

## 1. Error Code definition

MCAM_ERR_SUCCESS		0
MCAM_ERR_ERROR	-1001	
MCAM_ERR_NOT_INITIALIZED		-1002
MCAM_ERR_NOT_IMPLEMENTED		-1003
MCAM_ERR_RESOURCE_IN_USE		-1004
MCAM_ERR_ACCESS_DENIED		-1005
MCAM_ERR_INVALID_HANDLE		-1006
MCAM_ERR_INVALID_ID		-1007
MCAM_ERR_NO_DATA		-1008
MCAM_ERR_INVALID_PARAMETER		-1009
MCAM_ERR_IO		-1010
MCAM_ERR_TIMEOUT		-1011
MCAM_ERR_ABORT		-1012
MCAM_ERR_INVALID_BUFFER		-1013
MCAM_ERR_NOT_AVAILABLE		-1014
MCAM_ERR_NOT_OPEN_DEVICE		-10000
MCAM_ERR_NO_DEVICE		-10001
MCAM_ERR_RESOURCE_USED		-10002
MCAM_ERR_NO_SYSTEM		-10003
MCAM_ERR_NOT_OPEN_SYSTEM		-10004
MCAM_ERR_INVALID_BUFFER_SIZE		-10005

## 2. General functions

The return value for each function is " Error code "

```
__int32 ST_InitSystem()
```

Creates System module and initializes Library.

All functions can be used after call ST\_InitSystem()[IsInitSystem()]

```
__int32 ST_FreeSystem()
```

Quits all modules in using.

```
__int32 ST_IsInitSystem(bool* pFlag);
```

Checks that Library is available

– pFlag(IN, OUT) : IN : User value pointer, OUT : Init status(0 or 1)

```
__int32 ST_GetAvailableCameraNum(unsigned __int32* pNum)
```

Gets the quantity of connectable Device(Camera)

– pNum(IN, OUT) : IN : User value pointer, OUT : Quantity of connectable device.

```
__int32 ST_UpdateDevice()
```

Updates information of connected device.

```
__int32 ST_GetEnumDeviceID(unsigned __int32 EnumNum, char* pDeviceID, unsigned __int32* pSize)
```

Gets the unique ID of selected device.

- EnumNum(IN) : Enumeration number.
- pDeviceID(IN,OUT) : IN:User buffer pointer, OUT:Device ID.
- pSize(IN,OUT) : IN : User buffer size, OUT : Return string size.

```
__int32 ST_OpenDevice(unsigned __int32 EnumNum, __int32* hDevice)
```

Opens the device.

- EnumNum(IN) : Enumeration number.
- hDevice(IN,OUT) : IN:User value pointer, OUT:Device(camera) handle.

```
__int32 ST_IsOpenDevice(__int32 hDevice, bool* pFlag)
```

Checks the device open status.

- hDevice(IN) : Device handle.
- pFlag(IN, OUT) : IN : User value pointer, OUT : Init status(0 or 1)

```
__int32 ST_CloseDevice(__int32 hDevice)
```

Closes the device.

- hDevice(IN) : Device handle.

```
__int32 ST_AcqStart(__int32 hDevice)
```

Starts Acquisition.

- hDevice(IN) : Device handle.

```
__int32 ST_AcqStop(__int32 hDevice)
```

Stops Acquisition.

- hDevice(IN) : Device handle.

```
__int32 ST_DoAbortGrab(__int32 hDevice)
```

Forcely terminates current grab operation.

- hDevice(IN) : Device handle.

```
__int32 ST_SetAcqInvalidTime(__int32 hDevice, unsigned __int32 Time)
```

Sets the invalid image acquisition time after grab. The images which acquired in AcqInvalidTime are ignored. Please refer to *Fig. 1*, *Fig. 2*.

- hDevice(IN) : Device handle.
- Time(IN) : Acquisition invalid time(ms).

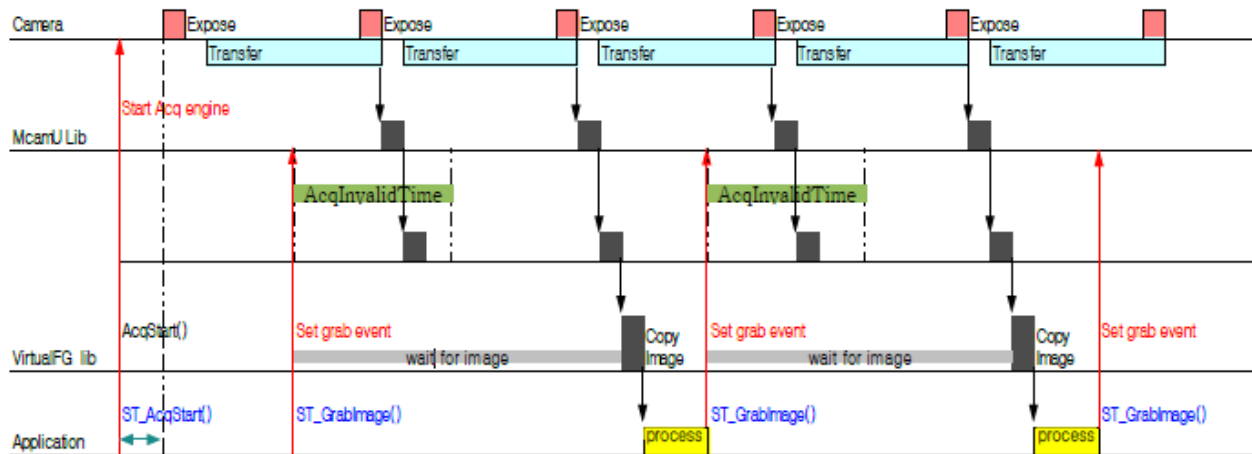


FIG. 1. GrabImage using parameter 'AcqInvalidTime'

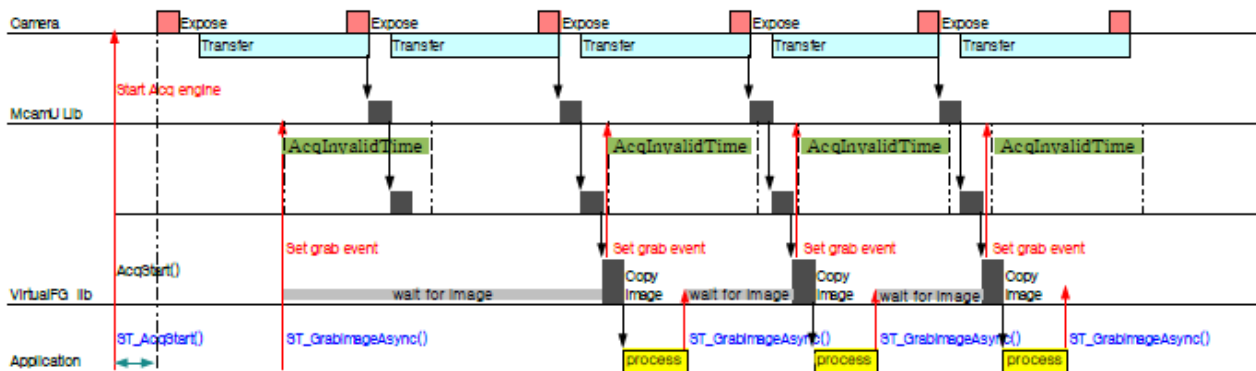


FIG. 2. GrabImageAsync using parameter 'AcqInvalidTime'

```
__int32 ST_GetAcqInvalidTime(__int32 hDevice, unsigned __int32* pTime)
```

Gets the setting `AcqInvalidTime` value.

- `hDevice(IN)` : Device handle.
- `pTime(IN, OUT)` : IN:User value pointer, OUT:Acquisition invalid time(ms).

```
__int32 ST_SetContinuousGrabbing(__int32 hDevice, unsigned __int32 Flag)
```

Continuously acquires images. When ContinuousGrabbing mode is enable(Flag = 1), all acquired images can be transmitted to use the maximum frame rate. Please refer to Fig. 3.

- `hDevice(IN)` : Device handle.

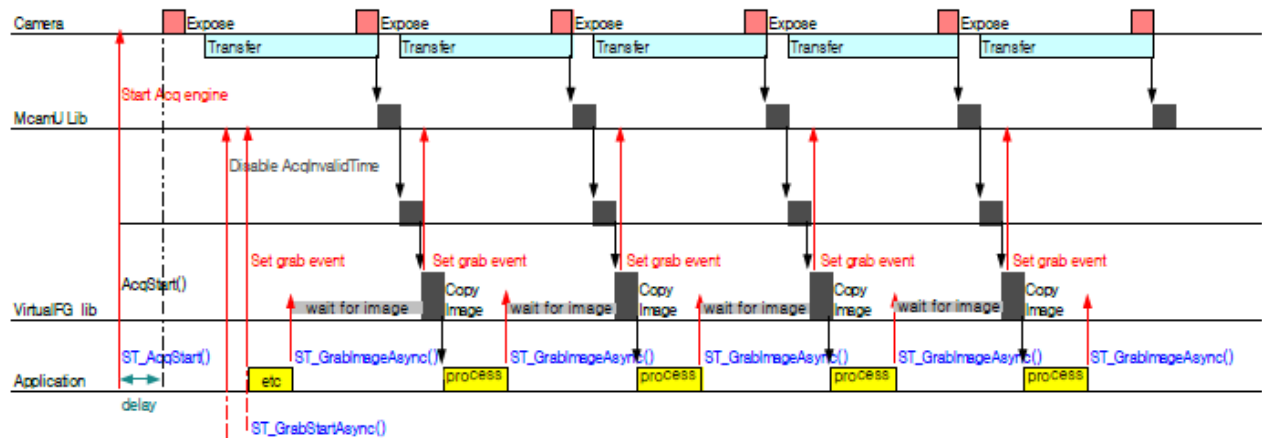


FIG. 3. GrabImageAsync using 'ContinuousGrabbing' mode

```
__int32 ST_GetContinuousGrabbing(__int32 hDevice, unsigned __int32* pFlag)
```

Gets the setting value of ContinuousGrabbing.

- hDevice(IN) : Device handle.
- pFlag(IN, OUT) : IN:User value pointer, OUT:ContinuousGrabbing (0 or 1).

```
__int32 ST_SetGrabTimeout(__int32 hDevice, unsigned __int32 Timeout)
```

Sets the timeout value of the Grab command. If the image can not be acquired until set time by SetGrabTimeout() after call GrabImage() or GrabImageAsync(), It returns MCAMU\_ERR\_TIMEOUT

- hDevice(IN) : Device handle.
- Timeout(IN) : Timeout(ms).

```
__int32 ST_GetGrabTimeout(__int32 hDevice, unsigned __int32* pTimeout)
```

Gets the currently set GrabTimeout value.

- hDevice(IN) : Device handle.
- pTimeout(IN, OUT) : IN:User value pointer, OUT:Grab timeout(ms).

```
__int32 ST_GrabStartAsync(__int32 hDevice, unsigned __int32 MaxDelay)
```

It used to acquire images in Asynchronous mode. It is useful for acquiring images in Trigger mode. If the acquired images are larger than MaxDelay value, the image is ignored and a new image is acquired again. Please refer to Fig. 4.

- hDevice(IN) : Device handle.
- MaxDelay(IN) : Maximum time for acquiring images(ms).

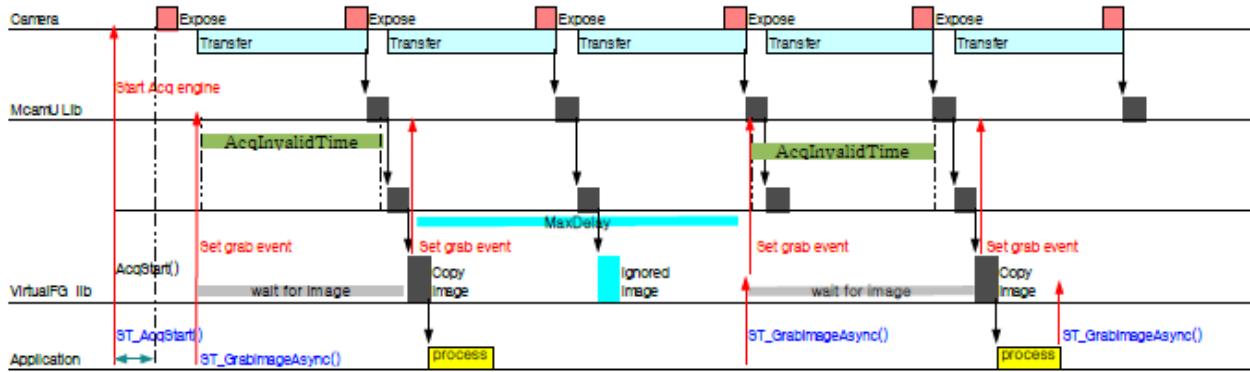


FIG. 4. GrabImageAsync using parameter 'MaxDelay'

```
__int32 ST_GrabImage(__int32 hDevice, void* pDest, unsigned __int32 Buffsize)
```

Acquires images in Synchronous mode. Please refer to Fig. 1.

- hDevice(IN) : Device handle.
- pDest(IN,OUT) : IN:User buffer pointer, OUT : Image data.
- Buffsize(IN) : User buffer size.

```
__int32 ST_GrabImageAsync(__int32 hDevice, void* pDest, unsigned __int32 Buffsize, unsigned __int32 MaxDelay)
```

Acquires images in Asynchronous mode. Please refer to Fig. 2.

- hDevice(IN) : Device handle.
- pDest(IN,OUT) : IN:User buffer pointer, OUT : Image data.
- Buffsize(IN) : User buffer size.
- MaxDelay(IN) : Maximum time for acquiring images(ms)

```
__int32 ST_GetImageAvailable(__int32 hDevice, unsigned __int32* pFlag)
```

It used to checking new image is acquired. When acquiring images in Asynchronous mode if pFlag value is 1, images can be get without waiting.

- hDevice(IN) : Device handle.
- pFlag(IN,OUT) : IN:User value pointer, OUT:ImageAvailable(0 or 1).

```
__int32 ST_CvtColor(void* pSrc, void* pDest, __int32 width, __int32 height, __int32 code)
```

Converts Bayer Pattern to RGB or BGR.

- Src(IN) : User value pointer.
- pDest(IN, OUT) : IN: User value pointer, OUT : Image Data.
- width(IN) : Image width.
- height(IN) : Image height.
- code(IN) : Convert Code :

CVP_BGR2RGB	: BGR → RGB
CVP_RGB2BGR	: RGB → BGR
CVP_BayerBG2RGB	: BayerBG8 → RGB
CVP_BayerGB2RGB	: BayerGB8 → RGB
CVP_BayerRG2RGB	: BayerRG8 → RGB
CVP_BayerGR2RGB	: BayerGR8 → RGB
CVP_YUV2RGB_UYVY	: YUV(UYVY) → RGB
CVP_YUV2BGR_UYVY	: YUV(UYVY) → BGR
CVP_YUV2RGB_YVYU	: YUV(YVYU) → RGB
CVP_YUV2BGR_YVYU	: YUV(YVYU) → BGR
CVP_YUV2RGB_YUYV	: YUV(YUYV) → RGB
CVP_YUV2BGR_YUYV	: YUV(YUYV) → BGR

```
__int32 ST_SetCallbackFunction(__int32 hDevice, __int32 Event, CALLBACKFUNC callbackFunc,  
void* UserData);
```

Registers the Callback function.

- hDevice(IN) : Device handle.
- Event(IN) : EventID :

EVENT\_NEW\_IMAGE

EVENT\_GRAB\_ERROR

- callbackFunc(IN) : Registers the Callback function to call.
- UserData(IN, OUT) : User Define Data.

### 3. Camera control functions

Return value of each function is "Error code".  
For Feature Name, please refer to item 4.

```
__int32 ST_SetIntReg (__int32 hDevice, const char* NodeName, __int32 val)
```

Sets the value on Integer type feature.

- hDevice(IN) : Device handle.
- NodeName : Feature Name
- val(IN) : Integer Value.

```
__int32 ST_GetIntReg (__int32 hDevice, const char* NodeName, __int32* pVal)
```

Gets the value of Integer type Feature.

- hDevice(IN) : Device handle.
- NodeName : NameFeature
- pVal(IN, OUT) : IN:User value pointer, OUT: Integer Value

```
__int32 ST_SetFloatReg(__int32 hDevice, const char* NodeName, double fVal)
```

Sets the value on Float type Feature.

- hDevice(IN) : Device handle.
- NodeName : Feature Name
- fVal(IN) : Float Value.

```
__int32 ST_GetFloatReg(__int32 hDevice, const char* NodeName, double *pFval)
```

Gets the value of Float type Feature.

- hDevice(IN) : Device handle.
- NodeName : NameFeature
- pFval(IN, OUT) : IN:User value pointer, OUT: Float Value.

```
__int32 ST_SetBoolReg(__int32 hDevice, const char* NodeName, bool bVal)
```

Sets the value on Boolean type Feature.

- hDevice(IN) : Device handle.
- NodeName : Feature Name
- fVal(IN) : True or False.

```
__int32 ST_GetBoolReg(__int32 hDevice, const char* NodeName, bool *pBval)
```

Gets the value of Boolean type Feature.

- hDevice(IN) : Device handle.
- NodeName : NameFeature
- pFval(IN, OUT) : IN:User value pointer, OUT: True or False.

```
__int32 ST_SetEnumReg(__int32 hDevice, const char* NodeName, char* val)
```

Sets the value on Enumeration type Feature.

- hDevice(IN) : Device handle.
- NodeName : Feature Name
- val(IN) : Enumeration Value. (refer to Feature Table.)

```
__int32 ST_GetEnumReg(__int32 hDevice, const char* NodeName, char* pInfo, unsigned __int32* pSize)
```

Gets the value of Enumeration type Feature.

- hDevice(IN) : Device handle.
- NodeName : Feature Name
- pInfo(IN,OUT) : IN : User buffer pointer, OUT : Enumeration Value.
- pSize(IN,OUT) : IN : User buffer size, OUT : pInfo length.

```
__int32 ST_GetStrReg(__int32 hDevice, const char* NodeName, char* pInfo, unsigned __int32* pSize)
```

Gets the value of String type Feature.

- hDevice(IN) : Device handle.
- NodeName : Feature Name
- pInfo(IN,OUT) : IN : User buffer pointer, OUT : String Value.
- pSize(IN,OUT) : IN : User buffer size, OUT : pInfo length.

```
__int32 ST_SetCmdReg(__int32 hDevice, const char* NodeName)
```

Sets the value on Command type Feature.

- hDevice(IN) : Device handle.
- NodeName : Feature Name

```
__int32 ST_GetIntRegRange(__int32 hDevice, const char* NodeName, __int32 *pMin, __int32 *pMax, __int32 *pInc)
```

Gets the settable range of Interger type Feature.

- hDevice(IN) : Device handle.
- NodeName : Feature Name
- pMin(IN, OUT) : IN : User value pointer, OUT : Settable minimum value.
- pMax(IN, OUT) : IN : User value pointer, OUT : Settable maximum value.
- pInc(IN, OUT) : IN : User value pointer, OUT : Settable increments.

```
__int32 ST_GetFloatRegRange(__int32 hDevice, const char* NodeName, double *pMin, double *pMax)
```

Gets the settable range of Float type Feature.

- hDevice(IN) : Device handle.
- NodeName : Feature Name
- pMin(IN, OUT) : IN : User value pointer, OUT : Settable minimum value.
- pMax(IN, OUT) : IN : User value pointer, OUT : Settable maximum value.

```
__int32 ST_GetEnumEntrySize(__int32 hDevice, const char* NodeName, __int32 *pVal)
```

Gets the quantity of Enumeration type Feature's Entry

- hDevice(IN) : Device handle.
- NodeName : Feature Name
- pVal(IN, OUT) : IN : User value pointer, OUT : Number of Entry.



```
__int32 ST_GetEnumEntryIntValue(__int32 hDevice, const char* NodeName, __int32 EntryIdx,
__int32 *pVal)
```

Gets the Integer value of Enumeration type Feature's Entry Index.

- hDevice(IN) : Device handle.
- NodeName : Feature Name
- EntryIdx : Entry Index.
- pVal(IN, OUT) : IN : User value pointer, OUT : Integer Value.

```
__int32 ST_GetEnumEntryValue(__int32 hDevice, const char* NodeName, __int32 EntryIdx, char*
pInfo, unsigned __int32 *pSize)
```

Gets the String value of Enumeration type Feature's Entry Index.

- hDevice(IN) : Device handle.
- NodeName : Feature Name
- EntryIdx : Entry Index.
- pInfo(IN,OUT) : IN : User buffer pointer, OUT : String Value.
- pSize(IN,OUT) : IN : User buffer size, OUT : pInfo length.

## 4. Feature Information

DeviceControl						
Feature	Interfaces		Type	AccessMode	Default Value	Range/Value
Description	GigE	USB2.0				
DeviceVendorName	O	O	String	RO		
DeviceVendorName						
DeviceModelName	O	O	String	RO		
DeviceModelName						
DeviceManufacturerInfo	O	O	String	RO		
DeviceManufacturerInfo						
DeviceVersion	O	O	String	RO		
DeviceVersion						
DeviceUserID	O	O	String	RW	0	
DeviceUserID						
DeviceScanType	O	O	Enumeration	RO		
DeviceScanType						

ImageFormatControl							
Feature	Interfaces		Type	AccessMode	Default Value	Range/Value	
Description	GigE	USB2.0					
Width	O	O	Integer	RW			
Width							
Height	O	O	Integer	RW			
Height							
PixelFormat	O	O	Enumeration	RW		GigE Mono	Mono8/10/12, Mono10/12 Packed
						GigE Color	BayerRG8/10/12 BayerRG10/12Packed,YUV422Packed RGB8Packed
						USB Mono	Mono8, Mono10, Mono12, Mono14
						USB Color	Mono8, Bayer8, BayerBG8, RGB8Packed
PixelFormat							
WidthMax	O	O	Integer	RO			
WidthMax							
HeightMax	O	O	Integer	RO			
HeightMax							
OffsetX	O	O	Integer	RW	0		
OffsetX							
OffsetY	O	O	Integer	RW	0		
OffsetY							
BinningHorizontal		Mono only	Integer	RW	1	1,2	
BinningHorizontal							
BinningVertical	Mono only	Mono only	Integer	RW	1	1,2	
BinningVertical							
TestPattern	O		Enumeration	RW	Off	On, Off	
Selects the TestPattern type generated by the device							

AcquisitionControl							
Feature	Interfaces		Type	AccessMode	Default Value	Range/Value	
Description	GigE	USB2.0					
AcquisitionStart	O		command	WO			
장치의 Acquisition을 시작							
AcquisitionStop	O		command	WO			
현재 프레임의 끝에서 Acquisition을 정지							
AcquisitionMode	O		Enumeration	RW	Continuous	Continuous	
Acquistion모드의 설정							
AcquisitionFrameCount	O		Integer	RW	0	0~255	
멀티프레임 Acquisition모드에서 요구한 프레임 수							
TriggerSelector	O		Enumeration	RW	Frame Start	Frame Start	
트리거 타입 설정							
TriggerMode	O		Enumeration	RW	Off	On, Off	
설정한 트리거 조정							
TriggerSource	O		Enumeration	RW	Line1	GigE	Line1, Software
						USB	Line1, TransferEnd, Software
트리거를 위해 사용하는 내부신호나 물리적 입력라인을 명시한다.							
TriggerActivation	O		Enumeration	RW	RisingEdge	GigE	RisingEdge, FallingEdge
						USB	RisingEdge, FallingEdge, Level High, Level Low
트리거 동작모드를 명시한다.							
TriggerSoftware	O		Command	WO			
내부 트리거를 생성한다.							
TriggerDelay	O		Float	RW	0		
트리거 설정 후 작동시키기 전 Delay를 설정한다.							
ExposureMode	O	X	Enumeration	RW	Timed	Timed, TriggerWidth	
노출이나 셔터의 작동시간을 설정							
ExposureAuto	O		Enumeration	RW	Off	Off, Once, Continuous	
노출모드가 시간설정되어있을 때의 자동 노출모드를 설정한다.							
ExposureTime	O		Float	RW	0		
노출모드가 시간설정되어 있고 자동노출이 꺼져 있을 때 노출 시간 설정							
TransferDelay	O	X	Integer	RW	0	0~2147483647	
AcquisitionFrameRateEnable	O		Enumeration	RW	Off	On, Off	
AcquisitionFrameRate	O		Float	RW	1		
프레임 캡처시 acquisition의 비율 조정							
AutoExposureTarget	O		Integer	RW	0	0~255	

DigitalloControl							
Feature	Interfaces		Type	AccessMode	Default Value	Range/Value	
Description	GigE	USB2.0					

LineSelector	O		Enumeration	RW		
환경설정을 위해 외부장치 커넥터의 라인을 설정						
LineMode	O		Enumeration	RW		
물리적 라인의 입출력을 설정						
LineInverter	O		Boolean	RW		
설정 한 입출력라인 신호의 변경을 조절						
LineStatus	O		Boolean	RO		
현재 설정된 입출력라인을 되돌린다.						
LineSource	O		Enumeration	RW		
내부 acquisition 이나 내외부 입력라인을 출력하기 위해 선택						
UserOutputSelector	O		Enumeration	RW	0	
사용자 출력값에 의해 설정된 사용자 출력기록 값을 선택						
UserOutputValue	O		Boolean	RW	0	
UserOutputSelector에 의해 선택된 비트의 값을 설정						

### CounterandTimerControl

Feature	Interfaces		Type	AccessMode	Default Value	Range/Value
Description	GigE	USB2.0				
TimerSelector	O		Enumeration	RW		
환경설정을 위한 타이머 선택						
TimerDuration	O		Float	RW		
Timer Pulse를 위한 기간 설정						
TimerDelay	O		Float	RW		
타이머를 시작 전 트리거 적용을 위한 Delay 기간 설정						

### AnalogControl

Feature	Interfaces		Type	AccessMode	Default Value	Range/Value
Description	GigE	USB2.0				
Gainselector	O		Enumeration	RW		
다양한 Gain 속성들을 조정하기위해 설정						
GainRaw	O		Integer	RW		
GainAuto	O		Enumeration	RW		
BlackLevelSelector	O		Enumeration	RW		
Black Level 설정						
BlackLevelRaw	O		Integer	RW		
BalanceRatioSelector	O		Enumeration	RW	0	
밸런스 비율을 선택						
BalanceRatio	O		Float	RW	0	
선택되어진 컬러요소의 비율을 조절						
BalanceWhiteAuto	O		Enumeration	RW	1	1,2

컬러채널중 화이트 밸런스 모드를 위한 컨트롤

**LUTControl**

Feature	Interfaces		Type	AccessMode	Default Value	Range/Value
Description	GigE	USB2.0				
LUTEnable	O		Boolean	RW		
선택되어진 LUT를 활성화						
LUTIndex	O		Integer	RW		
선택되어진 LUT에 접근하기 위한 offset을 조정						
LUTValue	O		Integer	RW		
LUTSelector에 선택된 LUT의 초기 인덱스 값을 반환한다.						

**TransportLayerControl**

Feature	Interfaces		Type	AccessMode	Default Value	Range/Value
Description	GigE	USB2.0				
PayloadSize	O		Integer	RO		
스트림 채널에 있는 각각의 전송되어진 이미지나 덩어리들의 수를 제공한다						

**GigEVision**

GevVersionMajor	O		Integer	RW		
사양의 주버전						
GevVersionMinor	O		Integer	RW		
사양의 부버전						
GevDeviceModelsBigEndian	O		Boolean	RW		
자이레지스터의 Endianess						
GevDeviceModeChracterSet	O		Enumeration	RW		
부트스트랩 레지스터들의 모든 스트링을 문자화						
GevInerfaceSelector	O		Integer	RW	0	
논리 링크 조절을 위한 선택						
GevMACAddress	O		Integer	RO	0	
논리 링크의 MAC 주소						
GevSupportedOptionSelector	O		Enumeration	RO	1	1,2
존재하는 support의 정보를 얻기위해 GEV옵션 선택						
GevSupportedOption	O		Integer	RO	1	1,2
GEV Option이 지원된다면 되돌린다.						
GevCurrentIPConfigurationLLA	O		Boolean	RO	Off	On, Off
링크 로컬주소 Ip구성방식 주어진 논리적 링크에서의 활성화 여부를 제어						
GevCurrentIPConfigurationDHCP	O		Boolean	RO		
동적IP 할당 구성방식을 주어진 논리적 링크에서의 활성화 여부 제어						
GevCurrentIPConfigurationPersistnetIP	O		Boolean	RO		
영구성IP 구성방식을 주어진논리적링크에서의 활성화 여부 제어						
GevCurrentIPAddress	O		Integer	RO		
주어진 논리적 링크를 위한 IP 주소						
GevCurrentSubnetMask	O		Integer	RO		
주어진 논리적 링크의 서브넷 마스크						

GevCurrentDefaultGateway	O		Integer	RO		
주어RO진 논리적 링크에서의 기본 게이트웨이 IP 주소						
GevFirstURL	O		String	RO		
GeROnIcam XML device descrption 파일을first URL로 가르킴						
GevSecondURL	O		String	RO		
GenIcam XML device descrption 파일을second URL로 가르킴						
GevNumberOfInterfaces	O		Integer	RO		
이 장치에 의해 지원되는 논리적링크의수를 가르킴						
GevPersistentIPAddress	O		Integer	RW		
영구성 IP 조정						
GevPersistentSubnetMask	O		Integer	RW		
영구성 Ip와 관련된 서브넷 마스크 조정						
GevPersistentDefaultGateway	O		Integer	RW		
영구성 기본게이트웨이 조정						
GevMessageChannelCount	O		Integer	RO		
장치에서 지원되는 메세지 채널수를 가르킴						
GevStreamChannelCount	O		Integer	RO		
장치에서 지원되는 스트림 채널의 수를 가르킴						
GevTimestampTickFrequency	O		Integer	RO		
1초안에 찍히는 타임스탬프의 수를 가르킴						
GevTimestampControlLatch	O		Command	WO		
GevTimestampValue 안에 있는 타임 스탬프 카운터를 래치						
GevTimestampControlReset	O		Command	WO		
타임 스탬프 카운터를 0으로 리셋						
GevTimestampValue	O		Integer	RO		
타임 스탬프 카운터의 래치화된 64비트 값을 반환						
GevHeartbeatTimeout	O		Integer	RW		
현재의 heartbeat timeout 값을 조정						
GevGVCPHeartbeatDisable	O		Boolean	RW		
GVCP heartbeat를 사용하지 않음						
GevCCP	O		Enumeration	RW		
Application의 장치접근 제어						
GevMCPHostPort	O		Integer	RW		
메세지를 보내야하는 장치의 포트를 제어						
GevMCTT	O		Integer	RW		
전송 Timeout 값을 제공						
GevMCRC	O		Integer	RW		
메세지 채널의 메세지 허용되는 타임아웃이 발생했을때 재전송되는 수를 컨트롤						
GevStreamChannelSelector	O		Integer	RW		
스트림 채널 선택						
GevSCPIfaceIndex	O		Integer	RW		
사용하기 위한 논리적 링크의 인덱스						
GevSCPHostPort	O		Integer	RW		
GVSP 송신기가 데이터 스트림 또는 GVSP 수신기는 데이터 스트림을 수신 할 수있는 포트를 전송해야하는 선택된 채널의 포트를 제어합니다.						
GevSCPSFireTestPacket	O		Boolean	RW		

테스트 패킷을 보낸다.

GevSCPSDoNotFragment	O		Boolean	RW		
이 기능의 상태는 각 스트림 패킷의 Ip헤더의 “조각 안 함”비트에 복사됩니다.						
GevSCPSBigEndian	O		Boolean	RW		
스트림에서 멀티 바이트 픽셀 데이터의 endianness						
GevSCSPPacketSize	O		Integer	RW		
GVSP transmitter가 선택한 채널에 전송하는 바이트 스트림 패킷 크기를 지정하거나 GVSP receiver에서 지원하는 최대 패킷 크기를 지정합니다.						
GevSCPD	O		Integer	RW		
스트림 채널에 대한 각 패킷 사이에 들어갈 Delay(타임 스탬프 카운터 단위)를 제어						
GevSCDA	O		Integer			
GVSP transmitter가 데이터 스트림 또는 GVSP receiver의 데이터 스트림을 받을 수 있는 대상 Ip주소를 전송해야 하는 선택된 스트림 채널의 Ip주소를 제어						

## UserSetControl

Feature	Interfaces		Type	AccessMode	Default Value	Range/Value
Description	GigE	USB2.0				
UserSetSelector	O		Enumeration	RW		
로드, 저장, 구성 설정 기능 사용을 선택						
UserSetLoad	O		Command	WO		
사용자세팅값을 불러들어오고 활성화 한다.						
UserSetSave	O		Command	WO		
장치의 비휘발성 메모리에 UserSetSelector에 의해 지정된 설정한 사용자 세팅값 저장						

## ColorTransformationControl

Feature	Interfaces		Type	AccessMode	Default Value	Range/Value
Description	GigE	USB2.0				
ColorTransformationEnable	O		Boolean	RW		
선택된 컬러 변환 모듈 활성화						
ColorTransformationValueSelector	O		Enumeration	RW		
선택된 컬러변환 모듈에 접근하는 변환 매트릭스의 Gain 요소나 offset 선택						
ColorTransformationValue	O		Float	RW		
변환 매트릭스 내부의 선택된 Gain요소나 offset값을 나타낸다.						

## DeviceOptionControl

Feature	Interfaces		Type	AccessMode	Default Value	Range/Value
Description	GigE	USB2.0				
DeviceFilterDriverMode	O		Enumeration	RW		
DeviceCommandTimeout	O		Integer	RW		
현재 특정링크의 Command timeout 을 알린다.						
DeviceCommandRetryNumber	O		Integer	RW		
DeviceStreamTimeout	O		Integer	RW		

DeviceMissingPacketReceive	O		Enumeration	RW		
DevicePacketResend	O		Boolean	RW		

MultiTriggerControl						
Feature	Interfaces		Type	AccessMode	Default Value	Range/Value
Description	GigE	USB2.0				
MultiTriggerEnable	O		Enumeration	RW		
멀티트리거함수 사용 유무						
MultiTriggerNumber	O		Integer	RW		
멀티트리거의 수						
MultiTriggerShutter	O		Integer	RW		
멀티트리거 모드의 셔터 시간						
MultiTriggerInterval0	O		Integer	RW		
첫번째와 두번째 펄스 사이의 간격 (Unit : 1line)						
MultiTriggerInterval1	O		Integer	RW		
두번째와 세번째 펄스 사이의 간격 (Unit : 1line)						
MultiTriggerInterval2	O		Integer	RW		
세번째와 네번째 펄스 사이의 간격 (Unit : 1line)						
MultiTriggerInterval3	O		Integer	RW		
네번째와 다섯번째 펄스 사이의 간격 (Unit : 1line)						
MultiTriggerInterval4	O		Integer	RW		
다섯번째와 여섯번째 펄스 사이의 간격 (Unit : 1line)						
MultiTriggerInterval5	O		Integer	RW		
여섯번째와 일곱번째 펄스 사이의 간격 (Unit : 1line)						
MultiTriggerInterval6	O		Integer	RW		
일곱번째와 여덟번째 펄스 사이의 간격 (Unit : 1line)						



## 5. Feature Name Definition

### - Device Control -

MCAM_DEVICE_ID	"DeviceID"	// String
MCAM_DEVICE_VENDOR_NAME	"DeviceVendorName"	// String
MCAM_DEVICE_MODEL_NAME	"DeviceModelName"	// String
MCAM_DEVICE_MANUFACTURER_INFO	"DeviceManufacturerInfo"	// String
MCAM_DEVICE_VERSION	"DeviceVersion"	// String
MCAM_DEVICE_USER_ID	"DeviceUserID"	// String
MCAM_DEVICE_SCANTYPE	"DeviceScanType"	// Enumeration

### - Image Format Control -

MCAM_WIDTH	"Width"	// Integer
MCAM_HEIGHT	"Height"	// Integer
MCAM_PIXEL_FORMAT	"PixelFormat"	// Enumeration
MCAM_WIDTH_MAX	"WidthMax"	// Integer
MCAM_HEIGHT_MAX	"HeightMax"	// Integer
MCAM_OFFSET_X	"OffsetX"	// Integer
MCAM_OFFSET_Y	"OffsetY"	// Integer
MCAM_BINNING_HORIZONTAL	"BinningHorizontal"	// Integer
MCAM_BINNING_VERTICAL	"BinningVertical"	// Integer
MCAM_TEST_PATTERN	"TestPattern"	// Enumeration

### - Acquisition Control -

MCAM_ACQUISITION_START	"AcquisitionStart"	// Command
MCAM_ACQUISITION_STOP	"AcquisitionStop"	// Command
MCAM_ACQUISITION_MODE	"AcquisitionMode"	// Enumeration
MCAM_ACQUISITION_FRAME_COUNT	"AcquisitionFrameCount"	// Integer
MCAM_TRIGGER_SELECTOR	"TriggerSelector"	// Enumeration
MCAM_TRIGGER_MODE	"TriggerMode"	// Enumeration
MCAM_TRIGGER_SOURCE	"TriggerSource"	// Enumeration
MCAM_TRIGGER_ACTIVATION	"TriggerActivation"	// Enumeration
MCAM_TRIGGER_SOFTWARE	"TriggerSoftware"	// Command
MCAM_TRIGGER_DELAY	"TriggerDelay"	// Float
MCAM_EXPOSURE_MODE	"ExposureMode"	// Enumeration
MCAM_EXPOSURE_AUTO	"ExposureAuto"	// Enumeration
MCAM_EXPOSURE_TIME	"ExposureTime"	// Float
MCAM_TRANSFER_DELAY	"TransferDelay"	// Integer
MCAM_ACQUISITION_FRAMERATE_ENABLE	"AcquisitionFrameRateEnable"	// Enumeration
MCAM_ACQUISITIONFRAMERATE	"AcquisitionFrameRate"	// Float
MCAM_AUTO_EXPOSURE_TARGET	"AutoExposureTarget"	// Integer

### - Digital I/O Control -

MCAM_LINE_SELECTOR	"LineSelector"	// Enumeration
MCAM_LINE_MODE	"LineMode"	// Enumeration
MCAM_LINE_INVERTER	"LineInverter"	// Boolean
MCAM_LINE_STATUS	"LineStatus"	// Boolean
MCAM_LINE_SOURCE	"LineSource"	// Enumeration
MCAM_USER_OUTPUT_SELECTOR	"UserOutputSelector"	// Enumeration
MCAM_USER_OUTPUT_VALUE	"UserOutputValue"	// Boolean

### - Counter and Timer Control -

MCAM_TIMER_SELECTOR	"TimerSelector"	// Enumeration
MCAM_TIMER_DURATION	"TimerDuration"	// Float
MCAM_TIMER_DELAY	"TimerDelay"	// Float

### - Analog Control -

MCAM_GAIN_SELECTOR	"GainSelector"	// Enumeration
MCAM_GAIN_RAW	"GainRaw"	// Integer
MCAM_GAIN_AUTO	"GainAuto"	// Enumeration
MCAM_BLACK_LEVEL_SELECTOR	"BlackLevelSelector"	// Enumeration
MCAM_BLACK_LEVEL_RAW	"BlackLevelRaw"	// Integer
MCAM_BALANCE_RATIO_SELECTOR	"BalanceRatioSelector"	// Enumeration
MCAM_BALANCE_RATIO	"BalanceRatio"	// Float
MCAM_BALANCE_WHITE_AUTO	"BalanceWhiteAuto"	// Enumeration

### - LUT Control -

MCAM_LUT_ENABLE	"LUTEnable"	// Boolean
MCAM_LUT_INDEX	"LUTIndex"	// Integer
MCAM_LUT_VALUE	"LUTValue"	// Integer
<b>- Transport Layer Control -</b>		
MCAM_PAYLOAD_SIZE	"PayloadSize"	// Integer
→ <b>GigEVision</b>		
GEV_VERSION_MAJOR	"GevVersionMajor"	// Integer
GEV_VERSION_MINOR	"GevVersionMinor"	// Integer
GEV_DEVICE_MODE_IS_BIGENDIAN	"GevDeviceModelsBigEndian"	// Boolean
GEV_DEVICE_MODE_CHARACTER_SET	"GevDeviceModeCharacterSet"	// Enumeration
GEV_INTERFACE_SELECTOR	"GevInterfaceSelector"	// Integer
GEV_MAC_ADDRESS	"GevMACAddress"	// Integer
GEV_SUPPORTED_OPTION_SELECTOR	"GevSupportedOptionSelector"	// Enumeration
GEV_SUPPORTED_OPTION	"GevSupportedOption"	// Integer
GEV_CURRENT_IP_CONFIGURATION_LLA	"GevCurrentIPConfigurationLLA"	// Boolean
GEV_CURRENT_IP_CONFIGURATION_DHCP	"GevCurrentIPConfigurationDHCP"	// Boolean
GEV_CURRENT_IP_CONFIGURATION_PERSISTENT_IP	"GevCurrentIPConfigurationPersistentIP"	// Boolean
GEV_CURRENT_IP_ADDRESS	"GevCurrentIPAddress"	// Integer
GEV_CURRENT_SUBNETMASK	"GevCurrentSubnetMask"	// Integer
GEV_CURRENT_DEFAULT_GATEWAY	"GevCurrentDefaultGateway"	// Integer
GEV_FIRST_URL	"GevFirstURL"	// String
GEV_SECOND_URL	"GevSecondURL"	// String
GEV_NUMBER_OF_INTERFACES	"GevNumberOfInterfaces"	// Integer
GEV_PERSISTENT_IP_ADDRESS	"GevPersistentIPAddress"	// Integer
GEV_PERSISTENT_SUBNETMASK	"GevPersistentSubnetMask"	// Integer
GEV_PERSISTENT_DEFAULT_GATEWAY	"GevPersistentDefaultGateway"	// Integer
GEV_MESSAGE_CHANNEL_COUNT	"GevMessageChannelCount"	// Integer
GEV_STREAM_CHANNEL_COUNT	"GevStreamChannelCount"	// Integer
GEV_TIME_STAMP_TICK_FREQUENCY	"GevTimestampTickFrequency"	// Integer
GEV_TIME_STAMP_CONTROL_LATCH	"GevTimestampControlLatch"	// Command
GEV_TIME_STAMP_CONTROL_RESET	"GevTimestampControlReset"	// Command
GEV_TIME_STAMP_VALUE	"GevTimestampValue"	// Integer
GEV_HEARTBEAT_TIMEOUT	"GevHeartbeatTimeout"	// Integer
GEV_GVCP_HEARTBEAT_DISABLE	"GevGVCPHeartbeatDisable"	// Boolean
GEV_CCP	"GevCCP"	// Enumeration
GEV_MCP_HOST_PORT	"GevMCPHostPort"	// Integer
GEV_MCTT	"GevMCTT"	// Integer
GEV_MCRC	"GevMCRC"	// Integer
GEV_STREAM_CHANNEL_SELECTOR	"GevStreamChannelSelector"	// Integer
GEV_SCP_INTERFACE_INDEX	"GevSCPInterfaceIndex"	// Integer
GEV_SCP_HOST_PORT	"GevSCPHostPort"	// Integer
GEV_SCPS_FIRE_TEST_PACKET	"GevSCPSFireTestPacket"	// Boolean
GEV_SCPS_DO_NOT_FRAGMENT	"GevSCPSDoNotFragment"	// Boolean
GEV_SCPS_BIG_ENDIAN	"GevSCPSBigEndian"	// Boolean
GEV_SCPS_PACKETSIZE	"GevSCSPacketSize"	// Integer
GEV_SCPD	"GevSCPD"	// Integer
GEV_SCDA	"GevSCDA"	// Integer
<b>- UserSet Control -</b>		
USER_SET_SELECTOR	"UserSetSelector"	// Enumeration
USER_SET_LOAD	"UserSetLoad"	// Command
USER_SET_SAVE	"UserSetSave"	// Command
<b>- Color Transformation Control -</b>		
MCAM_COLOR_TRANSFORMATION_ENABLE	"ColorTransformationEnable"	// Boolean
MCAM_COLOR_TRANSFORMATION_VALUE_SELECTOR	"ColorTransformationValueSelector"	// Enumeration
MCAM_COLOR_TRANSFORMATION_VALUE	"ColorTransformationValue"	// Float
<b>- Device Option Control -</b>		
MCAM_DEVICE_FILTER_DRIVER_MODE	"DeviceFilterDriverMode"	// Enumeration
MCAM_DEVICE_COMMAND_TIMEOUT	"DeviceCommandTimeout"	// Integer
MCAM_DEVICE_COMMAND_RETRY_NUMBER	"DeviceCommandRetryNumber"	// Integer
MCAM_DEVICE_STREAM_TIMEOUT	"DeviceStreamTimeout"	// Integer

MCAM_DEVICE_MISSING_PACKET_RECEIVE	"DeviceMissingPacketReceive"	// Enumeration
MCAM_DEVICE_PACKET_RESEND	"DevicePacketResend"	// Boolean
MCAM_DEVICE_MAX_PACKET_RESEND_COUNT	"DeviceMaxPacketResendCount"	// Integer
MCAM_DEVICE_FIND_MAX_PACKET_SIZE	"DeviceFindMaxPacketSize"	// Command
MCAM_DEVICE_MAX_PACKET_SIZE	"DeviceMaxPacketSize"	// Integer

## 6. Enumeration Feature Entry Definition

PIXEL_FORMAT_MONO8	"Mono8"
PIXEL_FORMAT_MONO10	"Mono10"
PIXEL_FORMAT_MONO12	"Mono12"
PIXEL_FORMAT_MONO14	"Mono14"
PIXEL_FORMAT_MONO10PACKED	"Mono10Packed"
PIXEL_FORMAT_MONO12PACKED	"Mono12Packed"
PIXEL_FORMAT_BAYERBG8	"BayerBG8"
PIXEL_FORMAT_BAYERBG10	"BayerBG10"
PIXEL_FORMAT_BAYERBG12	"BayerBG12"
PIXEL_FORMAT_BAYERBG10PACKED	"BayerBG10Packed"
PIXEL_FORMAT_BAYERBG12PACKED	"BayerBG12Packed"
PIXEL_FORMAT_BAYERRG8	"BayerRG8"
PIXEL_FORMAT_BAYERRG10	"BayerRG10"
PIXEL_FORMAT_BAYERRG12	"BayerRG12"
PIXEL_FORMAT_BAYERRG10PACKED	"BayerRG10Packed"
PIXEL_FORMAT_BAYERRG12PACKED	"BayerRG12Packed"
PIXEL_FORMAT_BAYERGR8	"BayerGR8"
PIXEL_FORMAT_BAYERGR10	"BayerGR10"
PIXEL_FORMAT_BAYERGR12	"BayerGR12"
PIXEL_FORMAT_BAYERGR10PACKED	"BayerGR10Packed"
PIXEL_FORMAT_BAYERGR12PACKED	"BayerGR12Packed"
PIXEL_FORMAT_BAYERGB8	"BayerGB8"
PIXEL_FORMAT_BAYERGB10	"BayerGB10"
PIXEL_FORMAT_BAYERGB12	"BayerGB12"
PIXEL_FORMAT_BAYERGB10PACKED	"BayerGB10Packed"
PIXEL_FORMAT_BAYERGB12PACKED	"BayerGB12Packed"
PIXEL_FORMAT_YUV422PACKED	"YUV422Packed"
PIXEL_FORMAT_RGB8PACKED	"RGB8Packed"
PIXEL_FORMAT_BGR8PACKED	"BGR8Packed"
TEST_PATTERN_OFF	"Off"
TEST_PATTERN_GREY_HORIZONTAL_RAMP	"GreyHorizontalRamp"
TEST_PATTERN_GREY_VERTICAL_RAMP	"GreyVerticalRamp"
ACQUISITION_MODE_CONTINUOUS	"Continuous"
ACQUISITION_MODE_SINGLE_FRAME	"SingleFrame"
ACQUISITION_MODE_MULTI_FRAME	"MultiFrame"
TRIGGER_SELECTOR_FRAME_START	"FrameStart"
TRIGGER_MODE_OFF	"Off"
TRIGGER_MODE_ON	"On"
TRIGGER_SOURCE_LINE1	"Line1"
TRIGGER_SOURCE_SOFTWARE	"Software"
TRIGGER_ACTIVATION_RISING_EDGE	"RisingEdge"
TRIGGER_ACTIVATION_FALLING_EDGE	"FallingEdge"
TRIGGER_ACTIVATION_LEVEL_LOW	"LevelLow"
TRIGGER_ACTIVATION_LEVEL_HIGH	"LevelHigh"
EXPOSURE_MODE_TIMED	"Timed"
EXPOSURE_MODE_TRIGGER_WIDTH	"TriggerWidth"
EXPOSURE_AUTO_OFF	"Off"
EXPOSURE_AUTO_ONCE	"Once"
EXPOSURE_AUTO_CONTINUOUS	"Continuous"
ACQUISITION_FRAMERATE_ENABLE_OFF	"Off"
ACQUISITION_FRAMERATE_ENABLE_ON	"On"
LINE_SELECTOR_LINE1	"Line1"
LINE_SELECTOR_LINE2	"Line2"
LINE_MODE_INPUT	"Input"
LINE_MODE_OUTPUT	"Output"
LINE_SOURCE_OFF	"Off"
LINE_SOURCE_EXPOSURE_ACTIVE	"ExposureActive"
LINE_SOURCE_TIMER_ACTIVE	"TimerActive"
LINE_SOURCE_USER_OUTPUT_1	"UserOutput1"
USER_OUTPUT_SELECTOR_USER_OUTPUT_1	"UserOutput1"
TIMER_SELECTOR_TIMER_1	"Timer1"
GAIN_SELECTOR_ALL	"All"
GAIN_AUTO_OFF	"Off"
GAIN_AUTO_ONCE	"Once"
GAIN_AUTO_CONTINUOUS	"Continuous"

BLACK_LEVEL_SELECTOR_ALL	"All"
BALANCE_RATIO_SELECTOR_RED	"Red"
BALANCE_RATIO_SELECTOR_GREEN	"Green"
BALANCE_RATIO_SELECTOR_BLUE	"Blue"
BALANCE_WHITE_AUTO_OFF	"Off"
BALANCE_WHITE_AUTO_ONCE	"Once"
BALANCE_WHITE_AUTO_CONTINUOUS	"Continuous"
GEV_DEVICE_MODE_CHARACTER_SET_UTF8	"UTF8"
GEV_SUPPORTED_OPTION_SELECTOR_USER_DEFINED_NAME	"UserDefinedName"
GEV_SUPPORTED_OPTION_SELECTOR_SERIAL_NUMBER	"SerialNumber"
GEV_SUPPORTED_OPTION_SELECTOR_HEART_BEAT_DISABLE	"HeartbeatDisable"
GEV_SUPPORTED_OPTION_SELECTOR_LINK_SPEED	"LinkSpeed"
GEV_SUPPORTED_OPTION_SELECTOR_CCP_APPLICATION_SOCKET	"CCPApplicationSocket"
GEV_SUPPORTED_OPTION_SELECTOR_MANIFEST_TABLE	"ManifestTable"
GEV_SUPPORTED_OPTION_SELECTOR_TEST_DATA	"TestData"
GEV_SUPPORTED_OPTION_SELECTOR_DISCOVERY_ACK_DELAY	"DiscoveryAckDelay"
GEV_SUPPORTED_OPTION_SELECTOR_DISCOVERY_ACK_DELAY_WRITABLE	"DiscoveryAckDelayWritable"
GEV_SUPPORTED_OPTION_SELECTOR_EXTENDED_STATUS_CODES	"ExtendedStatusCodes"
GEV_SUPPORTED_OPTION_SELECTOR_PRIMARY_APPLICATION_SWITCH_OVER	"PrimaryApplicationSwitchover"
GEV_SUPPORTED_OPTION_SELECTOR_ACTION	"Action"
GEV_SUPPORTED_OPTION_SELECTOR_PENDING_ACK	"PendingAck"
GEV_SUPPORTED_OPTION_SELECTOR_EVENT_DATA	"EventData"
GEV_SUPPORTED_OPTION_SELECTOR_EVENT	"Event"
GEV_SUPPORTED_OPTION_SELECTOR_PACKET_RESEND	"PacketResend"
GEV_SUPPORTED_OPTION_SELECTOR_WRITE_MEM	"WriteMem"
GEV_SUPPORTED_OPTION_SELECTOR_COMMANDS_CONCATENATION	"CommandsConcatenation"
GEV_SUPPORTED_OPTION_SELECTOR_IP_CONFIGURATION_LLA	"IPConfigurationLLA"
GEV_SUPPORTED_OPTION_SELECTOR_IP_CONFIGURATION_DHCP	"IPConfigurationDHCP"
GEV_SUPPORTED_OPTION_SELECTOR_IP_CONFIGURATION_PERSISTENT_IP	"IPConfigurationPersistentIP"
GEV_SUPPORTED_OPTION_SELECTOR_STREAM_CHANNEL_SOURCE_SOCKET	"StreamChannelSourceSocket"
GEV_SUPPORTED_OPTION_SELECTOR_MESSAGE_CHANNEL_SOURCE_SOCKET	"MessageChannelSourceSocket"
GEV_SUPPORTED_OPTION_SELECTOR_STREAM_CHANNEL_0_BIG_AND_LITTLE_ENDIAN	"StreamChannel0BigAndLittleEndian"
GEV_SUPPORTED_OPTION_SELECTOR_STREAM_CHANNEL_0_IP_REASSEMBLY	"StreamChannel0IPReassembly"
GEV_SUPPORTED_OPTION_SELECTOR_STREAM_CHANNEL_0_UNCONDITIONAL_STREAMING	"StreamChannel0UnconditionalStreaming"
GEV_SUPPORTED_OPTION_SELECTOR_STREAM_CHANNEL_0_EXTENDED_CHUNK_DATA	"StreamChannel0ExtendedChunkData"
GEV_CCP_OPEN_ACCESS	"OpenAccess"
GEV_CCP_EXCLUSIVE_ACCESS	"ExclusiveAccess"
GEV_CCP_CONTROL_ACCESS	"ControlAccess"
USER_SET_SELECTOR_DEFAULT	"Default"
USER_SET_SELECTOR_USER_SET_1	"UserSet1"
USER_SET_SELECTOR_USER_SET_2	"UserSet2"
USER_SET_SELECTOR_USER_SET_3	"UserSet3"
COLOR_TRANSFORMATION_VALUE_SELECTOR_HUEBYGP	"HUEBYGP"
COLOR_TRANSFORMATION_VALUE_SELECTOR_HUEBYGN	"HUEBYGN"
COLOR_TRANSFORMATION_VALUE_SELECTOR_HUEBYHP	"HUEBYHP"
COLOR_TRANSFORMATION_VALUE_SELECTOR_HUEBYHN	"HUEBYHN"
COLOR_TRANSFORMATION_VALUE_SELECTOR_HUERYGP	"HUERYGP"
COLOR_TRANSFORMATION_VALUE_SELECTOR_HUERYGN	"HUERYGN"
COLOR_TRANSFORMATION_VALUE_SELECTOR_HUERYHP	"HUERYHP"
COLOR_TRANSFORMATION_VALUE_SELECTOR_HUERYHN	"HUERYHN"
COLOR_TRANSFORMATION_VALUE_SELECTOR_HUECG	"HUECG"
DEVICE_FILTER_DRIVER_MODE_OFF	"Off"
DEVICE_FILTER_DRIVER_MODE_ON	"On"
DEVICE_MISSING_PACKET_RECEIVE_OFF	"Off"
DEVICE_MISSING_PACKET_RECEIVE_ON	"On"