

AT-Viewer User Manual

AiST Tech X-ray Detector

ver. 3.5.0

AiST Tech inc.

Tel : 063-852-5400 Web Site : <http://www.aisttech.co.kr>

Copyright © 2019 AiST Tech

목 차

1. AT-Viewer	3
1.1 프로그램 개요	3
1.2 설치	3
1.3 설치내용	3
2. User Interface	4
3. 버튼	5
4. 속성	6
5. 상태창	10
6. 캘리브레이션	11
7. 촬영	13
8. Software Development	16
8.1 소개	16
8.2 순서도	17
8.3 소프트웨어 컨트롤	18
8.4 구동 관련 명령어	19
8.5 속성 관련 명령어	22
8.6 출력 관련 명령어	33
9. 에러코드	35



1. AT-Viewer

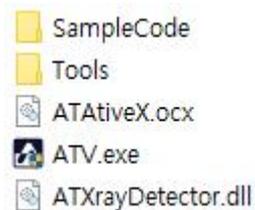
1.1 프로그램 개요

AiST Tech Viewer는 사용자가 X-ray Detector를 통해 쉽고 간단하게 피사체를 촬영하고, 촬영된 image를 검토할 수 있도록 만들어졌습니다.

1.2 설치

AT-ViewerSetup-3.5.0.exe 설치파일을 실행해 진행합니다.

1.3 설치내용



① Capture

촬영된 image가 저장되는 폴더입니다.

② SampleCode

소프트웨어 개발 도구(SDK)가 SampleCode\Lib 폴더에 포함 되어있습니다.
ATXrayDetector.dll 라이브러리를 이용한 샘플코드가 있습니다.

③ Tools

USB Driver 설치파일 및 사용자 매뉴얼이 포함 되어있습니다.

④ ATActiveX.ocx

객체 연결 및 맞춤형 컨트롤로 운영 체제에 등록하여 Detector를 제어 할 수 있습니다.

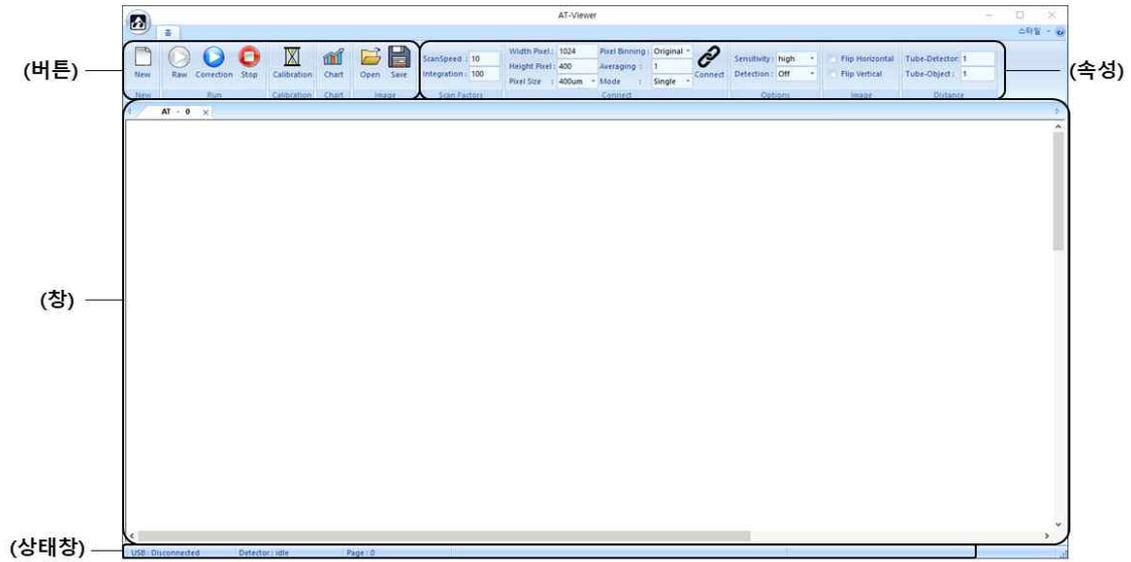
⑤ ATV.exe

Detector를 제어 할 수 있는 실행 파일입니다.



2. User Interface

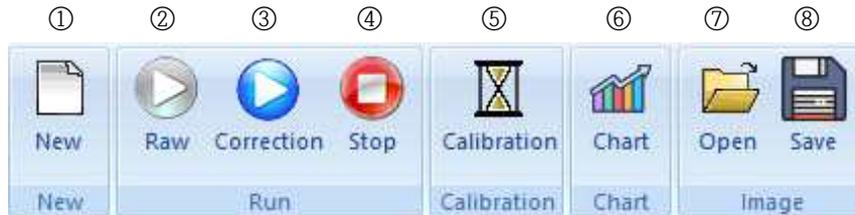
프로그램 실행 시 처음으로 보이는 화면입니다.





3. 버튼

촬영에 필요한 여러 가지 동작 기능을 가진 버튼입니다.



① New



새로운 Document를 생성합니다. Document당 하나의 Detector를 연결이 가능합니다.
선택된 Document 활성화하여 설정된 정보로 동작 시키거나 연결시킬 수 있습니다.

② Raw

Raw image 취득을 위한 동작모드를 실행합니다.

③ Correction

Correction image 취득을 위한 동작모드를 실행합니다.

Calibration이 선행 되어야합니다.

④ Stop

현재 동작중인 동작모드를 중지시켜 image의 취득을 중지합니다.

⑤ Calibration

Correction image를 얻기 위해 Calibration data를 추출하는 동작을 수행합니다.

⑥ Chart

실시간 취득되는 image(Raw, Correction)를 Line 단위로 에너지 그래프를 표현합니다.

⑦ Open

image 파일이 저장된 폴더를 불러옵니다.

⑧ Save

현재 동작모드로 수집되는 image를 tif 형식으로 저장합니다.



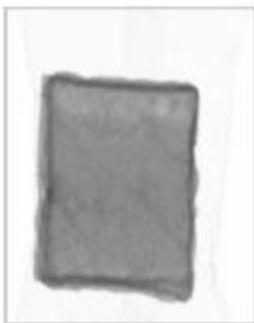
4. 속성

촬영에 필요한 여러 가지 속성을 변경하거나 확인 할 수 있는 창입니다.



① Scan Factors

(i) ScanSpeed : 일반적으로 컨베이어 속도와 같습니다. Scan Speed에 따라 내부적으로 Integration time이 변경됩니다. 값이 클수록 노출 시간이 짧아집니다. 피사체의 움직임과 스캔 스피드가 같아야 정상 크기의 image를 얻을 수 있습니다.



피사체 > ScanSpeed

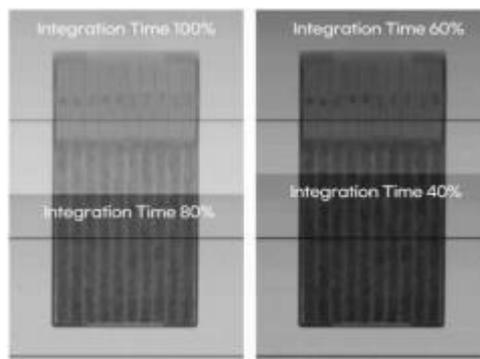


피사체 = ScanSpeed



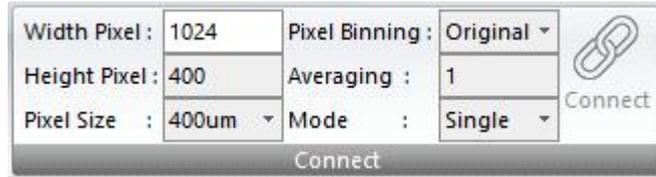
피사체 < ScanSpeed

(ii) Integration time : ScanSpeed에 따라 정해진 Integration Time을 Percentage(%) 단위로 조절합니다. Scan Speed는 변하지 않고 실제 촬영되는 노출 시간만 변화시켜 에너지를 조절 할 수 있습니다.

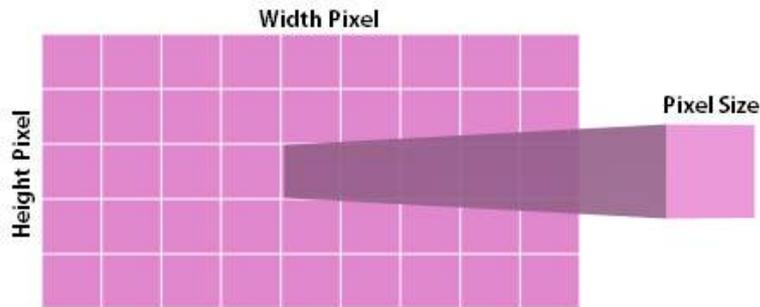




② Connect



- (i) Width Pixel : Detector에서 출력되는 하나의 Line을 구성하는 픽셀수입니다.
- (ii) Height Pixel : 임의의 Frame 구성을 위한 세로 Line 수이며 100 이상이어야 합니다.
- (iii) Pixel Size : Detector의 픽셀 사이즈입니다. 종류로는 400 μ m, 200 μ m, 100 μ m 가 있습니다.

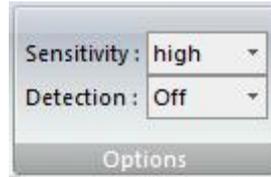


- (iv) Pixel Binning : 인접한 여러 픽셀을 하나의 픽셀로 표현합니다. 종류는 Original, Sum x2, Avg x2, Sum x4, Avg x4 가 있습니다.
예) Sum x2는 인접한 2개의 픽셀 값을 합하여 표현.
Avg x2는 인접한 2개의 픽셀 값을 평균하여 표현.
- (v) Mode : 디텍터의 종류를 선택합니다. 종류로는 Single, Dual, TDI가 있습니다.

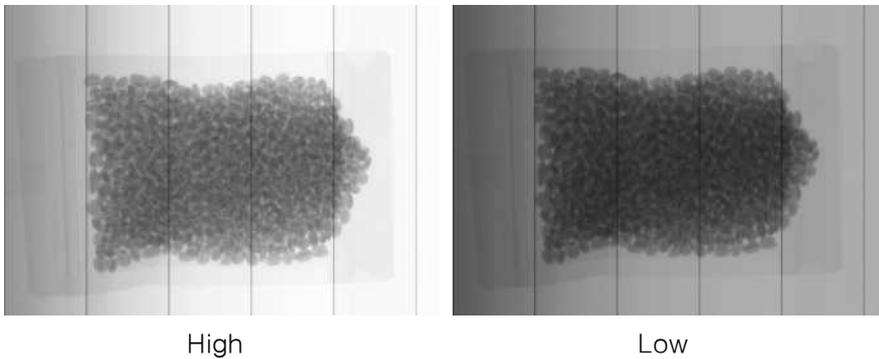
Connect 속성 값을 입력 후  Connect 버튼을 눌러 연결 시도합니다.
성공 혹은 실패 메시지가 출력됩니다. Connect 속성 변경할 때에만 버튼이 활성화 됩니다.
속성 값을 다르게 입력 시 잘못된 image가 출력 될 수 있습니다.



③ Option



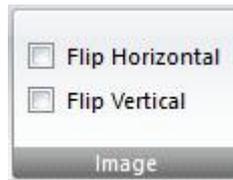
- (i) Sensitivity: 피사체를 찍을 때, 밀도가 높은 부분을 영상으로 추출할 것인지 낮은 부분을 추출할 것인지 선택할 수 있습니다.



- (ii) Detection : Detection 기능은 피사체가 감지되었을 때만 촬영을 하는 기능입니다. Calibration 데이터가 있어야하고 Correction 모드로 사용될 때 작동합니다. 배경 대비 감지하려는 피사체의 Gray 임계값을 설정할 수 있습니다. 임계값(Threshold)의 기본값은 30이며 범위는 10~100입니다.

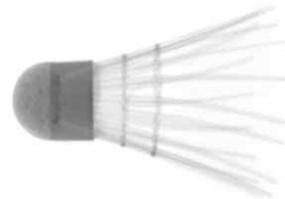
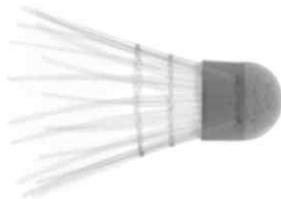


④ Image



(i) Flip horizontal : image를 좌우 반전 시킵니다.

(ii) Flip Vertical : image를 상하 반전 시킵니다.



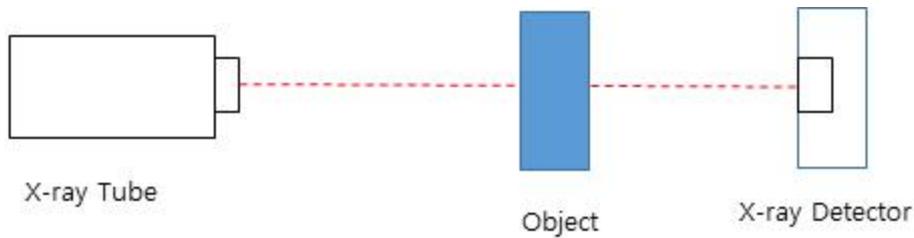
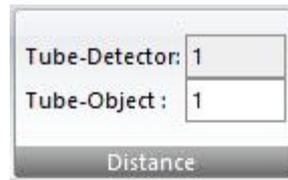
Flip horizontal



Flip Vertical



⑤ Distance



- (i) Tube - Detector : Tube와 Detector 사이의 거리
- (ii) Tube - Object : Tube와 Object 사이의 거리

$$A = (\text{Tube} - \text{Detector} \text{ 사이의 거리})$$

$$B = (\text{Tube} - \text{Object} \text{ 사이의 거리})$$

$$S = \text{Scan Speed} (= \text{Conveyor Speed})$$

$$R = \text{Real Scan Speed}$$

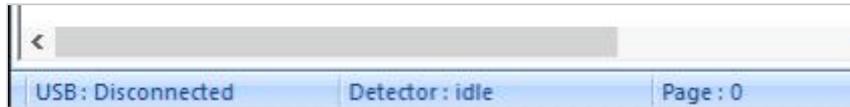
$$R = S \times \frac{A}{B}$$

Tube - Object - Detector 의 거리 비율로 인한 ScanSpeed를 보정합니다.



5. 상태창

연결 상태, 동작 상태를 확인 할 수 있는 창입니다.



USB : USB연결 상태를 Connected 혹은 Disconnected 으로 출력합니다.

Detector : Detector 의 동작 상태를 idle 혹은 running 으로 출력합니다.

Page : 현재까지 출력된 image의 프레임수를 출력합니다.



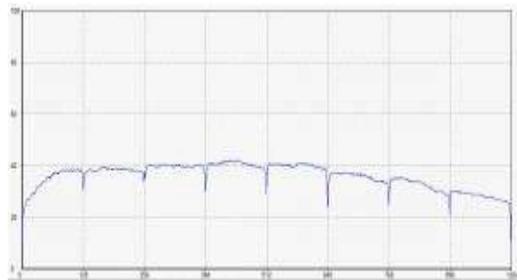
6. 캘리브레이션



Calibration은 X-ray Source를 켜진 상태에서 진행됩니다.



Scan Factors

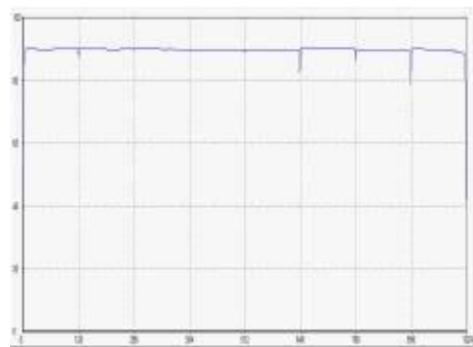


Chart

Scan Factors 및 X-ray Source의 관전압, 관전류를 조절해 Raw 동작모드의 Chart 창을 활용하여 에너지 분포를 확인합니다. 에너지 분포가 균일해야 하며 에너지는 Chart 범위의 20 ~ 80% 이내로 사용할 것을 권장합니다.



에너지가 부족함

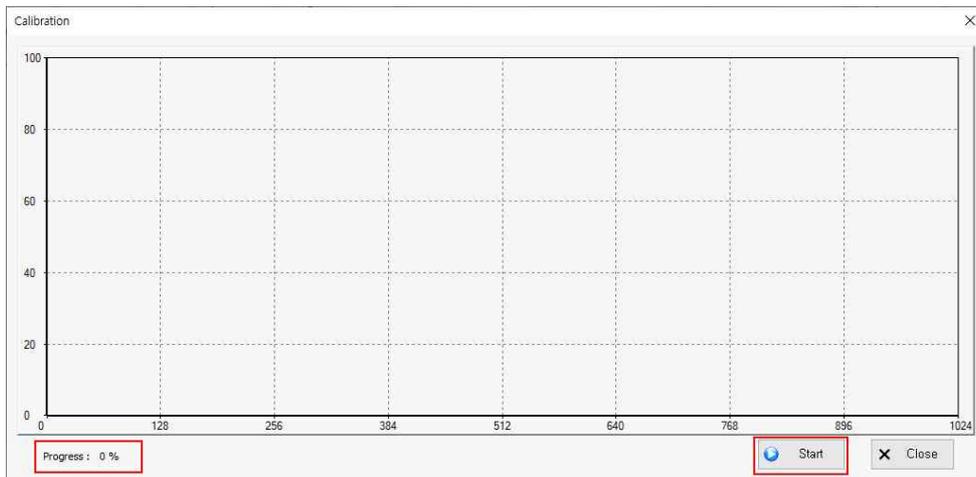


에너지가 과함

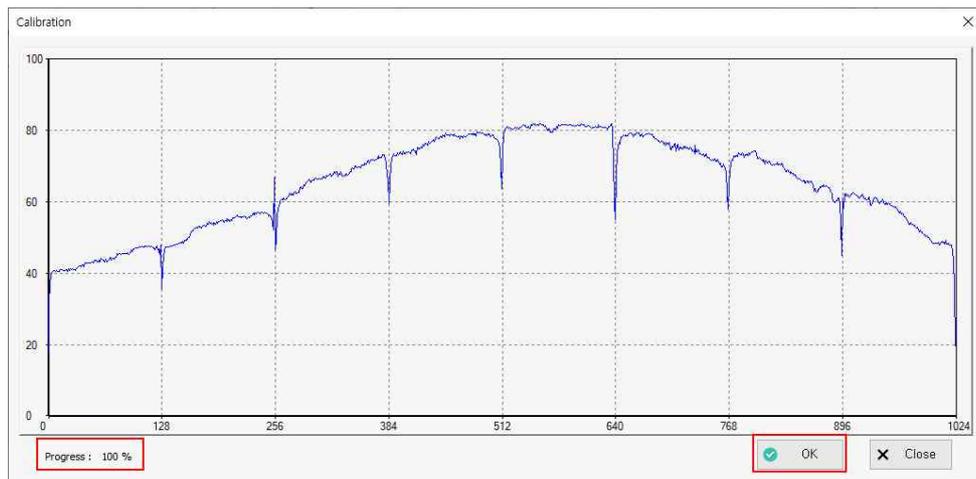
에너지가 부족할 경우 ScanSpeed를 낮추거나 관전압, 관전류를 높이고, 에너지가 과할 경우 ScanSpeed를 높이거나 관전압, 관전류를 낮춥니다.



버튼을 클릭해 Calibration 창을 띄웁니다.



Start 버튼을 클릭하여 Calibration을 진행 합니다.



실시간으로 Calibration을 차트로 확인 할 수 있으며 Progress를 통해 진행 정도를 알 수 있습니다. Calibration 완료 후 OK 버튼을 눌러 종료합니다.



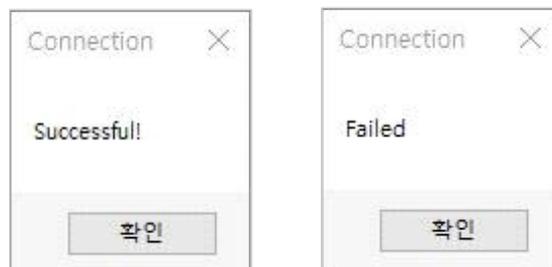
7. 촬영

- ① USB 연결
- ② Scan Factors 조절
- ③ Calibration
- ④ 촬영

7.1 USB 연결



Detector 정보에 맞게 입력 후 Connect 버튼을 클릭하여 연결 시도 합니다.



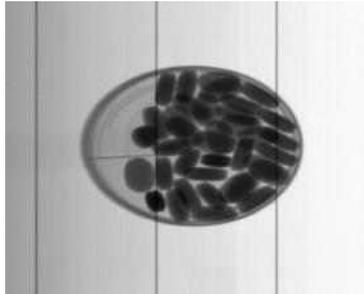
연결 성공하거나 실패시 메시지가 출력되며 하단의 상태창에 USB 연결 상태가 표시됩니다.



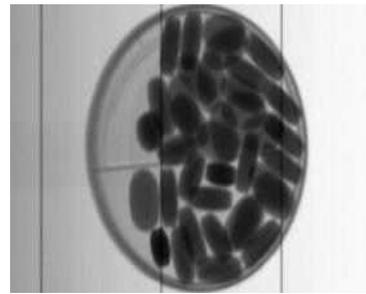
연결이 정상적으로 되어도 Connect 속성이 잘못 입력되면 비정상 image가 출력 될 수 있습니다.



7.2 Scan Factors 조절 및 에너지 조절



피사체속도 보다 느리게 찍힌 image



피사체속도 보다 빠르게 찍힌 image

Raw data로 촬영되는 영상을 확인하여 피사체의 모습이 늘어지거나 좁아지지 않도록 컨베이어 벨트의 속도를 적절히 맞춥니다.

컨베이어벨트의 속도보다 Scan Speed가 느리면 피사체가 좁게 촬영되고, 컨베이어벨트의 속도보다 Scan Speed가 빠르면 피사체가 넓게 촬영됩니다.

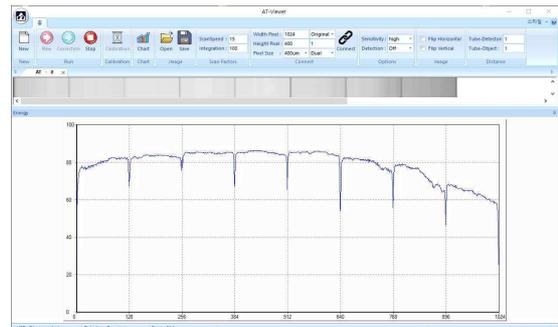
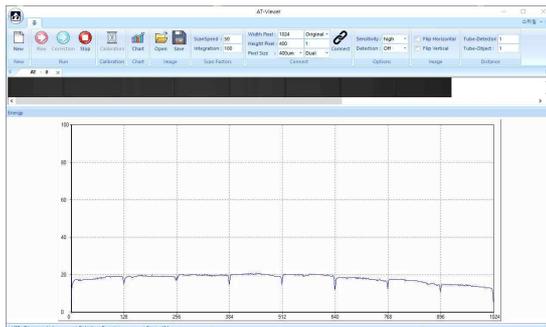
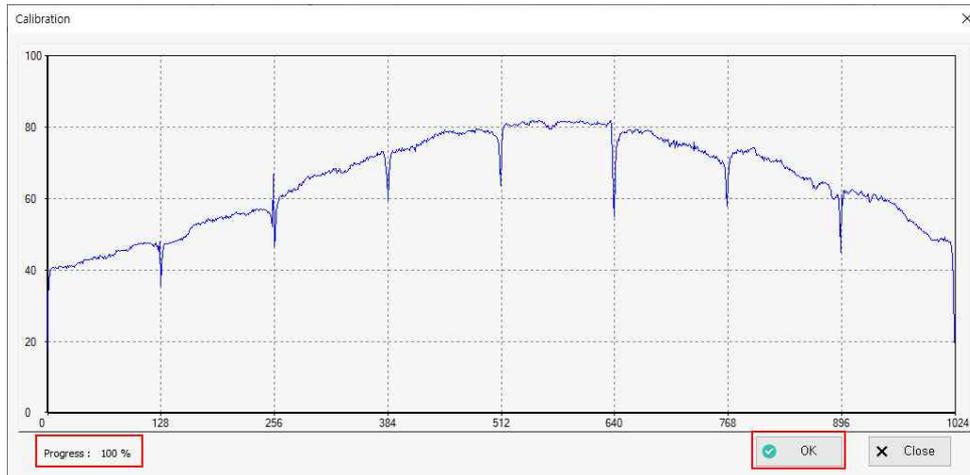


Chart 창을 활용하여 에너지 분포를 확인합니다. 에너지 분포가 균일해야 하며 에너지 범위는 20%~ 80% 이내 사용할 것을 권장합니다.



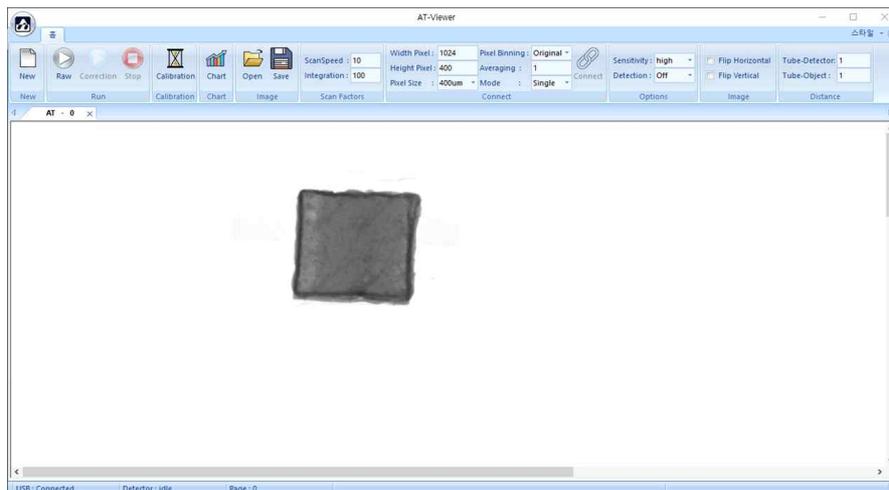
7.3 Calibration

적절한 속도와 밝기를 맞춘 후에는 컨베이어벨트를 멈추고, Detector와 X-ray tube 사이에 아무것도 없는 상태에서 Multi-Point Calibration을 시작합니다.



7.4 촬영

피사체를 촬영하여 출력되는 영상과 저장된 영상을 확인합니다.

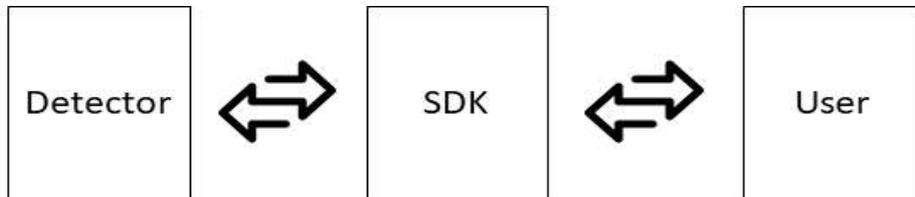




8. Software Development

8.1 소개

이 문서는 애플리케이션에서 사용할 수 있는 기능에 대한 설명을 제공합니다. 샘플 프로젝트와 USB 사용자를 위해 명령어 및 image를 제공합니다.



제공하는 SampleCode 안의 예제와 라이브러리를 통해 디텍터와 통신 할 수 있습니다. 디텍터와 통신할 수 있는 간단한 루틴을 보여주는 예제와 명령어들을 목록화 했습니다. 기능 설명에 대한 자세한 내용은 다음 목록을 참조하세요.

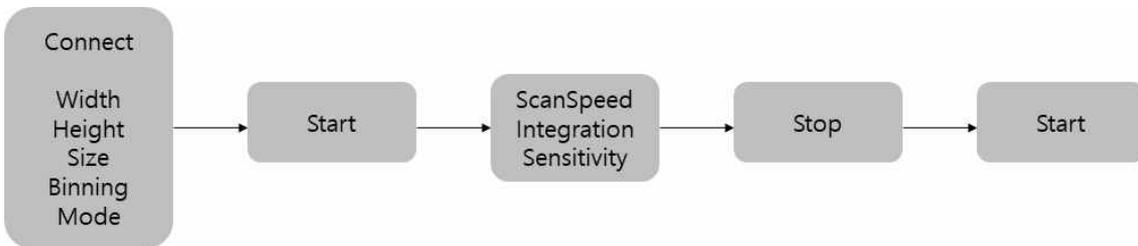
이름	수정한 날짜	유형
ConsoleDemo	2022-08-02 오전 10:57	파일 폴더
include	2022-08-02 오전 10:57	파일 폴더
Lib	2022-08-02 오전 10:57	파일 폴더
Release	2022-08-02 오전 10:57	파일 폴더
AIST_DEMO.sln	2020-09-07 오후 4:46	SLN 파일



8.2 순서도

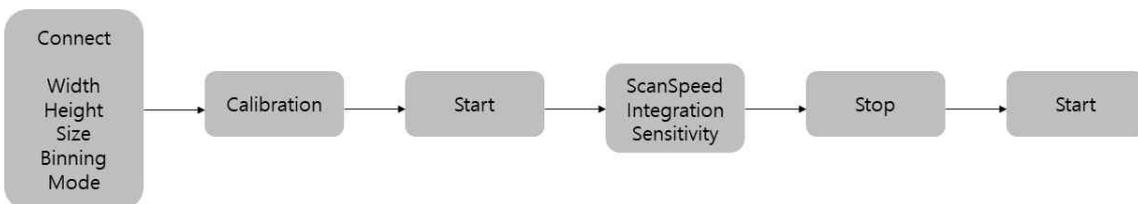
Connect 전에 Pixel에 관련된 속성을 입력합니다. 연결 후에는 ScanSpeed, Integration, Sensitivity, 등 image에 관련된 속성만 변경만 하는 것을 권장합니다.

1) Raw data



디텍터를 동작시켜 Raw image를 취득하는 과정입니다. 디텍터를 먼저 연결하고 속성 값을 변경시키고 데이터 수신과 중지를 반복하여 사용할 수 있습니다.

2) Calibration data



Calibration 실행 후 시작하면 Calibration image를 얻을 수 있습니다. 속성 값을 변경하면 Calibration을 다시 실행하는 것을 권장합니다.

Calibration은 적당한 에너지 대역에서 하는 것을 권장합니다. 우리는 20~80%의 대역을 추천합니다. 편의를 위해 Calibration data를 저장하거나 불러올 수 있습니다.



8.3 소프트웨어 컨트롤

Control	Description
ATXrayDetector	<p>디텍터 동작을 제어합니다.</p> <p>데이터를 처리하고 전달합니다.</p> <p>속성 값을 변경시키고 명령을 통한 동작 확인이 가능합니다.</p>



8.4 구동관련 명령어

8.4.1 atd_Connect()

Description :

데이터 송수신을 위해 USB채널을 통해 디텍터와 연결한다.

Return Value :

성공하면 true를 그렇지 않으면 false를 반환한다.

8.4.2 atd_Close()

Description :

연결된 채널을 닫는다.

Return Value :

성공하면 true를 그렇지 않으면 false를 반환한다.

8.4.3 atd_isOpen()

Description :

디텍터와의 연결 상태를 확인한다.

Return Value :

성공하면 true를 그렇지 않으면 false를 반환한다.

Sample Code

```
ATXrayDetector mATXrayDetector;  
  
if (mATXrayDetector.atd_Connect())  
    //printf("Connected!\n");  
  
if (mATXrayDetector.atd_isOpen())  
    //printf("Opened!\n");  
  
if (mATXrayDetector.atd_Close();)  
    //printf("Closed!\n");
```



8.4.4 atd_Start()

Description :

데이터 송수신을 위해 USB채널을 통해 디텍터와 연결한다.

Return Value :

성공하면 true를 그렇지 않으면 false를 반환한다.

8.4.5 atd_Stop()

Description :

연결된 채널을 닫는다.

Return Value :

성공하면 true를 그렇지 않으면 false를 반환한다.

8.4.6 atd_isRunning()

Description :

디텍터의 동작 상태를 확인한다.

Return Value :

성공하면 true를 그렇지 않으면 false를 반환한다.

Sample Code

```
ATXrayDetector mATXrayDetector;

if (mATXrayDetector.atd_Start())
    //printf("Start!\n");

if (mATXrayDetector.atd_isRunning())
    //printf("isRunning!\n");

if (mATXrayDetector.atd_Stop();)
    //printf("Stop!\n");
```



8.4.7 atd_Calibration()

Description :

캘리브레이션을 시작한다.

Return Value :

성공하면 true를 그렇지 않으면 false를 반환한다.

Sample Code

```
ATXrayDetector mATXrayDetector;

if (mATXrayDetector.atd_Calibration())
    //printf("Complete!\n");

// Message
Start Calibration, please Do nothing for a second Detector Start
Progress 0 %
Progress 10 %
Progress 20 %
Progress 30 %
Progress 40 %
Progress 50 %
Progress 60 %
Progress 70 %
Progress 80 %
Progress 90 %
Progress 100 % Calibration Completed!
```



8.5 속성관련 명령어

8.5.1 int atd_GetUsbIndex(), atd_SetUsbIndex(int USBIndex)

Description

연결할 USB 번호를 설정하거나 얻는다. USB 루트 포트 순으로 지정된다.

Parameter

USBIndex 기본 값은 0이다

Return Value

USBIndex을 반환한다.

8.5.2 int atd_GetWidth(), atd_SetWidth(int Width)

Description

image 넓이를 설정하거나 얻는다.

Parameter

Width 범위는 128 ~ 2048 이고 기본 값은 1024 이다.

Return Value

Width을 반환한다.

8.5.3 int atd_GetFrameHeight(), atd_SetFrameHeight(int Frame)

Description

프레임의 높이를 설정하거나 얻는다.

Parameter

Frame 기본 값은 400 이다.

Return Value

Frame을 반환한다.



8.5.4 int atd_GetPixelSize(), atd_SetPixelSize(int PixelSize)

Description

Pixel Size을 설정하거나 얻는다.

Parameter

PixelSize 기본 값은 400 (400 μ m) 이다.

Return Value

400 (400 μ m)

200 (200 μ m)

100 (100 μ m)

8.5.5 int atd_GetBinning(), atd_SetBinning(int Binning)

Description

Pixel Binning 방식을 선택한다.

Parameter

Binning 기본 값은 0 (Original) 이다.

Return Value :

0 (Original)

1 (Sum x2)

2 (Avg x2)

3 (Sum x4)

4 (Avg x4)

8.5.6 int atd_GetMode(), atd_SetMode(int Mode)

Description

디텍터의 종류를 설정하거나 얻는다.

Parameter

Mode 기본값은 0 (Single) 이다.

Return Value :

0 (Single)

1 (Dual)

2 (TDI)



Sample Code

```
ATXrayDetector mATXrayDetector;

mATXrayDetector.atd_SetUsbIndex(USBIndex);
mATXrayDetector.atd_SetWidth(Width);
mATXrayDetector.atd_SetFrameHeight(Frame);
mATXrayDetector.atd_SetPixelSize(PixelSize);
mATXrayDetector.atd_SetBinning(Binning);
mATXrayDetector.atd_SetMode(Mode)
printf("    USBIndex    : %d \n
        Width      : %d \n
        Frame      : %d \n
        PixelSize  : %d \n
        Binning    : %d \n
        Mode       : %d \n",
        atd_GetUsbIndex(),
        atd_GetWidth(),
        atd_GetFrameHeight(),
        atd_GetPixelSize(),
        atd_GetBinning(),
        atd_GetMode()
}
```



8.5.7 double atd_GetScanSpeed(), atd_SetScanSpeed(double ScanSpeed)

Description

ScanSpeed를 설정하거나 얻는다. 값에 따라 내부적으로 Integration time이 변경됩니다. 설정 변경시 IntegrationTimeRate은 100 으로 변경된다.

Parameter

ScanSpeed 기본값은 10 m/min 이다.

Return Value

ScanSpeed를 반환한다.

8.5.8 int atd_GetIntegrationTimeRate(), atd_SetIntegrationTimeRate(int Rate)

Description

Integration Time을 Percentage(%) 단위로 조절합니다.

Parameter

Rate 기본값은 100% 이다.

Return Value :

Rate를 반환한다.

8.5.9 int atd_GetSensitivity(), atd_SetSensitivity(int Sensitivity)

Description :

디텍터의 감도를 설정하거나 얻는다.

Parameter

Sensitivity 기본값은 1 (High) 이다.

Return Value :

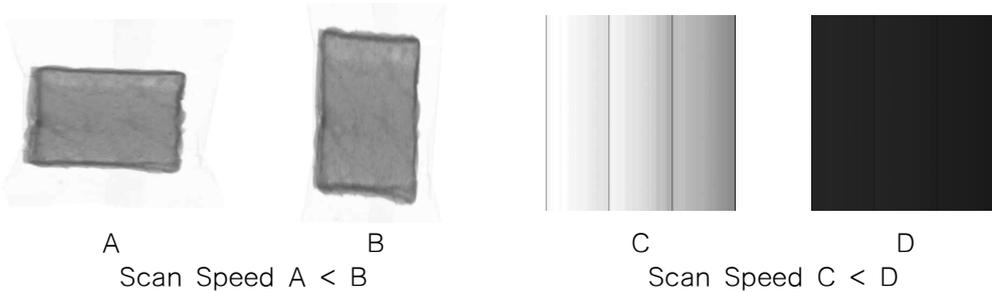
0 (Low)

1 (High)

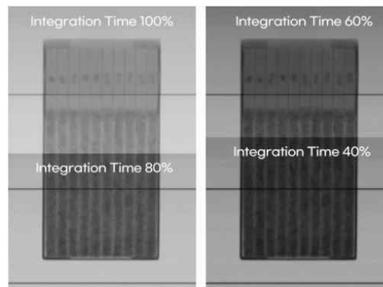


Sample Code

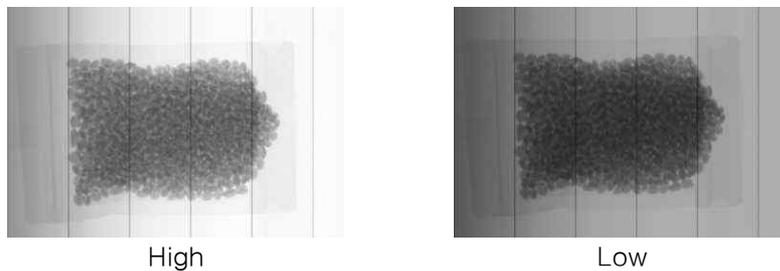
```
mATXrayDetector.atd_SetScanSpeed(ScanSpeed);  
printf(" ScanSpeed: %d \n", atd_GetScanSpeed());
```



```
mATXrayDetector.atd_SetIntegrationTimeRate(Rate);  
printf(" IntegrationTimeRate: %d \n", atd_GetIntegrationTimeRate());
```



```
mATXrayDetector.atd_SetSensitivity(Sensitivity);  
printf(" Sensitivity %d \n", atd_GetSensitivity());
```





8.5.10 int atd_GetImageFormat(), atd_SetImageFormat(int ImageFormat)

Description

image 포맷을 설정하거나 얻는다.

Parameter

ImageFormat 기본값은 0이다.

Return Value

- 0 (16bit Raw data)
- 1 (8bit Correction data)
- 2 (16bit Correction data)

Sample Code

```
ATXrayDetector mATXrayDetector;

printf("16bit Raw data\n");
mATXrayDetector.atd_SetImageFormat(0);
if (mATXrayDetector.atd_Start())
    ;

printf("8bit Correction data\n");
mATXrayDetector.atd_SetImageFormat(1);
if (mATXrayDetector.atd_Start())
    ;

printf("16bit Correction data\n");
mATXrayDetector.atd_SetImageFormat(2);
if (mATXrayDetector.atd_Start())
    ;
```



8.5.11 bool atd_GetFliphorizontal(), atd_SetFliphorizontal(bool Horizontal)

Description

좌우 반전을 설정한다.

Parameter

Horizontal 기본값은 False 이다.

Return Value

False (원본)

True (좌우반전)

8.5.12 bool atd_GetFlipvertical(), atd_SetFlipvertical(bool Vertical)

Description

상하 반전을 설정한다.

Parameter

Vertical 기본값은 False 이다.

Return Value :

False (원본)

True (상하반전)

8.5.13 int atd_GetAverage(), atd_SetAverage(int Average)

Description :

설정된 Line 수만큼 취득하여 평균값을 산출한 Line image를 표현한다.

Parameter

Average 범위는 1 ~ 50 이고 기본값은 1 이다.

Return Value :

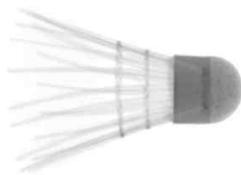
Average를 반환한다.



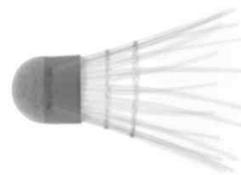
Sample Code

```
ATXrayDetector mATXrayDetector;
```

```
mATXrayDetector.atd_SetFliphorizontal(Horizontal);  
printf(" Horizontal: %d \n", atd_GetFliphorizontal());
```



False

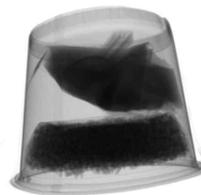


True

```
mATXrayDetector.atd_GetFlipvertical(Vertical);  
printf(" Vertical %d \n", atd_GetFlipvertical());
```

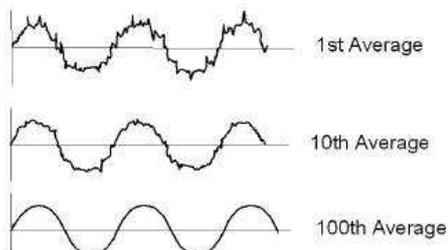


False



True

```
mATXrayDetector.atd_SetAverage(Average);  
printf(" Average %d \n", atd_GetAverage());
```





8.5.14 bool atd_GetObjectDetection(), atd_SetObjectDetection(bool Detection)

Description

피사체 감지 사용을 설정하거나 얻는다.

Parameter

Detection 기본값은 false 이다

Return Value

False (Off)

True (On)

8.5.15 int atd_GetObjectDetectionThreshold(), atd_SetObjectDetectionThreshold(int Threshold)

Description

피사체 감지시 임계값을 설정하거나 얻는다.

배경 대비 감지하려는 피사체의 Gray 임계값을 조정한다.

Parameter

Threshold 기본값은 30 이다. 범위는 10~100 이다.

Return Value :

Threshold 을 반환한다.

8.5.16 int atd_GetLinetoAvg(), atd_SetLinetoAvg(int LinetoAverage)

Description

Calibration 단계별로 설정된 Line 수만큼 취득된 데이터의 평균값을 산출하여 Calibration 데이터를 취득한다.

Parameter

LinetoAverage 범위는 10 ~ 100이고 기본값은 30이다.

Return Value

LinetoAverage



8.5.17 bool atd_SaveCalibrationData(const char FileName[])
bool atd_LoadCalibrationData(const char FileName[])

Description

Calibration data를 저장하거나 불러온다.

Parameter

FileName[] 저장하거나 불러올 파일이름을 설정한다.

Return Value :

성공하면 true를 그렇지 않으면 false를 반환한다.

8.5.18 bool atd_GetAutoCalibration(), atd_SetAutoCalibration(bool Auto)

Description

설정된 Source의 관전압, 관전류에서 Integration Time Rate를 점차적으로 낮춰가며 에너지가 포화되지 않는 영역에서 정상적으로 Calibration 동작을 실행한다.

Parameter

Auto 기본값은 False 이다.

Return Value :

False (자동 캘리브레이션 미실행)

True (자동 캘리브레이션 실행)

8.5.19 double atd_GetTotalLength(), atd_SetTotalLength(double TotalLength)

Description

튜브-디텍터 거리를 설정하거나 얻는다.

길이 정보를 통한 거리 비율로 ScanSpeed를 보정합니다.

Parameter

TotalLength 기본값은 1이다. ObjectLength 보다 작을 수 없다.

Return Value

TotalLength



8.5.20 double atd_GetObjectLength(), atd_SetObjectLength(double ObjectLength)

Description

튜브-피사체 거리를 설정하거나 얻는다.
길이정보를 통한 거리 비율로 ScanSpeed를 보정합니다.

Parameter

ObjectLength 기본값은 1이다. TotalLength 보다 클 수 없다.

Return Value

ObjectLength

8.5.21 double atd_GetTempCorrection(), atd_SetTempCorrection (bool Correction)

Description

온도 변화에 따른 값을 보정한다.

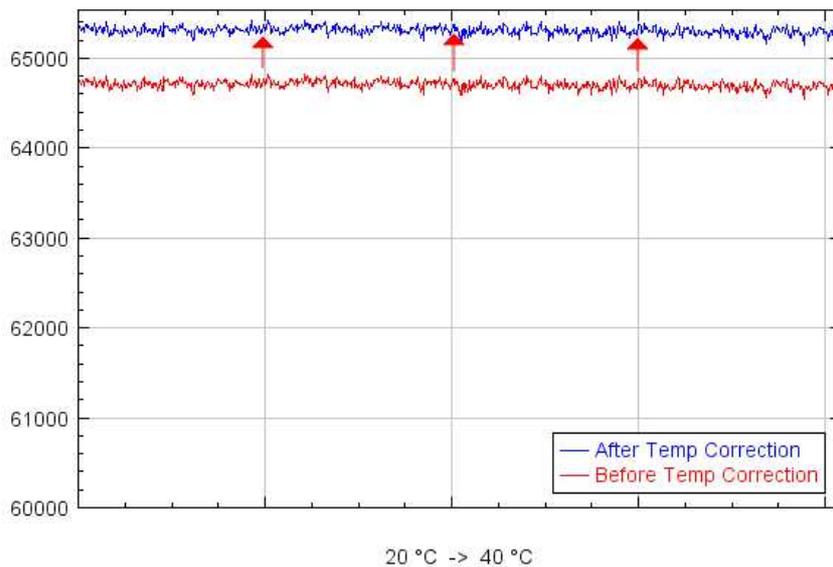
Parameter

Correction 기본값은 True 이다.

Return Value

False (온도 보정 미실행)

True (온도 보정 실행)



온도 변화로 발생한 변동된 값을 Temp Correction을 이용해 보정한다.



8.6 출력관련 명령어

8.6.1 `bool` `atd_HWND(HWND INhMDC)`

Description

콘솔 창 핸들 값을 설정한다.

Parameter

HWND image를 출력한 콘솔 창의 핸들값 전달

Return Value

성공하면 `true`를 그렇지 않으면 `false`를 반환한다.

8.6.2 `atd_SetCallBack(void(*CallBackFrame()),CallBackEvent(int code, const char* event))`

Description

이벤트시 메시지와 프레임 획득시 함수를 콜백 함수로 설정한다.

Parameter

CallBackFrame 프레임 획득시 함수

CallBackEvent 이벤트 메시지 및 오류코드 반환 함수

8.6.3 `unsigned short*` `atd_Getadress()`

Description

이벤트시 메시지와 프레임 획득시 함수를 콜백 함수로 설정한다.

Parameter

CallBackFrame 프레임 획득시 함수 CallBackEvent 이벤트 메시지 함수

Return Value

`unsigned short*` 획득한 프레임 주소값



8.6.4 atd_Draw(unsigned short* buf)

Description

콜백 함수 얻은 버퍼 프레임을 출력한다.

Parameter

Pixel Width x Frame 크기의 영상

Sample Code

```
//이벤트 시 메시지 함수
void CallBackEvent(int code, const char* event) {
    printf("%d, %s\n", code, event);
}

//프레임 획득 시 함수 void CallBackFrame() {
    memcpy(g_pBuf, (unsigned short*)atd_Getaddress(),sizeof(unsigned short)* g_Size);
}

//콜백함수설정
mATXrayDetector.atd_SetCallback(&CallBackFrame,&CallBackEvent);
```



9. 에러 코드

코드	설명	해결 방법
-1 = EC_FAILED	실패	
0 = EC_SUCCESS	성공	
1 = EC_USB_DEVICE_FAILED	USB 장치 찾기 실패	장치와 USB 상태를 확인하거나, USB 드라이버를 설치하세요.
2 = EC_USB_CONNECT_FAILED	USB 연결 실패	장치와 USB 상태를 확인하세요.
3 = EC_USB_SEND_FAILED	USB send 명령 실패	장치와 USB 상태를 확인하세요.
4 = EC_USB_REV_FAILED	USB receive 명령 실패	장치와 USB 상태를 확인하세요.
5 = EC_CALIBAION_SATURATION	X선 포화상태	X선이 포화상태입니다. 관전압, 관전류를 확인하세요.
6 = EC_CALIBAION_SHORTAGE	X선 부족	X선이 부족상태입니다. 관전압, 관전류를 확인하세요.
7 = EC_CALIBAION_FAILED	Calibration 실패	Calibration 도중 실패 했습니다. 통신 상태를 확인 하거나 다시 시도 하세요
8 = EC_SET_WIDTH_DISMATCH	Width 설정 값이 불일치	장치와 사용자 입력 값이 불일치 합니다. 다시 확인 하세요.

AiST Tech Inc.

tel. 063-852-5400

<http://aisttech.co.kr>

(주)아이스트