

1. Error Code definition

```
// Error Code
#define MCAM_ERR_SUCCESS 0
#define MCAM_ERR_INVALID_BUFFER_SIZE -1
#define MCAM_ERR_INVALID_HANDLE -2
#define MCAM_ERR_INVALID_ID -3
#define MCAM_ERR_ACCESS_DENIED -4
#define MCAM_ERR_NO_DATA -5
#define MCAM_ERR_ERROR -6
#define MCAM_ERR_INVALID_PARAMETER -7
#define MCAM_ERR_TIMEOUT -8
#define MCAM_ERR_INVALID_FILENAME -9
#define MCAM_ERR_INVALID_ADDRESS -10
#define MCAM_ERR_FILE_IO -11
#define MCAM_ERR_NOT_OPEN_DEVICE -10000
#define MCAM_ERR_NO_DEVICE -10001
#define MCAM_ERR_RESOURCE_USED -10002
#define MCAM_ERR_NO_SYSTEM -10003
#define MCAM_ERR_NOT_OPEN_SYSTEM -10004
```

2. General functions

각 함수의 Return 값은 "Error code" 입니다.

```
__int32 __stdcall ST_InitSystem()
```

System module 생성 및 Library를 초기화 합니다.

모든 함수들은 ST_InitSystem()[IsInitSystem()] 호출 후 사용 가능 합니다.

```
__int32 __stdcall ST_FreeSystem()
```

사용 중인 모든 Module을 종료합니다.

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

라이브러리가 사용 가능한지 확인합니다.

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

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

연결 가능한 Device(Camera)의 개수를 가져옵니다.

- pNum(IN, OUT) : IN : User value pointer, OUT : 연결 가능한 카메라 개수.

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

선택한 Device(Camera)의 고유 ID를 가져옵니다

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

N 번째 카메라를 Open 합니다.

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

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

Open된 Device를 종료합니다.

- hDevice(IN) : Device handle.

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

Acquisition을 시작합니다.

- hDevice(IN) : Device handle.

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

Acquisition을 정지합니다.

- hDevice(IN) : Device handle.

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

현재 Grab 작업을 강제 종료합니다.

- hDevice(IN) : Device handle.

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

Grab 명령 이후 유효하지 않은 영상 획득 시간을 설정합니다. Grab 명령 후부터 AcqInvalidTime 내에 획득한 영상은 무시됩니다. Fig. 1, Fig. 2 를 참고 바랍니다.

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

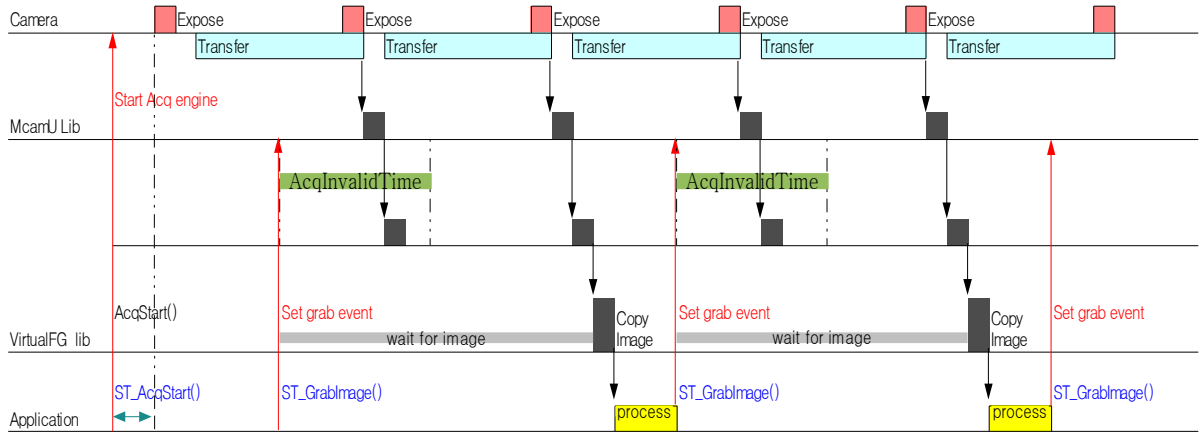


FIG. 1. GrabImage using parameter 'AcqInvalidTime'

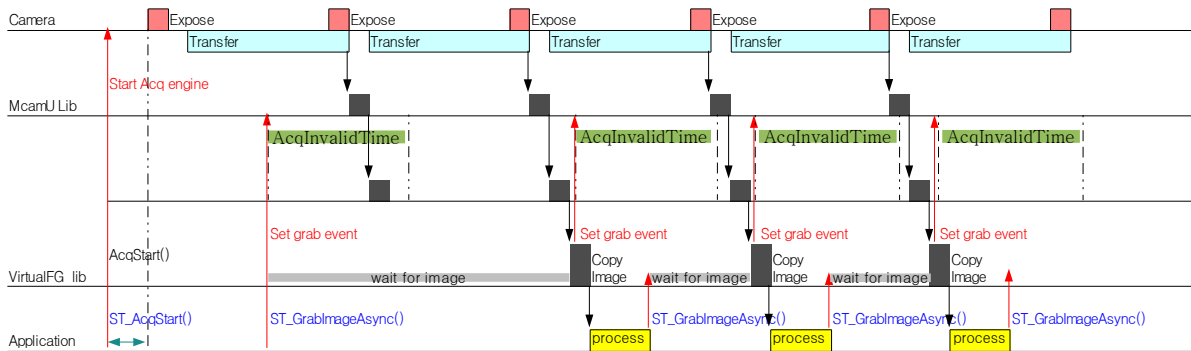


FIG. 2. GrabImageAsync using parameter 'AcqInvalidTime'

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

현재 설정 된 AcqInvalidTime 값을 가져옵니다.

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

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

카메라로 부터 영상을 연속적으로 획득합니다. ContinuousGrabbing 모드를 enable(Flag = 1) 하면 획득된 모든 영상을 전달하여 최대 frame rate를 사용 할 수 있습니다.

Fig. 3.를 참고 바랍니다.

- hDevice(IN) : Device handle.
- Flag(IN) : 0:disable, 1:enable.

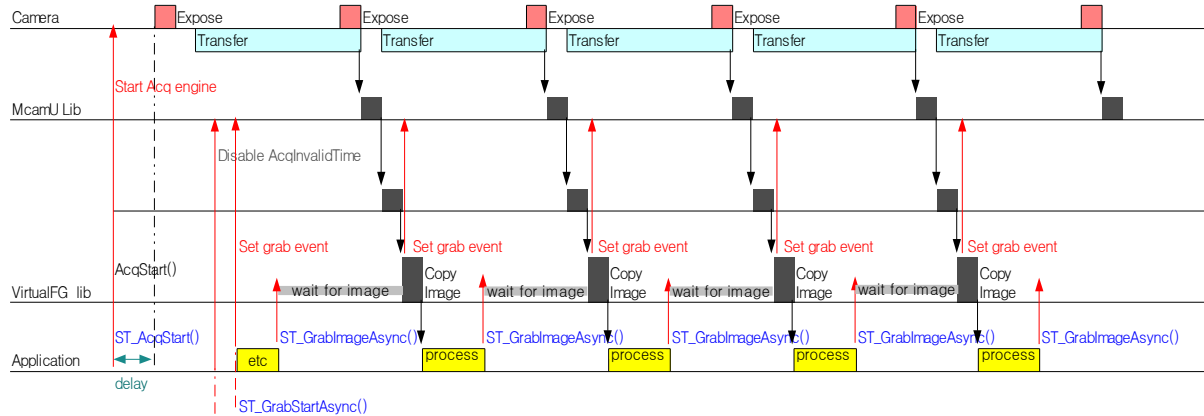


FIG. 3. GrabImageAsync using 'ContinuousGrabbing' mode

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

현재 설정된 ContinuousGrabbing 설정 값을 가져옵니다.

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

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

Grab 명령의 timeout 시간을 설정합니다. GrabImage() 혹은 GrabImageAsync() 함수 호출 후 SetGrabTimeout()에 의해 설정된 시간까지 영상을 획득 할 수 없을 경우 MCAMU_ERR_TIMEOUT 를 반환합니다.

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

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

현재 설정된 GrabTimeout 값을 가져옵니다.

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

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

Asynchronous 모드로 영상을 획득할 때 사용합니다. Trigger mode로 영상을 획득 시 유용합니다. 만약 획득된 영상이 MaxDelay보다 오래된 경우 이 영상은 무시되며 다시 새로운 영상을 획득합니다. Fig. 4.를 참고 바랍니다.

- hDevice(IN) : Device handle.
- MaxDelay(IN) : 영상 획득 최대 유효 시간(ms).

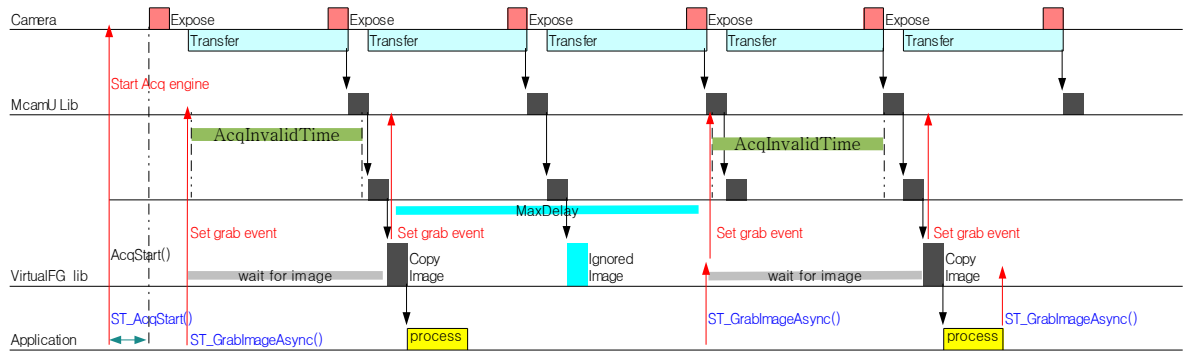


FIG. 4. `GrabImageAsync` using parameter 'MaxDelay'

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

Synchronous 모드로 영상을 획득합니다. Fig. 1.를 참고 바랍니다.

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

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

Asynchronous 모드로 영상을 획득합니다. Fig. 2.를 참고 바랍니다.

- hDevice(IN) : Device handle.
- pDest(IN,OUT) : IN:User buffer pointer, OUT : Image data.
- Buffsize(IN) : User buffer size.
- MaxDelay(IN) : 영상 획득 최대 유효 시간(ms). `ST_GrabStartAsync()` 함수의 파라미터와 같습니다.

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

현재 새로운 영상이 획득 되었는지를 확인할 때 사용 합니다. Asynchronous 모드로 영상을 획득 할 때 pFlag 값이 1 이면 기다리지 않고 영상을 가져올 수 있습니다.

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

3. Camera control functions

각 함수의 Return 값은 "Error code" 입니다.
Feature Name은 4번 항목을 참조 바랍니다.

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

Integer 타입 Feature에 값을 설정합니다.

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

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

Integer타입 Feature의 값을 가져옵니다.

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

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

Float 타입 Feature에 값을 설정합니다.

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

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

Float타입 Feature의 값을 가져옵니다.

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

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

Boolean 타입 Feature에 값을 설정합니다.

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

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

Boolean 타입 Feature의 값을 가져옵니다.

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

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

Enumeration 타입 Feature의 값을 설정합니다.

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

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

Enumeration 타입 Feature의 값을 가져옵니다.

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

String 타입 Feature의 값을 가져옵니다.

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

Command 타입 Feature의 값을 설정합니다.

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

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

Interger 타입 Feature의 설정 가능한 범위를 가져옵니다.

- hDevice(IN) : Device handle.
- NodeName : Feature Name
- pMin(IN, OUT) : IN : User value pointer, OUT : 설정 가능한 최소값.
- pMax(IN, OUT) : IN : User value pointer, OUT : 설정 가능한 최대값.
- pInc(IN, OUT) : IN : User value pointer, OUT : 설정 가능한 증가량.

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

Float 타입 Feature의 설정 가능한 범위를 가져옵니다.

- hDevice(IN) : Device handle.
- NodeName : Feature Name
- pMin(IN, OUT) : IN : User value pointer, OUT : 설정 가능한 최소값.
- pMax(IN, OUT) : IN : User value pointer, OUT : 설정 가능한 최대값.

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

Enumeration 타입 Feature의 Entry 개수를 가져옵니다.

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

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

Enumeration 타입 Feature의 Entry Index에 해당하는 Integer 값을 가져옵니다.

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

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

Enumeration 타입 Feature의 Entry Index에 해당하는 String 값을 가져옵니다.

- 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		
장치 생산자 이름						
DeviceModelName	O	O	String	RO		
장치모델이름						
DeviceManufacturerInfo	O	O	String	RO		
장치생산자 정보						
DeviceVersion	O	O	String	RO		
장치버전						
DeviceUserID	O	O	String	RW	0	
장치사용자 ID						
DeviceScanType	O	O	Enumeration	RO		
장치스캔타입						

ImageFormatControl							
Feature	Interfaces		Type	AccessMode	Default Value	Range/Value	
Description	GigE	USB2.0					
Width	O	O	Integer	RW			
픽셀 내의 가로							
Height	O	O	Integer	RW			
픽셀 내의 세로							
PixelFormat	O	O	Enumeration	RW		GigE Mono	Mono8/10/12, Mono10/12 Packed
						GigE Color	GigE Mono 포함, BayerRG8/10/12 BayerRG10/12Packed, YUV422Packed RGB8Packed
						USB Mono	Mono8, Mono10, Mono12, Mono14
						USB Color	Mono8, Bayer8, BayerBG8, RGB8Packed
장치에서 제공되는 픽셀 포맷							
WidthMax	O	O	Integer	RO			
이미지내 픽셀의 가로 최대							
HeightMax	O	O	Integer	RO			
이미지내 픽셀의 세로 최대							
OffsetX	O	O	Integer	RW	0		
가로 오프셋							
OffsetY	O	O	Integer	RW	0		
세로 오프셋							
BinningHorizontal		Mono only	Integer	RW	1	1,2	
가로픽셀 비닝							
BinningVertical	Mono only	Mono only	Integer	RW	1	1,2	
세로픽셀 비닝							

세로픽셀 비닝						
TestPattern	O		Enumeration	RW	Off	On, Off
장치에 의해 생성된 테스트패턴의 타입을 선택						

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	
Acquisition모드의 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	
설정된 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 시간설정되어 있을 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	

--

DigitalIoControl						
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		
설정된 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
	GigE	USB2.0				
LUTEnable	O		Boolean	RW		
선택되어진 LUT를 활성화						
LUTIndex	O		Integer	RW		
선택되어진 LUT에 접근하기 위한 offset 을 조정						
LUTValue	O		Integer	RW		
LUTSelector에 선택된 LUT의 초기 인덱스 값을 반환한다.						

TransportLayerControl						
Feature	Interfaces		Type	AccessMode	Default Value	Range/Value
	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						
GevDeviceModeCharacterSet	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	
영구성 IP 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	
GeROnlycam XML device descrption 파일을first URL로 가르킴					
GevSecondURL	O		String	RO	
Genlycam 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					
GevGVCPHearteatDisable	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						
GevSCPIInterfaceIndex	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						
GevSCPSPacketSize	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
	GigE	USB2.0				
Description						
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
	GigE	USB2.0				
Description						
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
	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
	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)						