

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Visit our website - *Click HERE*



APX-3302
User's Manual

AVAL DATA CORPORATION

Downloaded from www.Manualslib.com manuals search engine

Notes:

1. Reproduction of the contents of this manual in part or in whole without the permission of the copyright holder is prohibited.
2. Contents of this manual are subject to change without notice.
3. We made assurance doubly sure on the contents of this manual. However, if you find something questionable, incorrect, or omitted, please contact the shop of purchase.
4. Please note that AVAL DATA assumes no responsibility for any effects resulting from the application notwithstanding the article 3 above.

Copyright 2012 AVALDATA CORPORATION
Manufacturer names and product names used in this manual are trademarks or registered trademarks of each company.

AVAL DATA CORPORATION

Downloaded from www.Manualslib.com manuals search engine

Before Reading User's Manual

Following symbols and signal words are used in this manual to ensure that our customer can use this product safely and to prevent the harm to persons or property. Fully understand the meanings of these symbols before reading the body of the manual.

WARNING:

Indicates a potentially hazardous situation that can result in death or serious injury if you do not follow instructions.

CAUTION:

Indicates a potentially hazardous situation that can result in minor/moderate injury or property damage if you do not follow instructions.

	**Indicates a potentially hazardous situation that can result in electric shock if you do not follow the instruction.**
	**Indicates a potentially hazardous situation that can result in injury if you do not follow the instruction.**
	**Indicates a potentially hazardous situation that can result in heat-induced disorder if you do not follow the instruction.**
	**Instructs the operator to pull out the plug of power cable without fail for safety.**
	**Indicates a potentially hazardous situation that can result in smoking or fire if you do not follow the instruction.**

AVAL DATA CORPORATION

Downloaded from www.Manualslib.com manuals search engine

⚠ CAUTION:

◆♦Do not use this product in a watery or humid place. Otherwise it can cause electric shock.

◆ Do not hold the plug of equipment's power cord with wet hand when connecting it to AC outlet. Otherwise it can cause electric shock.

◆ Do not touch this equipment with wet hand when power cord is connected to AC outlet. Otherwise it can cause electric shock and damage to the equipment.

◆ Pull out the plug of power cord from AC outlet before installing/removing peripheral device. Installing/removing peripheral device with power cord connected to AC outlet can cause electric shock.

◆ Connect the plug of equipment's power cord to an independent AC outlet with a ground terminal. No connection or loose connection of ground can cause electric shock or malfunction. Many loads on one AC outlet can cause the outlet to overheat and result in fire.

◆ In the case of failure or abnormality, stop the use immediately and let us know at the specified contact address.

◆ Never repair or modify the product by yourself.

AVAL DATA CORPORATION

Downloaded from www.Manualslib.com manuals search engine

Table Of Contents

Chapter 1 Precautions 1
1.1 Handling of the Product 1
1.1.1 Place to use or store the product 1
1.1.2 Cautions for use of product 2
1.2 After-Sales Service 3
1.2.1 User's registration 3
1.2.2 Customer Support 4
1.2.3 User's manual 4
1.2.4 Serial number 5
1.2.5 Dealing with occurrence of failure or abnormality 5
1.2.6 Before connect us 6
1.2.7 Repair and maintenance service 6
1.2.8 In the case of transferring the product to a third party 7
1.3 Disposal Method of the Product 7
Chapter 2 General Description 8
2.1 Product Summary 8
2.2 Feature 8
2.3 Specifications 9
2.4 Block Diagram 10
2.5 Front Panel 11
2.6 Connector Pin Assignment 12
2.6.1 CameraLink connector CN1-CN2 12
2.6.2 Expansion connector CN6 13
Chapter 3 Settings and Interface 14
3.1 Hardware initial setting 14
3.1.1 SW1 setting 15
3.1.2 SW2 setting 15
3.1.3 SW3 setting 15
3.1.4 CN7, CN8 setting 15
3.1.5 Mount APX-3302 into PCI Express slot 16
3.1.6 Cable connection 17
3.2 Software initial setting 18
3.2.1 What is SDK-AcapLib2 18
3.2.2 SDK-AcapLib2 installation procedure 19
3.2.3 Serial communication driver (virtual COM driver) 23
3.2.4 Registration Tool 24

AVAL DATA CORPORATION

Downloaded from www.Manualslib.com manuals search engine

3.2.5 Firmware update tool 25
3.2.6 Uninstallation of each software 26
3.2.7 Installation trouble shooting 27
Chapter 4 AcapLib2Viewer Image acquisition 28
4.1 What is AcapLib2Viewer 28
4.2 What is ini file 28
4.3 Image acquisition using AcapLib2Viewer 29
Chapter 5 Programming guide 34
5.1 AcapLib2 flow chart 34
5.2 Programming example 35
5.2.1 Snap (one shot snap) 35
5.2.2 Grab (Streaming) 37
5.3 Sample capture application 39
5.3.1 SimpleApplication 39
5.3.2 OtherSample 40
Chapter 6 Function 42
6.1 CameraLink Configuration 42
6.1.1 How to configure for 2TAP output cameras 43
6.1.2 How to configure for 3TAP output cameras 44
6.2 Power Over CameraLink 45
6.3 Camera exposure control signal CC 47
6.4 Serial interface 49
6.5 External signal input 51
6.6 The noise rejection function of external input signal 56
6.7 Strobe output 57
6.8 Timing adjustment of CC output and strobe output 57
6.9 Synchronizing signal between frame-grabbers (SYNCLT) 60
6.9.1 Configuration steps using SYNCLT (Area sensor) 61
6.9.2 Configuration steps using SYNCLT (Line sensor) 61
6.10 Encoder input 69
6.10.1 Encoder input circuit 69
6.10.2 Encoder count timing 70
6.11 Encoder operation method 71
6.11.1 Relative position encoder 71
6.11.2 Absolute position encoder single point 73
6.11.3 Absolute position encoder multi point 76
6.12 MEGA-FIFO 84
6.13 Interruption 87
6.13.1 Event usage 88
6.13.2 Callback function usage 90

Downloaded from www.Manualslib.com manuals search engine

6.14 Notes for interrupt event, callback function 94
Chapter 7 Error codes 96
7.1 Common error codes 96
7.2 Board error code 99
7.3 Extended error code (Windows) 101
7.3.1 Extended error codes returned by driver 102
7.3.2 Extended error codes related to buffer 104
Firmware Update History 105
Revision History 106
Inquiry 107

Downloaded from www.Manualslib.com manuals search engine

Chapter 1 Precautions

Thank you for purchasing「APX-3302」
Please read this user's manual thoroughly to ensure the proper use of this product.

Cautions:
If you insert our products to x 16-lane port that is specified for the graphics card on the PC or motherboard, it may not work correctly. These lanes are not in operation guarantee because these lane's specification are not clear, and might be made to be able to maximize the capabilities of the graphics card.

1.1 Handling of the Product

1.1.1 Place to use or store the product



■Do not use this product at a place where it can be subject to spillage of chemicals etc. Spilled chemical on this product can cause short-circuit and result in fire, failure or deformation.

■ Do not use or store the product in a place where it can subject to spillage of liquids such as water and oil, a place exposed to steam, or a place of high humidity. Otherwise it can cause failure of the product.

■ Use or store the product in a flat place with sufficient strength. Do not apply vibration or impact on the product



■ Do not use or store the product in a place exposed to direct sunlight or a place close to fire or heaters such as stoves. Otherwise, it can cause failure or deformation.

■ Do not use or store the product in a dusty place.

■ Do not use the product at a place close to TV, radio or cordless phone. Otherwise, TV, radio or cordless phone can sound noise.

■Do not use or store the product at a place close to the equipment generating magnetism or electric wave. Otherwise, it can cause failure of the module.

■ Do not put heavy goods on the product. Otherwise it can cause failure of the module.



■ Do not pour liquid such as water in the product. Otherwise it can cause electric shock.

■ Static electricity can cause failure of the product. Take sufficient preventive measure against static electricity such as discharging static electricity of human body when handling the module.

Downloaded from www.Manualslib.com manuals search engine

■ Be careful not to get your hand injured by the front or rear surface of the module when handling the product.

1.1.2 Cautions for use of product

Cautions

Be sure to unplug the power code and turn off the power of the system before installing or removing this product. Otherwise it can cause electric shock.

■ Precautions for installing procedure of the product

a) Check the switch on the product is set correctly

b) After confirming the installation of the product in the system, turn on the power of the system and check the operation

■ Precautions for removing procedure of the product

a) Be sure to turn OFF the power of the system before removing the product, Otherwise it can cause failure

b) Right after turn OFF the power of the system, the product may be heated to a high temperature. please handle it with care.

* Refer to the manual for the use of the system

Downloaded from www.Manualslib.com manuals search engine

1.2 After-Sales Service

1.2.1 User's registration

User's registration is required to use the after sales service.
Otherwise, you may not obtain the maintenance service from AVAL DATA.
Select one of the following methods for registration.

Registration on the Website :
A maintenance service registration page is in the support homepage of our homepage.
A page for maintenance service registration in the support page on AVAL DATA's homepage.
http://www.avaldata.co.jp/support/regist_user.html

Registration by FAX :
Fill out a FAX registration form for maintenance service that is included with shipment.
FAX: 042-732-1032.

AVALDATA handles personal information appropriately and safely.
The handling of privacy policy is on the Website below:

http://www.avaldata.co.jp/policy/privacy.html

Downloaded from www.Manualslib.com manuals search engine

1.2.2 Customer Support

For the products inquiries, please contact to the support line below.

Customer service
AVAL DATA CORPORATION (URL http://www.avaldata.co.jp/) Head Quarters / Machida OFFICE SALES DEPARTMENT 1-25-10, Asahimachi, Machida-shi, Tokyo TEL 042-732-1030 FAX 042-732-1032 Business hours ：Monday-Friday 9:00AM-17：00PM(except for a national holiday) E-mail sales@avaldata.co.jp

For the Technical inquiries, please contact to the support line below.

Technical support Line
E-mail support@avaldata.co.jp

1.2.3 User's manual

A printed manual is not included with the shipment. The manual is provided for download on the website or provided for the purchase of the Software Development Kit(SDK).
It is based on concepts of environmental consideration and providing continuous updates in the ever-changing industry.
AVAL DATA appreciate your understanding.

Download from the Website. * Membership registration is required. http://www.avaldata.co.jp/support/manual.html

In addition, request a Support Line when it is difficult to download of the manual from the Website. AVAL DATA will send you it in electronic media, such as CD-R.
(The customer is responsible for the shipping cost of the items.)

Downloaded from www.Manualslib.com manuals search engine

1.2.4 Serial number

The serial number of the product is printed on the label on the main unit, the warranty certificate, the box, given each.
Location of the serial number label of the product is shown below.

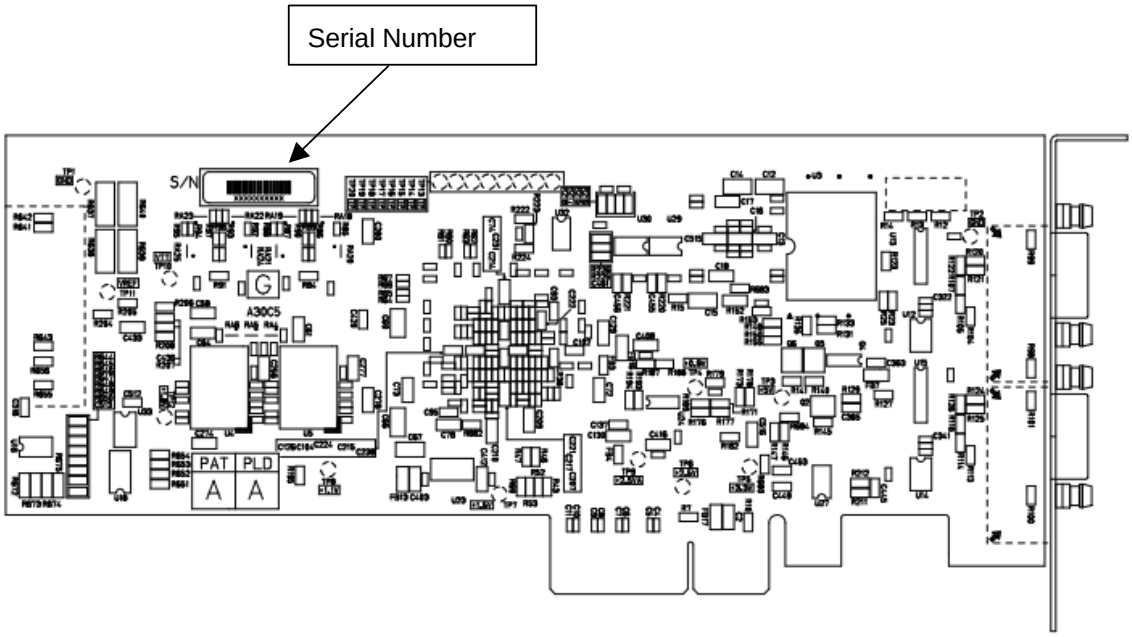


Figure1-1: Location of the serial number label

1.2.5 Dealing with occurrence of failure or abnormality

When the following failure or abnormality occurred on the module, turn off the power of the system immediately and pull out the plug of power cord from AC outlet.

A) When the system smokes or gives abnormal smell
B) When the system becomes too hot to touch by hand
C) When the system sounds abnormal noise
D) When the application that had been functioning correctly malfunctions

Contact to the company of purchase or customer service when failure or abnormality occurred.

Downloaded from www.Manualslib.com manuals search engine

1.2.6 Before connect us

Check the followings in advance when consulting with us for the failure or abnormality of the product

A) Check the following in advance when consulting with us for the failure or abnormality of the product
B) Have the Warranty certificate of the module at hand
C) Prepare a simple note on the state of failure or abnormality.
D) Prepare a note on the environment where the module is used (Hardware/ Software)
E) Have the manual of the module at hand.

1.2.7 Repair and maintenance service

・Warranty certificate
In case of failure of the product, we will repair the product in accordance with the contents of the certificate. Refer to the certificate for details.

・Repair parts
The minimum holding period of repair parts is 5 years after discontinuation of the product.

Downloaded from www.Manualslib.com manuals search engine

1.2.8 In the case of transferring the product to a third party

A) **User who transfers the product**

Following conditions shall be satisfied when the product is transferred to a third party.

- All attachments to the module (instruction manual, warranty certificate, accessories, etc.) shall be transferred. No reproduction shall be kept.
- Conditions related to transfer and alienation described in the condition to use that is attached to each software shall be satisfied.
- The software not permitted for its transfer and alienation shall be deleted before transfer.

B) **User to whom the product is transferred**

Be sure to inform us of the items necessary for changing the contents of user registration for the case when we send information to our users.

Items necessary for registration:

Warranty certificate number of the main unit, model name, and serial number.

The name of company you purchased the used product, its address and phone number

1.3 Disposal Method of the Product

Observe the related ordinances of the local government when discarding the module.
Consult with the local government for details.

Downloaded from www.Manualslib.com manuals search engine

Chapter 2 General Description

General description of APX-3302

2.1 Product Summary

APX-3302 is a LowProfile type PCI-Express frame grabber which supports CameraLink interface.
On-board mini SDR26 pin connector allows connecting up to two independent CameraLink Base Configuration cameras for each. It has an I/O interface circuit compatible with APX-3312 that provides image acquisition control in conjunction with external devises.

Cautions:
If you insert our products to x 16-lane port that is specified for the graphics card on the PC or motherboard, it may not work correctly. These lanes are not in operation guarantee because these lane's specification are not clear, and might be made to be able to maximize the capabilities of the graphics card.

2.2 Feature

Features of APX-3302

- High speed image acquisition by using PCI Express x4 bus (Gen2 2.5GT/s)
- Capable connection to the line camera or area camera that support the CameraLink standard for Base Configuration
- Capable of connecting up to two cameras
- Capable of synchronous / asynchronous image acquisition between camera
- PoCL function to supply power to the camera
- On-board APX-3312 compatible external I/O interface enables coordination with external device or generates interrupts
- Strobe timing output allows to notify the exposure timing to external lighting devices
- 256MB of on board MEGA-FIFO image buffer for each camera
- RoHS compliant

※MEGA-FIFO architecture

MEGA-FIFO architecture is a structure to store the image data from the camera in large capacity buffer on board temporarily, then it transfers the image data to host memory via the buffer.
Hardware offers fastest operation of buffer input by the camera.
(Base Configurationcamera:Max 85MHzx32bit＝340MByte/sec)
It is possible to acquire the image until the buffer capacity is full even the bandwidth of PCI-Express is lower then the frame rate of a camera.

Downloaded from www.Manualslib.com manuals search engine

2.3 Specifications

The specifications of APX-3302 are described as follows.

Item	Specification
Image Input	Two independent Camera Link Base Configuration CameraLink Clock 20MHz to 85MHz
Camera Control	CC output Shaft Encoder Interface On-board UART (communications to camera)
Power over Camera Link (PoCL)	Supported
I/O	Two TTL (trigger input/general-purpose input) Four RS422 (encoder/trigger input) Two Open-Collector (strobe/general-purpose output)
System bus	x4 PCI Express™ 2.0 Gen2 (2.5 GT/s)
MEGA-FIFO	DDR2 SDRAM 256MB
Interrupt	Start of acquisition End of DMA Falling edge of general-purpose input
Power Requirements	+12V 0.7A Max(PoCL OFF) +3.3V 0.3A Max
Environment Requirements	Temperature: 0ºC to 50ºC Humidity: 35% to 85% (non-condensing)
Dimension	167.65mm x 68.9mm (Low profile x4 PCI Express™ card)
Weight	100g
Software Support	Software Development Kit (SDK-AcapLib2) 32-bit and 64-bit Windows platform
RoHS compliant	Yes

Table2-1:Specifications

Downloaded from www.Manualslib.com manuals search engine

2.4 Block Diagram

The block diagram of APX-3302

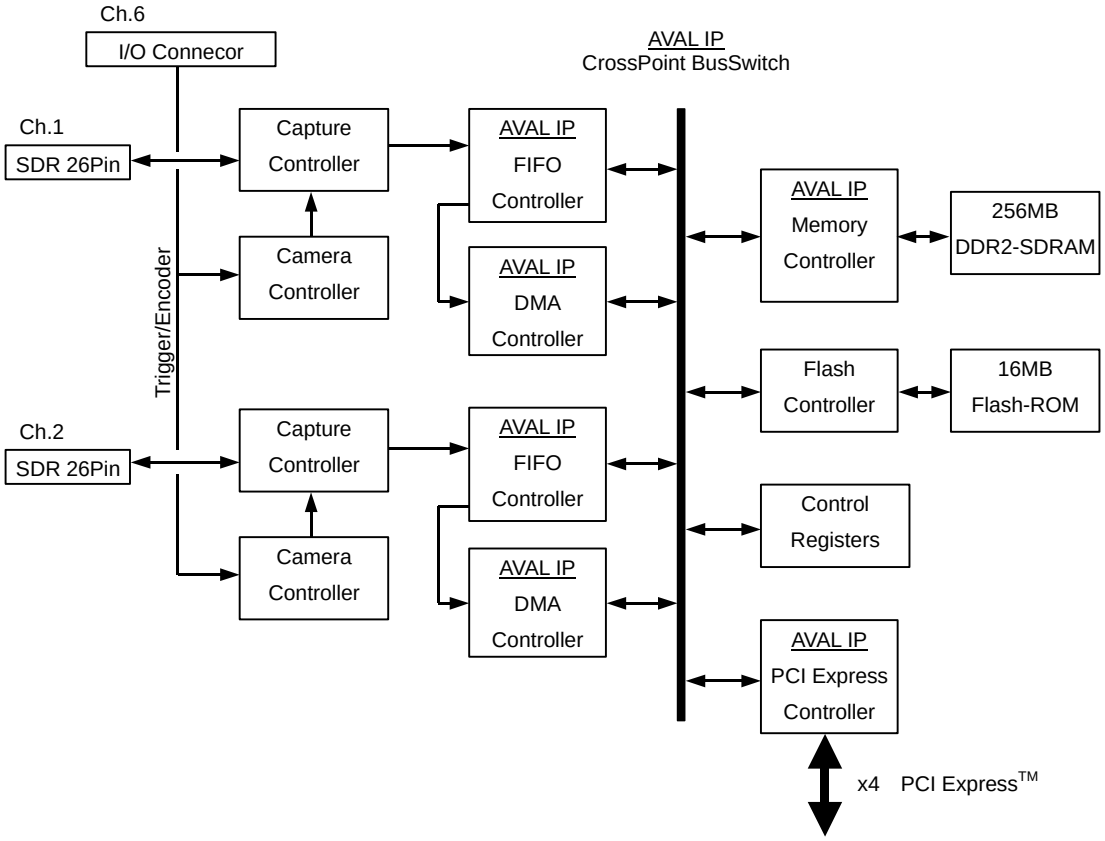


Figure2-1:Block Diagram

Downloaded from www.Manualslib.com manuals search engine

2.5 Front Panel

The front panel view of APX-3302 is shown below.

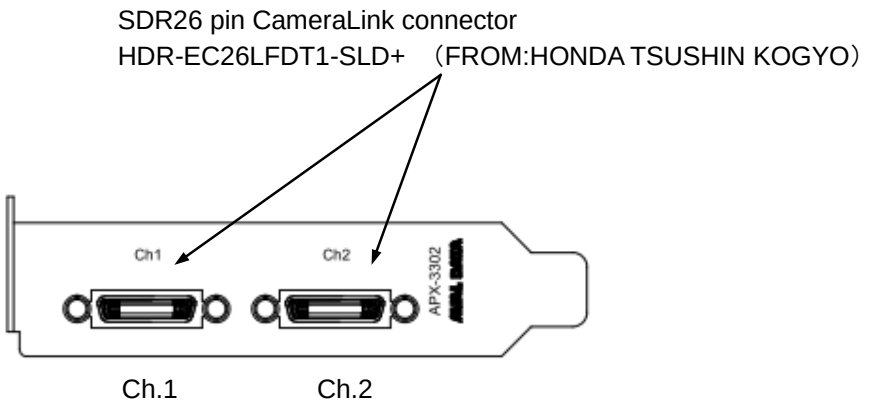


Figure2-2:Front panel

An optional PCI standard front panel is provided, replace a front panel according to the type of PC. Panel and board are connected with four of screw lock CameraLink connector.

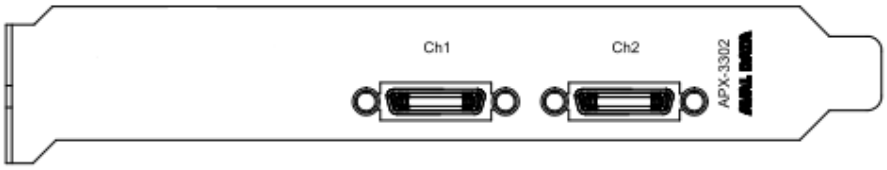


Figure2-3:Front panel (Standard)

Downloaded from www.Manualslib.com manuals search engine

2.6 Connector Pin Assignment

This chapter describes connector on APX-3302.

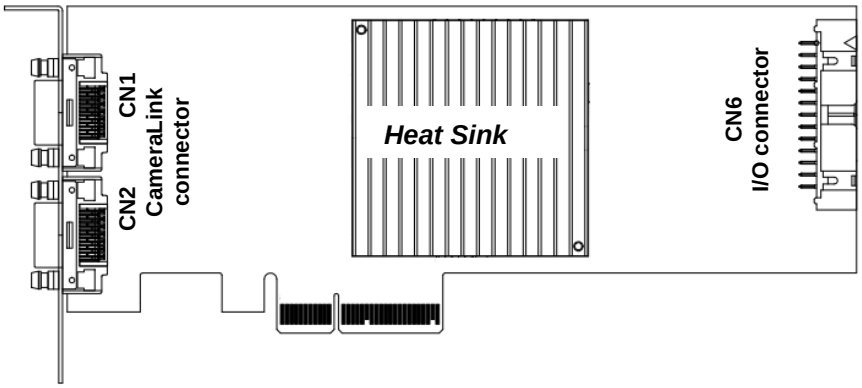


Figure2-4: Connector layout drawing

2.6.1 CameraLink connector CN1-CN2

CameraLink connector specifications are shown below. Compliant with CameraLink standard
Manufacturer : HONDA TSUSHIN KOGYO
Model : HDR-EC26LFDT1-SLD+

Pin	Signal	Pin	Signal
1	Inner_Shield/+12V	14	Inner_Shield
2	CC4-	15	CC4+
3	CC3+	16	CC3-
4	CC2-	17	CC2+
5	CC1+	18	CC1-
6	SerTFG+	19	SerTFG-
7	SerTC-	20	SerTC+
8	X3+	21	X3-
9	X_CLK+	22	X_CLK-
10	X2+	23	X2-
11	X1+	24	X1-
12	X0+	25	X0-
13	Inner_Shield	26	Inner_Shield/+12V

Table2-2: CameraLink connector pin assignment

Downloaded from www.Manualslib.com manuals search engine

2.6.2 Expansion connector CN6

The expansion connector is assigned various I/O signal to connect external devices.
Specifications are shown below.
Manufacturer : OMRON
Model : XG4C-2634

Pin	Signal	Description	Pin	Signal	Description
1	GND	Ground	2	GND	Ground
3	ExTrg_1	ch1 external trigger input (TTL)	4	ExTrg_2	ch2 external trigger input (TTL)
5	Enc. A+	Encoder A phase input (RS422+)	6	Enc. A-	Encoder A phase input (RS422-)
7	Enc. B+	Encoder B phase input (RS422+) or ch1 external trigger input (RS422+)	8	Enc. B-	Encoder B phase input (RS422-) or ch1 external trigger input (RS422-)
9	Enc. Z+	Encoder Z phase input (RS422+) or ch2 external trigger input (RS422+)	10	Enc. Z-	Encoder Z phase input (RS422-) or ch2 external trigger input (RS422+)
11	Diff+	External differential trigger input (RS422+)	12	Diff-	External differential trigger input (RS422-)
13	GND	Ground	14	GND	Ground
15	N.C		16	N.C	
17	N.C		18	N.C	
19	GP_IN1	Inputs (TTL) in general	20	GP_OUT1	Outputs (TTL) in general
21	SYNCLT_OUT	The synchronizing-signal output (TTL) between frame-grabbers	22	GND	Ground
23	GP_OUT2	Outputs (TTL/OC) in general	24	SYNCLT_IN	The synchronizing-signal input (TTL) frame-grabbers
25	IN_RES1	Inputs (TTL) in general	26	OUT_RES1	Outputs (TTL/OC) in general

Table2-3: Expansion connector is assignment

AVAL DATA prepares from expansion connector to expansion panel cable ”CBL-Z091A” wired PCI standard panel.

OUT_RES1 cannot be used by APX-3302.

Downloaded from www.Manualslib.com manuals search engine

Chapter 3 Settings and Interface

This chapter describes initial settings for Hardware and software of APX-3302.

3.1 Hardware initial setting

Set up the jumper and switch correctly before installing this product to the PC or devices.

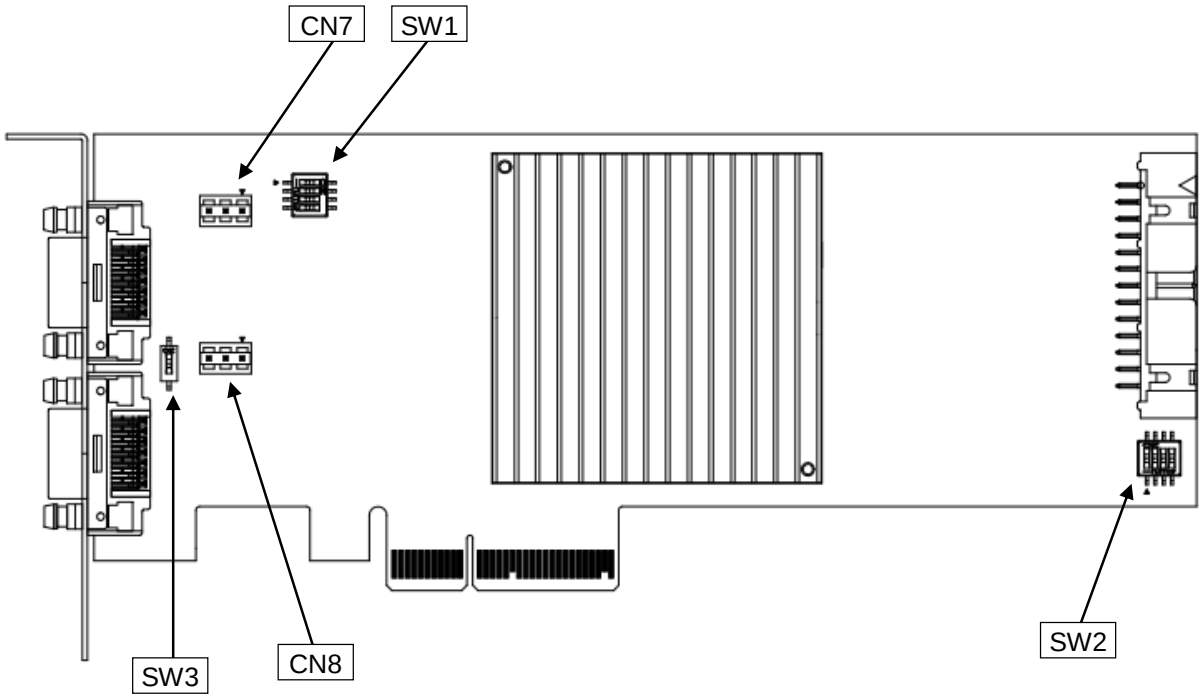


Figure3-1: jumper, switch layout drawing

Be sure to turn OFF the power of the system before change jumper or switch settings.

Downloaded from www.Manualslib.com manuals search engine

3.1.1 SW1 setting

SW1 should be used in OFF anytime.

3.1.2 SW2 setting

SW2 sets up Pull-Up as a general-purpose output.
Cut Pull-Up resistor off by OFF then connect Pull up by ON
Pull-Up resistor: 330Ω
The relation between the switch and signals is shown below

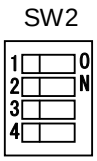


Figure3-2: SW2 setting

SW2-1: General-purpose output -1 GP_OUT1
SW2-2: Reserved General-purpose output-1 OUT_RES1
SW2-3: General-purpose output -2 GP_OUT2
SW2-4: (Unused)

SW2 is set as OFF, all Pull-up resistors are separated and are set as the open collector output in default setting.

3.1.3 SW3 setting

SW3 should be used in OFF anytime.

3.1.4 CN7, CN8 setting

Set up the validity or invalidity of PoCL function according to the channel of camera.
PoCL is valid in ON and is invalid in OFF.
CN7 corresponds to ch1 of camera and CN8 corresponds to ch2 of camera.

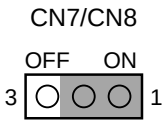


Figure3-3: CN7, CN8 setting

CN7 and CN8 are set as ON and PoCL function is valid in default setting.

* In order to use a PoCL function, it is necessary to control feeding power not only by this setting but also by software. Refer to [6.2 Power over CameraLink] for the details.

Downloaded from www.Manualslib.com manuals search engine

3.1.5 Mount APX-3302 into PCI Express slot

Please note following points in addition to the instructions of the system when mounting or removing this product to the system with the PCI Express connector.

- Notes
  a) Mount APC-3302 into PCI Express x4 or more slot.
  b) It is compliant with Gen1, Gen2.

- Notes for mounting APX-3302 into the system
  a) Make sure that all jumpers of APX-3302 are set correctly.
  b) Insert it into the PCI Express slot straight without adding the extra power at this time.
     Otherwise, it can cause of failure or damage the parts on the module by getting stuck with the input/output terminals assigned on the panel.
     Check a connector of module on the PCI slot and a connector of the system are connected correctly.
  c) After insert it into PCI slot, fix it with panel fixation screw tightly.
     Turn ON the power source of the system and check the operation after checking it is mounted into the system.

- Notes for removing APX-3302 from the system
  a) Be sure to turn OFF the power source of the system before removing it. it can cause of failure.
     Be cautious it becomes extremely hot soon after the power source OFF.
  b) Unfix panel fixing screw and remove it without touching the parts on a module and adding the extra power. It can cause damage to a module or the connector of the system.
  c) If it is getting stuck with the input/output terminals assigned on the panel when removing it, do not remove by force, it can cause of failure.

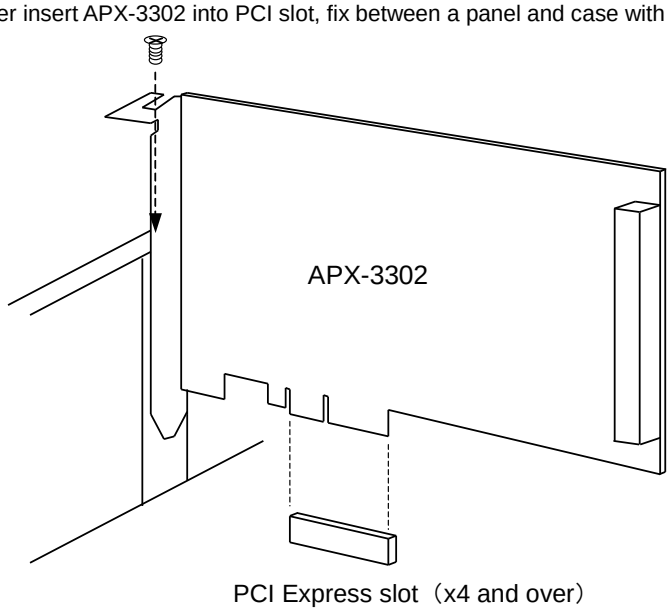


Figure3-4: Insert into PCI Express slot

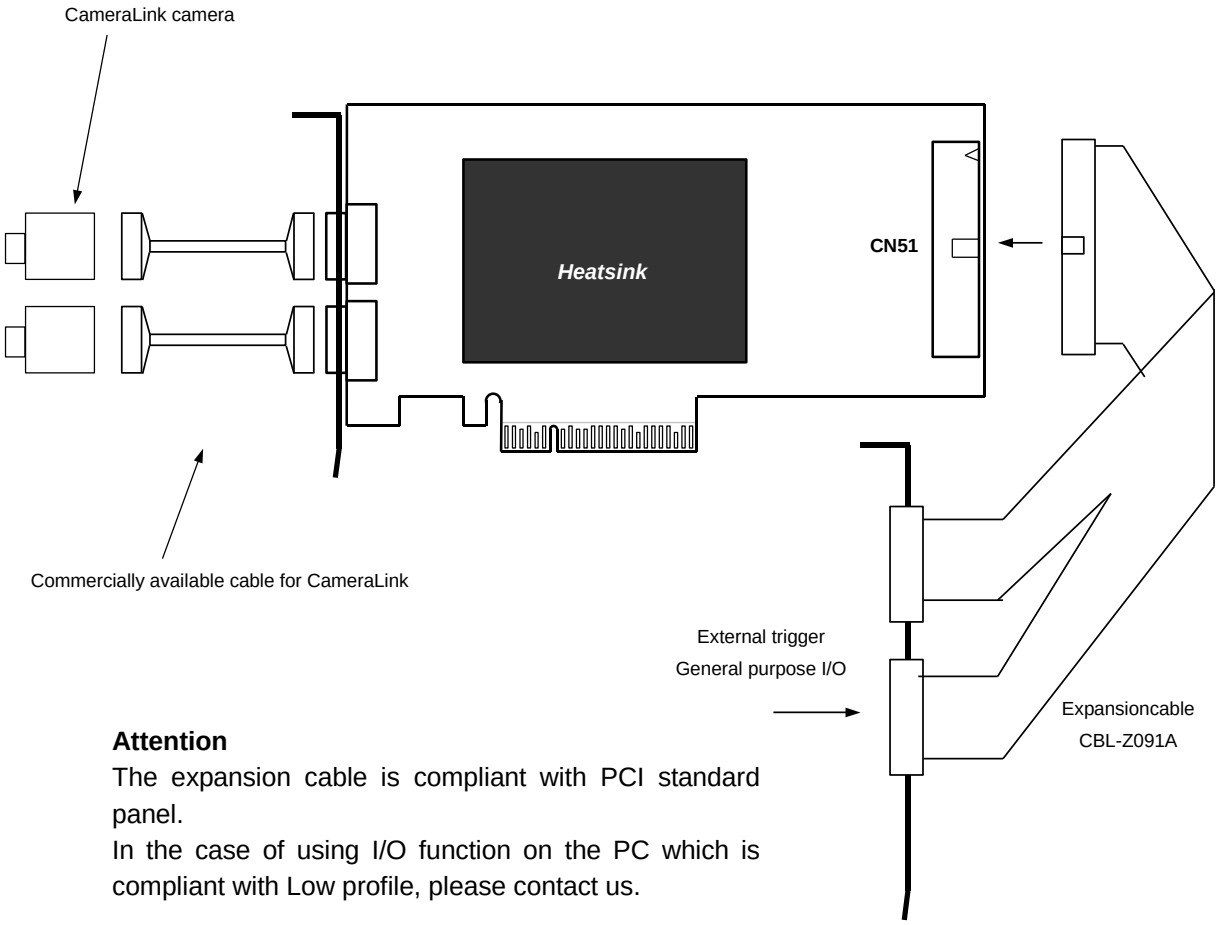
Downloaded from www.Manualslib.com manuals search engine

3.1.6 Cable connection

Description of basic connection method of APX-3302 is shown below.

- Connect between APX-3302 to camera using the cable which is compliant with CameraLink standard.

- In the case of connecting between APX-3302 to an external device, use expansion cable CBL-Z091A.
  Refer to the cable pamphlet for the detail specification of CBL-Z091A.



Attention
The expansion cable is compliant with PCI standard panel.
In the case of using I/O function on the PC which is compliant with Low profile, please contact us.

Figure3-5: Connection diagram of the power source connector and expansion cable.

Downloaded from www.Manualslib.com manuals search engine

3.2 Software initial setting

In order to control APX-3302, install SDK-AcapLib2.

3.2.1 What is SDK-AcapLib2

SDK-AcapLib2 (Software Development Kit AvalCaptureLibrary2) is a software package used for controlling many models of frame-grabbers including APX-3302.
The followings are contained in SDK-AcapLib2.

- Aval Capture Library 2 (hereafter referred as AcapLib2)
  AcapLib2 is a library which enables control of various models of frame-grabbers with a common interface.

- Drivers for various models of frame-grabbers
  The driver for various models of frame-grabbers including APX-3302

- Sample source
  Capture sample (VC++ / VC# / VB.NET) which supports any languages
  Sample for connecting various types of image-processing library

- Tool
  AcapLib2Viewer, virtual COM tool, firmware update tool, line profile tool, AZP-8100-01 (image-processing tool).

- Camera-setting (ini) file
  A file used for initial setting on frame-grabber side against the camera which has a connection record.

- Document
  Manuals, installation instructions, sample documents.

Downloaded from www.Manualslib.com manuals search engine

3.2.2 SDK-AcapLib2 installation procedure

Installation procedure of SDK-AcapLib2 is shown as below.

[Method 1]
The web pages as below is be displayed when a install disc is set to CD-ROM/DVD-ROM drive, [software] from a menu.



Figure3-6: Web page

Precautions before installation

■ Check whether the frame-grabber is mounted in the PC before starting installation.

■ Check **[DMA Master]** is selected as **[ON (YES)]** on a BIOS screen. (*)

■ When you log in, the user name should be "Administrator" or "**user name** authorized to install driver and others.

■ **Cannot use the standby (ACPI S1/S3 mode) which has power saving function.**
Turn sleeping state / hibernation off by using inside CD-ROM [Manual]-[Windows system standby cancellation procedure].

■ It does not support about for using 3G option in 32bit OS.
AVAL DATA recommends using it by removing **3G option**.

■ It is not supporting **Windows XP Mode** and **Windows XP compatibility mode** of Windwos7.

■ **If the another version is installed already, be sure to install SDK-AcapLib2 after uninstalling it**.
Refer to [3.2.6. Uninstallation of each software] for the procedure of uninstallation.

(*) Setting contents above may not included depending on the PC. If not, please disregard them.

Downloaded from www.Manualslib.com manuals search engine

[Method 2]
Click [install] at the side of [SDK-AcapLib2].

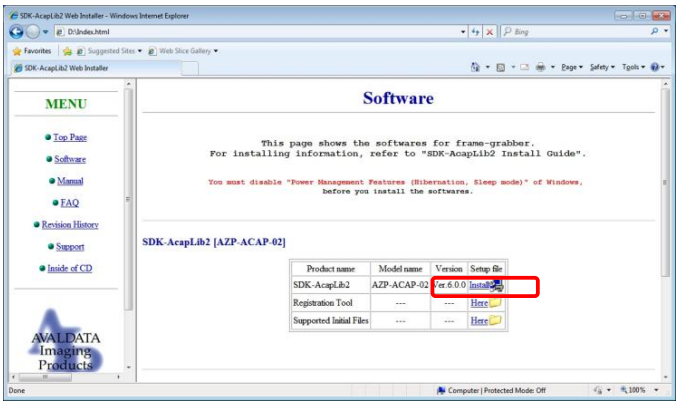


Figure3-7: Software

[Method 3]
Select [run] on [File download] is displayed.

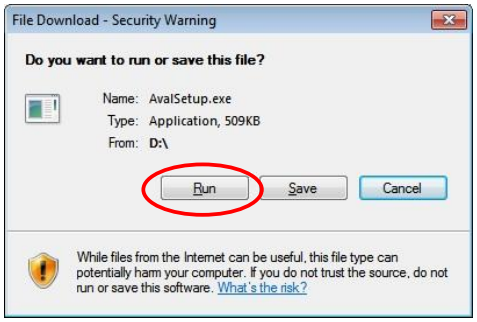


Figure3-8: Download of a file

[Method 4]
Click [Install].
*NOTE: If another version is already installed, [Uninstall] becomes effective.
You can continue to uninstall if the same version is used. However if the version is different, the warning will appear at the time.



Figure3-9: Menu

Downloaded from www.Manualslib.com manuals search engine

[Method 5]
Select [Browse] and chose other folder if you change the installation folder.
Select [OK] if you do not need to change any settings.

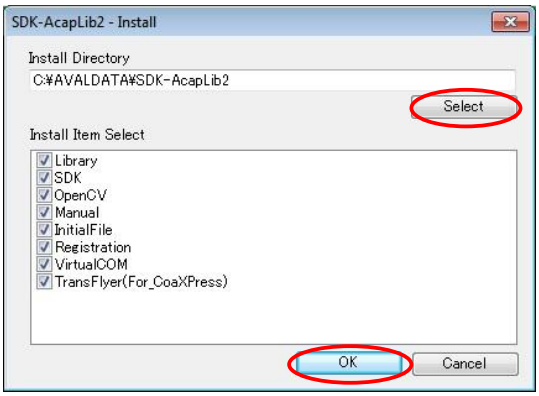


Figure3-10: Installation item

[Method 6]
When you check [Library] at installation selection, "Device Driver Installation Wizard" starts automatically.
Follow instruction s and continue installing.



Figure3-11: Library installation

[Method 7]
When you check [Registration] at installation selection, "AVALDATA Registration Tool" starts automatically.
When the product key is included, select [Product version], then input product key and select [OK].
Refer to [3.2.4 Registration Tool] for the details.

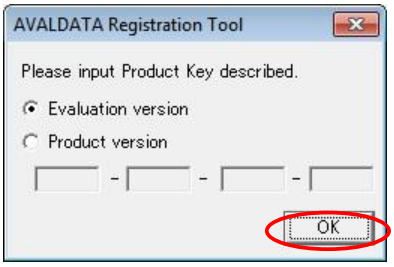


Figure3-12: Software registration

Downloaded from www.Manualslib.com manuals search engine

[Method 8]
When you check [TransFlyer (For_CoaXPress)] at installation selection, "AVALDATA TransFlyer" starts automatically. Follow instruction s and continue installing.
* It is not necessary to install this software other than CoaXPress frame-grabber.

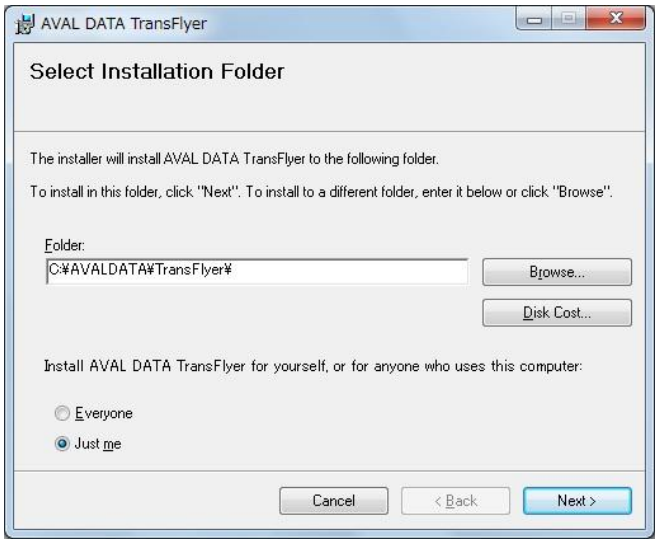


Figure3-13: TransFlyer installation

[Method 9]
Finish application by clicking " Finished"

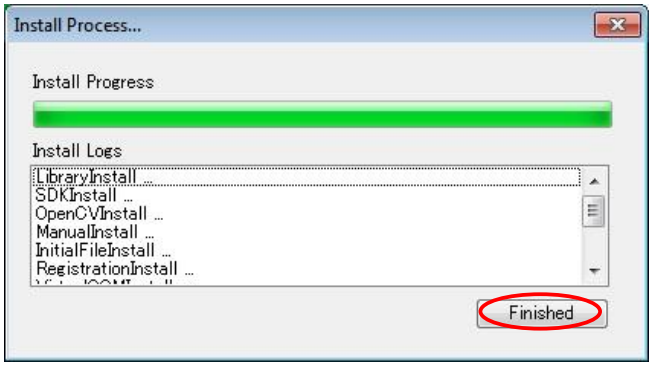


Figure3-14: Completion

Downloaded from www.Manualslib.com manuals search engine

[Method 10]
Click Quit then the installation is finished.

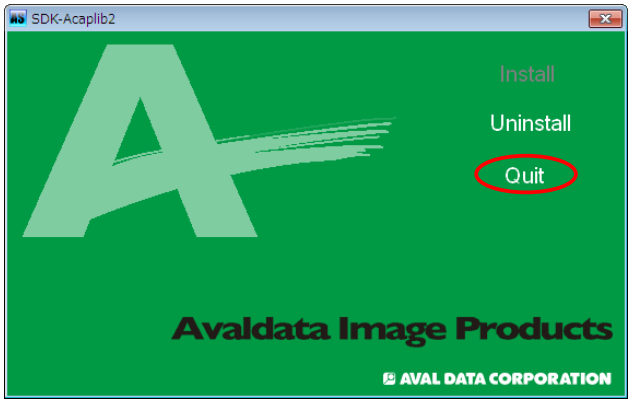


Figure3-15: Menu

Press [Quit] to finish the installation.

3.2.3 Serial communication driver (virtual COM driver)

Virtual COM is a software which has the function to assign each camera link port of AVALDATA's frame-grabber as a virtual COM port on Windows. It enables communicate from standard COM port on Windows to the camera.

Use virtual COM tool to assign virtual COM.
Execute from Start
[AVALDATA]=>[SDK-AcapLib2] =>[Tool]=>[VirtualCOM]=>[avdCOMmng]

Refer to a virtual COM tool user's manual(SDK-AcapLib2 VirtualComTool Manual (J2OM10016*).)for the installation or uninstallation procedure.

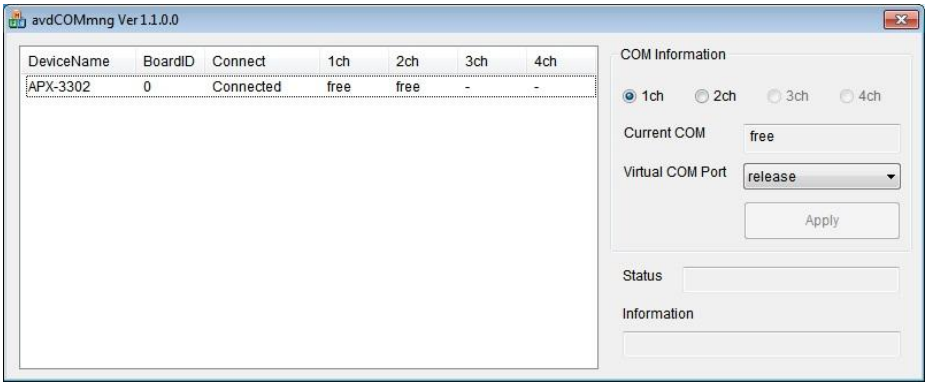


Figure3-16: Virtual COM tool

Downloaded from www.Manualslib.com manuals search engine

3.2.4 Registration Tool

When you using a library, it is necessary to perform registration to PC using "Registration Tool." There is no case that the function takes restriction when it is used without registration. However the message, which tells that registration is not completed, will appear when the function of specific library is called,



Figure3-17: Unregistered message

■ Registration method

Description of the registration method is shown below.

[Method 1]
Execute Start
[AVALDATA]=>[SDK-AcapLib2] =>[Tool]=>[Registration]=>[AvdRegistTool]

[Method 2]
Select [Product version]. then, enter 16 digit product key indicated on the yellow seal which attached to an install disc. Click [OK], when finished entering.

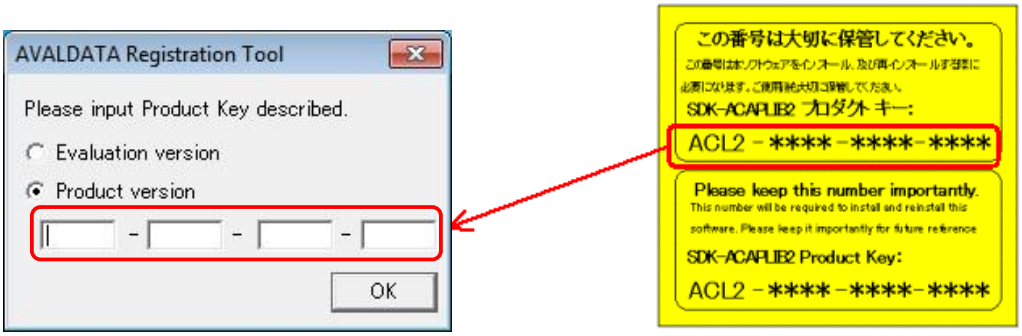


Figure3-18: Registration Tool and a product key

Downloaded from www.Manualslib.com manuals search engine

[Method 3]
Registration is completed if this message is appears.



Figure3-19: Registration success

3.2.5 Firmware update tool

Firmware update tool is the application for updating local FPGA firmware of APX-3302. It has the firmware update function as well as software switch setting function.

Execute Start
[AVALDATA]=>[SDK-AcapLib2] =>[Tool]=>[FirmwareUpdateTool]=>[FirmwareUpdateTool]

Refer to Firmware update tool user's manual (SDK-AcapLib2 FirmwareUpdateTool Manual(J3OM10026*) for the procedure of this application.

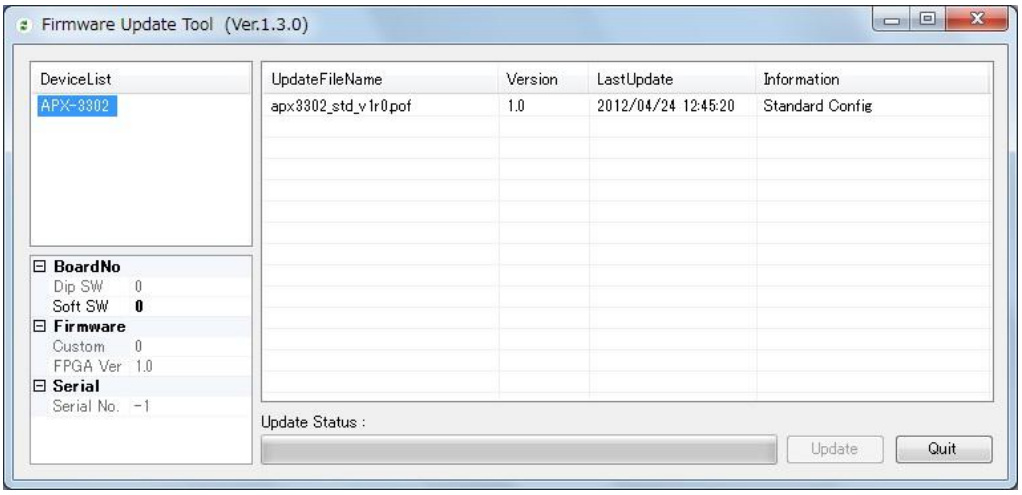


Figure3-20: Firmware update tool

Downloaded from www.Manualslib.com manuals search engine

3.2.6 Uninstallation of each software

■ Uninstalling Virtual COM Driver
[Method 1]
Execute [Start] =>[AVALDATA]=>[SDK-AcapLib2] =>[Tool]=>[VirtualCOM]=>[avdCOMmng]

[Method 2]
Cancel all assignment of COM. Refer to Virtual COM tool user's manual (SDK-AcapLib2 VirtualComTool Manual (J2OM10016*)) for the details.

[Method 3]
Select [Control Panel] - [Management Tool]-[Management of computer]-[Device manager]

[Method 4]
Right-click [AVALDATA Virtual COM Port (COM*)] in [Port (COM and LPT)], then select [delete]

[Method 5]
Click "OK" after "check of device deletion" box appeared.

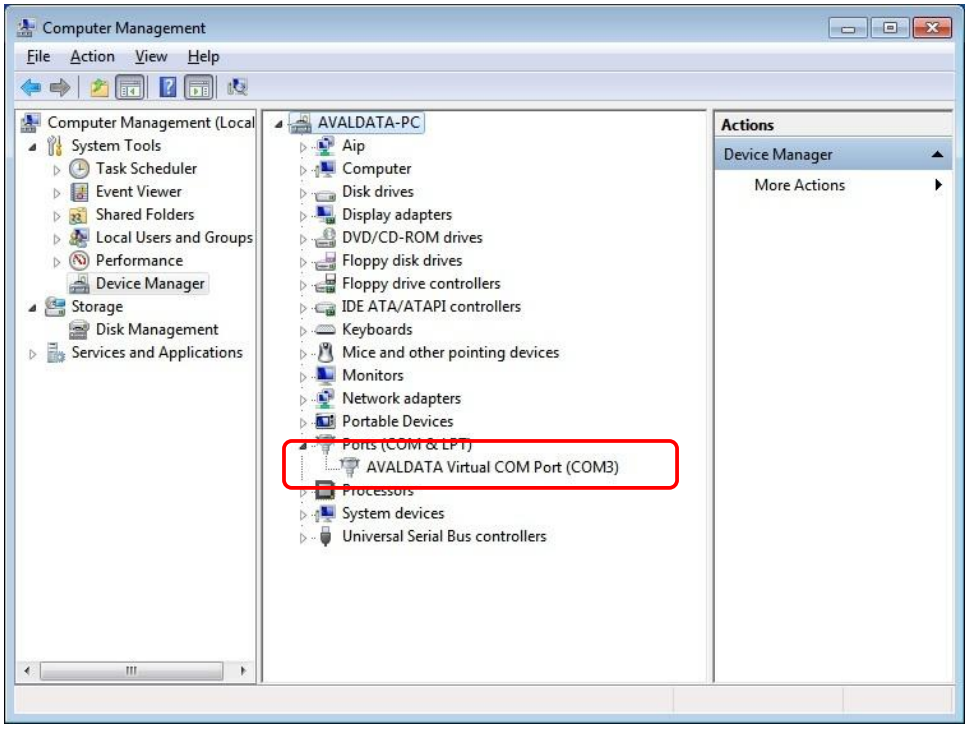


Figure3-21: Device manager

Downloaded from www.Manualslib.com manuals search engine

■ **Uninstallation of device driver and SDK-AcapLib2**

[Method 1]

Execute following programs.

<WindwosXP>

[Start] =>[Control panel] => [Addition or deletion of program]

<WindwosVista>

[Start] => [Setting] =>[Control panel] => [Uninstall a program]

<Windwos7>

[Start] => [Control panel] => [Uninstall a program]

[Method 2]

Select [AVALDATA SDK-AcapLib2] and delete it.

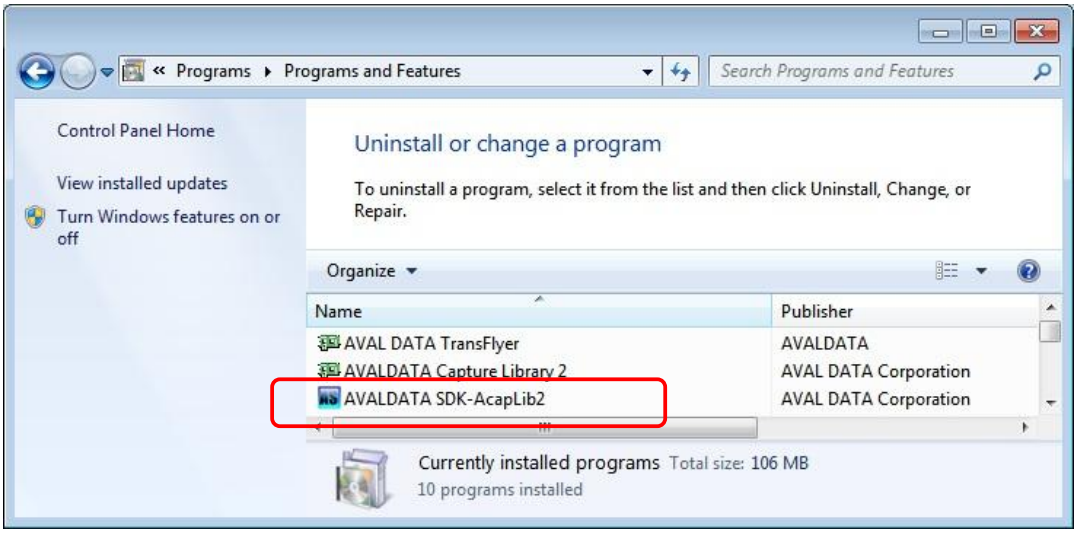


Figure3-22: Uninstallation of a program

[Method 3]

Close a program above once, execute a program again and uninstall if the following items remain.

- [AVALDATA Capture Library 2]
- [Windows driver package - AVAL DATA CORPORATION Aip (apx***)] (****)
- [AVAL DATA TransFlyer]

3.2.7 Installation trouble shooting

Check following points, if any problem occurs during installation.

- Check whether the hardware is connected correctly again.
- Check whether the same problem occurs when you mount APX-3302 into other PCI Express slot
- Refer to [Precautions befor installation]

Downloaded from www.Manualslib.com manuals search engine

Chapter 4 AcapLib2Viewer Image acquisition

This chapter describes the procedure until acquire images using APX-3302 and AcapLib2Viewer.

4.1 What is AcapLib2Viewer

AcapLib2Viewer is the software for acquiring images using the many models of frame-grabbers including APX-3302. Main functions of AcapLib2Viewer are shown below.

■ Frame-grabber initialization
Setting of Capture size, Shutter trigger, External trigger, Encoder, etc

■ Acquire images using the camera
Snap (One shot snap), AutoSnap (continuous snap-shot), and Grab (streaming) can be performed.

■ Display acquired images
Display enlarged, reduced, the same magnification images. Moreover, display the luminance value of specified pixels.

■ File output of acquired image
Output acquired images into various image formats such as (bmp, csv, bin, jpg, gif, tif, png).

■ Initialization file (ini file) output
Output each settings of AcapLib2Viewer as the ini file.

4.2 What is ini file

An ini file is a file which stored various setting points, such as capture size, shutter trigger, external trigger, and encoders. It can perform initialization of AcapLib2Viewer and AcapLib2.
Moreover, ini file which can perform basic capturing is including in the SDK-AcapLib2 for the cameras which has tested with frame-grabber.

Latest ini file is available to downloadable from [supported camera information] in the Web site.

[supported camera information] * Japanese version only available
http://www.avaldata.co.jp/products/z2_imaging/supportcamera/camera.html

Downloaded from www.Manualslib.com manuals search engine

4.3 Image acquisition using AcapLib2Viewer

Procedures until acquire images using AcapLib2Viewer.
Refer to [SDK-AcapLib2 Viewer Manual (J2OM08014*)] for the detailed setting method.

[Method 1]
Select [All programs] from the start menu of Windows.

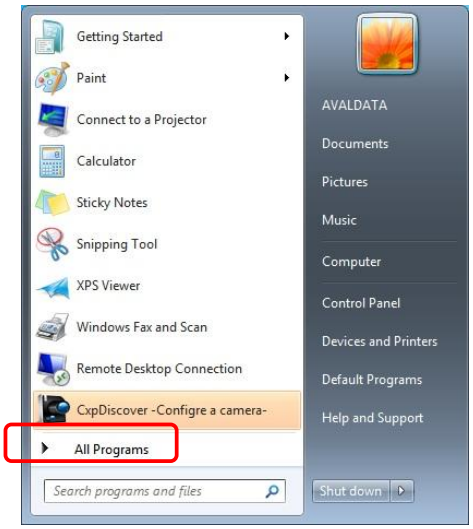


Figure4-1: Start menu

[Method 2]
Select AcapLib2Viewer in [AVALDATA]-> [SDK-AcapLib2]

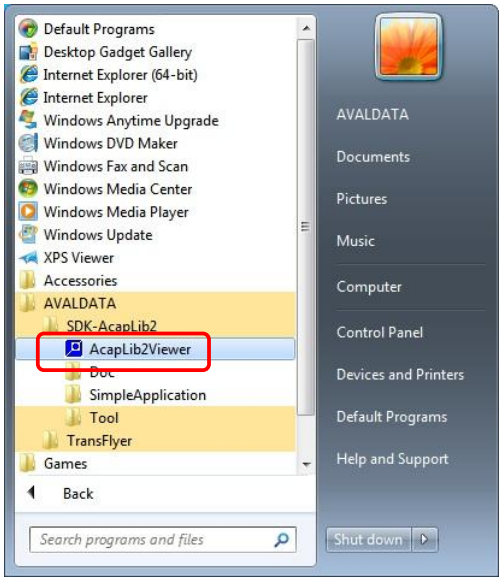


Figure4-2: All the programs

Downloaded from www.Manualslib.com manuals search engine

[Method 3]
Select each item when a dialog appears which can be selected ini file, Board (frame-grabber) name, Board (frame-grabber) ID number, input channel.

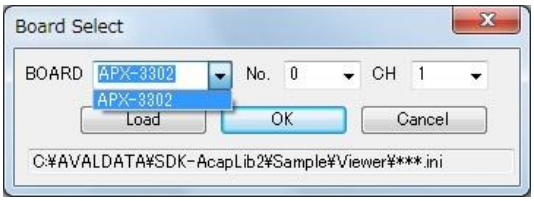


Figure4-3: Board selection

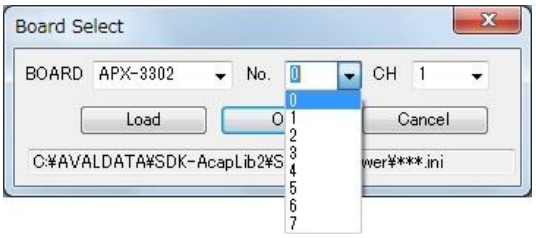


Figure 4-4: Board ID number selection

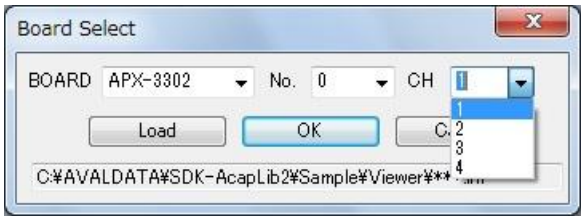


Figure4-5: Input-channel selection

■ Board ID number
Board ID number is a number which can be identified a solid when two or more APX-3302 are mounted into one PC. It is set as 0 at in the default. It can be changed into any value by using [Firmware Update Tool].

[Method 4]
ini file selection dialog appears when click [Load]

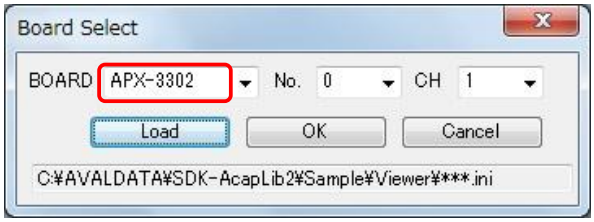


Figure4-6: Check dialog for the board

Downloaded from www.Manualslib.com manuals search engine

[Method 5]
Select ini file of the camera be used from ini file by some camera manufactures in the folder.
[SDK-AcapLib2 install folder] ->[AVALDATA]->[SDK-AcapLib2] ->[Sample]->[IniFile]

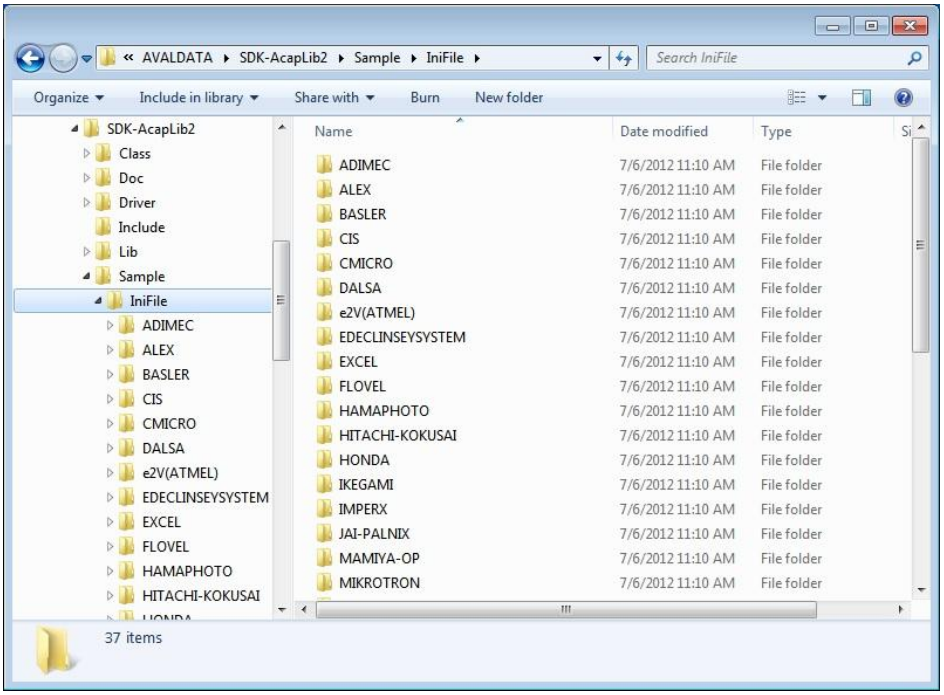


Figure4-7: Ini file selection dialog

[Method 6]
Click [OK] after choosing ini file.
Click [OK] then, initialize with ini file selected.

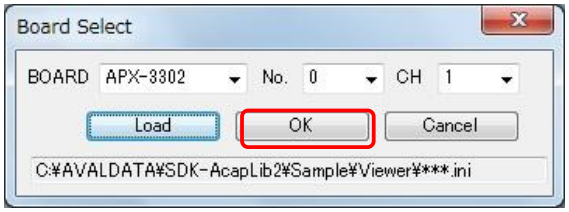


Figure4-8: Check dialog for the board

Downloaded from www.Manualslib.com manuals search engine

[Method 7]
The main window will be displayed when the initialization is successful.
Click a camera icon in the red frame, one shot snap will be performed.

* A camera shall be set as a factory-shipments state, free run, and the power source shall be ON.

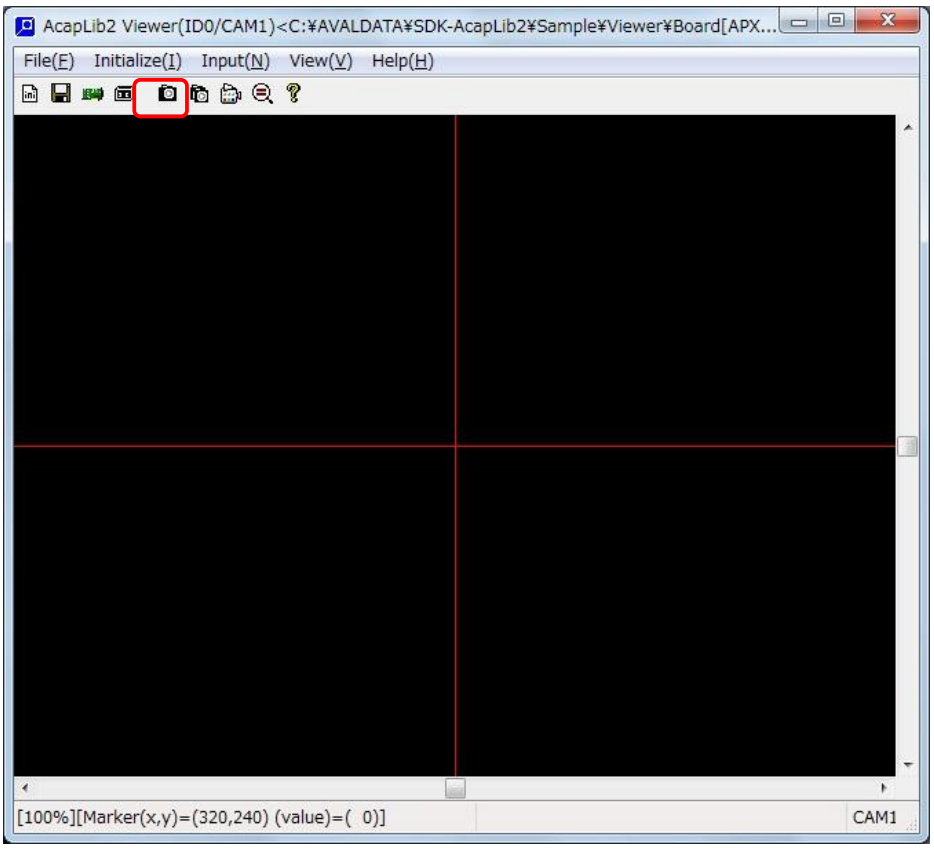


Figure4-9: Main window

Downloaded from www.Manualslib.com manuals search engine

[Method 8]
The acquired image, image acquisition time, frame rate will appear when the image acquisition succeeded.

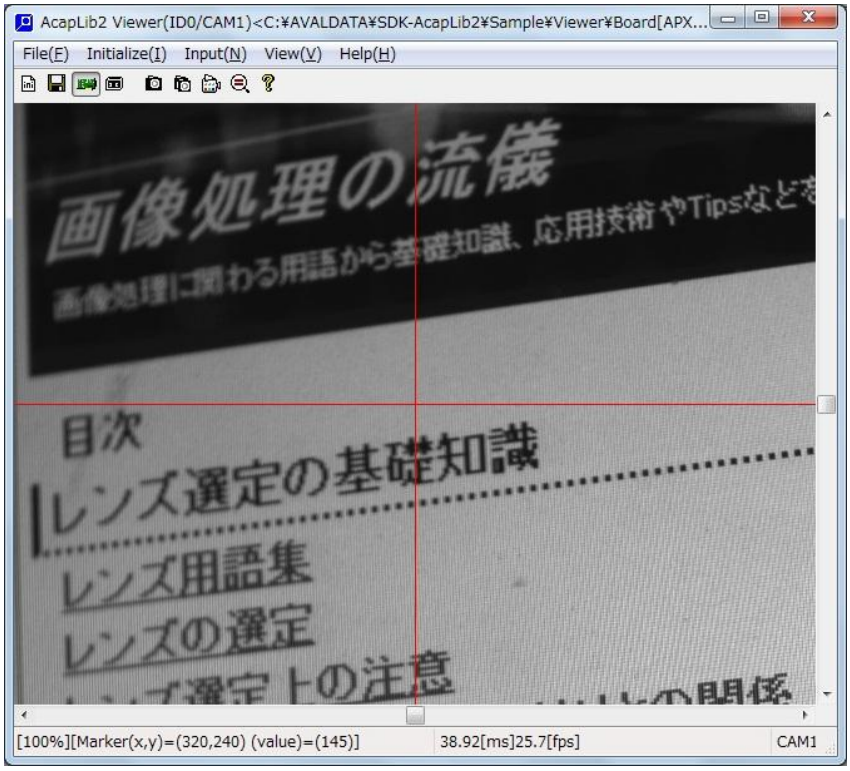


Figure4-10: Acquired image

Error code with error dialog as below will appear when the image acquisition failed.
Mis-setting of the frame-grabber or the camera, mis-connecting of the cable, input omission of external triggers or encoders, etc can cause of failure.
Refer to [Chapter7 Error code] for the details.

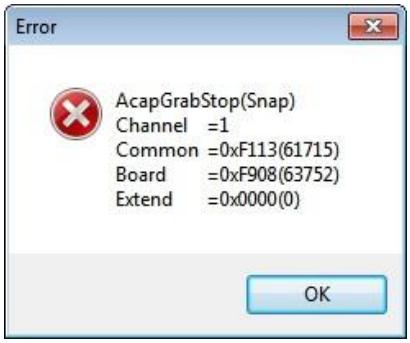


Figure4-11: Error dialog

Downloaded from www.Manualslib.com manuals search engine

Chapter 5 Programming guide

This chapter describes the programming procedure using AcapLib2 is shown below.

5.1 AcapLib2 flow chart

Processing procedure using AcapLib2

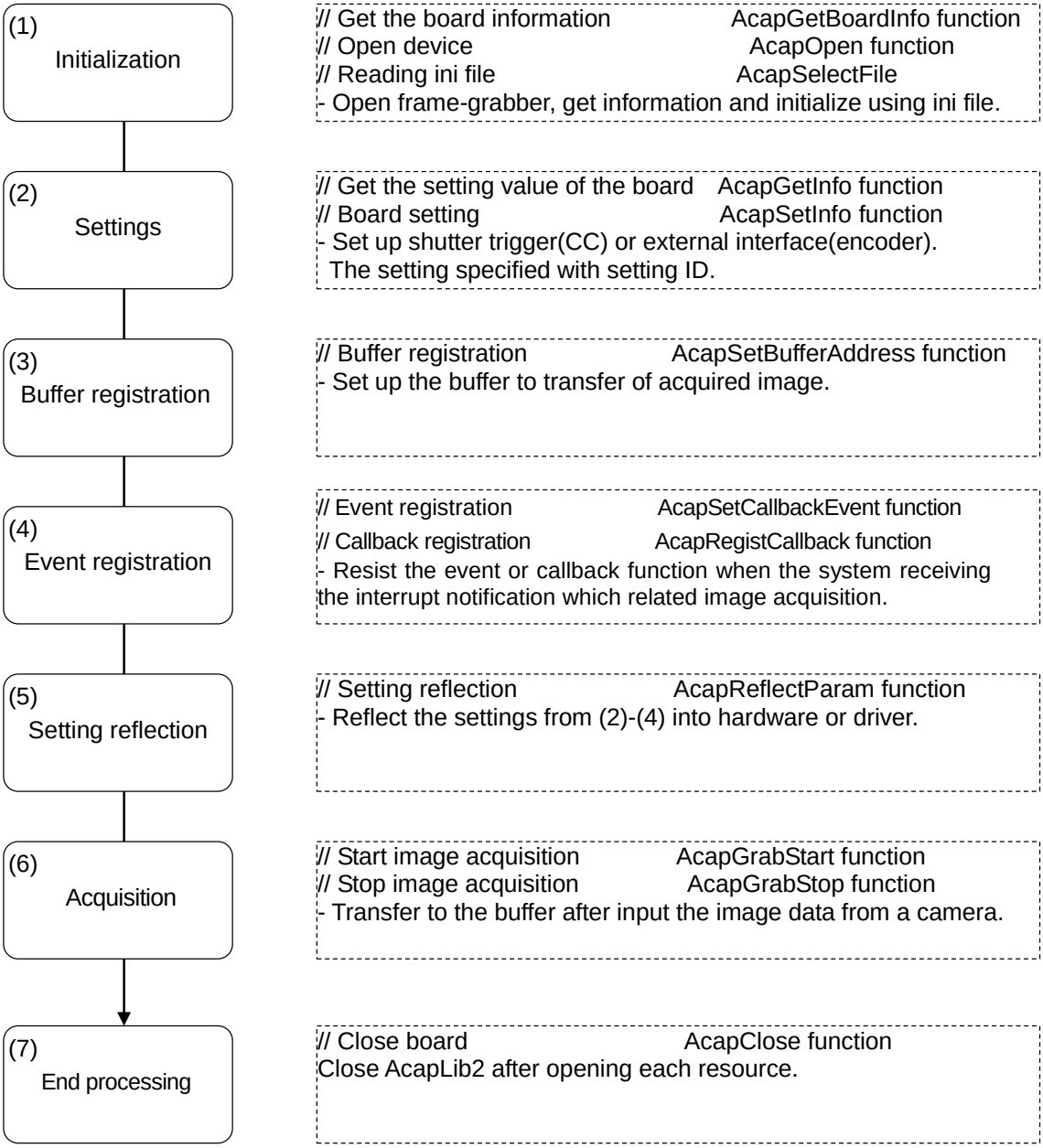


Figure5-1: Flow chart

Downloaded from www.Manualslib.com manuals search engine

5.2 Programming example

The programming examples for Snap (one shot snap) and Grab (streaming) for each are shown as below. GUI, such as a screen display or a dialog is not used. Moreover, error processing is not be performed here either.
The program is premised on being the following conditions.

Board : AVAL DATAs frame-grabber which found first by Windows,
if two or more frame-grabbers are mounted
Ini file : Default.ini (*)
Ini file path: Hierarchy with a project file or an executable file

(*) The main settings of Default.ini is as follows.
Input size : 640x480
XY delay : (x, y) = (0, 0)
Processing bit : 8 bits
External trigger, random shutter: Invalid

5.2.1 Snap (one shot snap)

The programming example of Snap (one shot snap) is shown as below.

<Sample code>

```
#include <stdio.h>
#include <conio.h>
#include "AcapLib2.h"
#pragma comment(lib, "AcapLib2.lib")

#define INPUT_CH 1
#define CAM_LINE 1

int main(int argc, char *argv[])
{
        HANDLE hDev;
        int nXSize, nYSize, nBoardBit;
        int nLoop = 1, nData = 0;
        int nCycle = 0, nScan = 0;
        int nCamera = 0;
        PUCHAR pUserBuffer = NULL;
        ACAPBOARDINFO abi;

        // Get board information
        AcapGetBoardInfo (&abi);)

        // Open device
        hDev = AcapOpen(abi.BOARDINDEX[0].pBoardName,
                        abi.BOARDINDEX[0].nBoardID, INPUT_CH);
        // Initialize by ini file.
        AcapSelectFile(hDev, INPUT_CH, "Default.ini", 0);
```


Downloaded from www.Manualslib.com manuals search engine

```
	// Check camera type
	AcapGetInfo (hDev, INPUT_CH, ACL_SCAN_SYSTEM, 0, &nCamera);)
	if(nCamera == CAM_LINE){
		// If the camera is Line sensor, confirm that exposure setting is performed.
		AcapGetInfo (hDev, INPUT_CH, ACL_EXP_CYCLE, 0, &nCycle);)
		AcapGetInfo (hDev, INPUT_CH, ACL_EXPOSURE, 0, &nScan);)
		if((nCycle == 0) ||    (nScan == 0)){
			printf ("exposure setting out is injustice ¥n");
			return 0;
		}
	}

	// Setting for performing DMA to the buffer.
	AcapGetInfo (hDev, INPUT_CH, ACL_X_SIZE, 0, &nXSize);)
	AcapGetInfo (hDev, INPUT_CH, ACL_Y_SIZE, 0, &nYSize);)
	AcapGetInfo (hDev, INPUT_CH, ACL_BOARD_BIT, 0, &nBoardBit);)
	AcapSetInfo(hDev, INPUT_CH, ACL_MEM_NUM, 0, 1);
	pUserBuffer = (PUCHAR)_aligned_malloc(nXSize * nYSize * (nBoardBit / 8), 16);
	ZeroMemory(pUserBuffer, nXSize * nYSize * (nBoardBit / 8));
	AcapSetBufferAddress(hDev, INPUT_CH, ACL_IMAGE_PTR, 1, pUserBuffer);
	printf ("in order to suspend an input, please input some keys ¥n");

	do{
		// Snap
		AcapGrabStart(hDev, INPUT_CH, 1);
		// Stop capturing
		AcapGrabStop(hDev, INPUT_CH);
		Sleep(1);

		// Dump the data of (x, y) = (100, 100)
		nData = pUserBuffer[100 + 100 * nXSize];
		printf ("taking-in number of sheets :    %4d data (100, 100)    : %3d¥n", nLoop++, nData);
	}while(!kbhit());
	printf ("¥n which suspended the input");

	// Close device
	AcapClose(hDev, INPUT_CH);

	// Release the buffer
	if(pUserBuffer != NULL){
		_aligned_free(pUserBuffer);
		pUserBuffer = NULL;
	}
	return 0;
}
```


Downloaded from www.Manualslib.com manuals search engine

5.2.2 Grab (Streaming)

The programming example of Grab (Streaming) is shown as below.

<Sample code>

```
#include <stdio.h>
#include <conio.h>
#include "AcapLib2.h"
#pragma comment(lib, "AcapLib2.lib")

#define INPUT_CH 1
#define CAM_LINE 1

void CALLBACK GrabCallbackFunc(int nChannel, DWORD dwEvent, int nParam1, int nParam2);
PUCHAR pUserBuffer[4];
int nXSize, nYSize, nBoardBit;

int main(int argc, char *argv[])
{
        HANDLE hDev;
        int nCount;
        int nCamera;
        int nCycle, nScan;
        DWORD dwEventMask;
        ACAPBOARDINFO abi;

        // Get board (frame-grabber) information
        AcapGetBoardInfo (&abi);)
        // Open device
        hDev = AcapOpen(abi.BOARDINDEX[0].pBoardName, abi.BOARDINDEX[0].nBoardID, INPUT_CH);
        // Initialize by ini file.
        AcapSelectFile(hDev, INPUT_CH, "Default.ini", 0);

        // Check camera type
        AcapGetInfo (hDev, INPUT_CH, ACL_SCAN_SYSTEM, 0, &nCamera);)
        if(nCamera == CAM_LINE){
                // If the camera is Line sensor, confirm that exposure setting is performed.
                AcapGetInfo (hDev, INPUT_CH, ACL_EXP_CYCLE, 0, &nCycle);)
                AcapGetInfo (hDev, INPUT_CH, ACL_EXPOSURE, 0, &nScan);)
                if((nCycle == 0) ||    (nScan == 0)){
                        printf ("exposure setting out is injustice ¥n");
                        return 0;
                }
        }
        // Set the number of memory as 4 frames for each ch
        AcapSetInfo(hDev, INPUT_CH, ACL_MEM_NUM, 0, 4);
```


Downloaded from www.Manualslib.com manuals search engine

```
		// ch1: Setting for performing DMA to the buffer.
		AcapGetInfo (hDev, INPUT_CH, ACL_X_SIZE, 0, &nXSize);)
		AcapGetInfo (hDev, INPUT_CH, ACL_Y_SIZE, 0, &nYSize);)
		AcapGetInfo (hDev, INPUT_CH, ACL_BOARD_BIT, 0, &nBoardBit);)
		for(nCount= 0; nCount < 4; nCount++){
			pUserBuffer[nCount] =
				(PUCHAR)_aligned_malloc(nXSize * nYSize * (nBoardBit / 8), 16);
			ZeroMemory(pUserBuffer[nCount], nXSize * nYSize * (nBoardBit / 8));
			 // Set nMemnum as negative value, for suspending reflection of setting at end.
			AcapSetBufferAddress(hDev, INPUT_CH, ACL_IMAGE_PTR,
					(nCount + 1)*(-1), &pUserBuffer[nCount][0]);
		}
		// Reflect configuration parameter
		AcapReflectParam(hDev, INPUT_CH);
		// Register callback function
		// ch1: Register ACL_INT_FRAMEEND (Frame decision interruption)
		dwEventMask = ACL_INT_FRAMEEND;
		AcapRegistCallback(hDev, INPUT_CH, dwEventMask, GrabCallbackFunc);

		// Grab start
		printf ("in order to suspend an input, please input some keys ¥n");
		AcapGrabStart(hDev, INPUT_CH, 0);

		do{
			Sleep(1);
		}while(!kbhit());

		// Grab stop
		AcapGrabStop(hDev, INPUT_CH);
		printf ("¥n which suspended the input");

		// Close device
		AcapClose(hDev, INPUT_CH);

		// Release the buffer
		for(nCount = 0; nCount < 4; nCount++){
			if(pUserBuffer[nCount] != NULL){
				_aligned_free(pUserBuffer[nCount]);
				pUserBuffer[nCount] = NULL;
			}
		}

		return 0;
}

//callback function
void CALLBACK GrabCallbackFunc(int nChannel, DWORD dwEvent, int nParam1, int nParam2)
{
		int nIndex;
		int nData;

		switch(dwEvent){
			case ACL_INT_FRAMEEND:
				//Processing of Frame decision interruption
				// Dump the data of (x, y) = (100, 100)
				nData = pUserBuffer[nParam2-1][100 + 100 * nXSize];
				printf ("taking-in number of sheets :   %4d data (100, 100)   : %3d¥n",
					nParam1, nData);
				break;
		}
}
```


Downloaded from www.Manualslib.com manuals search engine

5.3 Sample capture application

The capture sample applications for each programming languages, the samples for connecting various types of image-processing library are prepared within AcapLib2.

5.3.1 SimpleApplication

The capture sample applications for each programming languages using AcapLib2
It is stored in the folder as below.

[Capturing sample using AcapLib2]
-[SDK-AcapLib2 install destination]=>[AVALDATA]=>[SDK-AcapLib2]=>[Sample]=>[SimpleApplication]

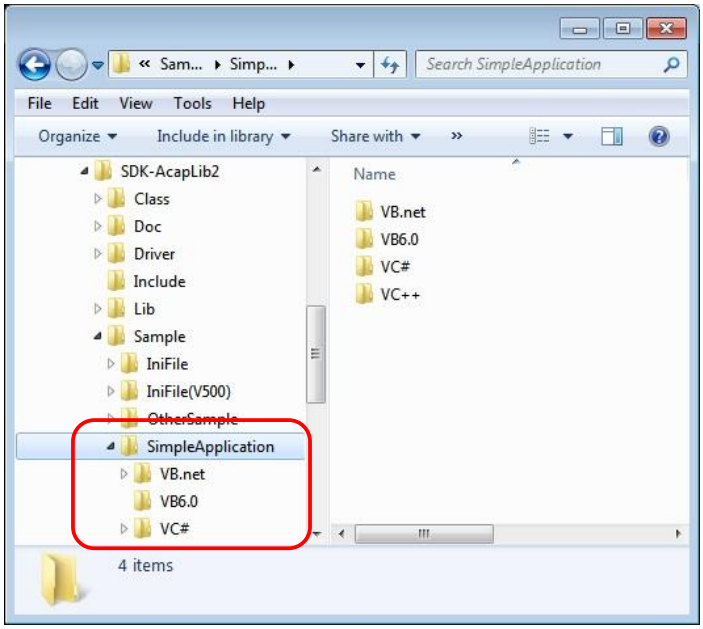


Figure5-2:SimpleApplication

[SimpleApplication] Folder	Sample contents
VB.net	VB.net capture sample using AcapLib2.
VB6.0	VB6.0 capture sample using AcapLib2.
VC#	VC# capture sample using AcapLib2.
VC++	VC++ capture sample using AcapLib2. Including free run, random shutter, external trigger.

* 64bitOS is not supported except VC++ sample.

Table5-1:SimpleApplication

Downloaded from www.Manualslib.com manuals search engine

5.3.2 OtherSample

The samples for connecting AcapLib2 with various types of image-processing library
It is stored in the folder as below.

[Connecting samples using various types of image-processing library]
- [SDK-AcapLib2 install destination] =>[AVALDATA] =>[SDK-AcapLib2] =>[Sample]=>[OtherSample]

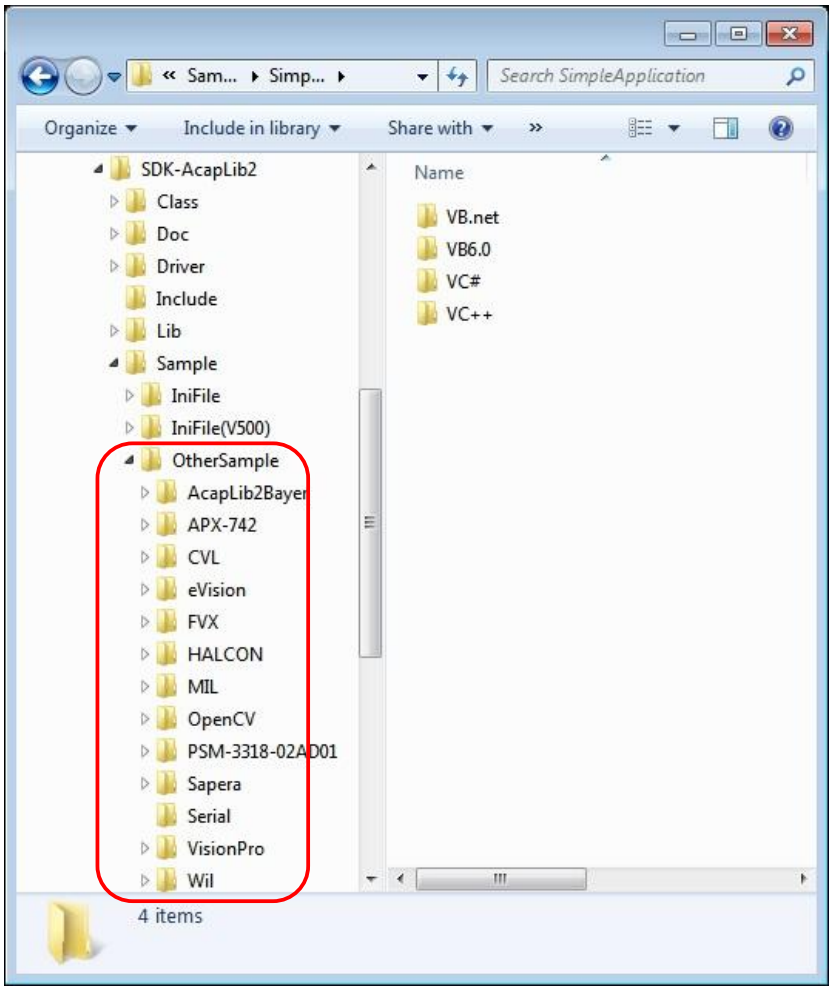


Figure5-3:OtherSample

Downloaded from www.Manualslib.com manuals search engine

Following samples are contained in folder
Refer to a manual of each sample for capturing samples when using other manufacture's library.
[SDK-AcapLib2 install destination] =>[AVALDATA] =>[SDK-AcapLib2] =>[Doc] =>[6_Technical]

[OtherSample] Folder	Sample contents
AcapLib2Bayer	VC++ sample which captures the image by AcapLib2 then performs bayer conversion.
APX-742	VC++ sample for image distribution with "APX-742."
CVL	A sample which captures the image by AcapLib2 then performs image processing in CVL.
eVision	VC++ sample which captures the image by AcapLib2 then performs image distribution in eVision.
FVX	VC++ sample which captures the image by AcapLib2 then performs image processing in FVX.
HALCON	VC++ sample which captures the image by AcapLib2 then performs image distribution in HALCON.
MIL	VC++ sample which captures the image by AcapLib2 then performs image distribution in MIL.
OpenCV	VC++, CLI sample which captures the image by AcapLib2 then performs image processing in OpenCV.
PSM-3318-02AD01	VC++ sample which performs image distribution in with "PSM-3318-02."
Sapera	VC++ sample which captures the image by AcapLib2 then performs image processing in Sapera.
Serial	VC++ which performs serial communication across the camera.
VisionPro	VB.net sample which captures the image by AcapLib2 then performs image processing in VisionPro.
Wil	VC++ sample which captures the image by AcapLib2 then performs image processing in WIL.

* 64bitOS is not supported except samples AcapLib2Bayer and PSM-3318-02AD01.

Table5-2:OtherSample

Downloaded from www.Manualslib.com manuals search engine

Chapter 6 Function

This chapter explains the function of APX-3302 and the examples how to use AcapLib2.

6.1 CameraLink Configuration

APX-3302 can acquire image from the camera corresponding to CameraLink Base Configuration.
APX-3302 can connect up to two cameras and it carries 26PinSDR connector.
The CameraLink input circuit is independent for each camera, so APX-3302 can connect with two cameras even if they are different models like area sensor camera and line sensor camera.

The graphics format that can be acquired by APX-3302 is as follows.

- 8bit x1tap Monochrome
- 8bit x2tap Monochrome
- 8bit x3tap Monochrome
- 10-16bit x1tap Monochrome
- 10-12bit x2tap Monochrome
- 24bitRGB x1tap Color

APX-3302 has pix rearrangement function for the cameras that have two or more tap outputs camera image.
APX-3302 corresponds to 20 MHz - 85 MHz clock. (Receiver is DS90CR288 manufactured by N.S Company.)
The CameraLink synchronizing-signal timing that can be acquired by APX-3302 is as follows.

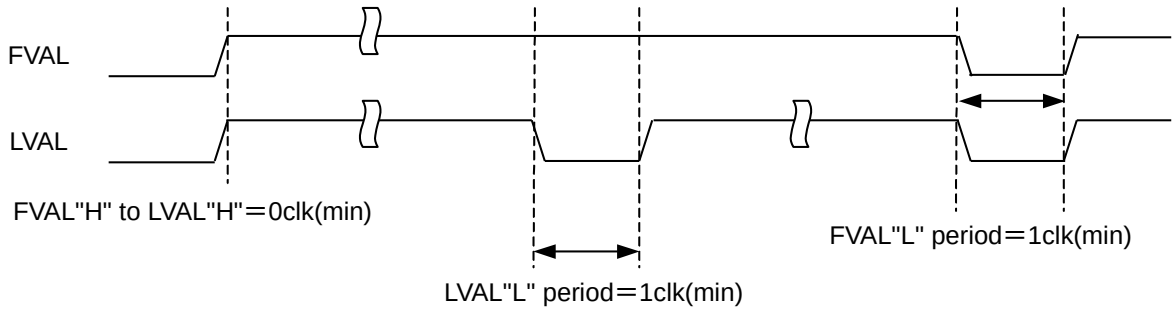


Figure6-1: CameraLink synchronizing signal

Downloaded from www.Manualslib.com manuals search engine

6.1.1 How to configure for 2TAP output cameras

■ The setting example for software

The sample code for 2TAP output cameras is as follows.

In this example, the data from camera is outputted in following order.

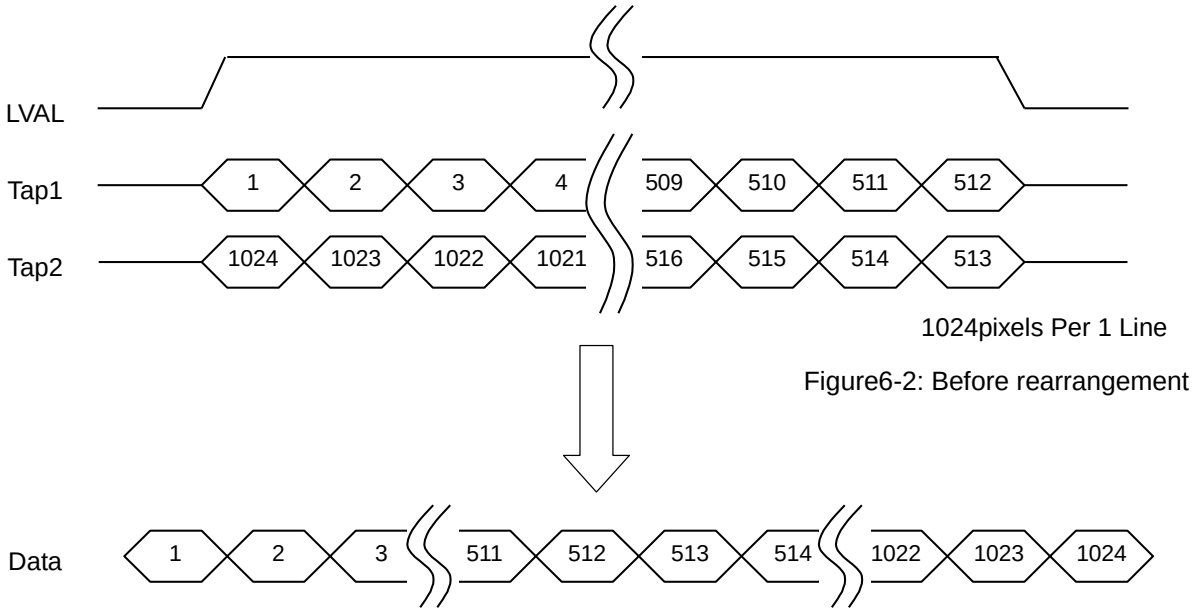


Figure6-2: Before rearrangement

Figure 6-3: After rearrangement

<Sample code>

```
// Configuration of tap number
AcapSetInfo(hDev, INPUT_CH, ACL_TAP_NUM, -1, 2)
// Configure rearranging method as all taps independent
AcapSetInfo(hDev, INPUT_CH, ACL_TAP_ARRANGE, -1, 1)
// Configure the rearrangement direction as "-><-."
AcapSetInfo(hDev, INPUT_CH, ACL_TAP_DIRECTION, -1, 0)
AcapSetInfo(hDev, INPUT_CH, ACL_TAP_DIRECTION, -2, 1)
// Disable Line reverse function
AcapSetInfo(hDev, INPUT_CH, ACL_LINE_REVERSE, -1, 0)
// Configure the X direction delay of LVAL as 0
AcapSetInfo(hDev, INPUT_CH, ACL_LVAL_DELAY, -1, 0)
// Configure the whole pixel number from camera for 1 line
AcapSetInfo(hDev, INPUT_CH, ACL_ARRANGE_XSIZE, -1, 1024)
// Reflect setting
AcapReflectParam(hDev, INPUT_CH)
```


Downloaded from www.Manualslib.com manuals search engine

6.1.2 How to configure for 3TAP output cameras

■ The setting example for software

The sample code for 3TAP output cameras is as follows.

In this example, the data from camera is outputted in following order.

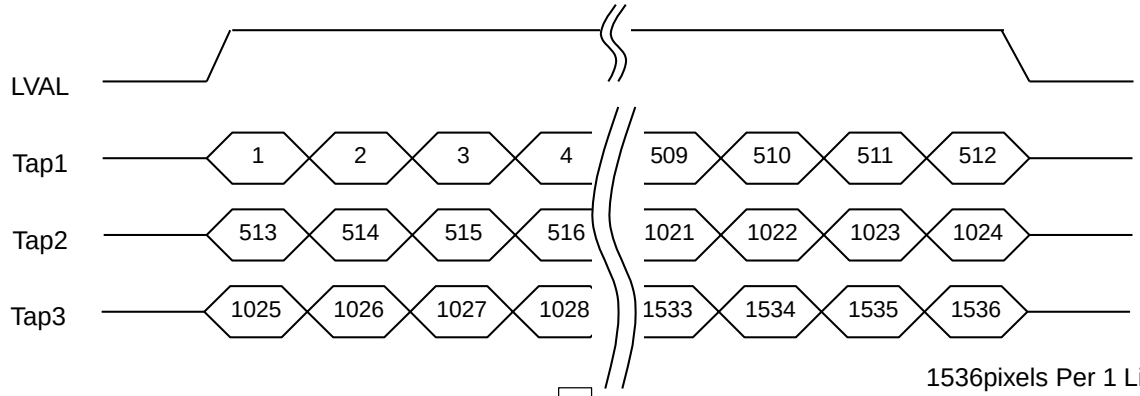


Figure6-4: Before rearrangement

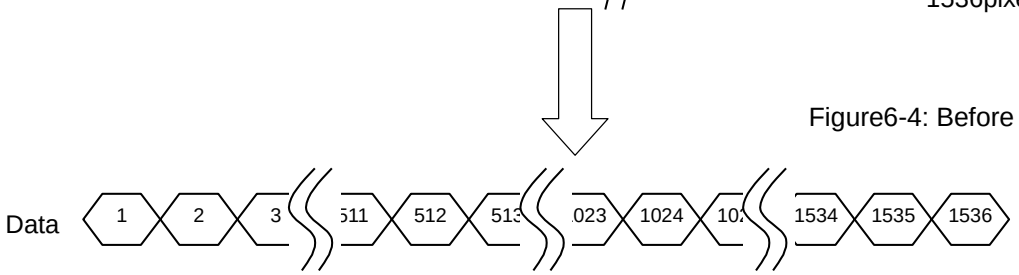


Figure 6-5: After rearrangement

<Sample code>

```
// Configuration of tap number
AcapSetInfo(hDev, INPUT_CH, ACL_TAP_NUM, -1, 3)
// Configure rearranging method as all taps independent
AcapSetInfo(hDev, INPUT_CH, ACL_TAP_ARRANGE, -1, 1)
// Configure the rearrangement direction as "->->."
AcapSetInfo(hDev, INPUT_CH, ACL_TAP_DIRECTION, -1, 0)
AcapSetInfo(hDev, INPUT_CH, ACL_TAP_DIRECTION, -2, 0)
AcapSetInfo(hDev, INPUT_CH, ACL_TAP_DIRECTION, -3, 0)
// Disable Line reverse function
AcapSetInfo(hDev, INPUT_CH, ACL_LINE_REVERSE, -1, 0)
// Configure the X direction delay of LVAL as 0
AcapSetInfo(hDev, INPUT_CH, ACL_LVAL_DELAY, -1, 0)
// Configure the whole pixel number from camera for 1 line
AcapSetInfo(hDev, INPUT_CH, ACL_ARRANGE_XSIZE, -1, 1536)
// Reflect setting
AcapReflectParam(hDev, INPUT_CH)
```


Downloaded from www.Manualslib.com manuals search engine

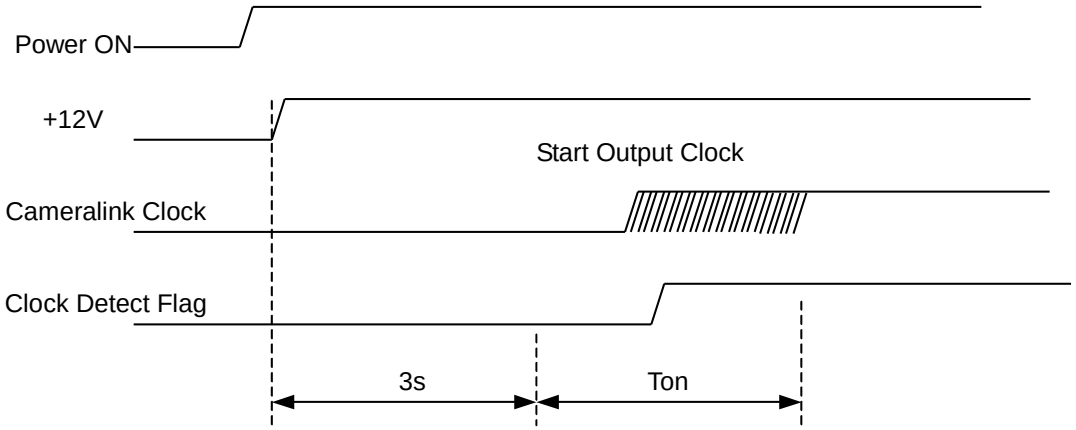
6.2 Power Over CameraLink

APX-3302 can supply +12V power source specified to the camera by PoCL standard.
You can control the power supply ON/OFF for each camera independently by software.

In the following case APX-3302 notifies invalid status and stop supplying power automatically.
*APX-3302 cannot detect CameraLink clock from the camera after supplying power within the set time.
*Camera stops CameraLink clock for any causes, and cannot begin to start CameraLink clock with in the set time.

APX-3302 begins to supply power, if it doesn't detect CameraLink clock when the driver starts.

The sequence of the power supply ON is as follows.



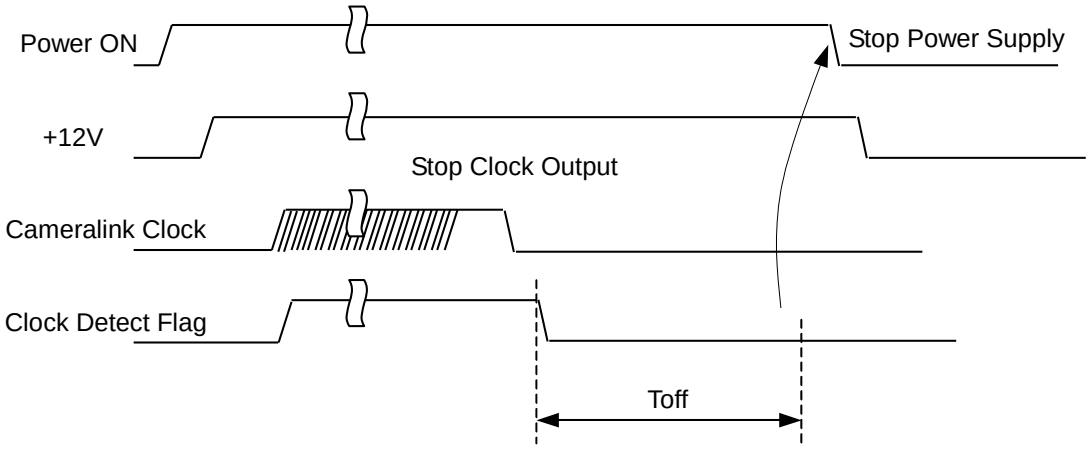
Wait 3 sec after power supply start

If it cannot detect clock detect flag within the set time of Ton, "Power ON Error" will occur and stop power supply.

Figure6-6: The sequence of the power supply ON

Downloaded from www.Manualslib.com manuals search engine

The invalid sequence under turning on electricity is as follows.



If it cannot detect clock detect flag within the set time of Toff, "Power OFF Error" will occur and stop power supply.

Figure6-7: The invalid sequence under turning on electricity

■ The setting example for software

The sample code which performs to supply 1ch side camera and to configure the set time as 3000ms is as follows.

Note: Supplying control is possible only when setting of the jumper (CN7/CN8) of APX-3302 is ON.

For more information of the jumper setting, refer to [3.1.4 CN7, CN8 setting].

<Sample code>

```
// Clear error status of camera power supplying
AcapSetInfo(hDev, INPUT_CH, ACL_POWER_STATE, 0, 1)
// Begin power supplying to camera
AcapSetInfo(hDev, INPUT_CH, ACL_POWER_SUPPLY, 3000, 1)
```


Downloaded from www.Manualslib.com manuals search engine

6.3 Camera exposure control signal CC

APX-3302 can output camera exposure control signal CC (Camera Control).
One of the external generation systems which operates as a factor can choose external signal inputs, such as exposure time, an internal generation system which sets up and operates an exposure cycle by software, or external triggering, an encoder.

As for an internal generation system and an external generation system, the setting-out unit of exposure time / exposure cycle is 1uS. The timing of an internal generation system is shown below.

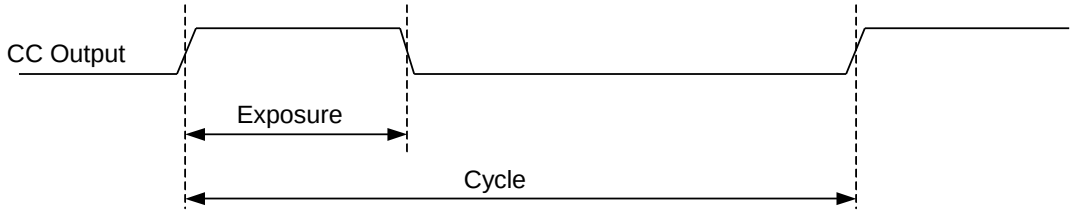


Figure6-8: Internal generation CC

The timing of an external generation system is shown below.

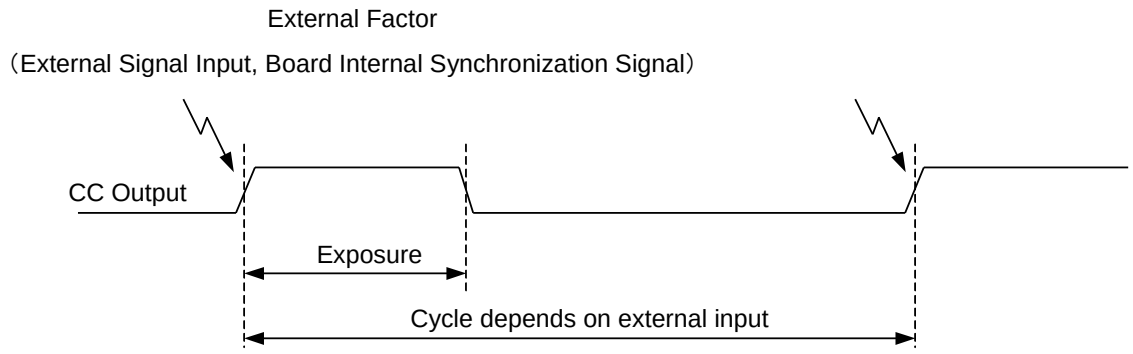


Figure6-9: External generation CC

The delay time from an external signal input to CC output start should follow the following tables as a rule of thumb.

External input IF type	Delay time to CC output start
TTL input	1uS(Max)
RS422 input	1uS(Max)

Table6-1: Delay time of external signal input

This table is a value having no noise rejection time setting (with no digital filter setting out). If you enable the noise rejection function, the delay for settling time is added.

Downloaded from www.Manualslib.com manuals search engine

■ The setting example for software

The sample code in the case of setting CC output cycle as 100 micro sec, and below, setting CC output width as 30 micro sec is described.

* The output width of a CC signal should set up a value smaller than an output cycle.

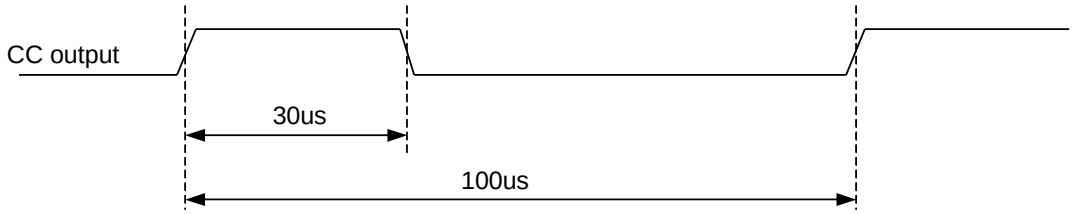


Figure6-10: CC signal output timing

<Sample code>

```
// Enable output CC signal
AcapSetInfo(hDev, INPUT_CH, ACL_EXP_EN, -1, 1);
// CC signal is set as positive logic
AcapSetInfo(hDev, INPUT_CH, ACL_EXP_POL, -1, 1);
// CC output width is set as 30us
AcapSetInfo(hDev, INPUT_CH, ACL_EXPOSURE, -1, 30);
// CC output cycle is set as 100us
AcapSetInfo(hDev, INPUT_CH, ACL_EXP_CYCLE, -1, 100);
// Reflect setting
AcapReflectParam(hDev, INPUT_CH)
```


Downloaded from www.Manualslib.com manuals search engine

6.4 Serial interface

APX-3302 carries the serial interface for camera communication for 2ch.

- SerTC : The signal from APX-3302 to camera direction (transmit data)
- SerTFG: The signal from camera to APX-3302 direction (receive data)

Serial interface is connected to UART independent every channel of APX-3302.
It is possible to assign UART to a COM port by using the virtual COM driver made by AVALDATA, and to communicate with a camera by two kinds of methods below.

- Communication using the communicating software which a camera manufacturer offers
- Communication using API (AcapLib2)

It can communicate by the following specifications between APX-3302 and a camera.
Communication speed: 9600bps(default) - Change by register setting is possible.

Start bit	: 1 bit
Data bit	: 8 bit
Stop bit	: 1 bit
Parity bit	: Nothing
Flow control	: Nothing

Please refer to "SDK-AcapLib2 virtual COM tool user's manual" for virtual COM.

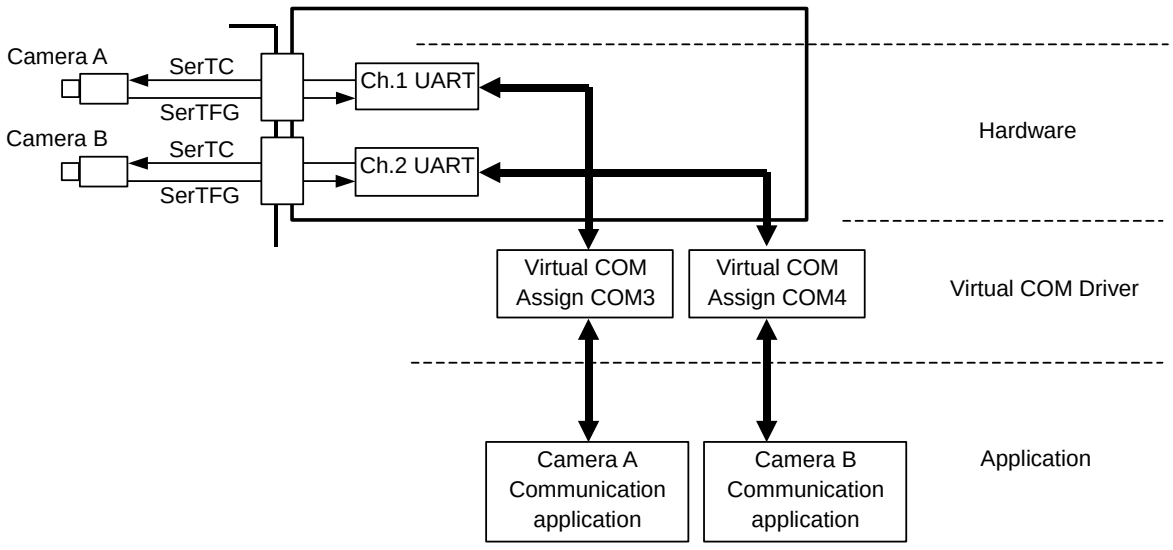


Figure6-11: Serial interface

Downloaded from www.Manualslib.com manuals search engine

■ The setting example for software

The sample code for sending command "AVAL" is as follows.
In this code, an alphabetic code, a baud rate, and a final character carry out the following.

Communication speed: 9600 bps
Final character : CR (¥r)
character code : ASCII

<Sample code>

```
// Open serial port
AcapSerialOpen(hDev, INPUT_CH);
// Configure serial parameter
AcapSerialSetParameter(hDev, INPUT_CH, 9600, 8, 0, 0, 0);
// Send command (writing)
AcapSerialWrite(hDev, INPUT_CH, TRUE, "AVAL¥r", NULL, NULL);
// Close serial port
AcapSerialClose(hDev, INPUT_CH);
```


Downloaded from www.Manualslib.com manuals search engine

6.5 External signal input

APX-3302 can interface with peripheral device by using expansion connector (CN6).
The signal names and the interfaces are shown below.

Signal name	Function
ExTrg_1	TTL level: External trigger input terminal for Ch.1. It is below available for three kinds of uses. - For Ch.1 Image input start trigger - For Ch.1 CC generating trigger - For Ch.1 Encoder startup trigger A negative going edge is recognized as a trigger.
ExTrg_2	TTL level: External trigger input terminal for Ch.2. It is below available for three kinds of uses. - For Ch.2 Image input start trigger - For Ch.2 CC generating trigger - For Ch.2 Encoder startup trigger A negative going edge is recognized as a trigger.
Enc.A±	RS422 level: Encoder input terminal. It is available as encoder A phase for Ch.1 and Ch.2.
Enc.B±	RS422 level: Encoder input terminal. It is below available for three kinds of uses. - Encoder B phase for Ch.1 and Ch.2 - Image input start trigger for Ch.1 - CC generating trigger for Ch.1
Enc.Z±	RS422 level: Encoder input terminal. It is below available for three kinds of uses. - Encoder Z phase for Ch.1 and Ch.2 - Image input start trigger for Ch.2 - CC generating trigger for Ch.2
Diff±	RS422 level: Common external trigger input terminal. It is below available for three kinds of uses. - Image input start trigger for Ch.1 and Ch.2 - CC generating trigger for Ch.1 and Ch.2 - Encoder startup trigger for Ch.1 and Ch.2 A negative going edge is recognized as a trigger.
GP_IN1	TTL level: General-purpose input terminal. It is available as an interruption factor. An interrupt occurs in a negative going edge.
IN_RES1	TTL level: General-purpose input terminal. The H/L state of a terminal can be checked with a register.
GP_OUT1	TTL / open-collector level: General-purpose output terminal for Ch.1. It is below available for three kinds of uses. - Outputs in general - Stroboscope output for Ch.1 - Capture flag (Output "L" level while capturing) for Ch.1
GP_OUT2	TTL / open-collector level: General-purpose output terminal. It is below available for three kinds of uses. - Outputs in general - Stroboscope output for Ch.2 - Capture flag (Output "L" level while capturing) for Ch.2
SYNCLT_IN	TTL level, synchronizing-signal input terminal between frame-grabbers. Connect the SYNCLT_OUT terminal of a master frame-grabber (synchronizing-signal generating origin).

Downloaded from www.Manualslib.com manuals search engine

SYNCLT_OUT	TTL level, synchronizing-signal output terminal between frame-grabbers. Connect with the SYNCLT_IN terminal of a slave frame-grabber (synchronizing-signal report destination).
OUT_RES1	It cannot be used in APX-3302.

Table6-2: I/F of an external signal input

Downloaded from www.Manualslib.com manuals search engine

The input circuit of an external signal is shown below.

1. TTL Level Input Terminal

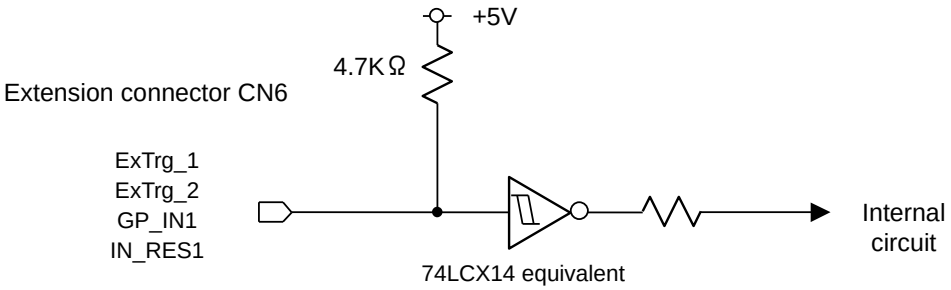


Figure6-12: TTL input terminal

Don't apply the voltage beyond +5V to a TTL terminal.

Since a TTL signal tends to be affected by the influence of transmission length or a peripheral noise,
Under the environment which a noise generates, use of a differential signal (RS422) is recommended strongly.

When you use external trigger
Input the "L" level stabilized more than the period of at least 25 uS(s).

■ The setting example for software

The sample code for using TTL signal is as follows.
In this example, External trigger mode is set as continuation external trigger mode
(CC signal is outputted whenever it detects external trigger.)

<Sample code>

```
// Configure to use TTL signal
AcapSetInfo(hDev, INPUT_CH, ACL_EXT_EN, -1, 1);
// Continuation external trigger mode
AcapSetInfo(hDev, INPUT_CH, ACL_EXT_MODE, -1, 0);
// Reflect setting
AcapReflectParam(hDev, INPUT_CH)
```


Downloaded from www.Manualslib.com manuals search engine

2. RS422 Level Input Terminal

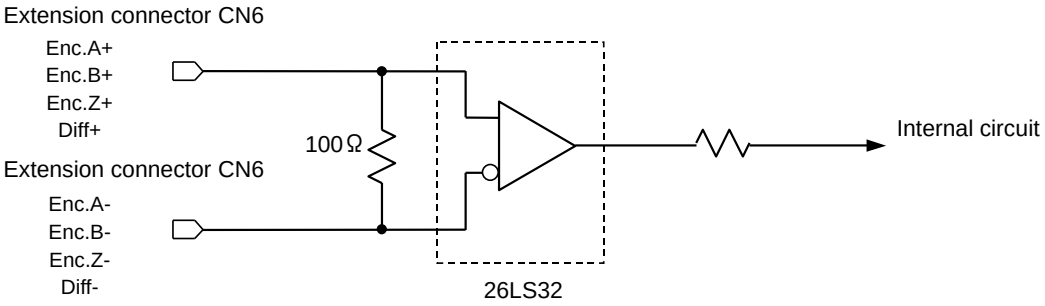


Figure6-13: RS422 input terminal

When you use external trigger
Input the "L" level stabilized more than the period of at least 25 uS(s).

In an intact RS422 input, an input level becomes unfixed.
If external triggering is validated in the unfixed state, a trigger may occur to unexpected timing. When you do not use a RS422 input, be sure to set up external triggering invalid.

■ The setting example for software

The sample code for using RS-422 signal is as follows.
In this example, External trigger mode is set as continuation external trigger mode
(CC signal is outputted whenever it detects external trigger.)

<Sample code>

```
// Configure to use RS-422 signal
AcapSetInfo(hDev, INPUT_CH, ACL_EXT_EN, -1, 2);
// Continuation external trigger mode
AcapSetInfo(hDev, INPUT_CH, ACL_EXT_MODE, -1, 0);
// Reflect setting
AcapReflectParam(hDev, INPUT_CH)
```


Downloaded from www.Manualslib.com manuals search engine

3. TTL / Open Collector Output Terminal

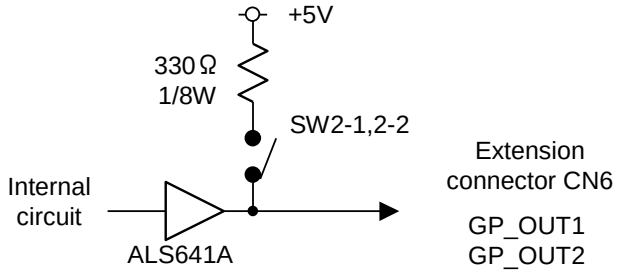


Figure6-14: Open collector output terminal

Don't apply the voltage beyond +5V to TTL / open collector output terminal.

■ The setting example for software

The sample code for using GP_OUT1 for below as a general-purpose output is as follows.

<Sample code>

```
// Configure to use GP_OUT1 as a general-purpose output
AcapSetInfo(hDev, INPUT_CH, ACL_GPOUT_SEL, -1, 1);
// Reflect setting
AcapReflectParam(hDev, INPUT_CH)
// Configure GP_OUT1 Low output.
AcapSetInfo(hDev, INPUT_CH, ACL_GPOUT_POL, 0, 0);
// Configure GP_OUT1 High output.
AcapSetInfo(hDev, INPUT_CH, ACL_GPOUT_POL, 0, 1);
```


Downloaded from www.Manualslib.com manuals search engine

6.6 The noise rejection function of external input signal

APX-3302 carries the noise rejection circuit of external trigger in each channel.
It is possible to cover a digital filter over the selected external trigger, and to remove a single noise and the noise at the time of level variation.

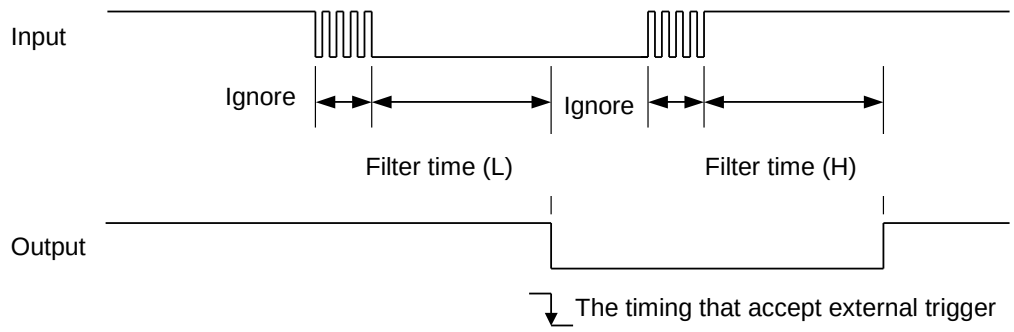


Figure6-15: Noise rejection of external trigger

Noise rejection time can be set up per uS.

On the specification of a digital filter, Keep in mind that the delay for filter settling time occurs in an output.

■ The setting example for software
The sample code that set external trigger noise removing time High and Low as 100us is as follows.

<Sample code>

```
// Configure noise rejection time as 100us.
AcapSetInfo(hDev, INPUT_CH, ACL_EXT_CHATTER, -1, 100);
// Configure external trigger period (the same value for High and Low period)
AcapSetInfo(hDev, INPUT_CH, ACL_EXT_CHATTER_SEPARATE, -1, 0);
// Reflect setting
AcapReflectParam(hDev, INPUT_CH)
```


Downloaded from www.Manualslib.com manuals search engine

6.7 Strobe output

APX-3302 can output strobe signal that need to connect with external illuminating equipment.
Strobe output is the output signal for adjusting camera exposure timing with external illuminating equipment and it is outputted from GPOUT terminal.
You can configure output period and delay time from basis signal to output start per 1uS.

■ The setting example for software
The sample code to enable strobe output is as follows.

<Sample code>

```
// Enable strobe output
AcapSetInfo(hDev, INPUT_CH, ACL_STROBE_EN, -1, 1);

// Configure strobe polarity as positive
AcapSetInfo(hDev, INPUT_CH, ACL_STROBE_POL, -1, 1);

// Reflect setting
AcapReflectParam(hDev, INPUT_CH);
```

6.8 Timing adjustment of CC output and strobe output

You can configure delay time of CC output and strobe output individually.

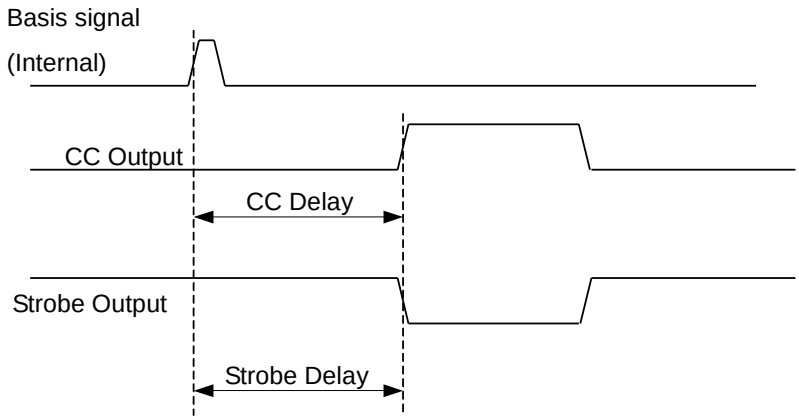


Figure6-16: CC / strobe output timing

CC output and strobe output are outputted as "0", they are outputted as same timing, when begin to supply power.

Downloaded from www.Manualslib.com manuals search engine

You can configure the illumination and exposure timing like the timing below by configuring delay time and output time.

1. Output strobe during camera exposure.

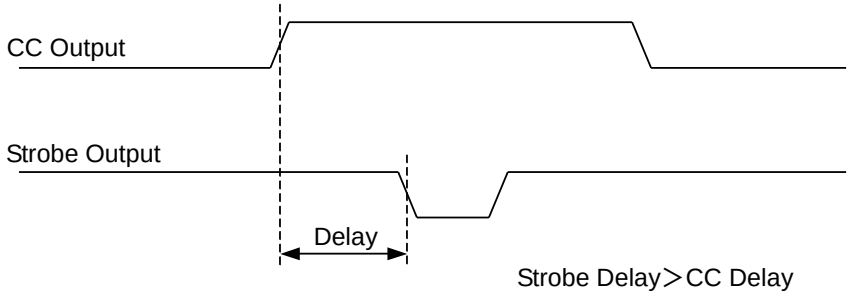


Figure6-17: Strobe output pattern 1

■ The setting example for software

The sample code to configure timing of CC output and strobe output is as follows.
In this code, delay time and output time are as following timing.

- Strobe delay time : 100us
- Strobe output time: 50us
- CC delay time : 10us

<Sample code>

```
// Strobe delay time
AcapSetInfo(hDev, INPUT_CH, ACL_STROBE_DELAY, -1, 100);
// Strobe output time
AcapSetInfo(hDev, INPUT_CH, ACL_STROBE_TIME, -1, 50);
// CC delay time
AcapSetInfo(hDev, INPUT_CH, ACL_CC_DELAY, -1, 10);
// Reflect setting
AcapReflectParam(hDev, INPUT_CH);
```


Downloaded from www.Manualslib.com manuals search engine

2. Exposure camera during output strobe.

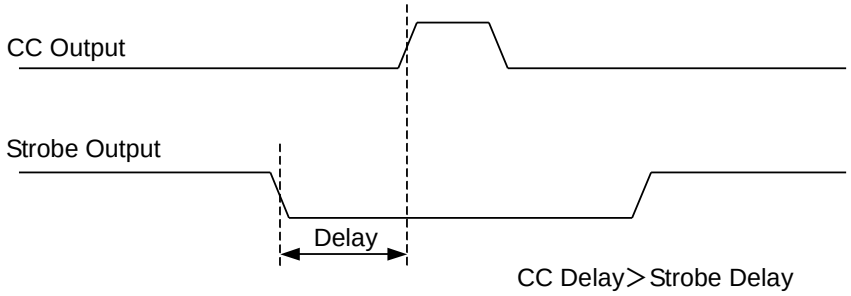


Figure6-18: Strobe output pattern 2

■ The setting example for software

The sample code to configure timing of CC output and strobe output is as follows.
In this code, delay time and output time are as a following timing.

- Strobe delay time : 30us
- Strobe output time: 20us
- CC delay time : 200us

<Sample code>

```
// Strobe delay time
AcapSetInfo(hDev, INPUT_CH, ACL_STROBE_DELAY, -1, 30);
// Strobe output time
AcapSetInfo(hDev, INPUT_CH, ACL_STROBE_TIME, -1, 20);
// CC delay time
AcapSetInfo(hDev, INPUT_CH, ACL_CC_DELAY, -1, 200);
// Reflect setting
AcapReflectParam(hDev, INPUT_CH);
```


Downloaded from www.Manualslib.com manuals search engine

6.9 Synchronizing signal between frame-grabbers (SYNCLT)

APX-3302 has SYNCLT function that can synchronize camera exposure timing between multiple frame-grabbers. The connection diagram using SYNCLT to synchronize is as follows.

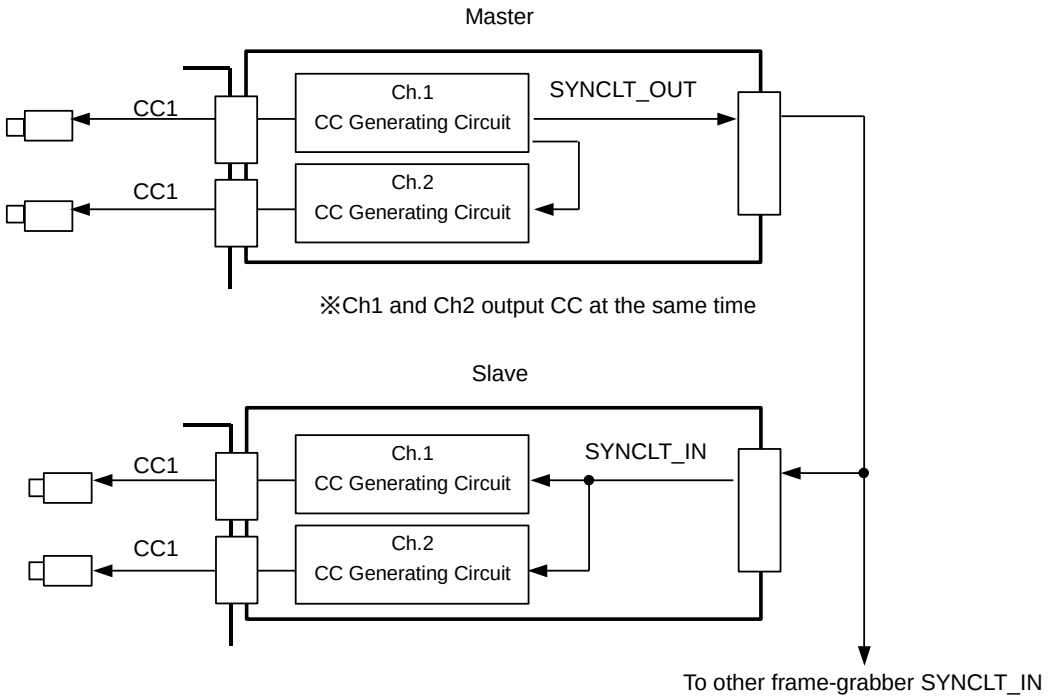


Figure6-19: SYNCLT connection diagram

Downloaded from www.Manualslib.com manuals search engine

6.9.1 Configuration steps using SYNCLT (Area sensor)

Configuration steps for using Area sensor are as follows.

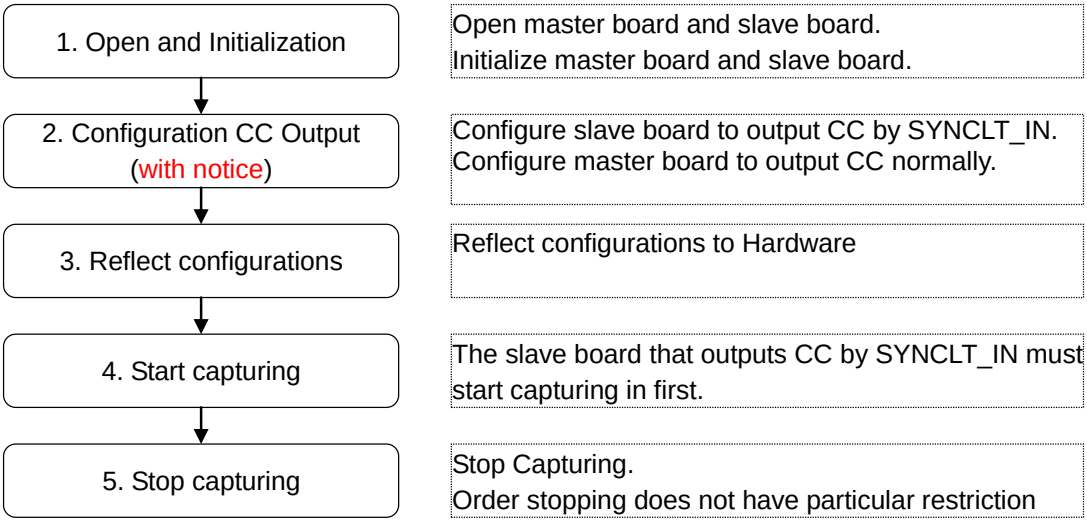


Figure6-20: Software configuration steps : Simple flow(Area sensor)

6.9.2 Configuration steps using SYNCLT (Line sensor)

Configuration steps for using Line sensor are as follows.

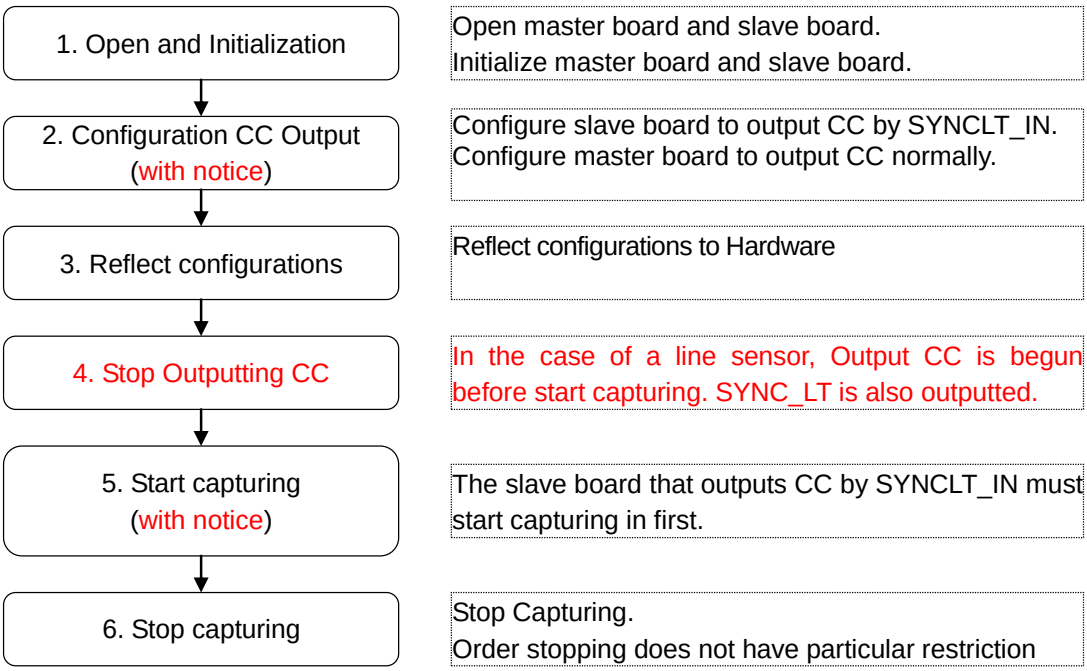


Figure6-21: Software configuration steps : Simple flow (Line sensor)

Downloaded from www.Manualslib.com manuals search engine

■ The setting example for software

The sample code to capture image with 4ch synchronization by SYNCLT function is as follows.

<Sample code>

```
////////////////////////////////////////////////////////////////////////////
//                  Open and initialization                  //
////////////////////////////////////////////////////////////////////////////
// The sample code capture image with 4ch synchronization of two APX-3302.
#define MASTER_BID 0
#define SLAVE_BID 1

// Not to use hDev_Master [0] and hDev_Slave [0]
HANDLE hDev_Master[3] = {INVALID_HANDLE_VALUE};
HANDLE hDev_Slave[3] = {INVALID_HANDLE_VALUE};
int ch_1 = 1;
int ch_2 = 2;
int ch;


// Open channel 1 and 2 of the master board(BoardID=0) and slave board(BoardID=1) respectively
// Configure Output CC enable
for (ch=ch_1; ch<=ch_2; ch++) {// Configure channel 1 and 2.
            hDev_Master[ch] = AcapOpen ("APX-3302", MASTER_BID, ch); // Master board
            hDev_Slave[ch] = AcapOpen ("APX-3302", SLAVE_BID, ch); // Slave board
}

// Initialize basic settings by initial file.
AcapSelectFile(hDev_Master[ch_1], ch_1, "camera.ini", 0);
AcapSelectFile(hDev_Master[ch_2], ch_2, "camera.ini", 0);
AcapSelectFile(hDev_Slave[ch_1], ch_1, "camera.ini", 0);
AcapSelectFile(hDev_Slave[ch_2], ch_2, "camera.ini", 0);
```


Downloaded from www.Manualslib.com manuals search engine

Configure outputting CC as a following timing.

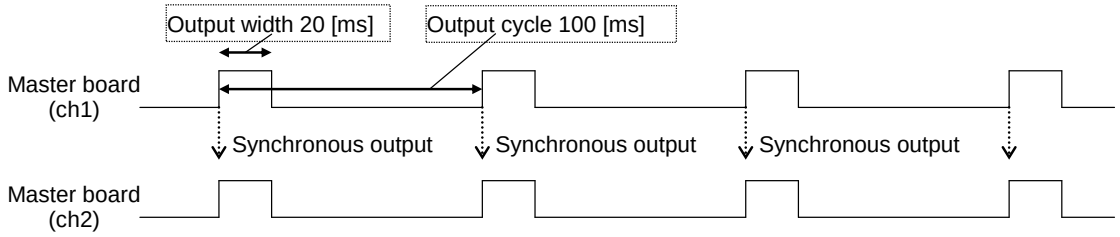


Figure6-22: Output CC cycle and output CC width of the master frame-grabber

```
////////////////////////////////////////////////////////////////////////////
// Configure output CC (master boards ch1 and ch2) //
////////////////////////////////////////////////////////////////////////////

int CCCycle_us;   // CC output cycle
int CCWidth_us;   // CC output width

// CC output setting out of a master board (ch1)
CCCycle_us = 100*1000; // 100[ms]
CCWidth_us = 20*1000;   //   20[ms]

AcapSetInfo(hDev_Master[ch_1], ch_1, ACL_EXP_CYCLE, -1, CCCycle_us);
AcapSetInfo(hDev_Master[ch_1], ch_1, ACL_EXPOSURE, -1, CCWidth_us);


// Configure output CC cycle of the master board (ch2) as output CC width + 1.
CCCycle_us = (20*1000) + 1;   // 20.001[ms]
CCWidth_us = 20*1000;         // 20[ms]

AcapSetInfo(hDev_Master[ch_2], ch_2, ACL_EXP_CYCLE, -1, CCCycle_us);
AcapSetInfo(hDev_Master[ch_2], ch_2, ACL_EXPOSURE, -1, CCWidth_us);


// Configure ch1 and ch2 of master board output CC enable.
AcapSetInfo(hDev_Master[ch_1], ch_1, ACL_EXP_EN, -1, 1);
AcapSetInfo(hDev_Master[ch_2], ch_2, ACL_EXP_EN, -1, 1);

////////////////////////////////////////////////////////////////////////////
// Configure synchronous channel (master board) //
////////////////////////////////////////////////////////////////////////////
// Configure master board ch1 as the parent of master board ch2.
AcapSetInfo(hDev_Master[ch_2], ch_2, ACL_SYNC_CH, -1, ch_1);
```

It makes 4th argument of AcapSetInfo to "-1"and suspends reflection of setting at end.
(Futility of processing time occurs if it is reflected every time.)

By followings, CC output cycle of master board ch2 must be set as "CC output width + 1".

* Both output cycle and width must be set to 1 or more if CC is outputted.
  (Cannot output if it is "0")
* Outputted cycle must be bigger than output width.
* **Cannot get next synchronous request from output CC to finish output cycle period.**

Downloaded from www.Manualslib.com manuals search engine

SYNCLT_OUT that master board outputs is the same wave as CC output of master board. Since master board is combined with CC output and SYNCLT_OUT is also outputted, it is not necessary to configure SyncLT.
The slave board needs to recognize the wave inputted into SYNCLT_IN as external triggering, and needs to set it up to output same CC.

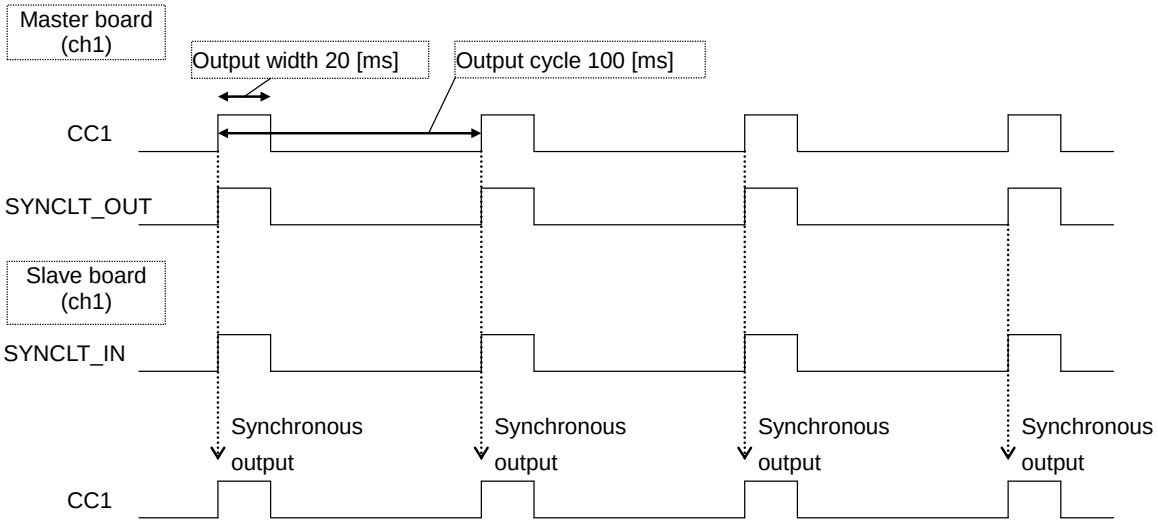


Figure6-23: SYNCLT_OUT and SYNCLT_IN

```
////////////////////////////////////////////////////////////////////////////////
// Configure SYNCLT (master board ch1) (slave board ch2) //
////////////////////////////////////////////////////////////////////////////////
// The following configuration is required to synchronize slave board with SYNCLT.
// Enable external trigger of Slave board ch1
AcapSetInfo(hDev_Slave[ch_1], ch_1, ACL_EXT_EN, -1, 1);

// Configure output CC method as Single-external trigger mode
AcapSetInfo(hDev_Slave[ch_1], ch_1, ACL_EXT_MODE, -1, 0);

// Enable SYNCLT
AcapSetInfo(hDev_Slave[ch_1], ch_1, ACL_SYNC_LT, -1, 1);

// There are some settings that depend on type of sensor.
int SensorType;

// Get sensor type
AcapGetInfo (hDev_Slave[ch_1], ch_1, ACL_SCAN_SYSTEM, 0, &SensorType);)
if(SensorType==0){
            // In the case of Area sensor, enable rolling shutter.
            AcapSetInfo(hDev_Slave[ch_1], ch_1, ACL_ROLLING_SHUTTER, -1, 1);
}else{
          //In the case of Line sensor, use external triggering (SyncLT) as coincidence pulse.
          AcapSetInfo(hDev_Slave[ch_1], ch_1, ACL_ENC_EN, -1, 1);
          AcapSetInfo(hDev_Slave[ch_1], ch_1, ACL_ENC_START, -1, 2);
    }
}
```


Downloaded from www.Manualslib.com manuals search engine

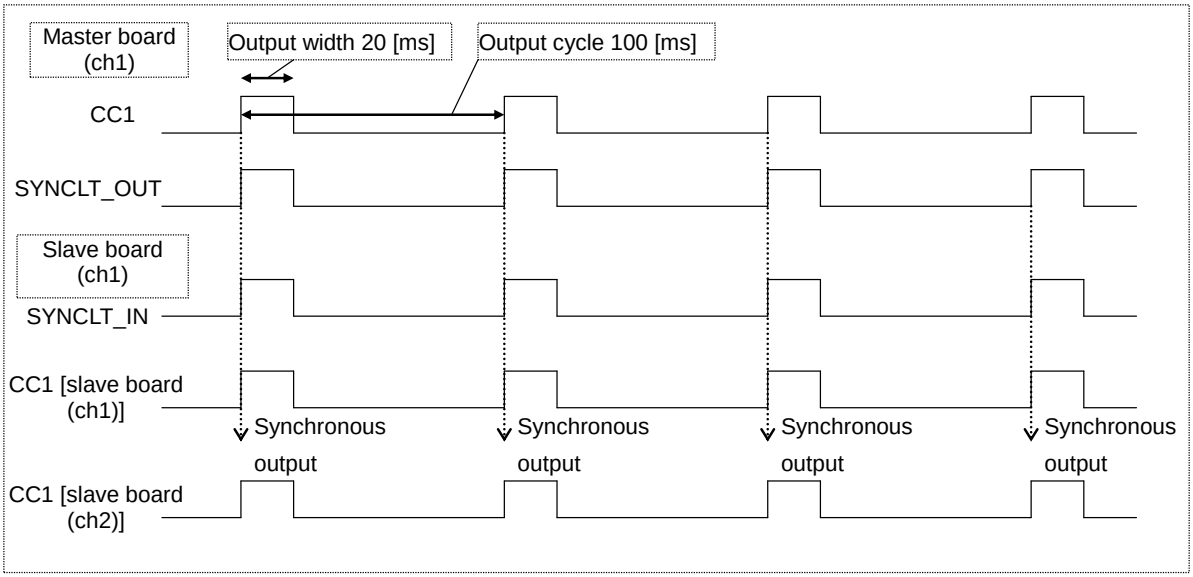


Figure6-24: Output CC cycle and output CC width of slave frame-grabber

```
////////////////////////////////////////////////////////////////////////
//    Configure CC output (slave boards ch1/ch2) //
////////////////////////////////////////////////////////////////////////
// Configure output CC of slave board ch1
// Configure output CC cycle of slave board (ch1) as output CC width + 1.
CCCycle_us = (20*1000) + 1;    // 20.001[ms]
CCWidth_us = 20*1000;          // 20[ms]

AcapSetInfo(hDev_Slave[ch_1], ch_1, ACL_EXP_CYCLE, -1, CCCycle_us);
AcapSetInfo(hDev_Slave[ch_1], ch_1, ACL_EXPOSURE, -1, CCWidth_us);
```

By followings, CC output cycle of slave board ch1 must be set as "CC output width + 1".

* Both output cycle and width must be set to 1 or more if CC is outputted.
  (Cannot output if it is "0")
* Outputted cycle must be bigger than output width.
* **Cannot get next synchronous request from output CC to finish output cycle period.**

```
// Configure output CC cycle of slave board (ch2) as output CC width + 1.
CCCycle_us = 20*1000) + 1;     // 20.001[ms]
CCWidth_us = 20*1000;          // 20[ms]

AcapSetInfo(hDev_Slave[ch_2], ch_2, ACL_EXP_CYCLE, -1, CCCycle_us);
AcapSetInfo(hDev_Slave[ch_2], ch_2, ACL_EXPOSURE, -1, CCWidth_us);

// Enable output CC
AcapSetInfo(hDev_Slave[ch_1], ch_1, ACL_EXP_EN, -1, 1);
AcapSetInfo(hDev_Slave[ch_2], ch_2, ACL_EXP_EN, -1, 1);

////////////////////////////////////////////////////////////////////////
// Configure synchronous channel (slave board) //
////////////////////////////////////////////////////////////////////////
// Configure slave board ch1 as the parent of slave board ch2.
AcapSetInfo(hDev_Slave[ch_2], ch_2, ACL_SYNC_CH, -1, ch_1);
```


Downloaded from www.Manualslib.com manuals search engine

```
////////////////////////////////////////////////////////////////////////
//                    Reflect setting                                 //
////////////////////////////////////////////////////////////////////////
// Reflect setting
AcapReflectParam(hDev_Master[ch_1], ch_1);
AcapReflectParam(hDev_Master[ch_2], ch_2);
AcapReflectParam(hDev_Slave[ch_1], ch_1);
AcapReflectParam(hDev_Slave[ch_2], ch_2);

////////////////////////////////////////////////////////////////////////
//            Specific setting of Line sensor                         //
////////////////////////////////////////////////////////////////////////

// In the case of Line sensor,
// Although you stop capturing, output CC is performed in order to prevent long exposure of camera.
// If you do not use synchronize function between boards, there are no problems in particular.
// But if you use it, capturing will be begun when slave board starts capturing
// because SYNCLT is also outputted with CC.
// There fore, you have to stop output CC before start capturing,
// when you use SYNC_LT with Line sensor.
// This processing is not supported in AcapLib2 standard function,
// you have to configure the register of APX-3302 directly.
// The contents of processing
// In the case of CH1: Set Bit0 of the register 1060h address to 0.
// In the case of CH2: Set Bit0 of the register 2060h address to 0.
DWORD cc_gen;

for (ch=ch_1; ch<=ch_2; ch++) {// Configure channel 1 and 2.
      AcapGetReg (hDev_Master [ch], ch, 0, 0x60+ (ch-1) (*0x1000), &cc_gen);)
      cc_gen = cc_gen & 0xFFFFFFFE;
      AcapSetReg(hDev_Master[ch], ch, 0, 0x60+((ch-1)*0x1000), cc_gen);
}
```


Downloaded from www.Manualslib.com manuals search engine

```
////////////////////////////////////////////////////////////////////
//                    Start capturing                             //
////////////////////////////////////////////////////////////////////

// Register event (For description)
AcapSetEvent(hDev_Master[ch_1], ch_1, ACL_INT_GRABEND, TRUE);
AcapSetEvent(hDev_Master[ch_2], ch_2, ACL_INT_GRABEND, TRUE);
AcapSetEvent(hDev_Slave[ch_1], ch_1, ACL_INT_GRABEND, TRUE);
AcapSetEvent(hDev_Slave[ch_2], ch_2, ACL_INT_GRABEND, TRUE);

// Master ch1 must start capturing in last.
AcapGrabStart(hDev_Slave[ch_2], ch_2, 5);     //Slave ch2
AcapGrabStart(hDev_Slave[ch_1], ch_1, 5);     //Slave ch1
AcapGrabStart(hDev_Master[ch_2], ch_2, 5);    //Master ch2
// Master board starts capturing after 100[ms] (For description)
Sleep(100);
AcapGrabStart(hDev_Master [ch_1], ch_1, 5);    //Master ch1
```

The timing chart to start capturing by the above setting is as follows.

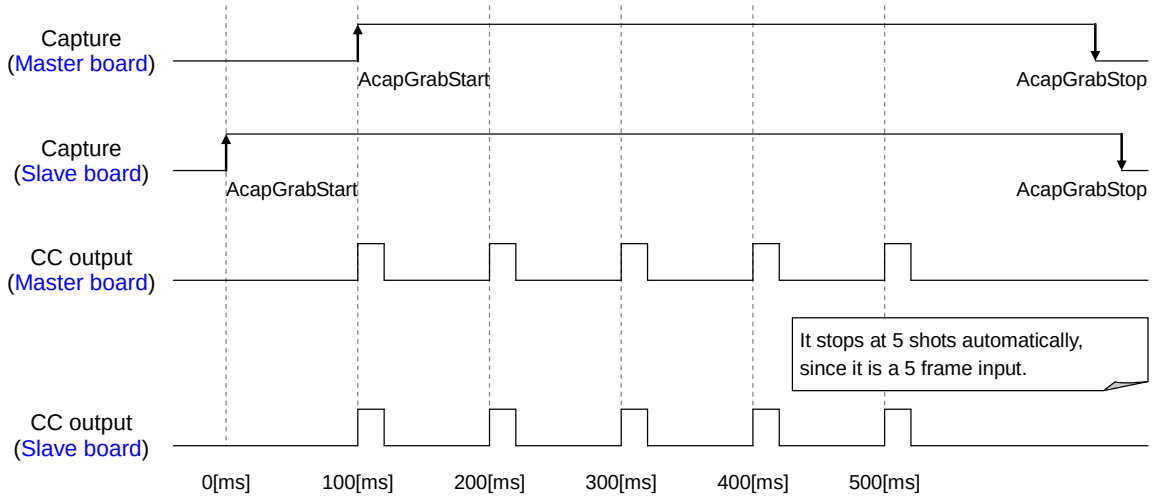


Figure6-25: Capture timing

```
// Waiting for GRABEND event (5 frame input end)
if(ACL_RTN_ERROR == AcapWaitEvent(hDev_Master[ch_1], ch_1, ACL_INT_GRABEND, 10000))
{; TimeOut etc.}
if(ACL_RTN_ERROR == AcapWaitEvent(hDev_Master[ch_2], ch_2, ACL_INT_GRABEND, 10000))
{; TimeOut etc.}
if(ACL_RTN_ERROR == AcapWaitEvent(hDev_Slave[ch_1], ch_1, ACL_INT_GRABEND, 10000))
{; TimeOut etc.}
if(ACL_RTN_ERROR == AcapWaitEvent(hDev_Slave[ch_2], ch_2, ACL_INT_GRABEND, 10000))
{; TimeOut etc.}
```


Downloaded from www.Manualslib.com manuals search engine

```
////////////////////////////////////////////////////////////////////////
//                    Stop capturing                                  //
////////////////////////////////////////////////////////////////////////
// After GRABEND event notified, order of stop is free
AcapGrabStop(hDev_Master[ch_1], ch_1);
AcapGrabStop(hDev_Master[ch_2], ch_2);
AcapGrabStop(hDev_Slave[ch_1], ch_1);
AcapGrabStop(hDev_Slave[ch_2], ch_2);

// Cancel parent setting and Disable SYNCLT for quitting synchronization capture
// Cancel parent setting of slave ch2
AcapSetInfo(hDev_Slave[ch_2], ch_2, ACL_SYNC_CH, 0, 0);

// Cancel parent setting of master ch2
AcapSetInfo(hDev_Master[ch_2], ch_2, ACL_SYNC_CH, 0, 0);

// Disable SYNCLT
AcapSetInfo(hDev_Slave[ch_1], ch_1, ACL_SYNC_LT, 0, 0);
```


Downloaded from www.Manualslib.com manuals search engine

6.10 Encoder input

APX-3302 has encoder input interface.

6.10.1 Encoder input circuit

APX-3302 has the incremental type encoder input circuit of RS422 interface.
It can control exposure and image input timing interlocked with external devices (shaft encoder etc.).

For the count of encoder signal, quad edge evaluation that using encoder A,B phase and double edge evaluation that using only encoder A phase are available. Encoder Z phase is available for using as start timing of encoder.
Moreover, it is also possible to use external trigger signal as a trigger to start encoder count up.(For using external trigger as a trigger to start encoder count up, you need to input trigger for each frame.)

The input frequency of encoder can be inputted to 1 MHz the maximum.

The encoder input timing must be kept the following regulations.

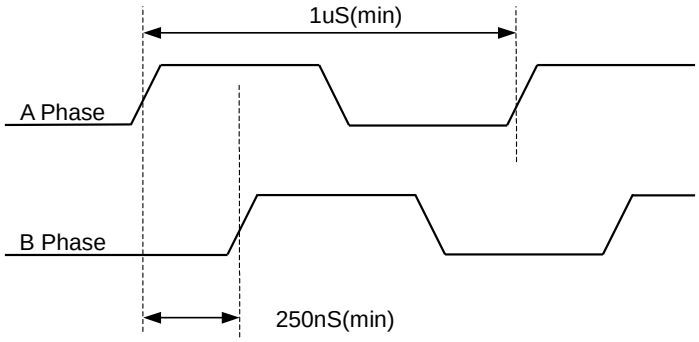


Figure6-26: Encoder input timing

The Duty ratio of an input impulse wave must be kept 50%.

Downloaded from www.Manualslib.com manuals search engine

6.10.2 Encoder count timing

Double edge evaluation count of using A phase only is as follows.

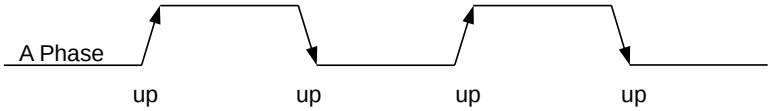


Figure6-27: Double edge evaluation counts of using A phase only

The encoder count of using A phase only performs only addition.

Quad edge evaluation count of using A and B phase is as follows.

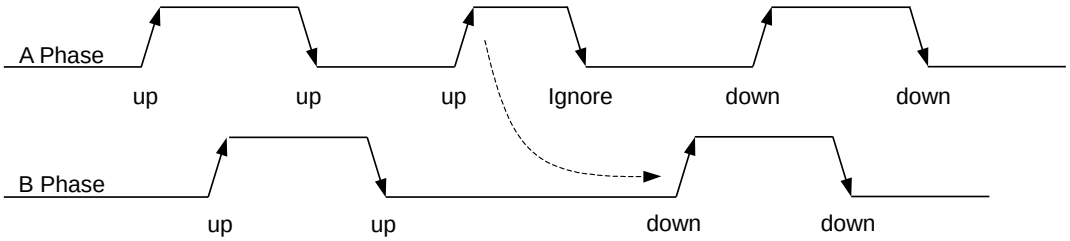


Figure6-28: Quad edge evaluation count of using A and B phase

Quad edge evaluation count performs addition when A phase is located 90 degrees ahead of B phase. And it performs subtraction when B phase is located 90 degrees ahead of A phase.
For preventing miscount by noise, it checks phase when it detects each edges, then it specifies the edge that should be detected.

As shown in the above figure, if phase changes after A phase rising then it waits falling edge of B phase. Even if the other edges are inputted, they will be ignored.

It is also possible to replace addition and subtraction timing by register control.

Downloaded from www.Manualslib.com manuals search engine

6.11 Encoder operation method

APX-3302 has three kinds of encoder operation methods. Each method is explained.

6.11.1 Relative position encoder

Relative position encoder is the mode suitable for the Web inspection using Line camera, etc.
It performs 1 line-scan every time the encoder proceed fixed distance.

It generates a coincidence pulse by using 32-bit counter, Comparison Register 1 (delay pulse configuration) and Comparison Register 2 (Length between coincidence pulses).
Coincidence pulse will be used as trigger generation factor in the internal circuit.

The relationship of pulse number, count value and coincidence pulse is as follows.

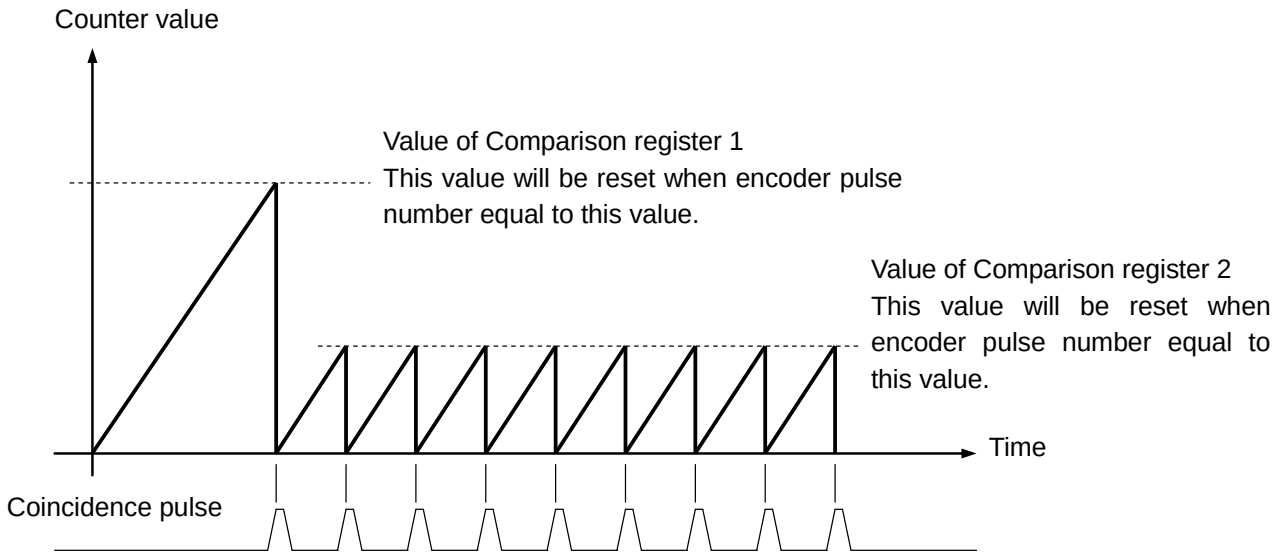


Figure6-29: Relative position encoder

Configure Comparison Register 1 to the number of delay pulses from start capturing to the first scan.
Configure Comparison Register 2 to the interval of line-scan.
Configure trigger for starting count. (Start with external trigger or Z phase, CPU)

Encoder count will be started after finishing configuration above.

Downloaded from www.Manualslib.com manuals search engine

■ The setting example for software

The sample code for capturing by Relative position encoder and Encoder scan mode is as follows.
The setting of encoder is shown below.

Select Encoder : Relative position encoder
Encoder starting method : Not using external trigger
Encoder operational mode : Encoder scan mode
Select Encoder input impulse-wave : Phase AB
Direction of encoder input impulse wave: CW direction
Encoder Z phase input : Not using Z phase
Encoder comparison register 1 : 10
Encoder comparison register 2 : 4

<Sample code>

```
// Relative position encoder
AcapSetInfo(hDev, INPUT_CH, ACL_ENC_EN, -1, 1);
// Not using external trigger
AcapSetInfo(hDev, INPUT_CH, ACL_ENC_START, -1, 0);
// Encoder scan mode
AcapSetInfo(hDev, INPUT_CH, ACL_ENC_MODE, -1, 0);
// Phase AB
AcapSetInfo(hDev, INPUT_CH, ACL_ENC_PHASE, -1, 0);
// CW
AcapSetInfo(hDev, INPUT_CH, ACL_ENC_DIRECTION, -1, 0);
// Not using Z phase
AcapSetInfo(hDev, INPUT_CH, ACL_ENC_ZPHASE_EN, -1, 0);
// Configure comparison register 1 as 10.
AcapSetInfo(hDev, INPUT_CH, ACL_ENC_COMPARE_1, -1, 10);
// Configure comparison register 2 as 4.
AcapSetInfo(hDev, INPUT_CH, ACL_ENC_COMPARE_2, -1, 4);
// Reflect setting
AcapReflectParam(hDev, INPUT_CH);
```


Downloaded from www.Manualslib.com manuals search engine

6.11.2 Absolute position encoder single point

Absolute position encoder single point is the mode that is suitable for capturing using stage.
For Relative position encoder mode, it resets counter every time pulse coincidence.
For this mode, it always keeps moving distance (counter value) from starting point, so it can be certain start position on the stage that performs proceeding and returning repeatedly.

It has 32-bit absolute counter and 16-bit interval counter.
For comparison register 1, configure absolute counter value that starts interval counter.
For comparison register 2, configure interval counter that counts number of pulse that generates coincidence pulse. After finishing configuration of comparison register 1 and 2, configure start condition of encoder count, then starts count.

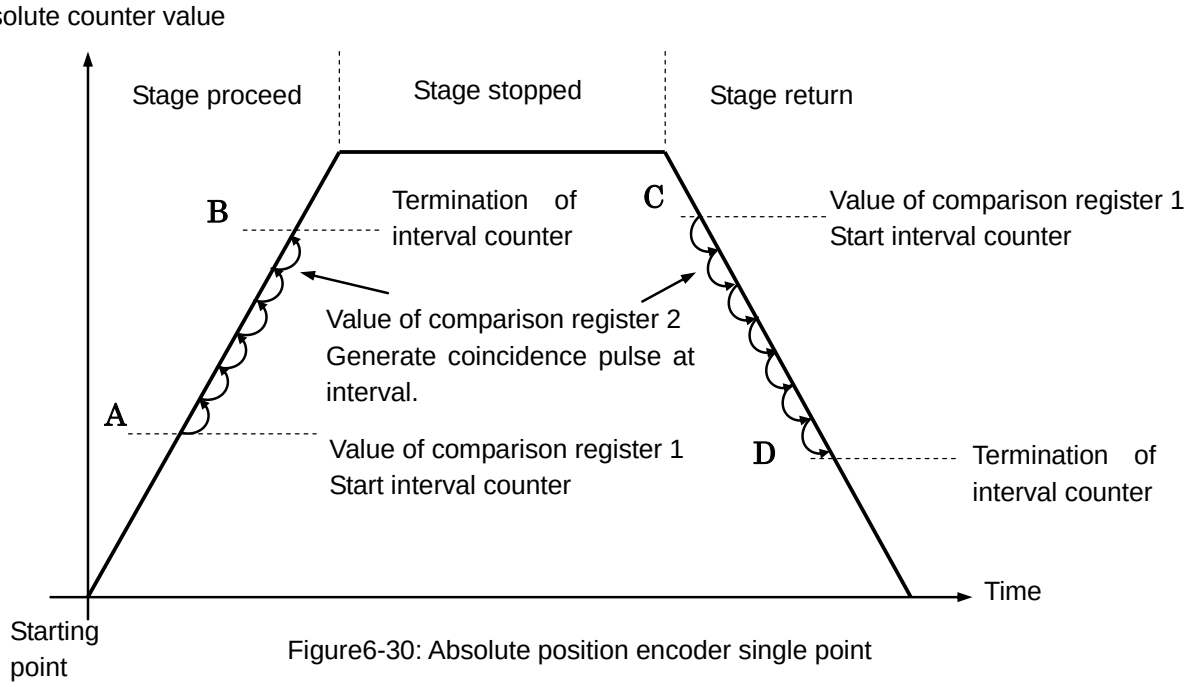


Figure6-30: Absolute position encoder single point

- Absolute counter is not reset even if it coincides with comparison register 1. It always keeps the value.
- Interval counter is not reset even if it coincides with comparison register 2.
- Coincidence pulse is outputted when the value of comparison register 1, 2 coincide with each counter value.

In the above-mentioned figure, configure comparison register 1 as start position A of proceeding direction, and configure comparison register 2 as generating interval of coincidence pulse.
After start capturing, when absolute counter becomes higher than B, it stops interval count.
Next, configure comparison register 1 as start position C of returning direction, and configure comparison register 2 as generating interval of coincidence pulse, then enable interval count.
It stops interval count and absolute count when capturing of returning direction is finished and absolute count becomes lower than D.

Downloaded from www.Manualslib.com manuals search engine

■ The setting example for software

The sample code for capturing 50 lines at round-trip each by Absolute position encoder single point and Encoder scan mode is as follows. The setting of encoder and the coordinate of each point that is shown at "Figure6-30: Absolute position encoder single point" is shown below.

[Encoder setting]

Select Encoder : Absolute position encoder single point
Encoder starting method : Not using external trigger
Encoder operational mode : Encoder scan mode
Select Encoder input impulse-wave : Phase AB
Direction of encoder input impulse wave: CW direction
Encoder Z phase input : Not using Z phase
Encoder comparison register 1 : 5000
Encoder comparison register 2 : 100

[Coordinates of every point]

A:point 5000 count
B:point 9900 count
C:point 9900 count
D:point 5000 count

<Sample code>

```
////////////////////////////////////////////////////////////////////////////////////////////
// starting point : Configure outward setting and start capturing //
////////////////////////////////////////////////////////////////////////////////////////////
// Outward trip : Start capturing at absolute encoder count "5000"
// After that, capture 1 line every time counting "100" counts, and capture "50" lines.

// Configure encoder setting (outward trip)
AcapSetEncoder(hDev, INPUT_CH, 2, 0, 0, 0, 0, 5000, 100);
// Start encoder count
AcapSetInfo(hDev, INPUT_CH, ACL_ENC_ABS_START, 0, 1);
// Snap
AcapGrabStart(hDev, INPUT_CH, 1)
```

```
////////////////////////////////////////////////////////////////////////
//    Point A :   Start capturing of outward trip    //
////////////////////////////////////////////////////////////////////////
// Coincidence pulse will be generated and start capturing, when absolute count reaches "5000".
// You can check current count value by the 11th argument of "AcapGetEncoder" or "ACL_ENC_ABC_COUNT".

// Get absolute position encoder count
ULONG ulAbsCount;
AcapGetEncoder (hDev, INPUT_CH, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, &ulAbsCount);)
AcapGetInfo (hDev, INPUT_CH, ACL_ENC_ABS_COUNT, 0, &ulAbsCount);)
```


Downloaded from www.Manualslib.com manuals search engine

```
////////////////////////////////////////////////////////////////////////////
//      Point B: Finish capturing of outward trip     //
////////////////////////////////////////////////////////////////////////////
// Outward trip : Finish "50" line acquiring. Then stop capturing and configure return trip setting
// Start capturing at absolute encoder count "9900". Configure as "CCW".

// Stop capturing (waiting for "50" line input)
AcapGrabStop(hDev, INPUT_CH);
// Configure encoder (return trip)
AcapSetEncoder(hDev, INPUT_CH, 2, 0, 0, 0, 1, 0, 9900, 100);

//--- Since encoder count is already started, no need to configure here. ---

// Snap
AcapGrabStart(hDev, INPUT_CH, 1)
```

```
////////////////////////////////////////////////////////////////////////////
//        Point C : Start capturing of return trip       //
////////////////////////////////////////////////////////////////////////////
// Coincidence pulse will be generated and start capturing, when absolute count reaches "9900".
// You can check current count value by the 11th argument of "AcapGetEncoder" or "ACL_ENC_ABC_COUNT".

// Get counting value
ULONG ulAbsCount;
AcapGetEncoder (hDev, INPUT_CH, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, &ulAbsCount);)
AcapGetInfo (hDev, INPUT_CH, ACL_ENC_ABS_COUNT, 0, &ulAbsCount);)
```

```
////////////////////////////////////////////////////////////////////////////
//        Point D : Finish capturing of return trip     //
////////////////////////////////////////////////////////////////////////////
// Return trip : Finish "50" line acquiring. Then stop capturing and configure return trip setting
// Start capturing at absolute encoder count "5000". Configure as "CW".

// Stop capturing (waiting for "50" line input)
AcapGrabStop(hDev, INPUT_CH);
// Configure as CW (outward trip)
AcapSetEncoder(hDev, INPUT_CH, 2, 0, 0, 0, 0, 0, 5000, 100);

//--- Since encoder count is already started, no need to configure here. ---

// Snap
AcapGrabStart(hDev, INPUT_CH, 1)
```


Downloaded from www.Manualslib.com manuals search engine

6.11.3 Absolute position encoder multi point

Absolute position encoder single point is the expanded mode of Absolute position encoder multi point.
It can generate coincidence pulse at any points you choose up to 256 during counting absolute count.
This mode is suitable for capturing using stage by Area camera, for example capturing interval is not constant.

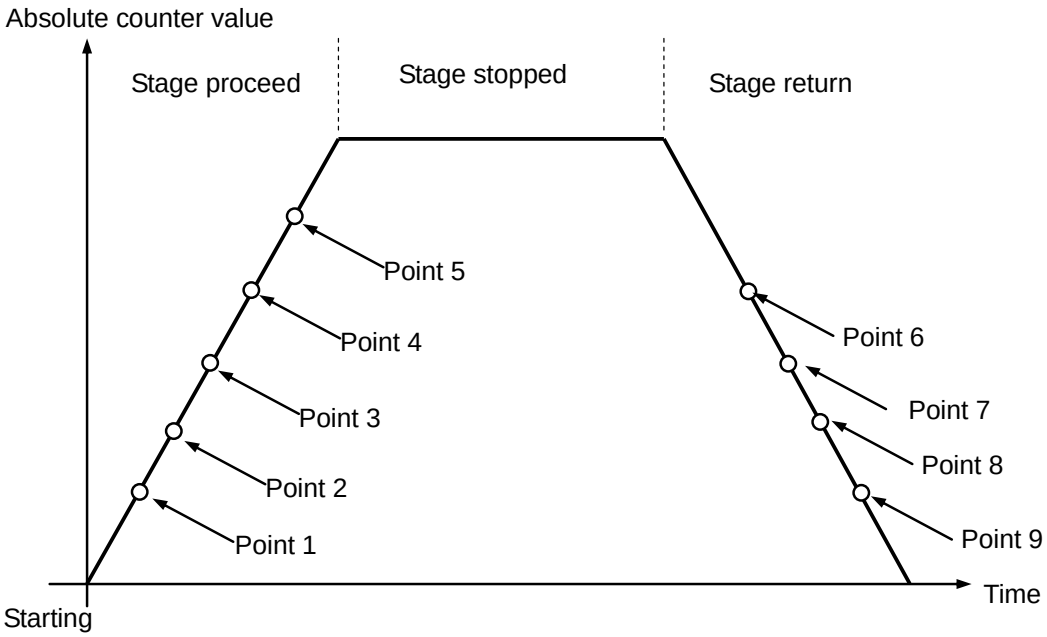


Figure6-31: Absolute position encoder multi point

Comparison register does not exist in this mode.
Instead of comparison register, it has 256 comparison memories that are mapped in internal register area, and it can generate coincidence pulse at certain count value.

■ The setting example for software

The sample code for capturing from point 1 to point 9 each by Absolute position encoder multi point is as follows. The setting of encoder and the coordinate of each point that is shown at "Figure6-31: Absolute position encoder multi point" is shown below.

[Encoder setting]

Select Encoder	: Absolute position encoder multi point
Encoder starting method	: Not using external trigger
Encoder operational mode	: Encoder scan mode
Select Encoder input impulse-wave	: Phase AB
Direction of encoder input impulse wave	: CW direction
Encoder Z phase input	: Not using Z phase

Downloaded from www.Manualslib.com manuals search engine

[Coordinates of every point]

Point 1: 1000 count
Point 2: 2000 count
Point 3: 3000 count
Point 4: 4000 count
Point 5: 5000 count
Point 6: 4000 count
Point 7: 3000 count
Point 8: 2000 count
Point 9: 1000 count

[Start capturing / Finish capturing point]

Outward-trip start point : Point 1 (1000 count)
Outward-trip finish point :Point 5 (5000 count)
Return-trip start point : Point 6 (4000 count)
Return-trip finish point : Point 9 (1000 count)

Downloaded from www.Manualslib.com manuals search engine

<Sample code>

```
////////////////////////////////////////////////////////////////////////////////////////////////////
// Starting point: Configure comparison memory (outward and return trip) //
////////////////////////////////////////////////////////////////////////////////////////////////////
// Outward trip : Start capturing at absolute encoder count "1000"
// After that, capture when counting reaches "2000","3000","4000" or "5000".
// It captures 5 times in total.

#define INPUT_CH 1
DWORD dwRegOffset= 0;       // register address
DWORD dwRegVal= 0;          // register setting value
int nStartPos= 0;           // starting position
int nEndPos= 0;             // end position
int i= 0;                   // variable for loop formations

// Outward trip: Configure comparison memory
// Standard functions of AcapLib2 do not support this processing,
// so you have to configure register of APX-3302 directly.
// CH1: Configure comparison memory from register address 5000h
// CH2: Configure comparison memory from register address 6000h

for(i = 0; i < 5; i++){
        // Specify the counter value for capturing
        dwRegVal = 1000 + (i * 1000);
        // Specify comparison memory
        dwRegOffset = 0x5000 + (i * 4) + (INPUT_CH - 1) * APX3302_CH_OFFSET;
        // Configure counter to specified comparison memory
        AcapSetReg(ghDev, INPUT_CH, 0, dwRegOffset, dwRegVal);
}

// Outward trip: Configure comparison memory
for(i = 5; i < 9; i++){
        // Specify the counter value for capturing
        dwRegVal = 4000 - ((i - 5) * 1000);
        // Specify comparison memory
        dwRegOffset = 0x5000 + (i * 4) + (INPUT_CH - 1) * APX3302_CH_OFFSET;
        // Configure counter to specified comparison memory
        AcapSetReg(ghDev, INPUT_CH, 0, dwRegOffset, dwRegVal);
}
```

The address of comparison memory register is 5000h for ch1 and 6000h for ch2.
For program above, it is configured as shown below table.

Comparison memory No	Address	Counter value
Comparison memory 0	5000h	1000
Comparison memory 1	5004h	2000
Comparison memory 2	5008h	3000
Comparison memory 3	500Ch	4000
Comparison memory 4	5010h	5000
Comparison memory 5	5014h	4000
Comparison memory 6	5018h	3000
Comparison memory 7	501Ch	2000
Comparison memory 8	5020h	1000

Table6-3: The example of comparison memory setting

Downloaded from www.Manualslib.com manuals search engine

```
// Outward trip: Specify comparison memory starting position and finish position
nStartPos = 0;
nEndPos = 4;
dwRegVal = (0x000000FF & nStartPos) << 8;
dwRegVal | = 0x000000FF & nEndPos;
// Configure the comparison memory starting position and finish position
AcapSetInfo(ghDev, INPUT_CH, ACL_ENC_COMPARE_2, 0, dwRegVal);
```

For configure comparison memory starting position and finish position, configure comparison register 2 by using AcapSetInfo function. The correspondence bit of starting position and finish position is as follows.

Comparison memory	Correspondence bit
Starting position	Bit15-Bit8
Finish position	Bit7 -Bit0

Table6-4: Correspondence bit

For instance, if you use starting position as comparison memory 0 value and finish position as comparison memory 4 value, configure comparison register 2 as "0004h".

Outward trip starting position →

Outward trip end position →

Comparison memory No	Address	Counter set value
Comparison memory 0	5000h	1000
Comparison memory 1	5004h	2000
Comparison memory 2	5008h	3000
Comparison memory 3	500Ch	4000
Comparison memory 4	5010h	5000
Comparison memory 5	5014h	4000
Comparison memory 6	5018h	3000
Comparison memory 7	501Ch	2000
Comparison memory 8	5020h	1000

Table6-5: Starting position and end position of outward trip

Downloaded from www.Manualslib.com manuals search engine

```
// Enable multipoint
// Standard functions of AcapLib2 do not support this processing,
// so you have to configure register of APX-3302 directly.

// CH1: Configure bit30 of register address 1028h as 1
// CH2: Configure bit30 of register address 2028h as 1

// Get the value of encoder control register
dwRegOffset = 0x1028 + (INPUT_CH - 1) * APX3302_CH_OFFSET;
AcapGetReg(ghDev, INPUT_CH, 0, dwRegOffset, &dwRegVal);)

// Change to multipoint mode (configure bit30 as 1)
dwRegVal | = 0x40000000;

// Configure encoder control register
AcapSetReg(ghDev, INPUT_CH, 0, dwRegOffset, dwRegVal);
```

Change absolute position encoder type to multi point. Configure bit30 of APX-3302 encoder control register as 1 by using AcapSetReg function. Do not change other bit.
The address of encoder control register is 1028h for ch1 and 2028h for ch2.

bit	Function
30	Select absolute position encoder type. 0: Single point 1: Multi point Power on status is "0": Single point

Table6-6: Encoder control register (bit30)

Downloaded from www.Manualslib.com manuals search engine

```
// Start counting of multi point
// Standard functions of AcapLib2 do not support this processing,
// so you have to configure register of APX-3302 directly.

// CH1: Configure bit17 of register address 1028h as 1
// CH2: Configure bit17 of register address 2028h as 1

// Get the value of encoder control register
dwRegOffset = 0x1028 + (INPUT_CH - 1) * APX3302_CH_OFFSET;
AcapGetReg(ghDev, INPUT_CH, 0, dwRegOffset, &dwRegVal);)

// Start counting of multi point mode (configure bit17 as 1)
dwRegVal | = 0x20000;

// Configure encoder control register
AcapSetReg(ghDev, INPUT_CH, REGTYPE_LOCAL, dwRegOffset, dwRegVal);

// Outward trip: Start frame number specifying capturing (Capture 5 frames)
AcapGrabStart(ghDev, INPUT_CH, 5);
```

Start counting of absolute position encoder multi point. Configure bit17 of APX-3302 encoder control register as 1 by using AcapSetReg function. Do not change other bit.
The address of encoder control register is 1028h for ch1 and 2028h for ch2.

bit	Function
17	Configure counting start/stop of absolute position encoder. 0: Stop multipoint Count 1: Start multipoint Count When you write this bit as "0" during operation, abort counting. Power on status is "0".

Table6-7: Encoder control register (bit17)

```
////////////////////////////////////////////////////////////////////////////
// point 1 - point 4: Outward trip, start capturing //
////////////////////////////////////////////////////////////////////////////
// When absolute count reaches "1000", coincidence pulse is generated and capture 1 time.
// After that, capture when counting reaches "2000","3000","4000" or "5000".
// It captures 5 times in total.
// You can get latest value of comparison memory by 10th argument of "AcapGetEncoder" function,
// and can get current value by "ACL_ENC_ABC_COUNT".
ULONG ulCompMem, ulAbsCount;
// Get the latest value of comparison memory
AcapGetEncoder(hDev, INPUT_CH, NULL, NULL, NULL, NULL, NULL, NULL, NULL, &ulCompMem, NULL);
// Get the value of absolute position encoder count
AcapGetInfo(hDev, INPUT_CH, ACL_ENC_ABS_COUNT, 0, &ulAbsCount);)
```


Downloaded from www.Manualslib.com manuals search engine

```
////////////////////////////////////////////////////////////////////////////
//       point 5: Outward trip, finish capturing        //
////////////////////////////////////////////////////////////////////////////
// Outward trip: Finish 5 times capturing, then stop capturing and configure return trip.
// Return trip: Specify comparison memory starting position and finish position.

// Stop capturing
AcapGrabStop(ghDev, INPUT_CH);

// Return trip: Specify comparison memory starting position and finish position.
nStartPos = 5;
nEndPos = 8;
dwRegVal    = (0x000000FF & nStartPos) << 8;
dwRegVal | = 0x000000FF & nEndPos;

// Configure comparison memory starting position and finish position.
AcapSetInfo(ghDev, INPUT_CH, ACL_ENC_COMPARE_2, 0, dwRegVal);

// Not to change to CCW, because counter will be count-up at return trip.
//--- Since encoder count is already started, no need to configure here. ---

// Return trip: Start frame number specifying capturing (Capture 4 frames)
AcapGrabStart(ghDev, INPUT_CH, 4);
```

For instance, if you use starting position as comparison memory 5 value and finish position as comparison memory 8 value, configure comparison register 2 as "0508h".

	Comparison memory No	Address	Counter set value
	Comparison memory 0	5000h	1000
	Comparison memory 1	5004h	2000
	Comparison memory 2	5008h	3000
	Comparison memory 3	500Ch	4000
Outward trip starting position →	Comparison memory 4	5010h	5000
	Comparison memory 5	5014h	4000
	Comparison memory 6	5018h	3000
	Comparison memory 7	501Ch	2000
Outward trip end position →	Comparison memory 8	5020h	1000

Table6-8: Starting position and end position of return trip

Downloaded from www.Manualslib.com manuals search engine

```
////////////////////////////////////////////////////////////////////////////
//    point 6 - point 8: Return trip, start capturing    //
////////////////////////////////////////////////////////////////////////////
// When absolute count reaches "4000", coincidence pulse is generated and capture 1 time.
// After that, capture when counting reaches "3000","2000" or "1000".
// It captures 5 times in total.
// You can get latest value of comparison memory by 10th argument of "AcapGetEncoder" function,
// and can get current value by "ACL_ENC_ABC_COUNT".
ULONG ulCompMem, ulAbsCount;
// Get the latest value of comparison memory
AcapGetEncoder(hDev, INPUT_CH, NULL, NULL, NULL, NULL, NULL, NULL, &ulCompMem, NULL);
// Get the value of absolute position encoder count
AcapGetInfo(hDev, INPUT_CH, ACL_ENC_ABS_COUNT, 0, &ulAbsCount);
```

```
////////////////////////////////////////////////////////////////////////////
//         point 9: Return trip, finish capturing          //
////////////////////////////////////////////////////////////////////////////
// Return trip: Finish 4 times capturing, then stop capturing and configure return trip.
// Outward trip: Specify comparison memory starting position and finish position.

// Stop capturing
AcapGrabStop(ghDev, INPUT_CH);

// Outward trip: Specify comparison memory starting position and finish position.
nStartPos = 0;
nEndPos = 4;
dwRegVal = (0x000000FF & nStartPos) << 8;
dwRegVal | = 0x000000FF & nEndPos;

// Configure comparison memory starting position and finish position.
AcapSetInfo(ghDev, INPUT_CH, ACL_ENC_COMPARE_2, 0, dwRegVal);

//--- Since encoder count is already started, no need to configure here. ---

// Outward trip: Start frame number specifying capturing (Capture 5 frames)
AcapGrabStart(ghDev, INPUT_CH, 5);
```


Downloaded from www.Manualslib.com manuals search engine

6.12 MEGA-FIFO

APX-3302 can extract part from effective pixels of the camera, and input it into the image buffer.

The image transfer function of APX-3302 performs that stores the image from the camera to large capacity image input buffer temporarily, and executes DMA transfer to PC host memory via buffer.
The DMA transfer function via large capacity buffer guarantees the acquiring image in the fastest operation of camera. This architecture is called MEGA-FIFO architecture.

By taking FIFO composition, the start of DMA is attained simultaneously with image input start, and a tact time can be shortened. Moreover, even when the band of PCI Express is temporarily less than the frame rate of camera, image input is possible until the image input buffer becomes full.
As shown in the following figure, store image into image buffer is configurable by parameters below.
* TRG_DELAY : Number of delay lines from FVAL to start capturing
* Y_TOTAL : Number of lines that frame-grabber acquire
* X_DELAY : Number of pixels from LVAL to start capturing
* X_SIZE : Number of pixels of width that frame-grabber acquire

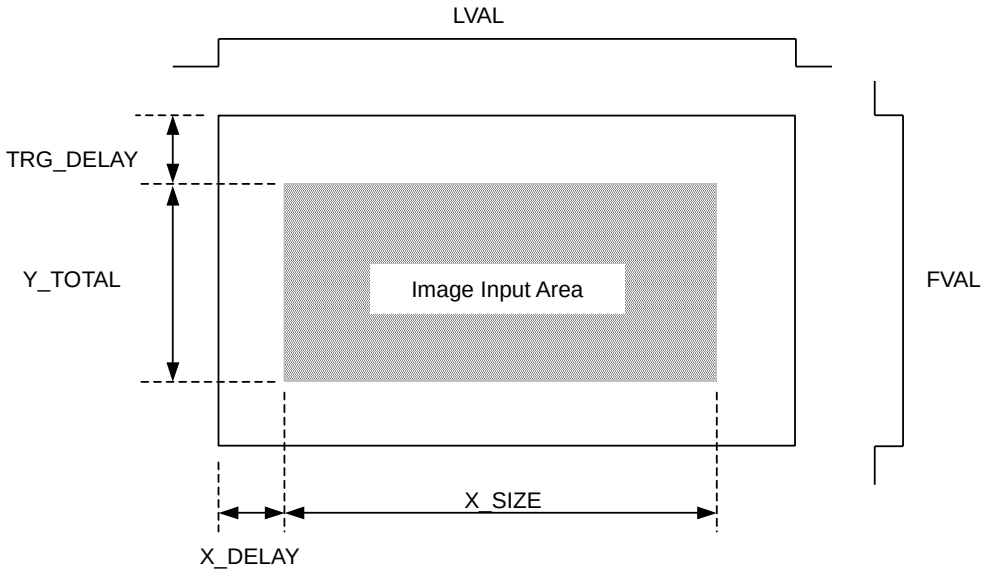


Figure6-31: Image input configuration

The image data stored in the image buffer is transmitted to a host by DMA function on frame-grabber.
There are DMA controllers of the same number as the number of channels, and operate independently.

DMA has transmission restrictions in a 128-bit unit. DMA transfer in a 64 bit/32bit unit may be supported in the future.

The image buffer has 64MByte per channel. If a buffer becomes full status, it outputs "FIFO overflow error" and stops capturing.

Downloaded from www.Manualslib.com manuals search engine

■ The setting example for software

The sample code to configure input size is as follows.

In this code, the image input range and each signal timing are as a following figure.

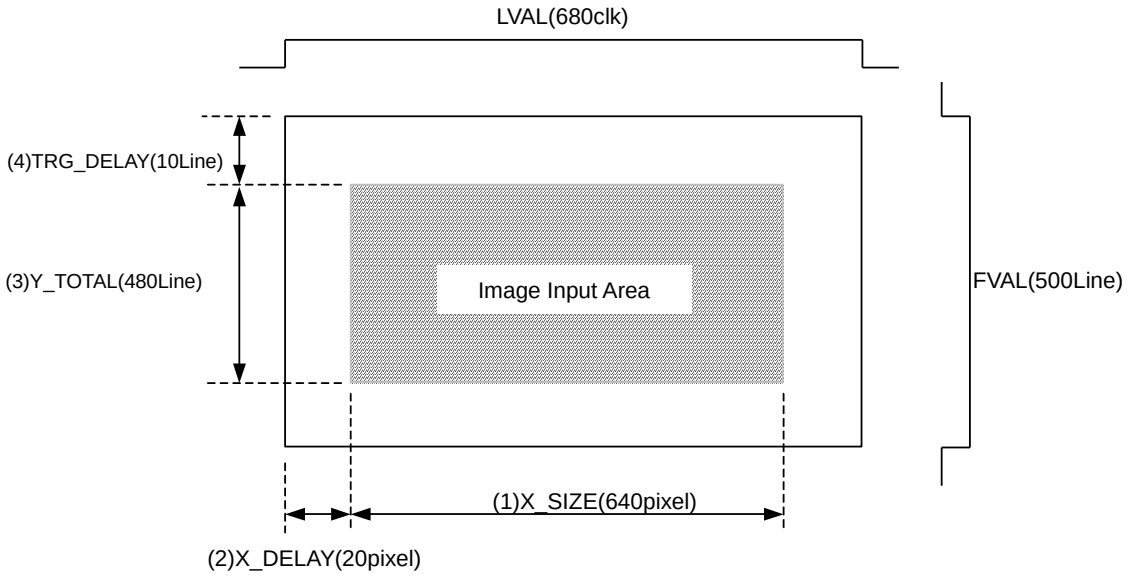


Figure6-32: Image input configuration

<Sample code>

```
// (1)X_SIZE
AcapSetInfo(hDev, INPUT_CH, ACL_X_SIZE, -1, 640); // Configure ACL_X_SIZE

// (2)X_DELAY
// For APX-3302, use not ACL_X_DELAY but ACL_LVAL_DELAY for configure ID.
// For 1TAP camera, configure LVAL_DELAY as same value of X_DELAY.
// For 2TAP camera, configure LVAL_DELAY as one-half value of X_DELAY.
// For 3TAP camera, configure LVAL_DELAY as one-third value of X_DELAY.
AcapSetInfo(hDev, INPUT_CH, ACL_LVAL_DELAY, -1, 20); // Configure ACL_LVAL_DELAY

// (3)Y_TOTAL
// Configure Y size and Y total as the same value, unless you use infinite length capturing.
AcapSetInfo(hDev, INPUT_CH, ACL_Y_SIZE, -1, 480); // Configure ACL_Y_SIZE
AcapSetInfo(hDev, INPUT_CH, ACL_Y_TOTAL, -1, 480); // Configure ACL_Y_TOTAL

// (4)TRG_DELAY:
AcapSetInfo(hDev, INPUT_CH, ACL_Y_DELAY, -1, 10); // Configure ACL_Y_DELAY

//Reflection of a set point
AcapReflectParam(hDev, INPUT_CH);
```


Downloaded from www.Manualslib.com manuals search engine

■ Setting range and Restriction

ACL_X_SIZE, ACL_Y_SIZE, ACL_Y_TOTAL, ACL_LVAL_DELAY, and ACL_Y_DELAY have the following restrictions, respectively. Therefore, Be aware to use each configuration ID.

Configuration ID	Setting range and Restriction	
ACL_X_SIZE	8bit/1tap	16-8192 (multiple of 16)
	8bit/2tap	16-16384 (multiple of 16)
	8bit/3tap	48-24576 (multiple of 48)
	10 bit - 16 bit / 1tap	8-8192 (multiple of 8)
	10 bit - 12 bit / 2tap	8-16384 (multiple of 8)
	24bitRGB	16-8192 (multiple of 16)
	32bitRGB	8-8192 (multiple of 8)
ACL_Y_SIZE	1-16777215	
ACL_Y_TOTAL	0-16777215	
ACL_LVAL_DELAY	0-65535	
ACL_Y_DELAY	0-1023	

Table6-9: Setting range and Restriction

Downloaded from www.Manualslib.com manuals search engine

6.13 Interruption

APX-3302 can output one level interruption to PCI Express.
You can choose multiple interrupts as a followings.
- DMA end interrupt
- Image input start interruption
- Input (GPIN) interruption in general

These interrupts is notified by the event or callback function.

- Event notification (AcapSetEvent/AcapWaitEvent)
- Callback function (AcapRegistCallback/AcapRegistCallbackEx)

Followings show definition of interrupts that able to notice.

Interrupt	Definition	Meaning
Input start interrupt	ACL_INT_GRABSTART	It notifies that starts data input of 1 frame.
1 frame input completion interrupt	ACL_INT_FRAMEEND	It notifies that finishes transferring data of 1 frame to buffer.
Input end interrupt	ACL_INT_GRABEND	It notifies that complete capturing of specified frame number if number of input frame is specified by InputNum of AcapGrabStart. It is not notified at streaming with InputNum=0.
GPIN interrupt	ACL_INT_GPIN	It notifies that negative going edge of GPIN (general input signal), which is assigned at specified channel, is detected.

Table6-10: The definition of interrupts

Downloaded from www.Manualslib.com manuals search engine

6.13.1 Event usage

This section describes standby method of interrupt by notifying event.

■ Setting

Use AcapSetEvent for setting up interrupt which wants to stand by.
Multiple interrupt is able to be set, but please register interrupt event one by one.

■ Standby

The Interrupt set by AcapSetEvent can be standby state by AcapWaitEvent.

■ The setting example for software

The sample code to wait "Input start interrupt" and "1 frame input completion interrupt " is as follows.

<Sample code>

```
// Open APX-3302
int rtn;
HANDLE hDevice;
int channel = 1;
hDevice = AcapOpen("APX-3302", 0, channel);

// Initialize by ini file.
rtn = AcapSelectFile(hDevice, channel, "camera.ini", 0);

// Set "Input start interrupt" and "1 frame input completion interrupt"
rtn = AcapSetEvent(hDevice, channel, ACL_INT_GRABSTART, TRUE);
rtn = AcapSetEvent(hDevice, channel, ACL_INT_FRAMEEND, TRUE);

// Start capturing (Grab)
rtn = AcapGrabStart(hDevice, channel, 0);

// Wait to start capturing
int GrabStartTimeout_ms = 1000;
rtn = AcapWaitEvent(hDevice, channel, ACL_INT_GRABSTART, GrabStartTimeout_ms);

// Wait to be notified "1 frame input completion interrupt"
int FrameEndTimeout_ms = 1000;
rtn = AcapWaitEvent(hDevice, channel, ACL_INT_FRAMEEND, FrameEndTimeout_ms);
```


Downloaded from www.Manualslib.com manuals search engine

■ Cancel standby (abort)

In AcapWaitEvent, process does not return until it receives interrupt or timeout occurs.
When AcapSetEvent is performed by using ACL_INT_ABORT flag from other thread, standby process of AcapWaitEvent can be abort.

■ The setting example for software

The sample code to abort the thread (Thread_WaitFrameEnd) that is waiting to be notified under ACL_INT_FRAMEEND is as follows.

<Sample code>

```
// Specify ACL_INT_ABORT and the event that will be aborted
rtn = AcapSetEvent(hDevice, channel, ACL_INT_FRAMEEND | ACL_INT_ABORT, TRUE);
```

```
//Thread which waits to be notified "1 frame input completion interrupt"
void Thread_WaitEvent(void* pParam)
{
   // Omit here
   // Wait to be notified "1 frame input completion interrupt"
   int FrameEndTimeout_ms = 10 * 1000;
   rtn = AcapWaitEvent(hDevice, channel, ACL_INT_FRAMEEND, FrameEndTimeout_ms);
   if(rtn == ACL_RTN_ERROR){
   ACAPERRORINFO AcapErrInfo;
   AcapGetLastErrorCode(&AcapErrInfo, 0);
   if(AcapErrInfo.dwBoardErrorCode==ACL_ABORT_EVENT_WAIT){
      // Function is returned as error if it is aborted
      printf("AcapWaitEvent was aborted.¥n");
   }
}
```

■ Cancellation

You need to cancel setting if setting event is unnecessary to stand by.
Perform AcapSetEvent, which 4th argument “bEventEnable” is set as FALSE, for cancel setting.

■ The setting example for software

The sample code to cancel event setting is as follows.

<Sample code>

```
// Cancel event setting
rtn = AcapSetEvent(hDevice, channel, ACL_INT_FRAMEEND, FALSE);
```

When AcapWaitEvent is under performing (under standby state), it becomes error since cannot cancel that.
In this case, cancels after execute “cancel standby”.

Downloaded from www.Manualslib.com manuals search engine

6.13.2 Callback function usage

This section describes standby method of interrupt by callback function.
Registration function has 2 kinds; AcapRegistCallback and AcapRegistCallbackEx.



You can use either AcapRegistCallback or AcapRegistCallbackEx.
You cannot use both at same time.

■ Prototype of Callback Function

The callback function registered into AcapRegistCallback must have following prototype.

```
void CALLBACK CallbackFunc (int nChannel, DWORD dwEvent, int nFrameNo, int nMemoryNo);
nChannel   : The channel which detected interruption
dwEvent    : Detected interrupt(ACL_INT_FRAMEEND and more)
nFrameNo   : Current “frame number finished input”(Last result during stop capturing )
nMemoryNo : Current “memory number(1-) which finished input lastly”(Last result at stop capturing)
```

The callback function registered into AcapRegistCallbackEx must have following prototype.

```
void CALLBACK CallbackExFunc (int nChannel, DWORD dwEvent, int nFrameNo, int nMemoryNo, void* pParam);
nChannel   : The channel which detected interruption
dwEvent    : Detected interrupt(ACL_INT_FRAMEEND and more)
nFrameNo   : Current “frame number finished input”(Last result during stop capturing)
nMemoryNo : Current “memory number gets from last input(1-)”(Last result during stop capturing)
pParam     : Address which is given to 5th argument by AcapRegistCallbackEx
             *Note:pParam becomes pointer of user buffer. It becomes cause of violation of segmentation if memory
                   which is given at registration is released.
```


Downloaded from www.Manualslib.com manuals search engine

■ Register callback function

It registers callback function for notice of interrupt by AcapRegistCallback/AcapRegistCallbackEx. Only 1 callback function can be registered, but it can specify multiple interrupts.

■ The setting example for software

The sample code to register callback function that occurs at "1 frame input completion interrupt" for ch1 is as follows.

<Sample code>

```
void CALLBACK CallbackFunc (int nChannel, DWORD dwEvent, int nFrameNo, int nMemoryNo)
{
    printf("CallbackFunc called. nChannel[%d] dwEvent[%d] nFrameNo[%d] nMemoryNo[%d]¥n",
                nChannel, dwEvent, nFrameNo, nMemoryNo);
    //Process at 1 frame input completion
}

// Register callback function at main routine
int rtn;
int channel = 1;
rtn = AcapRegistCallback(hDevice, channel, ACL_INT_FRAMEEND, (EVENT_FUNC)&CallbackFunc);

// Start capturing (Grab)
rtn = AcapGrabStart(hDevice, channel, 0);

// CallbackFunc is called as inputting 1 frame

<< Output result>>   * If number of transfer destination memory = 3
CallbackFunc called. nChannel[1] dwEvent[2] nFrameNo[1] nMemoryNo[1]
CallbackFunc called. nChannel[1] dwEvent[2] nFrameNo[2] nMemoryNo[2]
CallbackFunc called. nChannel[1] dwEvent[2] nFrameNo[3] nMemoryNo[3]
CallbackFunc called. nChannel[1] dwEvent[2] nFrameNo[4] nMemoryNo[1]
CallbackFunc called. nChannel[1] dwEvent[2] nFrameNo[5] nMemoryNo[2]
```


Downloaded from www.Manualslib.com manuals search engine

■ The setting example for software

The sample code to register extended callback function that occurs at "input start interrupt" or "1 frame input completion interrupt" or "input end interrupt" for ch2 is as follows.

<Sample code>

```
void CALLBACK CallbackFuncEx (int nChannel, DWORD dwEvent, int nFrameNo, int nMemoryNo, void*
pParam)
{
    DWORD Param = *((DWORD*)pParam);
    printf("CallbackFuncEx called. nChannel[%d] dwEvent[%d] nFrameNo[%d] nMemoryNo[%d]
*pParam[%d]¥n", nChannel, dwEvent, nFrameNo, nMemoryNo, Param);
}

// Register extended callback function at main routine
int rtn;
int channel = 2;
// Plural specifications by "OR"
DWORD dwEvent = ACL_INT_GRABSTART | ACL_INT_FRAMEEND | ACL_INT_GRABEND;
DWORD Param = 123456;   // Parameter that gives at 5th argument
rtn = AcapRegistCallback(hDevice, channel, dwEvent, (EVENT_FUNC_EX)&CallbackFuncEx, (void*)&Param);

// Input 5 frame
rtn = AcapGrabStart(hDevice, channel, 5);

// Use ACL_INT_GRABSTART at input start of each frame
// Use ACL_INT_FRAMEEND at input completion of each frame
// Callback function of ACL_GRABEND is called when 5 frame input is completed

<< Output result>>   * If number of transfer destination memory = 3
CallbackFuncEx called. nChannel[2] dwEvent[1] nFrameNo[0] nMemoryNo[3] *pParam[123456]    *Note
CallbackFuncEx called. nChannel[2] dwEvent[2] nFrameNo[1] nMemoryNo[1] *pParam[123456]
CallbackFuncEx called. nChannel[2] dwEvent[1] nFrameNo[1] nMemoryNo[1] *pParam[123456]
CallbackFuncEx called. nChannel[2] dwEvent[2] nFrameNo[2] nMemoryNo[2] *pParam[123456]
CallbackFuncEx called. nChannel[2] dwEvent[1] nFrameNo[2] nMemoryNo[2] *pParam[123456]
CallbackFuncEx called. nChannel[2] dwEvent[2] nFrameNo[3] nMemoryNo[3] *pParam[123456]
CallbackFuncEx called. nChannel[2] dwEvent[1] nFrameNo[3] nMemoryNo[3] *pParam[123456]
CallbackFuncEx called. nChannel[2] dwEvent[2] nFrameNo[4] nMemoryNo[1] *pParam[123456]
CallbackFuncEx called. nChannel[2] dwEvent[1] nFrameNo[4] nMemoryNo[1] *pParam[123456]
CallbackFuncEx called. nChannel[2] dwEvent[2] nFrameNo[5] nMemoryNo[2] *pParam[123456]
CallbackFuncEx called. nChannel[2] dwEvent[4] nFrameNo[5] nMemoryNo[2] *pParam[123456]

*Note : Since first GRABSTART interrupt does not input any frame, "0" is got for nFrameNo and number
of register memory (3) is got for nMemoryNo.
```


Downloaded from www.Manualslib.com manuals search engine

■ Registration cancellation of callback function

Registered callback function is able to be canceled if performs 4th argument of AcapRegistCallback/AcapRegistCallbackEx as “NULL”.

■ The setting example for software

The sample code to re-register callback function by different interrupt is as follows.

<Sample code>

```
// Register “GPIN interrupt“
int rtn;
rtn = AcapRegistCallback(hDevice, channel, ACL_INT_GPIN, (EVENT_FUNC)&CallbackFunc);

// Cancel “GPIN interrupt”
rtn = AcapRegistCallback(hDevice, channel, ACL_INT_GPIN, NULL);

// Register 1 frame input completion
rtn = AcapRegistCallback(hDevice, channel, ACL_INT_FRAMEEND, (EVENT_FUNC)&CallbackFunc);
```


Downloaded from www.Manualslib.com manuals search engine

6.14 Notes for interrupt event, callback function

■ About delayed event and lost an event

The thread in Acap library performs notifying interrupt event and calling callback function. So if delay occurs in performance of thread in state where CPU load are high, calling registered callback function and response of function AcapWaitEvent might be delayed. Notification would disappear by degree of load.

[Concept of "Notice"]

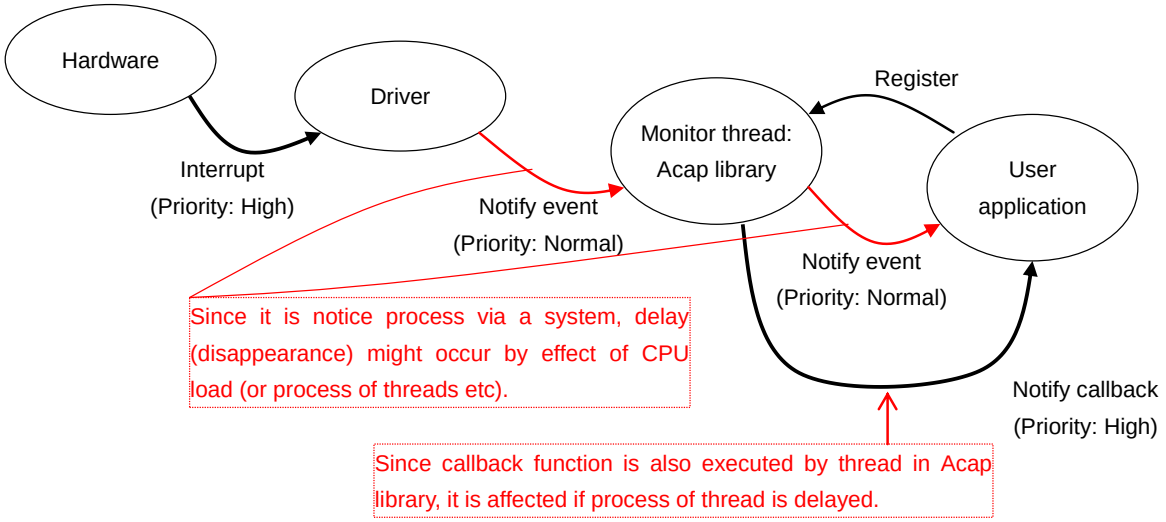


Figure6-33: Concept of "Notice"

[Troubleshoot issues of delayed event]

For example, when "the one-frame input completion interrupt" is standing by in state of heavy load, the followings might occur.

Example: Delay of callback function occurs by inputting 5 frames

```
// Callback function that is preformed every 1 frame input
void CALLBACK CallbackFunc (int nChannel, DWORD dwEvent, int nFrameNo, int nMemoryNo)
{
    printf("CallbackFunc called. nChannel[%d] dwEvent[%d] nFrameNo[%d] nMemoryNo[%d]¥n",
                nChannel, dwEvent, nFrameNo, nMemoryNo);
}

<< Output result>>   * If number of transfer destination memory = 3
CallbackFunc called. nChannel[1] dwEvent[2] nFrameNo[1] nMemoryNo[1]
CallbackFunc called. nChannel[1] dwEvent[2] nFrameNo[2] nMemoryNo[2]
CallbackFunc called. nChannel[1] dwEvent[2] nFrameNo[3] nMemoryNo[3]
CallbackFunc called. nChannel[1] dwEvent[2] nFrameNo[5] nMemoryNo[2]    *Delay occurs
CallbackFunc called. nChannel[1] dwEvent[2] nFrameNo[5] nMemoryNo[2]
```


Downloaded from www.Manualslib.com manuals search engine

Since “frame number which input completed (nFrameNo)”, the argument for callback function, counts up in real time at DMA transfer, it skips “frame number which input completed” if call of callback is delayed and executed after count-up.
Since DMA transfer itself is done without skipping, it can judge that number of images which frame counter skipped was decided. (It means that finish transferring 2 frames if it skips to 3->5)
So if image process is carried out to memory of the increment of frame counter, it can restore delay.

```
CallbackFunc called. nChannel[1] dwEvent[2] nFrameNo[1] nMemoryNo[1]
CallbackFunc called. nChannel[1] dwEvent[2] nFrameNo[2] nMemoryNo[2]
CallbackFunc called. nChannel[1] dwEvent[2] nFrameNo[3] nMemoryNo[3]
CallbackFunc called. nChannel[1] dwEvent[2] nFrameNo[5] nMemoryNo[2]   *1
CallbackFunc called. nChannel[1] dwEvent[2] nFrameNo[5] nMemoryNo[2]   *2
*1 : Perform process for 2 frames (MemoryNo 1 or 2)
*2 : Do not perform anything since “frame number finished input” does not increase
```

[Troubleshoot issues of lost event]
If status which load is very high continues, next event notification, which is same, is preformed and disappearance is occurred.

```
CallbackFunc called. nChannel[1] dwEvent[2] nFrameNo[1] nMemoryNo[1]
CallbackFunc called. nChannel[1] dwEvent[2] nFrameNo[2] nMemoryNo[2]
CallbackFunc called. nChannel[1] dwEvent[2] nFrameNo[3] nMemoryNo[3]
CallbackFunc called. nChannel[1] dwEvent[2] nFrameNo[5] nMemoryNo[2]   * Frame 4 does not come
```

Since disappearance is caused by overwriting same event notification, skipped event is always notified. It is able to follow if you do same measures as time of delay.

As these shows, it is able to restore delay by followings:
* Preparing multiple number of transfer destination memory
* Process the number of skipped frames

However, if skip exceeding number of transfer destination memory is occurred, overwrite of transfer destination memory occurs. Also it is hard to restore delay if image processing is heavy load or needs long time for performing.
In that situation, you need to do disposals like followings:
* Reduce CPU load
* Revise PC engine performance
* Reduce frame rate
* Reduce transferring amount (resolution) of 1 frame

Downloaded from www.Manualslib.com manuals search engine

Chapter 7 Error codes

These are error code values that can be gotten by AcapGetLastErrorCode.
- Common error code (dwCommonErrorCode): The error code about AcapLib2 library function
- Board error code (dwBoardErrorCode) : The error code peculiar to frame-grabber (APX-3302)
- Extended error code (dwExtendErrorCode) : The error code of the driver factor and others

7.1 Common error codes

These codes are stored in "dwCommonErrorCode" of ACAPERRORINFO structure.

Define value	Hexadecimal	Denotation
ACL_NO_ERROR	0	No error
ACL_BOARD_NOT_FOUND_ERROR	0xF101	Cannot find board
ACL_BOARD_NAME_ERROR	0xF102	Incorrect board name
ACL_BOARD_NUMBER_ERROR	0xF103	Specified board number is not existed
ACL_SAME_BOARDNO_ERROR	0xF104	Multiple boards have same board numbers
ACL_INVALID_HANDLE_ERROR	0xF105	Invalid handle
ACL_SELECT_CHANNEL_ERROR	0xF106	Invalid channel
ACL_MEM_ALLOCATE_ERROR	0xF107	Failed to reserve memory
ACL_INVALID_MUTEX_ERROR	0xF108	Invalid handle for exclusive control
ACL_MUTEX_LOCKED_ERROR	0xF109	Function is in the exclusive state
ACL_FILE_NOT_FOUND_ERROR	0xF10A	Cannot find a specified file
ACL_EVENT_TIMEOUT_ERROR	0xF10B	Times out for waiting event
ACL_OPEN_ERROR	0xF10C	Failed to open device
ACL_CLOSE_ERROR	0xF10D	Failed to close device
ACL_INITIALIZE_ERROR	0xF10E	Failed to initialize the board
ACL_REFLECT_PARAM_ERROR	0xF10F	Failed to reflect the setting value
ACL_SNAP_ERROR	0xF110	Failed for an image input
ACL_SNAP_WAIT_ERROR	0xF111	Failed for waiting an image input
ACL_GRAB_START_ERROR	0xF112	Failed for streaming
ACL_GRAB_STOP_ERROR	0xF113	Failed to abort streaming
ACL_GRAB_ABORT_ERROR	0xF114	Failed to abort process
ACL_GET_FRAMENO_ERROR	0xF115	Failed to get the total number of frame
ACL_SET_EVENT_ERROR	0xF116	Failed to register an event
ACL_WAIT_EVENT_ERROR	0xF117	Failed to hold an event

Downloaded from www.Manualslib.com manuals search engine

ACL_REGIST_CALLBACK_ERROR	0xF118	Failed to register a callback function
ACL_SET_ENCODER_ERROR	0xF119	Failed to set up an encoder
ACL_GET_ENCODER_ERROR	0xF11A	Failed to get an encoder
ACL_SET_SHUTTER_TRIGGER_ERROR	0xF11B	Failed to set up the trigger
ACL_SET_LINE_TRIGGER_ERROR	0xF11C	Failed to set up the line trigger
ACL_SET_EXTERNAL_TRIGGER_ERROR	0xF11D	Failed to set up the external trigger
ACL_ABORT_EXTERNAL_WAIT	0xF11E	Abort for waiting external trigger
ACL_SET_STROBE_ERROR	0xF11F	Failed to set up the strobe
ACL_SET_GPIO_ERROR	0xF120	Failed to set up the GPIO
ACL_IMAGE_CONVERT_ERROR	0xF121	Failed for picture bit conversion
ACL_SET_CAMERALINK_ERROR	0xF122	Failed to set up camera link
ACL_SET_DMA_ERROR	0xF123	Failed to set up DMA
ACL_SERIAL_OPEN_ERROR	0xF124	Failed for serial opening
ACL_SERIAL_CLOSE_ERROR	0xF125	Failed for serial closing
ACL_SERIAL_WRITE_ERROR	0xF126	Failed for serial sending (writing)
ACL_SERIAL_READ_ERROR	0xF127	Failed for serial receiving (reading)
ACL_SERIAL_SET_PARAM_ERROR	0xF128	Failed to set up a serial parameter
ACL_SERIAL_GET_PARAM_ERROR	0xF129	Failed to get the serial parameter
ACL_SERIAL_HANDLE_ERROR	0xF12A	Invalid serial handle
ACL_SERIAL_SELECT_PORT_ERROR	0xF12B	Specified port is unable to use
ACL_SET_BUFFER_ADRS_ERROR	0xF12C	Failed to set up the buffer address
ACL_GET_BUFFER_ADRS_ERROR	0xF12D	Failed to get the buffer address
ACL_SET_CAMERALINK_EX_ERROR	0xF12E	Failed to set up camera link (expand)
ACL_ABORT_EVENT_WAIT	0xF12F	Abort waiting event status
ACL_GET_BOARDINFO_ERROR	0xF130	Failed to get board information
ACL_SET_INFO_ERROR	0xF131	Failed to set up the value
ACL_GET_INFO_ERROR	0xF132	Failed to get the value
ACL_SELECT_FILE_ERROR	0xF133	Failed to read/write initial file
ACL_GET_FILE_VERSION_ERROR	0xF134	Failed to get the file version
ACL_REGIST_CALLBACK_EX_ERROR	0xF135	Failed to register callback function (expand)
ACL_ROI_CONVERT_ERRPR	0xF136	Failed for ROI image conversion
ACL_GET_CAMERALINK_ERROR	0xF137	Failed to get the camera link setting
ACL_GET_DMA_ERROR	0xF138	Failed to get the DMA setting
ACL_GET_LINE_TRIGGER_ERROR	0xF139	Failed to get the line trigger setting
ACL_GET_EXTERNAL_TRIGGER2_ERROR	0xF13A	Failed to get the external trigger setting
ACL_SET_DMA_OPTION_EX_ERROR	0xF13B	Failed to set DMA option EX
ACL_GET_DMA_OPTION_EX_ERROR	0xF13C	Failed to get DMA option EX setting

Downloaded from www.Manualslib.com manuals search engine

ACL_GET_GPOUT_ERROR	0xF13D	Failed to get GPOUT status
ACL_PARAM_1_ERROR	0xF201	Parameter error (1st argument)
ACL_PARAM_2_ERROR	0xF202	Parameter error (2nd argument)
ACL_PARAM_3_ERROR	0xF203	Parameter error (3rd argument)
ACL_PARAM_4_ERROR	0xF204	Parameter error (4th argument)
ACL_PARAM_5_ERROR	0xF205	Parameter error (5th argument)
ACL_PARAM_6_ERROR	0xF206	Parameter error (6th argument)
ACL_PARAM_7_ERROR	0xF207	Parameter error (7th argument)
ACL_PARAM_8_ERROR	0xF208	Parameter error (8th argument)
ACL_PARAM_9_ERROR	0xF209	Parameter error (9th argument)
ACL_PARAM_10_ERROR	0xF20A	Parameter error (10th argument)
ACL_PARAM_11_ERROR	0xF20B	Parameter error (11th argument)

Table7-1: Common error code

Downloaded from www.Manualslib.com manuals search engine

7.2 Board error code

These codes are stored in "dwBoardErrorCode" of ACAPERRORINFO structure.

Define value	Hexadecimal	Denotation
ACL_3302_NO_ERROR	0	No error
ACL_3302_DRIVER_ERROR	0xF901	Driver error
ACL_3302_INVALID_PARAM	0xF902	Invalid parameter
ACL_3302_NO_DEVICE	0xF903	Cannot find a device
ACL_3302_EVENT_ERROR	0xF904	Error for registering event
ACL_3302_INVALID_HANDLE	0xF905	Invalid handle
ACL_3302_INVALID_CAPTURE	0xF906	Invalid capture
ACL_3302_CAPTURE_RUNNING	0xF907	Under capturing
ACL_3302_TIME_OUT	0xF908	Time out
ACL_3302_INSUFFICIENT_CONFIG	0xF909	Insufficient setup
ACL_3302_NOT_CONNECTED_CAMERA	0xF90A	The camera is not connected
ACL_3302_NULL_POINTER_ERROR	0xF90B	The specified pointer is NULL
ACL_3302_X_SIZE_ERROR	0xF90C	Invalid X size
ACL_3302_Y_SIZE_ERROR	0xF90D	Invalid Y size
ACL_3302_BITSHIFT_ERROR	0xF90E	Invalid bit shift
ACL_3302_TAPNUM_ERROR	0xF90F	Invalid tap numbers
ACL_3302_SET_BUFFER_ERROR	0xF910	Failed to register the buffer
ACL_3302_CAMERA_KIND_ERROR	0xF911	Wrong camera classification (area or line)
ACL_3302_GRAB_ABORT	0xF912	Abort grabbing
ACL_3302_CH_NOT_OPENED	0xF913	Channel is not open
ACL_3302_FIFO_ERROR	0xF914	FIFO error
ACL_3302_DMA_ERROR	0xF915	DMA error
ACL_3302_GET_REGISTRY_KEY_ERROR	0xF916	Failed to get registry key
ACL_3302_DLL_DRIVER_ERROR	0xF917	DLL error
ACL_3302_INTERNAL_LIBRARY_ERROR	0xF918	Library error
ACL_3302_UNKNOWN_ERROR	0xF919	Unknown error
ACL_3302_INVALID_CAMBIT	0xF920	Invalid camera input bit
ACL_3302_OPEN_ALREADY	0xF921	Other processes are still open
ACL_3302_INVALID_EVENT	0xF924	Invalid event
ACL_3302_SETEVENT_ALREADY	0xF925	The event is already registered
ACL_3302_CREATEEVENT_FAILED	0xF926	Failed for "CreateEvent" in the library
ACL_3302_WAITEVENT_ALREADY	0xF927	Already in standby state, or process of ABORT
ACL_3302_EVENT_NOT_IDLE	0xF928	Cannot cancel since it is in standby state
ACL_3302_CALLBACK_ALREADY	0xF929	Call-back is already registered
ACL_3302_INVALID_CC	0xF92A	Invalid CC cycle or width relationship

Downloaded from www.Manualslib.com manuals search engine

ACL_3302_INVALID_BUFFER	0xF92B	Invalid buffer
ACL_3302_INVALID_XDELAY	0xF92C	Invalid X Delay
ACL_3302_INVALID_YDELAY	0xF92D	Invalid Y Delay
ACL_3302_INVALID_POWER_STATE	0xF92E	Invalid camera power state
ACL_3302_POWER_STATE_ERROR	0xF92F	Camera power state error
ACL_3302_POWER_ON_TIMEOUT	0xF930	Cannot detect camera clock in designated time although camera power was turned on
ACL_3302_POWER_OFF_TIMEOUT	0xF931	Detected camera clock after designated time pass although camera power was turned off
ACL_3302_TAP_SEL_ERROR	0xF932	Invalid Tap or Bit or Arrange relationship
ACL_3302_ARRANGE_X_ERROR	0xF933	Invalid arrangeXSize
ACL_3302_NOT_SUPPORT	0xF9FF	Unsupported

Table7-2: Board error code

Downloaded from www.Manualslib.com manuals search engine

7.3 Extended error code (Windows)

These codes are stored in "dwExtendErrorCode" of ACAPERRORINFO structure.
The meaning of inputting value is different by situation that error happens.

Occurring Status	Details	Meaning of extended error code
Internal Library Error	If this board error cord appears: ACL_3302_INTERNAL_LIBRARY_ERROR	Extended error code is shown as DWORD value. Mainly, error code that each driver outputs is stored. Refer to "Extended error codes returned by driver" for more information.
Buffer-related Error	If this board error cord appears: ACL_3302_SET_BUFFER_ERROR	Extended error code is shown as negatives of int type. Mainly, error code that shows setting error at buffer registration is stored. Refer to "Extended error codes related to buffer" for more information.
Argument-related Error	If this board error cord appears: ACL_3302_INVALID_PARAM	Extended error code is shown as DWORD value. It stores invalid argument's number if value specified by argument of function is invalid. ACL_PARAM_n_ERROR (n=Invalid argument's number)
Other Errors	If there is no classification, but need extended information.	The value, which is written at "extend" in description of each function, is returned. Contact us if other extended error code occurs and need details for that.

Table7-3: Occurrence situation of Extended error code

Downloaded from www.Manualslib.com manuals search engine

7.3.1 Extended error codes returned by driver

If error is stored at APX-3302, camera channel is stored at bottom.
Since it becomes as "0 : CAM1 / 1 : CAM2", if **"0x001E0001"** is stored,
→ Value of XSize at CAM2 is invalid
is shown.

It would be need physical change (change PC, add more memory, and more) for solve error, which is written as "fatal error". If this error occurs, execute application again after reboot. Contact us if you try it and still same error occurs.

Code	Error Cause / Point to Be Checked	
0x000B0000	Cause	Mapping is already executed by this file handle.
	Check	It is fatal error.
0x000C0000	Cause	Kernel does not make MDL of PCI space.
	Check	It is fatal error.
0x000D0000	Cause	Failed for MmMapLockedPagesSpecifyCache.
	Check	Value of loaded memory is not enough.
0x000E0000	Cause	It was going to re-register event that is already registered.
	Check	Avoid double registration for same event.
0x000F0000	Cause	Kernel memory is already mapped.
	Check	It is fatal error.
0x00100000	Cause	Cannot do mapping since MDL does not exist.
	Check	It is fatal error.
0x00110000	Cause	Failed for MmMapLockedPagesSpecifyCache.
	Check	Value of loaded memory is not enough.
0x00120000	Cause	MmMapLockedPagesSpecifyCache returns "NULL".
	Check	It is fatal error.
0x00130000	Cause	PCI unmap error.
	Check	It is fatal error.
0x001E0000	Cause	Value of X size is invalid.
	Check	Need to set this as 8-byte alignment (multiple by 8-byte).
0x001F0000	Cause	Value of Y size is invalid.
	Check	Do not set "0".
0x00200000	Cause	Inputting size is much bigger than registry registered value.
	Check	Change registry registered value.
0x00210000	Cause	Value of NumOfBuffer is invalid.
	Check	Do not set "0".
0x00220000	Cause	DMA status is not in Idle status.
	Check	Do not execute initialize during capturing.
0x00250000	Cause	Cannot begin capturing since selected CH is under inputting.
	Check	Do not take in again during taking in.
0x00270000	Cause	Cannot find DMA descriptor.
	Check	Capturing is executed without registering buffer.
0x00290000	Cause	Value of X size is invalid.

Downloaded from www.Manualslib.com manuals search engine

	Check	Do not set “0”.
0x002A0000	Cause	X size is not 8-byte alignment.
	Check	Need to set this as 8-byte alignment (multiple by 8-byte).
0x002B0000	Cause	Value of Y size is invalid.
	Check	Do not set “0”.
0x002D0000	Cause	Capturing size is much bigger than maximum of DIRECT_IO.
	Check	It was going to register 16 Mbytes or more buffers at once.
0x00300000	Cause	DMA status of AAE-B04 is not Idle.
	Check	Do not register buffer during capturing.
0x00310000	Cause	Other file object is under registering buffer.
	Check	Buffer will be registered by multiple processes.
0x00320000	Cause	Failed to MmMapLockedPagesSpecifyCache.
	Check	Value of loaded memory is not enough.
0x00360000	Cause	Failed for physical address calculation of user buffer during creating DMA descriptor.
	Check	Value of loaded memory is not enough.
0x00370000	Cause	Not enough memory for DMA descriptor.
	Check	Reduce buffer that try to register.
0x00380000	Cause	Buffer that is going to register is not 8-byte alignment.
	Check	Adjust start address of registering buffer to 8-byte alignment.
0x00390000	Cause	AAE-B04 is not in DMA idle state.
	Check	Do not cancel buffer during capturing.
0x00410000	Cause	Buffer is not registered.
	Check	Capturing is executed without registering buffer.
0x00430000	Cause	The other process is already registered in that channel.
	Check	Do not register same even in multiple processes.
0x00440000	Cause	Request is not mounted.
	Check	Do not register invalid event.
0x00450000	Cause	Failed to stop DMA.
	Check	It is fatal error.
0x00460000	Cause	There is no camera clock.
	Check	Check camera power supply, wiring, etc.
0x00470000	Cause	Returned value of MmMapLockedPages is “NULL”.
	Check	Value of loaded memory is not enough.
0x00480000	Cause	The size of continuing physical memory is too small.
	Check	Value of loaded memory is not enough.
0x004A0000	Cause	Cannot execute during capturing.
	Check	Do not register event during capturing.
0x004B0000	Cause	Cannot execute during capturing.
	Check	Do not initialize during capturing.
0x004C0000	Cause	Cannot execute during capturing.
	Check	Do not register buffer during capturing.
0x004D0000	Cause	Cannot execute during capturing.
	Check	Do not cancel buffer under capturing.

Table7-4: Extended error codes returned by driver

Downloaded from www.Manualslib.com manuals search engine

7.3.2 Extended error codes related to buffer

These are error codes that are stored when problem occurs at buffer registration process.

It is mainly occurred when function like AcapSelectFile or AcapReflectParam is executed or setting is reflected (initialized) by function like AcapSetBufferAddress or AcapSetInfo.

Definition value	Value	Content
ACL_BUFFINFO_ERR_NUMOFMEM	-1	Disagreement occurs in buffer number that is going to set up, and a number of buffers that is allocated by the AcapLib2. Example: Registers 5th buffer even ACL_MEM_NUM is"4".
ACL_BUFFINFO_ERR_ALLOCATE	-2	Fails to secure memory of Acap library buffer. It occurs when 1 frame size (X size x Y size) or number of memories are too big.
ACL_BUFFINFO_ERR_EXIST	-3	It occurs when buffer number, which is going to register, is already registered. Also when you perform "Buffer divided DMA", this error occurs if there is blank in Region rectangle.
ACL_BUFFINFO_ERR_XSIZE	-4	It is generated when X size of registering buffer is invalid.
ACL_BUFFINFO_ERR_YSIZE	-5	It is generated when Y size of registering buffer is invalid.
ACL_BUFFINFO_ERR_BITWIDTH	-6	It is generated when Bit width of registering buffer is invalid.
ACL_BUFFINFO_ERR_PTRNULL	-7	It is generated when pointer of registering buffer is "NULL".
ACL_BUFFINFO_ERR_NO_BUFFER	-8	It occurs when specified buffer does not exist. Also, it is occurred when buffer for storing pointer does not exist at getting buffer address.
ACL_BUFFINFO_ERR_REGION_SIZE	-9	It occurs when rectangle size of Region structure specified by "Non-linear DMA (ROI Transfer)"or "Buffer divided DMA" is invalid.
ACL_BUFFINFO_ERR_REGION_PTRNULL	-10	It occurs when buffer pointer, which is specified by "Non-linear DMA (ROI Transfer)"or "Buffer divided DMA", is NULL.
ACL_BUFFINFO_ERR_REGION_NUM	-11	It occurs when number of Region structure in "Buffer divided DMA" is 2 or less.
ACL_BUFFINFO_ERR_INITDMA	-12	Buffer, which cannot be initialized by driver, is registered. The following causes are thought for this problem. Cause 1 : Alignment limit of buffer start address is not followed Cause 2 : It is lower than alignment limit of X size Cause 3 : Multiple buffer's size are not same Cause 4 : Size information and actual buffer size are not same
ACL_BUFFINFO_ERR_SETBUFFER	-13	
ACL_BUFFINFO_ERR_SIZEOF	-14	The SizeOf, which is member of structure that specified in 4th argument of AcapSetDmaOptionEx, is invalid.

Table7-5: Extended error codes related to buffer

Downloaded from www.Manualslib.com manuals search engine

Firmware Update History

Version	Revised Date	Revised Contents
Ver.A(0x10)	2012.4.30	First edition

Downloaded from www.Manualslib.com manuals search engine

Revision History

Version	Revised Date	Revised Contents
A	2012.7.31	First edition

Downloaded from www.Manualslib.com manuals search engine

Inquiry

APX-3302 user's manual
July 31, 2012 first edition

Edited and published by AVAL DATA CORPORATION

©2012 AVALDATA CORPORATION

OM110000019A

株式会社 アバールデータ
Head Quarters Machida office Sales department
1-25-10, Asahimachi, Machida-shi, Tokyo, 194-0023 TEL 042-732-1030 FAX 042-732-1032
URL :http://www.avaldata.co.jp/

AVAL DATA CORPORATION

Downloaded from www.Manualslib.com manuals search engine

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP