SDK Document

2021

Revision 22.0

2021/10/20

http://www.otophotonics.com/

CONTENTS

MENU

1.SDK Instruction

1.1 API Brief Introduction

OTO spectrometer SDK(Software Development Kit) provides API in dynamic link library for using OTO spectrometer conveniently and efficiently .

SDK includes different class :

- Connection
- Device Information
- Spectrum Acquire
- Spectrum Correction
- Color Measurement
- Other

Refer to following steps to use these API functions

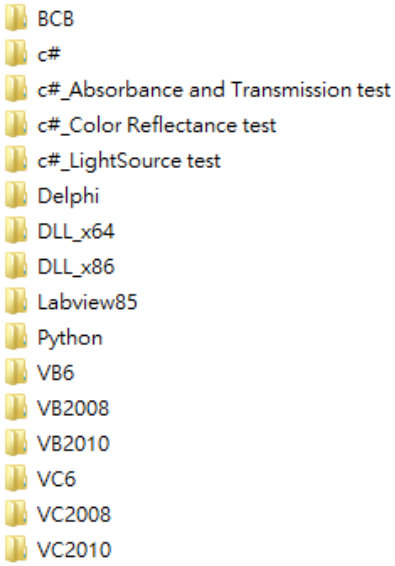
Step	API functions
Get spectrometer number and open spectrometer to get spectrometer handle	UAI_SpectrometerGetDeviceList() UAI_SepctrometerGetDeviceAmount() UAI_SpectrometerOpen()
Get Wavelength information and spectrum data	UAI_SpectromoduleGetFrameSize() UAI_SpectrometerWavelengthAcquire() UAI_SpectrometerDataAcquire() UAI_SpectrometerDataOneShot()
Do Calibration(Optional)	UAI_BackgroundRemove() UAI_LinearityCorrection() UAI_ContrastIntensityCorrection() UAI_AbsoluteIntensityCorrection()
Get Color Information(Optional)	UAI_ColorInformationAllocation() UAI_ColorOperation() UAI_ColorGetxyz() UAI_ColorGetCCT() UAI_ColorInformationFree()
Close spectrometer connection	UAI_SpectrometerClose()

1.2 Programming Language Support

SDK supports programming languages C++ , VB6 , c# , VB.net , Delphi , Python and Labview. This document introduces how to use c# sample . Please refer sample code for more applications in other languages.

1.3 Sample Code

SDK includes sample code to present in C++ , VB6 , c# ,VB.net , Delphi , python and Labview for your reference.



1.4 Header Files And Link Library

UserApplication.dll , SiUSBXp.dll must be included .

OTO Provides x86 and x64 library for developers which are located at

X86 : Application folder\SDK\DLL_x86

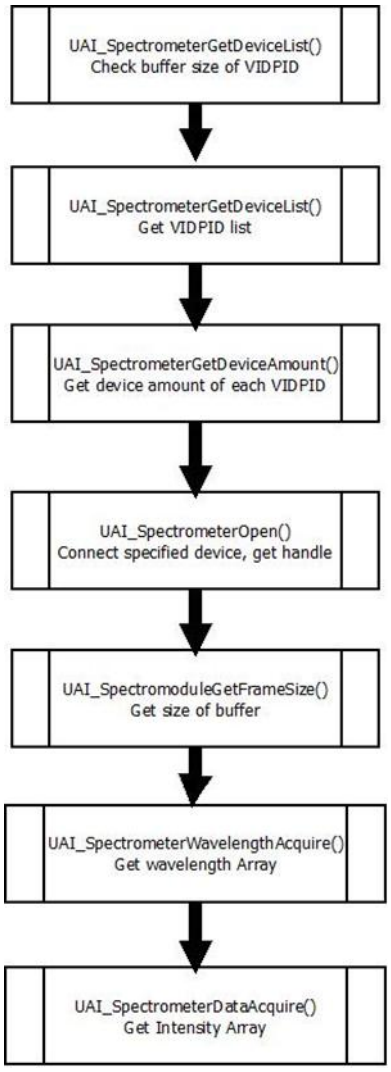
X64 : Application folder\SDK\DLL_x64

1.5 Platform

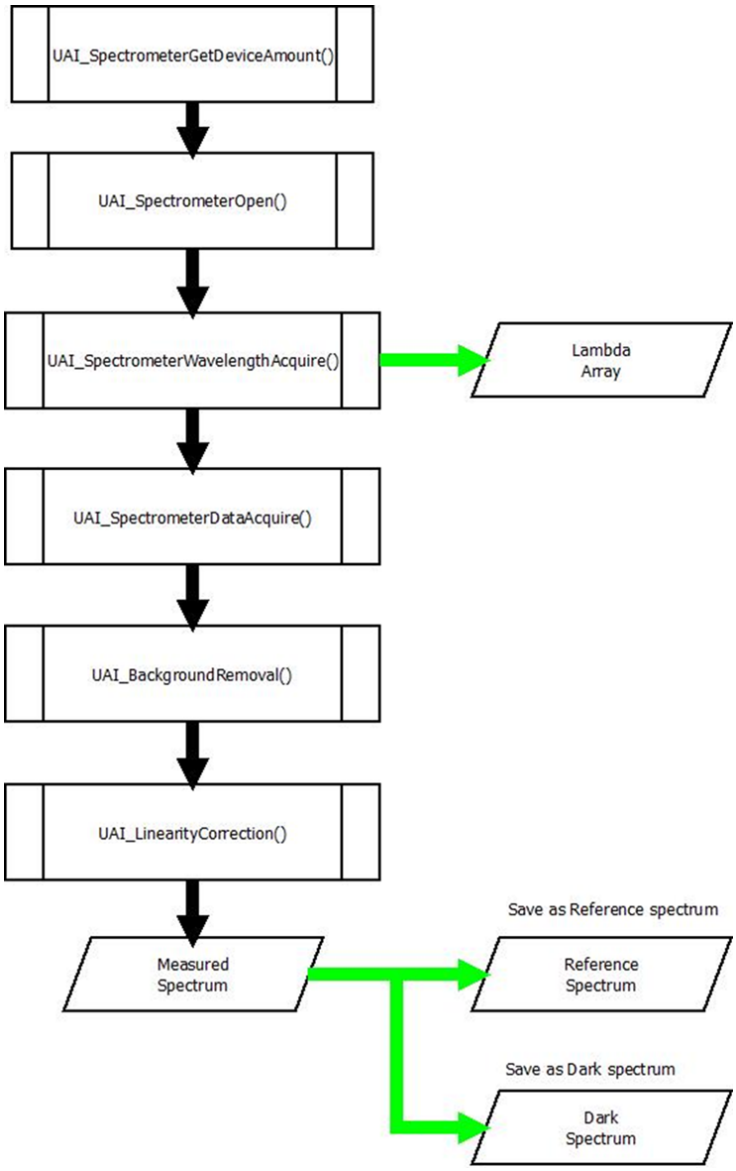
The SDK now supports Windows platform (Windows XP , Windows Vista and Windows7,8,10).

2. Application and measurement development flow chart

2.1. Spectrum Connection

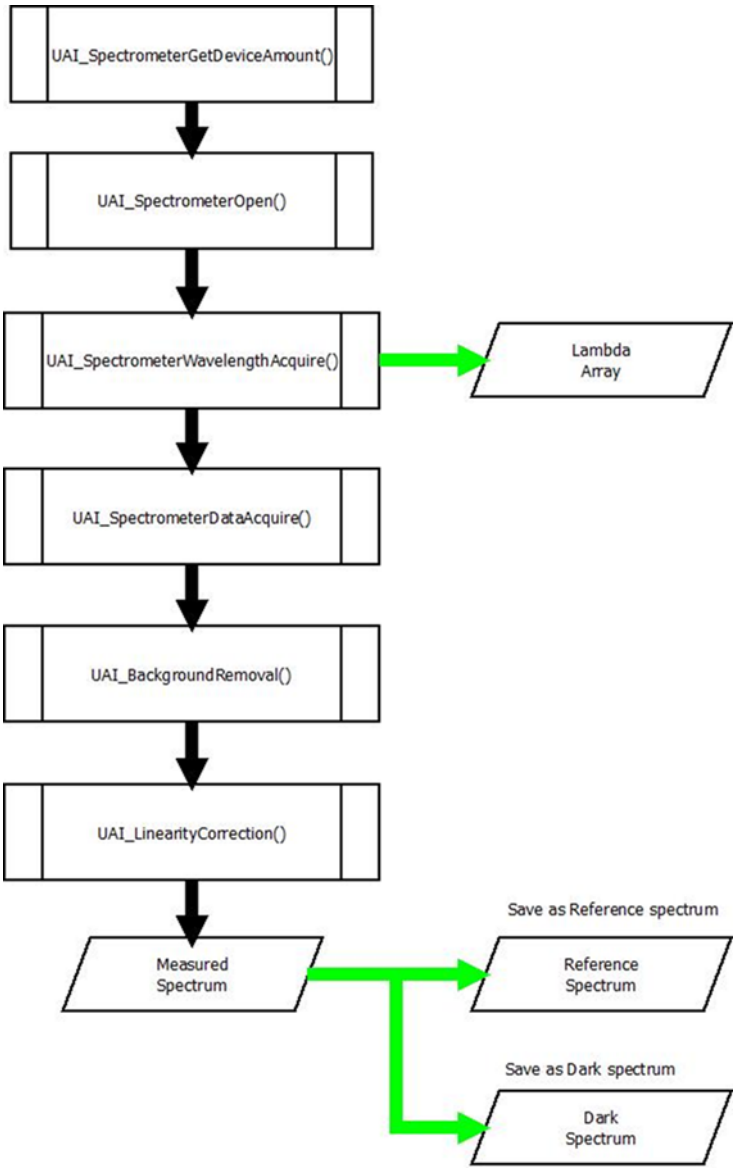


2.2. Absorbance Measurement



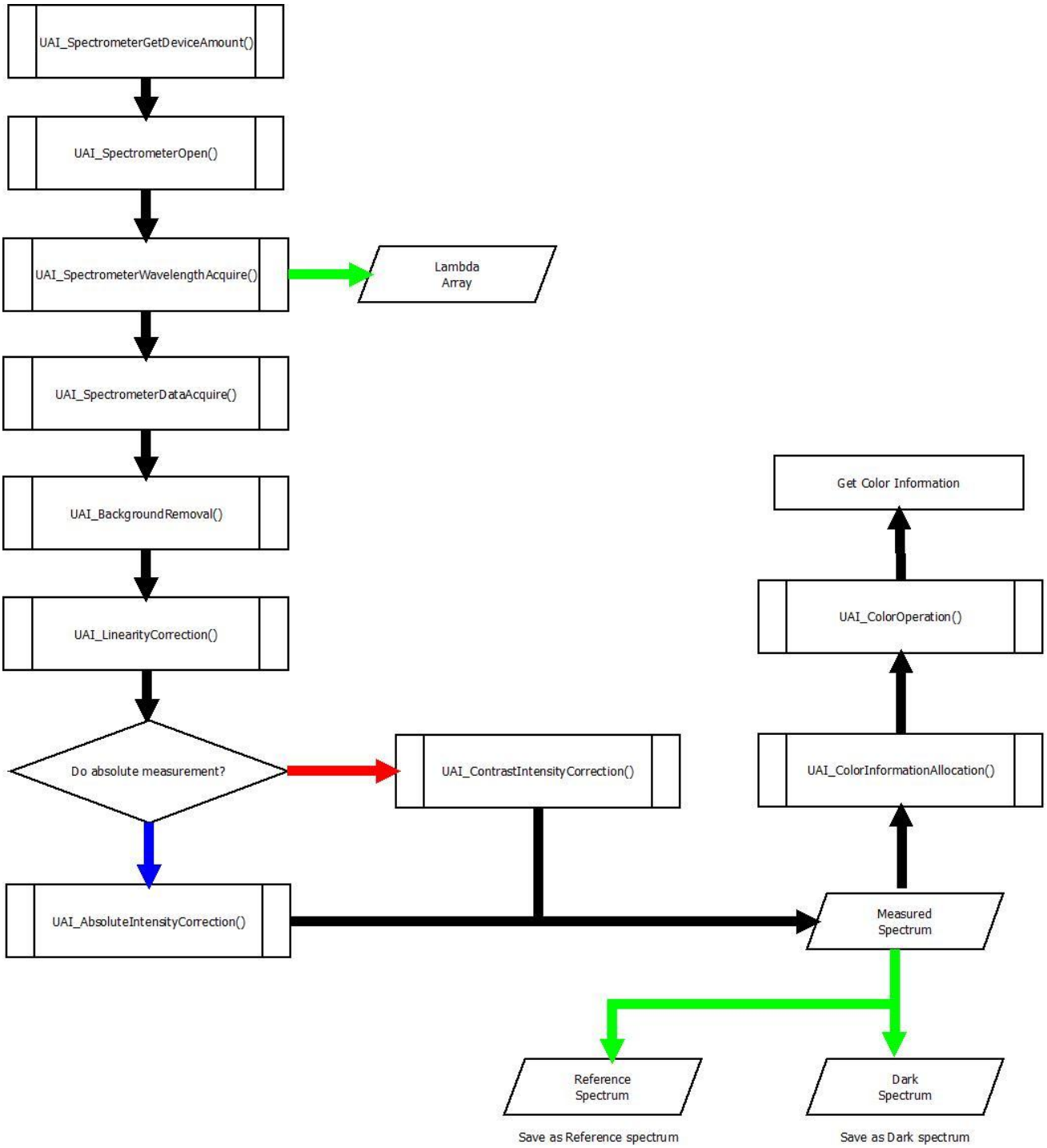
$$\text{Absorbance} = -\log \frac{\text{Measured-Dark}}{\text{Reference-Dark}}$$

2.3. Transmission & Reflection Measurement



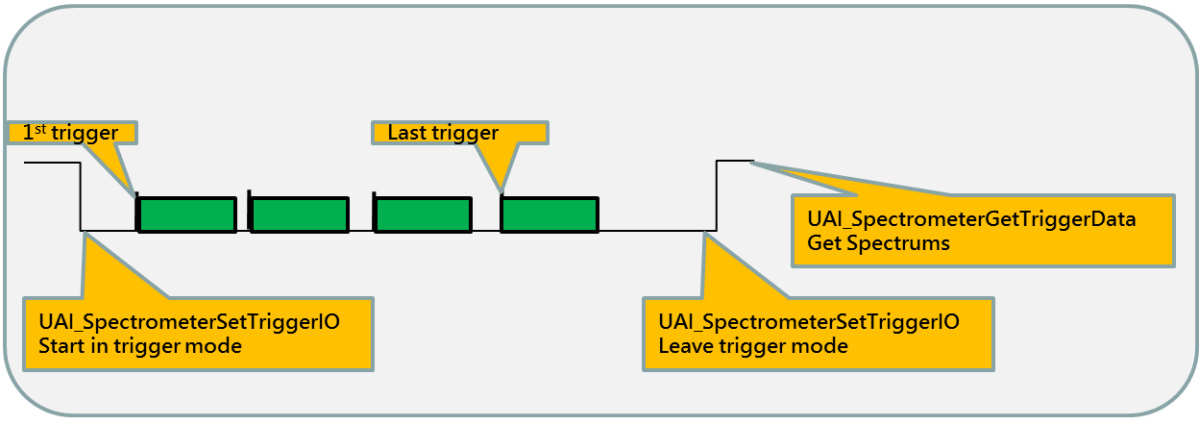
$$\text{Result}= \frac{\text{Measured-Dark}}{\text{Reference-Dark}} \boldsymbol{X}\ \mathbf{100\%}$$

2.4. Color Measurement

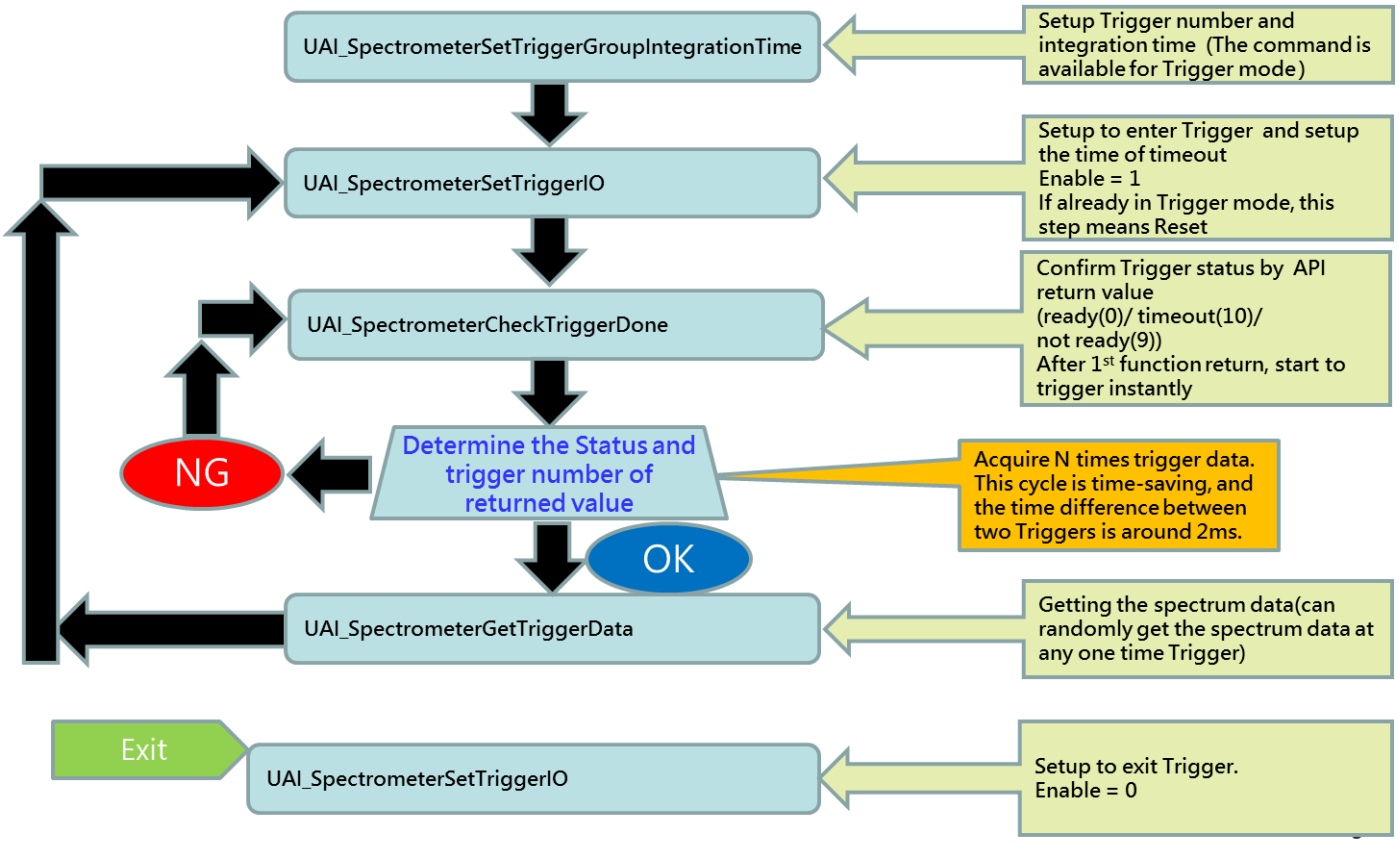

UAI_SpectrometerGetDeviceAmount()
UAI_SpectrometerOpen()
UAI_SpectrometerWavelengthAcquire()
Lambda Array
UAI_SpectrometerDataAcquire()
UAI_BackgroundRemoval()
UAI_LinearityCorrection()
Do absolute measurement?
UAI_ContrastIntensityCorrection()
UAI_AbsoluteIntensityCorrection()
Get Color Information
UAI_ColorOperation()
UAI_ColorInformationAllocation()
Measured Spectrum
Reference Spectrum
Dark Spectrum
Save as Reference spectrum
Save as Dark spectrum

2.5. Trigger Flow

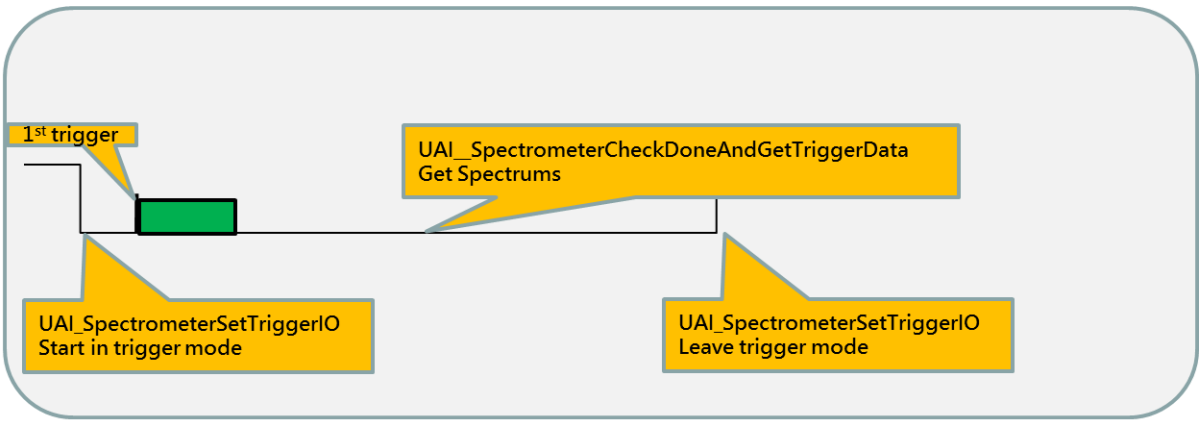
2.5.1. Multi Trigger Multi Data

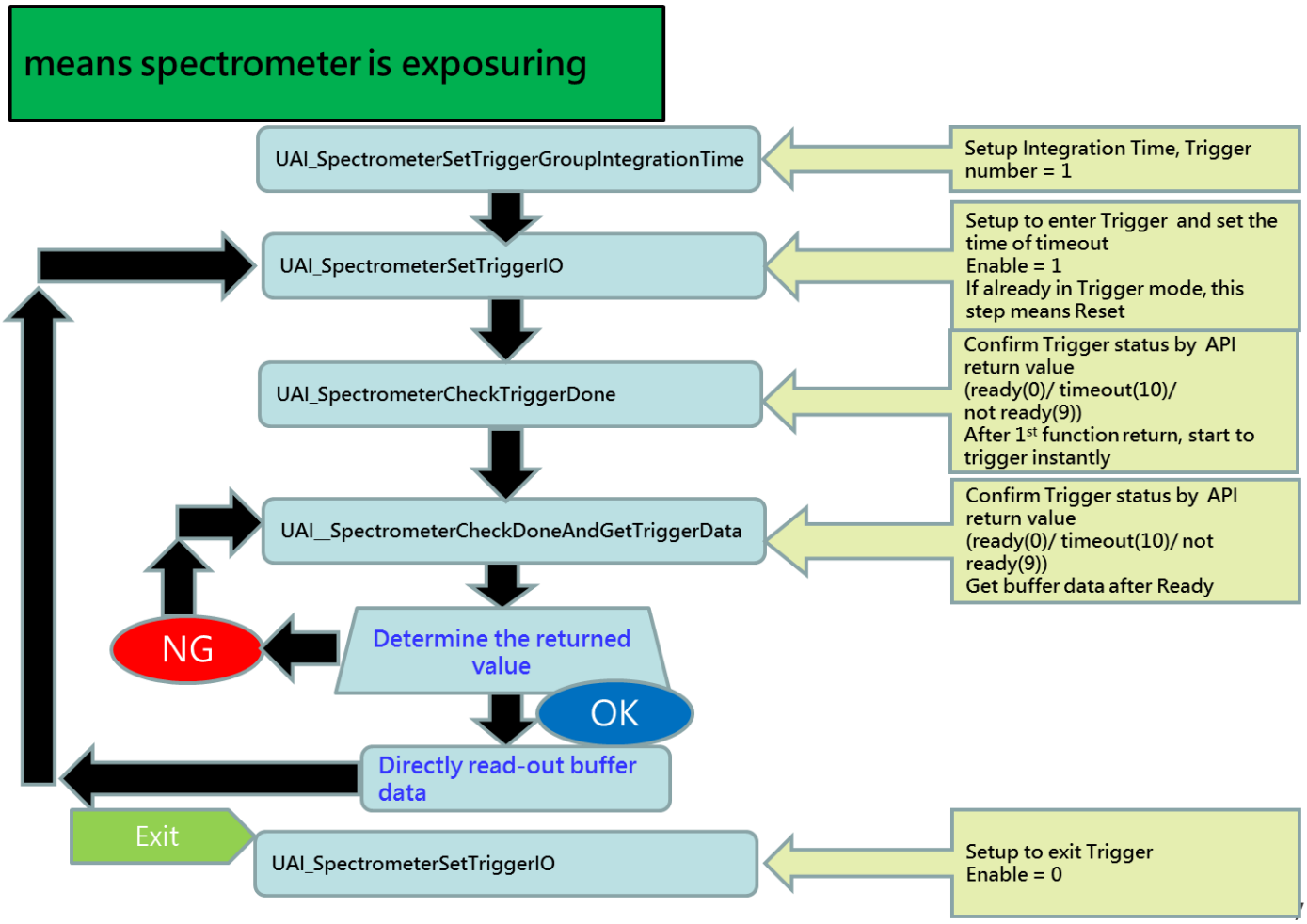


means spectrometer is exposuring

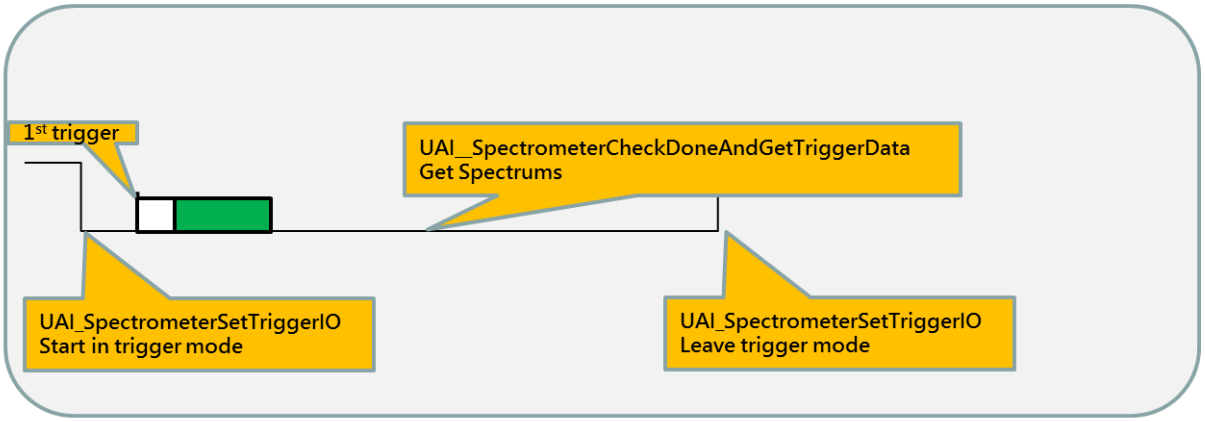


2.5.2. Single Trigger Single Data



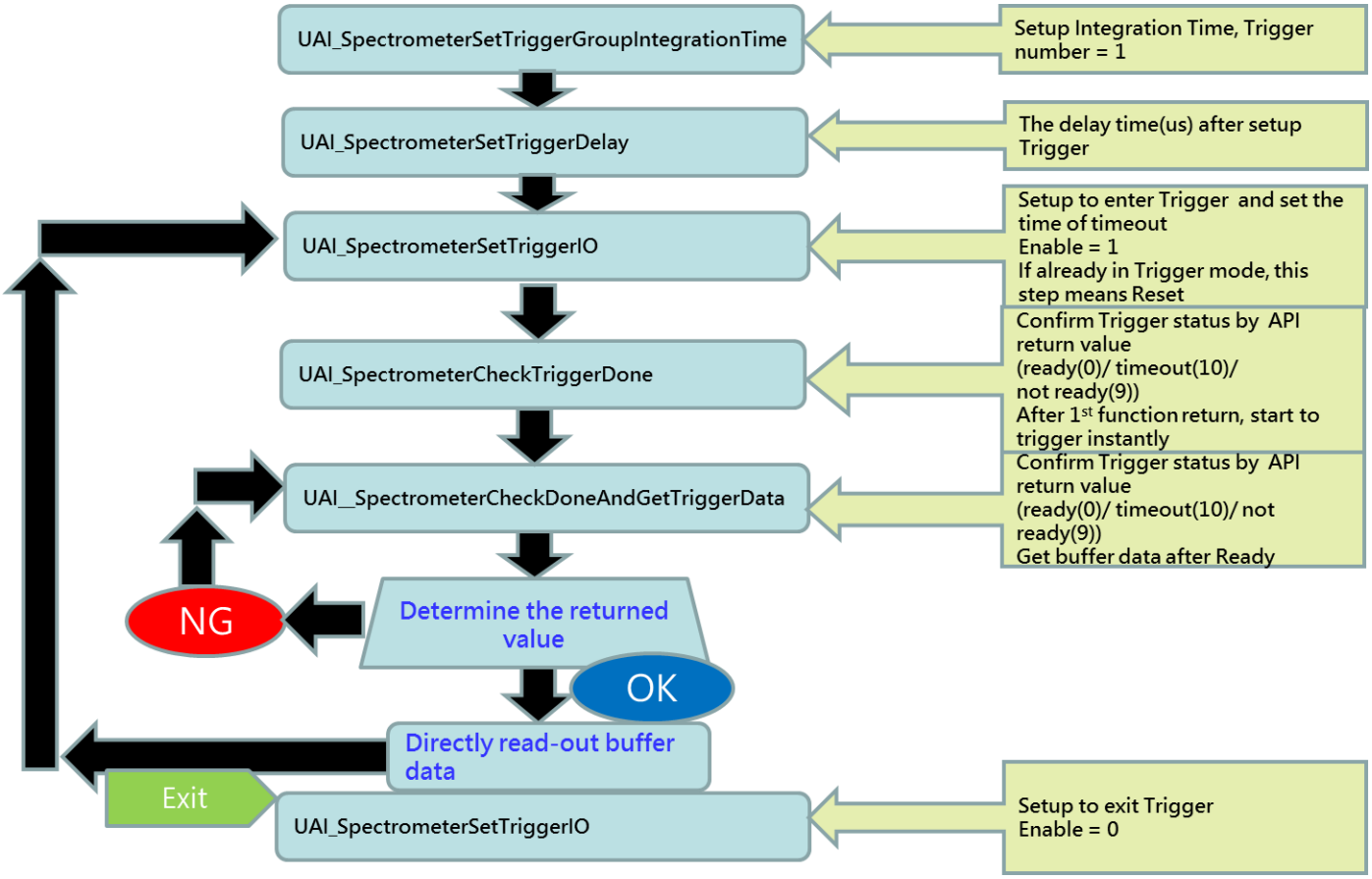


2.5.3. Single Trigger Single Data with Delay

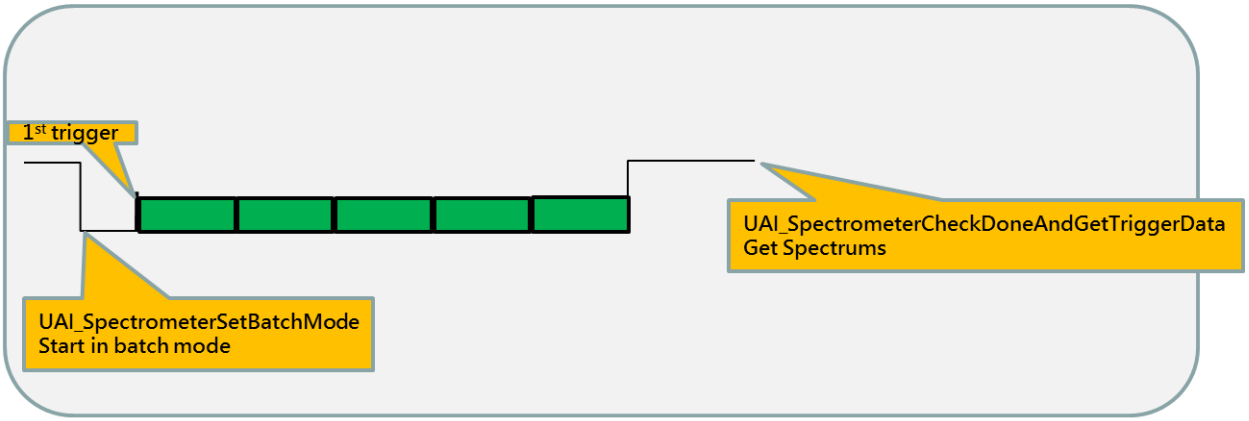


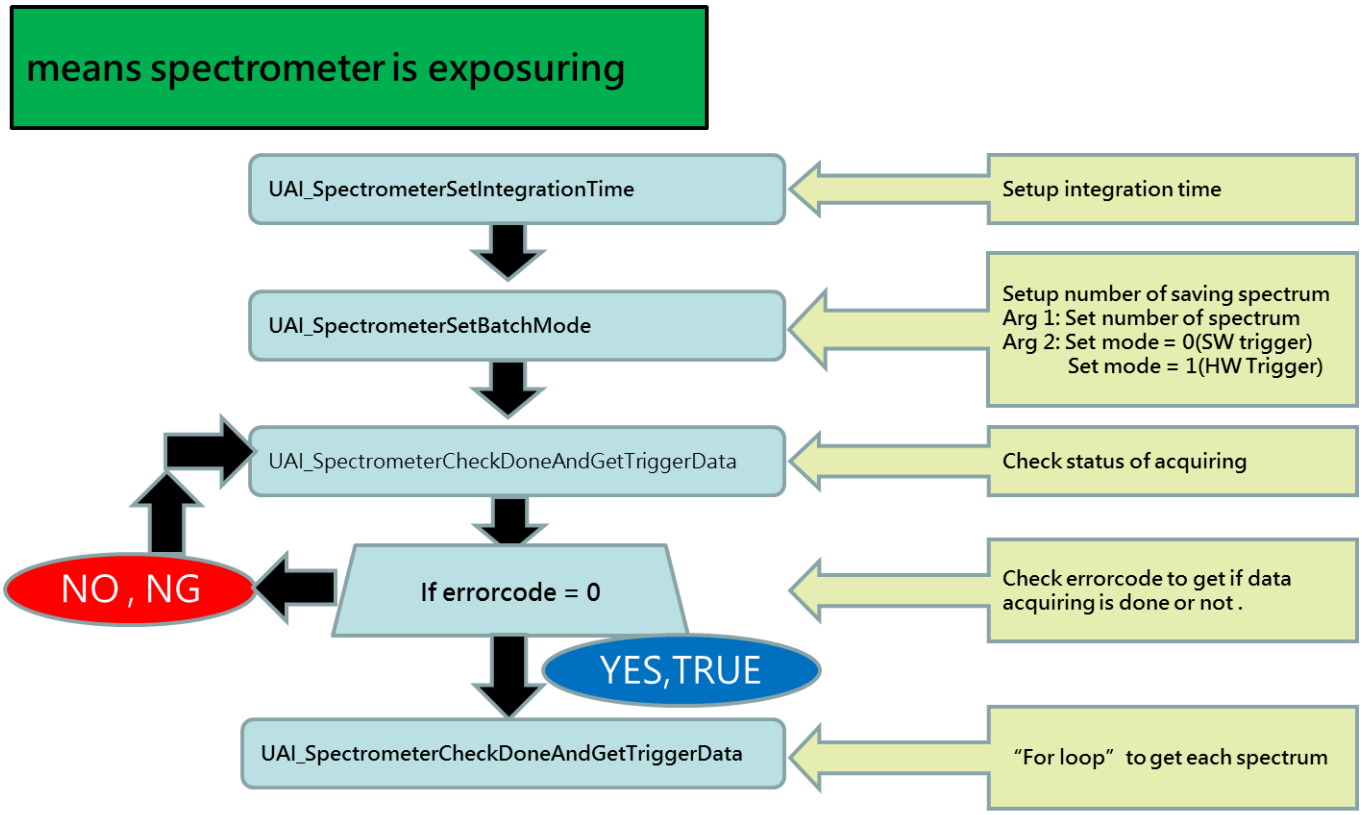
Means delay time between getting trigger and exposure.

means spectrometer is exposuring



2.5.4. Single Trigger Continuous Data

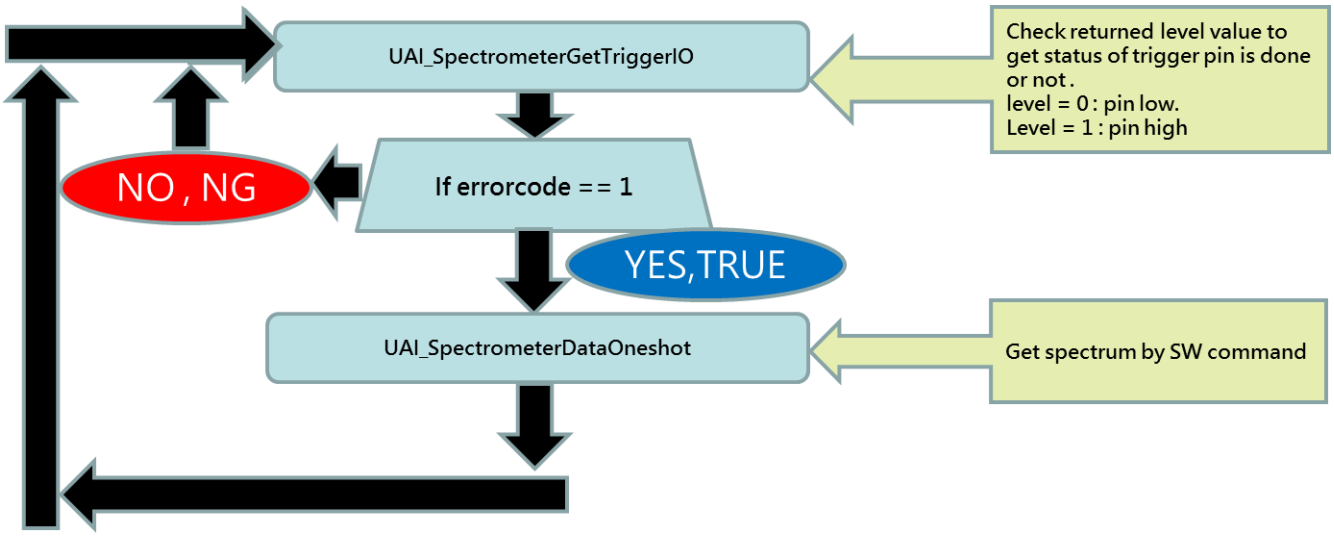




※The firmware of SE2020 , F2.1.33_A103 or later , supports hardware trigger in batch mode.

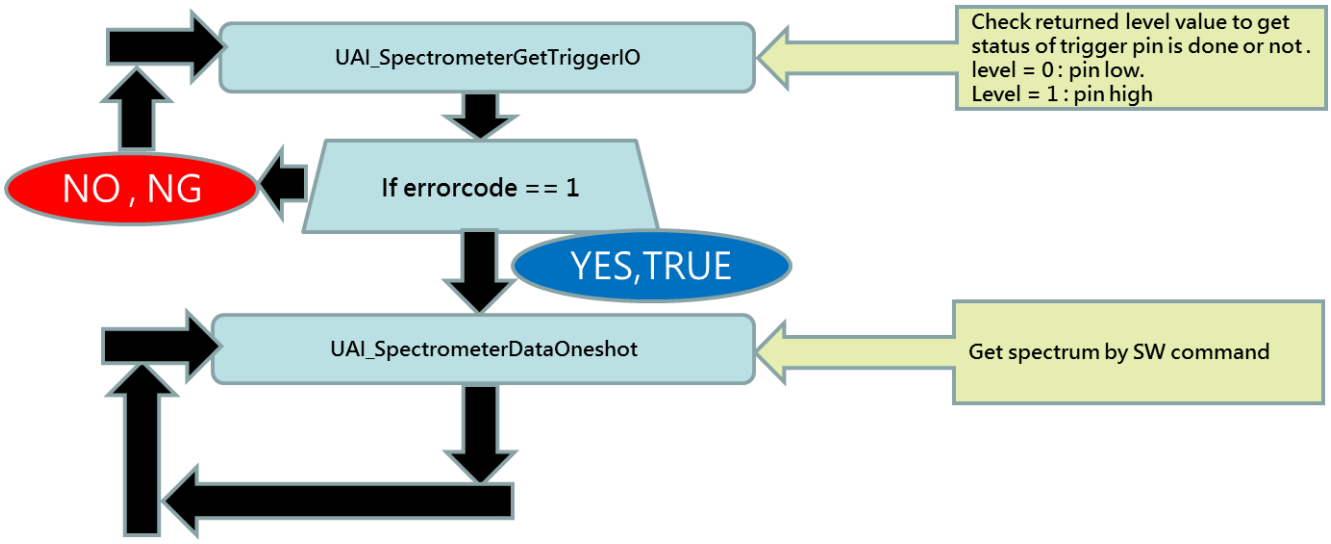
2.5.5. Software Level Trigger

Spectrometer in SW Level mode have to check status of external single.
If status of external single changes to high level , spectrasmart will keep data acquiring.
If status of external single changes to low level , spectrasmart will stop data acquiring.

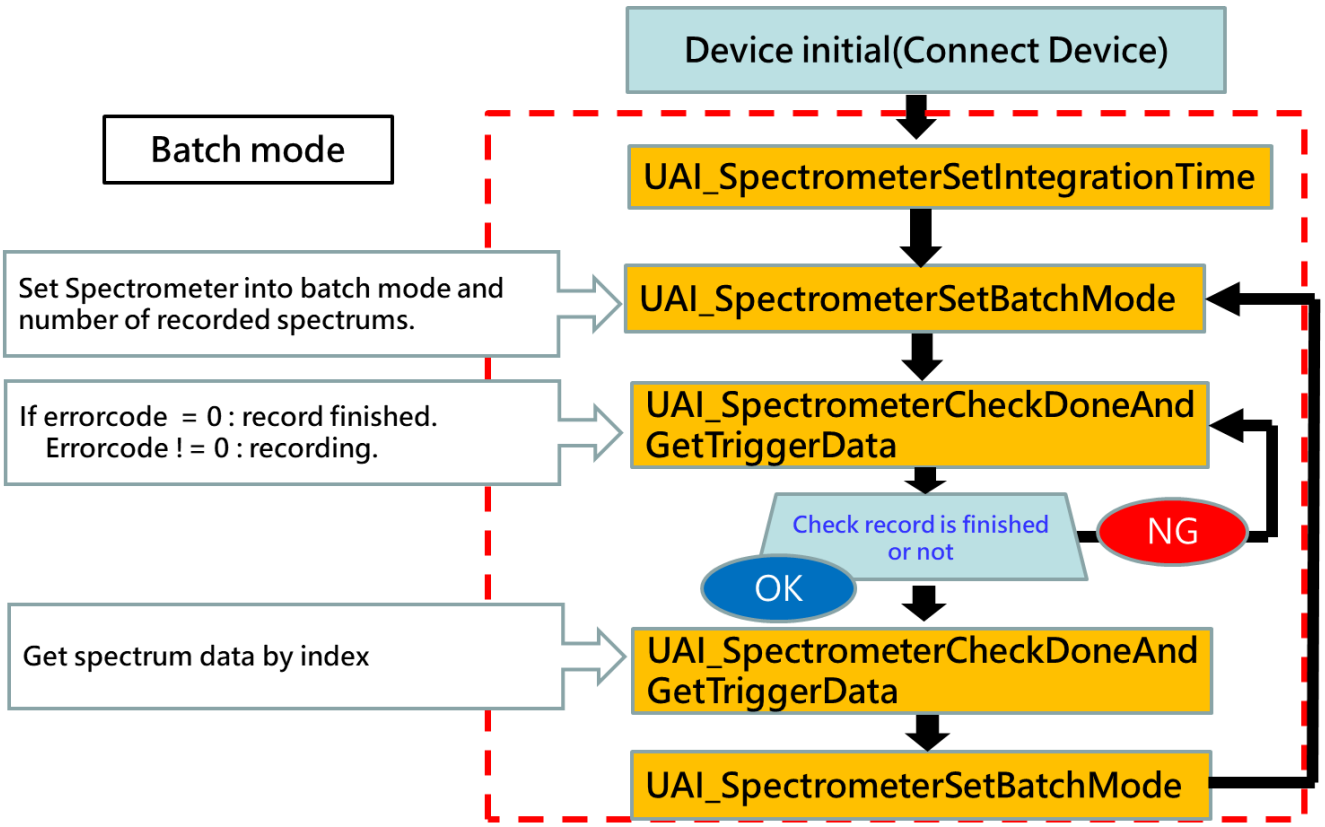


2.5.6. Software Continuous Trigger

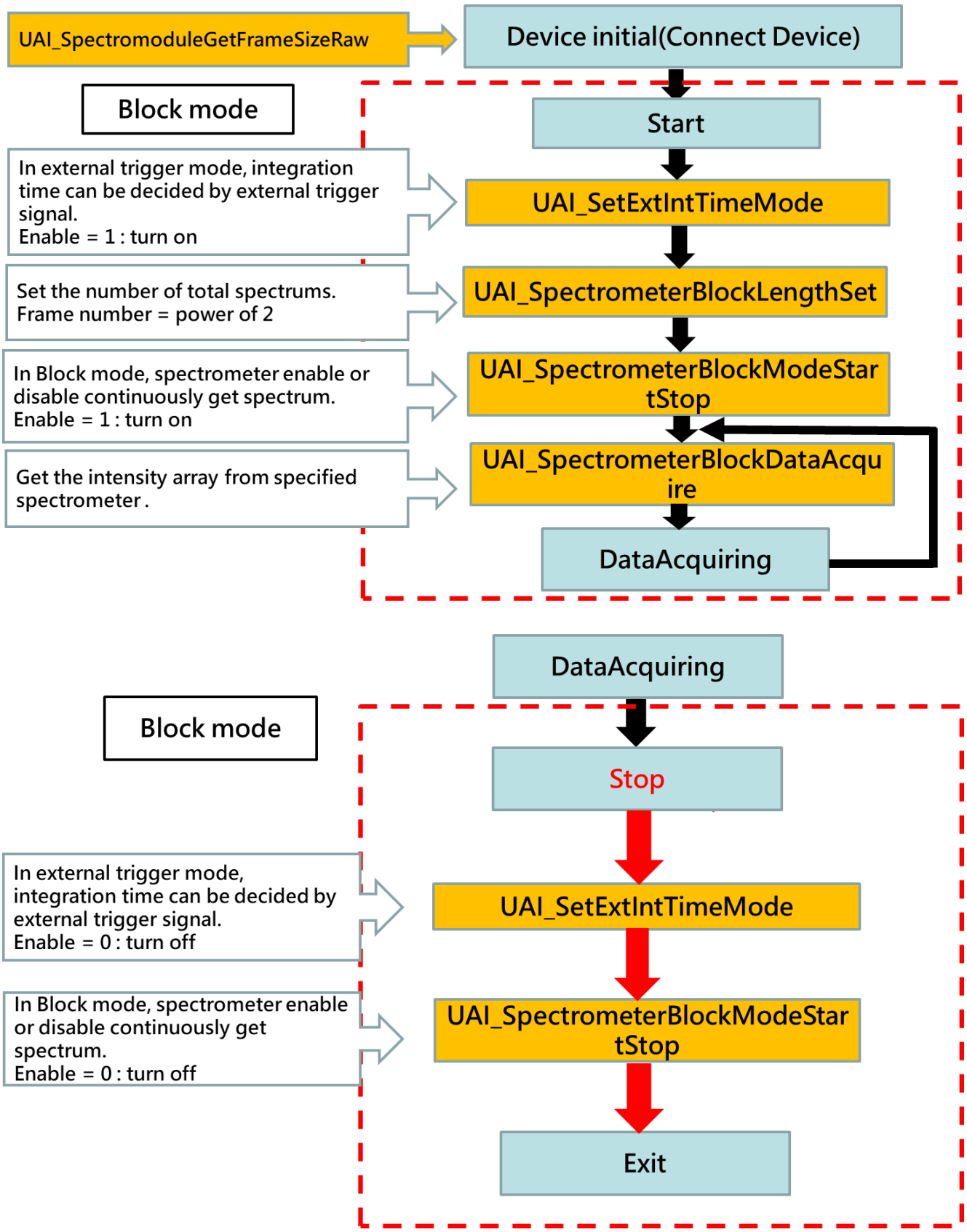
Spectrometer in SW continuous mode have to check status of external single.
If status of external single changes to high level , spectrasmart will start data acquiring and it will be stop until leaving SW continuous mode.



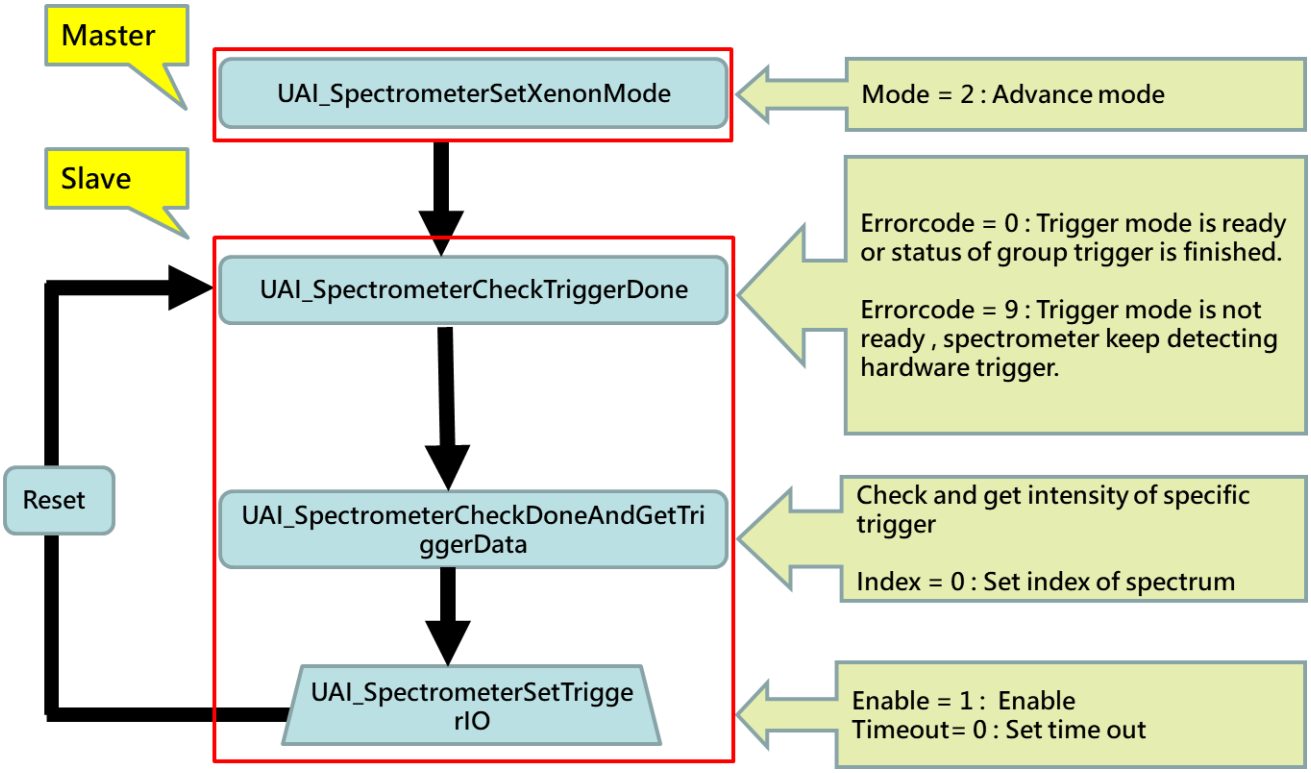
2.6. Batch mode Flow



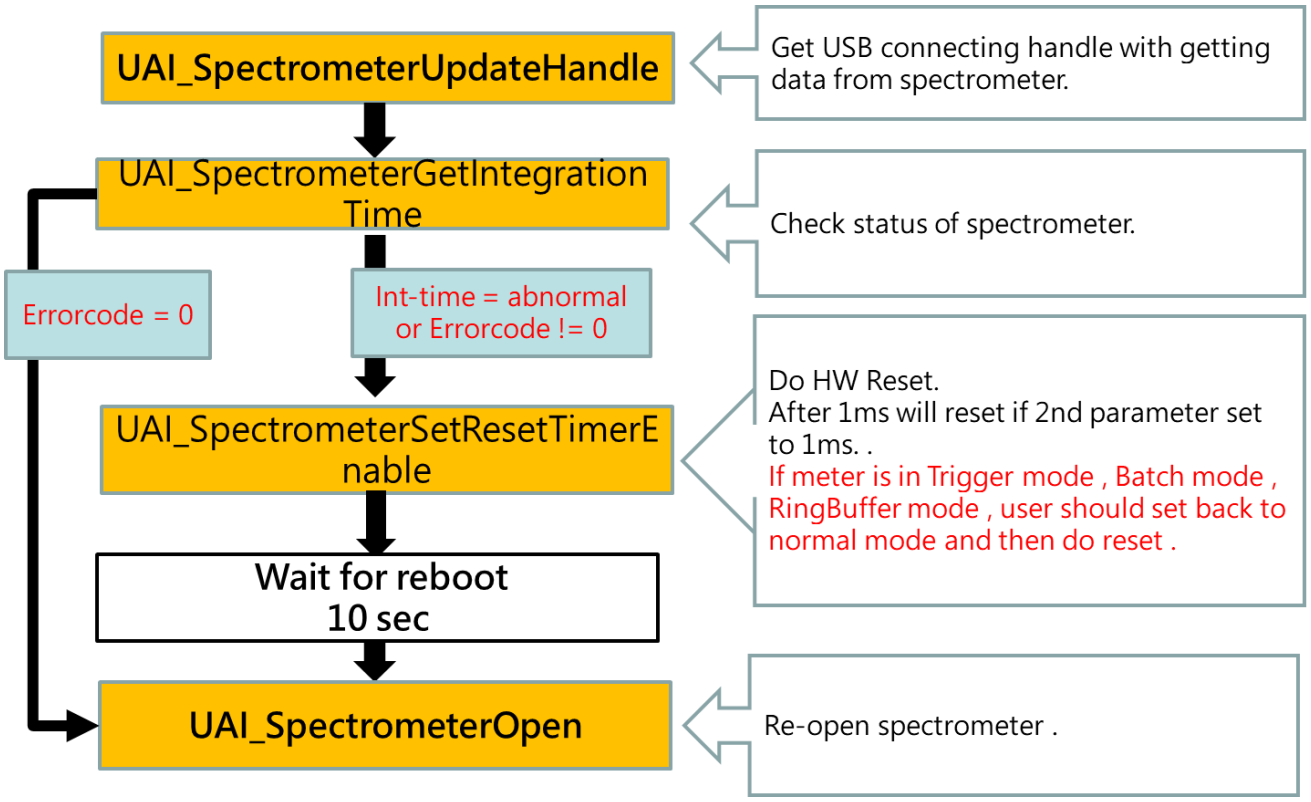
2.7. Ring Buffer Flow



2.8. Pulse Setting Flow



2.9. Hardware reset flow



3. API Function Instruction

3.1. Error code List

API function will return error code to show the function is successful or unsuccessful.

Errorcode	Description
0	API_SUCCESS
1	API_INT_BUFFER_INVALID
2	API_INT_FEATURE_UNSUPPORTED
3	API_INT_PROTOCOL_ERROR
4	API_INT_CALIBRATION_ERROR
5	API_INT_MEMORY_ERROR
6	API_INT_ARGUMENT_ERROR
7	API_INT_HANDLE_INVALID
8	API_INT_TIMEOUT
9	API_INT_DATA_NOT_READY
10	API_INT_DATA_TIME_OUT
11	API_INT_FILE_IO_ERROR
12	API_INT_FILE_Exist_ERROR
0x80000000	API_EXT_START

3.2. Connection

3.2.1. UAI_SpectrometerOpen

Declare :

UINT UAI_SpectrometerOpen (unsigned int dev, void handle,unsigned int VID , unsigned int PID)**

Visual Basic Declare :

_VB6UAI_SpectrometerOpen@16

Open the specified spectrometer and get the spectrometer handle.

Parameter

Name	Description
dev	Specify the index for the spectrometer.
handle	Return to the pointer of handle of the spectrometer.
VID	Provide specified VID.
PID	Provide specified PID.

Introduction

Before using the spectrometer, user have to create a spectrometer handle with this API.
The spectrometer handle will be related with the assigned spectrometer, all API's that configure or control this spectrometer have to base on this handle to work properly in the system with multiple spectrometer installed.
Once the spectrometer handle is created, the corresponding spectrometer is in use. If other thread or process use the same index of spectrometer and call UAI_spectrometerOpen(), it will create new spectrometer handle can release old spectrometer handle.
Please refer example code : Device Connection.

3.2.2. UAI_SpectrometerClose

Declare :

UINT UAI_SpectrometerClose (void* handle)

Visual Basic Declare :

_VB6UAI_SpectrometerClose@4

Close the specified spectrometer and release the spectrometer handle.

Parameter

Name	Description
handle	Specify the spectrometer handle.

Introduction

Once the thread or process use UAI_SpectrometerClose(), the spectrometer handle will be released. The corresponding spectrometer could be used again by other threads or processes.

3.2.3. UAI_SpectrometerUpdateHandle

Declare :

UINT UAI_SpectrometerUpdateHandle (unsigned int dev, void handle,unsigned int VID , unsigned int PID)**

Visual Basic Declare :

_VB6UAI_SpectrometerUpdateHandle@16

Get the spectrometer handle only for Hardware reset application.

Parameter

Name	Description
dev	Specify the index for the spectrometer.
handle	Return to the pointer of handle of the spectrometer.
VID	Provide specified VID.
PID	Provide specified PID.

Introduction

This function get USB handle from spectrometer without getting device information. User can't use device handle by update handle function .

3.2.4. UAI_SpectrometerSetResetTimerEnable

Declare :

UINT UAI_SpectrometerSetResetTimerEnable (void* handle,unsigned int time_ms)

Visual Basic Declare :

_VB6UAI_SpectrometerSetResetTimerEnable@8

Force to reset spectrometer.

Parameter

Name	Description
handle	Handle of the spectrometer.
Time_ms	Wait for time_ms to reset spectrometer

3.3. Device Information

3.3.1. UAI_FirmwareGetVersion

Declare :

UINT UAI_FirmwareGetVersion(void* handle , unsigned int* Version)

Visual Basic Declare :

_VB6UAI_FirmwareGetVersion@8

Get Firmware version of specified spctromodule

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
Version	Return to the pointer of version.

Introduction

Returned array can be used at index 0~2 for identifying firmware version.

C# sample

```
**********************************************************************************
byte[] temp_FWVersion = new byte[8];
UAI_FirmwareGetVersion(DeviceHandle, temp_FWVersion);
temp_FWVersion[2] = (byte)(temp_FWVersion[2] >> 4);
string FWVersion = Convert.ToChar(temp_FWVersion[0]).ToString() + temp_FWVersion[1].ToString().PadLeft(3,
'0') + "." + temp_FWVersion[2].ToString()"." + temp_FWVersion[3].ToString();
**********************************************************************************
```

3.3.2. UAI_SpectrometerGetDeviceAmount

Declare :

UINT UAI_SpectrometerGetDeviceAmount (unsigned int vid , unsigned int pid , unsigned int* dev)

Visual Basic Declare :

_VB6UAI_SpectrometerGetDeviceAmount@12

Get amount of connecting spectrometer in current system

Parameter

Name	Description
VID	Provide specified VID.
PID	Provide specified PID.
dev	Return to the amount of connecting spectrometer.

Introduction

The SDK supports multiple spectrometer (either in the same or different processes), which means users have to know the amount of connecting spectrometer in current system and how to operate desired one. Please refer example code : Device Connection.

3.3.3. UAI_SpectrometerGetDeviceList

Declare :

UINT UAI_SpectrometerGetDeviceList(unsigned int* number , unsigned int* list)

Visual Basic Declare :

_VB6UAI_SepctrometerGetDeviceList@8

Get number of supporting VID&PID list. Before software connects spectrometer , software have to get VIDPID list and uses the list to get amount of connecting spectrometer and open spectrometer.

Parameter

Name	Description
number	Return the size of VIDPID array.
list	Return the point of VIDPID array.

Introduction

The SDK supports several VIDPID , software have to get number of VIDPID group and prepare necessary buffer for this function . Function will return VIDPID array list . Please refer example code : Device Connection.

3.3.4. UAI_SpectrometerGetModelName

Declare :

UINT UAI_SpectrometerGetModelName(void* api_handle, unsigned char *model)

Visual Basic Declare :

_VB6UAI_SpectrometerGetModelName@8

Get the model name of specified spectrometer.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
model	Return to the pointer of model name. Length of data : 16 byte

3.3.5. UAI_SpectrometerGetSerialNumber

Declare :

UINT UAI_SpectrometerGetSerialNumber (void* api_handle, unsigned char *serial)

Visual Basic Declare :

_VB6UAI_SpectrometerGetSerialNumber@8

Get the serial number of specified spectrometer.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
serial	Return to the pointer of serial number. Length of data : 16 byte

3.3.6. UAI_SpectromoduleGetFrameSize

Declare :

UINT UAI_SpectromoduleGetFrameSize(void* api_handle, unsigned short *size)

Visual Basic Declare :

_VB6UAI_SpectromoduleGetFrameSize@8

Get the frame size of specified spectrometer.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
size	Return to the pointer of framesize.

Introduction

Before using UAI_SpectrometerWavelengthAcquire() and UAI_SpectrometerDataAcquire(), user have to use UAI_SpectromoduleGetFrameSize() to get correct frame size from spectrometer. Each spectrometer has different available framesize.

C# sample

Please refer sample code.

3.3.7. UAI_SpectromoduleGetFrameSizeRaw

Declare :

UINT UAI_SpectromoduleGetFrameSizeRaw(void* api_handle, unsigned short *size)

Visual Basic Declare :

_VB6UAI_SpectromoduleGetFrameSize@8

Get Full frame size of specified spectrometer.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
size	Return to the pointer of raw framesize.

Introduction

Full frame size of spectrometer means pixel number of CCD sensor in spectrometer. It depends on different type of CCD.

3.3.8. UAI_SpectromoduleGetMaximumIntegrationTime

Declare :

UINT UAI_SpectromoduleGetMaximumIntegrationTime(void* api_handle, unsigned int *time)

Visual Basic Declare :

_VB6UAI_SpectromoduleGetMaximumIntegrationTime@8

Get max supported integration time(ms)

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
time	Return to the pointer of time(ms).

3.3.9. UAI_SpectromoduleGetMinimumIntegrationTime

Declare :

UINT UAI_SpectromoduleGetMinimumIntegrationTime(void* api_handle, unsigned int *time)

Visual Basic Declare :

_VB6UAI_SpectromoduleGetMinimumIntegrationTime@8

Get min supported integration time(us)

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
time	Return to the pointer of time(us).

3.3.10.UAI_SpectromoduleGetWavelengthCalibrationCoefficients

Declare :

UINT UAI_ SpectromoduleGetWavelengthCalibrationCoefficients (void* api_handle, double *coefficients)

Visual Basic Declare :

_VB6UAI_SpectromoduleGetWavelengthCalibrationCoefficients@8

Get coefficients of wavelength calibration.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
coefficient	Return to the pointer of coefficients. Length of data : 11 double.

Introduction

User need to prepare buffer of coefficients and the length of buffer is 11 . Available number of wavelength coefficients will be different that is based on the wavelength calibration of each spectrometer.

3.3.11.UAI_SpectromoduleGetWavelengthStart

Declare :

UINT UAI_SpectromoduleGetWavelengthStart(void* api_handle, float* lambda)

Visual Basic Declare :

_VB6UAI_SpectromoduleGetWavelengthStart@8

Get first available wavelength.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
lambda	Return to the pointer of wavelength.(unit: nm)

3.3.12.UAI_SpectromoduleGetWavelengthEnd

Declare :

UINT UAI_SpectromoduleGetWavelengthEnd (void* api_handle, float* lambda)

Visual Basic Declare :

_VB6UAI_SpectromoduleGetWavelengthEnd@8

Get the latest available wavelength.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
lambda	Return to the pointer of wavelength.(unit: um)

3.4. Spectrum Acquire

3.4.1. UAI_SpectrometerWavelengthAcquire

Declare :

UINT UAI_SpectrometerWavelengthAcquire(void* api_handle, float *buffer)

Visual Basic Declare :

_VB6UAI_SpectrometerWavelengthAcquire@8

Get available array of wavelength from specified spectrometer.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
buffer	Return to the pointer of created 1D array buffer.(unit: nm)

Introduction

Before using UAI_SpectrometerWavelengthAcquire(), user have to create a 1D array buffer which have the same size as frame size. User could use UAI_SpectromoduleGetFrameSize() to get correct frame size from spectrometer.

Once using this API to get the wavelength list, the list is based on specified spectrometer.

C# sample

Please refer sample code.

3.4.2. UAI_SpectrometerWavelengthAcquireRaw

Declare :

UINT UAI_SpectrometerWavelengthAcquireRaw(void* api_handle, float *buffer)

Visual Basic Declare :

_VB6UAI_SpectrometerWavelengthAcquireRaw@8

Get full array of wavelength from specified spectrometer.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
buffer	Return to the pointer of created 1D array buffer.(unit: nm)

Introduction

Before using UAI_SpectrometerWavelengthAcquireRaw(), user have to create a 1D array buffer which have the same size as frame size. User could use UAI_SpectromoduleGetFrameSizeRaw() to get correct frame size from spectrometer.

Once using this API to get the wavelength list, the list is based on specified spectrometer.

3.4.3. UAI_SpectrometerGetIntegrationTime

Declare :

UINT UAI_SpectrometerGetIntegrationTime(void* api_handle, unsigned int * integration_time_us)

Visual Basic Declare :

_VB6UAI_SpectrometerGetIntegrationTime@8

Get the setting of integration time from specified spectrometer .

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
integration_time_us	Get integration time(unit: us)

3.4.4. UAI_SpectrometerSetIntegrationTime

Declare :

UINT UAI_SpectrometerSetIntegrationTime(void* api_handle, unsigned int integration_time_us)

Visual Basic Declare :

_VB6UAI_SpectrometerSetIntegrationTime@8

Set the setting of integration time from specified spectrometer.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
integration_time_us	Set integration time(unit: us)

3.4.5. UAI_SpectrometerDataAcquire

Declare :

UINT UAI_SpectrometerDataAcquire(void* api_handle, unsigned int integration_time_us, float *buffer, unsigned int average)

Visual Basic Declare :

_VB6UAI_SpectrometerDataAcquire@16

Get the intensity array with smoothing function from specified spectrometer .

Parameter

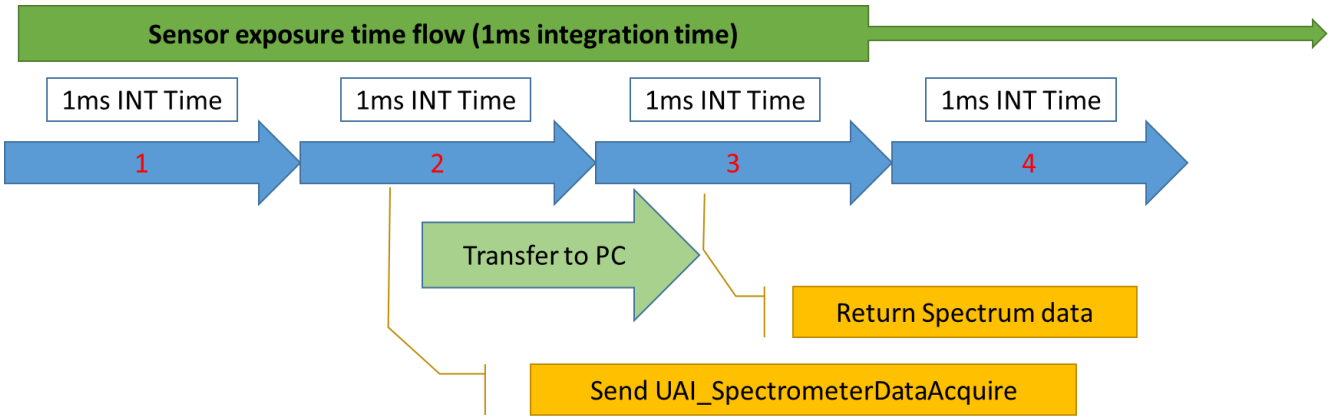
Name	Description
api_handle	Specify the spectrometer handle.
integration_time_us	Set integration time(unit: us)
buffer	Return to the pointer of created 1D array buffer.(unit: counts)
average	Set average times of acquire data.

Introduction

Before using UAI_SpectrometerDataAcquire(), user have to create a 1D array buffer which have the same size as frame size. User could use UAI_SpectromoduleGetFrameSize() to get correct frame size from spectrometer.

C# sample

Please refer sample code.



3.4.6. UAI_SpectrometerDataAcquires

Declare :

UINT UAI_SpectrometerDataAcquires(void* api_handle, unsigned int integration_time_us, float *buffer, unsigned int average)

Visual Basic Declare :

_VB6UAI_SpectrometerDataAcquires@16

Get the intensity array from specified spectrometer .

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
integration_time_us	Set integration time(unit: us)
buffer	Return to the pointer of created 1D array buffer.(unit: counts)
average	Set average times of acquire data.

Introduction

Before using UAI_SpectrometerDataAcquires(), user have to create a 1D array buffer which have the same size as frame size. User could use UAI_SpectromoduleGetFrameSize() to get correct frame size from spectrometer.

3.4.7. UAI_SpectrometerDataOneshot

Declare :

UINT UAI_SpectrometerDataOneshot(void* api_handle, unsigned int integration_time_us, float *buffer, unsigned int average)

Visual Basic Declare : **_VB6UAI_SpectrometerOneshot@16**

Get the intensity array with smoothing function from specified spectrometer .

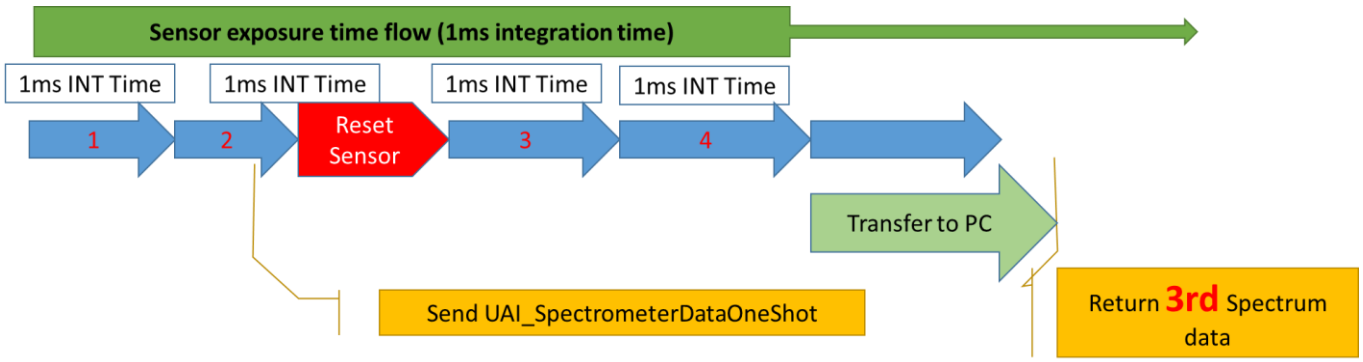
Parameter

Name	Description
api_handle	Specify the spectrometer handle.
integration_time_us	Set integration time(unit: us)
buffer	Return to the pointer of created 1D array buffer.(unit: counts)
average	Set average times of acquire data.

Introduction

Before using UAI_SpectrometerOneshot(), user have to create a 1D array buffer which have the same size as frame size. User could use UAI_SpectromoduleGetFrameSize() to get correct frame size from spectrometer.

The different between UAI_SpectrometerDataAcquire is spectrometer will erase buffer before data acquiring if user uses UAI_SpectrometerOneshot.



3.4.8. UAI_SpectrometerDataOneshots

Declare :

UINT UAI_SpectrometerDataOneshots(void* api_handle, unsigned int integration_time_us, float *buffer, unsigned int average)

Visual Basic Declare : **_VB6UAI_SpectrometerOneshots@16**

Get the intensity array from specified spectrometer .

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
integration_time_us	Set integration time(unit: us)
buffer	Return to the pointer of created 1D array buffer.(unit: counts)
average	Set average times of acquire data.

Introduction

Before using UAI_SpectrometerOneshots(), user have to create a 1D array buffer which have the same size as frame size. User could use UAI_SpectromoduleGetFrameSize() to get correct frame size from spectrometer.

The different between UAI_SpectrometerDataAcquires is spectrometer will erase buffer before data acquiring if user uses UAI_SpectrometerOneshots.

3.4.9. UAI_SpectrometerDataOneshotRaw

Declare :

UINT UAI_SpectrometerDataOneshotRaw(void* api_handle, unsigned int integration_time_us, float *buffer, unsigned int average)

Visual Basic Declare : **_VB6UAI_SpectrometerOneshotRaw@16**

Get full intensity array from specified spectrometer .

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
integration_time_us	Set integration time(unit: us)
buffer	Return to the pointer of created 1D array buffer.(unit: counts)
average	Set average times of acquire data.

Introduction

Before using UAI_SpectrometerOneshotRaw(), user have to create a 1D array buffer which have the same size as frame size. User could use UAI_SpectromoduleGetFrameSizeRaw() to get correct full frame size from spectrometer.

The different between UAI_SpectrometerDataAcquire is spectrometer will erase buffer before data acquiring if user uses UAI_SpectrometerOneshotRaw.

NOTE : Full intensity from UAI_SpectrometerOneshotRaw() can not be used in correction functions(chapter 3.6)

3.5. Trigger Mode

3.5.1. UAI_SpectrometerSetTriggerIO

Declare :

UINT UAI_SpectrometerSetTriggerIO(void* api_handle,unsigned int enable, unsigned int timeout,unsigned int level)

Visual Basic Declare :

_VB6UAI_SpectrometerSetTriggerIO@16

This function supports spectrometer getting triggered signal to acquire data from external pin.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
enable	1(Enable) , 0(Disable)
timeout	Set time out(ms)
level	Set trigger mode

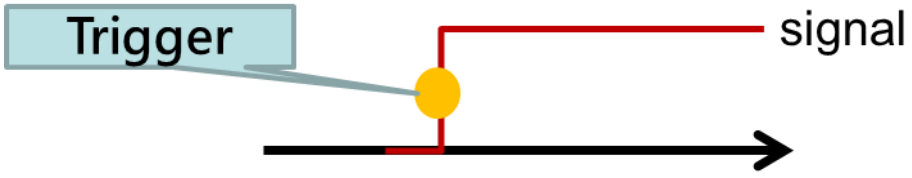
Introduction

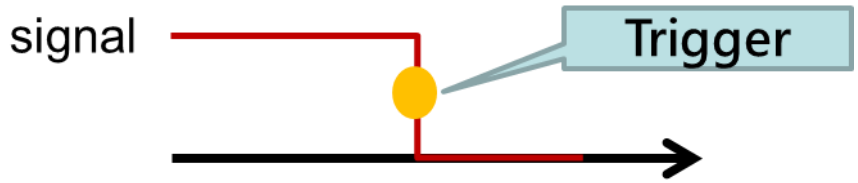
When Spectrameter is in Trigger mode , User can use external signal to ask spectrometer to acquire spectrum.

For example , if user want control multiple spectrometers and let spectrometers acquire data at the same time . Using API commands by software will be affected by performance of PC , but trigger mode makes sure that multiple spectrometer can execute at the same time.

User should use UAI_ SpectrometerTriggerDataAcquire to check the status of triggered signal.

Parameter “level” can set trigger edge by signal.

level	Item
1	

0	

3.5.2. UAI_SpectrometerGetTriggerIO

Declare :

UINT UAI_SpectrometerGetTriggerIO(void* api_handle,unsigned int* enable, unsigned int* timeout,unsigned int* level)

Visual Basic Declare :

_VB6UAI_SpectrometerGetTriggerIO@16

Get setting of TriggerIO.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
enable	Pointer of 1(Enable) , 0(Disable).
timeout	Pointer of time out value(ms).
level	Pointer of level value.

3.5.3. UAI_SpectrometerTriggerDataAcquire

Declare :

UINT UAI_SpectrometerTriggerDataAcquire(void* api_handle, float *buffer)

Visual Basic Declare :

_VB6UAI_SpectrometerTriggerDataAcquire@8

Check the status of triggered signal. This function will return intensity of light from spectrometer if it gets triggered signal.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
buffer	Return to the pointer of created 1D array buffer.

Introduction

Function will return value 8, if there isn't triggered signal during the period of time. User have to keep using the function to track the status. Function will return value 0, if there is triggered signal during the period of time. Buffer will be returned the pointer of light intensity.

3.5.4. UAI_SpectrometerGetTriggerGroupIntegrationTime

Declare :

UINT UAI_SpectrometerGetTriggerGroupIntegrationTime (void* api_handle, unsigned int * groupcount , unsigned int * integration_us)

Visual Basic Declare :

VB6UAI SpectrometerGetTriggerGroupIntegrationTime @12

Get information of group triggers.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
groupcount	Return to the pointer of count number of triggers.
integration_us	Return to the pointer of integration times of triggers(uints : us)

3.5.5. UAI_SpectrometerSetTriggerGroupIntegrationTime

Declare :

UINT UAI_SpectrometerSetTriggerGroupIntegrationTime (void* api_handle, unsigned int groupcount , unsigned int * integration_us)

Visual Basic Declare :

VB6UAI SpectrometerGetTriggerGroupIntegrationTime @12

Set information of group triggers.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
groupcount	Set count number of triggers.
integration_us	Set the pointer of integration times of triggers(uints : us)

3.5.6. UAI_SpectrometerCheckTriggerDone

Declare :

UINT UAI_ SpectrometerCheckTriggerDone(void* api_handle ,unsigned int * count)

Visual Basic Declare :

VB6UAI SpectrometerCheckTriggerDone@8

Get information of group triggers.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
count	Get count number how many trigger be detected.

Introduction

Parameter “count” shows how many hardware trigger that spectrometer has detected.

User can use errorcode to check the status of trigger mode.

Errorcode = 0 : Trigger mode is ready or status of group trigger is finished.

Errorcode = 9 : Trigger mode is not ready , spectrometer keep detecting hardware trigger.

Errorcode = 10 : Trigger mode is timeout , spectrometer needs to be reset trigger mode.

3.5.7. UAI_SpectrometerGetTriggerData

Declare :

UINT UAI_ SpectrometerGetTriggerData(void* api_handle, unsigned int framesize, unsigned int index, float *buffer)

Visual Basic Declare :

VB6UAI SpectrometerGetTriggerData@16

Get intensity of specific trigger.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
framesize	Set framesize of spectrum
index	Set index of spectrum
buffer	Return to the pointer of created 1D array buffer.

Introduction

Get specific index of spectrum in group trigger .

3.5.8. UAI_SpectrometerCheckDoneAndGetTriggerData

Declare :

UINT UAI_SpectrometerCheckDoneAndGetTriggerData(void* api_handle, unsigned int framesize, unsigned int index, float *buffer)

Visual Basic Declare :

VB6UAI SpectrometerCheckDoneAndGetTriggerData@16

Check and get intensity of specific trigger.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
framesize	Set framesize of spectrum
index	Set index of spectrum
buffer	Return to the pointer of created 1D array buffer.

Introduction

This function combine “UAI_SpectrometerCheckTriggerDone “ and “UAI_SpectrometerGetTriggerData”

Get specific index of spectrum in group trigger .

User can use errorcode to check the status of trigger mode.

Errorcode = 0 : Trigger mode is ready or status of group trigger is finished.

Errorcode = 9 : Trigger mode is not ready , spectrometer keep detecting hardware trigger.

Errorcode = 10 : Trigger mode is timeout , spectrometer needs to be reset trigger mode.

3.5.9. UAI_SpectrometerGetTriggerDelay

Declare :

UINT UAI_SpectrometerGetTriggerDelay (void* api_handle, unsigned int * integration_us)

Visual Basic Declare :

VB6UAI SpectrometerGetTriggerDelay@8

Get setting of delay time of trigger mode.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
integration_us	Get delay times of trigger(uints : us)

3.5.10.UAI_SpectrometerSetTriggerDelay

Declare :

UINT UAI_SpectrometerSetTriggerDelay (void* api_handle, unsigned int integration_us)

Visual Basic Declare :

VB6UAI SpectrometerSetTriggerDelay@8

Set setting of delay time of trigger mode.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
integration_us	Set delay times of trigger(uints : us)

3.5.11.UAI_SpectrometerSetBatchMode

Declare :

UINT UAI_SpectrometerSetBatchMode(void* api_handle, unsigned int count , unsigned int mode)

Visual Basic Declare :

VB6UAI SpectrometerSetBatchMode@12

Set into batch mode.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
count	Set acquired count in batch mode
mode	Set trigger mode in batch mode.

Introduction

Parameter “mode” used to define how to start acquiring spectrum in batch mode.

Mode = 0 : Start acquiring after command “UAI_SpectrometerSetBatchMode”.

Mode = 1: Start acquiring after detecting hardware trigger.

.

3.6. Spectrum Correction

3.6.1. UAI_BackgroundRemove

Declare :

UINT UAI_BackgroundRemove(void* api_handle, unsigned int integration_time, float* source)

Visual Basic Declare :

_VB6UAI_BackgroundRemove@12

Get the light intensity after electrical background removal.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
integration_time	Set integration time(unit: us)
source	Return to the pointer of light intensity(unit: counts) with background removal.

Introduction

The SDK supports electrical background removal function which could reduce noise of electrical system. API could receive the pointer of source of light intensity from user and return the pointer of light intensity with background removal. It will occurs abnormal error message or errorcode if user provide incorrect intensity data for API.

C# sample

Please refer sample code.

3.6.2. UAI_BackgroundRemoveWithAVG

Declare :

UINT UAI_BackgroundRemoveWothAVG(void* api_handle, unsigned int integration_time, float* source)

Visual Basic Declare :

_VB6UAI_BackgroundRemoveWithAVG@12

Get the light intensity after electrical background removal.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
integration_time	Set integration time(unit: us)
source	Return to the pointer of light intensity(unit: counts) with background removal.

Introduction

The SDK supports electrical background removal function which could reduce noise of electrical system. API could receive the pointer of source of light intensity from user and return the pointer of light intensity with background removal. It will occurs abnormal error message or errorcode if user provide incorrect intensity data for API.

The difference between UAI_BackgroundRemove and UAI_BackgroundRemoveWothAVG is that UAI_BackgroundRemoveWothAVG removes electrical background by averaging with 15 previous backgrounds.

3.6.3. UAI_LinearityCorrection

Declare :

UINT UAI_LinearityCorrection(void* handle , insighed int framesize , float* source)

Visual Basic Declare :

_VB6UAI_LinearityCorrection@12

Get light intensity after linearity correction.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
framesize	Set framesize
source	Return to the pointer of light intensity(unit: counts) with linearity correction.

Introduction

The SDK supports linearity correction function which could correction linearity issue from sensor . API could receive the pointer of source of light intensity from user and return the pointer of light intensity with linearity correction. It will occurs abnormal error message or errorcode if user provide incorrect intensity data for API.

C# sample

Please refer sample code.

3.6.4. UAI_AbsoluteIntensityCorrection

Declare :

UINT UAI_AbsoluteIntensityCorrection (void* api_handle, float* source, unsigned int integration_time)

Visual Basic Declare :

_VB6UAI_AbsoluteIntensityCorrection@12

Get the light intensity after absolute intensity correction.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
source	Return to the pointer of light intensity(unit: uW) with absolute intensity correction.
integration_time	Set integration time(unit: us)

Introduction

The SDK supports absolute intensity function which could correct absolute intensity to light unit(uW) of each pixel with standard light source. API could receive the pointer of source of light intensity from user and return the pointer of light intensity with absolute intensity correction. It will occurs abnormal error message or errorcode if user provide incorrect intensity data for API.

C# sample

Please refer sample code.

3.6.5. UAI_ContrastIntensityCorrection

Declare :

UINT UAI_ContrastIntensityCorrection(void* api_handle, float* source)

Visual Basic Declare :

_VB6UAI_ContrastIntensityCorrection@8

Get the light intensity with contrast intensity correction.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
source	Return to the pointer of light intensity(unit: counts) with contrast intensity correction.

Introduction

The SDK supports contrast intensity function which could correct intensity of each pixel with standard light source. API could receive the pointer of source of light intensity from user and return the pointer of light intensity with contract intensity correction. It will occurs abnormal error message or errorcode if user provide incorrect intensity data for API.

3.6.6. UAI_StrayLightCorrection

Declare :

UINT UAI_StrayLightCorrection(void* api_handle, float* source)

Visual Basic Declare :

_VB6UAI_StrayLightCorrection@8

Get the light intensity with straylight correction. This Function is only supported stray light table in spectrometer(UM series) . If user have CSV file of straylight table , please refer "UAI_SpectrometerSetStraylightCalibrationInformationF" and "UAI_SpectrometerStraylightCalibration"

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
source	Return to the pointer of light intensity(unit: counts) with straylightcorrection.

Introduction

The SDK supports straylight function which could correct intensity of each pixel. API could receive the pointer of source of light intensity from user and return the pointer of light intensity with straylight correction. It will occurs abnormal error message or errorcode if user provide incorrect intensity data for API.

3.6.7. UAI_DoIntensityCalibration

Declare :

UINT UAI_DoIntensityCalibration(void* api_handle, unsigned int std_size, float *std_lambda, float *std_intensity, float *m_intensity, unsigned int integration_time, unsigned short date)

Visual Basic Declare :

_VB6UAI_DoIntensityCalibration@28

This function supports system developer and user doing intensity calibration if system developer or user have to re-create intensity calibration table in measurement system .

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
std_size	The array length of standard light data.
std_lambda	The point of wavelength array of standard light.
std_intensity	The point of intensity array of standard light.
m_intensity	The point of intensity array of standard light which is measured by device.
Integration_time	Integration time of doing intensity calibration.
Date	Calibration date.

Introduction

The calibration range of intensity calibration depends on the range of standard light .

EX : If user provides wavelength from 390nm ~ 900nm . This function will do calculate gain table only between 390nm~900nm . Others will be 0 .

Std_lambda and Std_Intensity should get from provider who makes standard light source or measure by standard meter.

The format of date is 16 bits array . Index 0~6 means year . Index 7~10 means month . Index 11~15 means day . Year should minus 2010 .

EX: 2020/12/19 .

Format of year is 0001010. Binary from 10(2020-2010).

Format of Month is 1100. Binary from 12.

Format of Day is 10011.Binary from 19.

Format of date is 0001010110010011.

Y value means different absolute intensity , it's based on unit of standard light(std_intensity) in intensity calibration.

Unit of standard light	Unit of Y
uWatt	lumen
uWatt/m^2	Lux
uWatt/m^2.sr	Nit

3.6.8. UAI_SpectromoduleSetIntensityCalibration

Declare :

UINT UAI_SpectromoduleSetIntensityCalibration(void* api_handle, double *gain, unsigned short date, unsigned int integration_time)

Visual Basic Declare :

_VB6UAI_SpectromoduleSetIntensityCalibration@16

This function supports setting the gain table of intensity calibration from specified device .

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
gain	Provide calibration gain table . The array size = framesize of device.
date	Provide new calibration date.
Integration_time	Provide Integration time of doing intensity calibration.

Introduction

The format of date is same as function (UAI_DoIntensityCalibration) . Please refer the introduction of the function .

3.6.9. UAI_SpectromoduleGetIntensityCalibration

Declare :

UINT UAI_SpectromoduleGetIntensityCalibration(void* api_handle, double *gain, unsigned short *date, unsigned int *integration_time)

Visual Basic Declare :

_VB6UAI_SpectromoduleGetIntensityCalibration@16

This function supports getting the gain table of intensity calibration from specified device.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
gain	Return calibration gain table . The array size = framesize of device.
date	Return Calibration date.
Integration_time	Return Integration time of doing intensity calibration.

Introduction

The format of date is same as function (UAI_DoIntensityCalibration) . Please refer the introduction of the function.

3.6.10.UAI_SpectrometerSetStraylightCalibrationInformationF

Declare :

UINT UAI_SpectromoduleSetStraylightCalibrationInformationF(void* api_handle, char* fullfilename)

Visual Basic Declare :

VB6UAI SpectromoduleSetStraylightCalibrationInformationF@8

This function supports correcting straylight by existed file (CSV file).

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
fullfilename	String of straylight filename with full path.

Introduction

The SDK supports straylight function which could correct intensity of each pixel. API could receive the pointer of source of light intensity from user and return the pointer of light intensity with straylight correction. It will occurs abnormal error message or errorcode if user provide incorrect straylight file or intensity data for API.

3.6.11.UAI_SpectrometerStraylightCalibration

Declare :

UINT UAI_SpectromoduleStraylightCalibration (void* api_handle, unsigned int frame_size , float* source)

Visual Basic Declare :

VB6UAI SpectromoduleStraylightCalibration @12

Before using this function , user has to use

“UAI_SpectromoduleSetStraylightCalibrationInformationF: to load table from file.

This function supports correcting straylight by existed file (CSV file).

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
framesize	Value of framesize that get from device.
source	String of straylight filename with full path.

Introduction

The SDK supports straylight function which could correct intensity of each pixel. API could receive the pointer of source of light intensity from user and return the pointer of light intensity with straylight correction. It will occurs abnormal error message or errorcode if user provide incorrect straylight file or intensity data for API.

3.7. Color Measurement

3.7.1. UAI_ColorInformationAllocation

Declare :

UINT UAI_ColorInformationAllocation(void* api_handle, void **color, unsigned int type, unsigned int observer, unsigned int illuminant, float* Lambda, float* intensity_r, float* intensity_m, unsigned int size)

Visual Basic Declare :

_VB6UAI_ColorInformationAllocation@36

Set necessary parameters and data buffer of wavelength & light intensity.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
color	Return to the handle of color information.
type	Type of the color measurement. 0:Relative emission. 1:Relative reflection. 2:Absolute emission.
observer	Type of the observer degree. 0: 2 degree. 1: 10 degree.
illuminant	The CIE standard illuminant. 0:A light 1:B light 2:C light 3:D50 light 4:D55 light 5:D65 light 6:D75 light 7:E light 8:F1 light 9:F2 light 10:F3 light 11:F4 light

	12:F5 light 13:F6 light 14:F7 light 15:F8 light 16:F9 light 17:F10 light 18:F11 light 19:F12 light
Lambda	Pointer of wavelength data buffer. This parameter can't be NULL.
intensity_r	Pointer of reference intensity data buffer. If it's NULL, function will allocate one.
intensity_m	Pointer of measured intensity data buffer. If it's NULL, function will allocate one.
size	The size of the wavelength buffer.

Introduction

Before getting color information, user have to create a color information handle by UAI_ColorInformationAllocation . The color information handle will be related with the assigned spectrometer, all color API's that get color information from this spectrometer have to base on this handle to work properly in the system with multiple spectrometer installed.

User also have to set related parameter as observer . illuminant and data buffer as wavelength , reference intensity and measured intensity.

If type is "Relative emission" or "Relative reflection", user have to provide reference intensity and measurement intensity before doing UAI_ColorOperation(). User could use UAI_ColorInformationSet() to set intensity before this API .

If type is "absolute emission", reference intensity can be null.

C# sample

Please refer sample code.

3.7.2. UAI_ColorOperation

Declare :

UINT UAI_ColorOperation(void* color)

Visual Basic Declare :

_VB6UAI_ColorOperation@4

Parameter

Name	Description
color	Specified handle of color information.

Introduction

Before using UAI_ColorOperation(), user have to use UAI_ColorInformationAllocation() or UAI_ColorInformationSet() to set relate parameter and data buffer. These API will start to calculate all color information.

3.7.3. UAI_ColorInformationFree

Declare :

UINT UAI_ColorInformationFree(void* color)

Visual Basic Declare :

_VB6UAI_ColorInformationFree@4

Release color information handle.

Parameter

Name	Description
color	Specified handle of color information.

Introduction

Once the thread or process use UAI_ColorInformationFree(), the color information handle will be released.

3.7.4. UAI_ColorGetXYZ

Declare :

UINT UAI_ColorGetXYZ(void* color, double* XYZ)

Visual Basic Declare :

_VB6UAI_ColorGetXYZ@8

Get XYZ Color information.

$$X = \int_{380}^{780} I(\lambda)\,\overline{x}(\lambda)\,d\lambda$$

$$Y = \int_{380}^{780} I(\lambda)\,\overline{y}(\lambda)\,d\lambda$$

$$Z = \int_{380}^{780} I(\lambda)\,\overline{z}(\lambda)\,d\lambda$$

Parameter

Name	Description
color	Specified handle of color information.
XYZ	Return to the pointer of XYZ data buffer. Length of data : 3 double

Introduction

Before using UAI_ColorGetXYZ(), user have to use UAI_ColorOperation() to calculate color information. User have to create a 1D array buffer(3 double space) for this API.

Index of array	Item
0	X
1	Y
2	Z

Y value means different absolute intensity , it's based on unit of standard light in intensity calibration.

Unit of standard light	Unit of Y
uWatt	lumen
uWatt/m^2	Lux
uWatt/m^2.sr	Nit

3.7.5. UAI_ColorGetXYZRef

Declare :

UINT UAI_ColorGetXYZRef(void* color, double* XYZ)

Visual Basic Declare :

_VB6UAI_ColorGetXYZRef@8

Get XYZ Color information of Reference spectrum.

Parameter

Name	Description
color	Specified handle of color information.
XYZ	Return to the pointer of XYZ data buffer. Length of data : 3 double

Introduction

Before using UAI_ColorGetXYZRef(), user have to use UAI_ColorOperation() to calculate color information. User have to create a 1D array buffer(3 double space) for this API.

Index of array	Item
0	X
1	Y
2	Z

3.7.6. UAI_ColorGetxyz

Declare :

UINT UAI_ColorGetxyz(void* color, double* xyz)

Visual Basic Declare :

_VB6UAI_ColorGetxyz@8

Get CIE xyz color information.

$$x = \frac{X}{X+Y+Z}$$

$$y = \frac{Y}{X+Y+Z}$$

$$z = \frac{Z}{X+Y+Z} = 1 - x - y$$

Parameter

Name	Description
color	Specified handle of color information.
xyz	Return to the pointer of xyz data buffer. Length of data : 3 double

Introduction

Before using UAI_ColorGetxyz(), user have to use UAI_ColorOperation() to calculate color information. User have to create a 1D array buffer(3 double space) for this API.

Index of array	Item
0	x
1	y
2	z

3.7.7. UAI_ColorGetxyzRef

Declare :

UINT UAI_ColorGetxyzRef(void* color, double* xyz)

Visual Basic Declare :

_VB6UAI_ColorGetxyzRef@8

Get xyz Color information of Reference spectrum.

Parameter

Name	Description
color	Specified handle of color information.
xyz	Return to the pointer of xyz data buffer. Length of data : 3 double

Introduction

Before using UAI_ColorGetxyzRef(), user have to use UAI_ColorOperation() to calculate color information. User have to create a 1D array buffer(3 double space) for this API.

Index of array	Item
0	X
1	Y
2	Z

3.7.8. UAI_ColorGet1960UCS

Declare :

UINT UAI_ColorGet1960UCS(void* color, double* UVW)

Visual Basic Declare :

_VB6UAI_ColorGet1960UCS@8

Get 1960 UCS color information.

$$U = \frac{2}{3}X$$

$$V = Y$$

$$W = \frac{1}{2}(-X + 3Y + Z)$$

Parameter

Name	Description
color	Specified handle of color information.
UVW	Return to the pointer of UVW data buffer. Length of data : 3 double

Introduction

Before using UAI_ColorGet1960UCS(), user have to use UAI_ColorOperation() to calculate color information. User have to create a 1D array buffer(3 double space) for this API.

Index of array	Item
0	U
1	V
2	W

3.7.9. UAI_ColorGet1960ucs

Declare :

UINT UAI_ColorGet1960ucs (void* color, double* uvw)

Visual Basic Declare :

_VB6UAI_ColorGet1960ucs@8

Get 1960 ucs color information.

$$u = \frac{4x}{12y - 2x + 3}$$

$$v = \frac{6y}{12y - 2x + 3}$$

Parameter

Name	Description
color	Specified handle of color information.
uvw	Return to the pointer of uvw data buffer. Length of data : 3 double

Introduction

Before using UAI_ColorGet1960ucs(), user have to use UAI_ColorOperation() to calculate color information. User have to create a 1D array buffer(3 double space) for this API.

Index of array	Item
0	u
1	V
2	w

3.7.10.UAI_ColorGet1976UCS

Declare :

UINT UAI_ColorGet1976UCS(void* color, double* UVW)

Visual Basic Declare :

_VB6UAI_ColorGet1976UCS@8

Get 1976 UCS color information.

Parameter

Name	Description
color	Specified handle of color information.
UVW	Return to the pointer of UVW data buffer. Length of data : 3 double

Introduction

Before using UAI_ColorGet1976UCS(), user have to use UAI_ColorOperation() to calculate color information. User have to create a 1D array buffer(3 double space) for this API.

Index of array	Item
0	U
1	V
2	W

3.7.11.UAI_ColorGet1976ucs

Declare :

UINT UAI_ColorGet1976ucs (void* color, double* uvw)

Visual Basic Declare :

_VB6UAI_ColorGet1976ucs@8

Get 1976 ucs color information.

$$u' = \frac{4X}{X + 15Y + 3Z} = \frac{4x}{-2x + 12y + 3}$$
$$v' = \frac{9Y}{X + 15Y + 3Z} = \frac{9y}{-2x + 12y + 3}$$

Parameter

Name	Description
color	Specified handle of color information.
uvw	Return to the pointer of uvw data buffer. Length of data : 3 double

Introduction

Before using UAI_ColorGet1976 ucs(), user have to use UAI_ColorOperation() to calculate color information. User have to create a 1D array buffer(3 double space) for this API.

Index of array	Item
0	u
1	V
2	w

3.7.12.UAI_ColorGetCCT

Declare :

UINT UAI_ColorGetCCT (void* color, double* cct)

Visual Basic Declare :

_VB6UAI_ColorGetCCT@8

Get CCT(Correlated Color Temperature) color information.

Parameter

Name	Description
color	Specified handle of color information.
cct	Return to the pointer of CCT.

Introduction

Before using UAI_ColorGetCCT(), user have to use UAI_ColorOperation() to calculate color information.

3.7.13.UAI_ColorGetCIETint

Declare :

UINT UAI_ColorGetCIETint (void* color, double* Tcie)

Visual Basic Declare :

_VB6UAI_ColorGetCIETint@8

Get CIE Tint information.

Parameter

Name	Description
color	Specified handle of color information.
Tcie	Return to the pointer of Tcie.

Introduction

Before using UAI_ColorGetCIETint(), user have to use UAI_ColorOperation() to calculate color information.

3.7.14.UAI_ColorGetCIEWhiteness

Declare :

UINT UAI_ColorGetCIEWhiteness (void* color, double* Wcie)

Visual Basic Declare :

_VB6UAI_ColorGetCIEWhiteness@8

Get color whiteness information.

Parameter

Name	Description
color	Specified handle of color information.
Wcie	Return to the pointer of Wcie.

Introduction

Before using UAI_ColorGetCIEWhiteness(), user have to use UAI_ColorOperation() to calculate color information.

3.7.15.UAI_ColorGetColorRenderingIndex

Declare :

UINT UAI_ColorGetColorRenderingIndex (void* color, double* cri, double CCT)

Visual Basic Declare :

_VB6UAI_ColorGetColorRenderingIndex@16

Get Color Rendering Index(CRI) information.

Parameter

Name	Description
color	Specified handle of color information.
Cri	Return to the pointer of cri data buffer. Length of data : 17 double
CCT	Specified CCT value to calculate CCT.

Introduction

Before using UAI_ColorGetColorRenderingIndex(), user have to use UAI_ColorOperation() to calculate color information. User have to create a 1D array buffer(17 double space) for this API.

Index of array	Item
0	CRI_R1
1	CRI_R2
2	CRI_R3
3	CRI_R4
4	CRI_R5
5	CRI_R6
6	CRI_R7
7	CRI_R8
8	CRI_R9
9	CRI_R10
10	CRI_R11
11	CRI_R12
12	CRI_R13
13	CRI_R14
14	CRI_R15
15	CRI_Ra
16	CRI_DC

3.7.16.UAI_ColorGetColorQualityScale

Declare :

UINT UAI_ColorGetColorQualityScale (void* color, double* cqs)

Visual Basic Declare :

_VB6UAI_ColorGetColorQualityScale@8

Get Color Quality Scale(CQS) information.

Parameter

Name	Description
color	Specified handle of color information.
cqs	Return to the pointer of cqs data buffer. Length of data : 16 double

Introduction

Before using UAI_ColorGetColorQualityScale(), user have to use UAI_ColorOperation() to calculate color information. User have to create a 1D array buffer(16 double space) for this API.

Index of array	Item
0	CQS_1
1	CQS_2
2	CQS_3
3	CQS_4
4	CQS_5
5	CQS_6
6	CQS_7
7	CQS_8
8	CQS_9
9	CQS_10
10	CQS_11
11	CQS_12
12	CQS_13
13	CQS_14
14	CQS_15
15	CQS_Qa

3.7.17.UAI_ColorGetDominantWavelength

Declare :

UINT UAI_ColorGetDominantWavelength (void* color, double* Lambda_d)

Visual Basic Declare :

_VB6UAI_ColorGetDominantWavelength@8

Get dominant wavelength.

Parameter

Name	Description
color	Specified handle of color information.
Lambda_d	Return to the pointer of dominant wavelength.

Introduction

Before using UAI_ColorGetDominantWavelength(), user have to use UAI_ColorOperation() to calculate color information.

3.7.18.UAI_ColorGetHunterLab

Declare :

UINT UAI_ColorGetHunterLab(void* color, double* HLab)

Visual Basic Declare :

_VB6UAI_ColorGetHunterLab@8

Get CIE Hunter Lab color information.

Parameter

Name	Description
color	Specified handle of color information.
HLab	Return to the pointer of Hunter_Lab data buffer. Length of data : 3 double

3.7.19.UAI_ColorGetDuv

Declare :

UINT UAI_ColorGetDuv (double x , double y , double* duv)

Visual Basic Declare :

_VB6UAI_ColorGetDuv@20

Get delta uv by CIE x , y value.

Parameter

Name	Description
x	Input CIE x value.
Y	Input CIE y value.
duv	Return to the pointer of duv data buffer. Length of data : 3 double

3.7.20.UAI_ColorGetLab

Declare :

UINT UAI_ColorGetLab(void* color, double* Lab)

Visual Basic Declare :

_VB6UAI_ColorGetLab@8

Get CIE Lab color information.

Parameter

Name	Description
color	Specified handle of color information.
Lab	Return to the pointer of Lab data buffer. Length of data : 3 double

Introduction

Before using UAI_ColorGetLab(), user have to use UAI_ColorOperation() to calculate color information. User have to create a 1D array buffer(3 double space) for this API.

Index of array	Item
0	L
1	a
2	b

3.7.21.UAI_ColorGetLuv

Declare :

UINT UAI_ColorGetLuv (void* color, double* Luv)

Visual Basic Declare :

_VB6UAI_ColorGetLuv@8

Get CIE Luv color information.

$$L^* = \begin{cases} \left(\frac{29}{3}\right)^3 Y/Y_n, & Y/Y_n \le \left(\frac{6}{29}\right)^3 \\ 116\,(Y/Y_n)^{1/3} - 16, & Y/Y_n > \left(\frac{6}{29}\right)^3 \end{cases}$$

$$u^* = 13L^* \cdot (u' - u'_n)$$

$$v^* = 13L^* \cdot (v' - v'_n)$$

Parameter

Name	Description
color	Specified handle of color information.
Luv	Return to the pointer of Luv data buffer. Length of data : 3 double

Introduction

Before using UAI_ColorGetLuv(), user have to use UAI_ColorOperation() to calculate color information. User have to create a 1D array buffer(3 double space) for this API.

Index of array	Item
0	L
1	u
2	v

3.7.22.UAI_ColorGetPurity

Declare :

UINT UAI_ColorGetPurity (void* color, double* purity_e)

Visual Basic Declare :

_VB6UAI_ColorGetPurity@8

Get purity information.

Parameter

Name	Description
color	Specified handle of color information.
Purity_e	Return to the pointer of purity.

Introduction

Before using UAI_ColorGetPurity(), user have to use UAI_ColorOperation() to calculate color information.

3.7.23.UAI_ColorGetRadiantPower

Declare :

UINT UAI_ColorGetRadiantPower(void* color, double*RadiantPower)

Visual Basic Declare :

_VB6UAI_ColorGetRadiantPower@8

Get Radiant Power information

Parameter

Name	Description
color	Specified handle of color information.
RadiantPower	Return to the pointer of RadiantPower.

Introduction

Before using UAI_ColorGetRadiantPower(), user have to use UAI_ColorOperation() to calculate color information.

3.7.24.UAI_ColorGetUVW

Declare :

UINT UAI_ColorGetUVW (void* color, double*UVW)

Visual Basic Declare :

_VB6UAI_ColorGetUVW@8

Get CIE 1964 UVW color information.

Parameter

Name	Description
color	Specified handle of color information.
UVW	Return to the pointer of UVW data buffer. Length of data : 3 double

Introduction

Before using UAI_ColorGetUVW(), user have to use UAI_ColorOperation() to calculate color information. User have to create a 1D array buffer(3 double space) for this API.

Index of array	Item
0	U
1	V
2	W

3.7.25.UAI_ColorGetuvw

Declare :

UINT UAI_ColorGetuvw (void* color, double* uvw)

Visual Basic Declare :

_VB6UAI_ColorGetuvw@8

Get CIE 1964 uvw color information.

Parameter

Name	Description
color	Specified handle of color information.
uvw	Return to the pointer of uvw data buffer. Length of data : 3 double

Introduction

Before using UAI_ColorGet uvw (), user have to use UAI_ColorOperation() to calculate color information. User have to create a 1D array buffer(3 double space) for this API.

Index of array	Item
0	u
1	v
2	w

3.8. Ring Buffer

3.8.1. UAI_SetExtIntTimeMode

Declare :

UINT UAI_SetExtIntTimeMode(void* api_handle, unsigned int enable)

Visual Basic Declare :

VB6UAI SetExtIntTimeMode@8

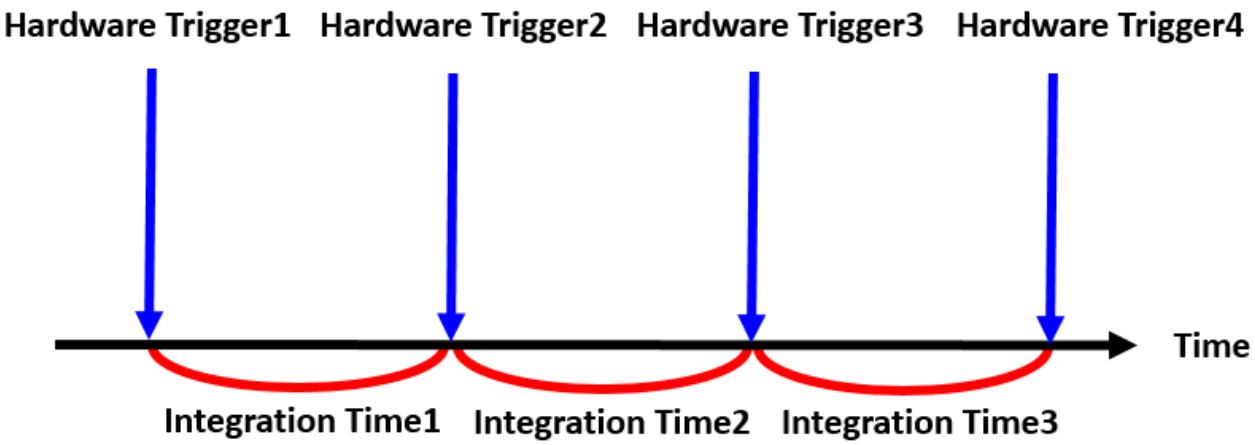
In external Trigger mode, integration time can be decided by external trigger signal.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
enable	Type of the enable 0: Turn off. 1: Turn on.

Introduction

When Spectrameter is in external Trigger mode , User can use external signal to get integration time for spectrum.



3.8.2. UAI_SpectrometerBlockLengthSet

Declare :

UINT UAI_SpectrometerBlockLengthSet(void* api_handle, unsigned int framenumber)

Visual Basic Declare :

_VB6UAI_SpectrometerBlockLengthSet@8

Set the frame number of total spectrums.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
framenumber	Set the spectrum frame number (Power of two).

3.8.3. UAI_SpectrometerBlockModeStartStop

Declare :

UINT UAI_SpectrometerBlockModeStartStop(void* api_handle, unsigned int enable)

Visual Basic Declare :

_VB6UAI_SpectrometerBlockModeStartStop@8

In Block mode, spectrometer enable or disable continuously get spectrum.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
enable	Type of the enable 0: Turn off. 1: Turn on.

3.8.4. UAI_SpectrometerBlockDataAcquire

Declare :

UINT UAI_SpectrometerBlockDataAcquire(void* api_handle, unsigned int integration_time_us, unsigned int frame_size, unsigned int framenumber , float *buffer)

Visual Basic Declare :

_VB6UAI_SpectrometerBlockDataAcquire@20

Get the intensity array from specified spectrometer .

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
integration_time_us	Set integration time(unit: us)
frame_size	Set the frame size of spectrum
framenumber	Set the spectrum frame number (Power of two).
buffer	Return to the pointer of created 1D array Buffer.(unit: counts)

3.9. TEC

3.9.1. UAI_SpectrometerSetTECOnOff

Declare :

UINT UAI_SpectrometerSetTECOnOff(void* api_handle, unsigned int onoff)

Visual Basic Declare :

_VB6UAI_SpectrometerSetTECOnOff@8

Enable/Disable TEC function of specified spectrometer .

Note : Only OTO TEC series spectrometer(EagleEye series) supports this function.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
onoff	Value 0 : disable TEC Value 1 : Enable TEC

3.9.2. UAI_SpectrometerGetTECOnOff

Declare :

UINT UAI_SpectrometerGetTECOnOff(void* api_handle, unsigned* int onoff)

Visual Basic Declare :

_VB6UAI_SpectrometerGetTECOnOff@8

Get status(Enable or Disable) of TEC function of specified spectrometer .

Note : Only OTO TEC series spectrometer(EagleEye series) supports this function.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
onoff	Value 0 : disable TEC Value 1 : Enable TEC

3.9.3. UAI_SpectrometerSetTECFansOnOff

Declare :

UINT UAI_SpectrometerSetTECFansOnOff(void* api_handle, unsigned int onoff)

Visual Basic Declare :

_VB6UAI_SpectrometerSetTECFansOnOff@8

Enable/Disable TEC's Fan of specified spectrometer .

Note : Only OTO TEC series spectrometer(EagleEye series) supports this function.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
onoff	Value 0 : disable TEC Value 1 : Enable TEC

3.9.4. UAI_SpectrometerGetTECFansOnOff

Declare :

UINT UAI_SpectrometerGetTECFansOnOff(void* api_handle, unsigned* int onoff)

Visual Basic Declare :

_VB6UAI_SpectrometerGetTECFansOnOff@8

Enable/Disable TEC's Fan of specified spectrometer .

Note : Only OTO TEC series spectrometer(EagleEye series) supports this function.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
onoff	Value 0 : disable TEC Value 1 : Enable TEC

3.9.5. UAI_SpectrometerSetTECTargetTemperature

Declare :

UINT UAI_SpectrometerSetTECTargetTemperature (void* api_handle, float degC)

Visual Basic Declare :

_VB6UAI_SpectrometerSetTECTargetTemperature@8

Set TEC's temperature value of specified spectrometer to control temperature of TEC.

Note : Only OTO TEC series spectrometer(EagleEye series) supports this function.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
degC	Set value of Temperature.

3.9.6. UAI_SpectrometerGetTECTargetTemperature

Declare :

UINT UAI_SpectrometerGetTECTargetTemperature (void* api_handle, float* degC)

Visual Basic Declare :

_VB6UAI_SpectrometerGet TECTargetTemperature @8

Get TEC's temperature value of specified spectrometer.

Note : Only OTO TEC series spectrometer(EagleEye series) supports this function.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
degC	Get value of Temperature.

3.9.7. UAI_SpectrometerGetTECTemperature

Declare :

UINT UAI_SpectrometerGetTECTemperature(void* api_handle, float* degC)

Visual Basic Declare :

_VB6UAI_SpectrometerGetTECTemperature@8

Get TEC sensor temperature of specified spectrometer.

Note : Only OTO TEC series spectrometer(EagleEye series) supports this function.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
degC	Get temperature of TEC sensor

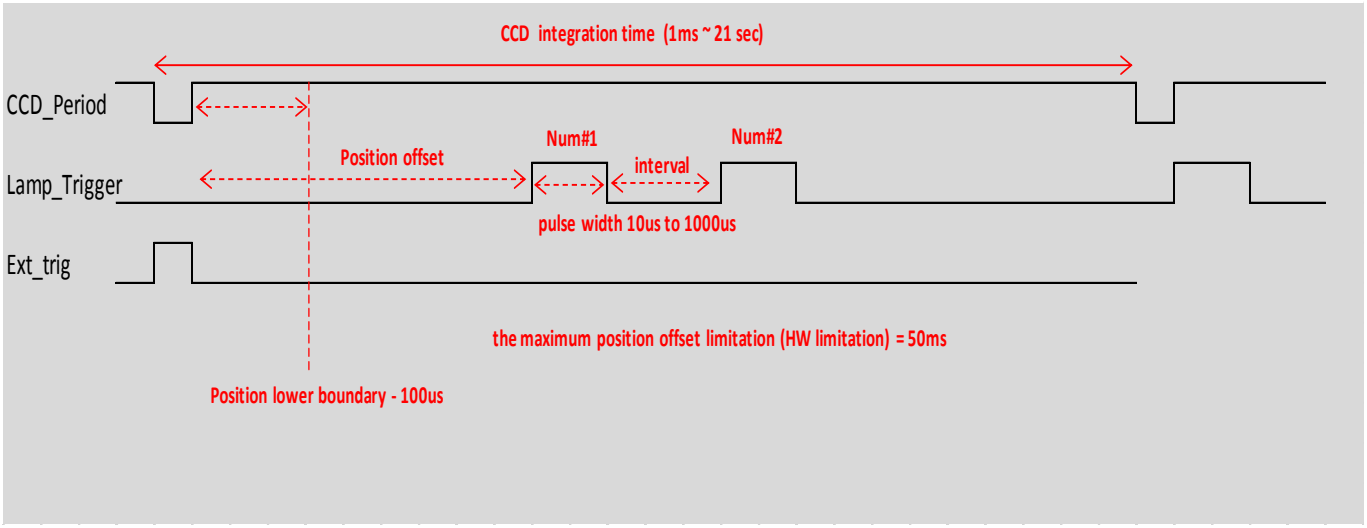
3.10. Pulse Setting

Ping LS_ON will send pulse signal for application.
Definition of LS_ON for each series spectrometer , please check chapter 5.

Pulse setting parameter definition
Pulse position(delay) : 100us~50ms
Pulse width : 10us~1000us
Pulse number : under integration period limited
Pulse interval : 10us~50ms

Acquiring period (integration time) >= position + (width + interval) * pulse number.

Pulse number = (integration time – position) / (pulse width+ pulse interval)
If pulse number is over available value , driver will automatically cut the number which is over above rule.



3.10.1. UAI_SpectrometerSetXenonPulseDelay

Declare :

UINT UAI_SpectrometerSetXenonPulseDelay(void* api_handle, unsigned int time_us)

Visual Basic Declare :

_VB6UAI_SpectrometerSetXenonPulseDelay@8

Set delay time for 1st pulse for each sensor's exposure.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
time_us	Set delay time value of 1st pulse

3.10.2. UAI_SpectrometerGetXenonPulseDelay

Declare :

UINT UAI_SpectrometerGetXenonPulseDelay(void* api_handle, unsigned int* time_us)

Visual Basic Declare :

_VB6UAI_SpectrometerGetXenonPulseDelay@8

Get delay time for 1st pulse for each sensor's exposure.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
time_us	Get delay time value of 1st pulse

3.10.3. UAI_SpectrometerSetXenonPulseNumber

Declare :

UINT UAI_SpectrometerSetXenonPulseNumber(void* api_handle, unsigned int number)

Visual Basic Declare :

_VB6UAI_SpectrometerSetXenonPulseNumber@8

Set pulse number during each sensor's exposure.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
number	Set pulse number.

3.10.4. UAI_SpectrometerGetXenonPulseNumber

Declare :

UINT UAI_SpectrometerGetXenonPulseNumber(void* api_handle, unsigned int* number)

Visual Basic Declare :

_VB6UAI_SpectrometerGetXenonPulseNumber@8

Get pulse number during each sensor’s exposure.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
number	Get pulse number.

3.10.5. UAI_SpectrometerSetXenonPulseInterval

Declare :

UINT UAI_SpectrometerSetXenonPulseInterval(void* api_handle, unsigned int time_us)

Visual Basic Declare :

_VB6UAI_SpectrometerSetXenonPulseInterval@8

Set pulse interval between each pulse.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
Time_us	Set pulse interval.

3.10.6. UAI_SpectrometerGetXenonPulseInterval

Declare :

UINT UAI_SpectrometerGetXenonPulseInterval(void* api_handle, unsigned int* time_us)

Visual Basic Declare :

_VB6UAI_SpectrometerGetXenonPulseInterval@8

Get pulse interval between each pulse.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
Time_us	Get pulse interval.

3.10.7. UAI_SpectrometerSetXenonPulseWidth

Declare :

UINT UAI_SpectrometerSetXenonPulseWidth(void* api_handle, unsigned int time_us)

Visual Basic Declare :

_VB6UAI_SpectrometerSetXenonPulseWidth@8

Set pulse width

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
Time_us	Set pulse width.

3.10.8. UAI_SpectrometerGetXenonPulseWidth

Declare :

UINT UAI_SpectrometerGetXenonPulseWidth(void* api_handle, unsigned int* time_us)

Visual Basic Declare :

_VB6UAI_SpectrometerGetXenonPulseWidth@8

Get pulse width

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
Time_us	Get pulse width.

3.10.9. UAI_SpectrometerSetXenonMode

Declare :

UINT UAI_SpectrometerSetXenonMode(void* api_handle, unsigned int mode)

Visual Basic Declare :

_VB6UAI_SpectrometerSetXenonMode@8

Set pulse or xenon mode

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
mode	Set mode.

Pulse Mode Select :

Mode = 0 : Disable pulse(xenon) Control(default).

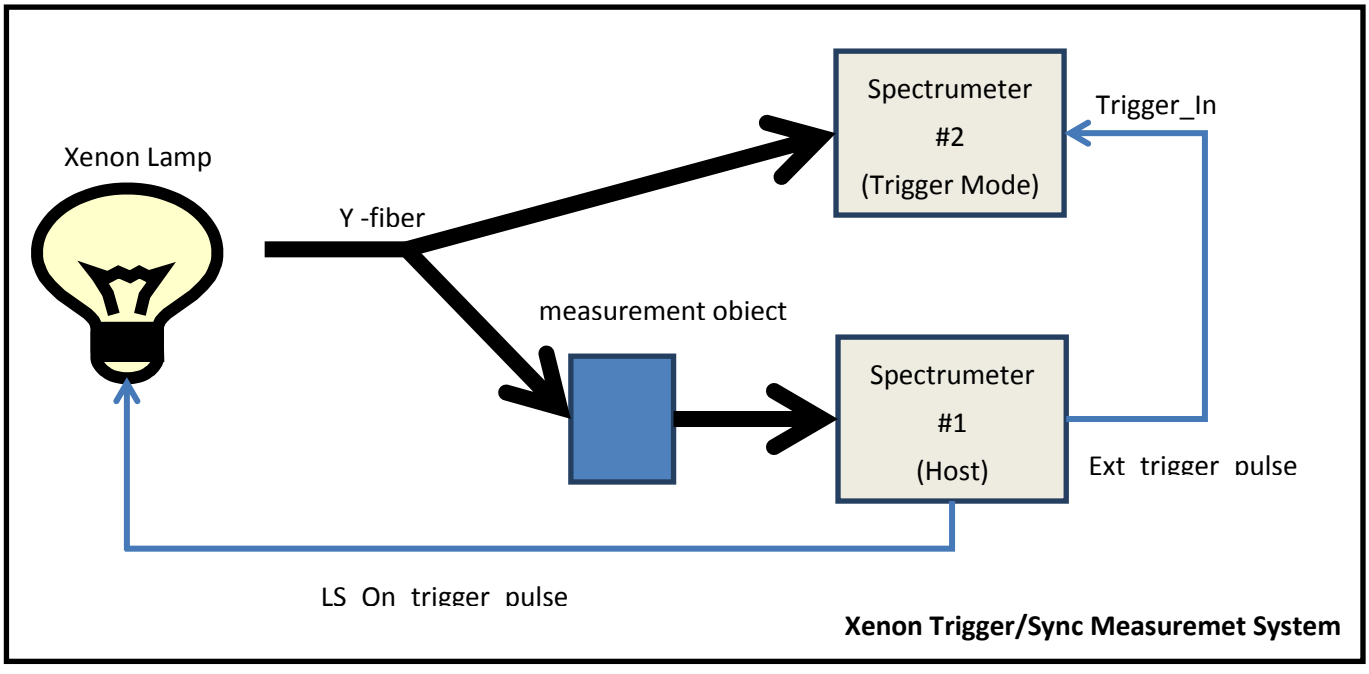
LS_ON pin will trun On(high)/Off(Low) by switching from SW. (light on/off)

Mode = 1 : Normal mode.

LS_ON pin will send pulses that synchronize to acquire period when SW issued light on.

Mode = 2 : Advance mode.

the GPIO next to LS_ON pin will send an external pulse every acquire period while light on. This is an advance application of Master/Slave control of Xenon light measurement.



3.10.10. UAI_SpectrometerGetXenonMode

Declare :

UINT UAI_SpectrometerGetXenonMode(void* api_handle, unsigned int* mode)

Visual Basic Declare :

_VB6UAI_SpectrometerGetXenonMode@8

Get pulse or xenon mode

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
mode	Get mode.

3.11. Other

3.11.1.UAI_SpectrometerSetExternalPort

Declare :

UINT UAI_SpectrometerSetExternalPort(void* api_handle, unsigned int port)

Visual Basic Declare :

_VB6UAI_SpectrometerSetExternalPort@8

This function supports spectrameter controls the status of 6 GPIO .

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
port	Provide the status .

Introduction

OTO spectrameter provide 6 GPIO for IO controlling that controller can set high(1) or low(0) of each pins .

Example :

GPIO 6	GPIO 5	GPIO 4	GPIO 3	GPIO 2	GPIO 1
0	1	0	0	0	0

Port value = 2

GPIO 6	GPIO 5	GPIO 4	GPIO 3	GPIO 2	GPIO 1
0	1	1	0	0	0

Port value = 6

GPIO 6	GPIO 5	GPIO 4	GPIO 3	GPIO 2	GPIO 1
0	1	0	0	0	1

Port value = 34

3.11.2.UAI_SpectrometerGetExternalPort

Declare :

UINT UAI_SpectrometerGetExternalPort(void* api_handle, unsigned int* port)

Visual Basic Declare :

_VB6UAI_SpectrometerGetExternalPort@8

This function supports spectrameter controls the status of 6 pins .

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
port	Return the status .

3.11.3.UAI_SpectrometerInitUserRom

Declare :

UINT UAI_SpectrometerInitUserRom(void* api_handle)

Visual Basic Declare :

_VB6UAI_SpectrometerInitUserRom@4

This function will allocate a 512 bytes buffer for user to Set or Get user rom . Application have to execute this function once when it's connecting device before using UAI_SpectrometerSetUserRom or UAI_SpectrometerGetUserRom.

Parameter

Name	Description
api_handle	Specify the spectrometer handle.

3.11.4.UAI_SpectrometerSetUserRom

Declare :

UINT UAI_SpectrometerSetUserRom(void* api_handle, unsigned char *buffer, unsigned int length, unsigned int offset)

Visual Basic Declare :

_VB6UAI_SpectrometerSetUserRom@16

This function supports function that system developer can create and set their own configuration table(512 bytes) into device .

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
buffer	Provide setting value.
length	Provide the length of buffer.
offset	Provide the index of 512 bytes array

3.11.5.UAI_SpectrometerGetUserRom

Declare :

UINT UAI_SpectrometerGetUserRom(void* api_handle, unsigned char *buffer, unsigned int length, unsigned int offset)

Visual Basic Declare :

_VB6UAI_SpectrometerGetUserRom@16

This function supports function that system developer can get their own configuration table(512 bytes) from device .

Parameter

Name	Description
api_handle	Specify the spectrometer handle.
buffer	Return setting value.
length	Provide the length of buffer.
offset	Provide the index of 512 bytes array

3.11.6.UAI_MATHGetCurveInfo

Declare :

UINT UAI_MATHGetCurveInfo(float* lambda ,float* Intensity,unsigned int size,float wavelength_start ,float wavelength_end ,float* LambdaP,float* CenterWavelength,float FWHM)

Visual Basic Declare :

_VB6UAI_MATHGetCurveInfo@32

This function calculates lambda of spectrum's peak , Center Wavelength and FWHM of the peak .

Parameter

Name	Description
Lambda	Provide lambda array of calculated spectrum.
Intensity	Provide intensity array of calculated spectrum.
Size	Provide size info of Lambda and intensity array
Wavelength_start	Provide start of wavelength for analysis
Wavelength_end	Provide end of wavelength for analysis
LambdaP	Return pointer of lambda of spectrum's max peak.
CenterWavelength	Return pointer of Center wavelength of peak.
FWHM	Return FWHM of peak.

Introduction

Size should be the length of Lambda or intensity array .The length of lambda and Intensity is same.

4.Example code

4.1. Device Connection

User have to connect Device and get related information before data acquiring . Please refer below sample code by c# .

```
private bool ConnectDevice()
{
      uint status;
      uint device_num = 0;
      uint i = 0;
      //Get compatiable VID and PID
      UInt32 buffersize = 0;
      UInt32[] VIDPID = new UInt32[3]; //For 2.5.0 dll , support FX2 series.
      unsafe
      {
            Link_UAI.UAI_SepctrometerGetDeviceList(ref buffersize, null); //Get how many series of OtO
      spectrometer is supported
            VIDPID = new UInt32[buffersize * 2]; //VIDPID buffersize should be 2 X VIDPID counts.
            fixed (UInt32* temp_VIDPID = VIDPID)
            {
                  Link_UAI.UAI_SepctrometerGetDeviceList(ref buffersize, temp_VIDPID);
                  //Get list of VIDPID of OtO spectrometer.

            }
      }
      for (int j = 0; j < buffersize * 2; j = j + 2)
      {
            device_num = 0;
            status = Link_UAI.UAI_SepctrometerGetDeviceAmount(VIDPID[j], VIDPID[j + 1], ref device_num);
            //Get number of connected device for each series.
            if (status == 0 && device_num > 0) //OTO_SUCCESS == status
            {
                  for (i = 0; i < device_num; i++)
                  {
                        status = Link_UAI.UAI_SpectrometerOpen(i, ref SD_Live.DeviceHandle, VIDPID[j], VIDPID[j +
                  1]);
                        //Open each spectrometer.

                        if (status == 0)
                        {
                        //Open device successfully then get frame size
                        Link_UAI.UAI_SpectromoduleGetFrameSize(SD_Live.DeviceHandle, ref SD_Live.framesize);
                        if (SD_Live.framesize == 0)
                              return false;
                        //Get serial
                        byte[] temp_SN = new byte[16];
                        Link_UAI.UAI_SpectrometerGetSerialNumber(SD_Live.DeviceHandle, temp_SN);
                        SD_Live.SerialNumber = System.Text.Encoding.Default.GetString(temp_SN).Replace(@"\s",
                  "");
                        SD_Live.SerialNumber = SD_Live.SerialNumber.Substring(0,
                  Link_UAI.LastIndexOfnumber(temp_SN));
                        //Get wavelength table
                        status = Link_UAI.UAI_SpectrometerWavelengthAcquire(SD_Live.DeviceHandle,
                  SD_Live.Lambda);
                        if (status != 0)
                        {
                              MessageBox.Show(this, "Error happens at UAI_SpectrometerWavelengthAcquire()",
                        "ERROR");
```

```
                        return false;
                    }
                    break;
                    }
                }
            return true;
            }
            else
            return false;
        }
        return true;
}
```

4.2. Acquire Intensity

Below c# example code shows how to get intensity data from specified device by providing devicehandle . For normal measurement like as Transmission , Reflection , Absorbance and Concentration , AP should apply related correction function .

```
public void GetData()
{
    try
    {
        uint ErrorCode = 0;

        //Get Data from Device
        ErrorCode = Link_UAI.UAI_SpectrometerDataAcquires(DeviceHandle, integration_time * 1000, Intensity, Avg);
        if (ErrorCode != 0)
        {
            MessageBox.Show("Error happen at bgW_GetData_DoWork(UAI_SpectrometerDataAcquire) : Errorcode(0x" +
Convert.ToString(ErrorCode, 16) + ").");
            ErrorCode = 0;
        }

        //Do Background_Remove correction
        ErrorCode = Link_UAI.UAI_BackgroundRemove(DeviceHandle, integration_time * 1000, Intensity);
        if (ErrorCode != 0)
        {
            MessageBox.Show("Error happen at bgW_GetData_DoWork(UAI_BackgroundRemove) : Errorcode(0x" +
Convert.ToString(ErrorCode, 16) + ").");
            ErrorCode = 0;
        }

        //Do linearity Calibration
        ErrorCode = Link_UAI.UAI_LinearityCorrection(DeviceHandle, framesize, Intensity);
        if (ErrorCode != 0)
        {
            MessageBox.Show("Error happen at bgW_GetData_DoWork(UAI_LinearityCorrection) : Errorcode(0x" +
Convert.ToString(ErrorCode, 16) + ").");
            ErrorCode = 0;
        }

        //Do Absolute Intensity Calibration. for Color measurement only.
ErrorCode = Link_UAI.UAI_AbsoluteIntensityCorrection(DeviceHandle, SD_Live.Intensity, integration_time * 1000);
if (ErrorCode != 0)
{
    MessageBox.Show("Error happen at bgW_GetData_DoWork(UAI_AbsoluteIntensityCorrection) : Errorcode(0x" +
Convert.ToString(ErrorCode, 16) + ").");
    ErrorCode = 0;
}

    }
    catch (Exception ex)
    {
        MessageBox.Show("Error happen at GetData()." + ex.Message);
    }
}
```

4.3. Get Color Information

Below c# example code shows how to get color information for color measurement . Before getting color information , user should do necessary correction (Background_Remove , linearity Calibration , Absolute Intensity Calibration) during intensity acquiring .

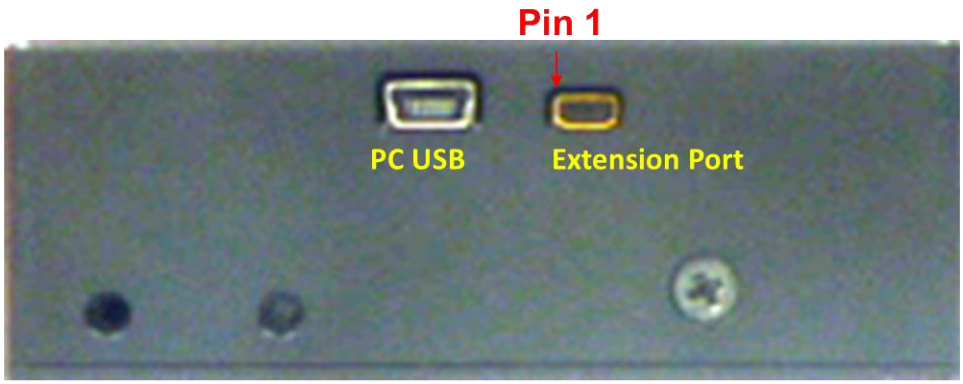
```
private void GetColorInfo()
{
    try
    {
        uint ErrorCode = 0;
        IntPtr ColorIntPtr;
        //Allocate and provide related buffer to API
        ErrorCode = Link_UAI.UAI_ColorInformationAllocation(ref ColorIntPtr, 2, (uint)observer, (uint)illuminant,
SD_Live.Lambda, null, Intensity, framesize);
        if (ErrorCode != 0)
        {
            MessageBox.Show(this, "Error happens at UAI_ColorInformationAllocation()");
        }
        //Do Color operation
        ErrorCode = Link_UAI.UAI_ColorOperation(ColorIntPtr);
        if (ErrorCode != 0)
        {
            MessageBox.Show(this, "Error happens at UAI_ColorOperation ()");
        }

        //Get Color Information of each items .
        double[] temp_B_3 = new double[3];

        ErrorCode = Link_UAI.UAI_ColorGetCIExyz(ColorIntPtr, temp_B_3);
        if (ErrorCode != 0)
        {
            MessageBox.Show("Error happen at OTO_DSPLI_ColorGetxyz : Errorcode(0x" +
Convert.ToString(ErrorCode, 16) + ").");
        }
        Colorinfo.x = temp_B_3[0];
        Colorinfo.y = temp_B_3[1];
        Colorinfo.z = temp_B_3[2];

        ErrorCode = Link_UAI.UAI_ColorGetXYZ(ColorIntPtr, temp_B_3);
        if (ErrorCode != 0)
        {
            MessageBox.Show("Error happen at UAI_ColorGetXYZ : Errorcode(0x" + Convert.ToString(ErrorCode, 16)
+ ").");
        }
        Colorinfo.X = temp_B_3[0];
        Colorinfo.Y = temp_B_3[1];
        Colorinfo.Z = temp_B_3[2];

        ErrorCode = Link_UAI.UAI_ColorGet1976ucs(ColorIntPtr, temp_B_3);
        if (ErrorCode != 0)
        {
            MessageBox.Show("Error happen at UAI_ColorGet1960ucs : Errorcode(0x" + Convert.ToString(ErrorCode,
16) + ").");
        }
```

```
            Colorinfo.CIE1976u = temp_B_3[0];
            Colorinfo.CIE1976v = temp_B_3[1];
            Colorinfo.CIE1976w = temp_B_3[2];

            ErrorCode = Link_UAI.UAI_ColorGetCCT(ColorIntPtr, ref Colorinfo.CCT);
            if (ErrorCode != 0)
            {
                MessageBox.Show("Error happen at OTO_DSPLI_ColorGetCCT : Errorcode(0x" +
Convert.ToString(ErrorCode, 16) + ").");
            }

            ErrorCode = Link_UAI.UAI_ColorGetDominantWavelength(ColorIntPtr, ref Colorinfo.DominantWavelength);
            if (ErrorCode != 0)
            {
                MessageBox.Show("Error happen at OTO_DSPLI_ColorGetDominantWavelength : Errorcode(0x" +
Convert.ToString(ErrorCode, 16) + ").");
            }

            ErrorCode = Link_UAI.UAI_ColorGetPurity(ColorIntPtr, ref Colorinfo.Purity);
            if (ErrorCode != 0)
            {
                MessageBox.Show("Error happen at OTO_DSPLI_ColorGetPurity : Errorcode(0x" +
Convert.ToString(ErrorCode, 16) + ").");
            }

            double[] CRI = new double[18];
            ErrorCode = Link_UAI.UAI_ColorGetColorRenderingIndex(ColorIntPtr, CRI);
            if (ErrorCode != 0)
            {
                MessageBox.Show("Error happen at OTO_DSPLI_ColorGetColorRenderingIndex : Errorcode(0x" +
Convert.ToString(ErrorCode, 16) + ").");
            }
            SD_Live.Colorinfo.CRI_R0 = CRI[0];
            SD_Live.Colorinfo.CRI_R1 = CRI[1];
            SD_Live.Colorinfo.CRI_R2 = CRI[2];
            SD_Live.Colorinfo.CRI_R3 = CRI[3];
            SD_Live.Colorinfo.CRI_R4 = CRI[4];
            SD_Live.Colorinfo.CRI_R5 = CRI[5];
            SD_Live.Colorinfo.CRI_R6 = CRI[6];
            SD_Live.Colorinfo.CRI_R7 = CRI[7];
            SD_Live.Colorinfo.CRI_R8 = CRI[8];
            SD_Live.Colorinfo.CRI_R9 = CRI[9];
            SD_Live.Colorinfo.CRI_R10 = CRI[10];
            SD_Live.Colorinfo.CRI_R11 = CRI[11];
            SD_Live.Colorinfo.CRI_R12 = CRI[12];
            SD_Live.Colorinfo.CRI_R13 = CRI[13];
            SD_Live.Colorinfo.CRI_R14 = CRI[14];
            SD_Live.Colorinfo.CRI_Ra = CRI[15];
            SD_Live.Colorinfo.CRI_DC = CRI[16];

            //Free memory .
        ErrorCode = Link_UAI.UAI_ColorInformationFree(ColorIntPtr);
        if (ErrorCode != 0)
        {
            MessageBox.Show("Error happen at UAI_ColorOperation : Errorcode(0x" + Convert.ToString(ErrorCode, 16)
+ ").");
        }
```

```
        }
        catch (Exception ex)
        {
            MessageBox.Show("Error happen at GetColorInfo()." + ex.Message);
        }
}
```

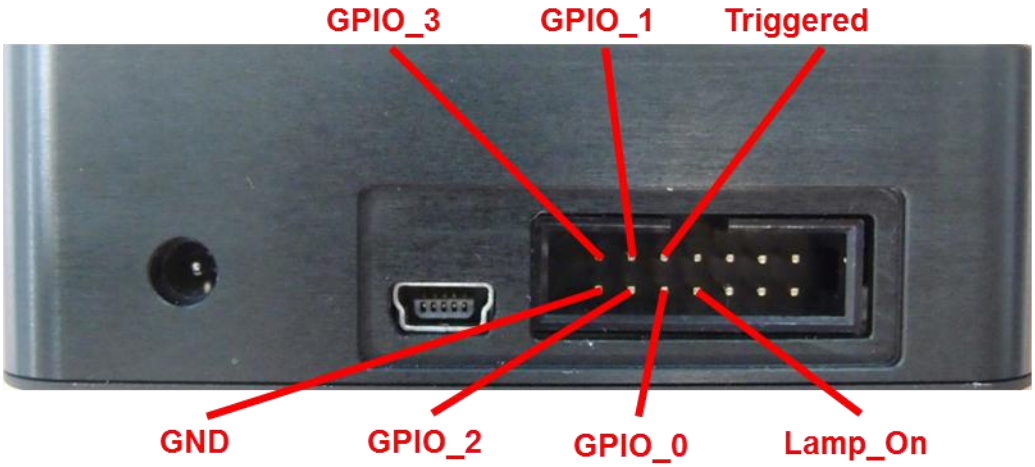
5.Pin definition

5.1. SD X200



Pin No.	Direction	Pin Name	SW Bit index
1	Power	5V Output	
2	Output	GPIO1	5
3	Output	GPIO2	4
4	Output	GPIO3	3
5	Output	GPIO4	2
6	Output	GPIO5	1
7	Input	GPIO6\Trigger	0
8	GND	GND	

5.2. SD X220



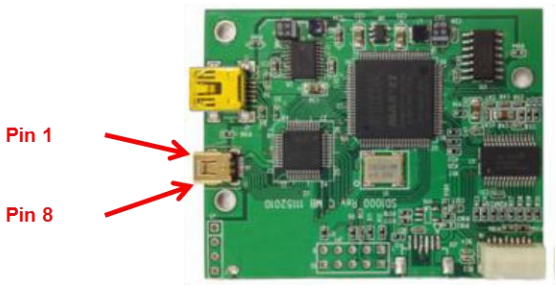
Direction	Pin Name	SW Bit index
Output	GPIO0	5
Output	GPIO1	4
Output	GPIO2	3
Output	GPIO3	2
Output	Lamp_On	1
Input	Trigger	0

5.3. SE XXXX

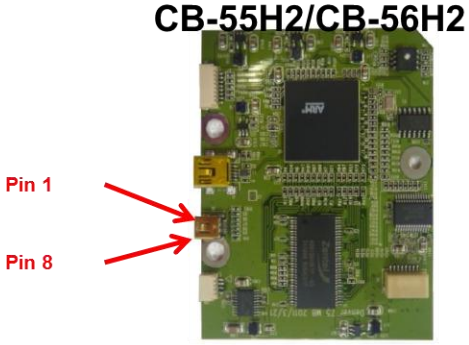
Pin No.	Direction	Pin Name	note
1	Power	5V Output	
2	Output	TX	RS232 TX. TX is the output from the RISC controller.
3	Input	RX	RS232 RX. RX is the input for the RISC controller.
4	Output	GPIO0	General Purpose Output 0.
5	Output	GPIO1	General Purpose Output 1.
6	Output	LS_ON	Light Source Turn ON.
7	Input	Trigger	External Trigger Input Signal.
8	GND	GND	

5.4. MS X200

CB-55H1/CB-56H1

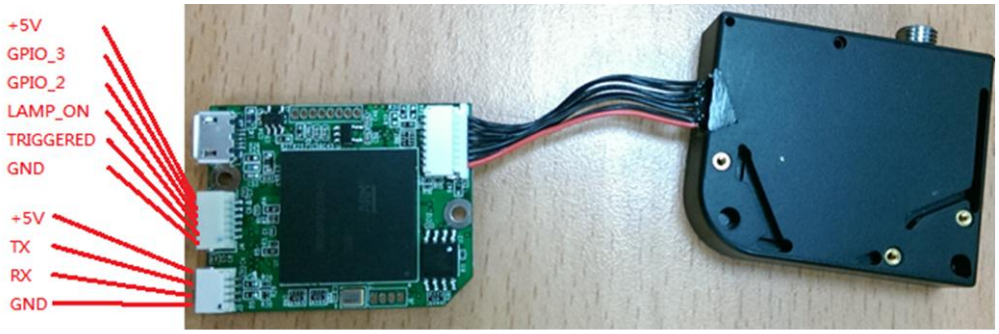


CB-55H2/CB-56H2



Pin No.	Direction	Pin Name	SW Bit index
1	Power	5V Output	
2	Output	GPIO1	5
3	Output	GPIO2	4
4	Output	GPIO3	3
5	Output	GPIO4	2
6	Output	GPIO5	1
7	Input	GPIO6\Trigger	0
8	GND	GND	

5.5. UM X280



Direction	Pin Name	SW Bit index
Output	GPIO2	3
Output	GPIO3	2
Output	Lamp_On	1
Input	Trigger	0

6.Error handling

Errorcode	Defination	DetailInformation
0	API_SUCCESS	
1	API_INT_BUFFER_INVALID	Parameter is NULL or incorrect format.
2	API_INT_FEATURE_UNSUPPORTED	Requested function isn't supported.
3	API_INT_PROTOCOL_ERROR	NET connection is failed.
4	API_INT_CALIBRATION_ERROR	Sample number of calibration is over framesize
5	API_INT_MEMORY_ERROR	Memory from device is incorrect
6	API_INT_ARGUMENT_ERROR	Parameter is incorrect value.
7	API_INT_HANDLE_INVALID	Device is inconnection.
8	API_INT_TIMEOUT	Device doesn't respones.
9	API_INT_DATA_NOT_READY	For Trigger mode.
10	API_INT_DATA_TIME_OUT	For Trigger mode.
11	API_INT_FILE_IO_ERROR	Read / Write file or device memory error.
12	API_INT_FILE_EXIST_ERROR	Read / Write file or device memory error.
13	API_INT_FUNCTION_HW_NOTSUPPORT	Requested function isn't supported.
14	API_INT_FUNCTION_FW_NOTSUPPORT	Requested function isn't supported.
15	API_INT_FILE_SIZE_ERROR_OVERSIZE	Read / Write file or device memory error.
16	API_INT_FWBURN_CHECK_ERROR	Write FW into device memory error.
17	API_INT_FILE_FORMAT_ERROR	Write FW into device memory error.
18	API_INT_FILE_DEVCON_EXIT_REBOOT	HW reset via devcon is incorrect
19	API_INT_FILE_DEVCON_EXIT_FAIL	HW reset via devcon is incorrect
20	API_INT_FILE_DEVCON_EXIT_USAGE	HW reset via devcon is incorrect
-1	LINUXUSB_ERROR_IO	Input/output error
-2	LINUXUSB_ERROR_INVALID_PARAM	Invalid parameter
-3	LINUXUSB_ERROR_ACCESS	Access denied (insufficient permissions)
-4	LINUXUSB_ERROR_NO_DEVICE	No such device (it may have been disconnected)
-5	LINUXUSB_ERROR_NOT_FOUND	Entity not found
-6	LINUXUSB_ERROR_BUSY	Resource busy
-7	LINUXUSB_ERROR_TIMEOUT	Operation timed out
-8	LINUXUSB_ERROR_OVERFLOW	Overflow
-9	LINUXUSB_ERROR_PIPE	Pipe error
-10	LINUXUSB_ERROR_INTERRUPTED	System call interrupted (perhaps due to signal)
-11	LINUXUSB_ERROR_NO_MEM	Insufficient memory
-12	LINUXUSB_ERROR_NOT_SUPPORTED	Operation not supported or unimplemented on this platform

-99	LINUXUSB_ERROR_OTHER	Other error