

VC-C50i SDK

Active X Controller

Function Manual

Ver1.1

OCX file name	Vcc5AXCtrlE.OCX
Component name	Vcc5AXCtrlE
Control Object name	Vcc5Ctrl

History of revisions

Version	Issue date	Description of revision	
Ver1.0	Dec. 25, 2003	First version issued.	
Ver1.1	Mar. 04, 2005	1. Exposure mode parameter added "02". 2. Dome mode setting and verification parameter changed to "00, 01" only.	Applicable to Ver.6-08

Contents

	<Communication Control>	5
CommOpen	Communication Port Open Function.....	5
CommClose	Communication Port Close	5
GetPicData	Received Data Information Acquisition.....	6
GetLastCmdStr	Transmitted Data Size Acquisition	6
GetPicLastCmd	Transmitted Command Data Acquisition	6
GetLastError	Extended Error Information Acquisition.....	7
GetCommModemStatus	Communication Terminal Information Acquisition.....	7
Terminate	Release Message Hook.....	7
	<Generated Event>	8
AckRxEvent	Received Event of ACK/NACK Command.....	8
NotifyRxEvent	Received Event of Notification Command.....	8
	<Camera Command>	9
SetCameraPower	Power Source Setting.....	9
SetFocusMode	Focus Mode Assignment.....	9
SetFocusPosition	Focus Position Assignment	10
GetFocusPosition	Focus Position Request.....	10
SetOnePushAF	One Push AF Assignment.....	11
GetFocusRange	Focus Range Request.....	11
SetZooming	Zoom Assignment	12
SetZoomPosition	Zoom Position Assignment 1	12
GetZoomPosition	Zoom Position Request 1	13
SetZoomPosition4	Zoom Position Assignment 2	13
GetZoomPosition4	Zoom Position Request 2.....	14
SetZoomSpeed	Zoom Speed Assignment	14
GetZoomSpeed	Zoom Speed Request.....	15
GetZoomMax	Zoom Position Movable Maximum Value Request.....	15
SetElectronicZoom	Electronic Zoom Setting.....	16
GetElectronicZoom	Electronic Zoom Setting Status Request	16
SetBackLight	Backlight Compensation Assignment	17
SetExposed	Exposure Mode Assignment	17
SetAELock	AE Lock Assignment	18
SetShutterSpeedx	Shutter Speed Manual Assignment.....	18
GetShutterSpeed	Shutter Speed Request.....	19
SetShutterSpeed	Shutter Speed Assignment	19
SetAGCGain	AGC Gain Assignment.....	20
GetAGCGain	AGC Gain Request.....	20
SetIris	Iris Assignment.....	21
GetIris	Iris Request.....	21
SetAEVal	AE Target Value Assignment.....	22
GetAEVal	AE Target Value Request.....	22
SetWhiteBalance	White Balance Mode Assignment	23
SetWhiteBalanceMx	White Balance Manual Assignment.....	23
GetWhiteBalance	White Balance Setting Value Request	24
SetFading	Fade Mode Assignment.....	24
CameraReset	Camera Reset.....	25
GetZoomRatio	Zoom Ratio Request.....	25
GetCCDSIZE	CCD Pixel Size Request.....	26

GetCameraVersion	Product Version Request	26
SetDomeData	Dome Data Writing	27
GetDomeData	Dome Data Request.....	27
<Pedestal Control Command> 28		
SetMotorSpeed	Motor Speed Assignment	28
GetMotorSpeed	Motor Speed Request.....	28
SetPanTilt1	Pan/Tilt Operation 1	29
GotoHomePosition	Home Position Movement.....	29
PedestalInitialize	Pedestal Initialization.....	30
GetLimitSpeed	Motor Maximum Minimum Speed Request	30
GetGearRatio	Gear Ratio Request.....	31
GetLimitAngle	Motor Maximum Minimum Angle Request	31
SetPanTilt2	Pan/Tilt Operation 2	32
SetPTAngle2	Pan/Tilt Angle Assignment	33
GetPTAngle2	Pan/Tilt Angle Request	33
SetPTAngle1	Pan/Tilt Movable Range Assignment.....	34
GetPTAngle1	Pan/Tilt Movable Range Request.....	35
<Common Control Command> 36		
SetIRRemote	Remote Control Assignment	36
GetOperateStatus	Operation Status Request	36
GetExtendedOperateStatus	Extended Operation Status Request.....	37
GetOperateStatus3	Operation Status3 Request	37
GetOperateStatus4	Operation Status4 Request	38
GetUnitName	Model Name Request	38
GetRomVersion	ROM Version Request	39
SetPresetMemory	Pre-Set Memory Assignment.....	39
GoToPreset	Pre-Set Movement.....	40
GetPresetStatus	Preset Status Request	40
GetExtendPreset	Extended Preset Status Request.....	41
RemoteCommandPass	Remote Control Through Assignment	41
SetLed	LED Indication Assignment.....	42
SetCascade	Cascade Mode Assignment	42
SetCmdMode	Command Mode Assignment.....	43
SetOnScreen	On Screen Assignment	44
SetTitleString	On Screen Indicated Character Assignment.....	45
GetTitleString	On Screen Indicated Character Read Out.....	45
SetDate	Date Assignment	46
GetDate	Assigned Date Read Out	46
SetTime	Time Assignment	47
GetTime	Assigned Time Read Out	47
GetCameraTime	Accumulated Power ON Time of Camera Request.....	48
GetPTTime	Accumulated Power ON Time of Pedestal Request.....	48
SetDefault	Default Assignment.....	49
SetNotify	Command Termination Notification Existence.....	49
SetGlobalNotify	Cascade Global Notification Existence	50
GetPedestalModel	Pedestal Model Request.....	51
GetCameraModel	Camera Model Request.....	51
SetAlarmOutput	Alarm Output Setting.....	52
GetAlarmOutput	Alarm Output Setting Status Request.....	52
SetExtSensorDetect	External Sensor Input Detection Setting	53

GetExtSensorDetect	External Sensor Input Detection Setting Information Request.....	53
GetExtSensorInput	External Sensor Input Status Request.....	54
SetIRLight	Internal InfraRed Light Setting	55
GetIRLight	Internal InfraRed Light Status Request.....	55
SetExtLightMode	External Light Output Setting.....	56
GetExtLightMode	External Light Output Status Request.....	56
SetIRCutFilter	InfraRed Cut Filter Insertion/Removal Setting	57
GetIRCutFilter	InfraRed Cut Filter Setting Status Request	57
SetNoiseReduction	Noise Reduction Setting	58
GetNoiseReduction	Noise Reduction Setting Status Request.....	58

<Communication Control>

CommOpen

Communication Port Open Function

This function is to open the communication port

```
CommOpen(  
    ByVal IPort As Long, _           ' Communication Port Number  
    ByVal ISpeed As Long, _         ' Communication Speed  
    ByVal IBytesize As Long, _      ' Communication Data Length  
    ByVal IParity As Long, _        ' Communication Parity  
    ByVal IStopbit As Long _        ' Communication Stop Bit length  
) As Long
```

Parameter

IPort

To assign Communication Port Number. In case of COM1, to assign 1

ISpeed

To assign Communication Speed

Assignable Values for VC-C5 are 9600bps, 14400bps and 19200bps

To assign directly the number of Communication Speed

IBytesize

To assign Communication Data Byte length between 4~8

Assignable Value for VC-C5 is only 8 (8bit)

IParity

To assign Communication Parity between 0~4

0: No parity, 1: Odd parity, 2: Even parity, 3: Mark parity, 4: Space parity

Assignable value for VC-C5 is only 0 (No parity)

IStopbit

To assign Communication Stop Bit length. Assignable value are 0: 1 bit, 1: 1.5bit, 2: 2bit.

For VC-C5, possible to assign 1 bit and 2 bit

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

CommClose

Communication Port Close

To close Communication Port

```
CommClose( ) As Long
```

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

GetPicData

Received Data Information Acquisition

To read out the received command

The guarantee term of received data is until the time of the event generated

```
GetPicData(  
    ByVal index As Long _           ' Assignment of command acquired position  
    ) As Integer
```

Parameter

index

This value is the buffer position data in order to acquire the command data

Possible to assign from 0 to the received data byte number -1

The received data byte number refers to the parameter of event function

Return Value

To returned the data of assigned buffer position

In case of the effective outrange assigned, returns 0

GetLastCmdStr

Transmitted Data Size Acquisition

To read out the transmitted data size of command transmitted lastly

```
GetLastCmdStr( ) As Long
```

Return Value

To return the transmitted command data size

GetPicLastCmd

Transmitted Command Data Acquisition

To read out the transmitted command data sent lastly

```
GetPicLastCmd(  
    ByVal index As Long _           ' Assignment of command buffer position  
    ) As Integer
```

Parameter

index

This value is the buffer position data in order to acquire the command data

Possible to assign from 0 to the transmitted data byte number -1

The transmitted data byte size is acquired by **GetLastCmdStr** function

Return Value

To return the data of the assigned buffer position

In case of the effective outrange assigned, return 0

GetLastError Extended Error Information Acquisition

To acquire the extended error information

```
GetLastError(  
    ByRef lpError As Long _          ' System error information  
    ) As Long
```

Parameter

lpError

To assign buffer referring to the buffer in order to acquire the system error information

In case that the returned error is "system error", set its error information

The error contents is its content acquired by **GetLastError** of Win32Api

Return Value

In case of function successful, return 0 and failed, return the value besides 0

Error Code

-1: System error

-2: Communication Port Open error

GetCommModemStatus Communication Terminal Information Acquisition

To acquire the status of DSR, CTS and other terminals

```
GetCommModemStatus( ) As Long
```

Return Value

To return the communication terminal information with bit assigned

Bit4: CTS terminal status (0 = OFF, 1 = ON)

Bit5: DSR terminal status (0 = OFF, 1 = ON)

※ Return Value is the same as **GetCommModemStatus** function of VC++

Terminate Release Message Hook

To release Hook Assignment of Windows Message initialized by ActiveX Control

By the environment of machines, it could present to un-release automatically at the moment of application terminated

In this case, release Message Hook by using this function

```
Terminate( )
```


<Generated Event>

AckRxEvent Received Event of ACK/NACK Command

This event is generated at the moment of receipt of ACK/NACK command

In order to acquire the received data, necessary to pick up at the moment of generating event by **GetPicData** function

```
Event AckRxEvent(  
    ByVal wParam As Long, _      ' Received Command Class  
    ByVal lBytes As Long _      ' Command data size  
)
```

Parameter

wParam

Classification of Command received is to be set. The value is 1 (fixed)

lBytes

To set data size of the received command

NotifyRxEvent Received Event of Notification Command

This event is generated at the moment of receipt of Notification Command

In order to acquire the received data, necessary to pick up at the moment of generating event by **GetPicData** function

```
Event NotifyRxEvent(  
    ByVal wParam As Long, _      ' Received Command Class  
    ByVal lBytes As Long _      ' Command data size  
)
```

Parameter

wParam

To set classification of Command received

1: [Header FAh] At the moment of receipt of Notification Command

2: [Header FDh] At the moment of receipt of Remote Control Through Command
(Remote ON)

3: [Header FCh] At the moment of receipt of Remote Control Through Command
(Remote OFF)

4: [Header F8h] At the moment of receipt of Result Notification for Global Command

5: [Header FBh] At the moment of receipt of Event Notification Command

lBytes

To set data size of received command

<Camera Command>

SetCameraPower Power Source Setting

To set power source of camera section

```
SetCameraPower(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IPower As Long _         ' Power source control  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IPower

To control power source Setting 0 is OFF and setting 1 is ON

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

SetFocusMode Focus Mode Assignment

To assign focus mode

```
SetFocusMode(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMode As Long _         ' Mode selection  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IMode

To assign focus mode as followings

0: Auto focus mode

1: Manual mode (to stop focus operation)

2: Focus Near mode (to move into near focus)

3: Focus Far mode (to move into far focus)

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

SetFocusPosition Focus Position Assignment

To move focus to the assigned position

```
SetFocusPosition(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IFocusPos As Long _       ' Focus position  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IFocusPos

To assign focus position

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

GetFocusPosition Focus Position Request

To request a focus position

```
GetFoucsPosition(  
    ByVal IUnit As Long _           ' Device number  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

SetOnePushAF**One Push AF Assignment**

To assign manual mode after assignment of focus by auto focus change

```
SetOnePushAF(  
    ByVal IUnit As Long _           ' Device number  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

GetFocusRange**Focus Range Request**

To request focus movable range

```
GetFocusRange(  
    ByVal IUnit As Long _           ' Device number  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

SetZooming

Zoom Assignment

To assign zoom operation

```
SetZooming(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMode As Long _           ' Mode selection  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IMode

To assign zoom mode as followings,

- 0: To stop zoom operation
- 1: To move zoom into Wide
- 2: To move zoom into Tele
- 3: To move zoom into Wide with Hi-speed
- 4: To move zoom into Tele with Hi-speed

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

SetZoomPosition

Zoom Position Assignment 1

To move zoom to the position assigned

```
SetZoomPosition(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IZoomPos As Long _         ' Zoom position  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IZoomPos

To assign zoom position

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

GetZoomPosition

Zoom Position Request 1

To request zoom position

```
GetZoomPosition(  
    ByVal IUnit As Long _           ' Device number  
    ) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status
In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0
In order to get extended error information, uses **GetLastError**

SetZoomPosition4

Zoom Position Assignment 2

To move into zoom position assigned

```
SetZoomPosition4(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IZoomPos As Long _        ' Zoom position  
    ) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status
In case of not Cascade connection, to set 0

IZoomPos

To assign zoom position

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0
In order to get extended error information, uses **GetLastError**

GetZoomPosition4

Zoom Position Request 2

To request zoom position

```
GetZoomPosition4(  
    ByVal IUnit As Long, _           ' Device number  
    ) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status
In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0
In order to get extended error information, uses **GetLastError**

SetZoomSpeed

Zoom Speed Assignment

To assign zoom running speed

```
SetZoomSpeed(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal ISpeed As Long, _         ' Zoom speed  
    ) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status
In case of not Cascade connection, to set 0

ISpeed

To assign zoom speed

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0
In order to get extended error information, uses **GetLastError**

Zoom Speed Request

To request zoom running speed

) As Long

Parameter

1Unit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

Zoom Position Movable Maximum Value Request

To request movable maximum value of zoom position

) As Long

Parameter

1Unit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

SetElectronicZoom Electronic Zoom Setting
Sets the zoom to the designated magnification.

```
SetElectronicZoom(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMode As Long _           ' Mode setting  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status
In case of not Cascade connection, to set 0

IMode

The magnification zoom is designated by the following numbers.

0: 1x (no electronic zoom)
1: 1x (no electronic zoom)
2: 2x
4: 4x
8: 8x
12: 12x

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0
In order to get extended error information, uses **GetLastError**

GetElectronicZoom Electronic Zoom Setting Status Request
Requests the setting status of the electronic zoom.

```
GetElectronicZoom(  
    ByVal IUnit As Long _           ' Device number  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status
In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0
In order to get extended error information, uses **GetLastError**

SetBackLight

Backlight Compensation Assignment

To assign backlight compensation mode

```
SetBackLight(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMode As Long _           ' Mode setting  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IMode

To assign backlight compensation mode as followings

0: Not to perform backlight compensation

1: To set backlight compensation mode

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

SetExposed

Exposure Mode Assignment

To assign exposure mode

```
SetExposed(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMode As Long _           ' Mode setting  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IMode

To assign exposure mode as followings,

0: To set exposure mode to auto mode

1: To set exposure mode to manual mode

2: To set exposure mode to the shutter speed is reduced to low with automatic exposure
(Ver.6-08 and up.)

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

SetAELock AE Lock Assignment

To assign AE Lock mode

```
SetAELock(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMode As Long _           ' Mode setting  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IMode

To assign AE mode as followings

0: To release AE lock ON

1: To lock the exposure in status of Auto-Exposure mode in the status

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

SetShutterSpeedx Shutter Speed Manual Assignment

To assign a shutter speed

```
SetShutterSpeedx(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal ISpeed As Long _           ' Shutter Speed  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

ISpeed

To assign shutter speed between 1 and 1/80000(sec).

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

GetShutterSpeed

Shutter Speed Request

To request shutter speed

```
GetShutterSpeed(  
    ByVal IUnit As Long _           ' Device number  
    ) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status
In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0
In order to get extended error information, uses **GetLastError**

SetShutterSpeed

Shutter Speed Assignment

To assign shutter speed mode

```
SetShutterSpeed(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMode As Long _           ' Mode setting  
    ) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status
In case of not Cascade connection, to set 0

IMode

To assign Shutter speed mode as followings,
0: To set program mode
1: To set 1/60 sec (PAL 1/50) mode
2: To set 1/100 sec (PAL 1/120) mode

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0
In order to get extended error information, uses **GetLastError**

SetAGCGain

AGC Gain Assignment

To assign AGC gain

```
SetAGCGain(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IGain As Long _          ' AGC gain  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IGain

To assign AGC magnification

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

GetAGCGain

AGC Gain Request

To request AGC gain

```
GetAGCGain(  
    ByVal IUnit As Long _           ' Device number  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

SetIris

Iris Assignment

To assign Iris

```
SetIris(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IIris As Long _           ' Iris  
    ) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IIris

To assign Iris value

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

GetIris

Iris Request

To request Iris value

```
GetIris(  
    ByVal IUnit As Long _           ' Device number  
    ) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

SetAEVal AE Target Value Assignment

To assign a target value of AE brightness

```
SetAEVal(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal lAEval As Long _         ' AE value  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

lAEval

To assign AE target value

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

GetAEVal AE Target Value Request

To request AE target value

```
GetAEVal(  
    ByVal IUnit As Long _           ' Device number  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

SetWhiteBalance White Balance Mode Assignment

To assign white balance mode

```
SetWhiteBalance(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMode As Long _           ' Mode setting  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IMode

To assign white balance mode as followings,

0: To set Auto mode

1: To stop white balance mode and set Lock mode

2: To set manual mode

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

SetWhiteBalanceMx White Balance Manual Assignment

To assign manual value of white balance

```
SetWhiteBalanceMx(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IGain As Long _           ' Setting value  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IGain

To assign white balance setting value

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

GetWhiteBalance White Balance Setting Value Request

To request setting value of white balance

```
GetWhiteBalance(  
    ByVal IUnit As Long _           ' Device number  
    ) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

SetFading Fade Mode Assignment

To assign fade mode

```
SetFading(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMode As Long _           ' Mode setting  
    ) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IMode

To assign fade mode as followings,

0: To set normal picture out status (To release fade status slowly)

1: To come out white picture with slow speed

2: To come out white picture with high speed

3: To come out black picture with high speed

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

CameraReset

Camera Reset

To Reset Camera section

```
CameraReset(  
    ByVal IUnit As Long _           ' Device number  
    ) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status
In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0
In order to get extended error information, uses **GetLastError**

GetZoomRatio

Zoom Ratio Request

To request zoom ratio

```
GetZoomRatio(  
    ByVal IUnit As Long _           ' Device number  
    ) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status
In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0
In order to get extended error information, uses **GetLastError**

GetCCDSize

CCD Pixel Size Request

To request CCD pixel size

```
GetCCDSize(  
    ByVal IUnit As Long, _           ' Device number  
    ) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status
In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0
In order to get extended error information, uses **GetLastError**

GetCameraVersion

Product Version Request

To request Product Version and ROM Version of Camera

```
GetCameraVersion(  
    ByVal IUnit As Long, _           ' Device Number  
    ByVal IMode As Long, _          ' Mode Setting  
    ) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status
In case of not Cascade connection, to set 0

IMode

To assign version type as followings,
0: To request Product version
1: To request EEPROM version

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0
In order to get extended error information, uses **GetLastError**

SetDomeData

Dome Data Writing

This is the control mode selector command based on dome data writing.

```
SetDomeData(  
    ByVal IUnit As Long _           ' Device number  
    ByVal IMode As Long _          ' Mode setting  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IMode

Dome data setting value

0: Select cam path 0 to 9

1: Select cam path 0 to 8

2: Select cam path 0 to 7

3: Select cam path 0 to 6

The following are cam paths 0 to 9.

0: INF

1: 830 cm

2: 380 cm

3: 240 cm

4: 230cm

5: 155 cm

6: 86 cm

7: 27.2 cm

8: 7.3 cm

9: 0.7 cm

[Ver.6-08 and up]

0: Dome mode OFF

1: Dome mode ON

•The Dome mode setting is retained even after the power is turned off

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

GetDomeData

Dome Data Request

Requests the dome data setting value.

```
GetDomeData(  
    ByVal IUnit As Long _           ' Device number  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

<Pedestal Control Command>

SetMotorSpeed Motor Speed Assignment

To assign motor speed of Pan/Tilt

```
SetMotorSpeed(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMode As Long, _          ' Mode selection  
    ByVal ISpeed As Long _          ' Speed assignment  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IMode

To assign Pan/Tilt motor

0: Pan motor assignment

1: Tilt motor assignment

ISpeed

To assign motor speed between 8h and 320h

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

GetMotorSpeed Motor Speed Request

To request the running speed of Pan/Tilt motor

```
GetMotorSpeed(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMode As Long _          ' Mode selection  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IMode

To assign Pan/Tilt motor

0: Pan motor assignment

1: Tilt motor assignment

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

SetPanTilt1

Pan/Tilt Operation 1

To assign Pan/Tilt operation

```
SetPanTilt1(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMode As Long _           ' Mode selection  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IMode

To assign Pan/Tilt operation as followings,

0: To stop operation

1: To start Pan right operation

2: To start Pan left operation

3: To start Tilt upper operation

4: To start Tilt lower operation

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

GotoHomePosition

Home Position Movement

To move Pan/Tilt positions to Home position

```
GotoHomePosition(  
    ByVal IUnit As Long _           ' Device number  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

PedestalInitialize

Pedestal Initialization

To initialize pedestal section

```
PedestalInitialize(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMode As Long _           ' Mode selection  
    ) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IMode

To initialize as followings,

0: Initialize operation 1 (After initialization, move to Home position)

1: Initialize operation 2 (After initialization, move to Prior position)

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

GetLimitSpeed

Motor Maximum Minimum Speed Request

To request maximum & minimum running speed of motor

```
GetLimitSpeed(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMode As Long _           ' Mode selection  
    ) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IMode

To assign the requested parameter as followings

0: Request Pan minimum speed

1: Request Pan maximum speed

2: Request Tilt minimum speed

3: Request Tilt maximum speed

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

GetGearRatio

Gear Ratio Request

To request a gear ratio of motor

```
GetGearRatio(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMode As Long _           ' Mode selection  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IMode

To assign the requested parameter as followings

0: Request Pan gear ratio

1: Request Tilt gear ratio

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

GetLimitAngle

Motor Maximum Minimum Angle Request

To request maximum & minimum operation angle of motor

```
GetlimitAngle(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMode As Long _           ' Mode selection  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IMode

To assign the requested parameter as followings,

0: Request minimum Pan angle

1: Request maximum Pan angle

2: Request minimum Tilt angle

3: Request maximum Tilt angle

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

SetPanTilt2

Pan/Tilt Operation 2

To assign Pan/Tilt operation

```
SetPanTilt2(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMode As Long _           ' Mode selection  
    ) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IMode

To assign Pan/Tilt operation as followings,

- 0: Stop operation
- 1: Stop Pan, start Tilt upper operation
- 2: Stop Pan, start Tilt lower operation
- 3: Start Pan right, stop Tilt operation
- 4: Start Pan left, stop Tilt operation
- 5: Start Pan right, Tilt upper operation
- 6: Start Pan right, Tilt lower operation
- 7: Start Pan left, Tilt upper operation
- 8: Start Pan left, Tilt lower operation

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

SetPTAngle2 Pan/Tilt Angle Assignment

To assign Pan/Tilt movement position by angle

```
SetPTAngle2(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal Angle1 As Long, _         ' Pan position angle  
    ByVal Angle2 As Long _         ' Tilt position angle  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

Angle1

To assign angle of Pan movement position

Angle2

To assign angle of Tilt movement position

To assign angle by \pm movement value as 8000h at the center

- value: movement for Pan left, Tilt lower

+ value: movement for Pan right, Tilt upper

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

GetPTAngle2 Pan/Tilt Angle Request

To request Angle of Pan/Tilt position

```
GetPTAngle2(  
    ByVal IUnit As Long _           ' Device number  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

SetPTAngle1

Pan/Tilt Movable Range Assignment

To assign angle of movable range for Pan/Tilt operation

```
SetPTAngle1(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMode As Long, _          ' Mode selection  
    ByVal Angle1 As Long, _         ' Movable operation angle range 1  
    ByVal Angle2 AS Long _         ' Movable operation angle range 2  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IMode

To assign Pan/Tilt

0: Assignment of Pan movable angle range

1: Assignment of Tilt movable angle range

Angle1

When IMode is Pan, assigns movable angle range of Pan left position

When IMode is Tilt, assigns movable angle range of Tilt lower position

Angle2

When IMode is Pan, assigns movable angle range of Pan right position

When IMode is Tilt, assigns movable angle range of Tilt upper position

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

GetPTAngle1

Pan/Tilt Movable Range Request

To request angle of movable range for Pan/Tilt operation

```
GetPTAngle1(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMode As Long _           ' Mode selection  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IMode

To assign the requested parameter as followings,

0: To request Pan angle of movable range

1: To request Tilt angle of movable range

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

<Common Control Command>

SetIRRemote Remote Control Assignment

To assign operation of remote control

```
SetIRRemote(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMode As Long _           ' Mode selection  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IMode

To assign the requested parameter as followings,

0: To assign remote control permitted

1: To assign remote control forbidden

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

GetOperateStatus Operation Status Request

To request information indicated inside status

```
GetOperateStatus(  
    ByVal IUnit As Long _           ' Device number  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

GetExtendedOperateStatus Extended Operation Status Request

To request information indicated inside status of pedestal

```
GetExtendedOperateStatus(  
    ByVal IUnit As Long _           ' Device number  
    ) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

GetOperateStatus3 Operation Status3 Request

To request information (Operation Status3) indicated inside status

```
GetOperateStatus3(  
    ByVal IUnit As Long _           ' Device number  
    ) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

GetOperateStatus4

Operation Status4 Request

To request information (Operation Status4) indicated inside status

```
GetOperateStatus4(  
    ByVal IUnit As Long _           ' Device number  
    ) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

GetUnitName

Model Name Request

To request model name

```
GetUnitName(  
    ByVal IUnit As Long _           ' Device Number  
    ) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

GetRomVersion ROM Version Request

To request ROM version

```
GetRomVersion(  
    ByVal IUnit As Long _           ' Device number  
    ) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

SetPresetMemory Pre-Set Memory Assignment

To assign motor and zoom position at receipt of command into memory number assigned

```
SetPresetMemory(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMemory As Long _         ' Memory number  
    ) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IMemory

To assign number of memory for save between 1 and 9

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

GoToPreset

Pre-Set Movement

To notify movement to preset position of memory number assigned

```
GoToPreset(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMemory As Long _         ' Memory number  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IMemory

To assign number of memory for save between 1 and 9

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

GetPresetStatus

Preset Status Request

To request status in use for preset position memory

```
GetPresetStatus(  
    ByVal IUnit As Long _           ' Device number  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

GetExtendPreset Extended Preset Status Request

To request the used status of pre-set position memory by word

```
GetExtendPreset(  
    ByVal IUnit As Long _           ' Device number  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

RemoteCommandPass Remote Control Through Assignment

To assign notification of remote control data

```
RemoteCommandPass(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMode As Long _           ' Mode selection  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IMode

To assign the requested parameter as followings

0: To assign notification forbidden of remote control data

1: To assign notification permitted of remote control data

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

SetLed LED Indication Assignment

To assign LED indication

```
SetLed(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMode As Long _           ' Mode selection  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IMode

To assign the requested parameter as followings,

0: Normal indication

1: Forced lighting ON of Green LED

2: Forced lighting OFF of All LED

3: Forced lighting ON of Red LED

4: Forced lighting ON of Orange LED

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

SetCascade Cascade Mode Assignment

To assign cascade connection

```
SetCascade(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMode As Long _           ' Mode selection  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IMode

To assign the requested parameter as followings

0: To assign termination of cascade connection

1: To assign start of cascade connection

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

SetCmdMode Command Mode Assignment

To assign a command control mode

```
SetCmdMode(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMode As Long _          ' Mode selection  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IMode

To assign the requested parameter as followings,

0: To assign Host control mode

1: To assign Local control mode

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

SetOnScreen

On Screen Assignment

To assign On screen indication of date and time

```
SetOnScreen(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMode As Long _           ' Mode selection  
    ) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IMode

To assign the requested parameter as followings,

- 0: To assign On Screen Indication OFF
- 1: To assign On Screen Indication ON
- 2: To assign Character indication OFF
- 3: To assign Character indication ON
- 4: To assign Time indication OFF
- 5: To assign Time indication ON (Indication Form 1)
- 6: To assign Time indication ON (Indication Form 2)
- 7: To assign Date indication OFF
- 8: To assign Date indication ON (Indication Form 1)
- 9: To assign Date indication ON (Indication Form 2)

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

SetTitleString On Screen Indicated Character Assignment

To assign an indication data of on screen character

```
SetTitleString(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IX As Long, _             ' Indication position (Horizontal)  
    ByVal IY As Long, _             ' Indication position (Vertical)  
    ByVal ICharacter _              ' Indication Character  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IX

To assign indicated horizontal position between 0 and 23 value

IY

To assign indicated vertical position between 0 and 10 value

ICharacter

To assign character code of indicated character

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

GetTitleString On Screen Indicated Character Read Out

To request read out of on screen indicated characters

```
GetTitleString(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IX As Long, _             ' Indicated position (Horizontal)  
    ByVal IY As Long _              ' Indicated position (Vertical)  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IX

To assign indicated horizontal position between 0 and 23 value

IY

To assign indicated vertical position between 0 and 10 value

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

SetDate Date Assignment

To assign a data of date indicated on screen

```
SetDate(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IYear As Long, _          ' Year  
    ByVal IMonth As Long, _         ' Month  
    ByVal IDay As Long _            ' Day  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IYear

To assign data of indicated year between 0 and 99 value

IMonth

To assign data of indicated month between 1 and 12 value

IDay

To assign data of indicated day between 1 and 31 value

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

GetDate Assigned Date Read Out

To request read out of on screen indicated date

```
GetDate(  
    ByVal IUnit As Long _           ' Device number  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

SetTime Time Assignment

To assign an indicated data of time on screen

```
SetTime(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IHour As Long, _          ' Hour  
    ByVal IMinutes As Long, _       ' Minute  
    ByVal ISecond As Long _        ' Second  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IHour

To assign indicated hour data between 0 and 23 value

IMinutes

To assign indicated minute data between 0 and 59

ISecond

To assign indicated second data between 0 and 59

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

GetTime Assigned Time Read Out

To request read out of on screen indicated time

```
GetTime(  
    ByVal IUnit As Long _           ' Device number  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

GetCameraTime Accumulated Power ON Time of Camera Request
To request total accumulated power ON time of Camera section

```
GetCameraTime(  
    ByVal IUnit As Long _                      ' Device number  
    ) As Long
```

Parameter

IUnit
To assign device number to 1~9 in case of Cascade status
In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0
In order to get extended error information, uses **GetLastError**

GetPTTime Accumulated Power ON Time of Pedestal Request
To request total accumulated power ON time of Pedestal section

```
BOOL GetPTTime(  
    ByVal IUnit As Long _                      ' Device number  
    ) As Long
```

Parameter

IUnit
To assign device number to 1~9 in case of Cascade status
In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0
In order to get extended error information, uses **GetLastError**

SetDefault Default Assignment

To return camera status into factory setting status

```
SetDefault(  
    ByVal IUnit As Long _           ' Device number  
    ) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

SetNotify Command Termination Notification Existence

To assign an answer to Notification of Termination after command executed

```
SetNotify(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMode As Long _           ' Mode selection  
    ) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IMode

To assign the requested parameter as followings

0: To assign notification not existed

1: To assign notification existed

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

SetGlobalNotify

Cascade Global Notification Existence

To assign an answer of Termination Notification for global command in cascade connected

```
SetGlobalNotify(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMode As Long _           ' Mode selection  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IMode

To assign the requested parameter as followings

0: To assign notification not existed

1: To assign notification existed

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

GetPedestalModel**Pedestal Model Request**

To request Pedestal Model mounted (Normal & Inverse position)

```
GetPedestalModel(  
    ByVal IUnit As Long _           ' Device Number  
    ) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

GetCameraModel**Camera Model Request**

To request the model of camera (NTSC, PAL)

```
GetCameraModel(  
    ByVal IUnit As Long _           ' Device Number  
    ) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

SetAlarmOutput Alarm Output Setting
Enables ON/OFF setting of the alarm output terminal.

```
SetAlarmOutput(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMode As Long _           ' Mode selection  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status
In case of not Cascade connection, to set 0

IMode

This designates the alarm output terminal status.
The 0 setting is alarm output OFF (\pm terminal Open) and 1 setting is alarm output ON (\pm terminal Closed).

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0
In order to get extended error information, uses **GetLastError**

GetAlarmOutput Alarm Output Setting Status Request
Requests the setting status of the alarm output terminal.

```
GetAlarmOutput(  
    ByVal IUnit As Long _           ' Device number  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status
In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0
In order to get extended error information, uses **GetLastError**

SetExtSensorDetect External Sensor Input Detection Setting
Enables setting of the external sensor input detection function.

```
SetExtSensorDetect(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMode As Long _           ' Mode selection  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status
In case of not Cascade connection, to set 0

IMode

00h: Notification OFF
01h: Notification ON when sensor terminal changes from Closed to Open
10h: Notification ON when sensor terminal changes from Open to Closed
11h: Notification ON when sensor terminal changes from both Closed to Open
and Open to Closed.

*When the VC-C50i is set to Notification ON and the sensor terminal has been set,
the occurrence of changes is notified by an Event Notification Command.

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0
In order to get extended error information, uses **GetLastError**

GetExtSensorDetect External Sensor Input Detection Setting Information Request
Requests the setting information of the external sensor input detection.

```
GetExtSensorDetect(  
    ByVal IUnit As Long _           ' Device number  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status
In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0
In order to get extended error information, uses **GetLastError**

GetExtSensorInput

External Sensor Input Status Request

Requests the setting information of the external sensor input detection function.

```
GetExtSensorInput(  
    ByVal IUnit As Long _           ' Device number  
    ) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

SetIRLight Internal InfraRed Light Setting

Enables ON/OFF setting of internal infrared light.

```
SetIRLight(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMode As Long _           ' Mode selection  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

IMode

Light control is set by the following numbers.

0: LED OFF

1: LED ON (Automatic OFF after 30 minutes)

2: LED ON (Automatic OFF after 1 hour)

3: LED ON (Automatic OFF after 2 hours)

4: LED ON (Automatic OFF after 3 hours)

5: LED ON (Automatic OFF after 4 hours)

6: LED ON (Automatic OFF after 8 hours)

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

GetIRLight Internal InfraRed Light Status Request

Requests the current ON/OFF status of the internal infrared light.

```
GetIRLight(  
    ByVal IUnit As Long _           ' Device number  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status

In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0

In order to get extended error information, uses **GetLastError**

SetExtLightMode External Light Output Setting
Enables setting of the external light output (ON/OFF).

```
SetExtLightMode(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMode As Long _           ' Mode selection  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status
In case of not Cascade connection, to set 0

IMode

This designates the external light output status. The 0 setting is external light output OFF,
and the 1 setting is external light output ON.

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0
In order to get extended error information, uses **GetLastError**

GetExtLightMode External Light Output Status Request
Requests the output setting status of external light.

```
GetExtLightMode(  
    ByVal IUnit As Long _           ' Device number  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status
In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0
In order to get extended error information, uses **GetLastError**

SetIRCutFilter InfraRed Cut Filter Insertion/Removal Setting
Enables setting of infrared cut filter insertion/removal.

```
SetIRCutFilter(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMode As Long _           ' Mode selection  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status
In case of not Cascade connection, to set 0

IMode

This designates the status of the cut filter.
The 0 setting removes the filter, and the 1 setting inserts the filter.

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0
In order to get extended error information, uses **GetLastError**

GetIRCutFilter InfraRed Cut Filter Setting Status Request
Requests the setting status of the infrared cut filter.

```
GetIRCutFilter(  
    ByVal IUnit As Long _           ' Device number  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status
In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0
In order to get extended error information, uses **GetLastError**

SetNoiseReduction Noise Reduction Setting
Enables noise reduction ON/OFF setting.

```
SetNoiseReduction(  
    ByVal IUnit As Long, _           ' Device number  
    ByVal IMode As Long _           ' Mode selection  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status
In case of not Cascade connection, to set 0

IMode

Noise reduction is set by the following numbers.

- 0: Noise reduction OFF
- 1: Low level noise reduction ON
- 2: High level noise reduction ON

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0
In order to get extended error information, uses **GetLastError**

GetNoiseReduction Noise Reduction Setting Status Request
Requests the noise reduction setting status.

```
GetNoiseReduction(  
    ByVal IUnit As Long _           ' Device number  
) As Long
```

Parameter

IUnit

To assign device number to 1~9 in case of Cascade status
In case of not Cascade connection, to set 0

Return Value

In case of function successful, returns the value besides 0 and failed, returns 0
In order to get extended error information, uses **GetLastError**