

# DNX64 DLL Control Reference

AnMo Electronics Corporation

---

## Contents

License agreement .....	3
Notice .....	4
Tutorial .....	5
Methods .....	7
EnableMicroTouch.....	7
FOVx .....	7
GetAETarget .....	7
GetAMR .....	7
GetAutoExposure .....	7
GetConfig.....	7
GetDeviceId .....	8
GetExposureValue .....	8
GetLensFinePosLimits.....	8
GetLensPosLimits .....	9
GetVideoDeviceIndex .....	9
GetVideoProcAmp.....	9
GetVideoProcAmpValueRange.....	9
GetWiFilmage .....	10
SetAETarget .....	10
SetAutoExposure .....	11
SetAimpointLevel .....	11
SetAXILevel .....	11
SetEFLC .....	11
SetExposureValue.....	12
SetLensFinePos .....	12
SetFLCLevel.....	12
SetFLCSwitch .....	13
SetLEDState .....	13
SetLensInitPos .....	13
SetLensPos.....	14

SetVideoDeviceIndex.....	14
SetVideoProcAmp.....	14
SetWiFiVideoRes .....	14
MicroTouchPressed.....	15

## License agreement

This Limited Use Software License Agreement (the "Agreement") is a legal agreement between you, the end-user ("Licensee"), and AnMo Electronics Corp ("AnMo"). By using or storing this program ("DNX64") on a computer hard drive or other media, you are agreeing to be bound by the terms of this Agreement.

Licensee may not alter this DNX64 in any way, including changing or removing any messages or windows.

Licensee may not decompile, reverse engineer, disassemble or otherwise reduce this DNX64 to a human perceivable form. Licensee may not modify, rent or resell the DNX64 for profit. Licensee may not publicize or distribute any registration code algorithms, information, or registration codes used by the DNX64 without permission of AnMo.

Licensee written applications containing embedded DNX64 control may be freely distributed, without royalty payments to AnMo, provided that such distributed product is bound into these applications in such a way so as to prohibit separate use in design mode, and that such product is distributed only in conjunction with the hardware manufactured by AnMo.

This DNX64 may be used as a constituent control only if the compound control thus created is distributed with and as an integral part of an application. The license may not be transferred to a third party under any circumstance.

This DNX64 is provided by AnMo on an "as is" basis. AnMo makes no warranty, expressed or implied, including without limitation the implied warranties of non-infringement, merchantability and fitness for a particular purpose, regarding the DNX64 or its use and operation alone or in combination with any product.

Under no circumstances shall AnMo be liable for any incidental or consequential damages, nor for any damages in excess of the original purchase price.

To the best of Vendor's knowledge, Customer's permitted use of the Licensed Software will not infringe the intellectual property rights of any third party.

Compliance with Applicable Law. Vendor warrants that the services provided under this Agreement and Customer's permitted use of the Licensed Software shall comply with applicable federal, state, and local laws and regulations.

## Notice

1. Please be sure the SDK is installed with Administrator's right.
2. The following files need to be included in your application setup:  
  
**DNX64.dll**, **libusbK.dll** and **DNX32.dll** are located in "DNX64" program folder, and need to be copied to Windows system folder on target PC.
3. Some built-in or external webcams may affect functionality controlling the sensor or LED. It is suggested to disable or unplug the webcam before using the program developed with the SDK.
4. Open third-party software to preview Dino-Lite video when using DNX64 SDK.

## Tutorial

The DNX64 provides basic methods and examples to build up app for controlling Dino-Lite on Windows 64bit operating systems.

### Get started

1. Connect Dino-Lite to the computer
2. Open third-party software to preview Dino-Lite video

#### **Visual C++ example:**

Add DNX64.h header file to your project for C++ and load DNX64.dll;

```
int main()

{

    char* strA;

    HINSTANCE handler = LoadLibrary(L"\\DNX64.dll");
    SetVideoDeviceIndex( 0);// set index of video device. Only this can do before the Init.
                                Or another place.

    Init(); //Init the control object. This method must do first, else the all return value will
            be failure or not correctly.

    strA = GetDeviceIDA(0); // Return unique device ID string
    EnableMicroTouch(true); //Enable MicroTouch Event
    SetEventCallback( HandleEvent );
    long nMin,nMax,nStep,nDef;
    // Retrieve value range for video property.
    GetVideoProcAmpValueRange(0, &nMin, &nMax,&nStep,&nDef);
    long nBrightness;
    //Get brightness here
    nBrightness = GetVideoProcAmp(0);
    long nBrightness;
    //Set brightness here
    SetVideoProcAmp(0, 50);
    SetLensInitPos(0); //Set lens to initial position.
```

```

SetFLCLevel(0,6); // Set the LED brightness level (range 1 to 6).

SetFLCSwitch(0,0xf); // Set the LED quadrant switch to partially turn on the LEDs.

//Waiting MicroTouch OnFire Message and get AMR Value
while(GetMessage(&Msg, NULL, 0, 0) > 0)
{
    TranslateMessage(&Msg);
    DispatchMessage(&Msg);

    fMag = GetAMR(0);
    printf("\nMag:%.2f",fMag);
}

```

## Methods

### EnableMicroTouch

Enable/disable MicroTouch button on the camera. When enabled, pressing the button will trigger MicroTouchPressed event.

Function EnableMicroTouch (Enable As Boolean) As Boolean

### FOVx

Return FOV (field of view) of current video frame

Property FoVx As (DeviceIndex As Integer, fMag As Double) As Double

### GetAETarget

Return Auto-Exposure target value of current video frame.

Function GetAETarget (DeviceIndex As Integer) As Long

### GetAMR

Return the magnification reading from models with AMR capability such as AM4515, AF4515, AF4915, AM4517, AM4917, AM7515, AM7915, AF7515, AF7915, AM73515, AM73915, AM8517, and AM8917 series.

Function GetAMR (DeviceIndex As Integer) As Double

### GetAutoExposure

Return AutoExposure of current video frame.

Function GetAutoExposure (DeviceIndex As Integer) As Long

AutoExposure value:

0=AE off

1=AE on

### GetConfig

Retrieves device configure parameters.

Function GetConfig(DeviceIndex As Long) As Long

Bit number	Name	Type	Value
[7]	EDOF	Read	1 = Supported 0 = Not supported
[6]	AMR	Read	1 = Supported 0 = Not supported
[5]	eFLC	Read-	1 = Supported 0 = Not supported
[4]	Aim Point Laseer(APL)	Read	1 = Supported 0 = Not supported
[3:2]	LED	Read	00: Not switchable 01: 2 segments (on & off) 10: 3 segments (LED 1, LED2, off) 11: Reserved
[1]	FLC	Read	1 = Supported 0 = Not supported
[0]	AXI	Read	1 = Supported 0 = Not supported

### GetDeviceId

Return unique device ID string

```
Function GetDeviceId (DeviceIndex As Integer) As String
```

### GetExposureValue

Return the relative Exposure value, which is proportional to the exposure time, of current video frame.

\* This method is available for Premier, Edge, Edge3.0 and EdgePLUS series.

```
Function GetExposureValue (DeviceIndex As Long) As Long
```

### GetLensFinePosLimits

Retrieve value range for lens fine property.

```
Function GetLensFinePosLimits (DeviceIndex As Long, UpperLmt As Long, LowerLmt As Long) As Long
```



This method is available with

1. EdgePLUS AM4917 series
2. Edge AF7915 series
3. EdgePLUS AM8917 series
4. Edge3.0 AM73915(R10, R10A or later) series

### GetLensPosLimits

Retrieve value range for lens property.

```
Function GetLensPosLimits (DeviceIndex As Long, UpperLmt As Long, LowerLmt As Long)
As Long
```

\* This method is available with AM4815, AF4915, AM4917, AM7815, AM7915, AF7915, AM73915 and AM8917 series.

### GetVideoDeviceIndex

Get index of video device to use for capture

```
Property VideoDeviceIndex As Integer
```

### GetVideoProcAmp

Get video properties

```
Property GetVideoProcAmp (ValueIndex As Long) As Long
```

ValueIndex parameter is:

Brightness = 0,  
 Contrast = 1,  
 Hue = 2,  
 Saturation = 3,  
 Sharpness = 4,  
 Gamma = 5,  
 ColorEnable = 6,  
 WhiteBalance = 7,  
 BacklightCompensation = 8,  
 Gain = 9

### GetVideoProcAmpValueRange

Retrieve value range for video property.

```
Function GetVideoProcAmpValueRange (ValueIndex As Long, Min As Long, Max As Long,
SteppingDelta As Long, Default As Long) As Long
```

ValueIndex parameter is:

Brightness = 0,  
 Contrast = 1,  
 Hue = 2,  
 Saturation = 3,  
 Sharpness = 4,  
 Gamma = 5,  
 ColorEnable = 6,  
 WhiteBalance = 7,  
 BacklightCompensation = 8,  
 Gain = 9

Min parameter is:

A variable that receives the minimum value of the property.

Max parameter is:

A variable that receives the maximum value of the property.

SteppingDelta parameter is:

variable that receives the step size for the property. The step size is the smallest increment by which the property can change.

Default

A variable that receives the default value of the property.

CapsFlags

A variable that receives a member of the VideoProcAmpFlags enumeration, indicating whether the property is controlled automatically or manually.

VideoProcAmp\_Flags\_Auto= 0x0001,  
 VideoProcAmp\_Flags\_Manual= 0x0002

|

## GetWiFiImage

Capture picture Wi-Fi streamer mode and save into file.

\* This method is available for WF10, and WF20 series.

```
Function GetWiFiImage(filename As String) As Boolean
```

## SetAETarget

Set Auto Exposure Target value of current video frame.

```
Function SetAETarget (DeviceIndex As Integer, Exposure As Long)
```

The AETarget can only be set when the AutoExposure is set to ON.

The Value can be set from 16 to 220.

### SetAutoExposure

Turn AutoExposure on or off of current video frame.

```
Function SetAutoExposure (DeviceIndex As Integer, AutoExposure As Long)
```

AutoExposure value:

0=AE off

1=AE on

### SetAimpointLevel

Set the aim point laser brightness level (range 0 to 6).

\* This method is only available for models equipped with Aim Point Laser(APL) to control the laser intensity.

```
Function SetAimpointLevel (DeviceIndex As Long, Value As Long)
```

### SetAXIlevel

Set the axial illumination brightness level (range 0 to 6).

\* This method is only available for axial light equipped Dino-Lite, such as AM7515T4A.

```
Function SetAXIlevel (DeviceIndex As Integer, Value As Long)
```

<u>Value</u>	<u>Bright</u>
<u>0</u>	<u>Off</u>
<u>1</u>	<u>3.125%</u>
<u>2</u>	<u>6.25%</u>
<u>3</u>	<u>12.5%</u>
<u>4</u>	<u>25%</u>
<u>5</u>	<u>50%</u>
<u>6</u>	<u>100%</u>

### SetEFLC

Set intensity level on each LED quadrant.

\* This method is available with device listed below.

1. EdgePLUS (R10 or later) series
2. Edge3.0 (R10, R10A, or later) series

```
Function SetEFLC (DeviceIndex As Long, Quadrant as Long, Value As Long)
```

Quadrant: 1-4, 0: all

Value: 1-31, 32: off

### SetExposureValue

Set relative exposure value, which is proportional to the exposure time, of current video frame.

\* This method is only available for Premier, Edge and EdgePLUS series.

```
Function SetExposureValue (DeviceIndex As Long) As Long
```

The ExposureValue can only be set when the AutoExposure is set to OFF.

The range of ExposureValue is varied with different series:

<u>Series</u>	<u>Range of Exposure Value</u>
0.3M Premier	8 to 30612
1.3M Premier	1 to 41771
1.3M Edge	1 to 63076
5M Premier / 5M Edge	1 to 30000
1.3M EdgePLUS	35 to 67498
8M EdgePLUS	10 to 20000

### SetLensFinePos

Set lens fine position.

```
Function SetLensFinePos (DeviceIndex As Integer, Value As Long)
```

\* This method is available with

1. EdgePLUS AM4917 series
2. EdgePLUS AM8917 series
3. Edge AF7915 series
4. Edge3.0 AM73915(R10, R10A or later) series

### SetFLCLevel

Set the LED brightness level

\* This method is available for FLC equipped Dino-Lite, such as AF3113, AF4x15, AM7x15, AF7x15, AM73x15 series (range 1 to 6).

\* This method is available for EFLC equipped Dino-Lite, such as AM4x17 series, AM73x15(R10, R10A or later) series and AM8x17 series (range 1 to 31).

```
Function SetFLCLevel (DeviceIndex As Integer, Value As Long)
```

### SetFLCSwitch

Set the LED quadrant switch to partially turn on the LEDs.

\* This method is available for FLC equipped Dino-Lite, such as AM7x15, AF7x15 and AF4x15 series.

\* This method is available for EFLC equipped Dino-Lite, such as AM4x17, AM73x15 (R10, R10A or later) and AM8x17 series.

```
Function SetFLCSwitch (DeviceIndex As Integer, Value As Long)
```

FLC switch argument values:

<u>Value</u>	<u>Switch-on Quadrant</u>
1	1
2	2
3	1, 2
4	3
5	1, 3
6	2, 3
7	1, 2, 3
8	4
9	1, 4
10	2, 4
11	1, 2, 4
12	3, 4
13	1, 3, 4
14	2, 3, 4
15	1, 2, 3, 4
16	All LEDs turn off

### SetLEDState

Switch the camera LED.

\* The LEDState will be controllable only when the camera preview is established.

\* This method may not be applicable to AM211, AM2011, AM2111, and Dino-Eye series.

```
Function SetLEDState (DeviceIndex As Integer, LEDState As Long) As Long
```

LEDState Values:

0 = LED off

1 = LED1 on

2 = LED2 on. The LED2 only exists on models with 2 switchable LEDs.

### SetLensInitPos

Set lens to initial position.

```
Function SetLensInitPos (DeviceIndex As Integer)
```

\* This method is available with AM4815, AF4915, AM4917, AM7815, AM7915, AF7915, AM73915 and AM8917 series.

### SetLensPos

Set lens position.

```
Function SetLensPos (DeviceIndex As Integer, Value As Long)
```

\* This method is available with AM4815, AF4915, AM4917, AM7815, AM7915, AF7915, AM73915 and AM8917 series.

### SetVideoDeviceIndex

Set index of video device to use for capture

```
Property SetVideoDeviceIndex As Integer
```

\*When VideoDeviceIndex = 90 for using WF10/WF20 control.

### SetVideoProcAmp

Set video properties

```
Property SetVideoProcAmp (ValueIndex As Long) As Long
```

ValueIndex parameter is:

Brightness = 0,  
 Contrast = 1,  
 Hue = 2,  
 Saturation = 3,  
 Sharpness = 4,  
 Gamma = 5,  
 ColorEnable = 6,  
 WhiteBalance = 7,  
 BacklightCompensation = 8,  
 Gain = 9

### SetWiFiVideoRes

Set video image dimensions

```
Function SetWiFiVideoRes (width As Long, height As Long) As Boolean
```

Return TRUE if it is successful or FALSE otherwise.

Because video formats are device-specific, applications should check the return value from this function to determine if the format is accepted by the driver.

\* This method is only available for WF10/WF20.

### **MicroTouchPressed**

This event is fired when MicroTouch button on the camera is pressed.

```
void SetEventCallback(EventCallback func);
```