Nanoeye 1.0

Mini-SEM 제어 라이브러리

SEC

2009-02-18

목차

[１. Nanoeye class의 instance 생성 1](#_Toc224549135)

[A) interface INanoeyeFactory 1](#_Toc224549136)

[B) class NanoeyeFactory : INanoeyeFactory 2](#_Toc224549137)

[C) enum ActiveScanTypeEnum : int 2](#_Toc224549138)

[D) enum ControllerTypeEnum : int 2](#_Toc224549139)

[E) enum SettingManagerTypeEnum : int 3](#_Toc224549140)

[F) enum SettingConverterTypeEnum : int 3](#_Toc224549141)

[２. Column 및 High Voltage 제어 4](#_Toc224549142)

[A) interface IController 4](#_Toc224549143)

[B) interface IControlValue<T> 15](#_Toc224549144)

[C) interface IControlValueDouble : IControlValue<T> 16](#_Toc224549145)

[D) interface IControlValueInt : IControlValue<T> 18](#_Toc224549146)

[E) interface IControllerTable 19](#_Toc224549147)

[F) delegate void ObjectArrayEventHandler(object sender, object[] value) 21](#_Toc224549148)

[G) delegate void ColumnEventHandler<T>(object sender, T value, T max, T min, T offset) 21](#_Toc224549149)

[H) delegate void CommunicationErrorEventHandler(object sender, string board, object obj) 21](#_Toc224549150)

[I) enum BeamShiftType : int 21](#_Toc224549151)

[J) enum GunAlingType : int 22](#_Toc224549152)

[K) enum HVcontroltype : int 22](#_Toc224549153)

[L) enum LensType : int 22](#_Toc224549154)

[M) enum StigType : int 22](#_Toc224549155)

[３. Scanning 24](#_Toc224549156)

[A) interface IActiveScan 24](#_Toc224549157)

[B) interface IScanItemEvent 28](#_Toc224549158)

[C) delegate void ScanDataUpdateDelegate(string nameInt32 startline, Int32 lines,) 30](#_Toc224549159)

[４. 설정 관리 31](#_Toc224549160)

[A) interface ISettingConverter 31](#_Toc224549161)

[B) interface ISettingManager 31](#_Toc224549162)

[５. 예제 34](#_Toc224549163)

[A) 객체 생성 34](#_Toc224549164)

[B) Column 초기화 34](#_Toc224549165)

[C) Scanner 초기화 35](#_Toc224549166)

[D) 설정 불러오기 35](#_Toc224549167)

[E) Scanner 동작 시키기 36](#_Toc224549168)

[F) 고압 켜기 36](#_Toc224549169)

[G) 배율 변경 37](#_Toc224549170)

[H) Stig-Wobbling for Stig-Align 37](#_Toc224549171)

# Nanoeye class의 instance 생성

Nanoeye library에 있는 interfce에 인스턴스를 할당한다.

Name Space는 “SEC.Nanoeye”이다.

**사용방법** :

NanoeyeFactory class의 인스턴스를 생성하여 필요한 인스턴스를 NanoeyeFactory 인스턴스를 통해 얻는다.

## interface **INanoeyeFactory**

NanoeyeFactory class에 대한 interface이다.

### Guid - C4A0B93B-7B78-4b3d-883C-36FA8E829C00

### Property

#### int **ActiveScanType**

읽고 쓸 수 있음.

GetNewActiveSan method를 통해 반환될 IActiveScan의 버전.

값은 ActiveScanTypeEnum을 참조 한다.

#### int **ControllerType**

읽고 쓸 수 있음.

GetNewController method를 통해 반환될 IController의 버전.

값은 ControllerTypeEnum을 참조 한다.

#### int **SettingManagerType**

읽고 쓸 수 있음.

GetNewSettingManager method를 통해 반환될 ISettingManager의 버전.

값은 SettingManagerTypeEnum을 참조 한다.

#### int **SettingConverterType**

읽고 쓸 수 있음.

GetNewSettingConverter method를 통해 반환될 ISettingConverter의 버전.

값은 SettingConverterTypeEnum을 참조 한다.

### Method

#### IActiveScan **GetNewActiveScan**()

새로운 IActiveScan의 인스턴스를 반환한다.

Return – ActiveScanType에 의해 선택된 IActiveScan에 할당 가능한 인스턴스.

#### IController **GetNewController**()

새로운 IController의 인스턴스를 반환한다.

Return – ControllerType에 의해 선택된 IController에 할당 가능한 인스턴스.

#### ISettingManager **GetNewSettingManager**()

새로운 ISettingManager의 인스턴스를 반환한다.

Return – SettingManageType에 의해 선택된 ISettingManager에 할당 가능한 인스턴스.

#### ISettingConverter **GetNewSettingConverter**()

새로운 ISettingConverter의 인스턴스를 반환한다.

Return – SettingConverterType에 의핸 선택된 ISettingConverter에 할당 가능한 인스턴스.

## class **NanoeyeFactory** : INanoeyeFactory

INanoeyeFactory interface의 구현 class. INanoeyeFactory와 동일함.

### Guid - 932416BB-3AFE-4517-83F1-3E112F6B96E7

## enum **ActiveScanTypeEnum** : int

### NotSelected = 0 : 선택 되지 않음.

### Version001 = 1 : 첫번째 버전

## enum **ControllerTypeEnum** : int

### NotSelected = 0 : 선택 되지 않음.

### Mini\_SEM = 1 : Mini-SEM Controller

### Carryer = 2 : Carryer Type SEM Controller

## enum **SettingManagerTypeEnum** : int

### NotSelected = 0 : 선택 되지 않음.

### Version001 = 1 : 첫번째 버전

## enum **SettingConverterTypeEnum** : int

### NotSelected = 0 : 선택 되지 않음.

### Version001 = 1 : 첫번째 버전.

# Column 및 High Voltage 제어

Column 및 High Voltage 제어를 위한 부분이다.

Name Space는 “SEC.Nanoeye.NanoView”이다.

**사용방법** :

먼저 IController interfcae를 상속한 ControllerMiniSEM를 생성한다.

IController에서 GetPortList method를 이용하여 사용가능한 Port List를 얻은 후, 그중 하나의 포트를 이용하여 ColumnInit(string port)를 호출, 객체를 초기화 한다.

이후에는 변경이 필요한 값을 변경 하면 된다.

앞으로 설명에중에 Read에 관한 특별한 설명이 없다면, 해당 IControlValue의 Read 및 RepeatRead 동작은 지원 되지 않거나, 지원 되더라도 가비지 값이 리턴 되므로 사용 하지 않는다.

## interface **IController**

Column 및 HV 제어를 위한 인터페이스

### Guid - 5CCD9C50-2202-4845-8CCF-5E5CA53C846B

### Property

#### IControlValueDouble **BeamShiftX**

읽기 전용

Beam Shift X의 이동 값을 설정 한다.

DefaultMin : -1.0

DefaultMax : 1.0

Precision : 0.00025

단위 : 비율

#### IControlValueDouble **BeamShiftY**

읽기 전용

Beam Shift Y의 이동 값을 설정 한다.

DefaultMin : -1.0

DefaultMax : 1.0

Precision : 0.00025

단위 : 비율

#### IControlValueDouble **Clt**

읽기 전용

Collector의 출력 전압을 설정 한다.

DefaultMin : 0

DefaultMax : 1.0

Precision : 약 0.004

단위 : 비율

Read : double[2] – double[0] = current

double[1] = voltage

#### IControlValueDouble **Eghv**

읽기 전용

Electoron Gun의 출력 전압을 설정 한다.

DefaultMin : 0

DefaultMax : 1.0

Precision : 약 0.004

단위 : 비율

Read : double[2] – double[0] = current

double[1] = voltage

#### IControlValueInt **FeedbackMode**

읽기 전용

Scan Coil에 흐르는 전류에 대한 feedback을 받을 때 사용하는 모드를 결정 한다.

0은 고배율 모드용(Scan Coil 출력이 약 0.1% 이하 일 때)이며, 1이면 저배율 모드용 이다.

DefaultMin : 0

DefaultMax : 1

단위 : 없음.

#### IControlValueDouble **Grid**

읽기 전용

Grid의 출력 전압을 설정 한다.

DefaultMin : 0

DefaultMax : 1.0

Precision : 약 0.004

단위 : 비율

Read : double[2] – double[0] = current

double[1] = voltage

#### IControlValueDouble **GunAlignX**

읽기 전용

Gun Aling X Coil에 흐르를 전류의 비율을 설정 한다.

DefaultMin : -1.0

DefaultMax : 1.0

Precision : 0.00025

단위 : 비율

#### IControlValueDouble **GunAlignY**

읽기 전용

Gun Aling Y Coil에 흐르를 전류의 비율을 설정 한다.

DefaultMin : -1.0

DefaultMax : 1.0

Precision : 0.00025

단위 : 비율

#### string **HVtext**

읽고 쓸수 있음.

HV 설정에 대한 표시 Text이다.

#### bool **Initialized**

읽기 전용

객체가 초기화 되었는지를 나타낸다.

객체가 초기화 되기 전에는 사용 할 수 없다.

#### IControlValueDouble **LensCondenser1**

읽기 전용

First Condenser Lens의 출력을 설정 한다.

DefaultMin : 0

DefaultMax : 1.0

Precision : 약 0.004

단위 : 비율

#### IControlValueInt **LensCondenser1Direction**

읽기 전용

First Condenser Lens의 출력 방향을 설정 한다.

DefaultMin : 0

DefaultMax : 1

단위 : 없음

#### IControlValueDouble **LensCondenser2**

읽기 전용

Second Condenser Lens의 출력을 설정 한다.

DefaultMin : 0

DefaultMax : 1.0

Precision : 약 0.004

단위 : 비율

#### IControlValueInt **LensCondenser2Direction**

읽기 전용

Seconde Condenser Lens의 출력 방향을 설정 한다.

DefaultMin : 0

DefaultMax : 1

단위 : 없음

#### IControlValueDouble **LensObject1**

읽기 전용

Object Lens의 출력을 coarse 하게 설정 한다.

DefaultMin : 0

DefaultMax : 1.0

Precision : 약 0.00025

단위 : 비율

#### IControlValueDouble **LensObject2**

읽기 전용

Object Lens의 출력을 fine 하게 설정 한다.

DefaultMin : 0

DefaultMax : 1.0

Precision : 약 0.00025

단위 : 비율

#### IControlValueInt **LensObjectDirection**

읽기 전용

Object Lens의 출력 방향 설정 한다.

DefaultMin : 0

DefaultMax : 1

단위 : 없음

#### IControlTable **Magnification**

읽기 전용

배율을 설정 한다. 이 값을 변경하면 MagnificationX, MagnificaionY, FeedbackMode가 설정값에 따라 변경된다.

TableAppend(object[] values)의 형태

values[0] – int. 배율

values[1] – double. MagnificationX의 값.

values[2] – double. MagnificationY의 값.

values[3] – int. FeedbackMode의 값.

SetStyle(…)의 parameters

Linear magnification mode : 배율의 Calibration 배율의 중간 배율로도 설정 가능하도록 함. SeletedIndex를 이용해서 증감을 하면 된다.

index = 0,

value = 0 – 사용 안함

1 – 실제 배율 차의 반씩 증감.

2 – 실제 배율 차의 1/4씩 증감.

WD releate mode : WD에 따른 배율 보상을 함. 보상 결과는 SelectedItems에 나타난다.

index = 1

value = 0 – 사용 안함

1 – 배율 보상 함.

#### IControlValueDouble **MagnificationX**

읽기 전용

Scan Coil X의 출력을 설정 한다.

DefaultMin : 0

DefaultMax : 1.0

Precision : 약 0.000001

단위 : 비율

#### IControlValueDouble **MagnificationY**

읽기 전용

Scan Coil Y의 출력을 설정 한다.

DefaultMin : 0

DefaultMax : 1.0

Precision : 약 0.000001

단위 : 비율

#### string **Name**

읽기고 쓸수 있음

객체의 이름

#### IControlValueDouble **Pmt**

읽기 전용

PMT의 출력을 설정 한다.

DefaultMin : 0

DefaultMax : 1.0

Precision : 약 0.004

단위 : 비율

Read : double[2] – double[0] = current

double[1] = voltage

#### IControlValueDouble **Rotation**

읽기 전용

Scan 출력을 회전 시킨다.

DefaultMin : -180.0

DefaultMax : 180.0

Precision : 약 0.001

단위 : degree

#### IControlValueDouble **ScanAmplitudeX**

읽기 전용

Scan reference X 신호의 증폭율 이다. 0.707 이상으로 설정 할 경우 Rotation 시 이미지가 찌그러 질 수 있다. 프로그램에서 0.707로 설정해 주어야 한다.

DefaultMin : 0

DefaultMax : 1.0

Precision : 약 0.0005

단위 : 비율

#### IControlValueDouble **ScanAmplitudeY**

읽기 전용

Scan reference Y 신호의 증폭율 이다. 0.707 이상으로 설정 할 경우 Rotation 시 이미지가 찌그러 질 수 있다. 프로그램에서 0.707로 설정해 주어야 한다.

기본값 : 0.707

DefaultMin : 0

DefaultMax : 1.0

Precision : 약 0.0005

단위 : 비율

#### IControlValueInt **ScanReference**

읽기 전용

Scan reference 신호의 포트 번호를 결정 한다.

현재 지원 되지 않고있다.

#### Int32 **Sequence**

읽고 쓸 수 있음

UI 표시 및 Setting Manager 지원을 위한 값.

#### IControlValueDouble **StigXab**

읽기 전용

Stig X Coil의 align을 위한 A-B 코일의 비율을 설정 한다.

DefaultMin : -1.0

DefaultMax : 1.0

Precision : 0.0002

단위 : 비율

#### IControlValueDouble **StigXcd**

읽기 전용

Stig X Coil의 align을 위한 C-D 코일의 비율을 설정 한다.

DefaultMin : -1.0

DefaultMax : 1.0

Precision : 0.0002

단위 : 비율

#### IControlValueDouble **StigXValue**

읽기 전용

Stig X Coil의 값을 설정 한다.

DefaultMin : -1.0

DefaultMax : 1.0

Precision : 0.00025

단위 : 비율

#### IControlValueDouble **StigYab**

읽기 전용

Stig Y Coil의 align을 위한 A-B 코일의 비율을 설정 한다.

DefaultMin : -1.0

DefaultMax : 1.0

Precision : 0.0002

단위 : 비율

#### IControlValueDouble **StigYcd**

읽기 전용

Stig Y Coil의 align을 위한 C-D 코일의 비율을 설정 한다.

DefaultMin : -1.0

DefaultMax : 1.0

Precision : 0.0002

단위 : 비율

#### IControlValueDouble **StigYValue**

읽기 전용

Stig Y Coil의 값을 설정 한다.

DefaultMin : -1.0

DefaultMax : 1.0

Precision : 0.00025

단위 : 비율

#### IControlValueDouble **Tip**

읽기 전용

TIP의 출력 전압을 설정 한다.

DefaultMin : 0

DefaultMax : 1.0

Precision : 약 0.004

단위 : 비율

Read : double[2] – double[0] = current

double[1] = voltage

#### IControlValueInt **VacuumState**

읽기 전용

현재의 진공 정보를 가져 온다.

Repeat Update및 Read 만 사용 할 수 있다.

Repeat Update및 Read 동작시 진공 상태에 대한 문자열이 리턴된다.

#### IControlTable **Workingdistance**

읽기 전용

작업 거리 설정. 이 값을 변경하면 MagnificationX, MagnificaionY, FeedbackMode가 설정값에 따라 변경된다.

TableAppend(object[] values)의 형태

values[0] – double. mm단위의 작업 거리

values[1] – double. LensObject1의 값

values[2] – double. reserved

SetStyle(…)의 parameters

NotSupported.

### Method

#### bool **ColumnInit**(string port)

객체를 초기화 한다.

Parameter :

BSTR port : 사용할 Mini-SEM과 연결된 port 이름

Return : 초기화 성공 여부

#### Int32 **ControlBoard**(out string[,] information)

Controller에 있는 각종 보드의 정보

string 2차원 배열 이며, 첫번째 row에는 보드의 이름, 두번째 row에는 Firmware의 compile 날짜, 세번째 row에는 firmware의 compile 시간이 들어 있다.

Parameter :

out string[,] information : Board 정보

Return : Board의 개수

#### string **GetControllerType**()

Controller의 vender 이름

return : vender 이름

#### string[] **GetPortList**()

Mini-SEM 연결된 Serial port의 이름 목록

return : string array로 Mini-SEM이 연결된 serial port의 이름 목록이다.

#### void **HVmonitor**(Int32 type, out double current, out double voltage)

HV의 구동 정보를 읽어 온다.

Parameter :

Int32 type : HV type 번호.

0 – Electorn Gun

1 – Tip

2 – Grid

3 – Collector (지원 하지 않음)

4 – PMT (지원 하지 않음)

out double current : 소비되는 전류

out double voltage : 인가된 전압

#### bool **LensSyncScan**(Int32 lenstype, bool enable, Int32 gain)

Condenser lens나 Object lens의 출력을 Scan 파형에 동기호 시킨다.

Parameter :

Int32 lenstype : 동기화 시킬 Lens

0 – Sync Scan를 사용할지 여부를 결정. 이와 함께 enable에 false가  
 false 전달되면 모든 lens의 sync scan이 동작 하지 않는다.

1 – First Condenser Lens

2 – Second Condenser Lens

3 – Reserved

4 – Object Lens Coarse

5 – Object Lens Fine

bool : 동시화 시킬지 여부

Int32 gain : reference scan 파형의 증폭율

#### bool **LensSyncScanState**(Int32 lenstype)

Condenser lens나 Object lens의 Sync Scan이 동작 중인지 여부를 확인 한다.

Parameter :

Int32 lenstype : 확인할 Lens

0 – Sync Scan를 사용할지 여부를 결정. 이와 함께 enable에 false가  
 false 전달되면 모든 lens의 sync scan이 동작 하지 않는다.

1 – First Condenser Lens

2 – Second Condenser Lens

3 – Reserved

4 – Object Lens Coarse

5 – Object Lens Fine

Return : 동기화 여부

#### bool **LensWobbling**(Int32 lenstype, bool enable, Int32 freq, Int32 amplitude)

Condenser lens나 Object lens의 wobbling 동작을 설정 한다.

Parameter :

Int32 lenstype : Wobbling 할 Lens

1 – First Condenser Lens

2 – Second Condenser Lens

3 – Object Lens

bool enable : Woobling을 사용할지 여부

Int32 freq : wobbling 주기

Int32 amplitude : wobbling 진폭

Return : 항상 true

#### bool **LensWobblingState**(Int32 lenstype)

Condenser lens나 Object lens의 wobbling이 동작 중인지 여부를 확인 한다.

Parameter :

Int32 lenstype : 확인할 Lens

1 – First Condenser Lens

2 – Second Condenser Lens

3 – Object

Return : wobbling 여부

#### bool **StigSyncScan**(Int32 stig, bool enable, Int32 gain)

Stig Coil을 reference scan 신호에 동기화(비점동기화)여부를 설정 한다.

Parameter :

Int32 stig : 동기화할 Stig Coil

0 – Stig X

1 – Stig Y

bool : 동기화 여부

Int32 gain : reserved

Return : 항상 true

#### bool **StigSyncScanState**(Int32 stig)

Stig Coil을 reference scan 신호에 동기화(비점동기화) 상태를 확인 한다.

Parameter :

Int32 stig : 동기화할 Stig Coil

0 – Stig X

1 – Stig Y

Return : 동기화 여부

#### bool **StigWobbling**(Int32 sitg, bool enable, Int32 freq, Int32 amplitude);

Stig Coil의 woobling 동작을 설정 한다.

Parameter :

Int32 stig : wobbling할 stig coil

0 – Stig X

1 – Stig Y

bool enable : wobbling 여부

Int32 freq : wobbling 주기

Int32 amplitude : wobbling 진폭

Return : 항상 true

#### bool **StigWobblingState**(Int32 stig);

Stig Coil의 wobbling 상태를 확인한다.

Int32 stig : 확인한 stig coil

0 – Stig X

1 – Stig Y

Return : wobbling 여부

#### double **VacuumGauge**(Int32 number);

진공 게이지의 값을 읽는다.

Parameter :

Int32 number : 진공 게이지 번호

Return : 진공 게이지의 값

### Event

#### event CommunicationErrorEventHandler **CommunicationError**

통신이 실패 할 경우 발생 한다.

### Indexer

#### object[] **this**[string board]

읽기 전용

보드 이름을 통해 보드가 제어하는 IControlValue의 목록을 가져옴.

Parameter :

string board – 보드 이름

Return :

IControlValue Array

## interface **IControlValue**<T>

제네릭으로 구현된 Controller 제어 인수의 interface 이다.

IControlValueDouble과 IControlValueInt의 기본형이다.

### Property

#### T **DefaultMax**

읽기 전용

System에서 지원하는 최대 값

#### T **DefaultMin**

읽기 전용

System에서 지원하는 최소 값

#### T **Maximum**

읽고 쓸수 있음

동작 가능한 최대 값

범위 : Minimum ~ DefaultMax

#### T **Minimum**

읽고 쓸수 있음

동작 가능한 최소 값

범위 : DefaultMin ~ Maximum

#### T **Offset**

읽고 쓸수 있음

offset. Value 가 변경 될 경우, Controller에는 Value+Offset의 값이 전달 된다.

범위 : MinValue + Value ~ MaxValue – Value

#### T **Precision**

읽고 전용

Value 제어의 정밀도.

#### T **Value**

읽고 쓸수 있음

설정 값. Controller에는 Value+Offset의 값이 전달 된다.

범위 : MinValue + Offset ~ MaxValue – Offset

#### Object[] **Read**

읽기 전용

동작에 의한 feedback

### Event

#### event ObjectArrayEventHandler **RepeatUpdated**

주기적으로 값을 읽어오는 동작에 의해 발생하는 이벤트

#### event ColumnEventHandler<T> **ValueChanged**

Maximum, Minimum, Value, offset이 바뀌었음을 알림.

## interface **IControlValueDouble** : IControlValue<T>

double 자료형을 사용하는 Controller 제어 인수의 interface 이다.

IController 객체로부터 BeamShiftX나 MagnificationX등의 interface를 이 interfce를 이용하여 query 후 interface에 정의된 property나 method등을 이용하여 동작을 취하면 된다.

### Guid - B58E3A5A-9220-461e-B179-780359C9D1EC

### Property

#### double **DefaultMax**

읽기 전용

System에서 지원하는 최대 값

#### double **DefaultMin**

읽기 전용

System에서 지원하는 최소 값

#### double **Maximum**

읽고 쓸수 있음

동작 가능한 최대 값

범위 : Minimum ~ DefaultMax

#### double **Minimum**

읽고 쓸수 있음

동작 가능한 최소 값

범위 : DefaultMin ~ Maximum

#### double **Offset**

읽고 쓸수 있음

offset. Value 가 변경 될 경우, Controller에는 Value+Offset의 값이 전달 된다.

범위 : MinValue + Value ~ MaxValue – Value

#### double **Precision**

읽고 전용

Value 제어의 정밀도.

#### double **Value**

읽고 쓸수 있음

설정 값. Controller에는 Value+Offset의 값이 전달 된다.

범위 : MinValue + Offset ~ MaxValue – Offset

#### Object[] **Read**

읽기 전용

동작에 의한 feedback

### Event

#### event ObjectArrayEventHandler **RepeatUpdated**

주기적으로 값을 읽어오는 동작에 의해 발생하는 이벤트

#### event ColumnEventHandler<double> **ValueChanged**

Maximum, Minimum, Value, offset이 바뀌었음을 알림.

## interface **IControlValueInt** : IControlValue<T>

int 자료형을 사용하는 Controller 제어 인수의 interface 이다.

IController 객체로부터 EGHV나 TIP등의 interface를 이 interfce를 이용하여 query 후 interface에 정의된 property나 method등을 이용하여 동작을 취하면 된다.

### Guid - BF7D8BDE-A1A8-431c-9C4B-AFD74D862508

### Property

#### Int32 **DefaultMax**

읽기 전용

System에서 지원하는 최대 값

#### Int32 **DefaultMin**

읽기 전용

System에서 지원하는 최소 값

#### Int32 **Maximum**

읽고 쓸수 있음

동작 가능한 최대 값

범위 : Minimum ~ DefaultMax

#### Int32 **Minimum**

읽고 쓸수 있음

동작 가능한 최소 값

범위 : DefaultMin ~ Maximum

#### Int32 **Offset**

읽고 쓸수 있음

offset. Value 가 변경 될 경우, Controller에는 Value+Offset의 값이 전달 된다.

범위 : MinValue + Value ~ MaxValue – Value

#### double **Precision**

읽고 전용

Value 제어의 정밀도.

기본 값 : 1

#### Int32 **Value**

읽고 쓸수 있음

설정 값. Controller에는 Value+Offset의 값이 전달 된다.

범위 : MinValue + Offset ~ MaxValue – Offset

#### Object[] **Read**

읽기 전용

### Event

#### event ObjectArrayEventHandler **RepeatUpdated**

주기적으로 값을 읽어오는 동작에 의해 발생하는 이벤트

#### event ColumnEventHandler<Int32> **ValueChanged**

Maximum, Minimum, Value, offset이 바뀌었음을 알림.

## interface IControllerTable

실제 Controller의 값을 조절 하지 않고, 배율이나 WD 같이 여러 값이 동시에 변경 되는 것에 대해 테이블을 이용해서 제어하는 인터패이스를 제공.

### Guid

### Property

#### int Length

읽기 전용.

테이블의 총 아이템의 개수를 가져 온다.

#### int SelectedIndex

읽고 쓸수 있음.

테이블의 특정 아이템을 선택한다.

#### object SelectedItem

읽고 쓸수 있음.

테이블에서 대표값을 이용해서 자료에 값을 선택하거나, 선택된 값을 가져온다.

### Method

#### void SetStyle(int index, object value)

부가적인 설정을 한다.

Parameter :

int index – 설정 번호

object value – 설정 값

#### void TableAppend(object[] values)

테이블에 값을 추가 한다. 배열의 첫번째 요소가 대표 값이다.

Parameter :

object[] values – 테이블 값

#### void TableChange(object preKey, object[] values)

특정 key의 값을 새로운 값으로 대체 한다.

Parameters :

object preKey – 값을 변경할 테이블 요소의 대표 값

object[] values – 변경할 새로운 값. 값의 현테는 “TableAppend(…)” method와 동일 하다.

#### object[,] TableGet()

테이블의 값을 가져온다.

Return :

테이블의 값. 각 행의 값은 “TableAppend(,,,)” method와 동일 하다.

#### void TableRemove(object key)

대표값을 이용해서 테이블의 요소를 삭제 한다ㅣ.

Parameters :

object key – 삭제할 요소의 대표 값.

#### void TableSet(object[,] values)

테이블의 기존 값을 삭제 하고, 새로운 값으로 대체 한다.

Parameters :

object[,] values – 새로운 테이블 값

### Event

#### EventHandler SelectedIndexChanged

특정 요소가 선택 되었음을 알림.

## delegate void **ObjectArrayEventHandler**(object sender, object[] value)

IControlValue의 RepeatRead 이벤트에 대한 delegate

Parameter :

object sender : 이벤트를 발생시킨 객체

object[] value : 읽어 들인 값들.

## delegate void **ColumnEventHandler<T>**(object sender, T value, T max, T min, T offset)

IControlValue의 ValueChanged 이벤트에 대한 delegate.

Parameter :

object sender : 이벤트를 발생시킨 객체

T value : value

T max : maximum

T min : minimum

T offset : offset

## delegate void **CommunicationErrorEventHandler**(object sender, string board, object obj)

IControlValue의 Communicationerror 이벤트에 대한 delegate

Parameter :

object sender : 이벤트를 발생시킨 IController 객체

string board : 통신에 실패한 보드 명

object obj : 통신에 실패한 IControlValue 객체

## enum **BeamShiftType** : int

Beam Shift Coil에 대한 제어 번호.

### BeamShiftX = 0 : Beam Shift X Coil

### BeamShiftY = 1 : Beasm Shift Y Coil

## enum **GunAlingType** : int

Gun Align Coil에 대한 제어 번호.

### GunAlignX = 0 : Gun Align X Coil

### GunAlignY = 1 : Gun Align Y Coil

## enum **HVcontroltype** : int

High Voltage에 대한 제어 번호.

### EghV = 0 : Electron Gun

### Tip = 1 : Tip

### Grid = 2 : Grid

### Clt = 3 : Collector

### Pmt = 4 : Photo multifly tube

## enum **LensType** : int

Lens에 대한 제어 번호

### All = 0 : All Lens

### Condenser1 = 1 : First Condenser Lens

### Condenser2 = 2 : Seconde Condenser Lens

### Oject = 3 : Object Lens

### ObjectCoarse = 4 : Object Lens control as coarse

### ObjectFine = 5 : Object Lens control as fine

## enum **StigType** : int

Stig 제어 번호

### StigX = 0 : Stig X Coil

### StigY = 1 : Stig Y Coil

# Scanning

주사 관련 부분 이다.

Name Space는 “SEC.Nanoeye.NanoImage”이다.

**사용방법** :

먼저 IActiveScan interfcae를 상속한 ActiveScan를 생성한다.

필요에 따라서는 ActiveScan에서 GetDevList method를 이용하여 사용가능한 DAQ 장치 목록을 얻은 후, 그중 하나의 장치를 이용하여 Initialize(string port)를 호출, 객체를 초기화 한다.

이후에 ScannerCreate method를 이용하여 주사 동작을 미리 정의 한 후, ScannerChange method를 이용하여 주사를 한다.

GetScanEvent method를 이용하여 각 주사 설정별 이벤트의 interface 객체를 얻은 후, 그 객체의 이벤트에 등록하여 획득한 이미지를 가져와 화면에 표시하면 된다.

## interface **IActiveScan**

scanning 과 이미지 획득을 관리 한다.

이름을 이용하여 접근 하며, 시작은 "ScannerChange"와 "ScannerSqeunceRun" method로 할 수 있고, 중지는 "Stop" method로 할 수 있다.

### Guid - 36125F73-E73F-433f-ABB9-944D1975876E

### Property

#### string **DaqDevice**

읽기 전용

사용되고 있는 DAQ 장치의 이름

### Method

#### string[] **GetDevList**()

사용 가능한 DAQ 장치 목록을 가져 옮.

Returns : DAQ 장치 목록

#### IScanItemEvent **GetScanEvent**(string name)

주사 설정의 event 연결용 interface를 가져옮.

Parameter : stirng name - 주사 설정 명

Return : event 연결용 interface

#### bool **Initialize**(string dev)

DAQ 장치를 이용하여 초기화 함.

Parameter : string dev - DAQ 장치 명

Return : 성공 여부

#### bool **IsRun**(string item)

주사 설정의 동작 여부를 확인 함

Parameter : string item - 주사 설정 명

Return : 동작 여부

#### void **ReStart**(string name)

주사 설정을 재시작 한다.

Parameter : string name - 설정 명.

#### bool **ScannerChange**(string name, Int32 count)

주사 설정을 실행 시킴

Parameter : string name - 주사 설정 명

Int32 count - 이미지 획득 프레임 수(0이면 계속 획득)

Return : 성공 여부

#### bool **ScannerClone**(string name, string newName)

특정 주사 설정과 동일한 주사 설정을 새로 생성 함.

Parameter : string name - 기존 주사 설정 명

string newName - 새로 생성할 주사 설정 명

Return : 성공 여부

#### bool **ScannerCreate**(string name, double[] sets)

주사 설정을 생성 함

설정 순서 –

0: int - AI Channel(0~3)

1: double - AI Sampling frequence(1~1250000)

2: int - AI Differencial(0-false, 1-true)

3: float - AI Signal Maximum(0~10)

4: float - AI Signal Minimum(-10~0)

5: double - AO Sampling frequence(half of AI Sampling frequence)

6: float - AO Signal Maximum(10)

7: float - AO Signal Minimum(-10)

8: int - Scanning Frame Height(pixels)

9: int - Scanning Frame Width(pixels)

10: double - PropergationDelay from ao to ai( > 0)

11: double - Scanning Signal X width (0~1)

12: double - Scanning Signal Y width (0~1)

13: double - Scanning Signal X Shift (-1~1)

14: double - Scanning Signal Y Shift (-1~1)

15: int - Frame Average(0~8)

16: int - Image Bluring(0~31)

17: int - Sample Composite(0~32)

18: int - Image Height(pixels)

19: int - Image X start position from Frame(pixels)

20: int - Image Y start position from Frame(pixels)

21: int - Image Width(pixels)

22: double - Display Start X(0~1)

23: double - Display Start Y(0~1)

24: double - Display Width(0~1)

25: double - Display Height(0~1)

Parameter : string name - 주사 설정 명

double[] sets - 설정 값들

Return : 성공 여부

#### bool **ScannerDelete**(string name)

주사 설정을 삭제 함

Parameter : string name - 주사 설정 명

Return : 성공 여부

#### bool **ScannerSqeunceRun**(string[] names)

여러 주사 설정을 동시에 실행 시킴.

Parameter : string[] names - 주사 설정 명들

Return : 성공 여부

#### bool **SettingChange**(string name, Int32 target, double value)

주사 설정의 특정 설정 값을 변경 함.

target 번호 –

0: int - AI Channel(0~3)

1: double - AI Sampling frequence(1~1250000)

2: int - AI Differencial(0-false, 1-true)

3: float - AI Signal Maximum(0~10)

4: float - AI Signal Minimum(-10~0)

5: double - AO Sampling frequence(half of AI Sampling frequence)

6: float - AO Signal Maximum(10)

7: float - AO Signal Minimum(-10)

8: int - Scanning Frame Height(pixels)

9: int - Scanning Frame Width(pixels)

10: double - PropergationDelay from ao to ai( > 0)

11: double - Scanning Signal X width (0~1)

12: double - Scanning Signal Y width (0~1)

13: double - Scanning Signal X Shift (-1~1)

14: double - Scanning Signal Y Shift (-1~1)

15: int - Frame Average(0~8)

16: int - Image Bluring(0~31)

17: int - Sample Composite(0~32)

18: int - Image Height(pixels)

19: int - Image X start position from Frame(pixels)

20: int - Image Y start position from Frame(pixels)

21: int - Image Width(pixels)

22: double - Display Start X(0~1)

23: double - Display Start Y(0~1)

24: double - Display Width(0~1)

25: double - Display Height(0~1)

Parameter : string name - 주사 설정 명

Int32 target - 설정 번호

double value - 값

Return : 성공 여부

#### double **SettingGet**(string name, Int32 target)

주사 설정의 설정 값을 가져 옴.

target 번호 –

0: int - AI Channel(0~3)

1: double - AI Sampling frequence(1~1250000)

2: int - AI Differencial(0-false, 1-true)

3: float - AI Signal Maximum(0~10)

4: float - AI Signal Minimum(-10~0)

5: double - AO Sampling frequence(half of AI Sampling frequence)

6: float - AO Signal Maximum(10)

7: float - AO Signal Minimum(-10)

8: int - Scanning Frame Height(pixels)

9: int - Scanning Frame Width(pixels)

10: double - PropergationDelay from ao to ai( > 0)

11: double - Scanning Signal X width (0~1)

12: double - Scanning Signal Y width (0~1)

13: double - Scanning Signal X Shift (-1~1)

14: double - Scanning Signal Y Shift (-1~1)

15: int - Frame Average(0~8)

16: int - Image Bluring(0~31)

17: int - Sample Composite(0~32)

18: int - Image Height(pixels)

19: int - Image X start position from Frame(pixels)

20: int - Image Y start position from Frame(pixels)

21: int - Image Width(pixels)

22: double - Display Start X(0~1)

23: double - Display Start Y(0~1)

24: double - Display Width(0~1)

25: double - Display Height(0~1)

Parameter : string name - 설정 명

Int32 target - 설정 번호

Return : 설정 값

#### double[] **SettingGets**(string name)

주사 설정의 설정 값들을 가져옴.

Parameter : string name - 설정 명

Return : 설정 값들

#### void **Stop**()

주사 동작을 중지 시킴

### Event

#### event EventHandler **ScanningStarted**

Scanning이 시작 되었음을 알림.

#### event EventHandler **ScanningStopped**

Scanning이 중지 되었음을 알림.

## interface **IScanItemEvent**

개별적 주사 설정의 동작과 관련된 인터페이스.

### Guid - 49C49484-0C82-44ed-81D1-74F0ECF9ADFF

### Property

#### IntPtr **ImageData**

읽기 전용

frame average와 bluring이 적용된 수집한 이미지

#### int ImageHeight

이미지의 높이(픽셀 단위)

#### int ImageWidth

이미지의 넓이(픽셀 단위)

#### string **Name**

읽기 전용

주사 설정의 이름

#### double PaintHeight

이미지의 표시 높이(0~1)

#### double PaintWidth

이미지의 포시 넓이(0~1)

#### double PaintX

이미지의 표시 시작 X축 위치(0~1)

#### double PaintY

이미지의 표시 시작 Y축 위치(0~1)

#### int ScanningScanned

수집한 이미지의 프레임 수

### \Event

#### event ScanDataUpdateDelegate **FrameUpdated**;

프레임 전체가 업데이트 됨을 알림.

#### event ScanDataUpdateDelegate **ScanLineUpdated**;

새로운 이미지 데이타가 업데이트 됨을 알림.

#### event EventHandler settingchangedEvent

설정이 바뀌었음을 알림.

## delegate void **ScanDataUpdateDelegate**(string nameInt32 startline, Int32 lines,)

스캔 데이타가 업데이트 됨을 알림

Parameter : string name - 주사 설정 이름

Int32 startline – 획득한 이미지의 시작 라인

Int32 lines – 획득한 라인 수

# 설정 관리

설정 파일의 버전을 변환 하고, 설정 파일을 이용해서 객체를 초기화 하거나, 객체의 정보를 저장한다.

## interface ISettingConverter

설정 파일의 버전을 변환한다.

### Guid - D8B76229-2274-4b71-89AC-3EBDC985AD1A

### Method

#### void **Convert**(string oriName, string newName, string newVer)

설정을 변환 한다.

Parameter : string oriName – 기존 설정 파일 명

string newName – 새로운 설정 파일 명

string newVer – 새로운 설정 버전

#### string **GetSettingVersion**(string filename)

설정 파일의 버젼을 가져 온다.

Parameter : string filename –파일 명

Return : 버전

## interface I**SettingManager**

설정 파일에서 설정을 불러와 IController나 IActiveScan을 초기화 하거나, 설정 파일에 저장한다.

### Guid - 0446E5A8-D31A-4b3b-8001-8E0FAEEEC17C

### Property

#### bool **Inited**

읽기 전용

초기화 여부

### Method

#### void **Save**(string fileName)

설정을 저장 함.

Parameter : string filename – 설정 파일 명

#### void **Load**(string fileName)

설정을 불러 옴.

Parameter : string filename – 설정 파일 명

#### void **ActiveScanInit**(NanoImage.IActiveScan scanner)

설정을 이용하여 ActiveScan에 주사 설정들을 생성 함

Parameter : IActiveScan scanner - 주사 설정을 생성할 ActiveScan

#### void **ActiveScanCreate**(NanoImage.IActiveScan scanner, string name)

새로운 주사 설정을 ActiveScan 및 설장 파일에 생성 함

Parameter : IActivveScan scanner - 새로운 주사 설정을 생성할 ActiveScan

string name - 주사 설정 명

#### void **ActiveScanDelete**(NanoImage.IActiveScan scanner, string name)

ActiveScan과 설정 파일에서 주사 설정을 삭제 함

Parameter : IActivveScan scanner - 주사 설정을 삭제할 ActiveScan

string name - 주사 설정 명

#### void **ActiveScanSettingSave**(NanoImage.IActiveScan scanner, string name)

ActiveScan의 특정 주사 설정을 설정 파일에 저장 함

Parameter : IActivveScan scanner - 주사 설정을 저장할 ActiveScan

string name - 주사 설정 명

#### string[] **ActiveScanList**()

설정 파일에 있는 주사 설정의 목록을 가져 옴.

Return : 주사 설정 목록

#### void **ControllerCreate**(string name)

새로운 Controller 설정을 생성 함.

Parameter : string name - Controller 설정 명

#### void **ControllerDelete**(string name)

Controller 설정을 삭제 함.

Parameter : string name - Controller 설정 명

#### string[] **ControllerList**()

사용 가능한 controller 설정 목록

Return : 설정 목록

#### void **ControllerChange**(NanoView.IController controller, string name)

Controller의 값을 설정 파일로 부터 user 설정으로 초기화 한다.

Parameter : IController controller - IController

string name - 설정 명

#### void **ControllerDefault**(NanoView.IController controller, string name)

Controller의 값을 설정 파일로 부터 Factory 설정으로 초기화 한다.

Parameter : IController controller - IController

string name - 설정 명

#### void **ControllerSettingSave**(NanoView.IController controller)

controller의 값을들 설정 파일에 저장 한다.

Parameter : IController controller - IController

# 예제

## 객체 생성

IActiveScan scanner;

ISettingManager setManage;

IController column;

INanoeyeFactory nanoeyeFct;

...

void CreateObjets()

{

nanoeyeFct = new NanoeyeFactory();

nanoeyeFct.ActiveScanType = (int)(ActiveScanTypeEnum.Version001);

nanoeyeFct.ControllerType = (int)(ControllerTypeEnum.Mini\_SEM);

nanoeyeFct.SettingConverterType = (int)(SettingConverterTypeEnum.Version001);

nanoeyeFct.SettingManagerType = (int)(SettingManagerTypeEnum.Version001);

setManage = nanoeyeFct.GetNewSettingManager();

scanner = nanoeyeFct.GetNewActiveScan();

column = nanoeyeFct.GetNewController();

}

## Column 초기화

void ColumnInit(IController column)

{

string[] items = column.GetPortList();

if ( items == null )

{

// Mini-SEM과 연결 되지 않은 경우 처리

}

string port = // items에서 하나를 선택 해야 함.

column.ColumnInit(port);

}

## Scanner 초기화

void ScannerInit(IActiveScan scanner)

{

string[] items = scanner.GetDevList();

if ( items == null )

{

// DAQ 장치가 연결 되지 않은 경우 처리

}

string dev = // items에서 하나를 선택 해야 함.

scanner.Initialize(dev);

}

## 설정 불러오기

void SettingLoad(ISettingManager setManage, IActiveScan scanner, IController column)

{

string setFileName = @".\Nanoeye001.config";

string oriName = @".\NanoeyeSEM.config";

if ( File.Exists(setFileName) )

{

}

else if ( File.Exists(oriName) )

{

ISettingConverter conv = nanoeyeFct.GetNewSettingConverter();

conv.Convert(oriName, setFileName, "Nanoeye001");

}

else

{

CreateNewConfig(setManage, \_SetFileName);

}

setManage.Load(newName);

setManage.ActiveScanInit(scanner);

string[] items = setManage.ControllerList();

setManage.ControllerChange(column, items[0]);

}

void CreateNewConfig(ISettingManager setManage, string filename)

{

setManage.ControllerCreate("Column 0");

setManage.ActiveScanCreate(scanner, "Fast Scan");

setManage.ActiveScanCreate(scanner, "Slow Scan");

setManage.ActiveScanCreate(scanner, "Fast Photo");

setManage.ActiveScanCreate(scanner, "Slow Photo");

setManage.ActiveScanCreate(scanner, "Scan Pause");

setManage.Save(filename);

setManage = null;

setManage = nanoeyeFct.GetNewSettingManager();

}

## Scanner 동작 시키기

void RunScanner(IActiveScan scanner, string mode)

{

scanner.ScannerChange(mode, 0);

IScanItemEvent isie = scanner.GetScanEvent(mode);

// Scanning 이벤트에 대해 isie에 이벤트 연결을 통해 이미지를 획득 해야 함.

}

## 고압 켜기

void HVon(IController column)

{

column.EgpsEnable(true);

column.ScanAmplitudeX.Value = 0.707;

column.ScanAmplitudeY.Value = 0.707;

System.Threading.Thread.Sleep(3000);

column.Eghv.Value = column.Eghv.Value;

column.Tip.Value = column.Tip.Value;

column.Grid.Value = column.Grid.Value;

column.Pmt.Value = column.Pmt.Value;

column.Clt.Value = column.Clt.Value;

System.Threading.Thread.Sleep(1000);

}

## 배율 변경

void MagChange(IController column, int mag)

{

column.Magnification.SelectedIndex = mag;

// "MagnificationX", "MagnificationY", "FeedbackMode"의 값이

// "Magnification"에 의해 자동으로 바뀌므로 위의 세 값을 건드릴 필요가 없음.

}

## Stig-Wobbling for Stig-Align

void StigWobblingOn()

{

column.StigWobbling((int)StigType.StigX, true, 1, 127);

column.StigWobbling((int)StigType.StigY, true, 1, 127);

}

void StigWobblingOff()

{

column.StigWobbling((int)StigType.StigX, false, 1, 127);

column.StigWobbling((int)StigType.StigY, false, 1, 127);

}