);
```

### Params

- u8videoInputNo : Video InputNo : 0-4
- u8videoInputNameType : Video Input Name Type

DEFAULT = 0

WORKSTATION = 1

WORKSTATION2 = 2

LAPTOP = 3

LAPTOP2 = 4

WINDOWS\_SYSTEM = 5

LINUX\_SYSTEM = 6

OS\_X\_SYSTEM = 7

MASTER = 8

PLAYBACK = 9

PREVIEW = 10

CUSTOM = 11

- CustomName : Custom name for the input. Used only if Video Input name type is Custom

VIDEO\_INPUT\_NAME\_LENGTH = 15

## Read Video Input Name

### Description

Read the Video Input Name.

### API

```
STATUS GetVideoInputName(UBYTE u8videoInputNo, BYTE byInputName[VIDEO_INPUT_NAME_LENGTH], UBYTE *pu8customNameStatus);
```

### Example

```
Status = GetVideoInputName(u8videoInputNo, byInputName, pu8customNameStatus);
```

### Params

- u8videoInputNo : Video InputNo : 0-4
- pbyInputName : Video Input Name
- pu8customNameStatus : Custom Name Status
  - 0- Default/Predefined Name
  - 1- Custom Name

## Change USB Port Binding Status

### Description

Set USB Port Binding Status

### API

```
STATUS SetUSBPortBindingStatus(UBYTE u8USBPortbindingStatus);
```

### Example

```
Status = SetUSBPortBindingStatus(u8USBPortbindingStatus);
```

### Params

- u8USBPortbindingStatus : Port Binding status are as shown below:
  - 0- USB Auto Detect
  - 1- Use USB 1(Type B ) only
  - 2- Use USB 2(Type C ) only
  - 3- Bind USB1 and 2 to specific Inputs

## Read USB Port Binding Status

### Description

Read USB Port Binding Status

### API

```
STATUS GetUSBPortBindingStatus(UBYTE *pu8USBPortbindingStatus);
```

### Example

```
Status = GetUSBPortBindingStatus(pSu8USBPortbindingStatus);
```



#### Params

- pu8USBPortbindingStatus : USB Port Binding Status

### Change USB Port Binding Configuration

#### Description

Set USB Port Binding Configuration. This API will work only if the Binding status is set to 3.

#### API

STATUS SetUSBBindingConfiguration(UBYTE u8USB1Binding , UBYTE u8USB2Binding);

#### Example

```
Status = SetUSBBindingConfiguration(u8USB1Binding, u8USB2Binding);
```

#### Params

- u8USB1Binding : Input Number bound to USB1, Input Number ranges from 0-3
  - 0 : DisplayPort 1
  - 1 : DisplayPort 2
  - 2 : HDMI 1
  - 3 : HDMI 2
- u8USB2Binding : Input Number bound to USB2, Input Number ranges from 0-4
  - 0 : DisplayPort 1
  - 1 : DisplayPort 2
  - 2 : HDMI 1
  - 3 : HDMI 2
  - 4 : USB Type-C

### Read USB Port Binding Configuration

#### Description

Read USB Port Binding Configuration

#### API

STATUS GetUSBBindingConfiguration(UBYTE \*pu8USB1Binding , UBYTE \*pu8USB2Binding);

#### Example

```
Status = GetUSBBindingConfiguration(pu8USB1Binding, pu8USB2Binding);
```

#### Params

- pu8USB1Binding : Input Number bound to USB1
- pu8USB2Binding : Input Number bound to USB2

## Image Adjustment APIs

### Change Image Scaling

#### Description

Change the aspect ratio settings.

## API

STATUS SetImageScaling(UBYTE u8AspectRatio, UBYTE u8PixelStatus);

## Example

```
Status = SetImageScaling(u8AspectRatio, u8PixelStatus);
```

## Params

- u8AspectRatio: Aspect Ratio settings

- 0 : Fit to source Aspect Ratio (Proportional)
- 1 : Fit to screen width (Proportional)
- 2 : Fit to screen height (Proportional)
- 3 : Fill to 17:9 (non-proportional)
- 4 : Fill to 16:9 (non-proportional)
- 5 : Fill to 1:85:1 (non-proportional)
- 6 : Fill to 2:39:1 (non-proportional)
- 7 : Pixel-for-pixel

Fill to 17:9 (non-proportional) is not available for Z27x-G2.

- u8PixelStatus: True 2k pixel Mapping status. Pixel status will enabled only if aspect ratio is “Pixel –for-pixel”  
This option is enabled only for the below resolutions

Pixel Format  
1280 × 720  
1920 × 1080  
2048 × 1080

This option is only valid for Z31x. It will be discarded in case of Z27x-G2.

## Read Image Scaling

### Description

Read the aspect ratio settings.

## API

STATUS GetImageScaling(UBYTE \*pu8AspectRatio, UBYTE \*pu8PixelStatus);

## Example

```
Status = GetImageScaling(pu8AspectRatio, pu8PixelStatus);
```

## Params

- pu8AspectRatio: Aspect Ratio settings
- pu8PixelStatus: True 2k pixel Mapping status

This option is only valid for Z31x. It will be discarded in case of Z27x-G2.

## Change DCI Region settings

### Description

DCI settings can be changed to any of the below options.

- 0 : Show Entire DCI Container
- 1 : Crop to DCI 1.85:1 Aspect Ratio
- 2 : Crop to DCI 2.39:1 Aspect Ratio

#### API

STATUS SetDciImageRegion(UBYTE u8Val)

#### Example

```
Status = SetDciImageRegion(u8Val);
```

#### Params

- u8Val : DCI region option to be set  
Ranges from 0 to 2. The values are:
  - 0 : Show Entire DCI Container
  - 1 : Crop to DCI 1.85:1 Aspect Ratio
  - 2 : Crop to DCI 2.39:1 Aspect Ratio

This option will be allowed for the following resolutions

1. 4096x2160
2. 2048x1080

## Read DCI Region settings

#### Description

Read the current DCI setting of the selected input resolution.

#### API

STATUS GetDciImageRegion(UBYTE \*u8Val)

#### Example

```
Status = GetDciImageRegion(u8Val);
```

#### Params

- u8Val : current DCI setting

This option will be allowed for the following resolutions

1. 4096x2160
2. 2048x1080

## Change Overscan status

#### Description

Enable/Disable the Overscan status

#### API

STATUS SetOverScanStatus(UBYTE u8Val)

#### Example

```
Status = SetOverScanStatus(u8Val);
```

#### **Params**

- u8Val : Enable/Disable status.  
0 or 1 can be set.

### **Read Overscan status**

#### **Description**

Read the Overscan status

#### **API**

STATUS GetOverScanStatus(UBYTE \*u8Val)

#### **Example**

```
Status = GetOverScanStatus(u8Val);
```

#### **Params**

- u8Val : Current Overscan Status

### **Change Overdrive status**

#### **Description**

Enable/Disable the Overdrive status

#### **API**

STATUS SetOverDriveStatus(UBYTE u8Val)

#### **Example**

```
Status = SetOverDriveStatus(u8Val);
```

#### **Params**

- u8Val : Enable/Disable the value  
0 or 1 can be set.

### **Read Overdrive status**

#### **Description**

Read the current Overdrive status.

#### **API**

STATUS GetOverDriveStatus(UBYTE \*u8Val)

#### **Example**

```
Status = GetOverDriveStatus(u8Val);
```

#### **Params**

- u8Val : Current Overdrive Status

### **Change Blue Channel Only status**

#### **Description**

Change the current Blue Channel Only status.

#### **API**

STATUS SetBlueChannelStatus(UBYTE u8Val)

#### **Example**

```
Status = SetBlueChannelStatus(u8Val);
```

#### **Params**

- u8Val : Enable/Disable the value  
0 or 1 can be set.

### **Read Blue Channel Only status**

#### **Description**

Read the current Blue Channel status.

#### **API**

STATUS GetBlueChannelStatus(UBYTE \*pu8Val)

#### **Example**

```
Status = GetBlueChannelStatus(pu8Val);
```

#### **Params**

- u8Val : Current Blue Channel Status

### **Change Deinterlacer setting**

#### **Description**

Change the current Deinterlacer setting.

#### **API**

STATUS SetDeinterlacerStatus(UBYTE u8Val)

#### **Example**

```
Status = SetDeinterlacerStatus(u8Val);
```

#### **Params**

- u8Val : Enable/Disable the value  
0 or 1 can be set.

### **Read Deinterlacer status**

#### **Description**

Read the current Deinterlacer status.

#### **API**

STATUS GetDeinterlacerStatus(UBYTE \*pu8Val)

#### **Example**

```
Status = GetDeinterlacerStatus(pu8Val);
```

#### **Params**

- u8Val : Current Deinterlacer Status

### **Change Cadence Detection setting**

#### **Description**

Change the current Cadence Detection setting.

#### **API**

STATUS SetCadenceDetectionStatus(UBYTE u8Val)

#### **Example**

```
Status = SetCadenceDetectionStatus(u8Val);
```

#### **Params**

- u8Val : Enable/Disable the value  
0 or 1 can be set.

### **Read Cadence Detection status**

#### **Description**

Read the current Cadence Detection status.

#### **API**

STATUS GetCadenceDetectionStatus(UBYTE \*pu8Val)

#### **Example**

```
Status = GetCadenceDetectionStatus(pu8Val);
```

#### **Params**

- u8Val : Current Cadence Detection Status

### **Change Show Cropped Region status**

#### **Description**

Change the show cropped region status.

#### **API**

STATUS SetShowCroppedRegionStatus(UBYTE u8Val)

#### **Example**

```
Status = SetShowCroppedRegionStatus(u8Val);
```

#### **Params**

- u8Val : Enable/Disable the value  
0 or 1 can be set.

This settings is available only if DCI Image region is set to 2.

### **Read Show Cropped Region status**

#### **Description**

Read the show cropped region status.

#### **API**

STATUS GetShowCroppedRegionStatus(UBYTE \*pu8Val)

#### **Example**

```
Status = GetShowCroppedRegionStatus(&pu8Val);
```

#### **Params**

- pu8Val : Current cropped region status

### **Change Cropped Region opacity**

#### **Description**

Change the cropped region opacity.

#### **API**

STATUS SetCroppedRegionOpacity(UBYTE u8Val)

### Example

```
Status = SetCroppedRegionOpacity(u8Val);
```

### Params

- u8Val : 10 to 80

This settings is available only if DCI Image region is set to 2

## Read Cropped Region Opacity

### Description

Read the cropped region opacity.

### API

```
STATUS GetCroppedRegionOpacity(UBYTE *pu8Val)
```

### Example

```
Status = GetCroppedRegionOpacity(&pu8Val);
```

### Params

- pu8Val : Current cropped region opacity

## Change Video Legal status

### Description

Change the video legal status.

### API

```
STATUS SetVideoLegalStatus(UBYTE u8Val)
```

### Example

```
Status = SetVideoLegalStatus(u8Val);
```

### Params

- u8Val : Enable/Disable the value  
0 or 1 can be set.

## Read Video Legal status

### Description

Read the video legal status.

### API

```
STATUS GetVideoLegalStatus(UBYTE *pu8Val)
```

### Example

```
Status = GetVideoLegalStatus(&pu8Val);
```

### Params

- pu8Val : Current video legal status

## Change Marker Details

### Description

Change the marker details.

## API

STATUS SetMarkerDetails(const MarkerStatus\* psMarkerDetails);

## Example

```
Status = SetMarkerDetails(psMarkerDetails);
```

## Params

- psMarkerDetails: pointer of type MarkerStatus structure filled accordingly.

Note :

“ClearAll” Member of the structure should be always 0 for this API.

16:9 Extraction Marker is not available for Z27x-G2

## Read Marker Details

### Description

Read the marker details.

## API

STATUS GetMarkerDetails( MarkerStatus\* psMarkerDetails);

## Example

```
Status = GetMarkerDetails(psMarkerDetails);
```

## Params

- psMarkerDetails : Current settings of marker.

## Change Marker Color

### Description

Change the marker color.

## API

STATUS SetMarkerColor(UBYTE u8MarkerColor);

## Example

```
Status = SetMarkerColor(u8MarkerColor);
```

## Params

- u8MarkerColor: Marker color. Available colors are
  - 0 : White
  - 1 : Red
  - 2 : Green
  - 3 : Blue
  - 4 : Cyan
  - 5 : Majenta
  - 6 : Yellow

## Read Marker Color

### Description

Read the marker Color.



### API

STATUS GetMarkerColor(UBYTE \*pu8MarkerColor);

### Example

Status = GetMarkerColor(pu8MarkerColor);

### Params

- pu8MarkerColor : Current marker Color

## Clear Markers

### Description

Clear all marker settings.

### API

STATUS ClearMarkers();

### Example

Status = ClearMarkers();

## PIP Control APIs

### Change PIP Position

#### Description

Change the position of PIP by specifying the position corners or flexible custom positions.

### API

STATUS SetPIPPosition(UBYTE u8PIPPositionStatus, SWORD16 s16XValue, SWORD16 s16YValue);

### Example

Status = SetPIPPosition(u8PIPPositionStatus, s16XValue, s16YValue);

### Params

- u8PIPPositionStatus : PIP Position identifiers whose value ranges from 0-4
  - 0 : Top Left
  - 1 : Top Right
  - 2 : Bottom Left
  - 3 : Bottom Right
  - 4 : Custom Position
- s16XValue : Horizontal value for Custom PIP position ranges from 204 to 4096 – (204+PIP width).
- s16YValue : Vertical value for Custom PIP position ranges from 108 to 2160 – (108+PIP height).

### Read PIP Position

#### Description

Read the current PIP Position.

### API

STATUS GetPIPPosition(UBYTE \*pu8PIPPositionStatus, SWORD16 \*ps16XValue, SWORD16 \*ps16YValue);

### Example

Status = GetPIPPosition(pu8PIPPositionStatus, ps16XValue, ps16YValue);

### Params

- pu8PIPPositionStatus : Current value of the PIP Position identifiers
- ps16XValue : Horizontal value for Custom PIP position
- ps16YValue : Vertical value for Custom PIP position

## Change PIP Status

### Description

Change the PIP status.

### API

```
STATUS SetPIPStatus(UBYTE u8Val);
```

### Example

```
Status = SetPIPStatus(u8Val)
```

### Params

- u8Val : PIP status to be set  
Ranges from 0 to 2. The values:  
0 : OFF  
1 : ON

## Read PIP Status

### Description

Read the current PIP status.

### API

```
STATUS GetPIPStatus(UBYTE *pu8Val);
```

### Example

```
Status = GetPIPStatus(pu8Val);
```

### Params

- pu8Val : Current PIP status

## Change PIP Mode

### Description

Change the PIP mode.

### API

```
STATUS SetPIPMODE(UBYTE u8Val);
```

### Example

```
Status = SetPIPMODE(u8Val)
```

### Params

- u8Val : PIP mode to be set  
Ranges from 0 to 2. The values:  
0 : OFF  
1 : Picture-in-Picture  
2 : Picture-beside-Picture

## Read PIP Mode

### Description

Read the current PIP mode.

### API

STATUS GetPIPMODE(UBYTE \*pu8Val);

### Example

```
Status = GetPIPMODE(pu8Val);
```

### Params

pu8Val : Current PIP mode

## Change PIP Size

### Description

Change the current PIP Size.

### API

STATUS SetPIPSize(UBYTE u8PIPSizeStatus, SWORD16 s16Height);

### Example

```
Status = SetPIPSize(u8PIPSizeStatus, s16Height);
```

### Params

- u8PIPPositionStatus : PIP Size identifiers whose value ranges from 0-3  
0 : Maximum  
1 : Default  
2 : Minimum  
3 : Custom Size
- s16Height : Height of Custom PIP Size ranges from 300 to 1080. Width will be automatically calculated.

## Read PIP Size

### Description

Read the current PIP Size.

### API

STATUS GetPIPSize(UBYTE \*pu8PIPSizeStatus, SWORD16 \*ps16XValue, SWORD16 \*ps16YValue);

### Example

```
Status = GetPIPSize(pu8PIPSizeStatus, ps16XValue, ps16YValue);
```

#### **Params**

- pu8PIPSizeStatus : Current PIP Size Status.
- ps16XValue : Width of PIP if Custom Size
- ps16YValue : Height of PIP if Custom Size

### **Change PIP Primary Input Source**

#### **Description**

Change PIP input source.

#### **API**

STATUS SetPIPPrimaryInput(UBYTE u8Val);

#### **Example**

```
Status = SetPIPPrimaryInput(u8Val);
```

#### **Params**

- u8Val : PIP Primary input source value to be set.  
Ranges from 0 to 4. Default values are
  - 0 : DP1
  - 1 : DP2
  - 2 : HDMI 1
  - 3 : HDMI 2
  - 4 : USB Type -C

### **Read PIP Primary Input Source**

#### **Description**

Read the current PIP Primary input.

#### **API**

STATUS GetPIPPrimaryInput(UBYTE \*pu8Val);

#### **Example**

```
Status = GetPIPPrimaryInput(pu8Val);
```

#### **Params**

- u8Val : Current PIP Primary input

### **Change PIP Secondary Input Source**

#### **Description**

Change PIP Secondary input source.

#### **API**

STATUS SetPIPSecondaryInput(UBYTE u8Val);

#### **Example**

```
Status = SetPIPSecondaryInput(UBYTE u8Val);
```

### Params

- u8Val : PIP Primary input source value to be set.  
Ranges from 0 to 3. Default values are
  - 0 : DP1
  - 1 : DP2
  - 2 : HDMI 1
  - 3 : HDMI 2
  - 4 : USB Type-C

## Read PIP Secondary Input Source

### Description

Read the current PIP Secondary input.

### API

STATUS GetPIPSecondaryInput(UBYTE \*pu8Val);

### Example

```
Status = GetPIPSecondaryInput(pu8Val);
```

### Params

- u8Val : Current PIP Secondary input

## Swap PIP and Inputs

### Description

Swap PIP and video inputs.

### API

STATUS SetSwapPIPInputsStatus(UBYTE u8Val)

### Example

```
Status = SetSwapPIPInputsStatus(u8Val);
```

### Params

- u8Val : PIP Swap input status

## Read Swap PIP Input Status

### Description

Read the current status of swap PIP/Primary inputs

### API

STATUS GetSwapPIPInputStatus(UBYTE \*pu8Val)

### Example

```
Status = GetSwapPIPInputStatus(pu8Val);
```

#### **Params**

- pu8Val : Current swap Status

Values – 0(OFF) and 1(ON)

### **Change PIP Video Legal Status**

#### **Description**

Change the current PIP Video Legal status.

#### **API**

STATUS SetPIPVideoLegalStatus(UBYTE u8Val)

#### **Example**

```
Status = SetPIPVideoLegalStatus(u8Val);
```

#### **Params**

- u8Val : PIP Video Legal Status

Values – 0(OFF) and 1(ON)

### **Read PIP Video Legal Status**

#### **Description**

Read the current PIP video legal status.

#### **API**

STATUS GetPIPVideoLegalStatus(UBYTE \*pu8Val)

#### **Example**

```
Status = GetPIPVideoLegalStatus(pu8Val);
```

#### **Params**

pu8Val : Current PIP video legal status, ON or OFF

### **Change PIP Overscan Status**

#### **Description**

Change the current PIP Overscan Status.

#### **API**

STATUS SetPIPOverscan(UBYTE u8Val)

#### **Example**

```
Status = SetPIPOverscan(u8Val);
```

#### **Params**

- u8Val : PIP Overscan status

Values – 0(OFF) and 1(ON)

### **Read PIP Overscan Status**

#### **Description**

Read the current PIP Overscan status.

#### **API**

STATUS GetPIPOverscanStatus(UBYTE \*pu8Val)

### Example

```
Status = GetPIPOverscanStatus(pu8Val);
```

### Params

- pu8Val : Current PIP overscan status, ON or OFF

## Change PIP DCI Status

### Description

Change the current PIP DCI option status.

### API

```
STATUS SetPIPDciStatus (SWORD16 s16Val)
```

### Example

```
Status = SetPIPDciStatus(s16Val);
```

### Params

- s16Val : PIP DCI region option to be set  
Ranges from 0 to 2. The values are:  
0 : Show Entire DCI Container  
1 : Crop to DCI 1.85:1 Aspect Ratio  
2 : Crop to DCI 2.39:1 Aspect Ratio

## Read PIP DCI Status

### Description

Read the current PIP DCI option status.

### API

```
STATUS GetPIPDciStatus(UBYTE *pu8Val)
```

### Example

```
Status = GetPIPDciStatus(&pu8Val);
```

### Params

- pu8Val : Current PIP DCI option

## Change PIP Marker Details

### Description

Change the PIP marker details.

### API

```
STATUS SetPIPMarkerDetails(const MarkerStatus* psPIPMarkerDetails);
```

### Example

```
Status = SetPIPMarkerDetails(pPIPMarkerDetails);
```

#### **Params**

- psPIPMarkerDetails: pointer of type MarkerStatus structure filled accordingly.  
Note : “ClearAll” Member of the structure should be always 0 for this API.

### **Read PIP Marker Details**

#### **Description**

Read the PIP marker details.

#### **API**

STATUS GetPIPMarkerDetails( MarkerStatus\* psPIPMarkerDetails);

#### **Example**

```
Status = GetPIPMarkerDetails(psMarkerDetails);
```

#### **Params**

- psPIPMarkerDetails: current settings of PIP marker.

### **Change PIP Marker Color**

#### **Description**

Change the PIP marker color.

#### **API**

STATUS SetPIPMarkerColor(UBYTE u8PIPMarkerColor);

#### **Example**

```
Status = SetPIPMarkerColor(u8PIPMarkerColor);
```

#### **Params**

- u8PIPMarkerColor: PIP Marker color. Available colors are
  - 0 : White
  - 1 : Red
  - 2 : Green
  - 3 : Blue
  - 4 : Cyan
  - 5 : Majenta
  - 6 : Yellow

### **Read PIP Marker Color**

#### **Description**

Read the PIP marker Color.

#### **API**

STATUS GetPIPMarkerColor(UBYTE \*pu8MarkerColor);

#### **Example**

```
Status = GetPIPMarkerColor(pu8MarkerColor);
```



**Params**

- pu8MarkerColor : Current PIP marker Color

**Clear PIP Markers****Description**

Clear all PIP marker settings.

**API**

STATUS ClearPIPMarkers();

**Example**

Status = ClearPIPMarkers();

## Management APIs

**Change Auto sleep mode****Description**

Enable/Disable the auto sleep mode. When there is no input signal, display goes to sleep mode based on this setting.

**API**

STATUS SetAutoSleepMode(UBYTE u8Val)

**Example**

Status = SetAutoSleepMode(u8Val);

**Params**

- u8Val : Enable/Disable the mode  
0 or 1 can be set.

**Read Auto sleep mode****Description**

Read the current setting of auto sleep mode. This is enabled by default so that display goes to sleep, when there is no input signal.

**API**

STATUS GetAutoSleepMode(UBYTE \*pu8Val)

**Example**

Status = GetAutoSleepMode(pu8Val);

**Params**

- pu8Val : Current setting for auto sleep mode

**Change Power on recall status****Description**

Change the current status of power on recall so as to control the display's ability to restore power to the unit, following unexpected removal of the supply AC voltage.

**API**

STATUS SetPowerOnRecallStatus(UBYTE u8Val)

**Example**

```
Status = SetPowerOnRecallStatus(u8Val);
```

**Params**

- u8Val : Enable/Disable the mode  
0 or 1 can be set.

**Read Power on recall status****Description**

Read the current setting of power on recall status. This is enabled by default.

**API**

```
STATUS GetPowerOnRecallStatus(UBYTE *pu8Val)
```

**Example**

```
Status = GetPowerOnRecallStatus(pu8Val);
```

**Params**

- pu8Val : Current setting of power on recall

**Change DDC/CI Communication status****Description**

Change the current status of DDC/CI communication.

**API**

```
STATUS SetDDCCICommnStatus(UBYTE u8Val)
```

**Example**

```
Status = SetDDCCICommnStatus(u8Val);
```

**Params**

- u8Val : Enable/Disable the mode  
0 or 1 can be set.

**Read DDC/CI Communication status****Description**

Read the current setting of DDC/CI communication. This is enabled by default.

**API**

```
STATUS GetDDCCICommnStatus(UBYTE *pu8Val)
```

**Example**

```
Status = GetDDCCICommnStatus(pu8Val);
```

**Params**

- pu8Val : Current setting of DDC/CI communication

**Change Auto EDID update status****Description**

Change the current status of Auto EDID update, to control the display EDID update during color space switching.

**API**

```
STATUS SetAutoEDIDUpdateStatus(UBYTE u8Val)
```

### Example

```
Status = SetAutoEDIDUpdateStatus(u8Val);
```

### Params

- u8Val : Enable/Disable the mode  
0 or 1 can be set.

## Read Auto EDID update status

### Description

Read the current setting of AutoEDID update.

### API

```
STATUS GetAutoEDIDUpdateStatus(UBYTE *pu8Val)
```

### Example

```
Status = GetAutoEDIDUpdateStatus(pu8Val);
```

### Params

- pu8Val : Current setting of Auto EDID update

## Change Hot Plug Initiate status

### Description

Change the current status of Hot Plug initiate. Determines whether or not a Hot Plug Event should occur when the color space switching is done.

### API

```
STATUS SetHotPlugInitiateStatus(UBYTE u8Val)
```

### Example

```
Status = SetHotPlugInitiateStatus(u8Val);
```

### Params

- u8Val : Enable/Disable the mode  
0 or 1 can be set.

## Read Hot Plug Initiate status

### Description

Read the current setting of Hot Plug Initiate. This is disabled by default.

### API

```
STATUS GetHotPlugInitiateStatus(UBYTE *pu8Val)
```

### Example

```
Status = GetHotPlugInitiateStatus(pu8Val);
```

### Params

- pu8Val : Current setting of Hot Plug Initiate

## Change DP HPD Status

### Description

Change the current status of DP Hot Plug. This is to prevent the computer from thinking the monitor has been disconnected when it is soft-powered off. If the user experiences issues – especially in multiple monitor configurations – with OS windows

jumping from monitor to monitor, when the monitor resumes from Off or Sleep mode they can change this setting to Always Active and eliminate this issue.

#### API

STATUS SetDPPHPDStatus(UBYTE u8Val)

#### Example

```
Status = SetDPPHPDStatus(u8Val);
```

#### Params

- u8Val : Low Power and Always Active
- 1 (Low Power) or 0 (Always Active) can be set.

## Read DP HPD Status

#### Description

Read the current setting of DP HPD.

#### API

STATUS GetDPPHPDStatus(UBYTE \*pu8Val)

#### Example

```
Status = GetDPPHPDStatus(pu8Val);
```

#### Params

- pu8Val : Current setting of DP HPD, Low Power or Always Active mode.

## Change Firmware update status

#### Description

Change the status of firmware update. The options available are:

- 1 : Enable, Via USB or Network
- 2 : Enable, Via USB Only
- 3 : Enable, Via Network Only
- 4 : Disable

#### API

STATUS SetFirmwareUpdateStatus(UBYTE u8Val)

#### Example

```
Status = SetFirmwareUpdateStatus(u8Val);
```

#### Params

- u8Val : Firmware update status to be set
- Ranges from 0 to 3. The values are
- 0 : Enable, Via USB or Network
- 1 : Enable, Via USB Only
- 2 : Enable, Via Network Only
- 3 : Disable

## Read Firmware update status

#### Description

Read the current setting of firmware update. Default option is 'Enable, Via USB or Network'.

#### **API**

STATUS GetFirmwareUpdateStatus(UBYTE \*u8Val)

#### **Example**

```
Status = GetFirmwareUpdateStatus(u8Val);
```

#### **Params**

- u8Val : current setting of firmware update

### **Change Calibration status**

#### **Description**

Change the current status of Calibration. This control only affects user calibration. Factory calibration will always run if the display is in Factory or Service mode, regardless of the setting of this control.

#### **API**

STATUS SetCalibrationStatus(UBYTE u8Val)

#### **Example**

```
Status = SetCalibrationStatus(u8Val);
```

#### **Params**

- u8Val : Enable/Disable the mode  
0 or 1 can be set.

### **Read Calibration status**

#### **Description**

Read the current setting of Calibration status. This is enabled by default.

#### **API**

STATUS GetCalibrationStatus(UBYTE \*u8Val)

#### **Example**

```
Status = GetCalibrationStatus(u8Val);
```

#### **Params**

- u8Val : Current setting of Calibration

### **Change WS-Management status**

#### **Description**

Change the WS Management status to enable or disable. This is disabled by default.

#### **API**

STATUS SetWSManagementStatus(UBYTE u8Val);

#### **Example**

```
Status = SetWSManagementStatus(u8Val);
```

#### **Params**

- u8Val : enable 0 and disable 1.

### **Read WS-Management status**

#### **Description**

Read the WS Management status. This is disabled by default.

#### **API**

STATUS GetWSManagementStatus(UBYTE \*pu8Val);

#### **Example**

```
Status = GetWSManagementStatus(pu8Val);
```

#### **Params**

- u8Val : Current status of WS management.

### **Change WS Identify Support status**

#### **Description**

Change the WS Identify status to enable or disable. This is disabled by default.

#### **API**

STATUS SetWSIdentifySupportStatus(UBYTE u8Val);

#### **Example**

```
Status = SetWSIdentifySupportStatus (u8Val);
```

#### **Params**

- u8Val : enable 0 and disable 1.

### **Read WS Identify Support status**

#### **Description**

Read the WS identify support status. This is disabled by default.

#### **API**

STATUS GetWSIdentifySupportStatus(UBYTE \*pu8Val);

#### **Example**

```
Status = GetWSIdentifySupportStatus(pu8Val);
```

#### **Params**

- u8Val : Current status of WS Identify support.

### **Change Web Server status**

#### **Description**

Change the Web server status to enable or disable. This is disabled by default.

#### **API**

STATUS SetWebServerStatus(UBYTE u8Val);

#### **Example**

```
Status = SetWebServerStatus(u8Val);
```

#### **Params**

- u8Val : enable 0 and disable 1.

### **Read Web Server status**

#### **Description**

Read the Web server status. This is disabled by default.

#### **API**

STATUS GetWebServerStatus(UBYTE \*pu8Val);

**Example**

```
Status = GetWebServerStatus(pu8Val);
```

**Params**

- u8Val : Current status of Web server.

**Change Dashboard Security****Description**

Change the Web dashboard security type

**API**

```
STATUS SetDashboardSecurity(UBYTE u8dashBoardStatus);
```

**Example**

```
Status = SetDashboardSecurity(u8dashBoardStatus);
```

**Params**

- u8Val :
  - 0 : Require Password only
  - 1 : Require Password and client certificate

**Read Web Dashboard Security****Description**

Read the Web dashboard security type.

**API**

```
STATUS GetDashboardSecurityStatus(UBYTE *u8dashBoardStatus);
```

**Example**

```
Status = GetDashboardSecurityStatus(u8dashBoardStatus);
```

**Params**

- u8Val : Current status of Web dashboard security type.

**Change IP configuration****Description**

Change the current IP configuration.

**API**

```
STATUS SetIPConfiguration(const IPConfig *psIPConfig);
```

**Example**

```
Status = SetIPConfiguration(psIPConfig);
```

**Params**

- psIPConfig : structure containing IP config details.
  - IP address, Gateway and Subnet Mask can be changed only for Static IP Mode.

**Read IP configuration****Description**

Read the IP configuration and MAC address assigned to the display.

#### **API**

STATUS GetIPConfiguration(IPConfig \*psIPConfig , BYTE \*pbyMacAddress);

#### **Example**

```
Status = GetIPConfiguration(psIPConfig , pbyMacAddress);
```

#### **Params**

- pbyMacAddress : Current MAC Address String.  
BYTE array of size 20
- psIPConfig : Current IP configuration

### **Switch OFF Manage Internal processor**

#### **Description**

Switches OFF the Internal Processor. Further communication via USB will be stopped after calling this API.

#### **API**

STATUS PowerOffManageInternalProcessor()

#### **Example**

```
Status = PowerOffManageInternalProcessor();
```

#### **Params**

- Nil

### **Change Date and Time**

#### **Description**

Change the date and time of the Z31x/Z27x-G2 monitor.

#### **API**

STATUS SetDateAndTime(const ClockSettings \*psClockSettings);

#### **Example**

```
Status = SetDateAndTime(psClockSettings);
```

#### **Params**

- psClockSettings : structure containing date and time details.

### **Read Date and Time**

#### **Description**

Read the current date and time of Z31x/Z27x-G2 Monitor.

#### **API**

STATUS GetDateAndTime(ClockSettings \*psClockSettings);

#### **Example**

```
Status = GetDateAndTime(psClockSettings);
```

#### **Params**

- psClockSettings: Current Date and time details

### **Change Auto Adjust Date and Time status**

#### **Description**

Change the Auto adjust date and time status of the Z31x/Z27x-G2 monitor.



#### API

```
STATUS SetAutoAdjustDateAndTimeStatus(UBYTE u8AutoAdjustDateTime);
```

#### Example

```
Status = SetAutoAdjustDateAndTimeStatus(u8AutoAdjustDateTime);
```

#### Params

- u8AutoAdjustDateTime : 0(OFF), 1 (ON)

### Read Auto Adjust Date and Time Status

#### Description

Read the current auto adjust date and time status of Z31x/Z27x-G2 Monitor.

#### API

```
STATUS GetAutoAdjustDateAndTimeStatus(UBYTE *pu8AutoAdjustDateTime);
```

#### Example

```
Status = GetAutoAdjustDateAndTimeStatus(pu8AutoAdjustDateTime);
```

#### Params

- pu8AutoAdjustDateTime: Current Auto Adjust Date and time status

### Change Auto Adjust Daylight Saving status

#### Description

Change the Auto adjust for daylight saving status of the Z31x/Z27x-G2 monitor.

#### API

```
STATUS SetAutoAdjustDayLightSavingStatus(UBYTE u8AutoAdjustDayLightSaving);
```

#### Example

```
Status = SetAutoAdjustDayLightSavingStatus(u8AutoAdjustDayLightSaving);
```

#### Params

- u8AutoAdjustDayLightSaving : 0(OFF), 1 (ON)

### Read Auto Adjust Daylight Saving Status

#### Description

Read the current auto adjust for daylight saving of Z31x/Z27x-G2 Monitor.

#### API

```
STATUS GetAutoAdjustDayLightSavingStatus(UBYTE *pu8AutoAdjustDayLightSaving);
```

#### Example

```
Status = GetAutoAdjustDayLightSavingStatus(pu8AutoAdjustDayLightSaving);
```

#### Params

- pu8AutoAdjustDayLightSaving: Current Auto Adjust for daylight saving status

### Change Device Time Format

#### Description

Change time format of the Z31x/Z27x-G2 monitor.

#### API

STATUS SetDeviceTimeFormat(UBYTE u8TimeFormatStatus);

#### Example

```
Status = SetDeviceTimeFormat(u8TimeFormatStatus);
```

#### Params

- u8TimeFormatStatus: 0(12 hour format), 1 (24 hour format)

### Read Device Time Format

#### Description

Read the current time format of Z31x/Z27x-G2 Monitor.

#### API

STATUS GetDeviceTimeFormat(UBYTE \*pu8TimeFormatStatus);

#### Example

```
Status = GetDeviceTimeFormat(pu8TimeFormatStatus);
```

#### Params

- pu8TimeFormatStatus: Current time format

### Change External Calibration Device Support

#### Description

Change external Calibration device support of Z31x/Z27x-G2 monitor.

#### API

STATUS SetAllowExternalInstrumentStatus(UBYTE u8EnableExternalInstrumentStatus);

#### Example

```
Status = SetAllowExternalInstrumentStatus(u8EnableExternalInstrumentStatus);
```

#### Params

- u8EnableExternalInstrumentStatus: 0(OFF), 1 (ON)

### Read External Calibration Device Support

#### Description

Read the current external Calibration device support status of Z31x/Z27x-G2 monitor.

#### API

STATUS GetAllowExternalInstrumentStatus(UBYTE \*pu8EnableExternalInstrumentStatus);

#### Example

```
Status = GetAllowExternalInstrumentStatus(pu8EnableExternalInstrumentStatus);
```

#### Params

- pu8EnableExternalInstrumentStatus: Current Allow use of External instrument status

### Change User Preset Support Status

#### Description

Change user preset support status of the Z31x/Z27x-G2 monitor.

#### API

STATUS SetUserPresetSupportStatus(UBYTE u8EnableUserPresetSupportStatus);

### Example

```
Status = SetUserPresetSupportStatus(u8EnableUserPresetSupportStatus);
```

### Params

- u8EnableUserPresetSupportStatus: 0(Disable), 1 (Enable)

## Read User Preset Support Status

### Description

Read the user preset support status of Z31x/Z27x-G2 Monitor.

### API

```
STATUS GetUserPresetSupportStatus(UBYTE *pu8UserPresetSupportStatus);
```

### Example

```
Status = GetUserPresetSupportStatus(pu8UserPresetSupportStatus);
```

### Params

- pu8UserPresetSupportStatus : Current user preset support status

## Change USB Type-C Input Configuration

### Description

Change USB Type-C Input configuration. This API is not available for Z27x-G2.

### API

```
STATUS SetUSBCInputConfiguration(UBYTE u8USBCConfig);
```

### Example

```
Status = SetUSBCInputConfiguration(u8USBCConfig);
```

### Params

- u8USBCConfig: Supported values are:  
0 : 4096 x 2160 60 Hz + USB 2.0 data  
1 : 4096 x 2160 30 Hz + USB 3.0 data

## Read USB Type-C Input Configuration

### Description

Read current USB Type-C Input configuration.

### API

```
STATUS GetUSBCInputConfiguration(UBYTE *pu8USBCConfig);
```

### Example

```
Status = GetUSBCInputConfiguration(pu8USBCConfig);
```

### Params

- pu8USBCConfig: Current USB Type-C Input configuration

## Change Time Zone

### Description

Change Time Zone of Z31x/Z27x-G2 Monitor

### API

```
STATUS SetTimezone(UBYTE u8Timezoneinfo);
```

**Example**

```
Status = SetTimezone(u8Timezoneinfo);
```

## Params

- u8Timezoneinfo : Timezone value ranges from 0 to 45.
  - 0- International Date Line West
  - 1- Coordinated Universal Time - 11
  - 2- Hawaii
  - 3- Alaska
  - 4- Pacific Time
  - 5- Mountain Time
  - 6- Arizona
  - 7- Central Time
  - 8- Eastern Time
  - 9- Indiana
  - 10- Bogota, Lima, Quinto, Branco
  - 11- Guadalajara, Mexico City
  - 12- Caracas
  - 13- Atlantic Time(Canada)
  - 14- Santiago
  - 15- Newfoundland
  - 16- Greenland
  - 17- Brasilia
  - 18- Coordinated Universal Time - 02
  - 19- Azores
  - 20- Coordinated Universal Time
  - 21- London, Dublin, Edinburg
  - 22- Berlin, Paris, Rome
  - 23- West Central Africa
  - 24- Athens, Cairo, E. Europe
  - 25- Harare, Pretoria
  - 26- Moscow, St.Petersberg
  - 27- Tehran
  - 28- Abu Dhabi,Muscat
  - 29- Kabul
  - 30- Chennai, Mumbai, New Delhi
  - 31- Islamabad, Karachi
  - 32- Coordinated Universal Time + 06
  - 33- Krasnoyarsk
  - 34- Bangkok, Hanoi, Jakarta
  - 35- Beijing, Hong Kong, Taipei
  - 36- Perth, Singapore
  - 37- Osaka, Tokyo, Seoul
  - 38- Adelaide, Darwin
  - 39- Brisbane, Melbourne, Sydney
  - 40- Petropavlovsk-Kamchatka
  - 41- Solomon Islands
  - 42- Auckland, Wellington
  - 43- Coordinated Universal Time + 12
  - 44- Samoa
  - 45- Kiritimati

## Read Time Zone

### Description

Read Time Zone of Z31x/Z27x-G2 Monitor.

### API

```
STATUS GetTimezone(UBYTE *pu8Timezoneinfo);
```

### Example

```
Status = GetTimezone(pu8Timezoneinfo);
```

### Params

pu8Timezoneinfo: Current TimeZone.

## Lock Management Menu

### Description

Lock Management Menu in OSD.

### API

```
STATUS LockManagementMenu(UBYTE u8Val);
```

### Example

```
Status = LockManagementMenu(u8Val);
```

### Params

– u8Val: Supported values are 0(Unlock), 1(Lock)

## Align Internal Calibration Device

### Description

Align Internal Calibration device to the reference device.

### API

```
STATUS AlignInternalCalibrationDevice();
```

### Example

```
Status = AlignInternalCalibrationDevice();
```

### Params

Nil

## Change USB Functionality during sleep

### Description

Enable/Disable USB ports during sleep.

### API

```
STATUS SetUSBSleepFunctionality(UBYTE u8Val)
```

### Example

```
Status = SetUSBSleepFunctionality(u8Val);
```

#### **Params**

- u8Val : Enable/Disable the USB ports during device sleep.  
0 or 1 can be set.

### **Read USB Functionality during sleep**

#### **Description**

Read the current setting of USB functionality during sleep.

#### **API**

STATUS GetUSBSleepFunctionality(UBYTE \*pu8Val)

#### **Example**

```
Status = GetUSBSleepFunctionality(pu8Val);
```

#### **Params**

pu8Val : Current setting for USB port status

### **Change DisplayPort EDID Configuration**

#### **Description**

Change Display Port EDID Configuration. This API is not available for Z27x-G2.

#### **API**

STATUS SetDPEDIDConfiguration(UBYTE u8Val)

#### **Example**

```
Status = SetDPEDIDConfiguration(u8Val);
```

#### **Params**

- u8Val : DisplayPort EDID Configuration Value  
0 - Display ID(4096 X 2160 default timing)  
1 - CEA-861(3840 X 2160 default timing)

### **Read Display Port EDID Configuration**

#### **Description**

Read the current setting of Display Port EDID Configuration. This API is not available for Z27x-G2.

#### **API**

STATUS GetDPEDIDConfiguration (UBYTE \*pu8Val)

#### **Example**

```
Status = GetDPEDIDConfiguration (pu8Val);
```

#### **Params**

pu8Val : Current setting for DisplayPort EDID Configuration

### **Change Luminance Uniformity**

#### **Description**

Enable/Disable luminance uniformity.

#### API

STATUS SetLuminanceUniformityStatus(UBYTE u8Val)

#### Example

```
Status = SetLuminanceUniformityStatus(u8Val);
```

#### Params

- u8Val : Enable/Disable Luminance Uniformity.  
0 or 1 can be set.

### Read Luminance Uniformity

#### Description

Read the current setting of Luminance Uniformity.

#### API

STATUS GetLuminanceUniformityStatus(UBYTE \*pu8Val)

#### Example

```
Status = GetLuminanceUniformityStatus(pu8Val);
```

#### Params

pu8Val : Current setting for Luminance Uniformity.

### Factory Reset

#### Description

Reset the Z31x/Z27x-G2 display to its factory settings.

#### API

STATUS FactoryReset()

#### Example

```
Status = FactoryReset();
```

#### Params

- Nil

## Menu and Message Control APIs

### Change OSD Position

#### Description

OSD Position can be changed by specifying the horizontal and vertical position.

#### API

STATUS SetOSDPosition(SWORD16 X, SWORD16 Y)

#### Example

```
Status = SetOSDPosition(X, Y)
```



#### **Params**

- X : Horizontal position x to be set  
Ranges from 0 to 100
- Y : Vertical position y to be set  
Ranges from 0 to 100

### **Read OSD Position**

#### **Description**

Read the current horizontal and vertical pixel position of OSD

#### **API**

STATUS GetOSDPosition(SWORD16 X, SWORD16 Y)

#### **Example**

```
Status = GetOSDPosition(X, Y)
```

#### **Params**

- X : Horizontal position x
- Y : Vertical position y

### **Change OSD Opacity**

#### **Description**

Adjust OSD opacity value. The minimum value the opacity can be set to be 25 and the maximum value can be set is 100. The default value for timeout is 100.

#### **API**

STATUS SetOSDOpacity(UBYTE u8Val)

#### **Example**

```
Status = SetOSDOpacity(u8Val);
```

#### **Params**

- u8Val : opacity value can be set  
Ranges from 25 to 100

### **Read OSD Opacity**

#### **Description**

Adjust OSD opacity value. The minimum value the opacity can be set to be 25 and the maximum value can be set is 100. The default value for timeout is 100.

#### **API**

STATUS GetOSDOpacity(UBYTE \*pu8Val)

#### **Example**

```
Status = GetOSDOpacity(pu8Val);
```

#### **Params**

- u8Val : opacity value  
Ranges from 25 to 100

### **Change OSD Timeout**

#### **Description**

Adjust OSD timeout value. The minimum value the timeout can be set to be 5 seconds and the maximum value can be set to be 60 seconds. The default value for timeout is 15 seconds.

#### **API**

STATUS SetOSDTimeout(UBYTE u8Val)

#### **Example**

```
Status = SetOSDTimeout(u8Val);
```

#### **Params**

- u8Val : Timeout value can be set  
Ranges from 5 to 60. Multiples of 5 will be only allowed.

### **Read OSD Timeout**

#### **Description**

Read the current OSD timeout value

#### **API**

STATUS GetOSDTimeout(UBYTE \*u8Val)

#### **Example**

```
Status = GetOSDTimeout(u8Val);
```

#### **Params**

- u8Val : Current OSD timeout

### **Change Power on Information OSD Settings**

#### **Description**

Change the power on info OSD settings. The power on information OSD will only be displayed if the value is set to 1. The OSD will not be displayed if the value is set to 0.

#### **API**

STATUS SetShowPowerOnInfo(UBYTE u8Val)

#### **Example**

```
Status = SetShowPowerOnInfo(u8Val);
```

#### **Params**

- u8Val : Value can be 0 or 1

### **Read Power on Information OSD Settings**

#### **Description**

Get the power on info OSD status. The power on information OSD will only be displayed if the value is set to 1. The OSD will not be displayed if the value is set to 0.

#### **API**

STATUS GetShowPowerOnInfoStatus(UBYTE \*pu8Val)

### Example

```
Status = GetShowPowerOnInfoStatus(pu8Val);
```

### Params

- u8Val : Value can be 0 or 1

## Change Input source Information OSD Settings

### Description

Change the Input source info OSD settings. The input source information OSD will only be displayed if the value is set to 1. The OSD will not be displayed if the value is set to 0

### API

STATUS SetShowInputSourceInfo(UBYTE u8Val)

### Example

```
Status = SetShowInputSourceInfo(u8Val);
```

### Params

- u8Val : Value can be 0 or 1

## Read Input Source Information OSD Settings

### Description

Get the input source info OSD status. The input source information OSD will only be displayed if the value is set to 1. The OSD will not be displayed if the value is set to 0.

### API

STATUS GetShowInputSourceInfoStatus(UBYTE \*pu8Val)

### Example

```
Status = GetShowInputSourceInfoStatus(pu8Val);
```

### Params

- u8Val : Value can be 0 or 1

## Change Colorspace Information OSD Settings

### Description

Change the Colorspace info OSD settings. The colorspace information OSD will only be displayed if the value is set to 1. The OSD will not be displayed if the value is set to 0.

### API

STATUS SetShowColorspaceInfoStatus(UBYTE u8Val)

### Example

```
Status = SetShowColorspaceInfoStatus(u8Val);
```

### Params

- u8Val : Value can be 0 or 1

## Read Colorspace Information OSD Settings

### Description

Get the colorspace info OSD status. The colorspace information OSD will only be displayed if the value is set to 1. The OSD will not be displayed if the value is set to 0.

## API

STATUS GetShowColorspaceInfoStatus(UBYTE \*pu8Val)

## Example

```
Status = GetShowColorspaceInfoStatus(pu8Val);
```

## Params

- u8Val : Value can be 0 or 1

## Change Warmup Time Reminder Info OSD Settings

### Description

Change the Warm up time info OSD settings. The warm up time information OSD will only be displayed if the value is set to 1. The OSD will not be displayed if the value is set to 0.

## API

STATUS SetShowWarmupTimeReminderStatus(UBYTE u8Val)

## Example

```
Status = SetShowWarmupTimeReminderStatus(u8Val);
```

## Params

- u8Val : Value can be 0 or 1

## Read WarmupTime Reminder Info OSD Settings

### Description

Get the warm up time reminder info OSD status. The warm up time information OSD will only be displayed if the value is set to 1. The OSD will not be displayed if the value is set to 0.

## API

STATUS GetShowWarmupTimeReminderStatus(UBYTE \*pu8Val)

## Example

```
Status = GetShowWarmupTimeReminderStatus(pu8Val);
```

## Params

- u8Val : Value can be 0 or 1

## Change Calibration Notification Info OSD Settings

### Description

Change the Calibration notification info OSD settings. The calibration process notification information OSD will only be displayed if the value is set to 1. The OSD will not be displayed if the value is set to 0.

## API

STATUS SetShowCalibrationNotificationStatus(UBYTE u8Val)

## Example

```
Status = SetShowCalibrationNotificationStatus(u8Val);
```

### Params

- u8Val : Value can be 0 or 1

## Read Calibration Notification Info OSD Settings

### Description

Get the calibration notification info OSD status. The calibration process notification information OSD will only be displayed if the value is set to 1. The OSD will not be displayed if the value is set to 0.

### API

STATUS GetShowCalibrationNotificationStatus(UBYTE \*pu8Val)

### Example

```
Status = GetShowCalibrationNotificationStatus(pu8Val);
```

### Params

- u8Val : Value can be 0 or 1

## Change Function Button Settings

### Description

Change the function button configuration of a particular bezel button. Set one of the sixteen different commands to a bezel button. Depending on the command, some will immediately execute a function while others will display a menu.

### API

STATUS SetFunctionButtonSettings(SWORD16 s16Button, SWORD16 s16Val)

### Example

```
Status = SetFunctionButtonSettings(s16Button, s16Val);
```

## Params

- s16Button: Bezel Button Id to be assigned with  
Ranges from
  - 1 – Top menu button
  - 2 – Second menu button (just below top button)
  - 3 – Third Menu button
  - 4 – Fourth menu button (just above bottom button)
- s16Val  
Value to be set for the Bezel button, Ranges from 1 -17  
Function Button Commands and their assigned values:
  - 0 – Color Space Select
  - 1 – Adjust Luminance
  - 2 – Low Blue Light
  - 3 – Video Input
  - 4 – Switch to Next Active Video Input
  - 5 – Image Scaling
  - 6 – Digital Cinema masking
  - 7 – Markers
  - 8 – Video Levels (16–235) On / Off
  - 9 – Overscan On / Off
  - 10 – Blue-Only Mode On / Off
  - 11 – Dual split On / Off
  - 12 – PIP On / Off
  - 13 – Swap Primary / PIP Inputs
  - 14 – Show Display Information
  - 15 – Show Color Space Information
  - 16 – Load User Preset
  - 17 – Test Pattern Generator
  - 18 – Empty (No Function)

## Read Function Button Settings

### Description

Read the function button configuration assigned to a bezel button.

### API

STATUS GetFunctionButtonSettings(SWORD16 s16Button, SWORD16 \*ps16Val)

### Example

```
Status = GetFunctionButtonSettings(s16Button, ps16Val);
```

### Params

- s16Button : Bezel Button Id to be assigned  
Ranges from:
  - 1 – Top menu button
  - 2 – Second menu button (just below top button)
  - 3 – Third Menu button
  - 4 – Fourth menu button (just above bottom button)
- ps16Val : Currently assigned command for the particular function button.

## Change Function Button Mode

### Description

Change the function button mode. This is to configure how the bezel buttons respond to touch. Two modes of operation are supported:

- Open Button Label First
- Execute Command on First Press

### API

STATUS SetFunctionButtonMode(UBYTE u8Val)

### Example

```
Status = SetFunctionButtonMode(u8Val);
```

### Params

- u8Val : Value can be 0 or 1,  
Open Button Label First (0) or Execute Command on First Press (1)

## Read Function Button Mode

### Description

Read the current function button mode.

### API

STATUS GetFunctionButtonMode(UBYTE \*pu8Val)

### Example

```
Status = GetFunctionButtonMode (pu8Val);
```

### Params

- u8Val : Value can be 0 or 1,  
Open Button Label First (0) or Execute Command on First Press (1)

## Change Bezel Button Color

### Description

Change the bezel button color to White, Red, or an auto-switch configuration for the illumination color.

### API

STATUS SetBezelButtonColor(UBYTE u8Val)

### Example

```
Status = SetBezelButtonColor(u8Val);
```

#### Params

- u8Val : Bezel button color, Value can be 0 , 1 or 2
  - 0 – Always White
  - 1 – Always Red
  - 2 – Auto-Switch to Red at 70 cd/m2

### Read Bezel Button Color

#### Description

Read the bezel button color setting.

#### API

STATUS GetBezelButtonColor(UBYTE \*pu8Val)

#### Example

```
Status = GetBezelButtonColor(pu8Val);
```

#### Params

- pu8Val : Values 0, 1 or 2

### Change Bezel Button Brightness

#### Description

Change the bezel button brightness to the desired level. The following range is permitted:

- The range of adjustment should be specified in percentage
- The maximum value the luminance can be set to is 100%.
- The minimum value the luminance can be set to is 0%.
- The default value for bezel button brightness is 100%.

#### API

STATUS SetBezelButtonBrightness(UBYTE u8Val)

#### Example

```
Status = SetBezelButtonBrightness(u8Val);
```

#### Params

- u8Val : brightness value in %. Multiples of 10 will be only allowed.

### Read Bezel Button Brightness

#### Description

Read the current setting for bezel button brightness

#### API

STATUS GetBezelButtonBrightness(UBYTE \*pu8Val)

#### Example

```
Status = GetBezelButtonBrightness(pu8Val);
```

#### Params

- u8Val : brightness value in %

### Change Bezel Button Auto-Fade

#### Description

Change the bezel button auto fade property. The bezel button illumination can either auto-fade in sync with the OSD timeout or is always on. This is to keep the LEDs on, so that the buttons are easier to find in a dark environment.



## API

STATUS SetBezelButtonAutoFadeStatus(UBYTE u8Val)

## Example

```
Status = SetBezelButtonAutoFadeStatus(u8Val);
```

## Params

- u8Val : Enable (1) or Disable (0) can be set

## Read Bezel Button Auto-Fade

### Description

Read the bezel button auto fade status

## API

STATUS GetBezelButtonAutoFadeStatus(UBYTE \*pu8Val)

## Example

```
Status = GetBezelButtonAutoFadeStatus(pu8Val);
```

## Params

- pu8Val : Enable (1) or Disable (0)

## Change Menu Language

### Description

Change the OSD language.

## API

STATUS SetMenuLanguage(UBYTE u8Val)

## Example

```
Status = SetMenuLanguage(u8Val);
```

## Params

- u8Val : Menu language value  
Ranges from 0 to 9. The values are:  
0 : Deutsch  
1 : Traditional Chinese (in char)  
2 : Simplified Chinese (in char)  
3 : English  
4 : Espanol  
5 : Francais  
6 : Italiano  
7 : Japanese(in Kanji)  
8 : Nederlands  
9 : Portugues

## Read Menu Language

### Description

Read the current OSD language

## API

STATUS GetMenuLanguage(UBYTE \*pu8Val)

**Example**

```
Status = GetMenuLanguage(pu8Val);
```

**Params**

- u8Val : Menu language value  
Ranges from 0 to 9

## Information APIs

### Read Firmware revision

**Description**

Read the firmware version of Lumen display. Firmware version is expected to be LISxyz format.

**API**

```
STATUS GetFirmwareRevision(BYTE *pbyVal)
```

**Example**

```
Status = GetFirmwareRevision(pbyVal)
```

**Params**

- pbyVal : Firmware version of the display

### Read Display Serial Number

**Description**

Read the serial number of Lumen display. Serial number is expected to be 12 digits.

**API**

```
STATUS GetDisplaySerialNumber(BYTE *pbyVal)
```

**Example**

```
Status = GetDisplaySerialNumber(pbyVal)
```

**Params**

- pbyVal : Display Serial Number

### Read Display Mode

**Description**

Read the Display mode details of Lumen display.

**API**

```
STATUS GetDisplayMode(BYTE *pbyVal)
```

**Example**

```
Status = GetDisplayMode(pbyVal)
```

**Params**

- pbyVal : Display Mode

### Read Calibration hours

**Description**

Read the time after last calibration in hours.

#### **API**

STATUS GetCalibrationHours(SWORD16 \*ps16Val)

#### **Example**

Status = GetCalibrationHours(ps16Val)

#### **Params**

- ps16Val : Time in hours

### **Read Backlight hours**

#### **Description**

Read the backlight hours of the display.

#### **API**

STATUS GetBacklightHours(SWORD16 \*ps16Val)

#### **Example**

Status = GetBacklightHours(ps16Val)

#### **Params**

- ps16Val : Time in hours

### **Read Remote Customer ID Asset Tag**

#### **Description**

Read the Custom ID asset tag of the monitor

#### **API**

STATUS GetCustomerIDAsset(BYTE byCustomerAssetTag[MONITOR\_PROFIE\_STRING\_LENGTH])

#### **Example**

Status = GetCustomerIDAsset(byCustomerAssetTag)

#### **Params**

- CustomerAssetTag : ID Asset tag of the monitor

MONITOR\_PROFIE\_STRING\_LENGTH = 19

### **Read Display Name**

#### **Description**

Read the remote display of the monitor

#### **API**

STATUS GetDisplayName(BYTE byDisplayName[MONITOR\_PROFIE\_STRING\_LENGTH])

#### **Example**

Status = GetDisplayName(byDisplayName)

#### **Params**

- DisplayName : Remote Display name of the monitor

### **Read Deployment Location**

#### **Description**

Read the deployment location

#### **API**

STATUS GetDeploymentLocation(BYTE byDeploymentLocation[MONITOR\_PROFIE\_STRING\_LENGTH])

#### **Example**

```
Status = GetDeploymentLocation(byDeploymentLocation)
```

#### **Params**

- DeploymentLocation : Deployment location of the monitor

### **Read Deployment Date**

#### **Description**

Read the deployment date

#### **API**

STATUS GetDeploymentDate(BYTE \*DeploymentDate)

#### **Example**

```
Status = GetDeploymentDate(DeploymentDate)
```

#### **Params**

- DeploymentDate : Deployment date of the monitor

### **Set Log**

#### **Description**

Enable or disable logging in USB SDK

#### **API**

STATUS SetLog(UBYTE u8Val)

#### **Example**

```
Status = SetLog(mode)
```

#### **Params**

- mode : 0 – no logging; 1 – enable logging.

## **Test Pattern Generator APIs**

### **Enable Test Pattern**

#### **Description**

Enable Test Pattern for User Mode or Service Mode.

#### **API**

STATUS EnableTestPattern();

#### **Example**

```
Status = EnableTestPattern();
```

### **Disable Test Pattern**

#### **Description**

Disable Test Pattern for User Mode or Service Mode.

#### **API**

STATUS DisableTestPattern();

### Example

```
Status = DisableTestPattern();
```

## Read Test Pattern Status

### Description

Get Current Test Pattern Status.

### API

```
STATUS GetTestPatternStatus(UBYTE *pu8Val);
```

### Example

```
Status = GetTestPatternStatus(pu8Val);
```

### Params

- u8Val : Test pattern status
  - 0 : Patten Disabled
  - 1 : Service Mode Test Pattern active
  - 2 : User mode Test Pattern active
  - 3 : Test Pattern Enabled.

## Select Test Pattern: Service Mode

### Description

Test patterns can be selected from the service mode test pattern options listed below. Test patterns can be displayed using display test pattern API for service mode. Nine test patterns are available:

### API

```
STATUS ShowTestPatternServiceMode(TEST_PATTERN Color)
```

### Example

```
Status = ShowTestPatternServiceMode(Color);
```

### Params

- u8Val : Test pattern ID
  - Ranges from 0 to 8.
  - 0 : Full Screen Black
  - 1 : Full Screen 50% Gray
  - 2 : Full Screen White
  - 3 : Full Screen Red
  - 4 : Full Screen Green
  - 5 : Full Screen Blue
  - 6 : Full Screen Cyan
  - 7 : Full Screen Magenta
  - 8 : Full Screen Yellow
  - 9 : Grayscale Linear Ramp

## Select Test Pattern: User Mode

### Description

Test patterns to be displayed can be selected by setting the required R, G and B values. Test patterns can be displayed using display test pattern API. 8-bit RGB values from 0–255 are supported for each channel

### API

STATUS ShowTestPatternforUserMode (SWORD16 R, SWORD16 G, SWORD16 B)

### Example

```
Status = ShowTestPatternforUserMode(R, G, B);
```

### Params

- R, G, B : Value can be 0 to 255.

## Remote Configuration APIs

These functionalities are only supported via USB SDK API's

### Read Colormangement Status

#### Description

Read the current status of color management engine.

### API

STATUS GetColorManagementEngineStatus(UBYTE \*pu8Val)

### Example

```
Status = GetColorManagementEngineStatus(pu8Val);
```

### Params

pu8Val : Current status of the colormangement

### Read Calibration Alert Limit

#### Description

Read the current calibration due alert limit.

### API

STATUS GetCalibrationAlertLimit(UBYTE \*pu8Val)

### Example

```
Status = GetCalibrationAlertLimit(pu8Val);
```

### Params

- pu8Val : Current calibration due alert limit

### Read Dithering Mode Status

#### Description

Read the Dithering mode Status.

### API

STATUS GetDitheringModeStatus(UBYTE \*pu8Val);

### Example

```
Status = GetDitheringModeStatus(UBYTE *pu8Val);
```

### Params

pu8Val : Current dithering mode status

## SDK Version API

### Release Version

#### Description

Read the version string of USB SDK release

#### API

STATUS GetSdkVersion(BYTE \*Version)

#### Example

```
Status = GetSdkVersion(Version)
```

### Params

– Version : Read the version string of USB SDK release

## Calibration APIs

### Change Calibration schedule

#### Description

Change the calibration schedule settings.

#### API

STATUS SetCalibrationSchedule(const ScheduledCalibInfo\* psScheduledCalibInfo);

#### Example

```
Status = SetCalibrationSchedule(psScheduledCalibInfo);
```

### Params

– psScheduledCalibInfo: pointer to ScheduledCalibInfo structure filled accordingly.

### Read Calibration schedule

#### Description

Read the calibration schedule settings.

#### API

STATUS GetCalibrationSchedule(ScheduledCalibInfo\* psScheduledCalibInfo);

#### Example

```
Status = GetCalibrationSchedule(psScheduledCalibInfo);
```

### Params

– psScheduledCalibInfo : Current Calibration schedule details.

### Change Warmup Settings

#### Description

Change the warmup settings of monitor.

#### API

STATUS SetDisplayWarmupSettings(const WarmupSettings\* psWarmupSettings);

#### Example

```
Status = SetDisplayWarmupSettings(psWarmupSettings);
```

#### Params

psWarmupSettings: pointer to WarmupSettings structure filled accordingly.

### Read Calibration schedule

#### Description

Read the warmup settings of display.

#### API

STATUS GetDisplayWarmupSettings(WarmupSettings\* psWarmupSettings);

#### Example

```
Status = GetDisplayWarmupSettings(psWarmupSettings);
```

#### Params

— psWarmupSettings : Current Warmup Settings.

### Recalibrate Active Preset

#### Description

Recalibrate Active preset of the display. Only internal calibration device is supported.

#### API

STATUS RecalibrateActivePreset();

#### Example

```
Status = RecalibrateActivePreset();
```

### Configure and Calibrate Preset

#### Description

Configure and Calibrate particular preset.

#### API

STATUS ConfigureAndCalibratePreset(UBYTE u8colorPreset, UBYTE u8colorGamut, UBYTE u8whitePoint, UBYTE u8gamma, UBYTE u8luminance);

#### Example

```
Status = ConfigureAndCalibratePreset(u8colorPreset, u8colorGamut, u8whitePoint, u8gamma, u8luminance);
```



### Params

- u8colorPreset : ranges from 1 – 8.
- u8colorGamut : ranges from 1 – 6.
  - 1 : DCI P3
  - 2 : sRGB / BT.709
  - 3 : BT.2020
  - 4 : AdobeRGB
  - 5 : BT.601
  - 6 : Display Native Gamut
- u8whitePoint : ranges from 1-4
  - 1 : D65
  - 2 : D50
  - 3 : D55
  - 4 : DCI P3
- u8gamma : ranges from 1-6
  - 1 : STD\_GAMMA\_2\_2
  - 2 : VIDEO\_DISPLAY\_2\_4
  - 3 : DIGITAL\_CINEMA\_2\_6
  - 4 : BT\_1886\_2\_4
  - 5 : SRGB\_STD
  - 6 : EPD\_IMAGING
- u8luminance : ranges from 48 – 250.

## Send File to Display

### Description

Sends the xml/lut/user/monitor settings file to the display.

### API

SWORD16 SendFileToDisplay(FILE\_TYPE FileType, char \*FileName);

### Example

```
Status = SendFileToDisplay(fileType, &fileName);
```

### Params

- FileType : The type of the file that is going to send to the monitor. Eg: xml file or zip (lut) file.
- FileName : Path of the file.

## Validate Studio XML File

### Description

Validates the XML file and decides which action need to be taken.

### API

SWORD16 ValidateStudiocalXML (void);

### Example

```
Status = ValidateStudiocalXML();
```

## Perform Studio Calibration

### Description

Performs the calibration process according to the processed xml file. Calibration operations using internal colorimeter is only supported.

### API

SWORD16 PerformStudioCalibration (void);

### Example

```
Status = PerformStudioCalibration ();
```

## Get File from Display

### Description

Downloads the resultant file from the monitor to a specified path.

### API

SWORD16 GetFileFromDisplay (FILE\_TYPE FileType, char \*FolderName);

### Example

```
Status = GetFileFromDisplay(fileType, &folderName);
```

### Params

- FileType : The type of the file that is going to send to the monitor. Eg: xml file or zip (lut) file.
- FileName : Folder name to which file will be downloaded.

## Variable Definitions

### Datatype Definitions

```
typedef unsigned char    UBYTE;
typedef unsigned short   UWORD16;
typedef unsigned int     UWORD32;
typedef signed short     SWORD16;
typedef signed int       SWORD32;
```

### STATUS

This enum defines the return status

```
typedef enum
{
    STATE_ER_OPEN = -11
    STATE_ER_ENUM = -10
    STATE_ER_NO_DEV = -9
    STATE_ER_TIMEOUT = -8
    STATE_ER_CLAIM_INTERFACE = -7
    STATE_ER_INVALID_ARG = -6
    STATE_ER_NOT_CONNECTED = -5
    STATE_ER_COMM = -4
    STATE_ER_API = -3
    STATE_ER_MONITOR_MODE = -2
    FAILURE = -1
    SUCCESS = 0
}

STATUS;

struct __MarkerStatus
{
    UBYTE ClearAll;
    UBYTE AR_1_85_1;
    UBYTE AR_2_39_1;
    UBYTE Extraction_16_9;
    UBYTE ActionSafe_16_9;
    UBYTE TitleSafe_16_9;
    UBYTE Extraction_4_3;
    UBYTE ActionSafe_4_3;
    UBYTE TitleSafe_4_3;
    UBYTE Centre;
    UBYTE Thirds;
    UBYTE User;

};

typedef struct __MarkerStatus MarkerStatus;
struct __DateSettings
{
    UBYTE Month;
    UBYTE Day;
    SWORD16 Year;
};
```

```

typedef struct __DateSettings DateSettings;

struct __TimeSettings
{
    UBYTE Hour;
    UBYTE Minute;
};
typedef struct __TimeSettings TimeSettings;

struct __ClockSettings
{
    DateSettings sDate;
    TimeSettings sTime;
    UBYTE Timezoneinfo;
    UBYTE Timeformat;
    UBYTE TimeMeridian;
};
typedef struct __ClockSettings ClockSettings;

struct __WarmupSettings
{
    UBYTE WarmupEnableStatus;
    UBYTE WarmupTimeHour;
    UBYTE WarmupTimeMinute;
    UBYTE WarmuptimeMeridian;
    UBYTE WarmupEnableSunday;
    UBYTE WarmupEnableMonday;
    UBYTE WarmupEnableTuesday;
    UBYTE WarmupEnableWednesday;
    UBYTE WarmupEnableThursday;
    UBYTE WarmupEnableFriday;
    UBYTE WarmupEnableSaturday;
};
typedef struct __WarmupSettings WarmupSettings;
struct __ScheduledCalibInfo
{
    UBYTE ScheduleStatus;
    UBYTE FrequencyType;
    SWORD16 FrequencyCount;
    UBYTE DayofWeek;
    UBYTE TimeHH;
    UBYTE TimeMM;
    UBYTE TimeMeridian;
};
typedef struct __ScheduledCalibInfo ScheduledCalibInfo;
struct __IpConfig
{
    UBYTE IPMode;
    UBYTE IPAddress[IP_STRING_LENGTH];
    UBYTE SubnetMask[IP_STRING_LENGTH];
    UBYTE Gateway[IP_STRING_LENGTH];
} ;
typedef struct __IpConfig IPConfig;

```



Share with colleagues



Rate this document

