# CSUN-U30-AR0234 USB カメラモジュール

# 操作マニュアル

株式会社日昇テクノロジー

http://www.csun.co.jp

info@csun.co.jp

作成・更新日（2026/11/10）

- 修正履歴

| NO | バージョン | 修正内容 | 修正日 |
|---|---|---|---|
| 1 | Ver1.0 | 新規作成 | 2026/11/10 |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

## 内容

CSUN-U30-AR0234 USB カメラモジュール ..... 1
操作マニュアル ..... 1
1 SDK の概要 ..... 4
1.1 SDK の紹介 ..... 4
1.2 SDK の構成 ..... 4
1.3 動作環境 ..... 4
1.4 SDK の設定 ..... 5
1.4.1 C++ からのライブラリの呼び出し ..... 5
1.4.2 C# からのライブラリの呼び出し ..... 5
2 呼び出しフロー ..... 6
2.1 共通機能の呼び出しフロー ..... 6
2.2 トリガー機能の呼び出しフロー ..... 7
2.3 デバイスファームウェア更新の呼び出しフロー ..... 8
2.4 ユーザー定義データをデバイスに書き込む際の呼び出しフロー ..... 8
3 関数の説明 ..... 9
3.1 SDK の初期化とデバイス管理 ..... 9
3.2 デバイス情報の取得 ..... 9
3.3 ファームウェアの更新とユーザーデータの読み書き ..... 9
3.4 画像取得とパラメーター制御 ..... 9
3.5 センサー制御 ..... 10
3.6 その他のデバイス機能 ..... 11
3.7 画像のデコード ..... 12
3.8 ファームウェア制御 ..... 12
4 デモの紹介 ..... 13
4.1 C++ ..... 13
4.1.1 Device_Command_Trigger_Control ..... 13
4.1.2 Device_Firmware_Update ..... 14
4.1.3 Device_GPIO_Control ..... 14
4.1.4 Device_Mirror_Flip_Control ..... 14
4.1.5 Device_Printf ..... 15
4.1.6 Device_Processing_Unit_Control ..... 15
4.1.7 Device_RAW_Output_Control ..... 16

4.1.8 Device_Shutter_Gain_Control ..... 16
4.1.9 Device_Single_3A_Control ..... 17
4.1.10 Device_Software_Trigger_Control ..... 17
4.1.11 Device_User_Data_Control ..... 18
4.1.12 Device_User_ESN_Control ..... 18
4.1.13 Device_WhiteBalance_Gain_Control ..... 18
4.1.14 Grab_Asynchronous ..... 19
4.1.15 Grab_CallBack ..... 19
4.1.16 Grab_Image ..... 20
4.1.17 GrabImage_Display ..... 21
4.1.18 GrabImage_Multi_Display ..... 22
4.1.19 Device_Trigger_Control ..... 23
4.1.20 Firmware_Control ..... 24
4.2 C# ..... 25
4.2.1 Device_Check ..... 25
4.2.2 Device_Printf ..... 25
4.2.3 Grab_Callback ..... 26
4.2.4 Grab_Image ..... 26
5 外部パルストリガのタイミング ..... 27
6 外部自動トリガのタイミング ..... 27
7 ハードウェア仕様 ..... 28
接続方法 ..... 28

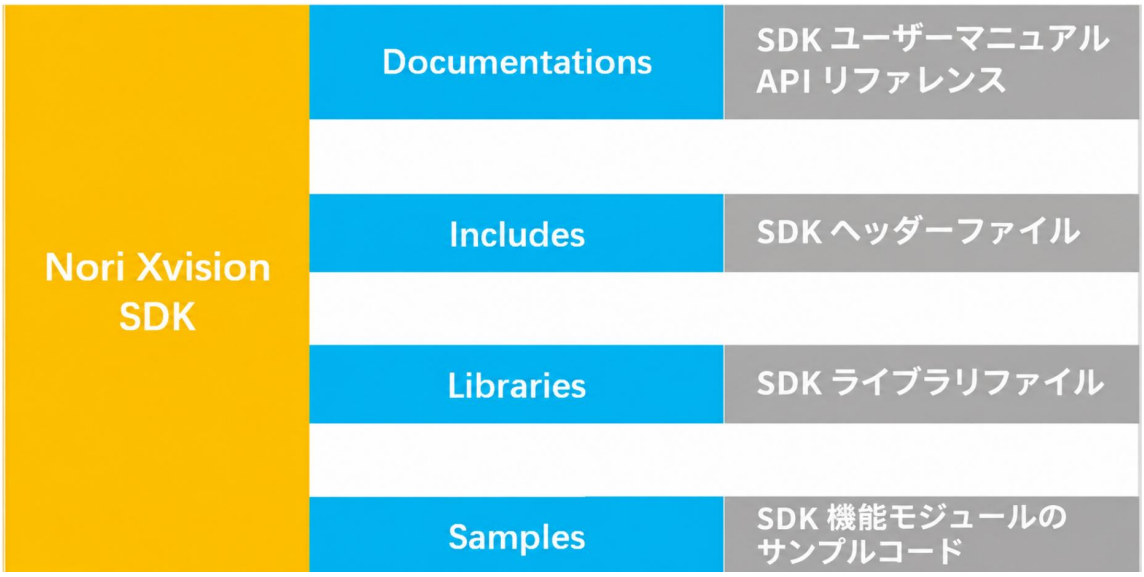
# 1 SDK の概要

## 1.1 SDK の紹介

Nori Xvision SDK は、当社の USB インターフェース搭載カメラ向けの Windows 用ソフトウェア開発キットです。特定の種類のアプリケーション開発に必要な機能インターフェースをライブラリとして提供し、開発者がすべての機能を一から実装することなく、効率よくコードを記述できるようにします。

## 1.2 SDK の構成

SDK は、Documentations、Includes、Libraries、Samples で構成されています。




## 1.3 動作環境

Windows 10/11（32/64 ビット）

## 1.4 SDK の設定

### 1.4.1 C++ からのライブラリの呼び出し

図 2 に示すように、C++ で SDK ライブラリを使用するには、ヘッダーファイルの宣言、外部 DLL 関数の宣言、および対応するバージョンのライブラリの指定が必要です。詳細は下図を参照してください。

```
#include <iostream>
#include <process.h>
#include <conio.h>
#include "../../../../Includes/Nori_Xvision_API/Nori_Xvision_API.h"

#ifndef _WIN64
#pragma comment(lib, "../../../../Libraries/win32/Nori_Xvision_API.lib")

#else
#include <initguid.h>
#pragma comment(lib, "../../../../Libraries/win64/Nori_Xvision_API_X64.lib")

#endif
```

### 1.4.2 C# からのライブラリの呼び出し

図に示すように、C# で SDK ライブラリを使用するには、ヘッダーファイルの宣言、外部 DLL 関数の宣言、および対応するバージョンのライブラリの指定が必要です。詳細は下図を参照してください。

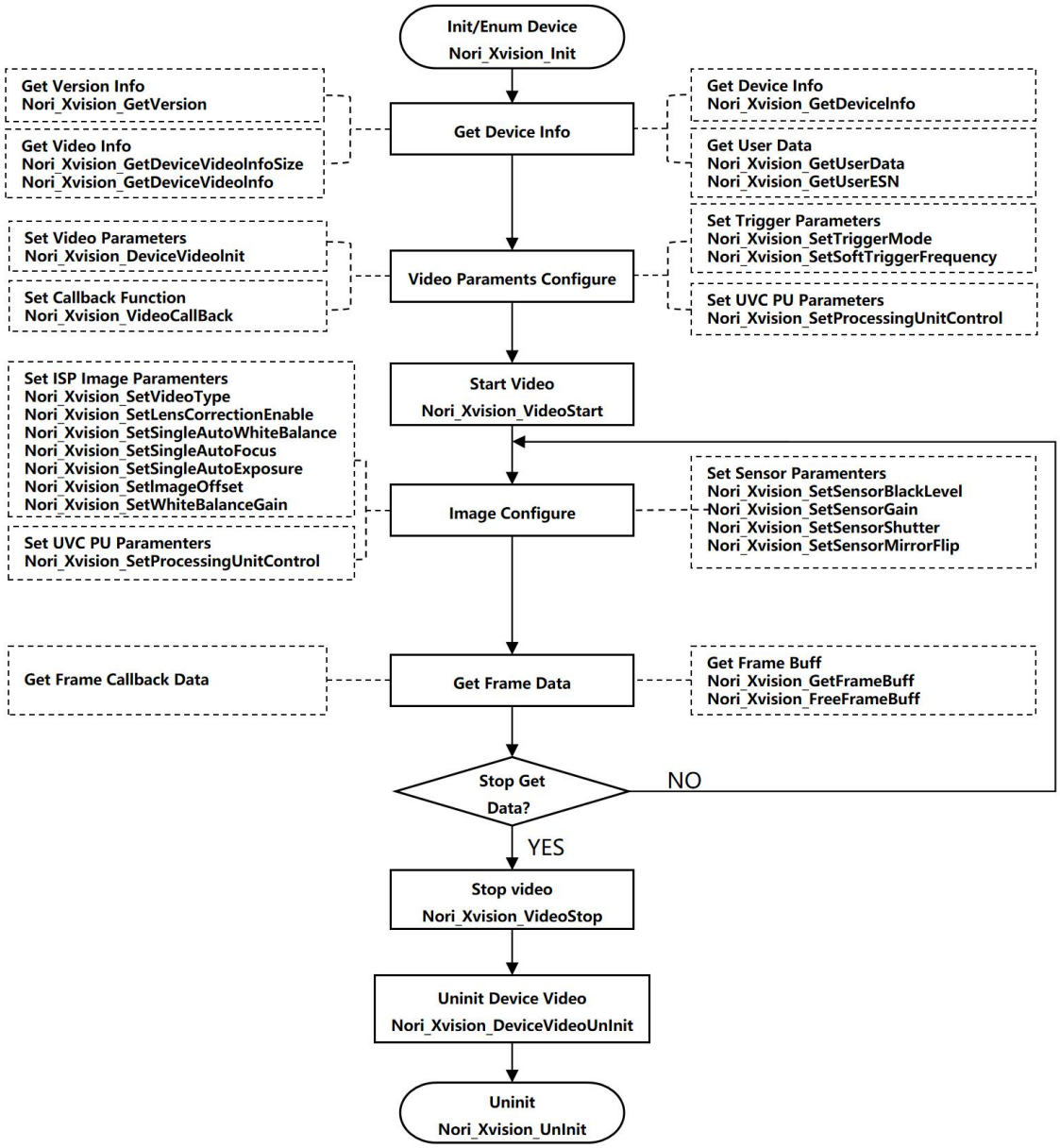
```
// ========================================
[UnmanagedFunctionPointer(CallingConvention.Cdecl)]
public delegate uint PCall_Back_Frame(uint Media_Type, IntPtr pdata, UInt32 size, UInt32 width, UInt32 heigh

// ========================================
// DLL インポート
// ========================================
5 個の参照
public static class NativeMethods
{
    #if X86 && DEBUG
            private const string DllName = @"../../../Libraries/win32\Nori_Xvision_API.dll";
    #elif X86 && RELEASE
            private const string DllName = @"../../../Libraries/win32\Nori_Xvision_API.dll";
    #elif X64 && DEBUG
            private const string DllName = @"../../../../Libraries/win64\Nori_Xvision_API_x64.dll";
    #elif X64 && RELEASE
            private const string DllName = @"../../../../Libraries/win64\Nori_Xvision_API_x64.dll";
    #else
    #endif

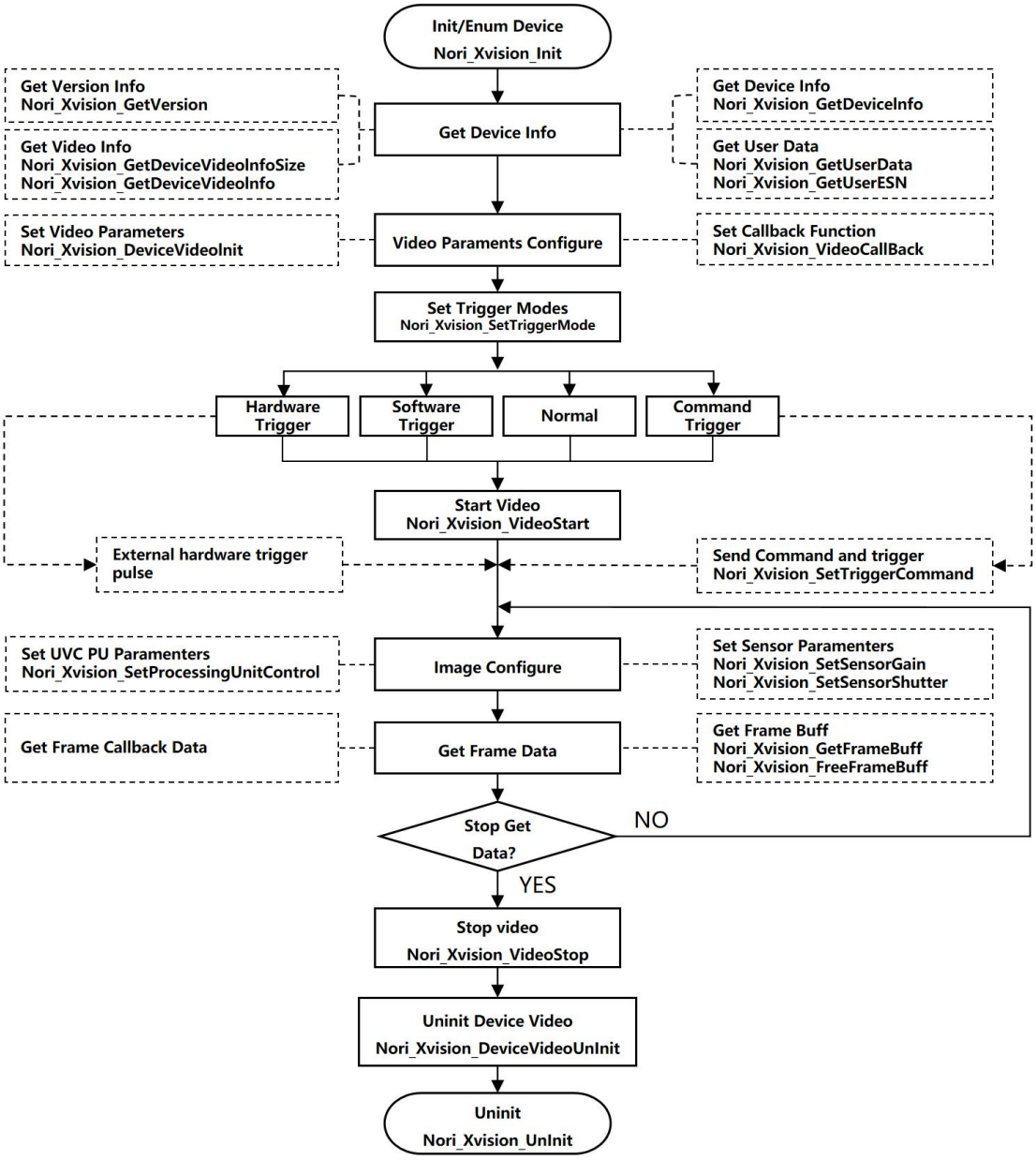
    [DllImport(DllName, CallingConvention = CallingConvention.Cdecl)]
    1 個の参照
    public static extern uint Nori_Xvision_Init(uint uLayerType, ref uint pDeviceNum);
```

# 2 呼び出しフロー

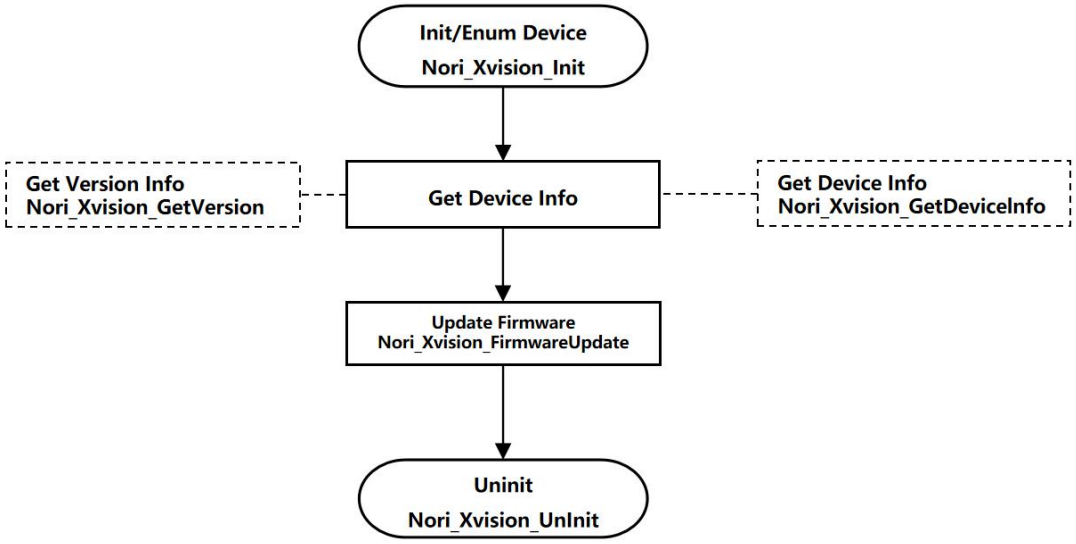
## 2.1 共通機能の呼び出しフロー



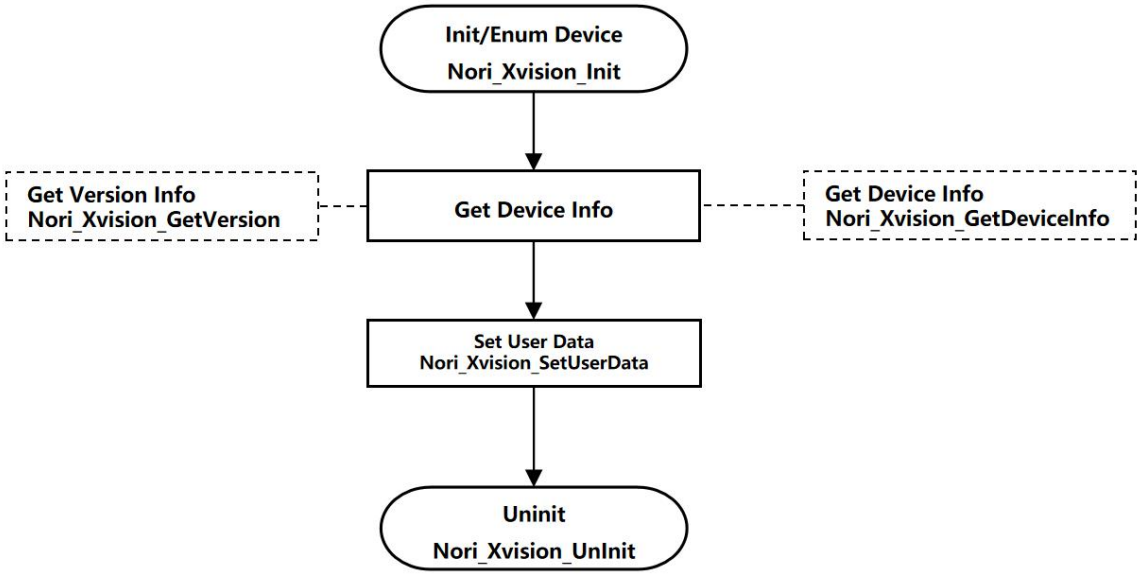
## 2.2 トリガー機能の呼び出しフロー



## 2.3 デバイスファームウェア更新の呼び出しフロー




## 2.4 ユーザー定義データをデバイスに書き込む際の呼び出しフロー



# 3 関数の説明

## 3.1 SDK の初期化とデバイス管理

| 関数名 | 機能説明 |
|---|---|
| Nori_Xvision_Init | カメラ SDK を初期化し、現在接続されているカメラデバイスを列挙する |
| Nori_Xvision_UnInit | SDK のリソースを解放し、カメラシステムを終了する |

## 3.2 デバイス情報の取得

| 関数名 | 機能説明 |
|---|---|
| Nori_Xvision_GetVersion | SDK のバージョン、デバイスの種類、ISP／FPGA のバージョンなどの情報を取得する |
| Nori_Xvision_GetDeviceInfo | カメラのメーカー、型番、シリアル番号、対応機能などのデバイス情報を取得する |
| Nori_Xvision_GetDeviceVideoInfoSize | カメラが対応する映像仕様の数を取得する |
| Nori_Xvision_GetDeviceVideoInfo | 解像度、フレームレート、映像形式の詳細を取得する |

## 3.3 ファームウェアの更新とユーザーデータの読み書き

| 関数名 | 機能説明 |
|---|---|
| Nori_Xvision_FirmWareUpdate | 接続中のカメラのファームウェアを更新する |
| Nori_Xvision_GetUserData | カメラ内部のユーザー定義データを読み出す |
| Nori_Xvision_SetUserData | ユーザー定義データをカメラに書き込む |
| Nori_Xvision_GetUserESN | ユーザー定義のデバイスシリアル番号を取得する |
| Nori_Xvision_SetUserESN | ユーザー定義のデバイスシリアル番号を設定する（Gen060 は非対応） |
| Nori_Xvision_DeviceCheck | ユーザー定義のハンドシェイク処理を行う |

## 3.4 画像取得とパラメーター制御

| 関数名 | 機能説明 |
|---|---|
| Nori_Xvision_DeviceVideoInit | 映像ストリームのパラメーター（形式、解像度、フレームレート）を初期化する |

| 関数名 | 機能説明 |
|---|---|
| Nori_Xvision_DeviceVideoUnInit | 映像取得の終了処理を行い、関連するリソースを解放する |
| Nori_Xvision_VideoStart | 映像の取得を開始する |
| Nori_Xvision_VideoStop | 映像の取得を停止する |
| Nori_Xvision_GetFrameBuff | 呼び出しにより 1 フレームの画像データを取得する |
| Nori_Xvision_FreeFrameBuff | 画像フレームのバッファーを解放する |
| Nori_Xvision_VideoCallBack | リアルタイム映像データのコールバックを登録する |
| Nori_Xvision_SetProcessingUnitControl | 明るさ、コントラスト、彩度、ゲインなどの画像パラメーターを設定する |
| Nori_Xvision_GetProcessingUnitControl | 明るさ、コントラスト、ゲインなどの画像パラメーターを取得する |
| Nori_Xvision_GetCameraTerminalControl | 露光、低照度補正などのカメラパラメーターを取得する |
| Nori_Xvision_SetCameraTerminalControl | 露光モード、低照度補正などを設定する |
| Nori_Xvision_VideoStreamStatus | 映像ストリームの状態（フレームの到着）を照会する、または待機する |

## 3.5 センサー制御

| 関数名 | 機能説明 |
|---|---|
| Nori_Xvision_GetSensorRawInfo | RAW のビット深度、Bayer 形式などの情報を取得する |
| Nori_Xvision_GetSensorBlackLevel | センサーのブラックレベルを取得する |
| Nori_Xvision_SetSensorBlackLevel | センサーのブラックレベルを設定する |
| Nori_Xvision_GetSensorShutter | センサーの露光時間を取得する |
| Nori_Xvision_SetSensorShutter | センサーの露光時間を設定する |
| Nori_Xvision_GetSensorGain | センサーのゲインを取得する |
| Nori_Xvision_SetSensorGain | センサーのゲインを設定する |
| Nori_Xvision_GetSensorMirrorFlip | 画像のミラー／反転状態を取得する |
| Nori_Xvision_SetSensorMirrorFlip | 画像のミラー／反転を設定する |
| Nori_Xvision_GetWhiteBalanceGain | ホワイトバランスのゲインを取得する（Gen056 は非対応） |

| 関数名 | 機能説明 |
|---|---|
| Nori_Xvision_SetWhiteBalanceGain | ホワイトバランスのゲインを設定する（Gen056 は非対応） |

## 3.6 その他のデバイス機能

| 関数名 | 機能説明 |
|---|---|
| Nori_Xvision_GetTriggerMode | カメラの現在のトリガーモードを取得する |
| Nori_Xvision_SetTriggerMode | カメラのトリガー動作モードを設定する |
| Nori_Xvision_SetTriggerCommand | コマンドにより 1 フレームの取得をトリガーする |
| Nori_Xvision_GetSoftTriggerFrequency | ソフトウェアトリガーの周波数を取得する |
| Nori_Xvision_SetSoftTriggerFrequency | ソフトウェアトリガーの周波数を設定する |
| Nori_Xvision_SetIO | GPIO の入出力と信号レベルを設定する |
| Nori_Xvision_GetIO | GPIO の現在の状態を取得する |
| Nori_Xvision_SetPWMFrequencyRatio | PWM の周波数とデューティ比を設定する（Gen056 は非対応） |
| Nori_Xvision_GetPWMFrequencyRatio | PWM のパラメーターを取得する（Gen056 は非対応） |
| Nori_Xvision_GetVideoType | 現在の映像出力タイプを取得する |
| Nori_Xvision_SetVideoType | 映像出力タイプを設定する |
| Nori_Xvision_GetLensCorrectionEnable | レンズの歪み補正の状態を取得する（Gen056 は非対応） |
| Nori_Xvision_SetLensCorrectionEnable | レンズの歪み補正を有効／無効にする（Gen056 は非対応） |
| Nori_Xvision_SetSingleAutoWhiteBalance | 自動ホワイトバランスを 1 回実行し、その結果を固定する（Gen056 は非対応） |
| Nori_Xvision_GetSingleAutoWhiteBalance | 単発の自動ホワイトバランスの状態を取得する（Gen056 は非対応） |
| Nori_Xvision_SetSingleAutoFocus | オートフォーカスを 1 回実行する（Gen056 は非対応） |
| Nori_Xvision_GetSingleAutoFocus | 単発のオートフォーカスの状態を取得する（Gen056 は非対応） |
| Nori_Xvision_SetSingleAutoExposure | 自動露光を 1 回実行する（Gen056 は非対応） |
| Nori_Xvision_GetSingleAutoExposure | 単発の自動露光の状態を取得する（Gen056 は非対応） |

| 関数名 | 機能説明 |
|---|---|
| Nori_Xvision_GetImageOffset | 画像のオフセット（ROI オフセット）を取得する（Gen056 は非対応） |
| Nori_Xvision_SetImageOffset | 画像のオフセット（ROI）を設定する（Gen056 は非対応） |
| Nori_Xvision_GetMultipleVideoSwap | 複数の映像間での画像位置の入れ替え状態を取得する（Gen056 は非対応） |
| Nori_Xvision_SetMultipleVideoSwap | 複数の映像間での画像位置の入れ替え状態を設定する（Gen056 は非対応） |

## 3.7 画像のデコード

| 関数名 | 機能説明 |
|---|---|
| Nori_Xvision_Raw10ToBGR24 | RAW10 データを BGR24 形式にデコードする |
| Nori_Xvision_YUYVTOBGR24Decode | 生の YUYV 画像データを BGR24 形式にデコードする |
| Nori_Xvision_SaveBGR24ToBMP | BGR24 画像データをローカルの BMP ファイルに保存する |

## 3.8 ファームウェア制御

| 関数名 | 機能説明 |
|---|---|
| Nori_Xvision_FirmwareInit | ファームウェア設定を初期化する |
| Nori_Xvision_FirmwareUnInit | ファームウェア設定の終了処理を行い、関数内部のファームウェアデータを解放する |
| Nori_Xvision_FirmwareGetVID | デバイスの Vendor ID を取得する |
| Nori_Xvision_FirmwareSetVID | デバイスの Vendor ID を設定する |
| Nori_Xvision_FirmwareGetPID | デバイスの Product ID を取得する |
| Nori_Xvision_FirmwareSetPID | デバイスの Product ID を設定する |
| Nori_Xvision_FirmwareGetBCD | デバイスの BCD バージョン文字列を取得する |
| Nori_Xvision_FirmwareGetManufacturer | メーカーの文字列を取得する |
| Nori_Xvision_FirmwareSetManufacturer | メーカーの文字列を設定する |
| Nori_Xvision_FirmwareGetSerialNumber | デバイスのシリアル番号文字列を取得する |
| Nori_Xvision_FirmwareSetSerialNumber | デバイスのシリアル番号文字列を設定する |
| Nori_Xvision_FirmwareGetProduct | 製品の文字列を取得する |

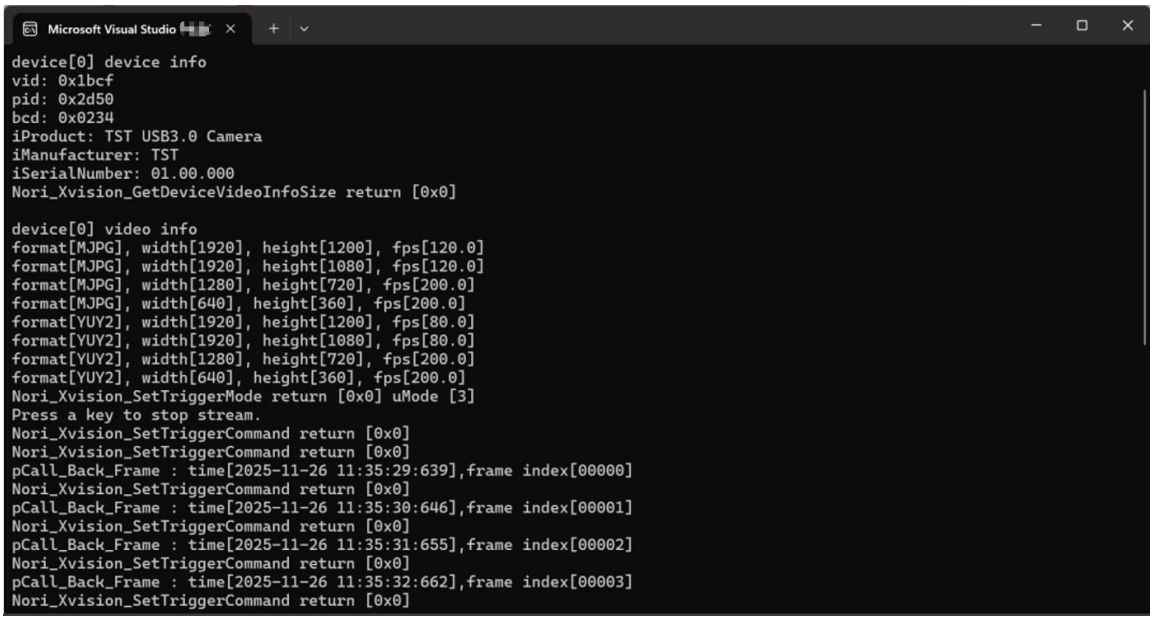
| 関数名 | 機能説明 |
|---|---|
| Nori_Xvision_FirmwareSetProduct | 製品の文字列を設定する |
| Nori_Xvision_FirmwareGetAudioProduct | オーディオ製品の文字列を取得する |
| Nori_Xvision_FirmwareSetAudioProduct | オーディオ製品の文字列を設定する |
| Nori_Xvision_FirmwareGetESN | デバイスの ESN／EXID 文字列を取得する |
| Nori_Xvision_FirmwareSetESN | デバイスの ESN／EXID 文字列を設定する |
| Nori_Xvision_FirmwareGetData | ファームウェアデータのポインターを取得する |
| Nori_Xvision_FirmwareGetLen | ファームウェアデータの長さを取得する |

# 4 デモの紹介

## 4.1 C++

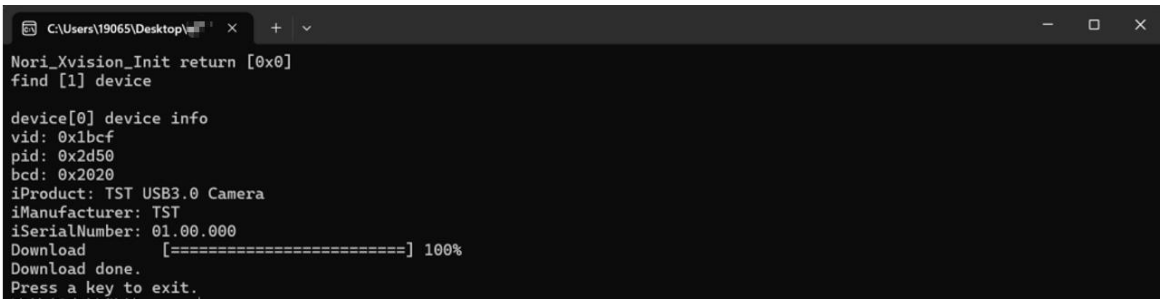
### 4.1.1 Device_Command_Trigger_Control

説明：コマンドトリガーのサンプル



### 4.1.2 Device_Firmware_Update

説明：USB カメラのファームウェア更新サンプル



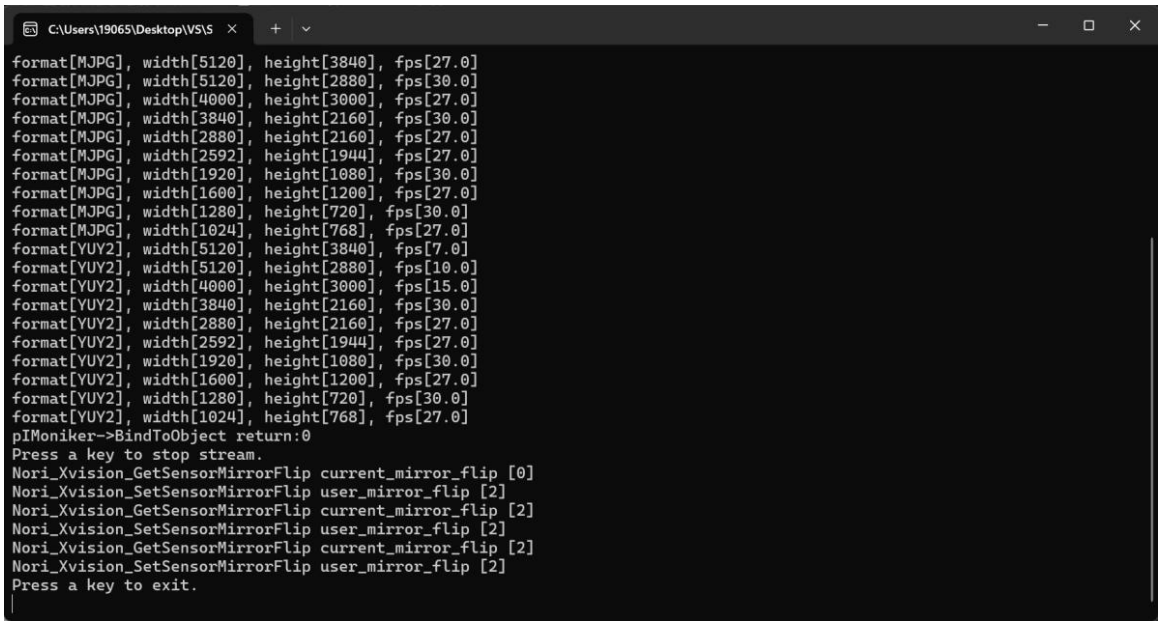

### 4.1.3 Device_GPIO_Control

説明：GPIO の読み取り・設定サンプル




### 4.1.4 Device_Mirror_Flip_Control

説明：画像のミラー表示・反転状態の取得・設定サンプル



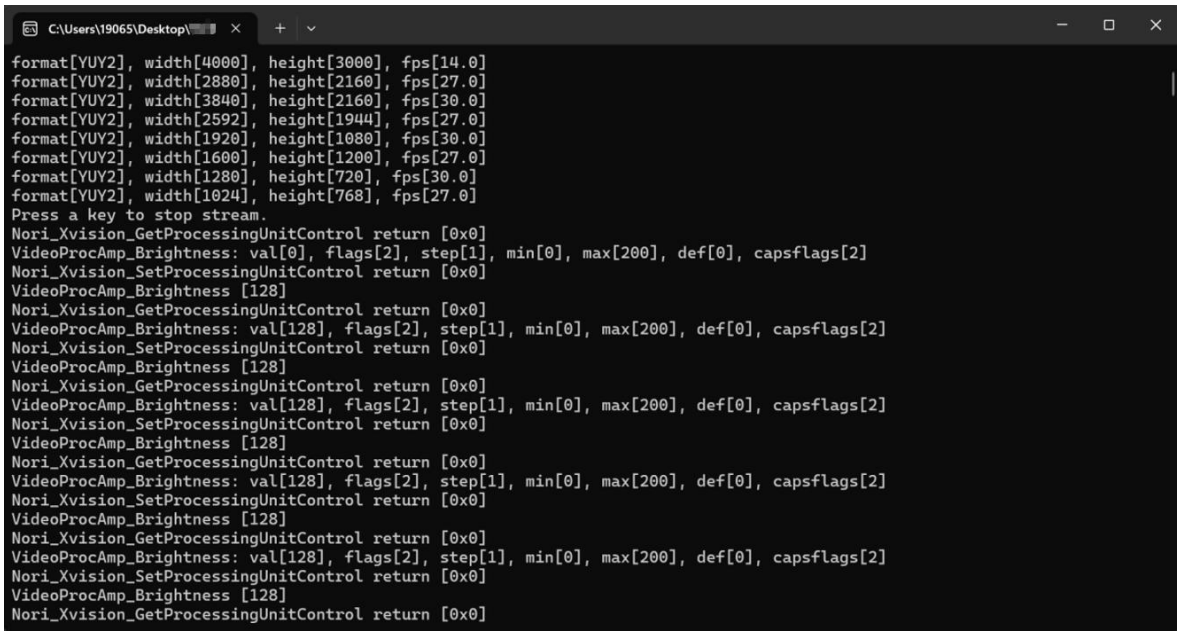
### 4.1.5 Device_Printf

説明：カメラのデバイス情報を取得するサンプル



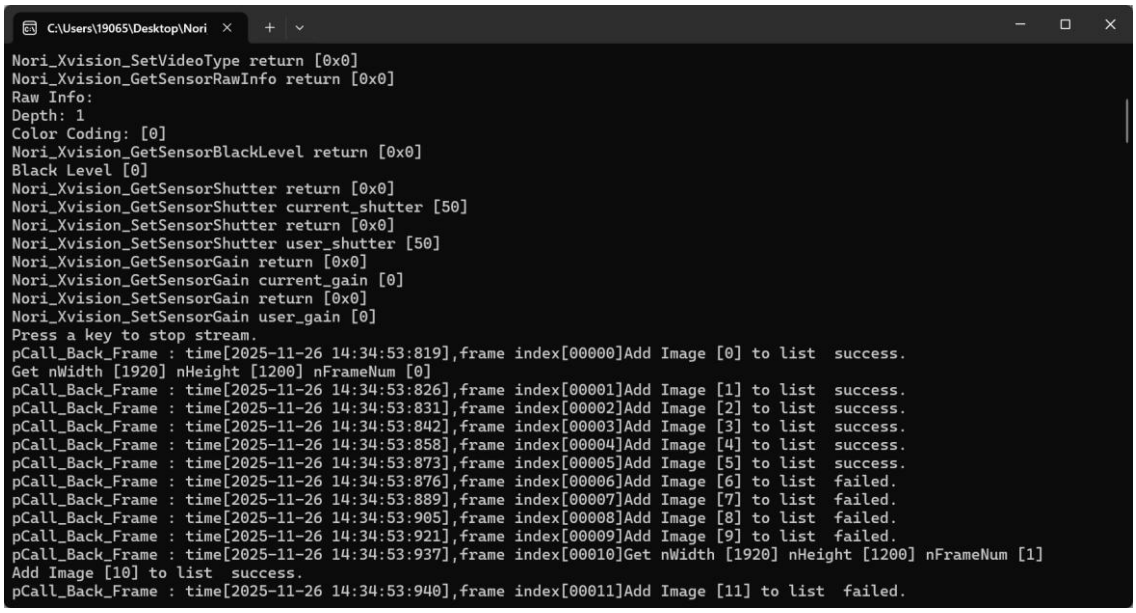

### 4.1.6 Device_Processing_Unit_Control

説明：UVC Processing Unit パラメーターの取得・設定サンプル



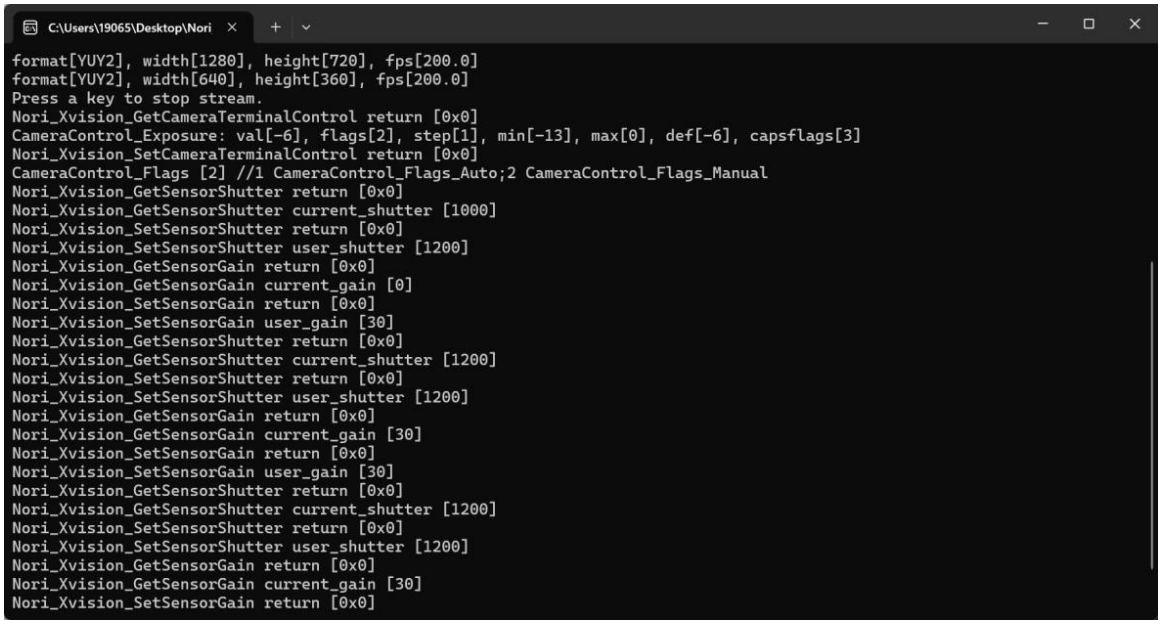
### 4.1.7 Device_RAW_Output_Control

説明：RAW データの取得サンプル



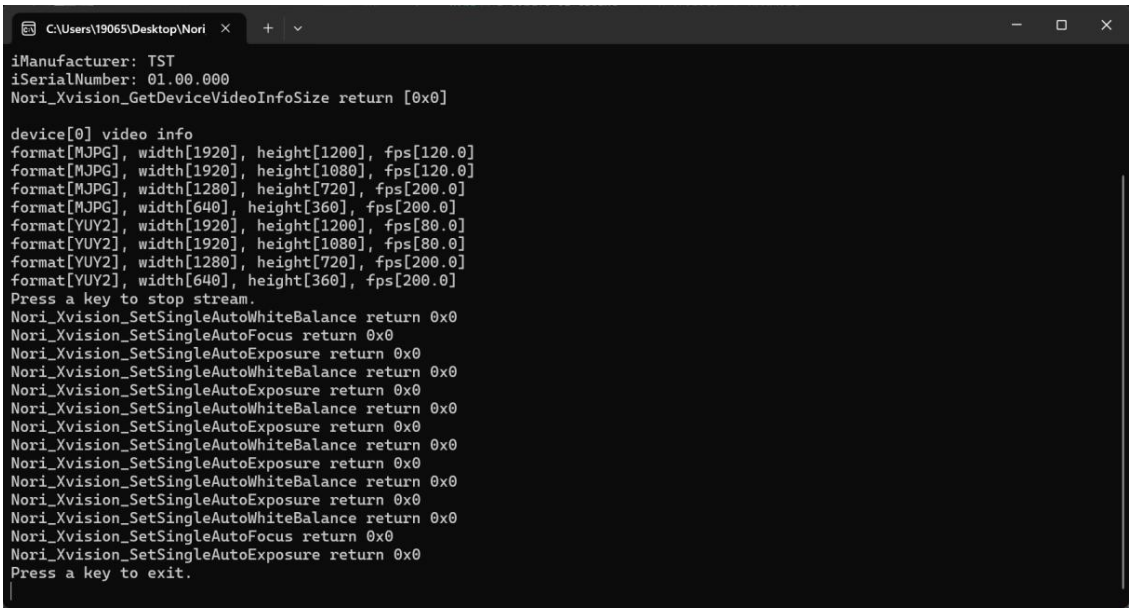

### 4.1.8 Device_Shutter_Gain_Control

説明：露光時間・ゲイン倍率の取得・設定サンプル



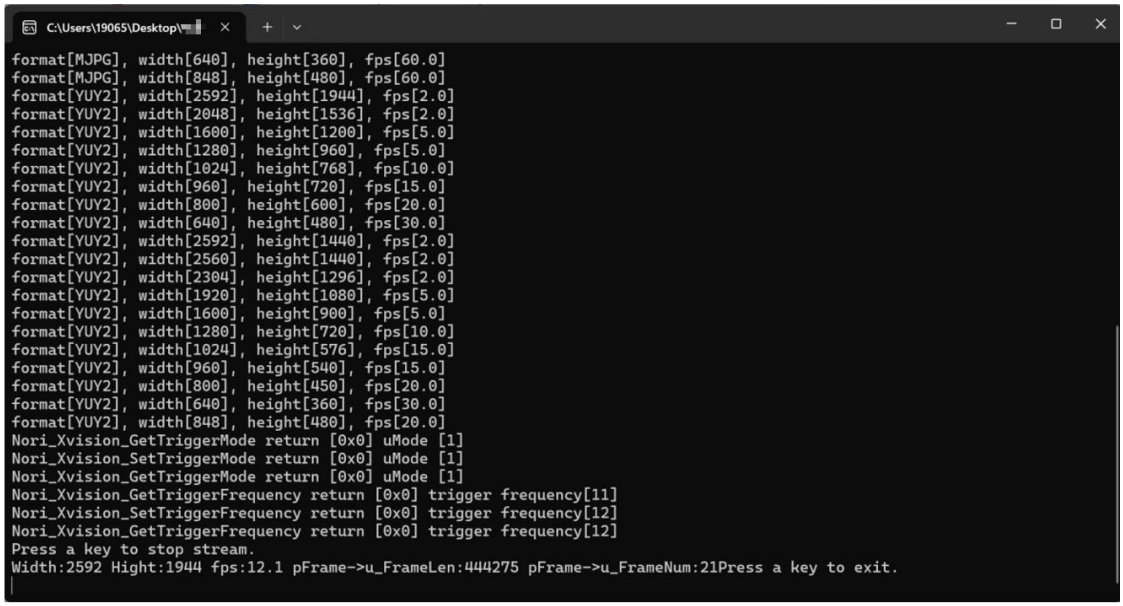
### 4.1.9 Device_Single_3A_Control

説明：単発の自動露光・オートフォーカス・自動ホワイトバランスの取得・設定サンプル




### 4.1.10 Device_Software_Trigger_Control

説明：ソフトウェアトリガーモードおよびソフトウェアトリガー周波数の設定サンプル



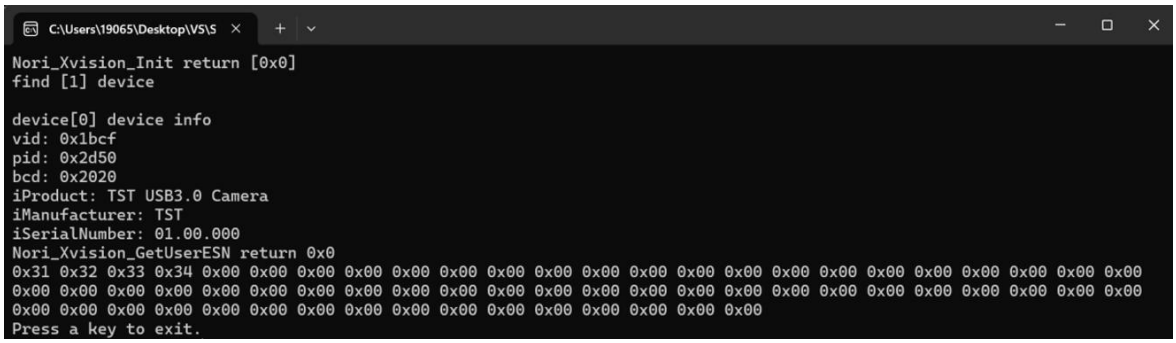
### 4.1.11 Device_User_Data_Control

説明：カメラのユーザー定義データ領域の内容を取得・設定するサンプル



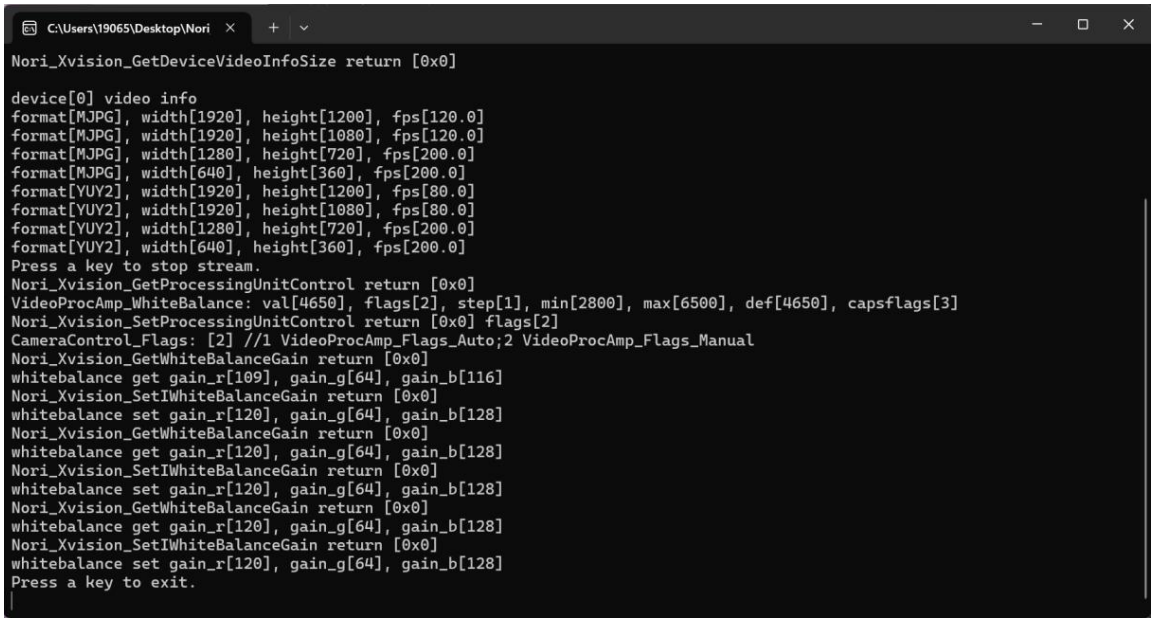

### 4.1.12 Device_User_ESN_Control

説明：ユーザー定義の非標準シリアル番号を読み取るサンプル（読み取りのみ）



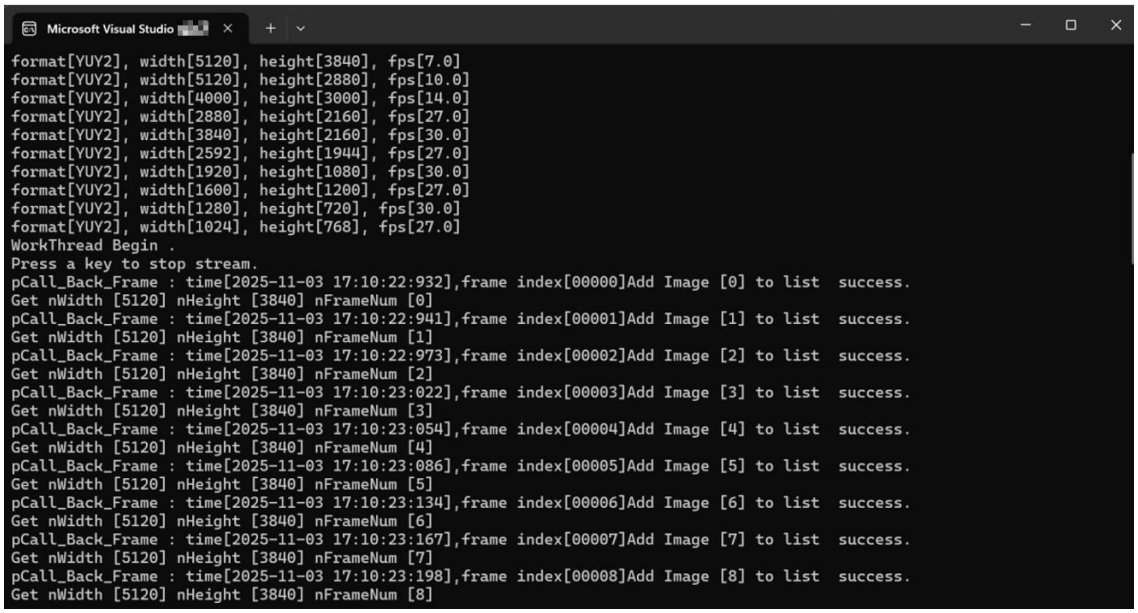

### 4.1.13 Device_WhiteBalance_Gain_Control

説明：ホワイトバランスの R/G/B ゲインを取得・設定するサンプル



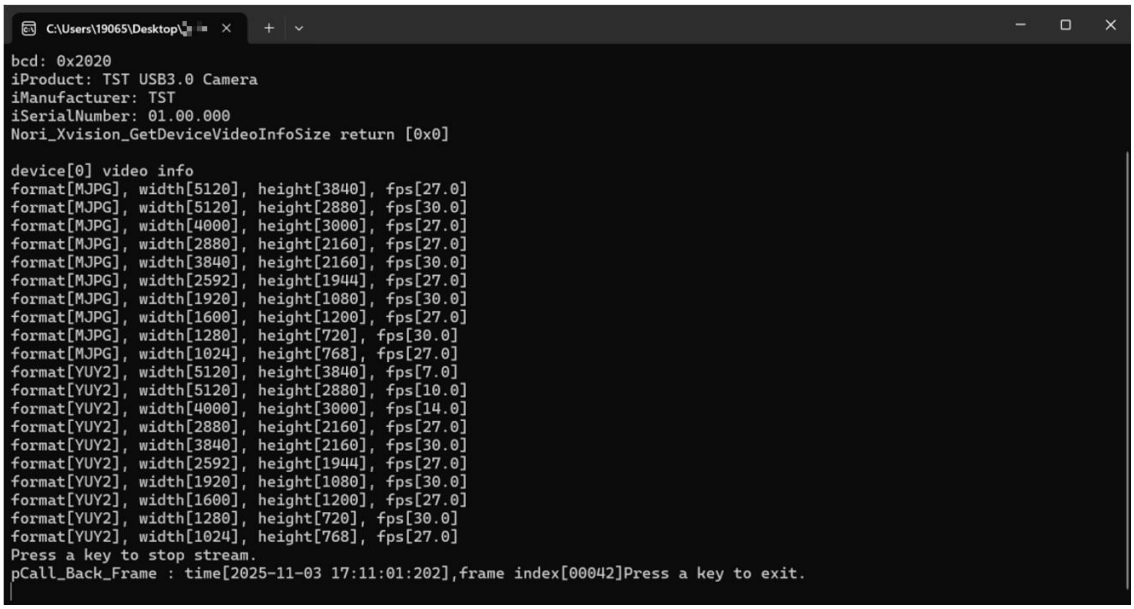
### 4.1.14 Grab_Asynchronous

説明：カメラ用スレッドで映像ストリームを非同期処理するサンプル



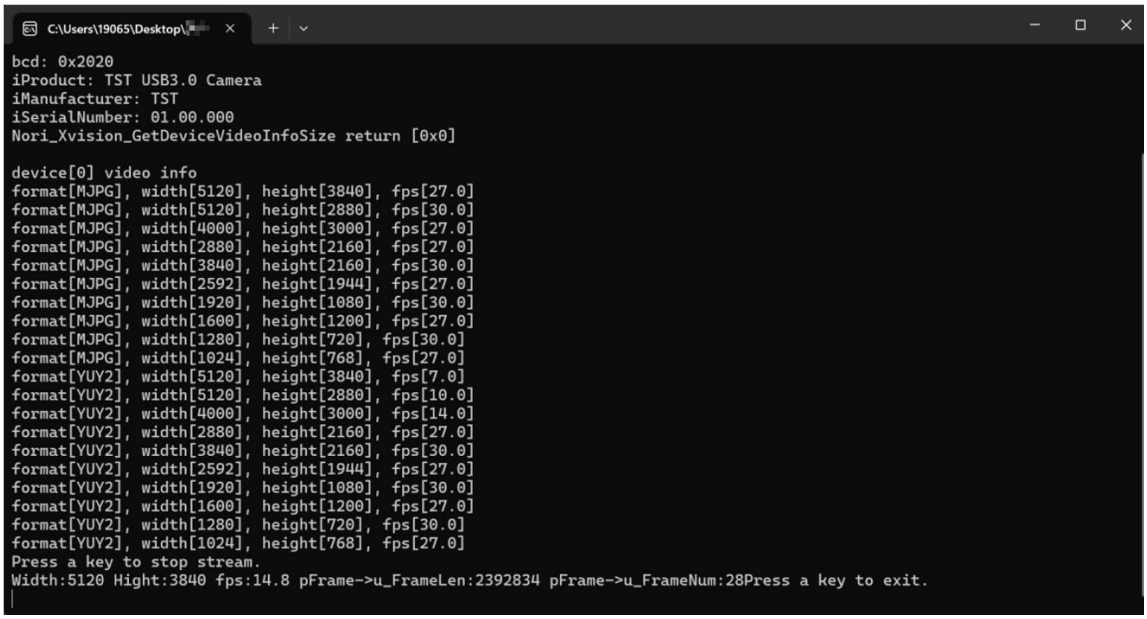

### 4.1.15 Grab_CallBack

説明：コールバック関数を用いてカメラの映像ストリームを取得するサンプル



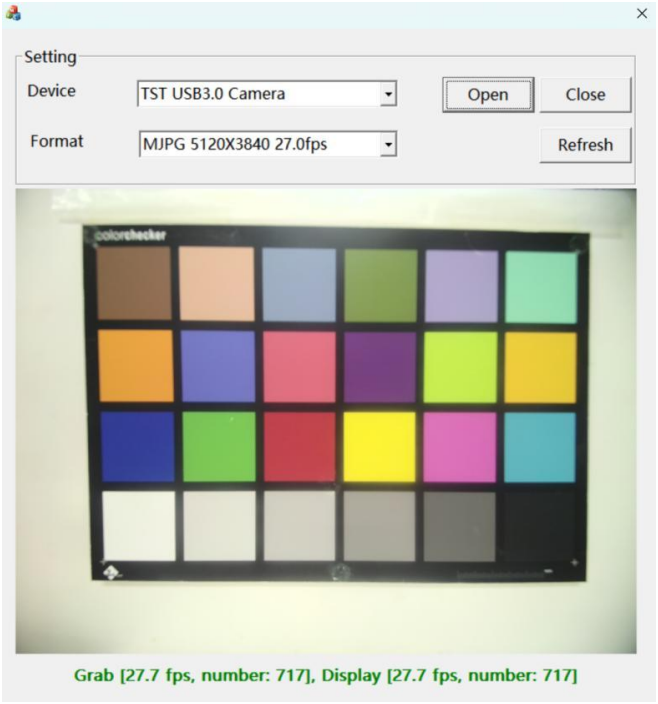
### 4.1.16 Grab_Image

説明：映像ストリームを能動的に取得するサンプル



### 4.1.17 GrabImage_Display

説明：カメラの映像ストリームを取得する UI サンプル

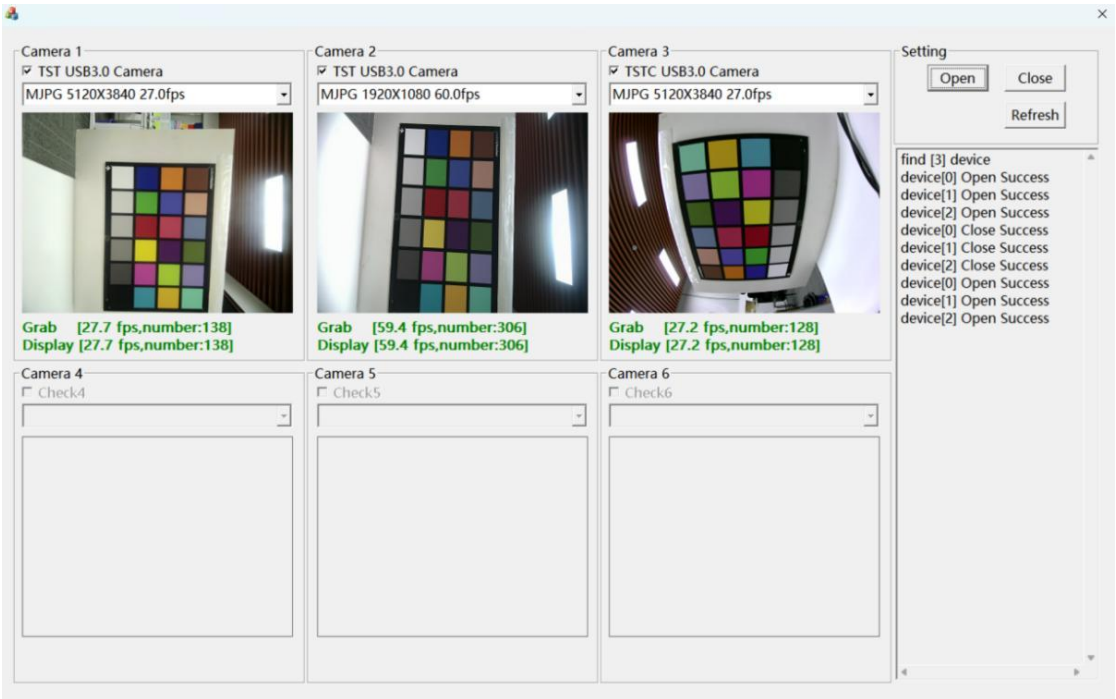



注意：フレームレートが動的に変動する場合は、Nori_Xvision_SetCameraTerminalControl を使用して UVC Camera Terminal の「低照度補正（Low Light Compensation）」を無効にするか、USB ビデオプレビューソフトウェア AMCAP で当該項目のチェックを外してください。



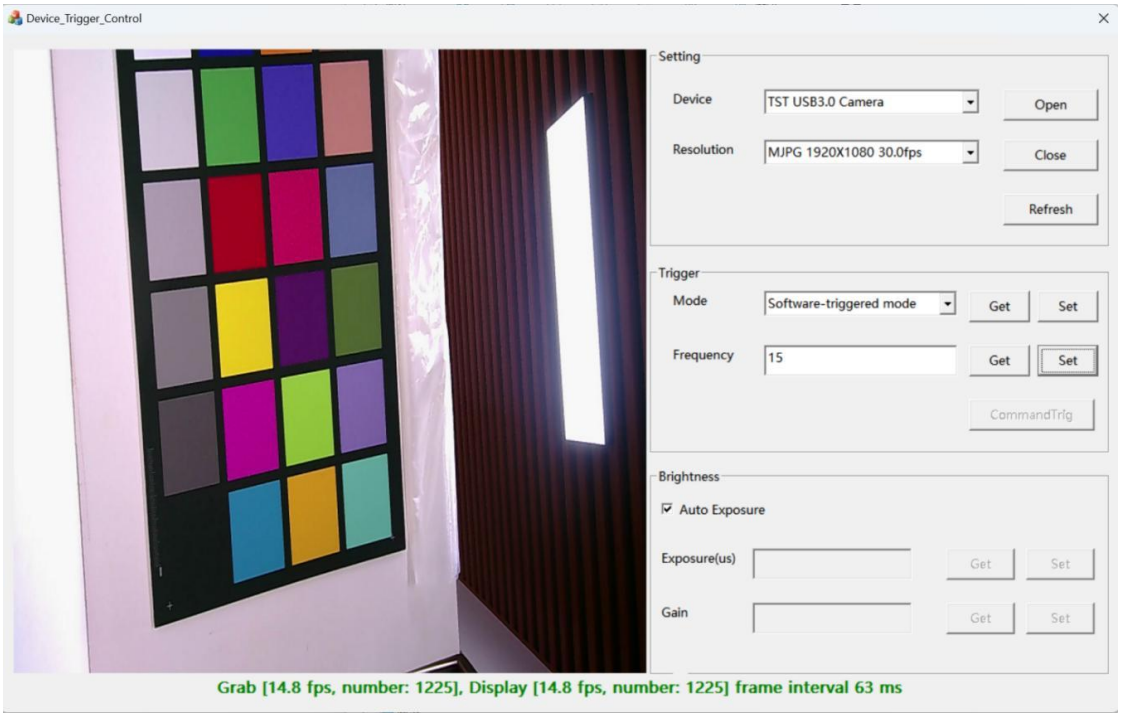

### 4.1.18 GrabImage_Multi_Display

説明：複数のカメラから同時に映像ストリームを取得する UI サンプル



### 4.1.19 Device_Trigger_Control

説明：カメラのトリガー機能、露光時間、ゲインを制御する UI サンプル



### 4.1.20 Firmware_Control

説明：ファームウェア情報の取得と変更

```
***fopen sf.bin success***
src_info.idVendor:      0x1bcf
src_info.idProduct:     0x2d50
src_info.bcdDevice:     13.02.00
src_info.iManufacturer: TSTC
src_info.iSerialNumber: 01.00.000
src_info.iProduct:      TSTC USB3.0 Camera
src_info.AudioProduct:  TSTC USB3.0 Camera
src_info.EXID:
src_info.FwType:        115


*** user info write to firmware***


*** read new firmware info***
user_info_get.idVendor:      0x1bcf
user_info_get.idProduct:     0x2d50
user_info_get.bcdDevice:     13.02.00
user_info_get.iManufacturer: Manufacturer Test
user_info_get.iSerialNumber: SerialNumber Test
user_info_get.iProduct:      Product Test
user_info_get.AudioProduct:  AudioProduct Test
user_info_get.EXID:          EXID Test
***save new firmware***

***save user.bin success***
Press a key to exit.
```

## 4.2 C#

### 4.2.1 Device_Check

説明：ユーザー定義のハンドシェイクのサンプル

```
Nori_Xvision_Init return [0x0]
find [1] device

device[0] device info
vid: 0x1BCF
pid: 0x2D50
bcd: 0x0264
iProduct: atom fluid camera
iManufacturer: TST
iSerialNumber: 01.00.000

device[0] version info
SDKVersion: 10.00.09
DeviceType: Gen60
ISPVersion: 13.00.00
FPGAVersion: Raw12-20251124
Enter 8 bytes in hexadecimal, separated by spaces:
** ** ** ** ** ** ** **
Nori_Xvision_DeviceCheck return [0x0]
Press a key to stop stream.
Press a key to exit.[2026-01-29 11:12:29:775], frame index[00080]
```

### 4.2.2 Device_Printf

説明：カメラのデバイス情報を取得するサンプル

```
Nori_Xvision_Init return [0x0]
find [1] device

device[0] device info
vid: 0x1BCF
pid: 0x2D50
bcd: 0x0510
iProduct: TST USB3.0 Camera
iManufacturer: TST
iSerialNumber: 01.00.000

device[0] version info
SDKVersion: 10.00.08
DeviceType: Gen60
ISPVersion: 13.00.00
FPGAVersion: NULL
Press a key to exit.
```

### 4.2.3 Grab_Callback

説明：コールバック関数を用いてカメラの映像ストリームを取得するサンプル

```
Nori_Xvision_Init return [0x0]
find [1] device

device[0] device info
vid: 0x1BCF
pid: 0x2D50
bcd: 0x0510
iProduct: TST USB3.0 Camera
iManufacturer: TST
iSerialNumber: 01.00.000

device[0] version info
SDKVersion: 10.00.08
DeviceType: Gen60
ISPVersion: 13.00.00
FPGAVersion: NULL
Press a key to stop stream.
Frame callback: time[2026-01-12 19:52:15:481], frame index[00073]
```

### 4.2.4 Grab_Image

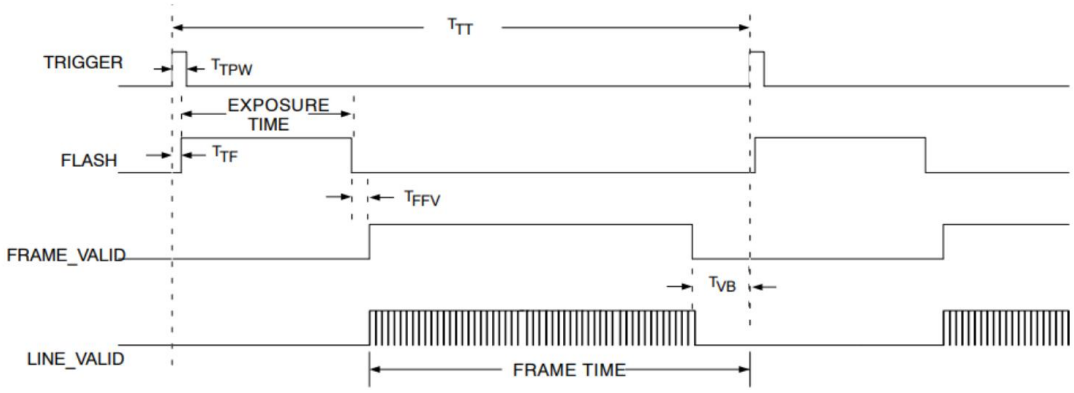
説明：映像ストリームを能動的に取得するサンプル

```
Nori_Xvision_Init return [0x0]
find [1] device

device[0] device info
vid: 0x1BCF
pid: 0x2D50
bcd: 0x0510
iProduct: TST USB3.0 Camera
iManufacturer: TST
iSerialNumber: 01.00.000

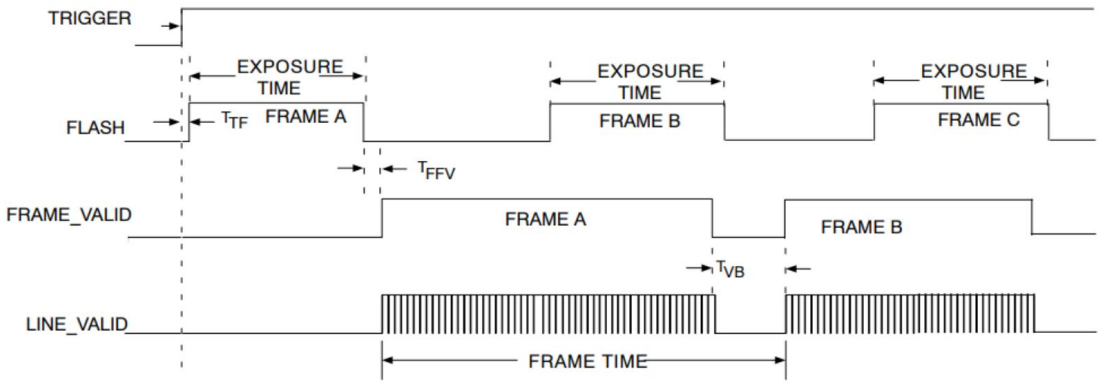
device[0] version info
SDKVersion: 10.00.08
DeviceType: Gen60
ISPVersion: 13.00.00
FPGAVersion: NULL
Press a key to stop stream.
Width:2592 Height:1944 fps:59.4 FrameLen:712958 FrameNum:266
```

# 5 外部パルストリガのタイミング




1) 露光およびフラッシュの遅延時間 $T_{TF}$ は、1.456 ms です。
2) トリガパルス幅 $T_{TPW}$ の最小値は、3 ms です。
3) トリガ周期 $T_{TT}$ は、外部トリガ信号の周波数によって決まります。
4) フラッシュ信号（Flash）はセンサーから出力されます。その有効期間は露光時間と等しくなります。
5) 露光時間（Exposure）は手動で設定する必要があります。「露光制御」を参照してください。露光が完了すると、フレームの出力が開始されます。

# 6 外部自動トリガのタイミング




1) 露光およびフラッシュの遅延時間 $T_{TF}$ は、1.456 ms です。
2) フラッシュ信号（Flash）はセンサーから出力されます。その有効期間は露光時間と等しくなります。
3) 露光時間（Exposure）は手動で設定する必要があります。「露光制御」を参照してください。露光が完了すると、フレームの出力が開始されます。

図中の表記：TRIGGER＝トリガ信号、FLASH＝フラッシュ信号、EXPOSURE TIME＝露光時間、FRAME TIME＝フレーム時間、FRAME_VALID＝フレーム有効信号、LINE_VALID＝ライン有効信号、FRAME A／B／C＝フレーム A／B／C

# 7 ハードウェア仕様

PCB 端子のピンのシルク印刷表示と、その定義を以下に示します。

| ピン番号 | 定義 | 入出力 | 説明 |
|---|---|---|---|
| 1 | GND | | グラウンド |
| 2 | Trigger | 入力 | ハードウェアトリガ入力、1.8 V 基準 |
| 3 | Strobe | 出力 | フラッシュ信号出力、1.8 V 基準 |

## 接続方法

1) 外部トリガ入力には、2 番ピン（Trigger）と 1 番ピン（GND）を使用してください。

2) フラッシュ信号出力には、3 番ピン（Strobe/Flash）と 1 番ピン（GND）を使用してください。