SDK User Manual

Table of Contents

6. Reference S/W Description

1. Introduction

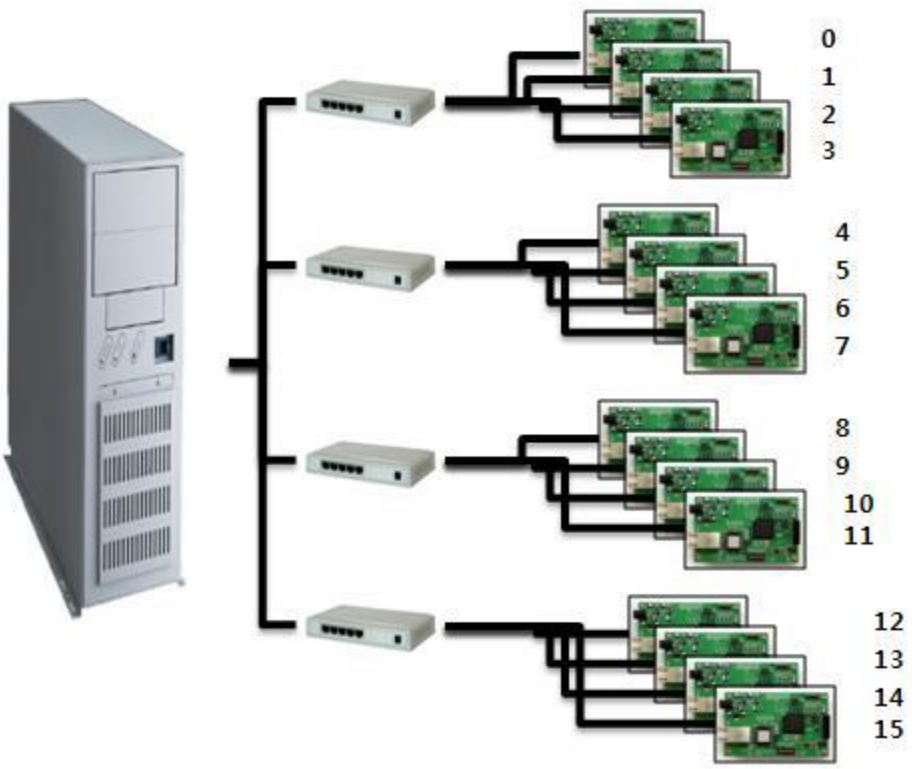
This document is the SDK User Guide for Bontech Detector. It consists of Network configuration for API, API Usage, and Reference S/W explanations.

2. SDK Overview

Provided APIs are divided into two kinds. The first one is the ImageCaptureDDL which connects with Sensor in order to acquire Images, and the second one is CalibrationDLL for calibration of the acquired Images.

2.1. System Configuration

Detector can be directly connected to PC or indirectly connected through Switch Hub to be used. In connection, Gagabit Ethernet technology is utilized, and it can transfer up to 1 Gbps using a large amount Image Data theoretically.



The picture above is an example of System Configuration. Maximum 4 Sensors can be connected to 1 PC.

2.2. System Requirement

- OS : 32bit or 64bit Windows operating systems
- Processor : Dual Core 2.5GHz or faster
- RAM : 1GB or 2GB
- NIC : NIC which support Jumbo Packet

2.3. API Configuration

Header / Library	ImageCapDllEx.h ImageCaptureDLL.lib ImageCaptureDLL.dll GCDLL.dll	– Interface with Sensor – Image calibration
	CalibrationDllEx.h CalibrationDLL.lib CalibrationDLL.dll	
Data File	ImageCapture.ini	–Sensor initialization information file
	Ref1.dat	
	Ref2.dat	
Calibration reference file	[Mode]_Dark.raw	– Calibration file by each Mode – 2 types of modes: Trigger mode & AED mode
	[Mode]_DefectMap.raw	
	[Mode]_Reference00.raw	

- The function for Detector Interface such as Connection with Sensor, Shooting for Images, and etc. are defined in ImageCapDllEx.h.

  When connecting with Detector, you need to set the path of Data file.

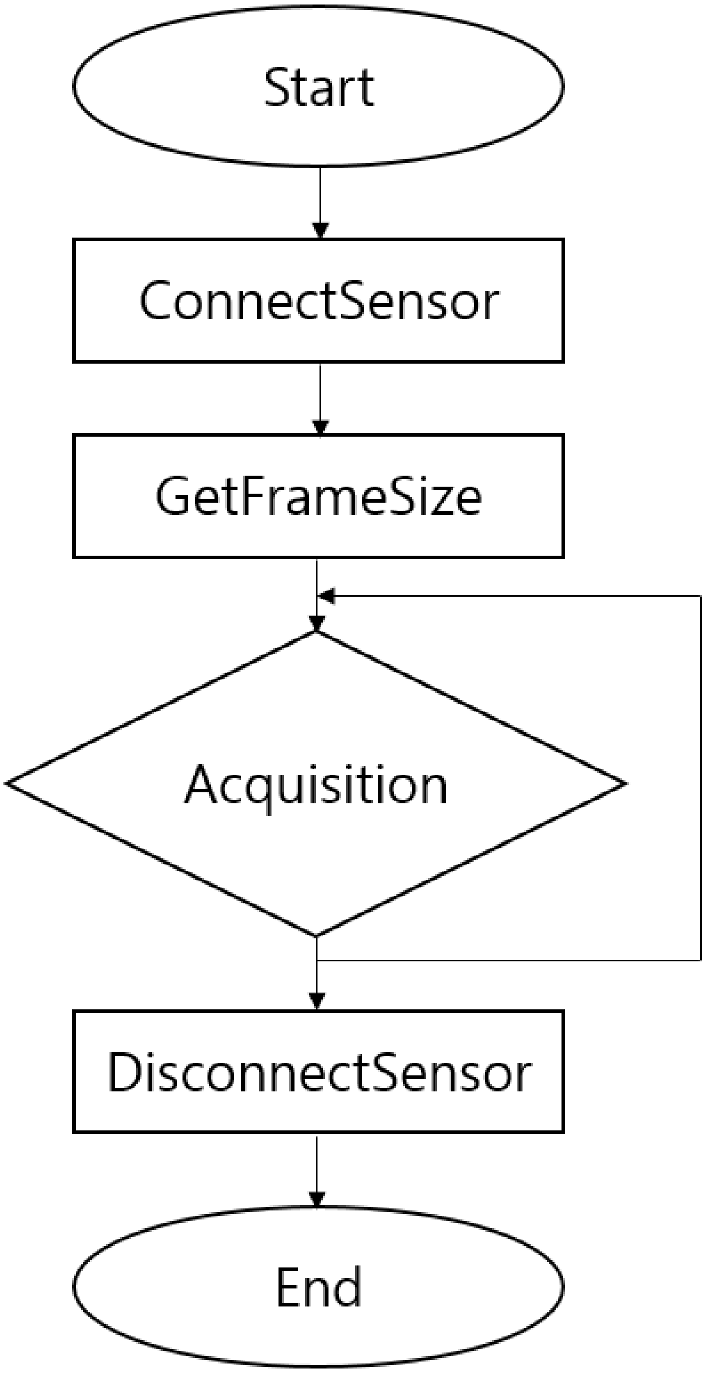
- The function for calibrating the X–ray image is defined in CalibrationDllEx.h.

  When acquiring the Image, you need to set the path for Calibration reference file.

  Calibration reference file can be acquired by using the RawImageViewer program.

- In the Sample S/W, Ref1.dat, Ref2.dat files, and Calibration reference file are not provided.

2.4. Data Acquisition Flow chart



It is a procedure to acquire X-ray images. It is possible to acquire the image if the function is called in order as the flow chart above.

The other functions can be called if necessary.

2.5. Acquisition method

The detector has three methods of image acquisition.

- AED Mode, Soft Trigger Mode, Average Mode, Continuous Mode(Dynamic)

The function to change the mode is ImageCapModeChange.

The image acquisition function is ImageCapImageAcquistion.

The image acquisition function of the Continuous Mode is ImageCapStartCapture.

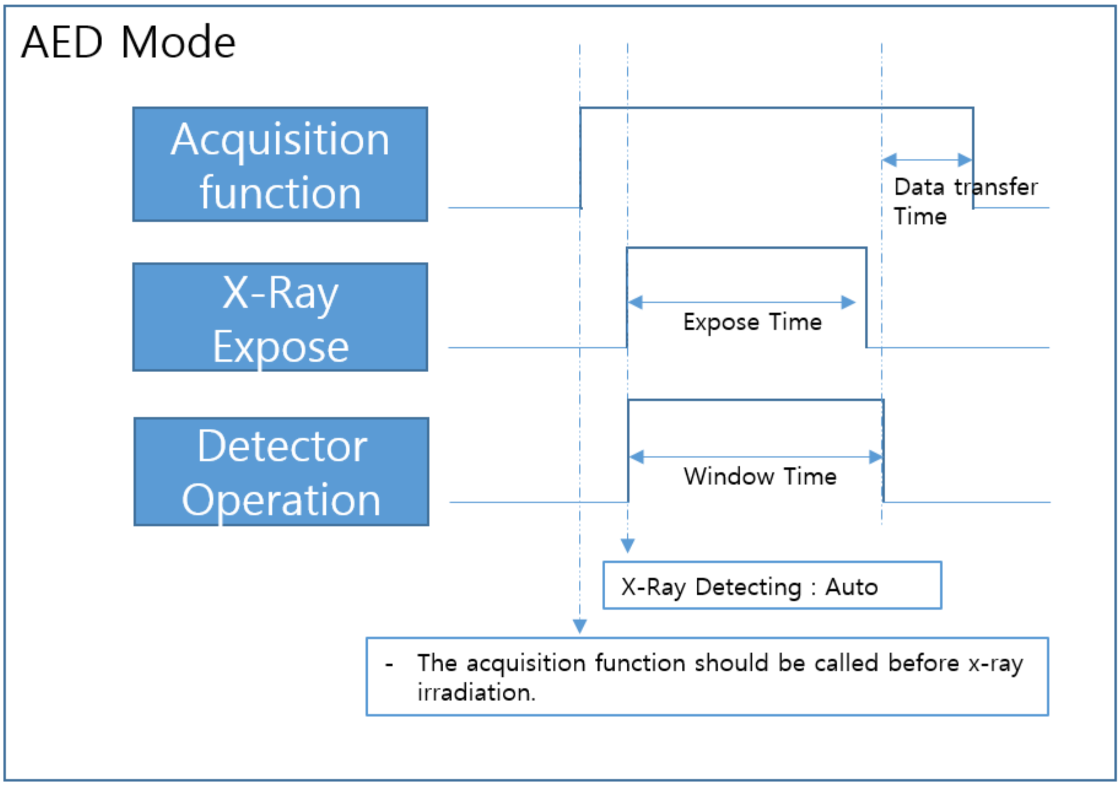
The following explains when to call ImageCapImageAcquistion and the parameters of the function.

2.6. AED Mode

A method of acquiring images by automatically detecting X−rays.

- Mode change method: ImageCapModeChange (IMAGECAP_AED_MODE);

- Acquisition function: ImageCapImageAcquistion (BRIGHT_MODE);



2.7. Soft Trigger Mode

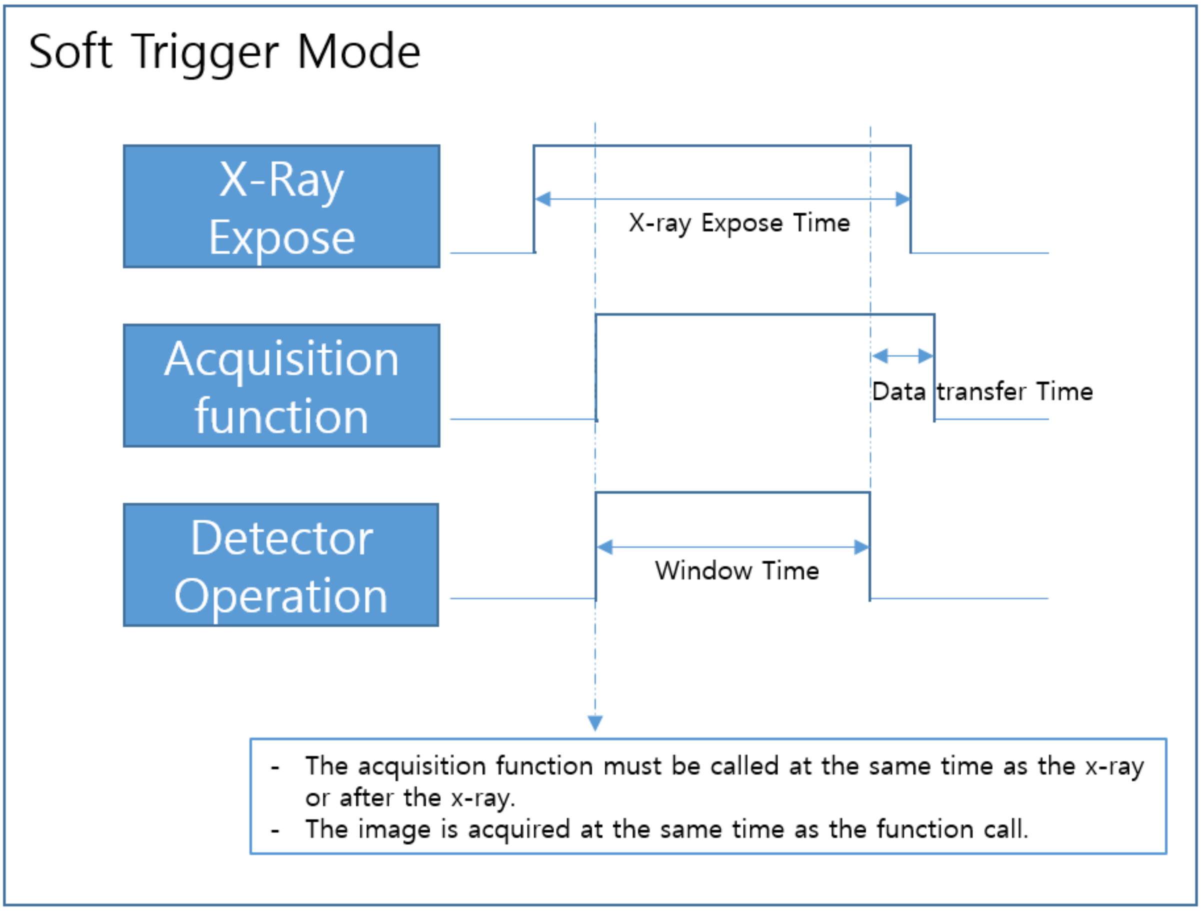
This is a method of acquiring images from the time of soft trigger occurrence.

- Mode change method:

  ImageCapModeChange (IMAGECAP_SOFTTRIGGER_MODE);

- Acquisition function:

  ImageCapImageAcquistion (SOFTTRIGGER_INSTANT_MODE);



- The acquisition function must be called at the same time as the x-ray or after the x-ray.
- The image is acquired at the same time as the function call.

2.8. Average Mode

Average mode is a mode in which the average image is returned after acquiring the set number of frames.

- Mode change method:

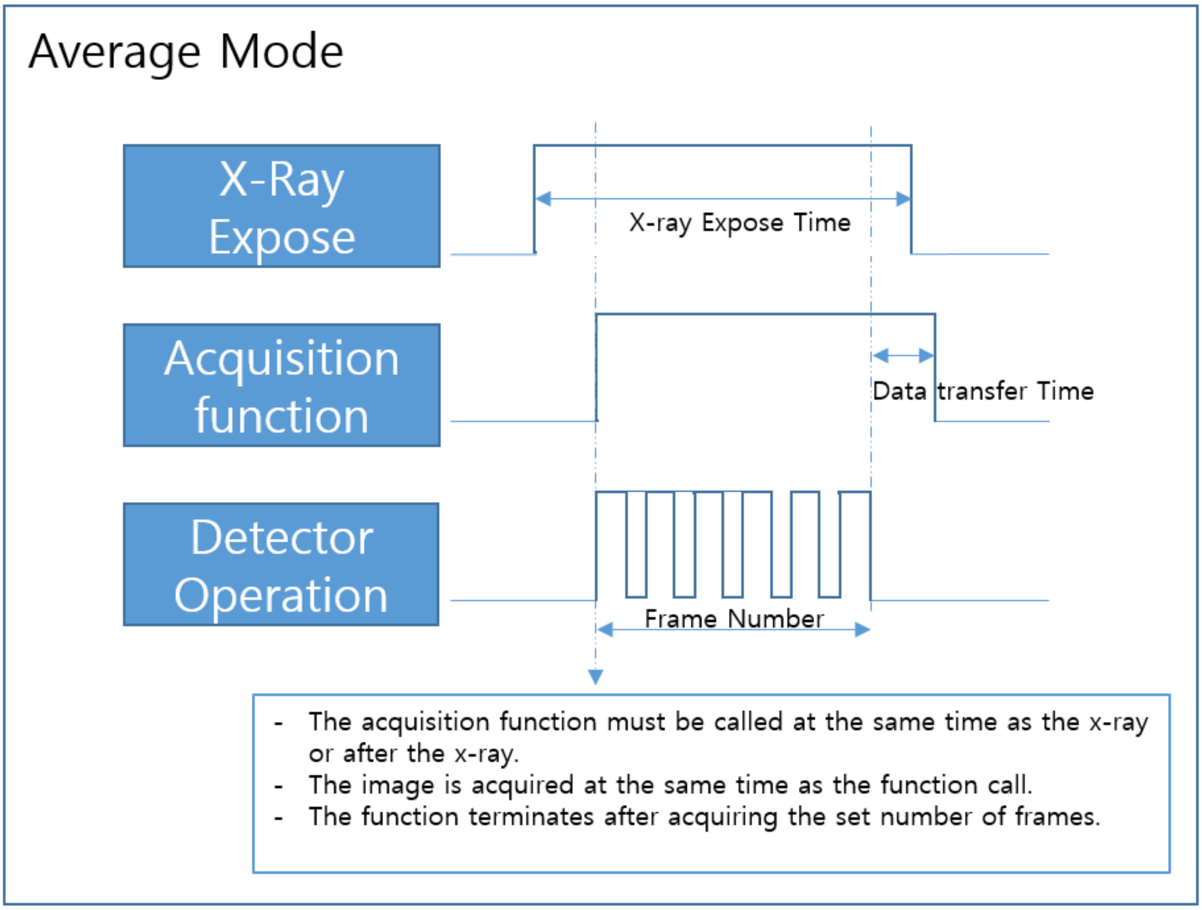
  ImageCapModeChange (IMAGECAP_SOFTTRIGGER_MODE);

- Acquisition function:

  ImageCapImageAcquistion (SOFTTRIGGER_INSTANT_MODE);

- Frame number setting method:

  ImageCapSetAVGModeFrameNum(int nFrameNum);



- The acquisition function must be called at the same time as the x-ray or after the x-ray.
- The image is acquired at the same time as the function call.
- The function terminates after acquiring the set number of frames.

2.9. Continuous Mode (4FPS)

The continuous mode starts by a acquisition command and continuously acquires an image until a stop command is received.

- Mode change method:

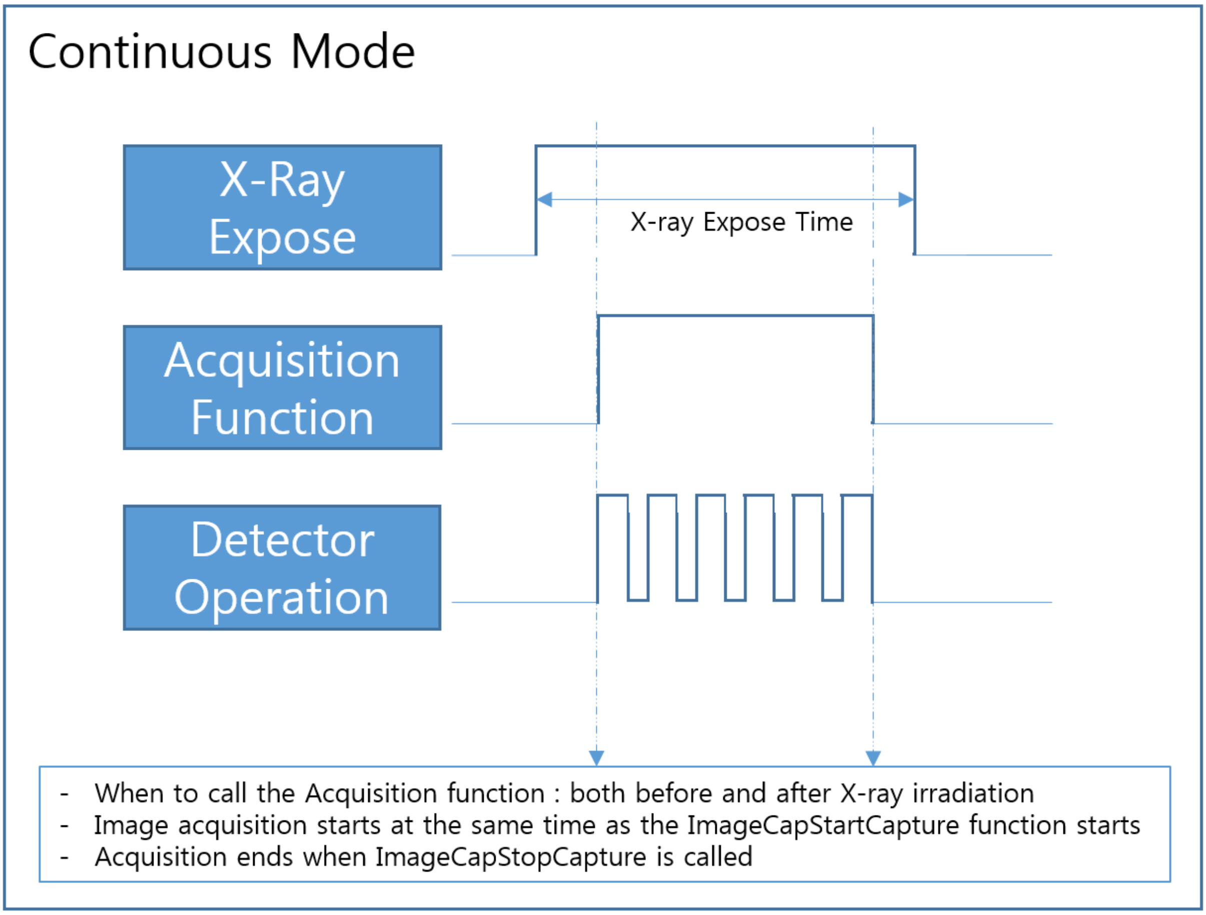
  ImageCapModeChange (IMAGECAP_CONTINUOUS_MODE);

- Acquisition function:

  ImageCapCommandSend(DARK_MODE);

  ImageCapStartCapture(DARK_MODE);

  ImageCapStopCapture();



Continuous Mode

- When to call the Acquisition function : both before and after X-ray irradiation
- Image acquisition starts at the same time as the ImageCapStartCapture function starts
- Acquisition ends when ImageCapStopCapture is called

2.10. Continuous - Binning Mode (8FPS)

The continuous−binning mode starts by a acquisition command and continuously acquires an binning−image until a stop command is received.

- Mode change method:

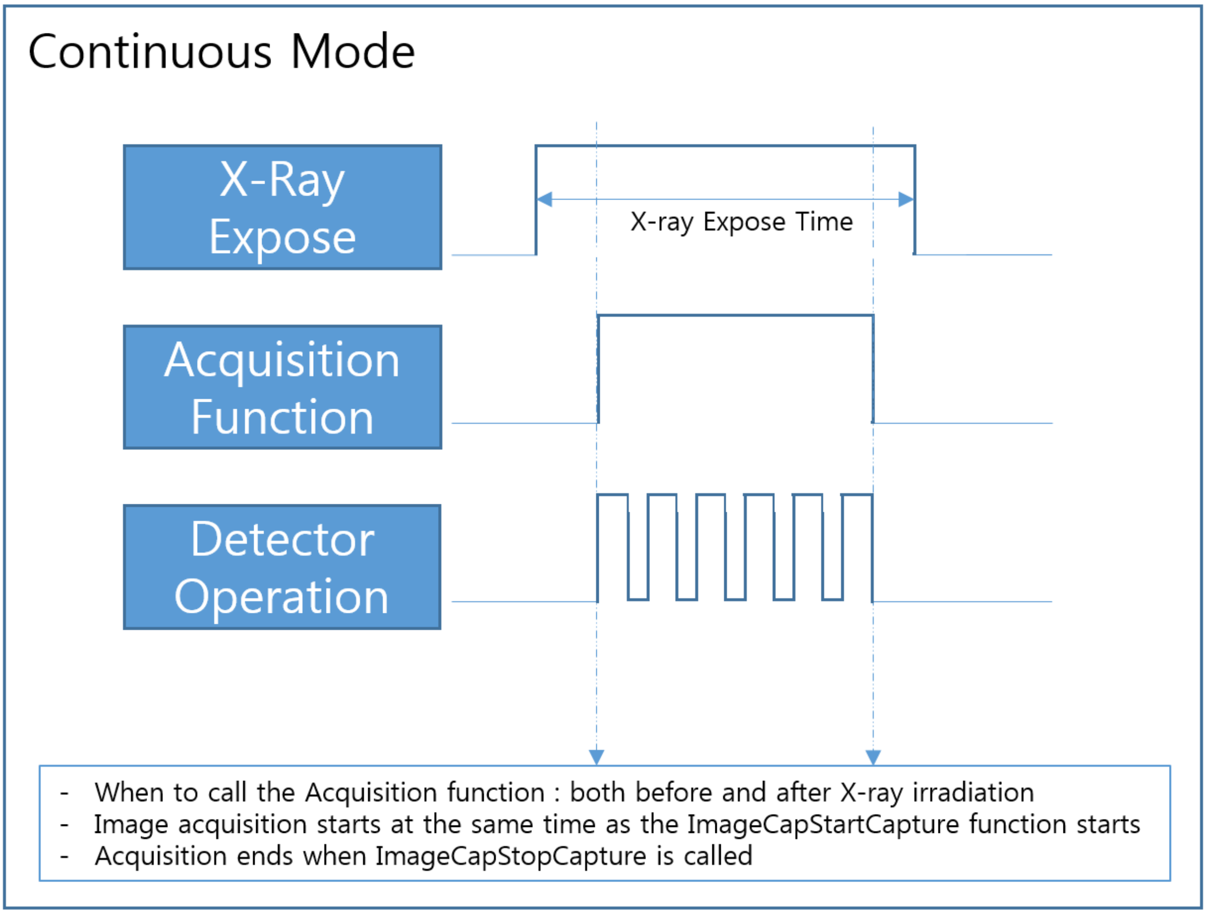
  ImageCapModeChange (IMAGECAP_CONTINUOUS_BINNING_MODE);

- Acquisition function:

  ImageCapCommandSend(DARK_MODE);

  ImageCapStartCapture(DARK_MODE);

  ImageCapStopCapture();



- When to call the Acquisition function : both before and after X-ray irradiation
- Image acquisition starts at the same time as the ImageCapStartCapture function starts
- Acquisition ends when ImageCapStopCapture is called

3. API Main Function Description

3.1. ImageCapConnectSensor

Header ImageCAPDllEx.h

Syntax

```
int     ImageCapConnectSensor (
            unsigned int     nHostIP,
            unsigned int     nSensorIP
            LPCTSTR          lpszDataPath
);
```

Parameters

nHostIP	Host IP address
nSensorIP	Sensor IP address
lpszDataPath	Sensor Configuration File Path

Return value

IMAGECAP_OK	Success
Etc.	Error Code

Description

By passing the PC IP address and Sensor IP, try to connect to Sensor.

In *lpszDataPath,* ImageCapture.ini, and Ref1.dat, and Ref2.dat files should exist. When the connection is successful, the initialization is performed in *ImagecapConnectsensor,* therefore, image acquisition is possible through ImageCapImageAcquisition() function without calling or configuring for further functions. It does not need to call for ImageCapGetHost() and ImageCapSearchSensor() functions.

It is designed to acquire Image by connecting with Sensor through this function in the Smaple S/W

3.2. ImageCapGetFrameSize

Header ImageCAPDllEx.h

Syntax

```
int     ImageCapGetFrameSize (
            unsigned int        nSensorIP,
            unsigned short*     nWidth
            unsigned short*     nHeight
);
```

Parameters

nSensorIP	Sensor IP
nWidth	Frame width
nHeight	Frame height

Return value

IMAGECAP_OK	Success
Etc.	Error Code

Description

Obtain the horizontal and vertical size of the frame.

It is 3072x3072 for 4343 sensor, and it is 2500x3052 for 3543 sensor.

3.3. ImageCapImageAcquisition

Header ImageCAPDllEx.h

Syntax

```
int    ImageCapImageAcqusition (
           unsigned int        nSensorIP,
           unsigned short*     pImage,
           unsigned int        nMode,
           LPCTSTR             lpszRefPath
);
```

Parameters

nSensorIP	Sensor IP
pImage	Data buffer to receive Image data
nMode	DARK_MODE, BRIGHT_MODE, SOFTTRIGGER_INSTANT_MODE
lpszRefPath	Path of Calibration reference file

Return value

IMAGECAP_NEW_FRAME	Image data is received.
IMAGECAP_RECEIVING	Image data is not received.
IMAGECAP_CANCEL	Canceled by a user command
IMAGECAP_COMMAND_RECV_ERROR	Data reception error
IMAGECAP_RECV_TIMEOUT_ERROR	ACK is not transmitted in a certain time.
IMAGECAP_NOTREADY_RO	Not ready to receive Image
Etc.	Error Code

Description

It receives the Image data relevant to ***nMode*** into ***pImage***. ***pImage*** should be called allocating the size of image data. Since Image size is 16bit, 3072 by 3072, it becomes 18874368byte(3072*3072*(16/8)).

Depending on lpszRefPath configuration, it may obtain calibrated images or non-calibrated images.

nMode	DARK_MODE	Acquires a dark image
	BRIGHT_MODE	Acquires bright images. When X-ray Expose signal is input, it receives X-ray and then image data. At this time, an infinite waiting for Exposure signal can be cancelled by calling ImageCapAcqusitionCancel() function. Used in AED and Average mode
	SOFTTRIGGER_ INSTANT_MODE	Used in Soft Trigger Mode. The image is acquired immediately.
lpszRefPath	NULL	Acquires un-calibrated images
	Calibration path	It acquires calibrated images with reference to file of the inputted path In the Calibration path, the reference file should be prepared in advance.

Calibration reference files : It is possible to acquire through RawImageViewer program.

- Trigger_Dark.raw, Trigger_DefectMap.raw, Trigger_Referenc00.raw

- AED_Dark.raw, AED_DefectMap.raw, AED_Reference00.raw

3.4. ImageCapCommandSend

Header ImageCAPDllEx.h

Syntax

```
int    ImageCapCommandSend (
            unsigned int        nSensorIP,
            int                 nCmd,
);
```

Parameters

nSensorIP	Sensor IP
nCmd	DARK_MODE

Return value

IMAGECAP_OK	Success
IMAGECAP_NOTREADY_RO	Not ready to receive Image
Etc.	Socket Error Code

Description

Transmit command for acquiring images in continuous mode.

When the command is sent, the detector starts to acquire the image.

3.5. ImageCapStartCapture

Header ImageCAPDllEx.h

Syntax

```
int     ImageCapImageStartCapture (
            unsigned int        nSensorIP,
            unsigned short**    pRefImage,
            unsigned int        nMode,
            LPCTSTR             lpszRefPath
);
```

Parameters

nSensorIP	Sensor IP
pRefImage	Start address of image buffer.
nMode	DARK_MODE
lpszRefPath	Path of Calibration reference file

Return value

IMAGECAP_NEW_FRAME	Image data is received.
IMAGECAP_RECEIVING	Image data is not received.
IMAGECAP_CANCEL	Canceled by a user command
IMAGECAP_COMMAND_RECV_ERROR	Data reception error
IMAGECAP_RECV_TIMEOUT_ERROR	ACK is not transmitted in a certain time.
IMAGECAP_NOTREADY_RO	Not ready to receive Image
Etc.	Error Code

Description

This function should be called after ImageCapCommandSend.

10 buffers are used internally, and pRefImage receives the starting address of the image being created.

You can acquire a continuous image through the loop statement.

After calling this function, the necessary tasks must be implemented in a thread to maintain the maximum FPS.

It must end the loop statement and call ImageCapStopCapture to end the image acquisition

3.6. ImageCapStopCapture

Header ImageCAPDllEx.h

Syntax

```
int     ImageCapStopCapture (
            unsigned int        nSensorIP,
);
```

Parameters

nSensorIP	Sensor IP

Return value

IMAGECAP_OK	Success

Description

Image acquisition is stopped in continuous mode.

3.7. ImageCapAcqusitionCancel

Header ImageCAPDllEx.h

Syntax

```
int    ImageCapAcqusitionCancel (
           unsigned int      nSensorIP
);
```

Parameters

nSensorIP	Sensor IP

Return value

IMAGECAP_OK	Success

Description

After calling ImageCapImageAcqusition function, cancel the waiting status to receive ready signal. When it is normaly canceled, the ImageCapImageAcqusition function will return *IMAGECAP_CANCEL*.

3.8. ImageCapDisConnectSensor

Header ImageCAPDllEx.h

Syntax

```
int     ImageCapDisConnectSensor (
            unsigned int            nSensorIP
);
```

Parameters

nSensorIP	Sensor IP

Return value

IMAGECAP_OK	Success
IMAGECAP_STOP_WAIT	Reception thread is not completely finished.
Etc.	Error Code

Description

Close buffer and reception thread allocated in ImageCapConnectSensor() function.

Use only when detector is in single image acquisition mode.

Not used in Dynamic mode.

4. API Sub Function Description

4.1. ImageCapEnableLog

Header ImageCAPDllEx.h

Syntax

```
int    ImageCapEnableLog (
           BOOL           nEnable
);
```

Parameters

bEnable	Saving.	TRUE: log is saved FALSE: log is not saved

Return value

IMAGECAP_OK	Success

Description

Set whether to save Log. If you want to save the Log, set nEnable as TRUE. Log files are created according to created dates inside [./log] folder, and as API function is called, the name of function and the return value are printed on log.

4.2. ImageCapGetSAEDSensitivity

Header ImageCAPDllEx.h

Syntax

```
int    ImageCapGetSAEDSensitivity (
           unsigned int          nSensorIP
           int*                  nSens
);
```

Parameters

nSensorIP	Sensor IP
nSens	Receive the current AED sensitivity value LOW_SENSITIVITY MID_SENSITIVITY HIGH_SENSITIVITY

Return value

IMAGECAP_OK	Success
Etc.	Error Code

Description

Receive the currently set AED sensitivity value.

Sensitivity is divided into 3 stages, and the default value is MID_SENSITIVITY.

(LOW_SENSITIVITY, MID_SENSITIVITY, HIGH_SENSITIVITY)

4.3. ImageCapSetSAEDSensitivity

Header ImageCAPDllEx.h

Syntax

```
int     ImageCapSetSAEDSensitivity (
            unsigned int            nSensorIP
            int                     nSens
);
```

Parameters

nSensorIP	Sensor IP
nSens	Sensitivity settings LOW_SENSITIVITY MID_SENSITIVITY HIGH_SENSITIVITY

Return value

IMAGECAP_OK	Success
Etc.	Error Code

Description

Change AED Sensitivity of the sensor. This setting may be one of three values.

LOW_SENSITIVITY, MID_SENSITIVITY, HIGH_SENSITIVITY

4.4. ImageCapModeChange

Header　　ImageCAPDllEx.h

Syntax

```
int     ImageCapModeChange (
            unsigned int      nSensorIP,
            int                nMode
);
```

Parameters

nSensorIP	Sensor IP
nMode	IMAGECAP_TRIGGER_MODE IMAGECAP_AED_MODE IMAGECAP_SOFTTRIGGER_MODE IMAGECAP_AVG_COUNT_MODE IMAGECAP_AVG_AED_MODE IMAGECAP_CONTINUOUS_MODE

Return value

IMAGECAP_OK	Success
IMAGECAP_CANCEL	nMode setting error
Etc.	Error Code

Description

Change the Sensor mode relevant to ***nSensorIP***.

4.5. ImageCapGetCurrentMode

Header ImageCAPDllEx.h

Syntax

```
int    ImageCapGetCurrentMode (
           unsigned int      nSensorIP,
           int*               nMode
);
```

Parameters

nSensorIP	Sensor IP
nMode	Pointer to receive Mode value

Return value

IMAGECAP_OK	Success
Etc.	Error Code

Description

It receives the current image acquisition method into ***nMode***.

nMode – IMAGECAP_TRIGGER_MODE

IMAGECAP_AED_MODE

If it’s not a Success, receive –1 from the nMode.

4.6. ImageCapGetHostIP

Header ImageCAPDllEx.h

Syntax

```
int     ImageCapGetHostIP (
            unsigned int*      pHostIP,
            unsigned int*      pNumberofHost
);
```

Parameters

pHostIP	Pointer of Host IP List
pNumberofHost	Pointer of number of Network Card

Return value

IMAGECAP_OK	Success
IMAGECAP_GET_HOST_NAME_ERROR	Get Host Name Error
IMAGECAP_NIC_CHECK	No Network cards
Etc.	Error Code

Description

It transmit Network IP List and the number of Network cards of the PC to User Application.

pHostIP has to capture memory as much as the number of Network cards.

4.7. ImageCapSearchSensor

Header ImageCAPDllEx.h

Syntax

```
int    ImageCapSearchSensor (
            unsigned int       nHostIP,
            unsigned int*      pSensorIP,
            unsigned int*      pNumberofSensor
);
```

Parameters

nHostIP	IP of Network Card which will check connection
pSensorIP	Pointer of Sensor IP connected to nHostIP
pNumberofSensor	Pointer of number of Sensors connected to nHostIP

Return value

IMAGECAP_OK	Success
Etc.	Error Code

Description

It checks whether Network Card for ***nHostIP*** is connected to Sensor, and when it is connected, it brings IP Address List that is used for communication.

If there are more than one Sensors to Network Card, you can choose which device to use in the User Application.

4.8. ImageCapInitSensor

Header ImageCAPDllEx.h

Syntax

```
int     ImageCapInitSensor (
            unsigned int        nSensorIP,
            LPCTSTR             lpszDataPath
);
```

Parameters

nSensorIP	Sensor IP
lpszDataPath	Sensor Configuration File Path

Return value

IMAGECAP_OK	Success
Etc.	Error Code

Description

Refer to the data file in the path of *lpszDataPath,* and initialize the Sensor relevant for ***nSensorIP***. The major values are Sensor signal control, M Clock control, Device ID, and etc. *nSensorIP* is the SensorIP found by using ImageCapSearchSensor() function.

Since then, buffer to transmit Image data is internally assigned, and the thread to receive data is practiced. ImageCapDisconnect() function can be summoned and the allocated buffer can be released or the thread can be closed.

4.9. ImageCapSetWindowTime

Header ImageCAPDllEx.h

Syntax

```
int    ImageCapSetWindowTime (
             unsigned int        nSensorIP,
             int                 nWindowTime
);
```

Parameters

nSensorIP	Sensor IP
nWindowTime	Window Time to change. Unit is milliseconds

Return value

IMAGECAP_OK	Success
Etc.	Error Code

Description

This function changes the window time of the detector.

4.10. ImageCapSetAVGModeFrameNum

Header ImageCAPDllEx.h

Syntax

```
int     ImageCapSetAVGModeFrameNum (
            unsigned int        nSensorIP,
            int                 nFrameNum
);
```

Parameters

nSensorIP	Sensor IP
nFrameNum	frame number

Return value

IMAGECAP_OK	Success
Etc.	Error Code

Description

This function changes the frame number in Average mode.

4.11. ImageCapGetAVGModeFrameNum

Header ImageCAPDllEx.h

Syntax

```
int     ImageCapGetAVGModeFrameNum (
            unsigned int        nSensorIP,
            int*                nFrameNum
);
```

Parameters

nSensorIP	Sensor IP
nFrameNum	frame number

Return value

IMAGECAP_OK	Success
Etc.	Error Code

Description

This function gets the frame number of Average mode

4.12. ImageCalibration

Header CalibrationDllEx.h

Syntax

```
int     ImageCalibration (
            unsigned short*     pObjectImage,
            LPCTSTR             lpszRefPath,
            int                 nWidth,
            int                 nHeight
            int                 nMode,
            CRect                rectMargin,
            BOOL                 bBackGround
);
```

Parameters

pObjectImage	Object image
lpszRefPath	Path of Reference file for Calibration
nWidth	Image data width
nHeight	Image data height
nMode	IMAGECAP_TRIGGER_MODE IMAGECAP_AED_MODE
rectMargin	Specify an area to calibrate
bBackGround	Set Background processing or not TRUE : Perform Background processing FALSE: Does not perform Background processing

Return value

0	Success
-1	Fail

Description

Perform Calibration to the X−ray image acquired through Detector.

Necessary reference files are three types by each mode, and the file name should be fixed.

Trigger_Dark.raw
Trigger_DefectMap.raw
Trigger_Reference00.raw

If three types of reference files are all existing in the path specified by *lpszRefPath*, perform calibration using these files. The result is written in *pObjectImage*.

You need to call *pObjectImage* after allocating the memory by the size of image size.

If there is no Reference file or if you fail to open the file, −1 will be returned.

Currently this function is not used.

Image correction can be done with ImageCapImageAcquistion().

4.13. GenerateReferenceFile

Header CalibrationDllEx.h

Syntax

```
int     GenerateReferenceFile (
            unsigned short*      pDarkImage,
            unsigned short**     pBrightImage,
            int                  nBrightNum,
            int                  nAcqMode,
            LPCTSTR              lpszSavePath,
            int                  nWidth,
            int                  nHeight
);
```

Parameters

pDarkImage	Dark image buffer
pBrightImage	Bright image buffer
nBrightNum	Number of Bright images
nAcqMode	Acquisition mode
lpszSavePath	Save path
nWidth	Image width
nHeight	Image height

Return value

0	No file created
Other	Number of generated Reference files

Description

Generates a reference file for calibration.

Create Gain Map and Defect Map by using Dark image and Bright image.

The file name is different for each mode, and Mode can be checked through ImageCapGetCurrentMode.

The file is saved in SavePath. If this path is passed in during image acquisition, calibrated image acquisition is possible.

5. Return Value Description

Most of the API functions return the IMAGECAP_OK when they succeed.

At the time of Failure, the Error codes are as follows:

IMAGECAP_SOCKET_LOAD_ERROR	Socket Load Error
IMAGECAP_INVALID_SOCKET_ERROR	Socket Formation Error
IMAGECAP_BIND_SOCKET_ERROR	Socket Bind Error
IMAGECAP_SOCKET_OPTION_ERROR	Socket Option Configuration Error
IMAGECAP_COMMAND_SEND_ERROR	Command Transmission Error
IMAGECAP_COMMAND_RECV_ERROR	Data Reception Error
IMAGECAP_RECV_TIMEOUT_ERROR	ACK is not transmitted in certain time.
IMAGECAP_ACK_SEND_ERROR	ACK Transmission Error
IMAGECAP_NO_SENSOR_ERROR	No Sensor in Lan area
IMAGECAP_REGISTER_TYPE_ERROR	A Wrong Register Type
IMAGECAP_GET_HOST_NAME_ERROR	Get Host Name Error
IMAGECAP_NIC_CHECK	No Network Cards
IMAGECAP_STOP_WAIT	Receiving thread is not completed.
IMAGECAP_OK	Success
IMAGECAP_CANCEL	Canceled Image Acquisition
IMAGECAP_NEW_FRAME	Acquisition of a New Image
IMAGECAP_RECEIVING	Failed in Image Acquisition
IMAGECAP_NOTREADY_RO	Tried Image Acquisition within 10 sec Image Cycle Time
Undefined Error Code	Same as Windows System error code

6. Reference S/W description

6.1. Overview

It is a simple MFC dialog which can acquire image data and perform calibration using the acquired data through Detector.

6.2. File Configuration

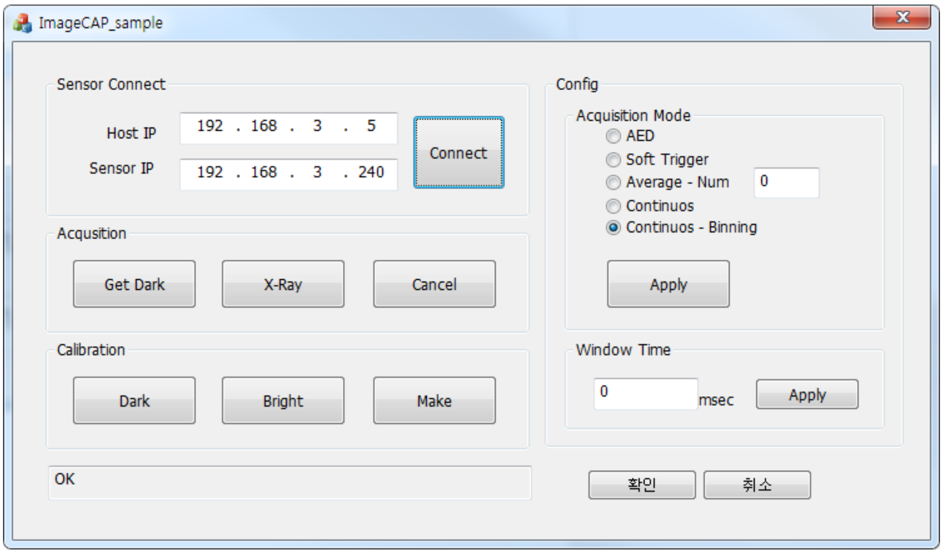
The files necessary for practice consist of 7 files, and the files necessary for Calibration consist of 3 files.

EXE file / DLL	ImageCAP_sample.exe	EXE file
	ImageCaptureDLL.dll	It must be located in the same path as EXE file.
	CalibrationDLL.dll	
	GCDLL.dll	
Data file	ImageCapture.ini	Data file necessary for EXE file. The path should be set when Sensor is connected.
	Ref1.dat	
	Ref2.dat	

- Ref1.dat, Ref2.dat : It is possible to execute without these in the sample SW, however, these are required for the actual Sensor test.

- Calibration reference file : Without this file, Non-calibrated images will be acquired. It is not provided in the sample SW

6.3. Screen Configuration



ImageCAP_sample

Sensor Connect

Host IP 192 . 168 . 3 . 5

Sensor IP 192 . 168 . 3 . 240

Config

Acquisition Mode

○ AED
○ Soft Trigger
○ Average - Num 0
○ Continuos
● Continuos - Binning

Window Time 0 msec

OK

6.4. Sensor Connect

When the connect button is clicked, ImageCapConnectSensor function is called to connect with the detector.

m_strReferencePath is set to ./A_Data.

You should have an ImageCapture.ini file inside the folder.

Then check the FrameSize.

6.5. Acquisition (AED, Soft Trigger, Average Mode)

- Get Dark

This button acquires a dark image and saves it in the A_Cal folder.

The file name is different for each mode.

The file in A_Cal is used to calibrate the X-ray image.

Allocating memory to receive the dark image and using the ImageCapImageAcqusition () function to obtain the image.

If the function returns IMAGECAP_NEW_FRAME, the acquisition was successful.

- X-ray

This button acquires an X-ray image. It is saved as [Image.raw] file in the same path as the executable file.

It works differently depending on the third argument of ImageCapImageAcquisition.

Acquisition Mode	Argument of ImageCapImageAcquisition
AED	BRIGHT_MODE
Soft Trigger	SOFTTRIGGER_INSTANT_MODE
Average	BRIGHT_MODE

AED mode recognizes X-ray automatically, so if you call Acquisition function, it will be in infinite standby. To cancel this, you need to call ImageCapCancel.

The reference file for calibration was set to [. \ A_Cal] folder.

6.6. Acquisition (Continuous Mode & Continuous Binning Mode)

- Get Dark

The dark image is the same as other mods.

However, 10 dark images were obtained for calibration.

- X-Ray

If you click the x-ray button, Continuous mode starts through ImageCapCommandSend, and Image data receives the address value through ImageCapStartCapture function and copies the memory.

You can stop the acquisition through the cancel button.

6.7. Mode Change

The acquisition mode changes to the selected mode.

6.8. Window Time

The window time changes to the time entered.

6.9. Calibration - Make

Obtain a dark image through the Get Dark button, obtain a bright image through the X-ray button, and pass the buffer to the GerateReferenceFile function to create a calibration file.

For samples s/w, the generated calibration file is stored in A_Cal.

The file name is stored differently depending on the acquisition mode.