



APX-3326 User's Manual

본 매뉴얼은 아발데이터의 대리점에서 고객 참고용으로 번역한 것으로,
원본 매뉴얼과는 내용이 상이할 수 있습니다.

 **AVAL DATA CORPORATION**

주의]

1. 본 유저 매뉴얼의 내용 일부 또는 전체를 무단으로 복사하는 것은 금지 되어 있습니다.
2. 본 유저 매뉴얼의 내용에 관해서는 사전 예고 없이 변경 될 수 있습니다.
3. 본 유저 매뉴얼의 내용에 대해 문의 사항이 있으시면 구매하신 대리점으로 연락 부탁드립니다.
4. 잘 못된 사용으로 인해 발생한 문제에 대해서는 당사 워런티 조항과 관계없이 제조사에서 책임을 지지 않습니다.

Copyright 2013 AVALDATA CORPORATION

본 유저 매뉴얼에서 사용되는 회사명 및 제품명은 당사의 상표 또는 등록 상표입니다.

사용자 매뉴얼을 읽기 전에

이 매뉴얼에서는 제품을 안전하게 사용하고, 고객의 위험, 재산피해 등을 사전에 방지하기 위해 아래와 같은 안전표시를 사용하고 있습니다. 안전표시의 의미를 잘 숙지한 후에 매뉴얼을 이용 바랍니다.



경고 :

이 표시를 무시하고 제품을 잘못 취급할 경우, 사망 또는 중상을 입을 가능성이 있음.



주의 :

이 표시를 무시하고 제품을 잘 못 취급할 경우 상해를 입을 가능성이 있으며, 물적 피해가 발생할 가능성이 있음.



잘못 사용할 경우 감전의 가능성이 있음.



잘못 사용할 경우 상처를 입을 가능성이 있음.



잘못 사용할 경우, 고온에 의한 화상을 입을 가능성이 있음.



안전을 위해 전원 케이블의 플러그를 반드시 뽑 것.



잘못 사용할 경우 발연, 발화의 가능성이 있음.

 **주의 :**

- ◆ 본 제품을 수분 및 습기가 많은 장소에서의 사용은 감전의 원인이 됩니다.
- ◆ 장치 전원코드의 플러그를 젖은 손으로 취급 하는 것은 감전의 원인이 됩니다.
- ◆ 전원 코드가 AC 콘센트에 접속 되어 있을 경우에, 젖은 손으로 본체에 접촉하는 것은 감전의 원인이 됩니다.
- ◆ 주변기기를 장착/분해 할 경우에는 제품의 전원 코드를 제거 하여 주세요. 전원이 들어 있는 상태에서의 장착/분해를 하는 것은 감전의 원인이 됩니다.
- ◆ 장치 전원코드의 접속은 접지단자가 있는 독립된 AC 콘센트에 접속해 주세요. 접지의 미 접속 또는 불안전한 접지는 감전 및 오작동의 원인이 되며, 하나의 콘센트에 다수의 전원을 동시에 사용하는 것은 콘센트의 과열로 화재의 원인이 됩니다.
- ◆ 고장 및 이상이 발생한 경우에는 즉시 사용을 중지하고, 구매처로 문의 바랍니다.
- ◆ 고객센터에서의 수리 및 개조는 금지하고 있습니다.

목 차

第 1 章	사용상의 주의사항	1
1.1	본 제품 취급에 대해	2
1.1.1	제품의 사용 및 보관 장소	2
1.1.2	제품 사용시의 주의사항	3
1.2	애프터 서비스	4
1.2.1	유저등록	4
1.2.2	문의처	5
1.2.3	메뉴얼	5
1.2.4	제품등록	6
1.2.5	고장 및 이상이 발생한 경우	7
1.2.6	문의 전 확인사항	8
1.2.7	수리 및 유지보수 서비스	8
1.2.8	제품을 3자에게 양도할 경우	9
1.3	제품의 폐기 방법	9
第 2 章	개요	10
2.1	제품개요	10
2.2	특징	10
2.3	사양 (Specifications)	12
2.4	블록 다이어그램	14
2.5	프론트 패널 외관도	15
2.6	커넥터핀 배치	16
2.6.1	CameraLink 커넥터	17
2.6.2	외부전원 커넥터 CN17	19
2.6.3	확장 커넥터 CN16	20
第 3 章	초기설정	21
3.1	하드웨어 초기설정	21
3.1.1	SW1 설정	22
3.1.2	SW2 설정	22
3.1.3	CN11 ~ CN14 설정	23
3.1.4	CN15 설정	23
3.1.5	CN7 설정	26

3.1.6 PCI Express 슬롯에 실장	27
3.1.7 케이블 접속	28
3.1.8 확장 패널 케이블 CBL-Z102	29
3.2 소프트웨어 초기설정	30
3.2.1 SDK-AcapLib2 에 대해	30
3.2.2 SDK-AcapLib2 인스톨 순서	31
3.2.3 시리얼 통신 드라이버(가상 COM 드라이버)	36
3.2.4 Registration Tool	36
3.2.5 펌웨어 갱신 툴	38
3.2.6 각 소프트웨어의 언인스톨	39
3.2.7 인스톨 관련 트러블 슈팅	41
第 4 章 AcapLib2Viewer 로 촬영	42
4.1 AcapLib2Viewer 에 대해	42
4.2 ini 에 대해	43
4.3 AcapLib2Viewer 를 사용하여 촬영	44
第 5 章 프로그래밍 가이드	49
5.1 AcapLib2 Flow 차트	49
5.2 프로그래밍 예	50
5.2.1 한개 영상촬영(Snap)	50
5.2.2 연속 촬영(Grab)	52
5.3 취득 샘플에 대해	54
5.3.1 SimpleApplication	54
5.3.2 OtherSample	55
第 6 章 기능	57
6.1 CameraLink Configuration	57
6.1.1 2TAP 출력시의 설정	58
6.1.2 3TAP 출력시의 설정	59
6.1.3 4TAP 출력시의 설정	60
6.1.4 8TAP 출력시의 설정	61
6.2 Power Over CameraLink	63
6.3 카메라 노출제어 신호 CC	65
6.4 시리얼 인터페이스	68
6.5 외부 신호 출력	70
6.6 외부입력 신호의 노이즈제거기능	75
6.7 스트로브 출력	76

6.8	CC 출력과 스트로브 출력 타이밍 조정	77
6.9	엔코더 입력	80
6.9.1	엔코더 입력회로	80
6.9.2	엔코더 카운트 타이밍	81
6.10	엔코더 카운트 모드	82
6.10.1	상대 카운트 모드	82
6.10.2	절대 카운트 · 싱글 포인트 모드	84
6.10.3	절대 카운트 · 멀티 포인트 모드	87
6.11	엔코더 동작 모드	95
6.11.1	엔코더 스캔 모드	95
6.11.2	엔코더 라인선택 모드	98
6.12	MEGA-FIFO	102
6.13	끼워넣기	105
6.13.1	이벤트 사용방법	106
6.13.2	콜백함수의 상용방법	108
6.14	끼워넣기 이벤트 · 콜백 함수의 주의점	112
第 7 章	설정 ID 일람	113
第 8 章	에러 코드	118
8.1	공통 에러코드	118
8.2	보드 에러 코드	122
8.3	확장 에러코드 (Windows)	125
8.3.1	드라이버가 출력하는 확장 에러코드	125
8.3.2	버퍼관련 확장 에러코드	126
	펌웨어 갱신이력	127
	개정이력	128
	お問い合わせ	129

第 1 章 사용상의 주의사항

「APX-3326」의 구입을 감사드립니다. 사용전에 본 매뉴얼을 반드시 읽고 바른 사용법을 숙지바랍니다.

주의 1) 일반적으로 시판되고 있는 마더보드·PC 에 사용되는 그래픽 카드에 지정된 16 레인보드에 당사 제품을 장착하여 사용할 경우, 정상적으로 동작하지 않을 가능성이 있습니다. 이는 그래픽 카드의 성능을 최대한 발휘할 수 있도록 튜닝되어 있는 경우가 있으므로, 이와 같이 사양이 명확하지 않는 경우는 정상적으로 동작하지 않을 수 있습니다

주의 2) PoCXP 가 대응되지 않는 카메라를 사용 할 경우에는 카메라 전원을 PC 전원보다 먼저 입력 해 주세요. 카메라의 파손 및 PC의 동작 불안정의 원인이 될 수 있습니다.

주의 3) 2 대 이상의 PoCL 카메라를 접속할 경우에는 아래 설정을 실행 해 주세요.
· CN15 을 EXT Con 으로 설정
· CN17 에 PCI Express 용의 6 핀 전원 커넥터 접속

주의 4) 전원을 입력하고 CameraLink 케이블을 탈 부착하는 것은 카메라와 보드의 파손의 원인이 됩니다.

주의 5) APX-3326 의 패널 개구부 치수는 PCI Express 규격이 아닙니다. 이 때문에 사용하는 PC 에 따라서는 간섭이 있을 수 있으니 이점 주의 바랍니다.

1.1 본 제품 취급에 대해

1.1.1 제품의 사용 및 보관 장소



■ 제품에 약품등이 묻을 수 있는 곳에서는 사용을 하지 말아 주세요. 제품이 약품이 묻을 경우, 회로의 쇼트로 인해 화재를 일으킬 수 있으며, 고장 및 변형의 원인이 됩니다.

■ 제품에 물과 기름등 액체가 묻을 수 있는 장소, 증기 및 습기가 많은 장소에서는 사용 및 보관을 하지 말아 주세요. 고장의 원인이 됩니다.

■ 평평하고 충분히 강도가 있는 곳에서 사용, 보관해 주세요. 또한 진동 및 충격이 가하지 않도록 주의 바랍니다. 고장 및 변형의 원인이 됩니다.



■ 직사광선이 비치는 장소, 화기 및 스토브등 난방기구의 근처에서는 사용 및 보관하지 말아 주세요. 고장 및 변형의 원인이 됩니다.

■ 먼지가 많은 장소에서의 사용 및 보관은 고장의 원인이 됩니다.



■ TV, 라디오, 무선 전화기등의 근처에서 사용을 하지 말아 주세요. 주변 기기들의 노이즈가 제품에 영향을 미칠 수 있습니다.

■ 자기, 전파가 발생하는 기기의 근처에서는 사용 및 보관은 고장의 원인이 됩니다.

■ 제품의 위에 무거운 것을 올리는 것을 고장의 원인이 됩니다.




■ 제품에 물 또는 액체를 넣는 것은 고장의 원인이 됩니다

■ 정전기는 제품 고장의 원인이 됩니다. 본 제품을 취급할 경우에는 신체의 정전기가 제품에 영향을 주지 않도록 사전에 조치 바랍니다.

■ 제품을 취급시에 모서리 부분등에 의해 손에 상처를 입을 수 있으니 주의 바랍니다.

1.1.2 제품 사용시의 주의사항

 주의

감전의 위험이 있으니, 제품을 장치에 장착, 분해 할 경우에는 반드시 장치 전원 케이블을 제거한 상태에서 장치 전원이 꺼진 것을 확인 후에 작업을 하십시오.

■ 제품을 장착할 때 주의사항.

a) 제품의 스위치가 바르게 입력 되어 있는지 확인 해 주세요.



b) 장치에 장착된 것을 확인 후, 장치 전원을 켜서 동작 하는지 확인 해 주세요.

■ 제품을 분해 할 경우 주의사항.



a) 제품을 분해 하기전에 장치 전원을 반드시 꺼 주세요.

전원이 입력된 상태에서 분해 하는 것은 고장의 원인이 됩니다.

b) 전원을 끈 직후에는, 제품이 상당히 뜨거울수 있으니 취급에 주의 바랍니다.



※ 장치 취급에 대해서는 장치 매뉴얼을 참조해 주세요.

1.2 애프터 서비스

1.2.1 유저등록

당사의 서비스를 이용하기 위해서는 유저 등록이 필요합니다. 유저 등록을 하지 않은 경우에는 유지보수 서비스등을 받을 수 없으므로, 반드시 하기를 참조하여 등록을 부탁드립니다

홈페이지에서 등록 :

하기 당사 서포트 홈페이지의 유지보수 등록 페이지에서 등록 가능합니다.

http://www.avaldata.co.jp/support/regist_user.html

FAX 에 의한 등록 :

제품에 첨부되어 있는 유지보스 서비스 FAX 등록용지의 내용을 기입후 +81-42-732-1032 로 FAX 를 보내주시면 등록 가능합니다.

또한, 개인정보취급에 대해서는 철저히 보호를 하고 있습니다.

당사의 개인정보 취급관련 방침은 하기 웹 사이트를 참조 바랍니다.

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

1.2.2 문의처

본 제품 및 당사 제품에 대한 문의는 아래 연락처로 문의 바랍니다.

문의처

주식회사 아발데이터 (URL <http://www.avaldata.co.jp/>)

본사/마찌다 사무소 영업부

東京都町田市旭町 1 丁目 25 番 10 号

TEL 042-732-1030 (문의 시간 월요일 ~ 금요일 (휴일 제외) / 오전 9:00 ~ 오후 5:00)

FAX 042-732-1032

E-mail sales@avaldata.co.jp

또한, 기술문의는 하기로 연락 바랍니다.

기술 문의

E-mail support@avaldata.co.jp

1.2.3 메뉴얼

당사 제품은 종이 매뉴얼을 제공하지 않습니다. 웹사이트에서 다운로드, 또는 소프트웨어 개발 키 (SDK)을 구매하시면 포함되어 있습니다. 이는 환경보호를 고려하고 변화가 많은 업계 상황을 고려하여 항상 최신의 매뉴얼을 제공하기 위해 시행하고 있으니, 이해를 부탁드립니다

Web 에서 다운로드

※회원등록이 필요합니다.

<http://www.avaldata.co.jp/support/manual.html>

혹, 홈페이지에서 다운로드가 어려울 경우에는 제품 구매처로 연락을 부탁 드립니다

1.2.4 제품등록

본 제품의 제조번호는 본체, 보증서, 박스에 붙어 있는 라벨에 적혀 있습니다.

본 제품의 제품번호는 제품본체, 유지보수 서비스 FAX 등록 용지 및 박스에 적혀 있습니다.

본 제품본체의 시리얼이 적힌 라벨의 부착위치는 그림 1-1 을 참조 바랍니다.

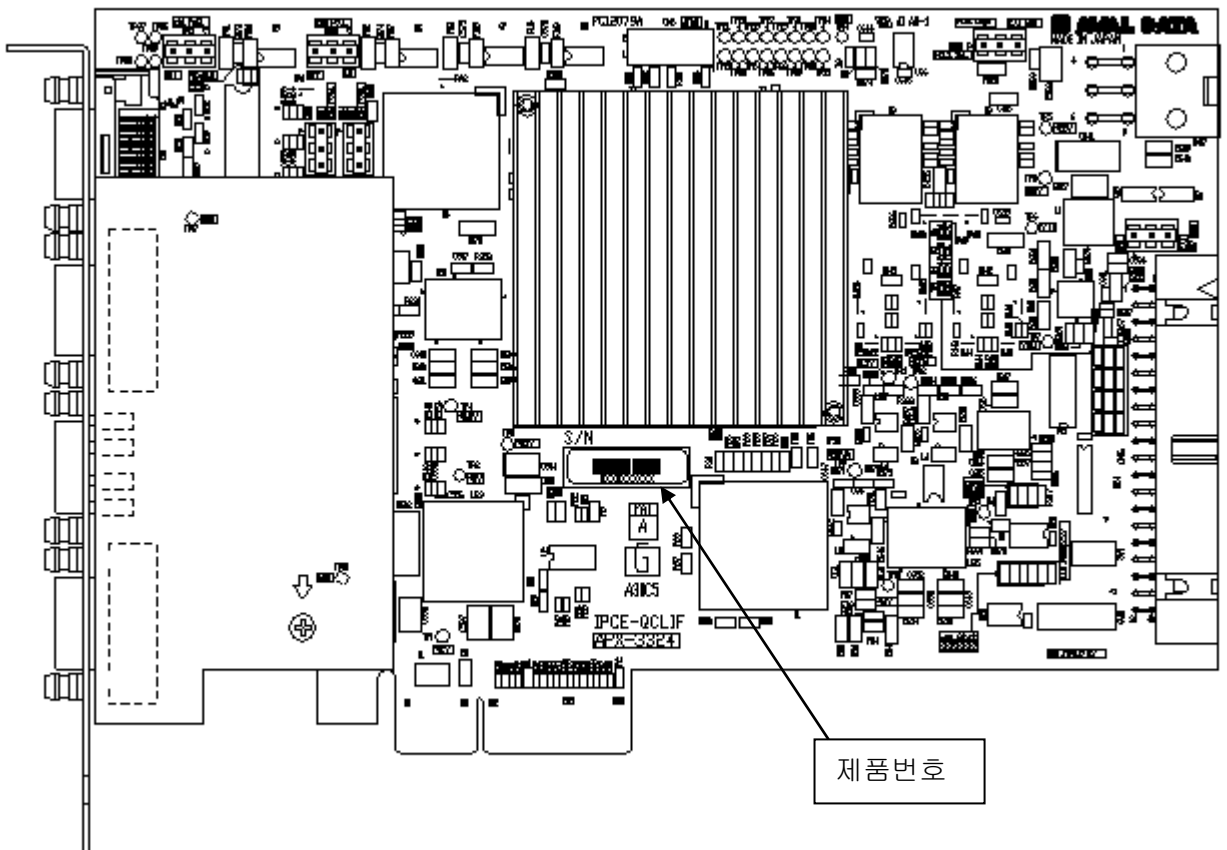


Figure 1-1 : serial 스티커 부착위치

1.2.5 고장 및 이상이 발생한 경우

본 제품을 장착한 장치가 하기와 같은 문제가 발생 하였을 경우는 즉시 장치의 전원을 끄고, 전원 코드를 빼주세요.

A)장치에서 연기가 나거나, 악취가 날 경우.

B)장치가 만질 수 없을 정도로 뜨거운 경우.

C)장치에서 이상한 소리가 날 경우.

D)사용 어플리케이션이 오작동 할 경우.

고장 및 문제를 일으킬 경우, 구입처 또는 당사 서비스센터로 연락 바랍니다.

1.2.6 문의 전 확인사항

본 제품에 대한 고장 및 문제에 대해 문의하기 전에 하기 내용을 확인 바랍니다.

- A) 제품명 및 제품번호를 확인해 주세요.
- B) 본 제품의 보증서를 준비해 주세요.
- C) 고장 및 문제의 상황을 간단히 메모해 주세요.
- D) 본 제품을 사용하는 환경 (하드웨어환경 및 소프트웨어 환경) 을 메모해 주세요.
- E) 본 제품의 매뉴얼을 준비해 주세요.

1.2.7 수리 및 유지보수 서비스

보증서에 대해

보증기간 중에 고장이 난 경우에는 보증서 기재내용에 준하여 수리를 실시합니다. 상세내용은 보증서를
참고 바랍니다.

유지보수용 부품에 대해

유지보수용 부품의 최소 보유 기간은 제품 단종 후 5년입니다.

1.2.8 제품을 3자에게 양도할 경우

A) 양도하는 고객에게

본 제품을 제 3자에게 양도할 경우에는 하기 내용을 유념해 주십시오.

- 본 제품에 첨부되어 있는 모든 것 (취급 설명서, 보증서, 부속품등) 을 양도하고, 복사물을 보유하지 않을 것.
- 본 제품에 소프트웨어가 첨부되어 있는 경우에는, 소프트웨어의 양도, 이전에 대한 조항을 확인 할 것.
- 양도 이전이 허락되지 않은 소프트웨어는 삭제할 것.

B) 양도를 받는 고객에게

당사로부터 고객에게 안내장을 보내는 경우가 있습니다. 반드시 유저등록 내용의 변경에 필요한 항목을 당사로 연락 바랍니다.

등록에 필요한 항목

본 제품의 보증서번호 및 제품명, 제품번호, 시리얼 번호.

사전에 사용한 고객의 성함/회사명, 주소, 전화번호 또는 중고품을 구입하신 점포의 명칭, 주소 및 전화번호

1.3 제품의 폐기 방법

본 제품을 폐기할 경우에는 사용 지자체의 조례에 따라 처리 바랍니다. 상세 내용은 가용 지자체에 문의 바랍니다.

第 2 章 개요

본 제품의 개요에 대해 설명합니다.

2.1 제품개요

본 제품은 CameraLink 인터페이스를 대응하는 PCI Express 영상입력 보드입니다. 소형 SDR26 핀 커넥터를 4개 탑재하여 CameraLink Base/Medium/Full Configuration 카메라를 2대 접속 가능합니다. 또한 I/O 인터페이스를 구비하여 외부기기와 연동하여 영상입력제어 가능합니다.

주의) 일반적으로 시판되고 있는 마더보드·PC에 사용되는 그래픽 카드에 지정된 16 레인보드에 당사 제품을 장착하여 사용할 경우, 정상적으로 동작하지 않을 가능성이 있습니다. 이는 그래픽 카드의 성능을 최대한 발휘할 수 있도록 튜닝되어 있는 경우가 있으므로, 이와 같이 사양이 명확하지 않는 경우는 정상적으로 동작하지 않을 수 있습니다

2.2 특징

본 제품은 하기와 같은 특징을 가지고 있습니다.

- PCI Express 2.0 (Gen2) 5.0GT/s x4에 의해 고속 영상취득이 가능합니다
- CameraLink Base/Medium/Full Configuration에 대응한 라인, 에어리어 카메라를 접속 가능합니다.
- PoCL 기능을 실장하여, 카메라에 전원을 공급합니다.
- 외부 I/O 인터페이스를 탑재, 외부 기기와 연계와 끼어넣기 생성이 가능합니다.
- 스트로브 타이밍 출력을 탑재하여 외부 조명장치에 노출 타이밍을 통지 가능합니다 .

- 512MByte 의 대용량 버퍼를 실장하고 있습니다.
- RoHS 대응 제품입니다.

※MEGA-FIFO 논리 구조에 대해

MEGA-FIFO 논리구조는 카메라에서 의 영상데이터를 On board 의 대용량 버퍼에 일시 저장하여,

버퍼를 경유하여 호스트 메모리에 전송하는 구조입니다.

하드웨어는 카메라의 최고속도 동작에서 버퍼입력을 보증합니다.

(Base Configuration 카메라의 경우, 최대 데이터 Rate 는 $85\text{MHz} \times 32\text{bit} = 340\text{MByte/sec}$ 가 됩니다.)

PCI Express 의 대역폭이 일시적으로 카메라의 Frame rate 보다 낮을 경우에도 버퍼가 Full 이 될때까지 영상입력이 가능합니다.

2.3 사양 (Specifications)

The specifications of this frame-grabber are described as follows.

Item	Specification
Image Input	Camera Link Base/Medium/Full 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	4 TTL (trigger input) 4 RS422 (encoder/trigger input) 4 Opto (general-purpose input) 4 Open-Collector/TTL (strobe output) 8 Open-Collector (general-purpose output)
System bus	PCI Express® Base Specification Revision 2.0 (Gen2) Link Speed 5.0GT/s (Max) 4 Lane
MEGA-FIFO	DDR2 SDRAM 512MB
Interrupt	Start of acquisition End of DMA

	Falling edge of general-purpose input
Power Requirements	+12V 1.5A Max(PoCL OFF)
Environment Requirements	Temperature: 0°C to 50°C Humidity: 35% to 85% (non-condensing)
Dimension	167.65mm x 111.15mm (Half Size x4 PCI Express™ card)
Weight	144g
Software Support	Software Development Kit(SDK-AcapLib2) 32-bit and 64-bit Windows platform
RoHS compliant	Yes

Table 2-1 : Specifications

2.4 블록 다이어그램

본 제품의 블록 다이어그램은 아래와 같습니다.

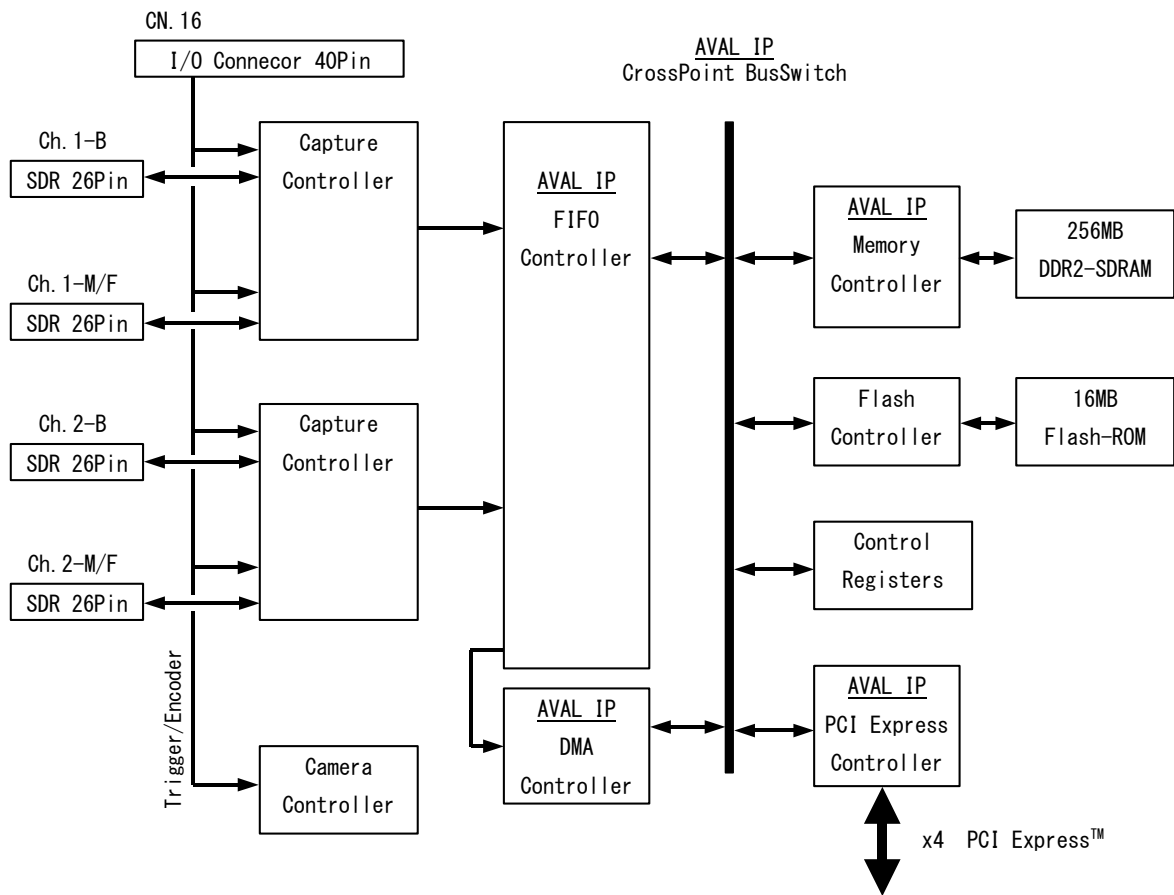


Figure 2-1 : Block Diagram

2.5 프론트 패널 외관도

본 제품의 프론트 패널 외관도는 아래와 같습니다.

SDR26 핀 CameraLink 커넥터
HDR-EC26LFDT1-SLD+ (혼다 통신공업제품)

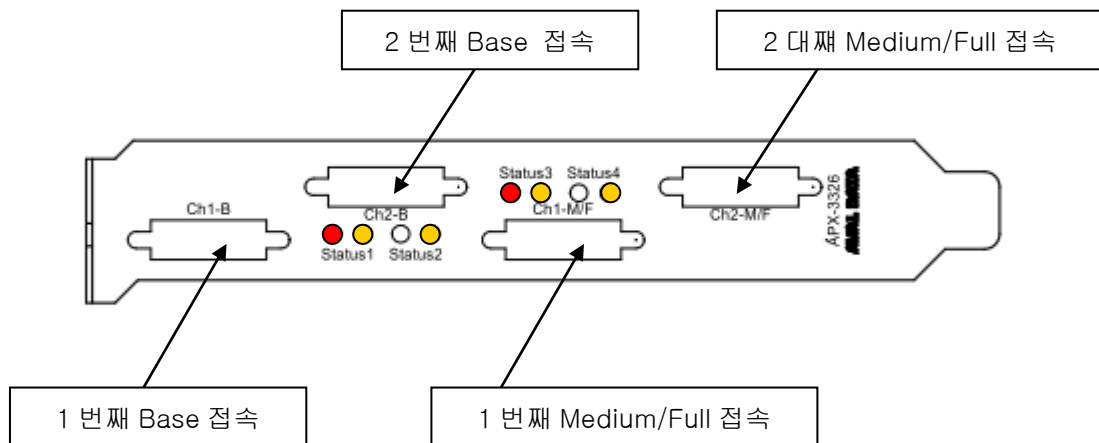


Figure 2-2 : Status 확인 LED(적/황)

주의) 사용하는 PC와 카메라링크 커넥터에 따라서는 PC와 CameraLink 케이블 커넥터의 간섭으로 인해 사용할 수 없는 경우가 있습니다.

LED (LED)	STATE (상태)
RED LED OFF (적색소등)	NOT SUPPLY POWER TO CAMERA 카메라에 전원 공급이 되지 않고 있음
PULSE RED LED (적색점등)	SUPPLYING POWER 카메라에 전원 공급중
RED LED BLINKING (적색고속점멸)	SUPPLYING POWER ERROR PoCL ON ERROR or OFF ERROR 발생
YELLOW LED OFF (황색소등)	CAMERA NOT CONNECT 카메라를 검출하지 못함
PULSE RED YELLOW (황색점등)	CAMERA CONNECT 카메라를 검출함
YELLOW LED BLINKING (황색 고속점멸)	IMAGE INPUT OPERATION 영상입력중
RED AND YELLOW LED SLOW BLINKING (적황 저속점멸)	LICENCE ERROR 라이선스 에러 발생

Figure 2-3 : Front panel

2.6 커넥터핀 배치

본 제품에 탑재되어 있는 커넥터에 대해 설명합니다.

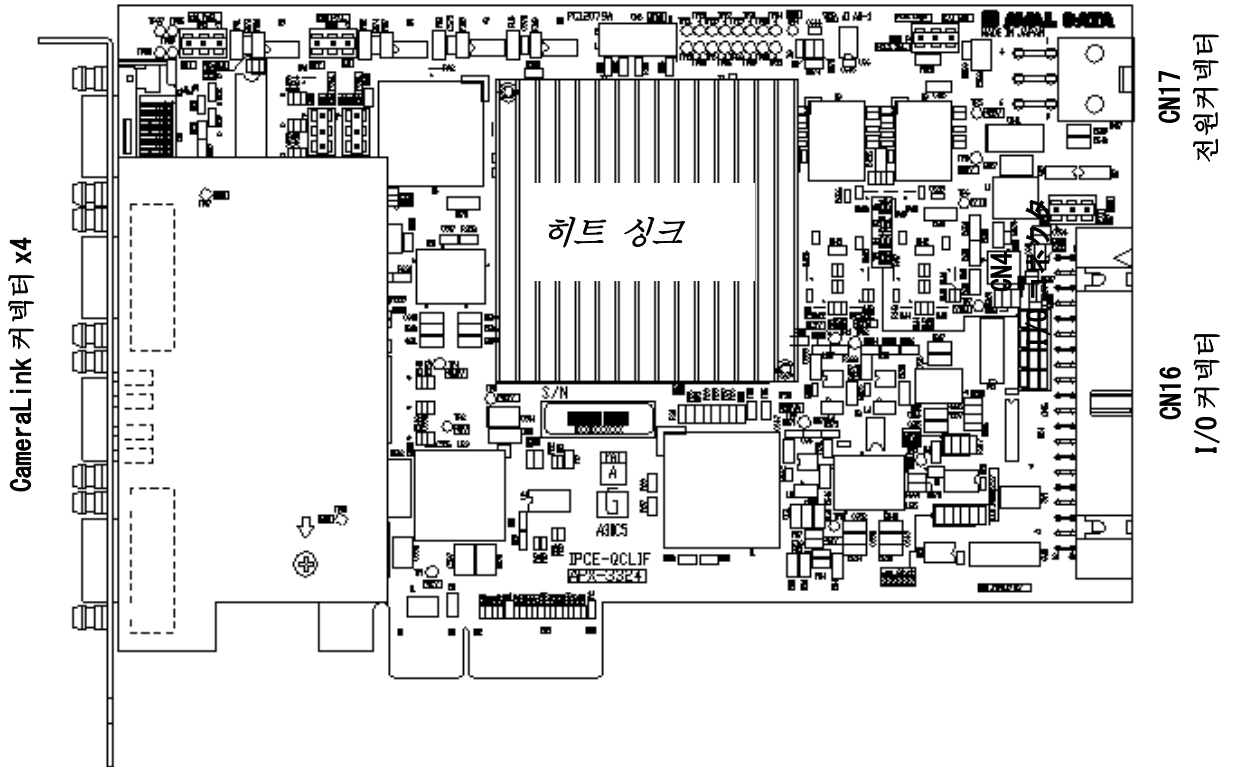


Figure 2-4 : 커넥터 배치도

2.6.1 CameraLink 커넥터

CameraLink 커넥터의 사양은 아래와 같습니다. CameraLink 규격에 준합니다.

메이커 : 혼다 통신공업

모델명 : 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

Table 2-2 : CameraLink 커넥터 핀배치 (Base)

Pin	Signal	Pin	Signal
1	Inner_Shield/+12V	14	Inner_Shield
2	Z3+	15	Z3-
3	Z_CLK+	16	Z_CLK-
4	Z2+	17	Z2-
5	Z1+	18	Z1-
6	Z0+	19	Z0-
7	Inner_Shield	20	100Ω
8	Y3+	21	Y3-
9	Y_CLK+	22	Y_CLK-
10	Y2+	23	Y2-
11	Y1+	24	Y1-
12	Y0+	25	Y0-
13	Inner_Shield	26	Inner_Shield/+12V

Table 2-3 : CameraLink 커넥터 핀배치 (Medium/Full)

2.6.2 외부전원 커넥터 CN17

외부전원 커넥터의 사양은 아래와 같습니다.

메이커 : 몰렉스

모델명 : 5569-06A2-210

Pin	Signal
1	+12V
2	+12V
3	+12V
4	GND
5	GND
6	GND

Table 2-4 : 외부전원 커넥터 핀배치

CN17에는 PCI Express 용의 6핀 전원 커넥터를 접속 할 수 있습니다.

2.6.3 확장 커넥터 CN16

확장 커넥터에는 외부기기와의 접속용으로 각종 I/O 신호가 배치되어 있습니다.

사양은 아래와 같습니다.

모델명 : 3M

모델명 : 7640-5002PL

Pin	Signal	Description	Pin	Signal	Description
1	P_COM/+5V	영상용 Common 입력/+5V 출력	2	GND	Ground
3	OPTO_IN1	포토커플러 외부 트리거입력 1	4	OPTO_IN2	포토커플러 외부 트리거입력 2
5	OPTO_IN3	미접속	6	OPTO_IN4	미접속
7	TTL_IN1	TTL 외부 트리거 1	8	TTL_IN2	TTL 외부 트리거 2
9	TTL_IN3	미접속	10	TTL_IN4	미접속
11	GND	Ground	12	GND	Ground
13	RS422_IN1+	RS422 엔코더 A+	14	RS422_IN1-	RS422 엔코더 A-
15	RS422_IN2+	RS422 엔코더 B+	16	RS422_IN2-	RS422 엔코더 B-
17	RS422_IN3+	RS422 엔코더 Z+	18	RS422_IN3-	RS422 엔코더 Z-
19	RS422_IN4+	RS422 외부트리거+	20	RS422_IN4-	RS422 외부트리거-
21	GND	Ground	22	GND	Ground
23	STB_OUT1	스트로브출력 1(TTL/OC)	24	STB_OUT2	스트로브출력 2(TTL/OC)
25	STB_OUT3	미접속	26	STB_OUT4	미접속
27	GND	Ground	28	GND	Ground
29	GP_OUT1	범용출력 1(TTL/OC)	30	GP_OUT2	범용출력 2(TTL/OC)
31	GP_OUT3	범용출력 3(TTL/OC)	32	GP_OUT4	범용출력 4(TTL/OC)
33	GP_OUT5	범용출력 5(TTL/OC)	34	GP_OUT6	범용출력 6(TTL/OC)
35	GP_OUT7	범용출력 7(TTL/OC)	36	GP_OUT8	범용출력 8(TTL/OC)
37	GND	Ground	38	GND	Ground
39	NC	No connected	40	NC	No connected

Table 2-5 : 확장 커넥터 핀배치

당사에서는 확장 커넥터부터 PCI 표준 패널까지 배선이 완료된 확장패널 케이블 「CBL-Z102」을 표준으로 하고 있습니다.

포토 커플러용 Common 입력은 +12V 입니다.

I/O 에 대한 상세내용은 [6.5 항 외부신호 입출력](#)을 참조 하세요.

第 3 章 초기설정

본제품의 하드웨어 초기설정과 소프트웨어 초기설정에 대해 설명합니다.

3.1 하드웨어 초기설정

본 제품을 PC 등의 장치에 장착하기 전에 필요한 스위치, 점퍼설정을 실행합니다.

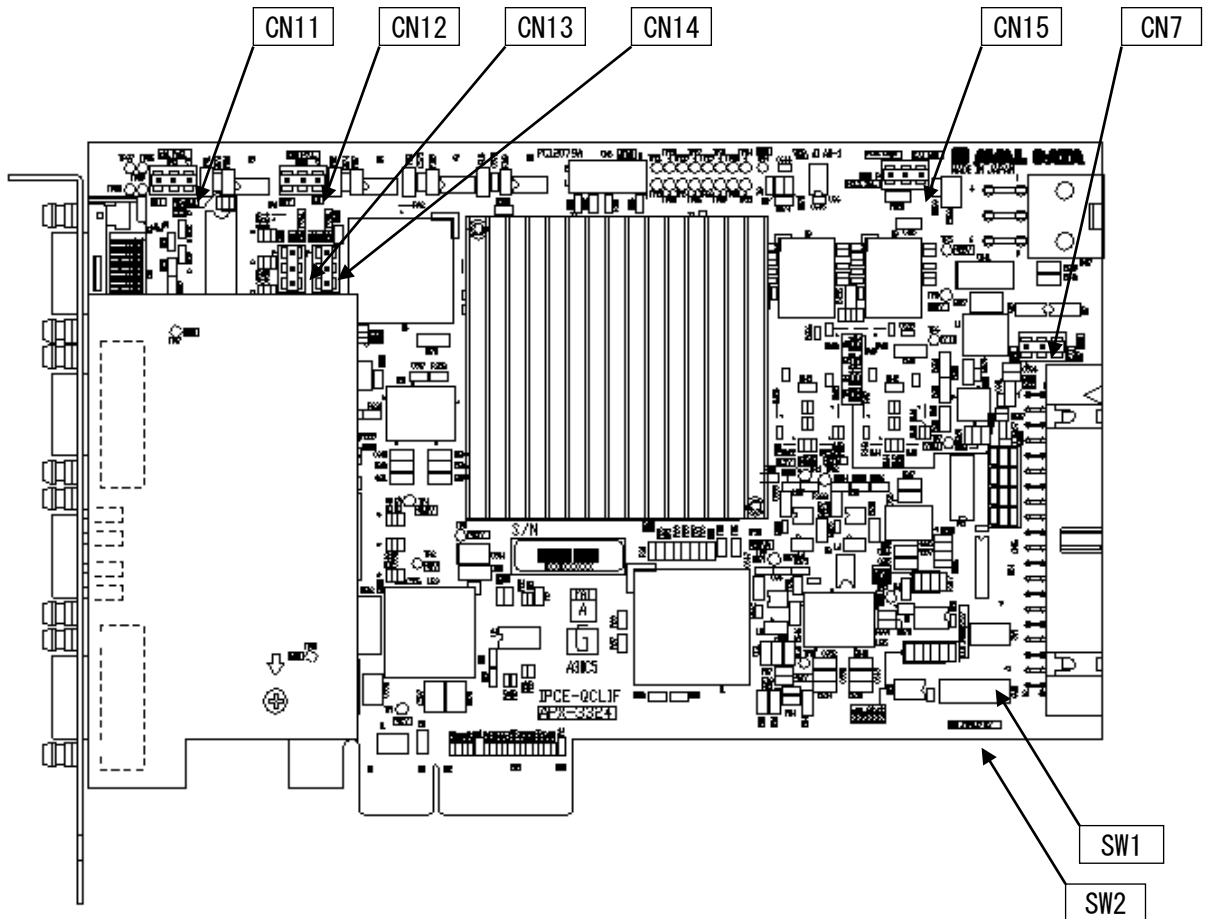


Figure 3-1 : 점퍼, 스위치 배치도

점퍼, 스위치 설정변경은 반드시 전원을 끈 상태에서 실행해 주세요

3.1.1 SW1 설정

SW1 은 스트로브 출력의+5VFull up 설정을 실행합니다. OFF 는 Full up 저항을 분리하고, ON 은 Full up 저항을 접속합니다.

Full up 저항값은 330Ω입니다.

스위치와 신호의 관계는 아래와 같습니다.

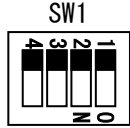


Figure 3-2 : SW1 설정

- SW1-1 : 스트로브출력-1 Ch1
- SW1-2 : 스트로브출력-2 Ch2
- SW1-3 : 스트로브출력-3 Ch3
- SW1-4 : 스트로브출력-4 Ch4

출하시에는 모두 OFF, Full up 저항은 분리되어, Open collector 출력으로 설정되어 있습니다.

3.1.2 SW2 설정

SW2 는 범용출력의+5VFull up 설정을 실행합니다. OFF 는 Full up 저항을 분리하고, ON 은 Full up 저항을 접속합니다.

Full up 저항값은 330Ω입니다.

스위치와 신호의 관계는 아래와 같습니다.



Figure 3-3 : SW2 설정

- SW2-1 : 범용출력-1 GP_OUT1
- SW2-2 : 범용출력-2 GP_OUT2
- SW2-3 : 범용출력-3 GP_OUT3
- SW2-4 : 범용출력-4 GP_OUT4
- SW2-5 : 범용출력-5 GP_OUT5
- SW2-6 : 범용출력-6 GP_OUT6
- SW2-7 : 범용출력-7 GP_OUT7
- SW2-8 : 범용출력-8 GP_OUT8

출하시에는 모두 OFF, Pull up 저항은 분리되어, Open collector 출력으로 설정되어 있습니다.

3.1.3 CN11 ~ CN14 설정

카메라 채널마다 PoCL 기능의 유효, 무효를 설정합니다. ON 은 PoCL 유효, OFF 는 PoCL 무효입니다. CN11 는 카메라 ch1 에 CN12 는 카메라 ch2 에 CN13 는 카메라 ch3 에 CN14 는 카메라 ch4 에 대응합니다.

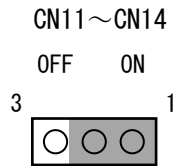


Figure 3-4 : CN11 ~ CN14 설정

출하시에는 CN11 ~ CN14 모두 ON 으로 PoCL 기능은 유효로 설정되어 있습니다.

※PoCL 기능을 사용하기 위해서는 본 설정만이 아니라, 소프트웨어에서 전원공급을 제어해야 합니다. 상세 내용은 [6.2 Power Over CameraLink](#) 를 참조해 주세요

PoCL 비 대응 카메라를 PoCL 유효설정된 상태에서 사용하는 것은 가능하지만, 이 경우 PC 에 전원을 넣기전에 카메라의 전원을 먼저 입력하여 주세요. PC 부팅시에 카메라의 전원이 입력되지 않은 경우는 PC 에 따라서는 +12V 전원이 내려가서 OS 가 정상적으로 작동하지 않을 수 있습니다.

3.1.4 CN15 설정

PoCL 전원 공급원을 설정합니다. PCI Edge 에서 PCI Express 카드 엣지로부터 전원공급, EXT Con 에서 CN17 로부터 전원을 공급합니다.

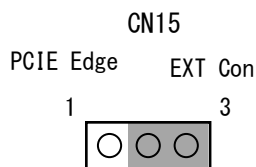


Figure 3-5 : CN15 설정

출하시에는 EXT Con 로 설정되어 있습니다.

PCI Edge 로 설정한 경우에는 본 제품과 카메라의 총전력을 25W 이하에서 사용해 주세요

PoCL 비 대응 카메라를 접속할 경우에도 EXT Con 으로 설정해 주세요

(PoCL 비 대응 카메라만 접속한 경우에는 CN17 로부터의 전원공급은 필요 없습니다. 점퍼 설정만 실행 해 주세요)

※본 제품의 소비전력은 18W 입니다. PoCL 카메라 1 대만 접속할 경우에는 PCI Express 슬롯에서 전원공급이 가능합니다.

주의 3) 2대 이상의 PoCL 카메라를 접속할 경우에는 아래 설정을 실행해 주세요

- CN15을 EXT Con 에 설정 (디폴트)
- CN17 에 PCI Express 용 6 핀 전원 커넥터를 접속

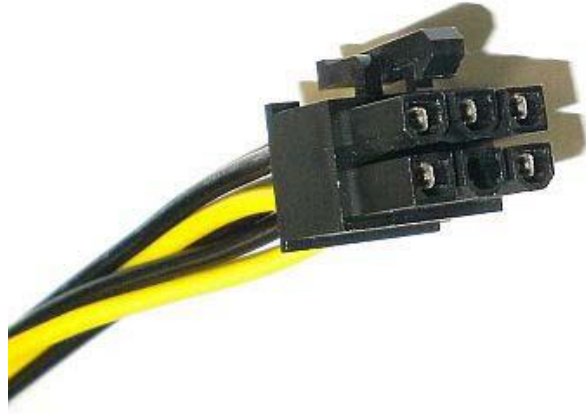


Figure 3-6 : PCI Express 용 6 핀 전원 커넥터

PC에 6 핀 전원 커넥터가 없는 경우에는, 일반적으로 판매되는 변환 케이블을 사용해 주세요

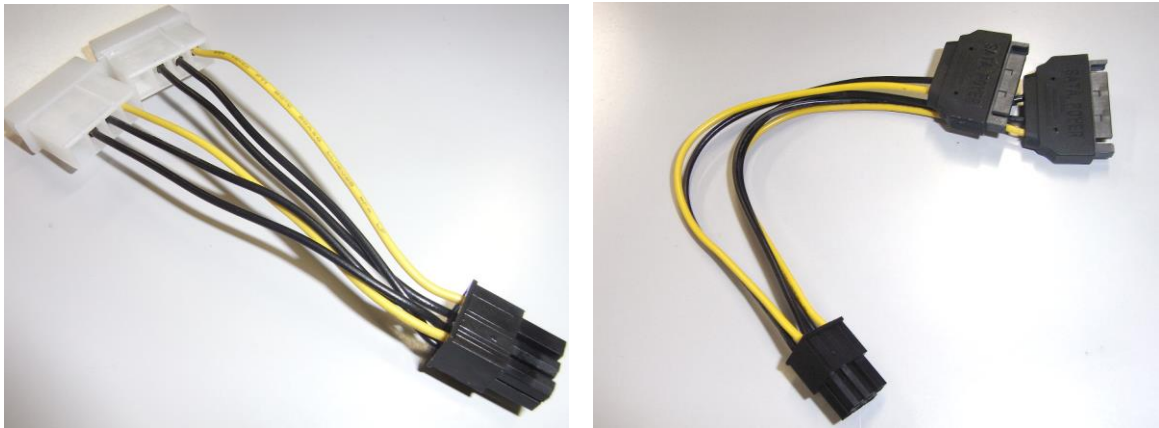


Figure 3-7 : 전원 커넥터 변환 케이블(예)

카메라 접속과 점퍼 설정에 대해서는 아래 표를 참조해 주세요

접속 상태		CN15	CN11 ~ 14
PoCL 대수	비 PoCL 대수		
0	1	EXT Con (전원공급 필요없음)	모든 Ch 을 OFF
0	2	EXT Con (전원공급 필요없음)	모든 Ch 을 OFF
1	0	EXT Con 또는 PCI Edge	PoCL 카메라를 접속 하는 Ch 을 ON, 카메라를 접속하지 않는 Ch 을 OFF
2	0	EXT Con	모든 Ch 을 ON
1	1	EXT Con 또는 PCI Edge	PoCL 카메라를 접속하는 Ch 을 ON 비 PoCL 카메라를 접속하는 Ch 을 OFF

Table 3-1 : PoCL 점퍼 설정

카메라에 +12V 접속은 아래와 같습니다.

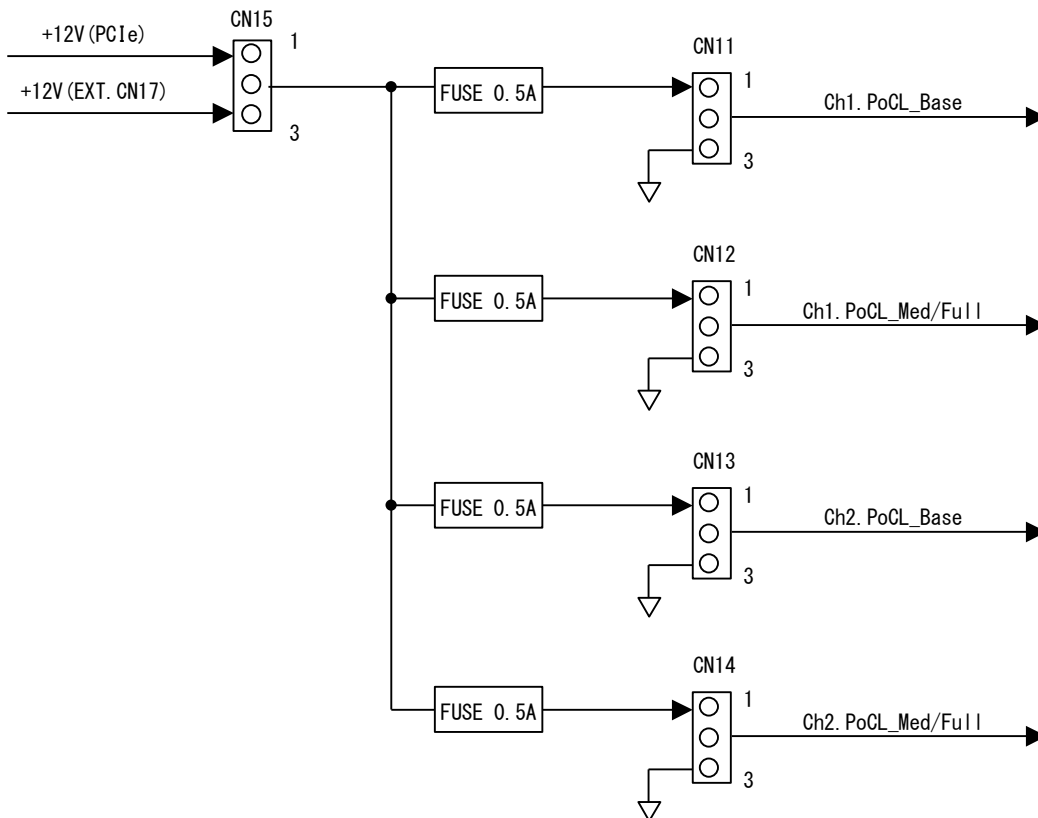


Figure 3-8 : PoCL 전원공급

3.1.5 CN7 설정

확장 커넥터 1pin 의 영상용 Common 입력/+5V 출력을 변환합니다. INT 에서+5V 출력설정, EXT 에서 영상용 Common 설정을 합니다.

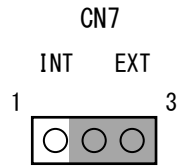


Figure 3-9 : CN7 설정

출하시에는 CN7 는 INT 로 설정되어 있습니다.

※INT 설정시에는 확장 커넥터 1pin 에 전원을 입력하지 마십시오.

※+5V 출력설정으로 한 경우에는 APX-3324 에서 최대 500mA 공급 가능합니다.

3.1.6 PCI Express 슬롯에 실장

본 제품을 PCI Express 커넥터가 탑재되어 있는 시스템에 탈 부착을 할 경우에는 아래 내용을 주의해 주세요.

- PCI Express 슬롯 선정시의 주의점

- a) PCI Express x4 이상의 슬롯에 장착해 주세요
- b) Gen1/Gen2 모두 사용 가능합니다.

- 본 제품을 장착할 때 주의점

- a) 본 제품의 점퍼가 바르게 설정되어 있는지 확인 바랍니다.
- b) 사용 시스템의 PCI Express 슬롯에 제품을 장착 합니다.
이때 무리하게 힘을 가하지 않고, 직선방향으로 장착해 주세요. 패널상에 배치된 입력단자가 걸려서 고장의 원인이 되거나 모듈상의 부품을 파손시킬 위험이 있습니다.
또한, 장착한 모듈의 커넥터와 시스템의 커넥터가 바르게 접속되어 있는지 확인해 주세요
- c) 본 제품을 장착한 후에는 PC의 패널에 나사로 고정해 주세요.
시스템에 장착이 완료되면 시스템 전원을 켜서 동작확인을 해 주세요.

- 본 제품을 탈착할 때의 주의점

- a) 본 제품을 탈착하기전에 반드시 시스템 전원을 꺼 주세요. 고장의 원인이 됩니다.
전원을 끈 직후에는 제품이 뜨거움으로 취급에 주의 바랍니다.
- b) 패널에 고정된 나사를 빼내고 기판상의 부품을 만지지 않은 상태에서 무리한 힘을 가하지 않고 탈착해 주세요.
무리하게 탈착을 하면 모듈 및 시스템의 커넥터가 파손될 위험이 있습니다.
- c) 본 제품을 탈착할 때에는 패널상에 배치된 입출력단자가 걸릴 위험이 있으므로 무리하게 탈착을 하지 말아 주세요. 고장의 원인이 됩니다.

APX-3324 슬롯에 장착 후, 패널에 나사로 고정해 주세요

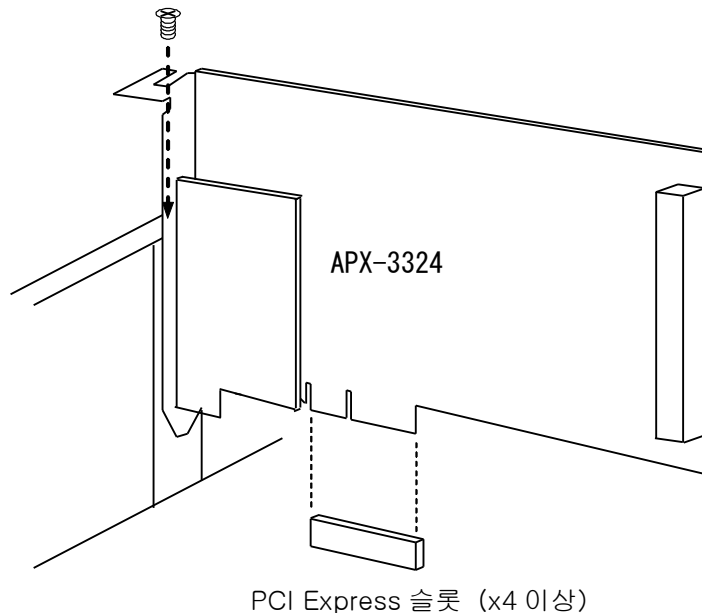


Figure 3-10 : 장착

3.1.7 케이블 접속

본제품의 기본적인 접속방법은 아래와 같습니다.

- 본 제품과 카메라 사이에는 CameraLink 규격에 준한 케이블을 접속해 주세요
- 본 제품과 외부기기를 접속 할 경우에는 확장케이블 CBL-Z102 를 이용해 주세요
 확장 케이블 CBL-Z102 의 상세사양은 케이블 팸플렛을 참조 해 주세요

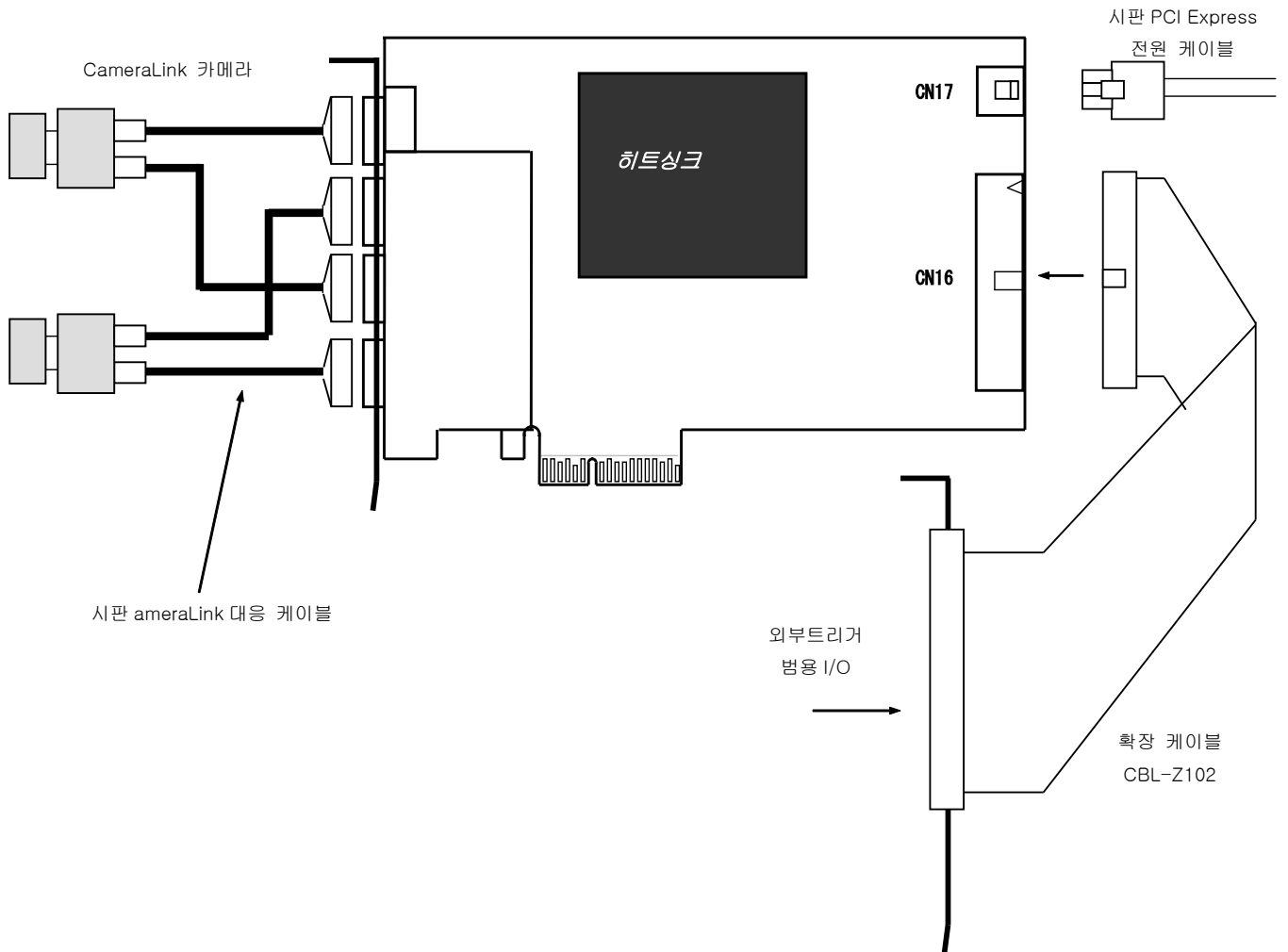


Figure 3-11 : 전원 커넥터, 확장 케이블 접속

3.1.8 확장 패널 케이블 CBL-Z102

APX-3324의 뒷면에 실장된 확장 커넥터에 접속하는 Dsub37 핀 PCI 블랑켓 포함 케이블입니다.
(제품에 포함되어 있지 않습니다. 옵션으로 판매)

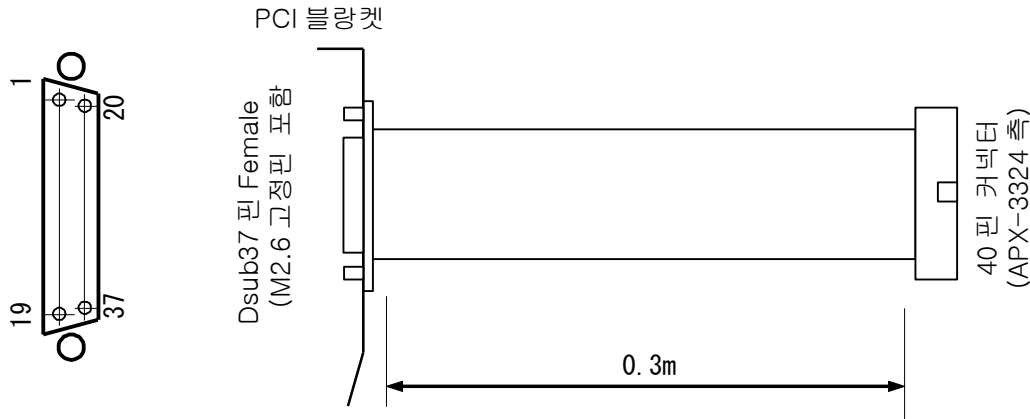


Figure 3-12 : 확장 케이블 CBL-Z102

Dsub 커넥터의 핀배치는 아래와 같습니다.

Pin	Signal	Description	Pin	Signal	Description
1	P_COM/+5V	영상용 Common 입력/+5V 출력	20	GND	Ground
2	OPTO_IN1	포토커플러 외부 트리거입력 1	21	OPTO_IN2	포토커플러 외부 트리거입력 2
3	OPTO_IN3	포토커플러 외부 트리거입력 3	22	OPTO_IN4	포토커플러 외부 트리거입력 4
4	TTL_IN1	TTL 외부 트리거 1	23	TTL_IN2	TTL 외부 트리거 2
5	TTL_IN3	TTL 외부 트리거 3	24	TTL_IN4	TTL 외부 트리거 4
6	GND	Ground	25	GND	Ground
7	RS422_IN1+	RS422 엔코더 A+	26	RS422_IN1-	RS422 엔코더 A-
8	RS422_IN2+	RS422 엔코더 B+	27	RS422_IN2-	RS422 엔코더 B-
9	RS422_IN3+	RS422 엔코더 Z+	28	RS422_IN3-	RS422 엔코더 Z-
10	RS422_IN4+	RS422 외부트리거+	29	RS422_IN4-	RS422 외부트리거-
11	GND	Ground	30	GND	Ground
12	STB_OUT1	스트로브출력 1 (TTL/OC)	31	STB_OUT2	스트로브출력 2 (TTL/OC)
13	STB_OUT3	스트로브출력 3 (TTL/OC)	32	STB_OUT4	스트로브출력 4 (TTL/OC)
14	GND	Ground	33	GND	Ground
15	GP_OUT1	범용출력 1 (TTL/OC)	34	GP_OUT2	범용출력 2 (TTL/OC)
16	GP_OUT3	범용출력 3 (TTL/OC)	35	GP_OUT4	범용출력 4 (TTL/OC)
17	GP_OUT5	범용출력 5 (TTL/OC)	36	GP_OUT6	범용출력 6 (TTL/OC)
18	GP_OUT7	범용출력 7 (TTL/OC)	37	GP_OUT8	범용출력 8 (TTL/OC)
19	GND	Ground			

Table 3-2 : Dsub 커넥터 핀배치

3.2 소프트웨어 초기설정

본 제품의 제어를 위해서는 SDK-AcapLib2 의 인스톨이 필요합니다.

3.2.1 SDK-AcapLib2 에 대해

SDK-AcapLib2(Software Development Kit AvalCaptureLibrary2)는 본 제품을 포함한 여러기종의 보드를 제어하기 위해 사용하는 소프트웨어 패키지 입니다.

SDK-AcapLib2 에는 아래 내용이 포함 되어 있습니다.

■Aval Capture Library 2(이하, AcapLib2 라고 합니다.)

각종 보드를 공통 인터페이스로써 제어 가능한 라이브러리 입니다.

■각종 보드용 드라이버

본 제품을 포함하는 각종 보드용 드라이버 입니다.

■샘플 소스

각종 언어에 대응한 취득 샘플(VC++ / VC# / VB.NET). 각종 영상처리 라이브러리 접속 샘플등이 있습니다.

■툴

AcapLib2Viewer, 가상 COM 툴, 펌웨어 갱신 툴, 라인 프로파일 툴 AZP-8100-01(영상처리 툴)등이 있습니다.

■카메라 설정(ini)파일

접속실적이 있는 카메라에 대해서 보드측의 초기설정을 실행할때 사용하는 파일입니다.

■문서

각종 매뉴얼, 인스톨 가이드, 샘플 도큐먼트등이 있습니다.

3.2.2 SDK-AcapLib2 인스톨 순서

SDK-AcapLib2 의 인스톨 순서에 대해 설명합니다.

[手順1] - 순서 1

인스톨 디스크를 CD-ROM/DVD-ROM 드라이브에 셋업하면 아래의 웹 페이지가 표시됩니다. 메뉴에서 [ソフトウェア-소프트웨어]를 선택해 주세요

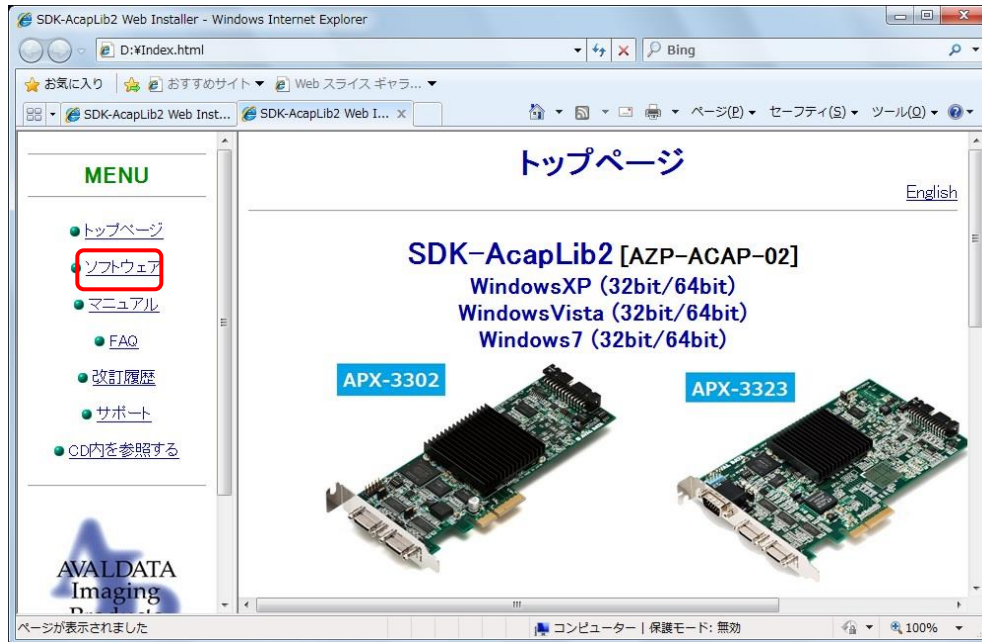


Figure 3-13 : 웹페이지

인스톨 전 주의 사항

- 인스톨 작업을 실행하기전에 보드가 PC 에 실장되어 있는지 확인해 주세요
- BIOS 화면에서「DMA Master」가「ON(YES)」로 되어 있는 것을 확인해 주세요. (*)
- 로그인 할 경우에는 사용자명이「Administrator」또는 「드라이버등의 인스톨이 허가된 사용자명」으로 로그인 해 주세요.
- 저 전력기능의 대기상태(ACPI S1/S3 모드)는 사용할 수 없습니다.
CD-ROM 내 [マニュアル-메뉴얼]-[Windows Stand by 유무화 순서]를 참조하여 Sleep 상태/ 휴정상태를 무효화로 설정하여 주세요
- 32 비트 OS 에서는 3G 옵션에 대해서는 지원 하고 있지 않습니다.
3G 옵션을 해제하고 사용하는 것을 권장합니다.
- 64 비트 OS 에서의 WOW64 에 대해서는 지원하고 있지 않습니다.
- Windwos7 의 Windows XP Mode 및 Windows XP 호환 모드에 대해서는 지원하지 않습니다.

AVAL DATA CORPORATION

■이미 다른 버전이 인스톨 된 경우에는 반드시 먼저 언인스톨을 하고 인스톨을 실행 해 주세요.
 언인스톨의 순서에 대해서는「[3.2.6 각 소프트웨어의 언 인스톨](#)」을 참조하여 실행해 주세요.

(*)상기 설정항목은 PC 에 따라서는 항목이 없는 경우가 있습니다. 없을 경우에는 무시하여도 됩니다.

[手順2] - 순서 2

[手順3]

[SDK-AcapLib2]의 옆의[インストール - 인스톨]을 선택해 주세요

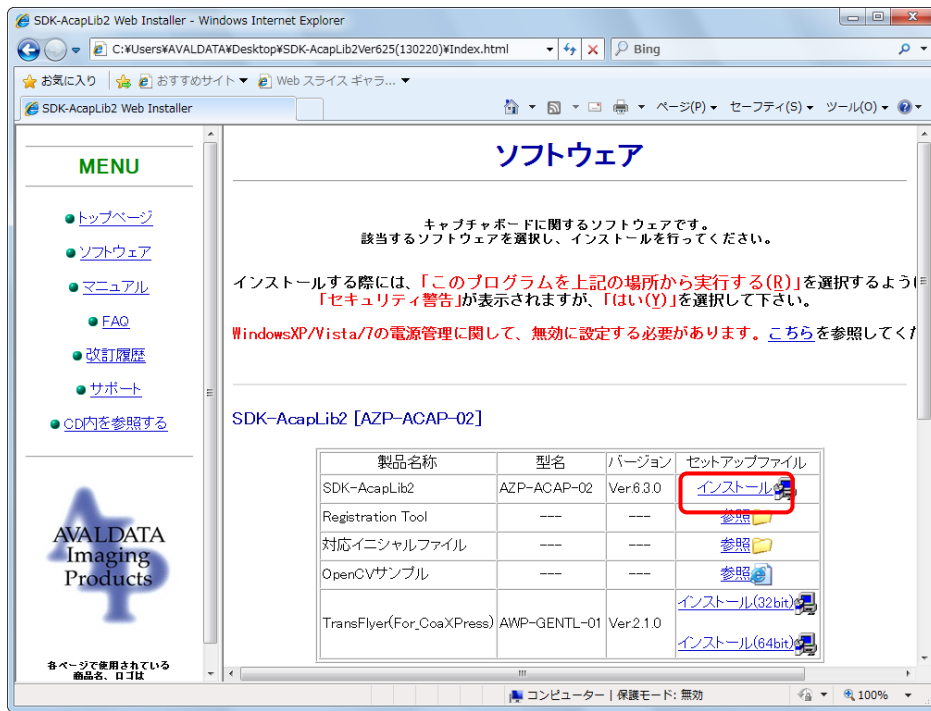


Figure 3-14 : 소프트웨어

[手順4] - 순서 3

[파일 다운로드]가 표시되면 [実行-실행]을 선택해 주세요



Figure 3-15 : 파일 다운로드

[手順5] - 순서 4

[Install]을 선택해 주세요

※이미 다른 버전이 인스톨 되어 있는 경우에는 [Uninstall]이 유효화 됩니다.

동일 버전일 경우에는 그 대로 Uninstall의 실행이 가능합니다만, 다른 버전일 경우에는 Uninstall시에 경고화면이 나타납니다.

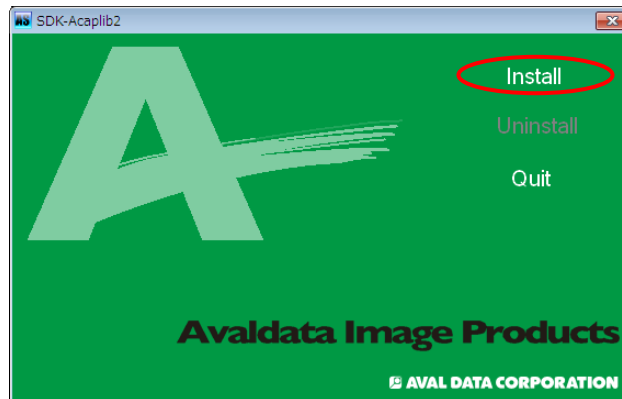


Figure 3-16 : 메뉴화면

[手順6] - 순서 5

인스톨 폴더를 변경할 경우에는 「変更-변경」을 선택하고 임의의 장소로 변경합니다. 변경이 필요하지 않을 경우에는 [OK]를 선택해 주세요

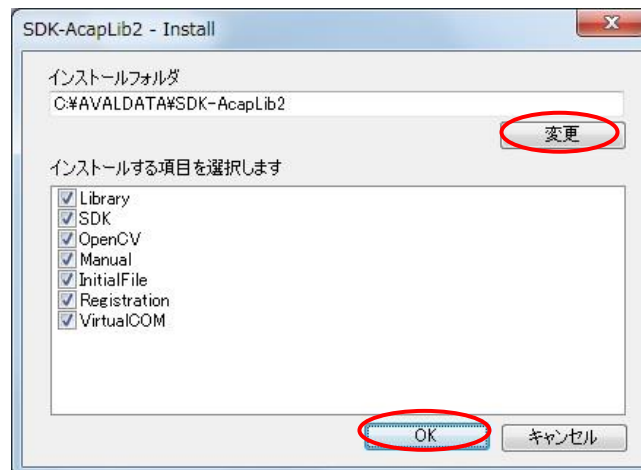


Figure 3-17 : 인스톨 항목

[手順7] - 순서 6

인스톨 선택에서 [Library]에 체크가 되어 있을 경우에는 자동적으로 「デバイスドライバのインストールウィザード - 디바이스 드라이버의 인스톨 윈도우」가 나타납니다. 지시에 따라 인스톨 해주세요.

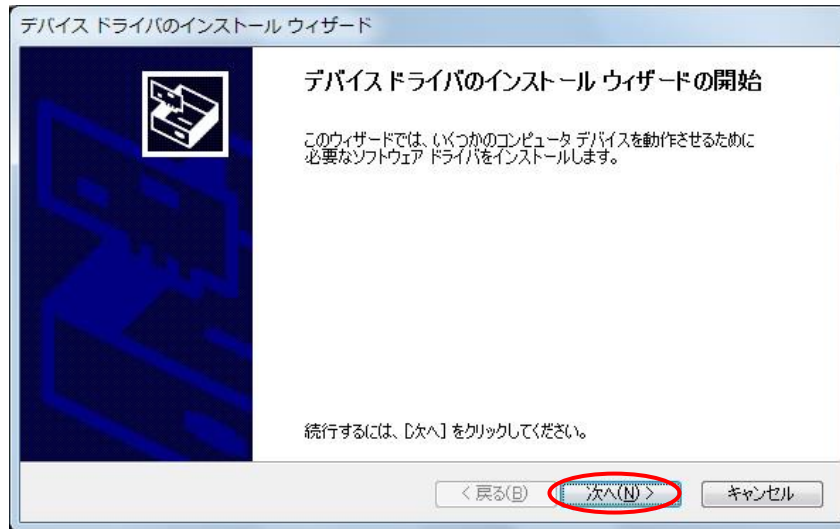


Figure 3-18 : 라이브러리 인스톨 화면

[手順8] 순서 7

인스톨 선택에서 [Registration]에 체크가 되어 있는 경우에는 자동적으로 「AVALDATA Registration Tool」이 나타납니다. Product Key 를 가지고 있는 경우에는 [製品版-제품판]을 선택하고 Pproduct Key 를 입력후 [OK]를 선택합니다. 상세내용은 [3.2.4 Registration Tool](#) 를 참조해 주세요. Product Key 가 없는 경우는 [評価版-평가판]을 선택하고 [OK]를 선택합니다.



Figure 3-19 : 소프트웨어 레지스터 등록 화면

[手順9] - 순서 8

완료 버튼을 눌러서 인스톨을 종료합니다.

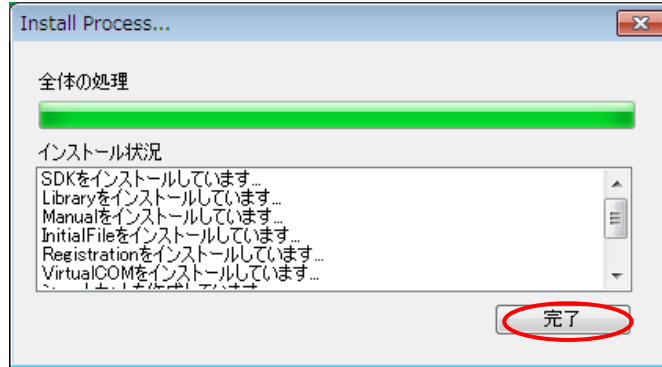


Figure 3-20: 완료화면

[手順10] - 순서 9

Quit 버튼을 눌러서 종료 합니다.



Figure 3-21 : 메뉴 화면

이상으로 인스톨 작업은 완료됩니다.

3.2.3 시리얼 통신 드라이버(가상 COM 드라이버)

가상 COM 이란 Windows 상에서 당사 보드의 각 CameraLink 포트를 가상의 COM 포트로서 할당하여 인식 하는 기능입니다. Windows 상의 표준 COM 포트에서 카메라와 통신 가능하도록 합니다.

가상 COM 을 할당하기 위해서는 가상 COM 툴을 사용합니다.

가상 COM 툴은 Stand menu 에서 [AVALDATA]⇒[SDK-AcapLib2]⇒[Tool]⇒[VirtualCOM]⇒[avdCOMmng]을 실행 해 주세요

가상 COM 툴의 인스톨/언인스톨 및 사용방법에 대해서는, 가상 COM 툴 사용자 메뉴얼(SDK-AcapLib2 VirtualComTool Manual(J2OM10016*))를 참조해 주세요.

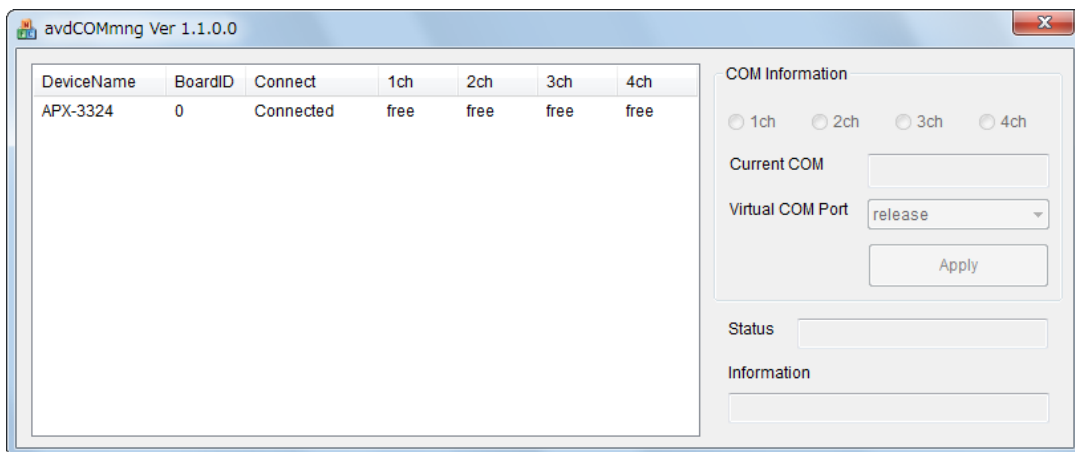


Figure 3-22 : 가상 COM 툴

3.2.4 Registration Tool

라이브러리를 사용하는 경우 「Registration Tool」을 사용하여 PC 에 등록작업을 실행할 필요가 있습니다. 등록 작업을 실행하지 않을 경우에는 기능에 제한이 있을 수 있으며, 특정 라이브러리의 함수를 불러들일 때 아래와 같은 등록이 되지 않았다는 메시지 박스가 표시됩니다.



Figure 3-23 : 미 등록 메시지

■ 등록방법

등록 방법에 대해서는 아래에 설명합니다.

[手順1] - 순서 1

시작 메뉴에서

[AVALDATA] ⇒ [SDK-AcapLib2] ⇒ [Tool] ⇒ [Registration] ⇒ [AvdRegistTool] 를 실행 해주세요

[手順2] - 순서 2

[제품판]을 선택합니다. 다음에 인스톨 디스크에 부착된 황색 스티커에 표시된 16 자리의 Product Key 를 입력해 주세요. 입력후에 [OK]를 클릭하여 주세요

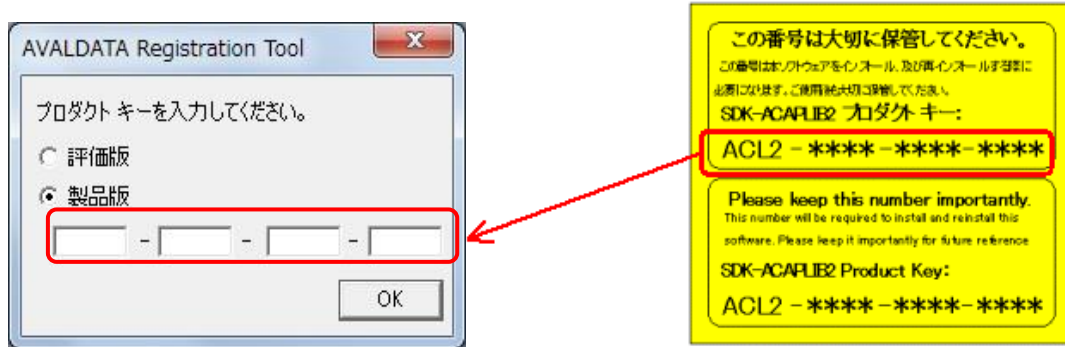


Figure 3-24 : Registration Tool 과 Product Key

[手順3] - 순서 3

아래와 같은 메시지가 표시되면 등록이 완료되었습니다.



Figure 3-25 : 등록 성공화면

3.2.5 펌웨어 갱신 툴

펌웨어 갱신틸이란 본 제품의 로컬 FPGA 펌웨어를 갱신하기위한 어플리케이션 입니다. 펌웨어 갱신기능 이외에도 소프트웨어 스위치 설정기능을 탑재하고 있습니다.

펌웨어 갱신 툴은 시작메뉴에서

[AVALDATA] ⇒ [SDK-AcapLib2] ⇒ [Tool] ⇒ [FirmwareUpdateTool] ⇒ [FirmwareUpdateTool] 을 실행해 주세요.

펌웨어 갱신틸의 상용방법에 대해서는 펌웨어 갱신틸 사용자 매뉴얼(SDK-AcapLib2 FirmwareUpdate Tool Manual(J3OM10026*))을 참조해 주세요.

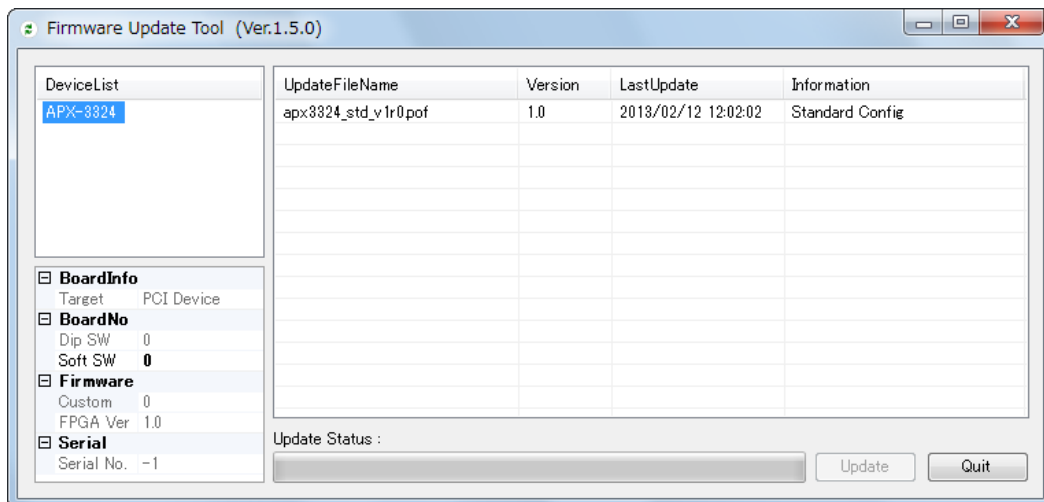


Figure 3-26 : 펌웨어 갱신 툴

3.2.6 각 소프트웨어의 언인스톨

■가상 COM 드라이버의 언인스톨에 대해

[手順1] - 순서 1

[スタート]⇒[AVALDATA]⇒[SDK-AcapLib2]⇒[Tool]⇒[VirtualCOM]⇒[avdCOMmng]을 실행합니다.

[手順2] - 순서 2

모든 COM에 할당을 삭제합니다. 상세내용은 가상 COM 툴 사용자 메뉴얼(SDK-AcapLib2 VirtualComTool Manual(J2OM10016*))을 참조해 주세요

[手順3] - 순서 3

[제어판] - [장치 및 프린터] - [장치관리자]를 선택합니다.

[手順4] - 순서 4

[포트(COM과 LPT)]안에[AVALDATA Virtual COM Port(COM*)]을 오른쪽을 클릭하고 [削除]를 선택하여주세요

[手順5] - 순서 5

「디바이스 삭제 확인」박스가 표시되면 「OK」를 누릅니다.

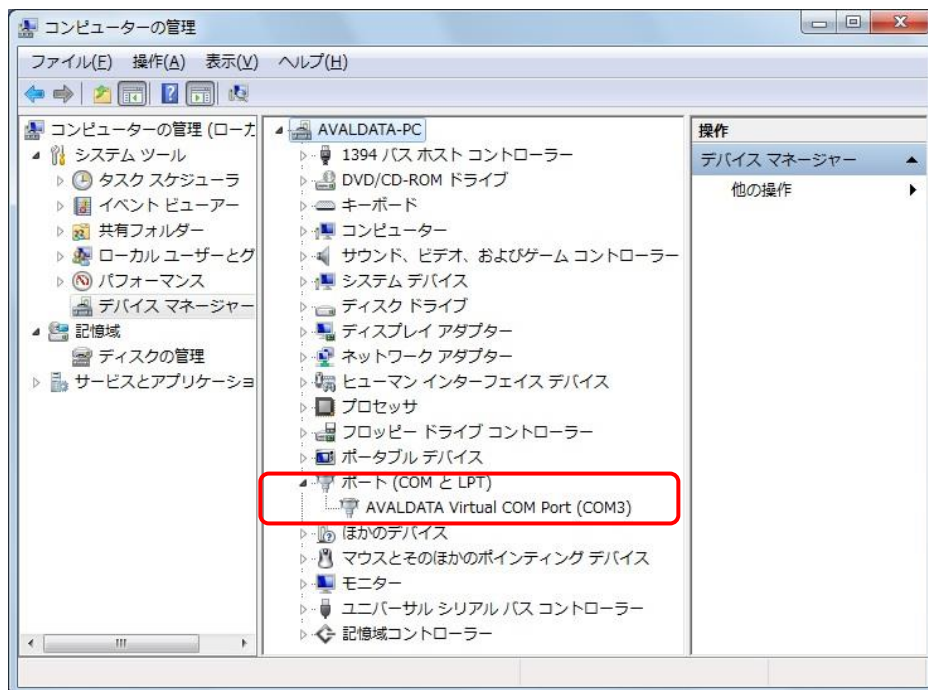


Figure 3-27 : 장치 관리자

■디바이스 드라이버 및SDK-AcapLib2의 언인스톨에 대해서

[手順1] - 순서 1

아래 프로그램을 실행해 주세요.

〈WindowsXP〉

[시작]⇒[제어판]⇒[프로그램 추가와 제거]

〈WindowsVista〉

[시작]⇒[설정]⇒[제어판]⇒[프로그램 제거]

〈Windows7〉

[시작] ⇒[제어판]⇒[프로그램 제거]

[手順2] - 순서 2

[AVALDATA SDK-AcapLib2]를 선택하고 삭제하여 주세요



Figure 3-28 : 프로그램 언인스톨

[手順3] - 순서 3

일단 위의 프로그램을 닫고, 다시 프로그램을 실행하여 아래 항목이 남아 있으면, 언인스톨을 실행해 주세요

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

3.2.7 인스톨 관련 트러블 슈팅

인스톨 작업중에 문제가 발생할 경우에는 아래 내용을 확인 바랍니다.

- 하드웨어가 바르게 장착되어 있는지 다시 확인 바랍니다.
- 다른 PCI Express 슬롯에 장착하여, 동일한 증상이 발생하는지 확인 바랍니다.
- 「[인스톨 전 주의 사항](#)」을 다시 확인 바랍니다

第 4 章 AcapLib2Viewer 로 촬영

본 제품과 AcapLib2Viewer 를 사용하여 촬영을 하기까지의 순서를 설명합니다.

4.1 AcapLib2Viewer 에 대해

AcapLib2Viewer 란 본 제품을 포함한 각종 보드를 사용하여 촬영을 하기위한 소프트웨어 입니다.

AcapLib2Viewe 에는 주로 아래와 같은 기능이 있습니다.

■보드의 초기화

취득 사이즈, 셔터 트리거, 외부 트리거, 엔코더등의 설정을 할 수 있습니다.

■카메라를 사용한 촬영

Snap(1 개화면 촬영), Grab(연속 촬영)을 실행 할 수 있습니다.

■촬영영상의 표시

확대, 축소 표시가 가능합니다. 또한 지정한 화소의 휘도값을 표시 할 수 있습니다.

■촬영 영상의 파일 출력

촬영한 영상을 각종영상 포맷(bmp, csv, bin, jpg, gif, tif, png)으로 저장 가능합니다.

■설정 파일(ini 파일)저장

AcapLib2Viewer 의 각종 설정을 ini 파일로써 저장, 불러들일 수 있습니다.

4.2 ini 에 대해

ini 파일은 취득 사이즈, 셔터 트리거, 외부트리거, 엔코더등 각종 설정값이 저장된 파일입니다.

AcapLib2Viewer 의 초기설정, AcapLib2 의 함수에서 ini 파일을 이용 할 수 있습니다.

또한 당사에서 접속한 실적이 있는 카메라에 대해서는 기본적인 영상취득이 가능한 ini 파일이

SDK-AcapLib2 에 저장되어 있습니다.

또한 당사 홈페이지의 [영상 제품대응 카메라 정보]에서 최신판 ini 파일을 다운로드 가능합니다.

[영상제품 대응 카메라 정보]

http://www.avaldata.co.jp/products/z2_imaging/supportcamera/camera.html

4.3 AcapLib2Viewer 를 사용하여 촬영

AcapLib2Viewer 를 사용하여 촬영을 하기까지의 순서를 설명합니다.

AcapLib2Viewer 의 각종설정 방법에 대해서는[SDK-AcapLib2 Viewer Manual(J2OM08014*)]를 참조해 주세요.

[手順1] - 순서 1

Windows 의 시작 메뉴에서[모든 프로그램]을 선택합니다.

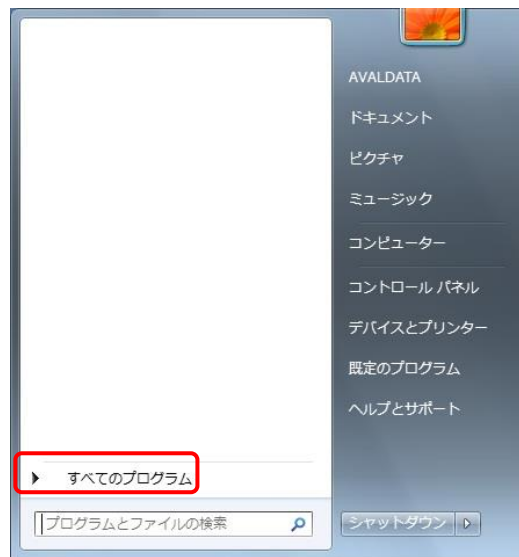


Figure 4-1 : 시작메뉴

[手順2] - 순서 2

[AVALDATA] → [SDK-AcapLib2] 내의 AcapLib2Viewer 을 선택합니다.

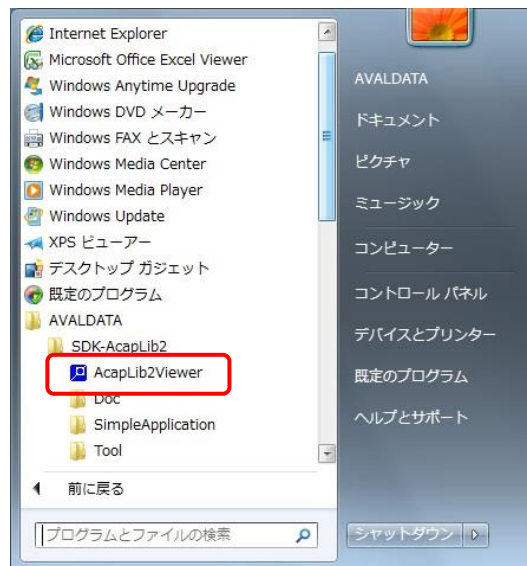


Figure 4-2 : 모든 프로그램

[手順3] - 순서 3

ini 파일, 사용하는 포트, 보드 번호, 입력채널을 선택하는 창이 표시되면 각 항목을 선택합니다.

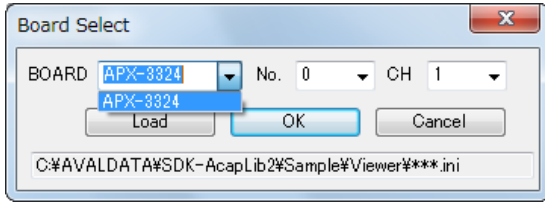


Figure 4-3 : 보드 선택

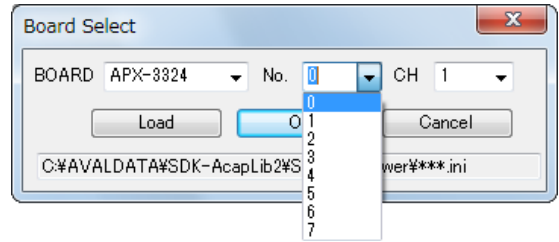


Figure 4-4 : 보드 번호 선택

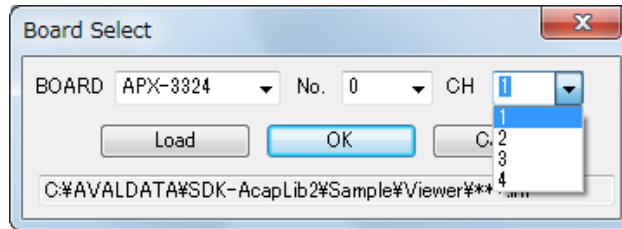


Figure 4-5 : 입력 채널 선택

■ 보드 번호에 대해

보드 번호란 1 개의 PC 에 제품이 여러 개 장착된 경우에 각 보드를 식별하는 번호 입니다. 출하시에는 0 으로 설정되어 있습니다만,「[3.2.5 펌웨어 갱신](#)」을 사용하여 임의값으로 변경 할 수 있습니다.

[手順4] - 순서 4



[Load] 버튼을 누르면 ini 파일 선택 창이 표시됩니다.

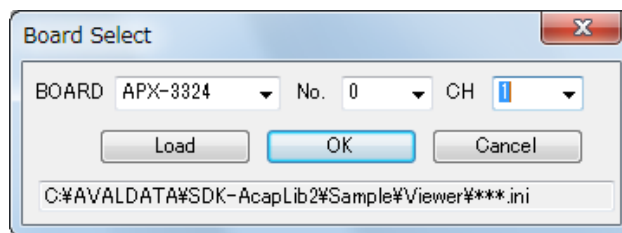


Figure 4-6 : 사용 보드등의 확인 창

[手順5] - 순서 5

[SDK-AcapLib2] → [AVALDATA] → [SDK-AcapLib2] → [Sample] → [IniFile] 폴더내에 각종 카메라 메이커의 ini 파일이 있으므로, 사용할 카메라의 ini 파일을 선택해 주세요.

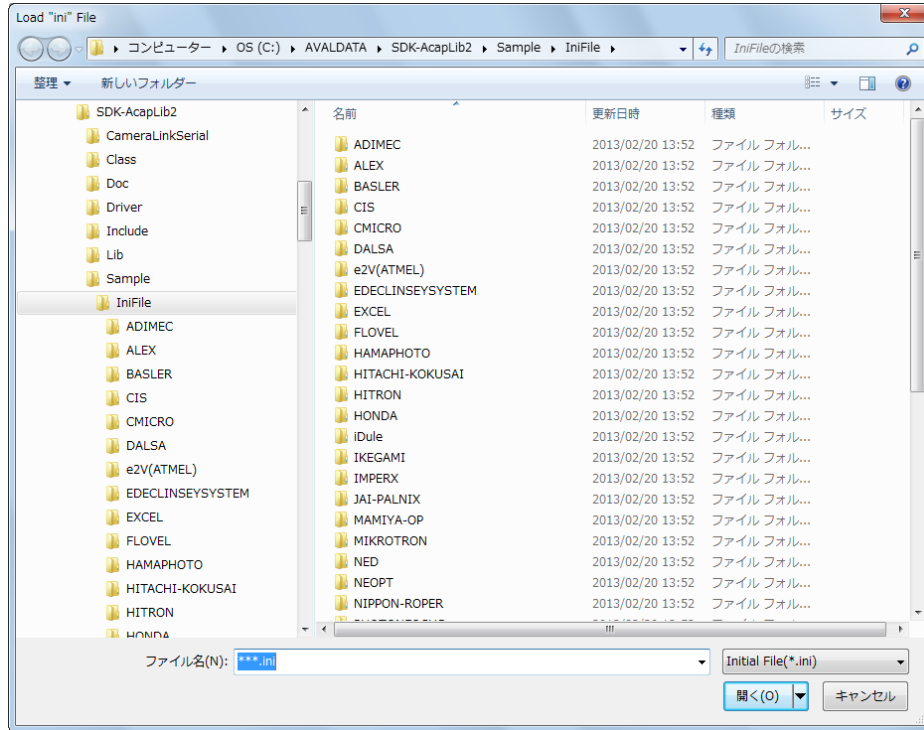


Figure 4-7 : ini 파일 선택 창

[手順6] - 순서 6

ini 파일을 선택한 후 [OK] 버튼을 눌러 주세요

[OK] 버튼을 누르면 선택된 ini 파일로 초기화를 실행 합니다.

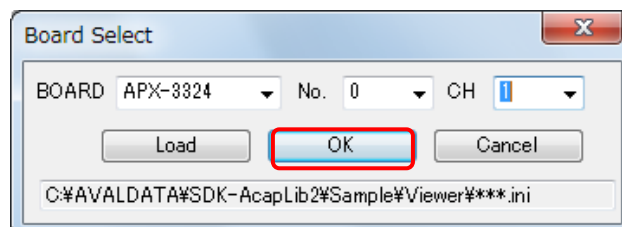


Figure 4-8 : 사용 보드등의 확인 창

[手順7] - 순서 7

초기화에 성공하면 Main Window 가 표시됩니다.
 빨강박스의 카메라 아이콘을 누르면 1 개의 영상이 취득됩니다.

※카메라는 공장출하 및 Free run 모드로 설정되어, 전원이 입력되어 있어야 합니다.

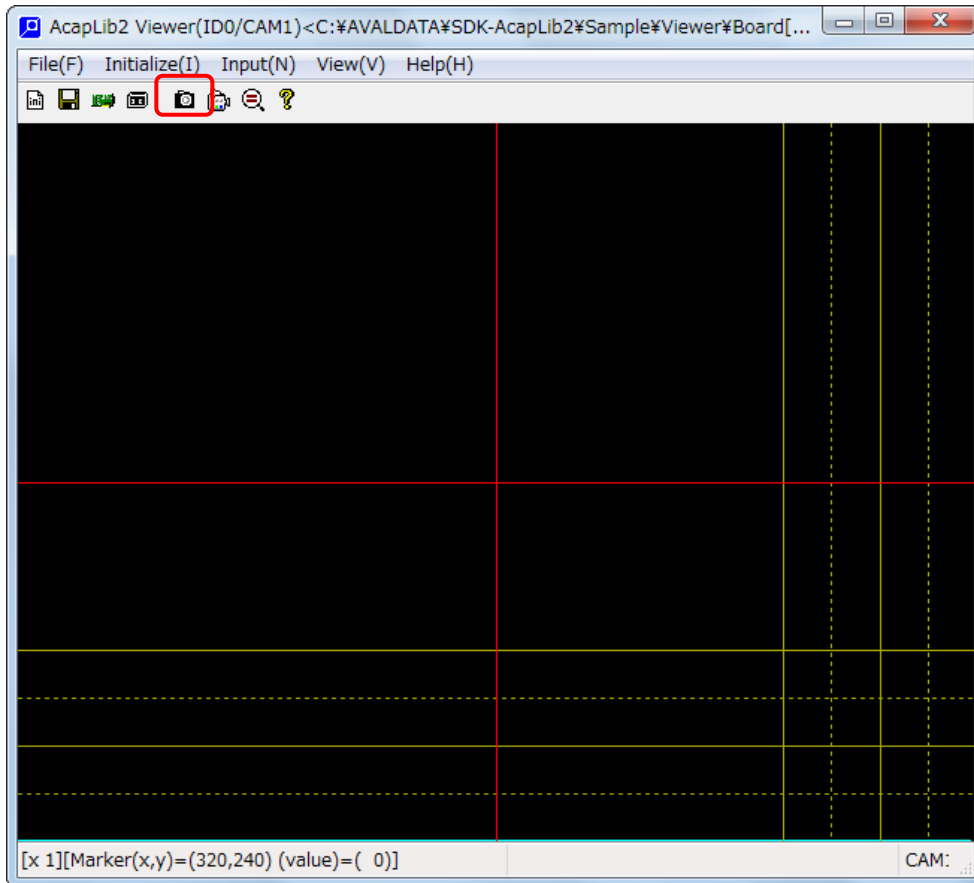


Figure 4-9 : Main Window

[手順8] - 순서 8

촬영에 성공하면 Main window 에 촬영영상, 영상입력 시간, Frame rate 가 표시 됩니다.

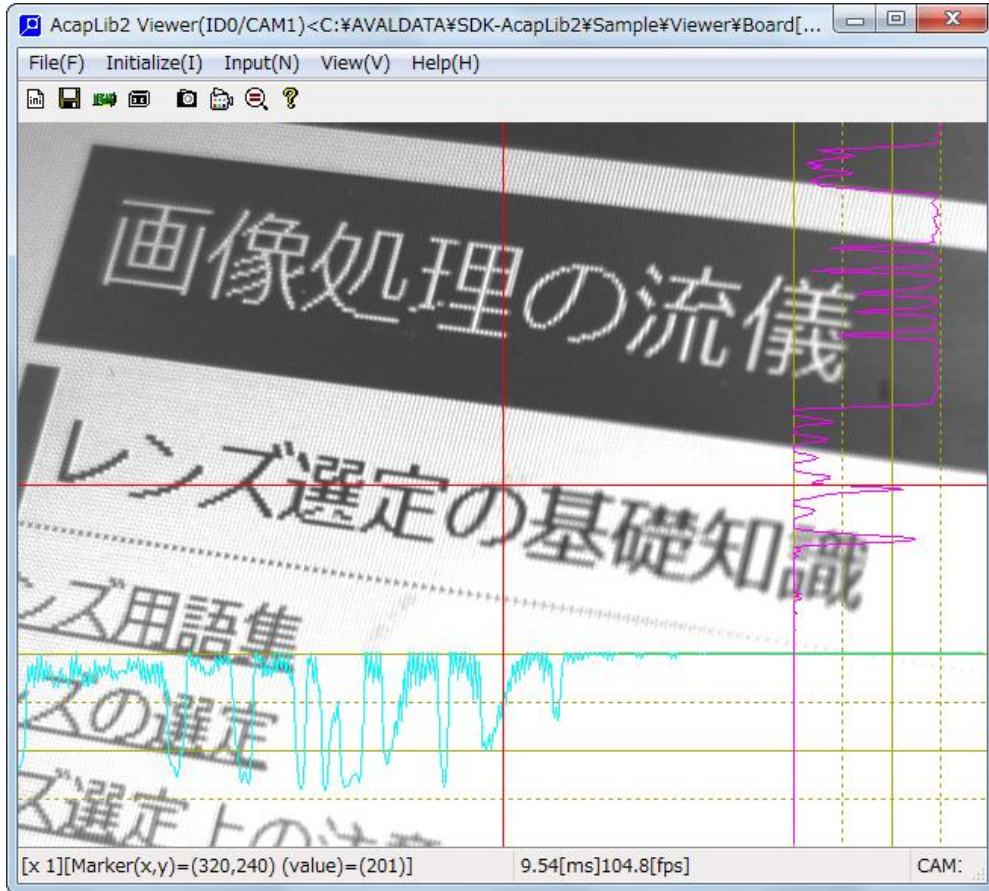


Figure 4-10 : 촬영 영상

또한, 촬영에 실패한 경우에는 아래와 같은 에러창과 함께 에러코드가 표시됩니다.
실패원인은 보드와 카메라의 잘못된 설정, 케이블이 연결되지 않음, 외부 트리거 및 엔코더가 입력되지 않은 등이 있습니다. 에러 코드에 대해서는 [第 8 章 의 에러코드](#)를 참조 해 주세요

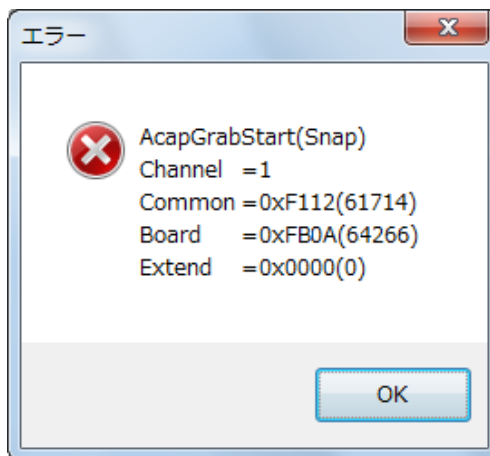


Figure 4-11 : 에러코드 창

第 5 章 프로그래밍 가이드

AcapLib2 를 사용하여 프로그래밍 순서를 설명합니다.

5.1 AcapLib2 Flow 차트

AcapLib2 를 사용하여 기본적인 호출 순서 및 처리순서를 설명 합니다.

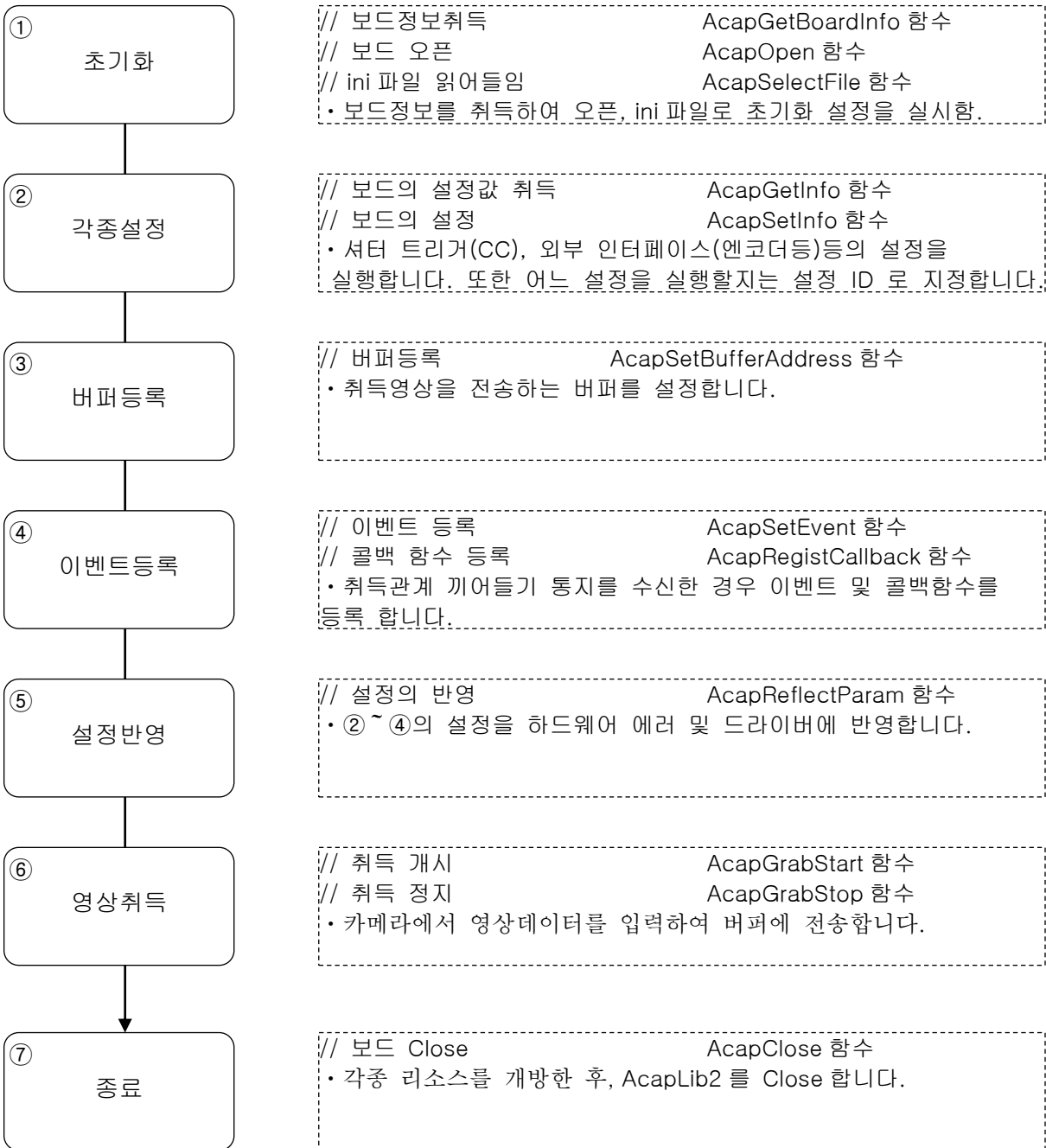


Figure 5-1 : Flow 차트

5.2 프로그래밍 예

한개 영상촬영, 연속촬영의 각각의 프로그래밍 예를 설명합니다.
화면표시, 다이얼로그등의 GUI는 사용하지 않습니다. 또한, 이에 대해 에러처리도 실행하지 않습니다.
프로그램은 아래의 조건을 전제로 합니다.

보드 : 복수 보드가 실장된 경우, Windows가 최초 검색한 당사 보드
ini 파일 : Default.ini(*)
ini 파일 버스 : 프로젝트 드라이버 파일 또는 실행 파일과 동일 단층

(*)Default.ini의 주요 설정은 아래와 같습니다.

입력사이즈 : 640×480
XY Delay : (x, y) = (0, 0)
처리 비트 : 8bit
외부 트리거, 랜덤 셔터 : 무효

5.2.1 한개 영상촬영(Snap)

한개 영상촬영(Snap)의 프로그래밍 예는 아래와 같습니다.

<샘플 코드>

```
#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;
    PCHAR pUserBuffer = NULL;
    ACAPBOARDINFO abi;

    // 보드 정보취득
    AcapGetBoardInfo( &abi );

    // 보드 오픈
    hDev = AcapOpen( abi.BOARDINDEX[0].pBoardName, abi.BOARDINDEX[0].nBoardID,
INPUT_CH );
    // ini 파일 읽어들이
    AcapSelectFile( hDev, INPUT_CH, "Default.ini", 0 );
```

```

// 카메라 종류 인식
AcapGetInfo( hDev, INPUT_CH, ACL_SCAN_SYSTEM, 0, &nCamera );
if( nCamera == CAM_LINE ){
    // 라인센서의 경우, 노광설정이 되어 있는지 확인함
    AcapGetInfo( hDev, INPUT_CH, ACL_EXP_CYCLE, 0, &nCycle );
    AcapGetInfo( hDev, INPUT_CH, ACL_EXPOSURE, 0, &nScan );
    if( ( nCycle == 0 ) || ( nScan == 0 ) ){
        printf( "노광설정이 부정\n" );
        return 0;
    }
}

// 여기에서 확보한 버퍼에 DMA 를 실행하도록 설정함
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( "입력을 중지하기 위해서는 아무키나 입력해 주세요\n" );

do{
    // 한개 영상촬영 개시
    AcapGrabStart( hDev, INPUT_CH, 1 );
    // 한개 영상촬영 중지
    AcapGrabStop( hDev, INPUT_CH );
    Sleep( 1 );

    // (x, y) = (100, 100)의 데이터를 Dump
    nData = pUserBuffer[100 + 100 * nXSize];
    printf( "취득 매수 : %4d 데이터(100, 100) : %3d\n", nLoop++, nData );
}while( !kbhit() );
printf( "입력을 중지하였습니다\n" );

// 보드 Close
AcapClose( hDev, INPUT_CH );

// 확보한 버퍼의 개방
if( pUserBuffer != NULL ){
    _aligned_free( pUserBuffer );
    pUserBuffer = NULL;
}
return 0;
}

```

5.2.2 연속 촬영(Grab)

연속 촬영(Grab)의 프로그래밍 예는 아래와 같습니다.

〈샘플 코드〉

```
#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;

    // 보드 정보취득
    AcapGetBoardInfo( &abi );
    // 보드 오픈
    hDev = AcapOpen( abi.BOARDINDEX[0].pBoardName, abi.BOARDINDEX[0].nBoardID, INPUT_CH );
    // ini 파일을 읽어들임
    AcapSelectFile( hDev, INPUT_CH, "Default.ini", 0 );

    // 카메라 종류 인식
    AcapGetInfo( hDev, INPUT_CH, ACL_SCAN_SYSTEM, 0, &nCamera );
    if( nCamera == CAM_LINE ){
        // 라인센서의 경우, 노광설정이 되어 있는지 확인함
        AcapGetInfo( hDev, INPUT_CH, ACL_EXP_CYCLE, 0, &nCycle );
        AcapGetInfo( hDev, INPUT_CH, ACL_EXPOSURE, 0, &nScan );
        if( ( nCycle == 0 ) || ( nScan == 0 ) ){
            printf( "노광설정이 부정\n" );
            return 0;
        }
    }

    // 메모리를 각 CH 마다 4 번씩 확보하도록 설정함
    AcapSetInfo( hDev, INPUT_CH, ACL_MEM_NUM, 0, 4 );
}
```

```

// CH1 은 여기서 확보한 버퍼에 DMA 를 실행하도록 설정
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 ));
    // 설정을 반영은 최후에 실행함으로 nMemNum 는 부수
    AcapSetBufferAddress( hDev, INPUT_CH, ACL_IMAGE_PTR, (nCount + 1)*(-1),
        &pUserBuffer[nCount][0] );
}
// 설정의 반영
AcapReflectParam( hDev, INPUT_CH );
// 콜백 함수의 등록
// CH1 에 「프레임 종료(ACL_INT_FRAMEEND)」를 등록
dwEventMask = ACL_INT_FRAMEEND;
AcapRegistCallback( hDev, INPUT_CH, dwEventMask, GrabCallbackFunc );

// 연속 촬영 개시
printf( "입력을 중지하기 위해서는 아무키나 입력해 주세요\n" );
AcapGrabStart( hDev, INPUT_CH, 0 );

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

// 연속 촬영 정지
AcapGrabStop( hDev, INPUT_CH );
printf( "촬영을 중지합니다\n" );

// 보드 Close
AcapClose( hDev, INPUT_CH );

// 확보한 버퍼의 함수
for( nCount = 0; nCount < 4; nCount++ ){
    if( pUserBuffer[nCount] != NULL ){
        _aligned_free( pUserBuffer[nCount] );
        pUserBuffer[nCount] = NULL;
    }
}

return 0;
}
// 콜백 함수
void CALLBACK GrabCallbackFunc( int nChannel, DWORD dwEvent, int nParam1, int nParam2 )
{
    int nIndex;
    int nData;

    switch( dwEvent ){
        case ACL_INT_FRAMEEND:
            // 프레임 종료시의 처리
            // (x, y) = (100, 100)의 데이터를 Dump
            nData = pUserBuffer[nParam2-1][100 + 100 * nXSize];
            printf( "취득 매수 : %4d 데이터(100, 100) : %3d\n", nParam1, nData );
            break;
    }
}
}

```

5.3 취득 샘플에 대해

AcapLib2에서는 각 개발언어의 취득 샘플, 타사 영상처리 라이브러리에 접속 샘플을 제공합니다.

5.3.1 SimpleApplication

AcapLib2를 사용한 각 언어의 영상취득 샘플입니다. 아래의 폴더에 저장되어 있습니다.

[AcapLib2를 사용하여 영상취득 샘플]

- [SDK-AcapLib2]⇒[AVALDATA]⇒[SDK-AcapLib2]⇒[Sample]⇒[SimpleApplication]

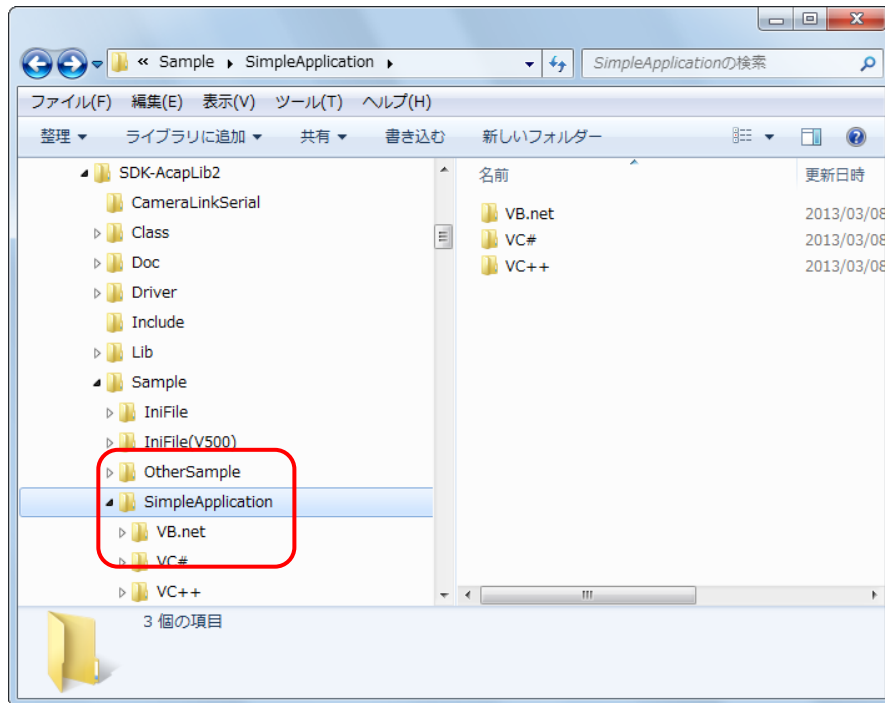


Figure 5-2 : SimpleApplication

[SimpleApplication] 폴더	샘플 내용
VB.net	AcapLib2를 이용하여 영상취득을 실행 VB.net의 샘플
VC#	AcapLib2을 이용하여 영상취득을 실행 VC#의 샘플
VC++	AcapLib2을 이용하여 영상취득 실행 VC++의 샘플 Free run, 셔터, 각종 트리거 등이 있습니다.

Table 5-1 : SimpleApplication

5.3.2 OtherSample

AcapLib2 와 타사 라이브러리를 이용한 영상취득 샘플입니다. 아래 폴더에 저장되어 있습니다.

[타사 라이브러리를 사용하여 영상취득 샘플 등]

- [SDK-AcapLib2] ⇒ [AVALDATA] ⇒ [SDK-AcapLib2] ⇒ [Sample] ⇒ [OtherSample]

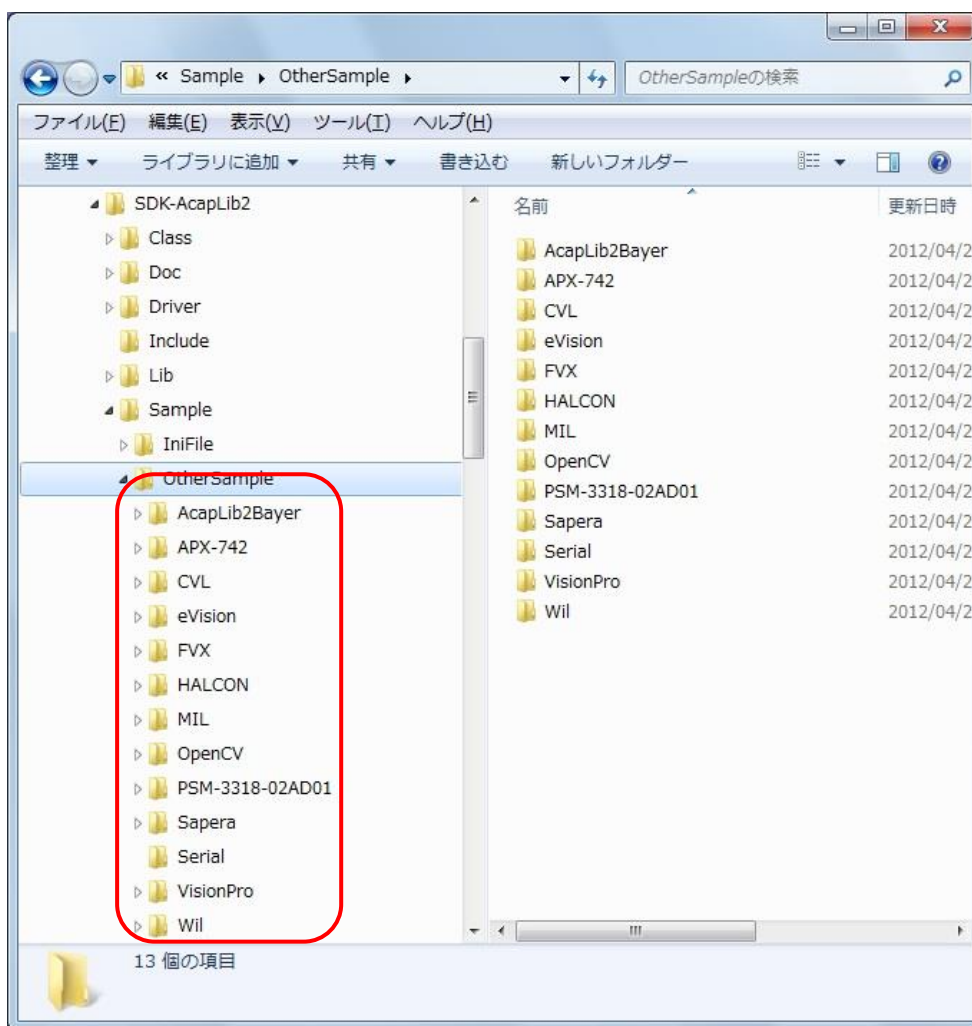


Figure 5-3 : OtherSample

각 폴더에서 아래 샘플이 있습니다.

또한, 타사 라이브러리를 사용한 영상취득 샘플에 대해서는 아래 폴더의 각 샘플 매뉴얼이 있으니, 참조 바랍니다.

- [SDK-AcapLib2]⇒[AVALDATA]⇒[SDK-AcapLib2]⇒[Doc]⇒[6_Technical]

[OtherSample] 폴더	샘플 내용
AcapLib2Bayer	AcapLib2 로 영상취득, 베이어 변환 VC++ 샘플
APX-742	「APX-742」로 영상배분을 실행하기 위한 VC++ 샘플
CVL	AcapLib2 로 영상취득 CVL 로 영상 처리하는 샘플
eVision	AcapLib2 로 영상취득 eVision 으로 영상 처리하는 VC++ 샘플
FVX	AcapLib2 로 영상취득 FVX 로 영상 처리하는 VC++ 샘플
HALCON	AcapLib2 로 영상취득 HALCON 으로 영상 처리하는 VC++ 샘플
MIL	AcapLib2 에서 영상취득 MIL 로 영상 처리하는 VC++ 샘플
OpenCV	AcapLib2 에서 영상취득 OpenCV 에서 영상 처리하는 VC++, CLI 샘플
PSM-3318-02AD01	「PSM-3318-02」에서 영상배분을 실행하기 위한 VC++ 샘플 ※본 제품에서는 사용할 수 없습니다.
Sapera	AcapLib2 로 영상취득 Sapera 에서 영상처리하는 VC++ 샘플
Serial	카메라와 시리얼 통신을 실행하는 VC++ 샘플
VisionPro	AcapLib2 로 영상취득 VisionPro 로 영상처리하는 VB.net 샘플
Wil	AcapLib2 로 영상취득 WIL 에서 영상처리하는 VC++ 샘플

※AcapLib2Bayer, PSM-3318-02AD01 샘플 이외는 64bitOS 대응하지 않습니다.

Table 5-2 : OtherSample

第 6 章 기능

본 제품의 기능에 대해 설명합니다.
또한, 각 기능에 있어서 AcapLib2 를 사용한 설정 예를 설명합니다.

6.1 CameraLink Configuration

본 제품은 CameraLink Base ~ Full Configuration 에 대응하는 카메라의 영상입력이 가능합니다.
26PinSDR 커넥터를 탑재하고 있어 1 대의 보드에 2 대의 카메라가 접속 가능합니다.

본 제품이 입력가능한 영상 포맷은 아래와 같습니다.

- 8bit x1tap, x2tap, x3tap, x4tap, x8tap, x10tap Monochrome
- 10 ~ 12bit x1tap, x2tap, x4tap Monochrome
- 14 ~ 16bit x1tap Monochrome
- 24bitRGB x1tap, Color
- 30bitRGB x1tap Color
- 32bitRGB x1tap x2tap Color
- 10bit x8tap Monochrome

복수 탭 출력을 가진 카메라 영상의 화소 재배열 기능도 있습니다.

클럭은 20MHz ~ 85MHz 까지 대응하고 있습니다. (N.S 사 DS90CR288 를 Receiver 로 사용하고 있습니다.)
본 제품으로 영상취득 가능한 CameraLink 동기 신호 타이밍은 아래와 같습니다.

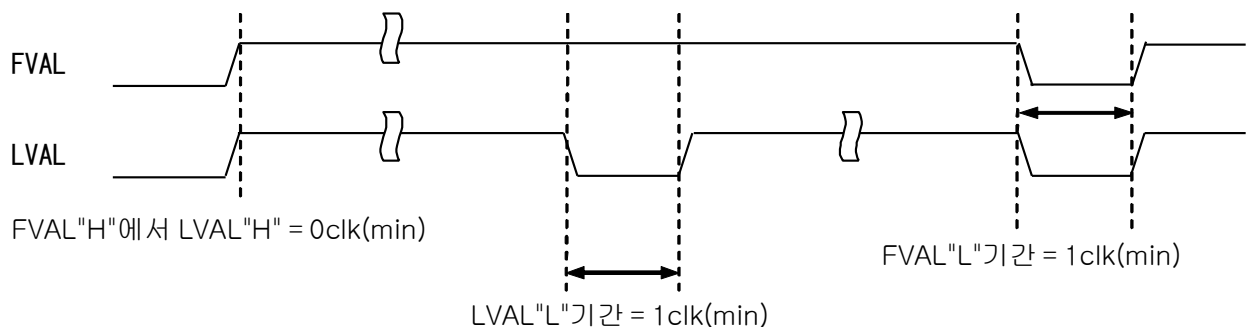


Figure 6-1 : CameraLink 동기 신호

6.1.1 2TAP 출력시의 설정

■하드웨어 설정 예

아래에 2TAP 출력 카메라에 있어서, 재 배열을 실행하는 경우의 샘플 코드를 설명합니다.
또한, 카메라에서 아래의 순서로 데이터가 출력되는 것을 전제로 합니다.

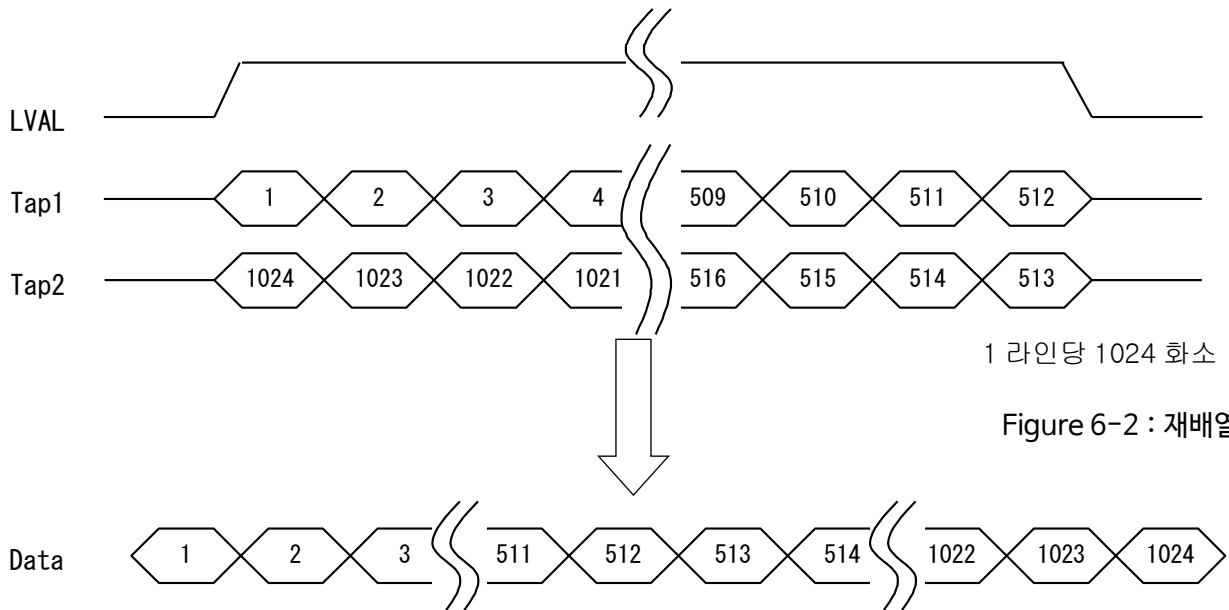


Figure 6-2 : 재배열전

Figure 6-3 : 재배열후

〈샘플 코드〉

```

//Tap 수 설정
AcapSetInfo(hDev, INPUT_CH, ACL_TAP_NUM, -1, 2);
//재배열 방법을 모든 Tap 독립적으로 설정
AcapSetInfo(hDev, INPUT_CH, ACL_TAP_ARRANGE, -1, 1);
//재배열 방향을「→←」로 설정
AcapSetInfo(hDev, INPUT_CH, ACL_TAP_DIRECTION, -1, 0);
AcapSetInfo(hDev, INPUT_CH, ACL_TAP_DIRECTION, -2, 1);
//라인 Reverse 기능 무효
AcapSetInfo(hDev, INPUT_CH, ACL_LINE_REVERSE, -1, 0);
//LVAL의 X 방향 지연량을 0으로 설정
AcapSetInfo(hDev, INPUT_CH, ACL_LVAL_DELAY, -1, 0);
//카메라가 1 라인으로 써 출력하는 총 화소수를 설정
AcapSetInfo(hDev, INPUT_CH, ACL_ARRANGE_XSIZE, -1, 1024);
//설정의 반영
AcapReflectParam(hDev, INPUT_CH);
    
```

6.1.2 3TAP 출력시의 설정

■소프트웨어 설정 예

아래에 3TAP 출력 카메라에 있어서 재배열을 실행할 경우의 샘플코드를 설명합니다.
또한 카메라에서 아래 순서로 데이터가 출력되는 것을 전제로 합니다.

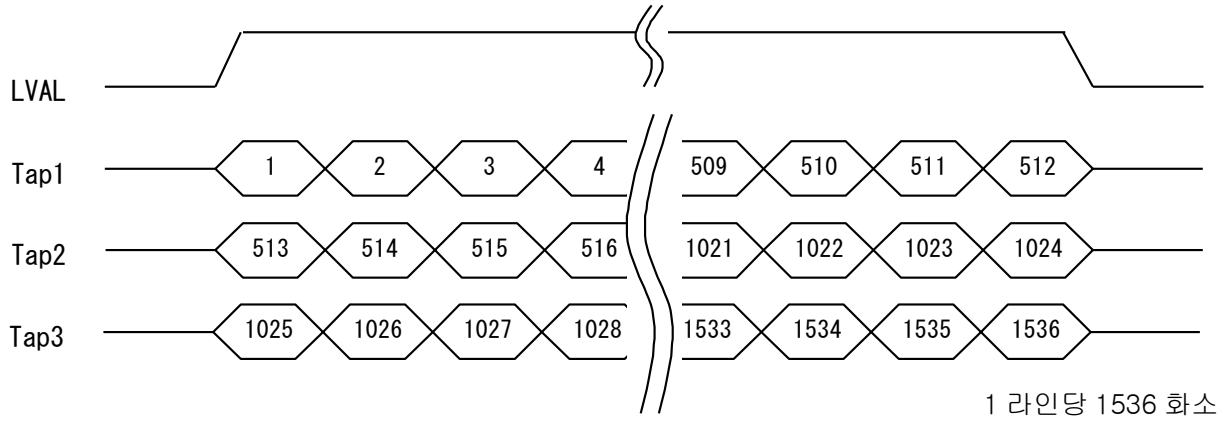


Figure 6-4 : 재배열전

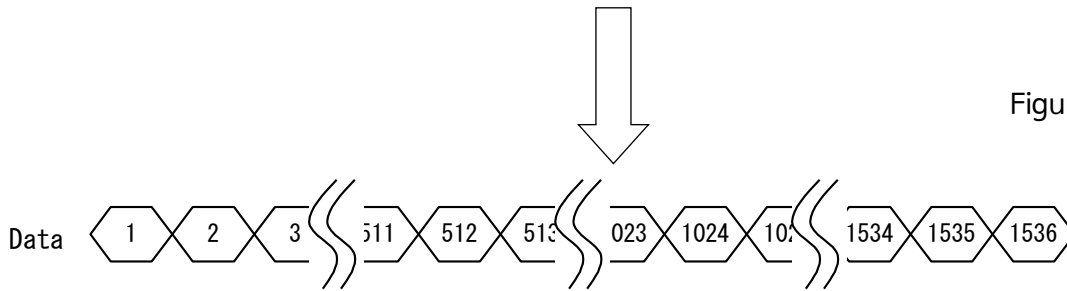


Figure 6-5 : 재배열후

<샘플 코드>

```

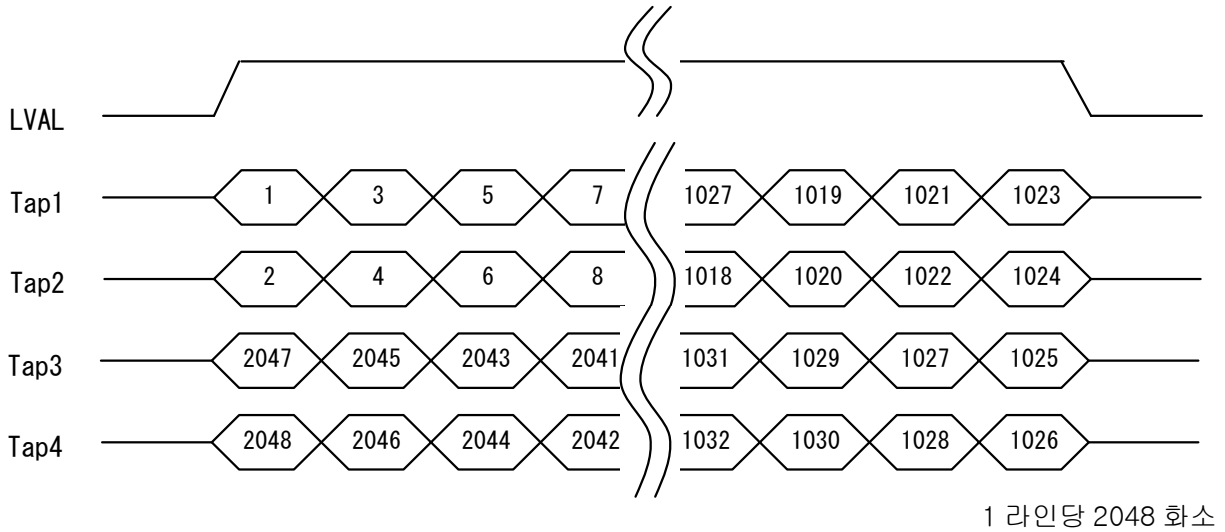
//Tap 수 설정
AcapSetInfo(hDev, INPUT_CH, ACL_TAP_NUM, -1, 3);
//재배열 방법을 모든 Tap 독립적으로 설정
AcapSetInfo(hDev, INPUT_CH, ACL_TAP_ARRANGE, -1, 1);
//재배열 방향을「→→」로 설정
AcapSetInfo(hDev, INPUT_CH, ACL_TAP_DIRECTION, -1, 0);
AcapSetInfo(hDev, INPUT_CH, ACL_TAP_DIRECTION, -2, 0);
AcapSetInfo(hDev, INPUT_CH, ACL_TAP_DIRECTION, -3, 0);
//라일 Reverse 기능 무효
AcapSetInfo(hDev, INPUT_CH, ACL_LINE_REVERSE, -1, 0);
//LVAL 의 X 방향 지연량을 0 으로 설정
AcapSetInfo(hDev, INPUT_CH, ACL_LVAL_DELAY, -1, 0);
//카메라가 1 라인으로써 출력하는 총 화소수를 설정
AcapSetInfo(hDev, INPUT_CH, ACL_ARRANGE_XSIZE, -1, 1536);
//설정 반영
AcapReflectParam(hDev, INPUT_CH);
    
```

6.1.3 4TAP 출력시의 설정

■ 소프트웨어 설정 예

아래에 4TAP 출력 카메라에 있어서 재배열을 실행할 경우의 샘플코드를 설명합니다.

또한 카메라에서 아래 순서로 데이터가 출력되는 것을 전제로 합니다.



1 라인당 2048 화소

Figure 6-6 : 재배열 전

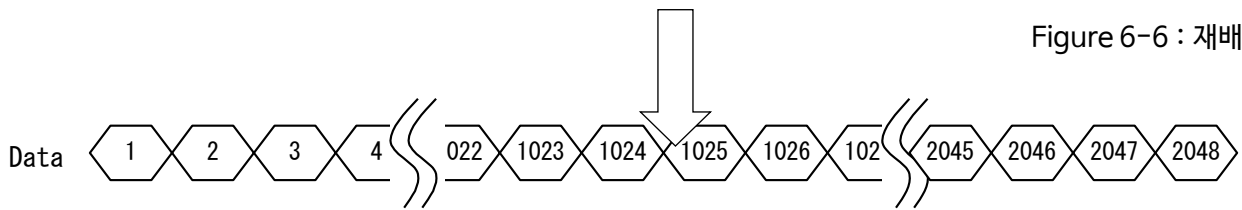


Figure 6-7: 재배열 후

〈샘플 코드〉

```

//Tap 수 설정
AcapSetInfo(hDev, INPUT_CH, ACL_TAP_NUM, -1, 4 );
//재배열 방법을 2Tap 동시에 설정
AcapSetInfo(hDev, INPUT_CH, ACL_TAP_ARRANGE, -1, 2 );
//재배열 방향을「→→」로 설정
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, 1 );
AcapSetInfo(hDev, INPUT_CH, ACL_TAP_DIRECTION, -4, 1 );
//라일 Reverse 기능 무효
AcapSetInfo(hDev, INPUT_CH, ACL_LINE_REVERSE, -1, 0 );
//LVAL 의 X 방향 지연량을 0 으로 설정
AcapSetInfo(hDev, INPUT_CH, ACL_LVAL_DELAY, -1, 0 );
//카메라가 1 라인으로써 출력하는 총 화소수를 설정
AcapSetInfo(hDev, INPUT_CH, ACL_ARRANGE_XSIZE, -1, 2048 );
//설정 반영
AcapReflectParam(hDev, INPUT_CH);
    
```

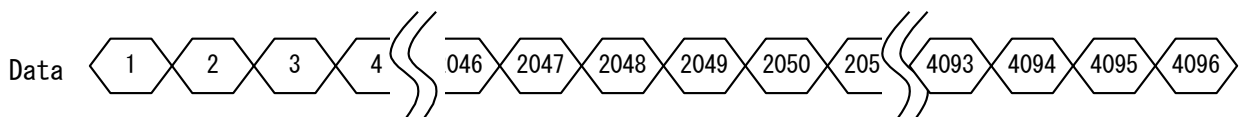
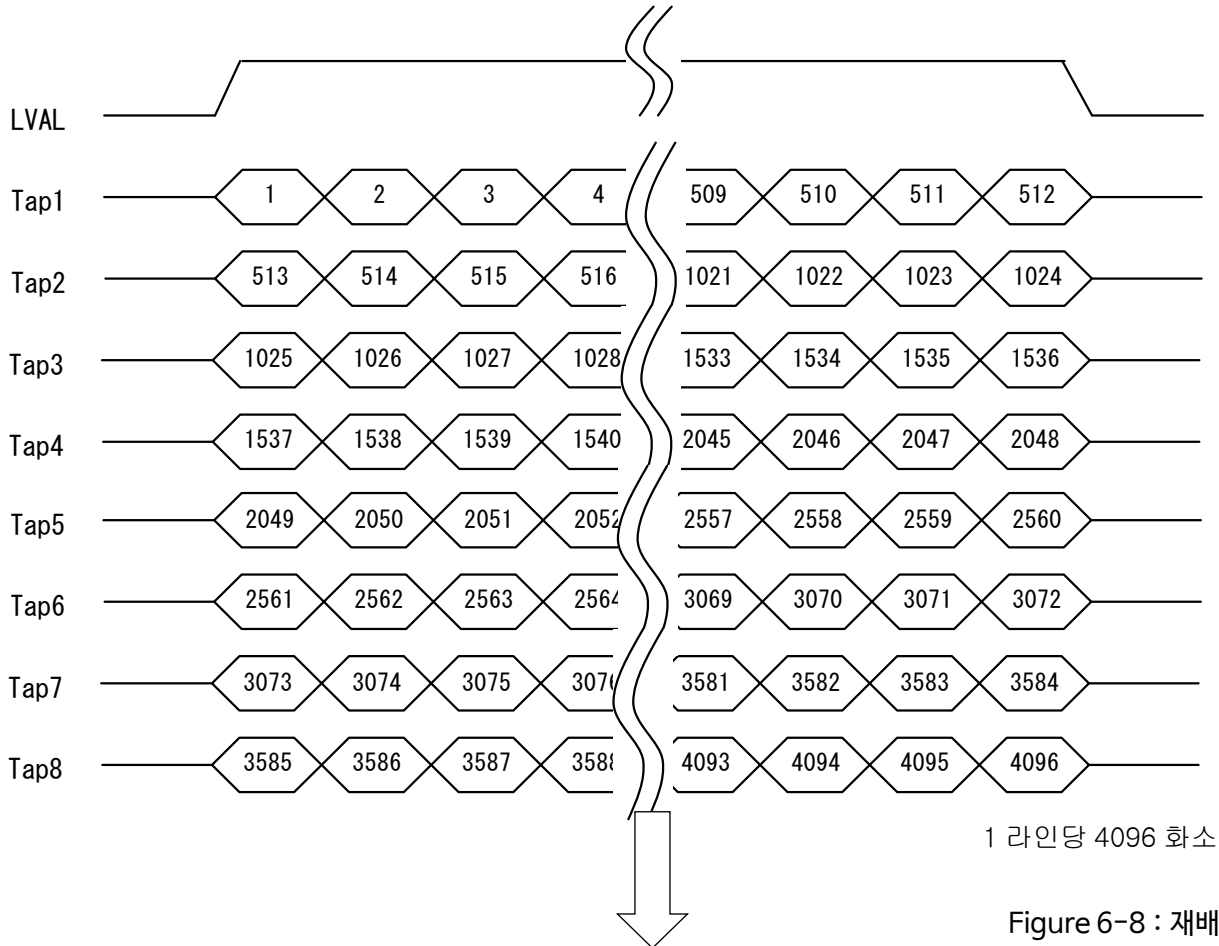


6.1.4 8TAP 출력시의 설정

■ 소프트웨어 설정 예

아래에 8TAP 출력 카메라에 있어서 재배열을 실행할 경우의 샘플코드를 설명합니다.

또한 카메라에서 아래 순서로 데이터가 출력되는 것을 전제로 합니다.



〈샘플 코드〉

```

//Tap 수 설정
AcapSetInfo(hDev, INPUT_CH, ACL_TAP_NUM, -1, 8 );
//재배열 방법을 모든 Tap 독립적으로 설정
AcapSetInfo(hDev, INPUT_CH, ACL_TAP_ARRANGE, -1, 1 );
//재배열 방향을「→→」로 설정
for(i = 1; i <= 8; i++){
    AcapSetInfo(hDev, INPUT_CH, ACL_TAP_DIRECTION, -i, 0 );
}
//라일 Reverse 기능 무효
AcapSetInfo(hDev, INPUT_CH, ACL_LINE_REVERSE, -1, 0 );
//LVAL 의 X 방향 지연량을 0 으로 설정
AcapSetInfo(hDev, INPUT_CH, ACL_LVAL_DELAY, -1, 0 );
//카메라가 1 라인으로써 출력하는 총 화소수를 설정
AcapSetInfo(hDev, INPUT_CH, ACL_ARRANGE_XSIZE, -1, 4096 );
//설정 반영
AcapReflectParam(hDev, INPUT_CH);

```

6.2 Power Over CameraLink

본 제품은 카메라에 PoCL 규격으로 설정된 +12V 전원 공급이 가능합니다.

전원 ON, OFF 제어는 소프트웨어에서 카메라마다 독립적으로 제어 가능합니다.

전원입력후, 설정시간내에 CameraLink 클럭을 검출하지 못한 경우, 또는 통전중에 어떠한 이유로 인해 CameraLink 클럭이 정지하여, 그 후 설정시간내에 클럭이 복구하지 않은 경우에는 문제를 통지하고 전원 공급을 자동으로 중단합니다.

또한, 드라이버 기동시에 CameraLink 클럭이 검출되지 않을 시에는 전원공급을 개시합니다.

전원 ON 시에 시퀀스는 아래와 같습니다.

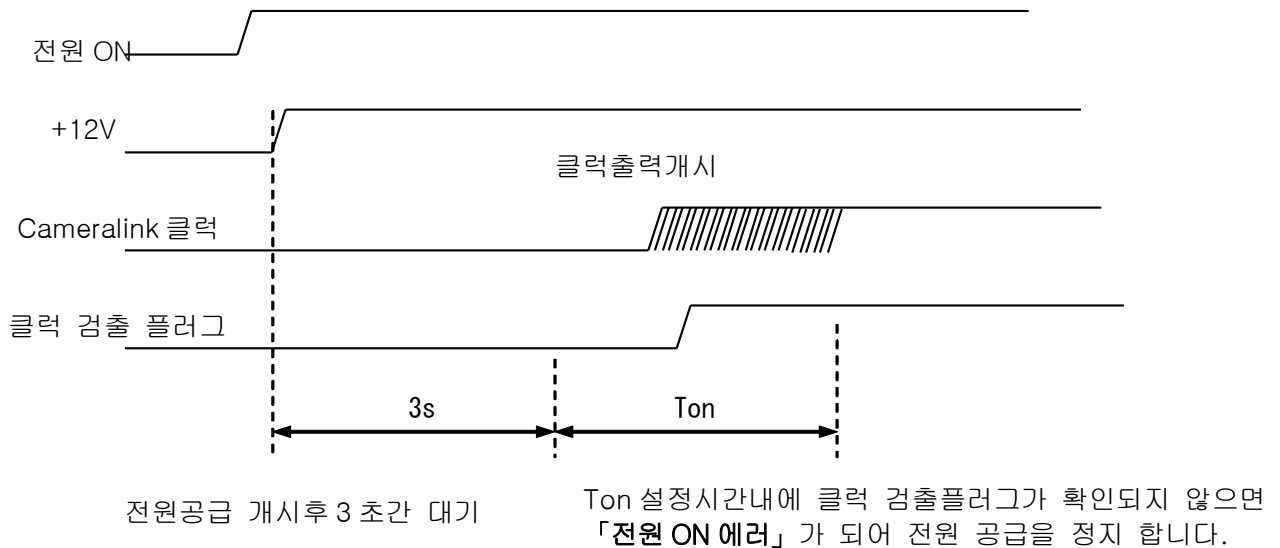
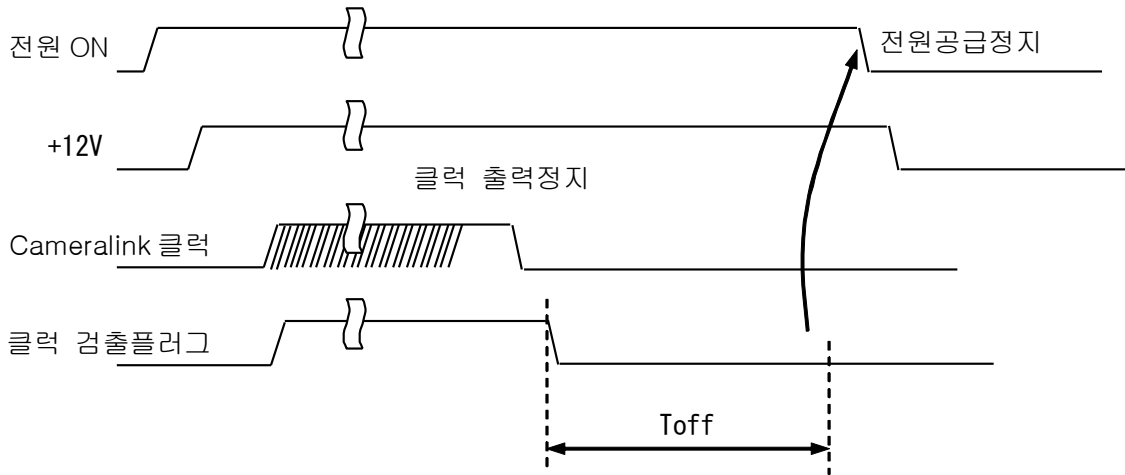


Figure 6-10 : 전원 ON 시의 동작

통전중의 이상 시퀀스는 아래와 같습니다.



Toff 설정시간내에 클럭 검출플러그가 확인되지 않으면 「전원 OFF 에러」가 되어 전원 공급을 정지합니다.

Figure 6-11 : 이상시 동작

■ 소프트웨어 설정 예

아래에 카메라 클럭을 확인하기까지의 대기시간을 3000ms, 1ch 측의 카메라에 전원 공급을 실행할 경우에 샘플 코드를 설명합니다.

또한, 보드 본체의 점퍼(각 채널에 대응한 CN)의 설정이 ON 인 경우에만 전원공급 제어가 가능합니다.

점퍼에 대한 상세내용은 [\[3.1.3 CN11 ~ CN14설정\]](#)을 참조해 주세요

<샘플 코드>

```
//카메라 전원 에러상태를 클리어 합니다.
AcapSetInfo(hDev, INPUT_CH, ACL_POWER_STATE, 0, 1);
//카메라에 전원공급을 실시합니다.
AcapSetInfo(hDev, INPUT_CH, ACL_POWER_SUPPLY, 3000, 1);
```


6.3 카메라 노출제어 신호 CC

본 제품은 카메라의 노출제어 신호 CC (Camera Control) 을 출력 할 수 있습니다. 노출시간, 노출주기를 소프트웨어로 설정하여 동작하는 내부생성방식 또는 외부 트리거와 엔코더 등 외부신호입력을 원인으로써 동작하는 외부 생성방식 중 선택이 가능합니다.

내부생성방식 및 외부생성방식과 같이 노광시간/노광주기의 설정단위는 100nS 입니다. 내부 생성방식의 타이밍은 아래와 같습니다.

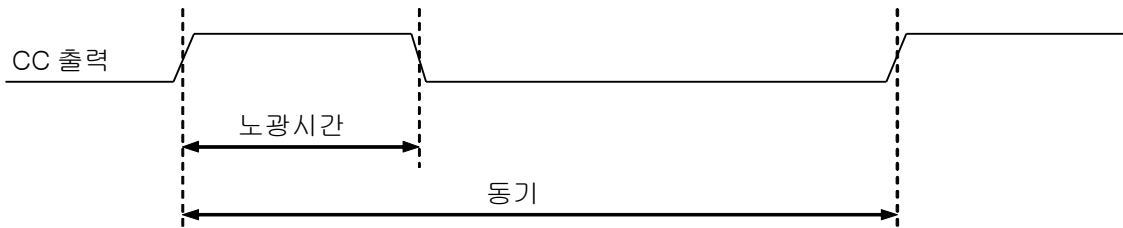


Figure 6-12 : 내부생성 CC

외부생성방식의 타이밍은 아래와 같습니다.

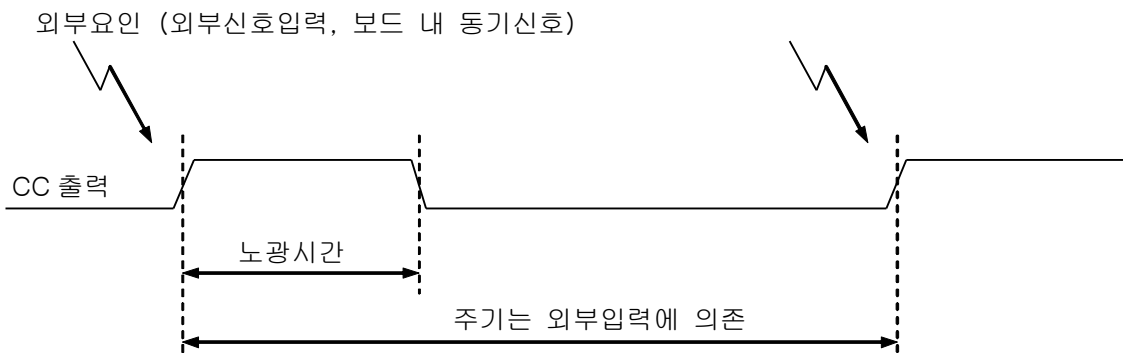


Figure 6-13 : 외부생성 CC

외부신호 입력에서 CC 출력 개시까지의 지연시간은 대략적으로 아래와 같습니다.

외부입력 IF 의 종류	CC 출력 개시까지의 지연시간
TTL 입력	1uS (Max)
RS422 입력	1uS (Max)

Table 6-1 : 외부신호입력의 지연시간

이 표는 노이즈제거 시간설정없이 (디지털 필터 없음) 에서의 값입니다. 노이즈제거 기능을 유효로 한 경우에는 설정한 시간만큼 지연이 추가됩니다.

CC 신호는 각 채널 독립적으로 출력 가능합니다.

또한, 별도 채널의 CC를 외부요인으로써 동작시키는 마스터, 슬레이브 구성도 가능합니다.

CC 신호 출력회로의 조합에 대한 예는 아래와 같습니다.

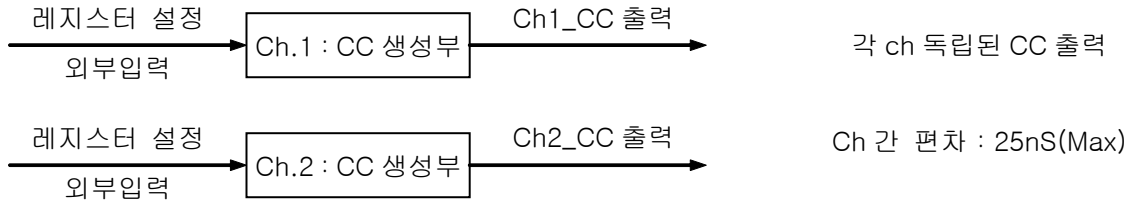


Figure 6-14 : 전 ch 독립 CC 출력

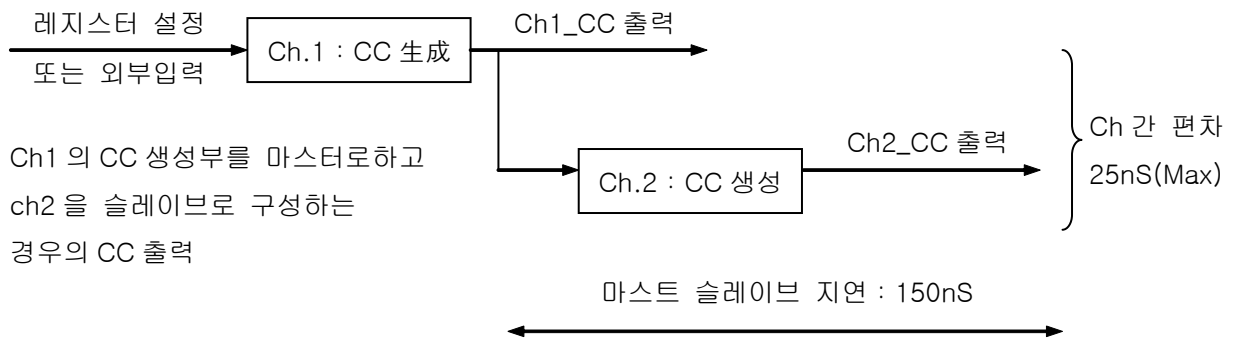


Figure 6-10 : 마스터, 슬레이브 구성 CC 출력

■ 소프트웨어 설정예

아래에 CC출력 주기를100마이크로, CC출력폭을 30마이크로로 설정한 경우의 샘플을 설명합니다.

※CC 신호의 출력폭은 출력주기보다 작은 값으로 설정해 주세요

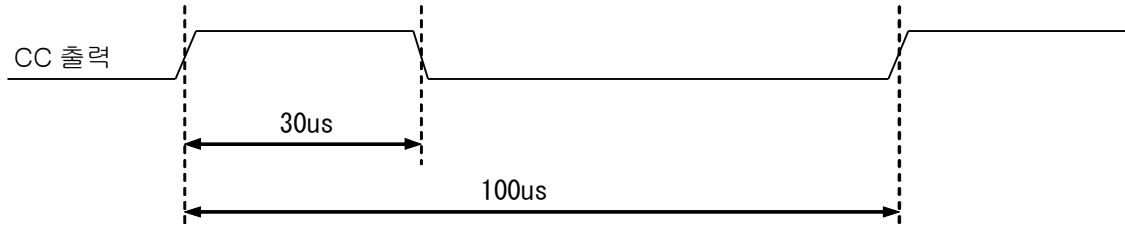


Figure 6-16 : CC 신호 출력 타이밍

<샘플 코드>

```
//CC신호 유효
AcapSetInfo( hDev, INPUT_CH, ACL_EXP_EN, -1, 1);
//CC신호를 정논리로 설정
AcapSetInfo( hDev, INPUT_CH, ACL_EXP_POL, -1, 1);
//CC출력폭을30us로 설정
AcapSetInfo( hDev, INPUT_CH, ACL_EXPOSURE, -1, 30);
//CC출력 주기를 100us로 설정
AcapSetInfo( hDev, INPUT_CH, ACL_EXP_CYCLE, -1, 100);
//설정 반영
AcapReflectParam(hDev, INPUT_CH);
```

6.4 시리얼 인터페이스

본 제품은 카메라 통신용 시리얼 인터페이스를 4ch 탑재하고 있습니다.

- SerTC 보드에서 카메라 방향의 신호 (송신 데이터)
- SerTFG 카메라에서 보드 방향의 신호 (수신 데이터)

AVALDATA 제품 가상 COM 드라이브를 사용하는 것으로 UART 를 COM 포트에 할당하여, 아래 2 개의 방법으로 카메라와 통신하는 것이 가능합니다.

- 카메라 메이커가 제공하는 통신 소프트웨어를 이용한 통신
- API(AcapLib2)를 이용한 통신
-

본 제품과 카메라간은 아래와 같은 사양으로 통신 가능합니다.

통신속도 : 9600bps (디폴트) ~ 레지스터 설정으로 변경가능
 Start bit : 1bit
 Data bit : 8bit
 Stop bit : 1bit
 Parity bit : 없음
 Flow 제어 : 없음

가상 COM 에 대해서는「SDK-AcapLib2 가상 COM 툴 사용자 메뉴얼」을 참조해 주세요

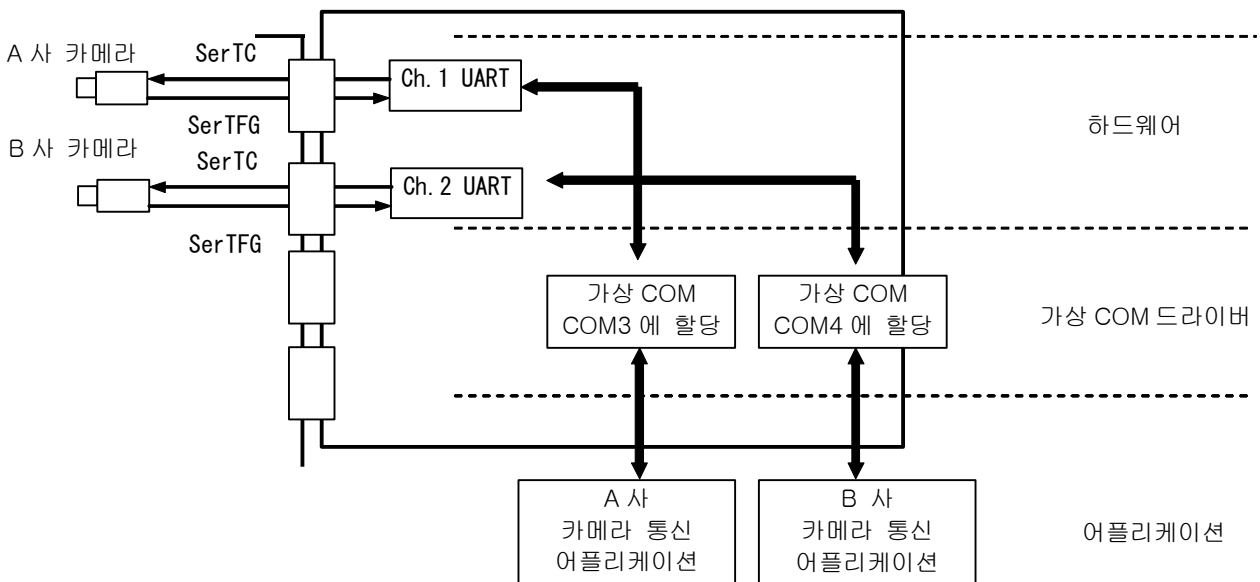


Figure 6-17 : 시리얼 인터페이스

■소프트웨어 설정예

아래에 명령어「AVAL」을 송신할 경우의 샘플 코드를 설명합니다.

또한 문자 코드, Baud rate, 종단문자는 아래와 같습니다.

통신속도 : 9600bps

종단문자 : CR(Wr)

문자코드 : ASCII

〈샘플 코드〉

```
//시리얼 포트를 오픈함
AcapSerialOpen( hDev, INPUT_CH );
//시리얼 파라미터 설정
AcapSerialSetParameter( hDev, INPUT_CH, 9600, 8, 0, 0, 0);
//커맨드 송신(입력)
AcapSerialWrite( hDev, INPUT_CH, TRUE, "AVALWr", NULL, NULL);
//시리얼 포트를 Close 함
AcapSerialClose( hDev, INPUT_CH );
```

6.5 외부 신호 출력

본 제품은 확장 커넥터 (CN16) 를 사용하여, 외부기기와의 인터페이스가 가능합니다.
아래에 신호명과 인터페이스를 표시합니다.

신호명	기능
OPTOIN_1 OPTOIN_2 (ch 대응)	~ 포토커플러 : 외부 트리거 입력단자입니다. 아래 3 종류의 용도로 사용가능합니다. · 영상입력 개시 트리거 · CC 발생 트리거 · 엔코더 기동 트리거 상하 엷지를 트리거로써 인식합니다.
TTL_IN1 ~ TTL_IN2 (ch 대응)	TTL 레벨 : 외부 트리거 입력단자입니다. 아래 3 종류의 용도로 사용가능합니다. · 영상 입력 개시 트리거 · CC 발생 트리거 · 엔코더 기동 트리거 상하 엷지를 트리거로써 인식합니다.
RS422_IN1 ±	RS422 레벨 : 아래 용도로 사용가능합니다. · 엔코더 A 상
RS422_IN2 ±	RS422 레벨 : 아래 용도로 사용가능합니다. · 엔코더 B 상
RS422_IN3 ±	RS422 레벨 : 아래 용도로 사용가능합니다. · 엔코더 Z 상
RS422_IN4 ±	RS422 레벨 : 아래 트리거 용도로 사용 가능합니다. · ch1 ~ ch4 공통 영상입력개시 트리거
STB_OUT1 STB_OUT2 (ch 대응)	~ 오픈 컬렉터/TTL 레벨 : 스트로브 출력단자 입니다.
GP_OUT1 GP_OUT2 (ch 대응)	~ 오픈 컬렉터/TTL 레벨 : 범용출력단자 입니다. 아래 2 종류의 용도로 사용 가능합니다. · 레지스터에 의한 논리출력 · 영상입력중 신호
GP_OUT3 GP_OUT8	~ 오픈 컬렉터/TTL 레벨 : 범용출력단자 입니다. 레지스터에 의한 논리출력만 가능합니다.

Table 6-1 : 외부신호 입력 I/F

외부신호 입력회로는 아래와 같습니다.

1. TTL 레벨 입력단자

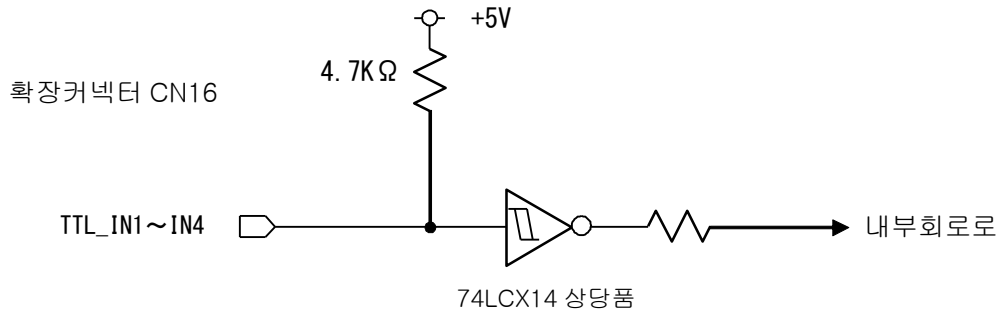


Figure 6-18 : TTL 입력단자

TTL 단자에는 +5V 이상의 전원을 인가하지 마십시오

TTL 신호는 전송길이와 주변노이즈의 영향을 받기 쉬움으로 노이즈가 발생하는 환경에서는 차동신호 (RS422)의 사용을 추천합니다.

외부트리거 사용시에는,
최저 25uS 시간 이상의 안정된 "L"레벨을 입력해 주세요

■ 소프트웨어 설정 예

아래에 TTL신호를 사용할 경우의 샘플 코드를 설명합니다.

또한, 외부 트리거모드는 연속외부 트리거모드 (외부 트리거를 검출하는 것으로 CC 신호 출력)로 합니다.

<샘플 코드>

```
//TTL신호사용
AcapSetInfo( hDev, INPUT_CH, ACL_EXT_EN, -1, 1);
//연속 외부 트리거 모드
AcapSetInfo( hDev, INPUT_CH, ACL_EXT_MODE, -1, 0);
//설정 반영
AcapReflectParam(hDev, INPUT_CH);
```

2.RS422 레벨 입력단자

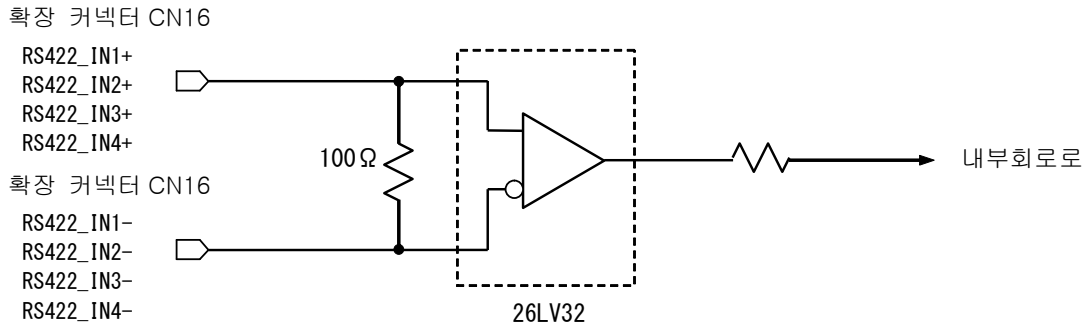


Figure 6-19 : RS422 입력단자

외부 트리거 사용시는
 최저 25uS 의 시간이상의 안정된"L"레벨을 입력해주세요

사용하지 않는 RS422 입력은 입력레벨이 부정이 됩니다.
 부정상태에서 외부 트리거를 유효로하면, 예기치 않는 타이밍에서 트리거가 발생
 해 버리는 것이 됩니다. RS422 입력을 사용하지 않을 때에는 반드시 외부 트리거를
 무효로 설정해 주세요

■소프트웨어 설정 예

아래에 RS-422신호를 사용하는 경우의 샘플코드를 설명합니다.

또한, 외부 트리거모드는 연속외부 트리거모드(외부 트리거 1 개에 CC 를 1 개 발생)합니다.

<샘플 코드>

```
//RS-422신호사용(INPUT_CH의 값은 관계없고 ch1~ch4동시 출력이 됩니다.)
AcapSetInfo( hDev, INPUT_CH, ACL_EXT_EN, -1, 2);
//연속 외부 트리거모드
AcapSetInfo( hDev, INPUT_CH, ACL_EXT_MODE, -1, 0);
//설정 반영
AcapReflectParam(hDev, INPUT_CH);
```


3. 포토커플러 입력단자

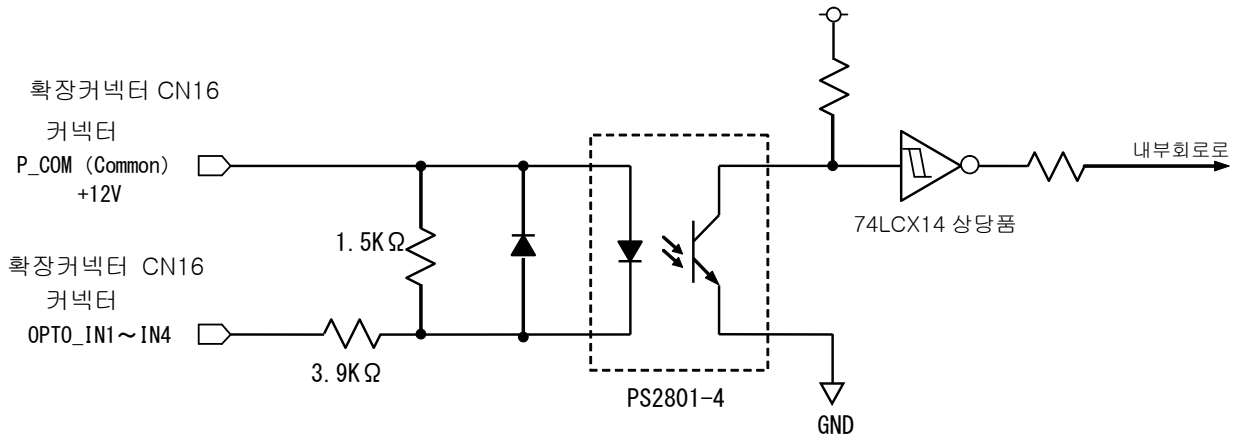


Figure 6-20 : 포토 입력단자

포토 입력부의 사양은 아래와 같습니다.

- 입력저항 : 3.9KΩ
- 입력전압 : DC+12V
- 입력레벨 : ON (단락) → "L", OFF (개방) → "H"
- 입력전류 : 3mA
- 입력응답시간 : Ton=8uS (TYP) , Toff=45uS (TYP)

외부 트리거 사용시는
최저 25uS 의 기간 이내의 안정된 "L"레벨을 입력해 주세요

■ 소프트웨어 설정 예

```
//포토 입력을 외부 트리거로써 사용
AcapSetInfo( hDev, INPUT_CH, ACL_EXT_EN, -1, 4);
//연속 외부트리거 모드
AcapSetInfo( hDev, INPUT_CH, ACL_EXT_MODE, -1, 0);
//설정반영
AcapReflectParam(hDev, INPUT_CH);
```

4.TTL/오픈 컬렉터 단자입력

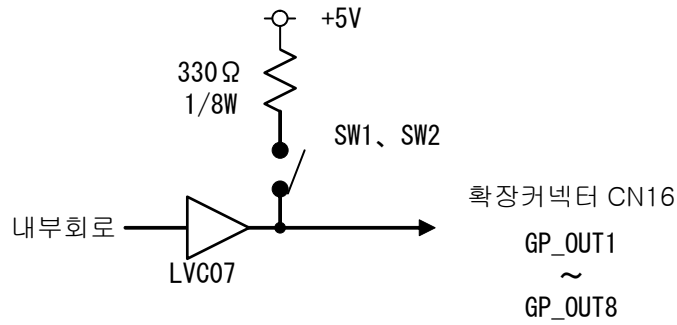


Figure 6-21 : 오픈 컬렉터 출력단자

TTL/오픈 컬렉터 출력단자에는 +5V 이상의 전압을 가하지 마십시오

■ 소프트웨어 설정예

아래에 GP_OUT의 범용출력을Low출력, 또는 High출력으로써 사용할 경우 샘플코드를 설명합니다.

〈샘플코드〉

```

1.현재 상태를 취득
DWORD GP_OUT_level;
AcapGetInfo(hDev, ACL_GPOUT_POL, 0, (int*)&GP_OUT_level);

2.GP_OUT[5]를 High 로 함
GP_OUT_level |= 0x10; // Bit4 가 GP_OUT[5]
AcapSetInfo(hDev, ACL_GPOUT_POL, 0, (int)GP_OUT_level);

3.GP_OUT[5][6][7][8]를 Low 로 함
GP_OUT_level &= 0x0F; // Bit4 ~ 7 을 0 로 하면 좋음
AcapSetInfo(hDev, ACL_GPOUT_POL, 0, (int)GP_OUT_level);
    
```

GP_OUT_level 의 각 bit 에 GP_OUT 신호가 할당되어 Low 면 0, High 는 1 을 설정합니다.

bit	GP_OUT 신호의 할당
31-8	미사용
7	GP_OUT8
6	GP_OUT7
5	GP_OUT6
4	GP_OUT5
3	GP_OUT4(향후 대응예정)
2	GP_OUT3(향후 대응예정)
1	GP_OUT2(향후 대응예정)
0	GP_OUT1(향후 대응예정)

Table 6-2 : 할당 일람

6.6 외부입력 신호의 노이즈제거기능

본 제품은 외부 트리거의 노이즈 제거회로를 탑재하고 있습니다.

선택된 외부 트리거에 디지털 필터를 걸어서 단발노이즈와 레벨 변화시의 노이즈를 제거할 수 있습니다.

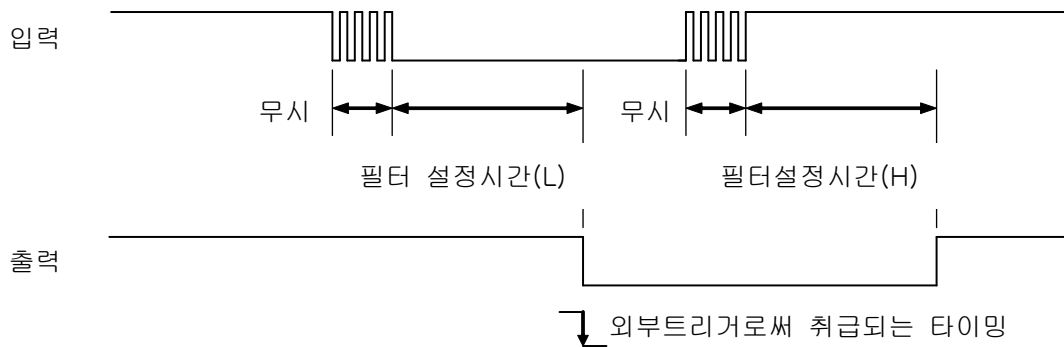


Figure 6-22 : 외부 트리거의 노이즈제거

노이즈 제거시간은 uS 단위로 설정 가능합니다.

디지털 필터의 사양상
출력은 필터 설정시간의 진연이 발생함으로 주의 바랍니다.

■소프트웨어 설정예

아래에 외부 트리거 노이즈 제거시간을 High와 Low 각각 100us로 하는 경우의 샘플코드를 설명합니다.

<샘플 코드>

```
//노이즈제거 시간을100us으로 설정
AcapSetInfo( hDev, INPUT_CH, ACL_EXT_CHATTER, -1, 100);
//외부 트리거High시간, Low와 같이 동일 값
AcapSetInfo( hDev, INPUT_CH, ACL_EXT_CHATTER_SEPARATE, -1, 0);
//설정반영
AcapReflectParam(hDev, INPUT_CH);
```

6.7 스트로브 출력

본 제품은 외부조명기기와 접속에 사용 가능한 스트로브 출력을 할 수 있습니다.

스트로브출력은 STB 단자에서 출력되어 외부조명기기와 카메라 노광타이밍을 조정시키기 위한 출력 신호입니다. 기준신호에서 출력개시까지의 지연시간 및 출력시간을 1uS 단위로 설정 가능합니다.

■소프트웨어 설정에

아래에 스트로브를 정극성에 있는 경우의 샘플을 설명합니다.

〈샘플코드〉

```
// 스트로브 정극성
AcapSetInfo(hDev, INPUT_CH, ACL_STROBE_POL, -1, 1);

/설정반영
AcapReflectParam(hDev, INPUT_CH);
```

6.8 CC 출력과 스트로브 출력 타이밍 조정

CC 출력과 스트로브출력은 출력개시까지의 지연시간을 각각 개별적으로 설정 가능합니다.

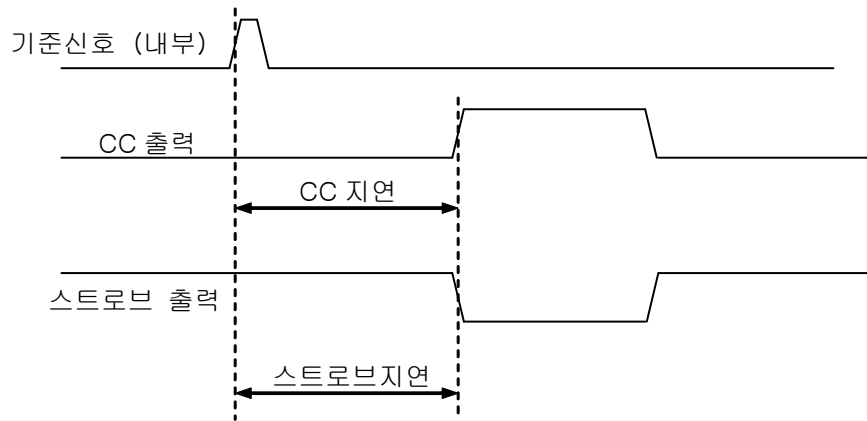


Figure 6-23 : CC/스트로브 출력 타이밍

전원입력시는 CC 출력, 스트로브 출력과의 지연시간은"0", 동일 타이밍에서 출력됩니다.

지연시간, 출력시간을 설정하는 것으로 조명과 노출관계를 아래와 같은 타이밍으로 할 수 있습니다.

1. 카메라 노광중에 스트로브 발광

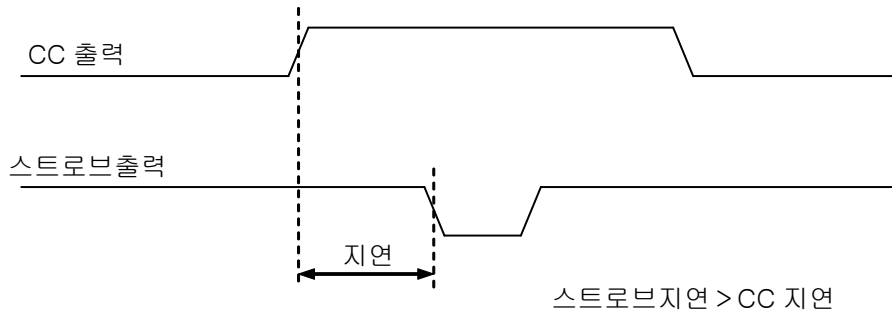


Figure 6-24 : 스트로브 출력 패턴 1

■ 소프트웨어 설정예

아래와 같이 CC 출력과 스트로브 출력 타이밍을 아래로 할 경우의 샘플 코드를 설명합니다.

또한, 지연시간, 노출시간은 아래와 같습니다.

- 스트로브 지연시간 : 100us
- 스트로브 출력시간 : 50us
- CC 지연시간 : 10us

<샘플 코드>

```
//스트로브 지연
AcapSetInfo(hDev, INPUT_CH, ACL_STROBE_DELAY, -1, 100);
//스트로브 출력시간
AcapSetInfo(hDev, INPUT_CH, ACL_STROBE_TIME, -1, 50);
//CC 지연
AcapSetInfo(hDev, INPUT_CH, ACL_CC_DELAY, -1, 10);
//설정반영
AcapReflectParam(hDev, INPUT_CH);
```

2. 스토브 발광중에 카메라 노광

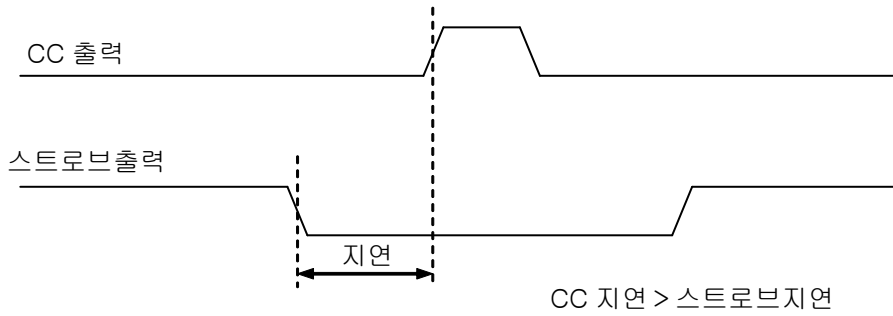


Figure 6-25 : 스토브 출력 패턴 2

■ 소프트웨어 설정에

아래와 같이 CC 출력과 스토브 출력 타이밍을 아래로 할 경우의 샘플 코드를 설명합니다.

또한, 지연시간, 노출시간은 아래와 같습니다.

- 스토브 지연시간 : 30us
- 스토브 출력시간 : 20us
- CC 지연시간 : 200us

〈샘플 코드〉

```
//스토브 지연
AcapSetInfo(hDev, INPUT_CH, ACL_STROBE_DELAY, -1, 30);
//스토브 출력시간
AcapSetInfo(hDev, INPUT_CH, ACL_STROBE_TIME, -1, 20);
//CC 지연
AcapSetInfo(hDev, INPUT_CH, ACL_CC_DELAY, -1, 200);
//설정반영
AcapReflectParam(hDev, INPUT_CH);
```

6.9 엔코더 입력

본 제품은 엔코더 입력 인터페이스에 대해 설명합니다.

6.9.1 엔코더 입력회로

본 제품은 RS422 인터페이스의 incremental 타이밍 엔코더 입력회로를 탑재하여 외부기기(Shaft 엔코더등)와 연동하여 노광제어와 영상입력 타이밍 제어가 가능합니다.

엔코더신호 카운트는 엔코더 A 상, B 상을 사용하여 4 체배 카운트, 또는 A 상만 사용하여 2 체배 카운트가 가능합니다. 추가로 Z 상은 엔코더의 카운트 개시 타이밍으로 사용 가능합니다.

또한 외부 트리거신호를 요인으로한 엔코더 카운트를 개시 할 수 있습니다.

(외부 트리거를 사용하여 엔코더 카운트를 개시할 경우는 한 화면마다 트리거입력이 필요합니다.)

엔코더 입력 주파수는 최대 1MHz 까지 입력 가능합니다.

엔코더 입력 타이밍은 아래 규정을 지켜 주세요.

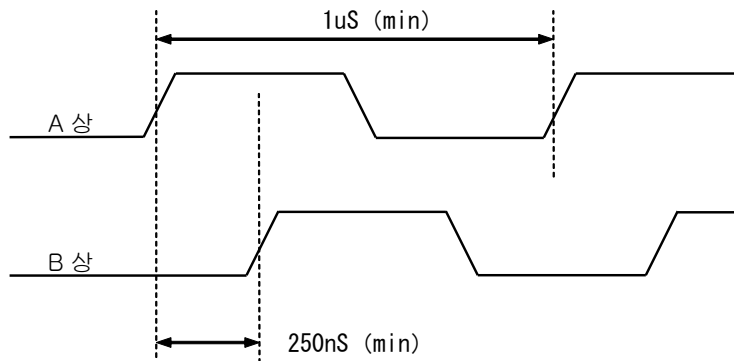


Figure 6-26 : 엔코더 입력 타이밍

입력 펄스의 Duty 비는 50%를 지켜주세요

6.9.2 엔코더 카운트 타이밍

A 상만의 엔코더 카운트 타이밍은 아래와 같습니다.

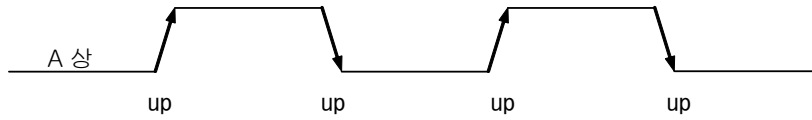


Figure 6-27 : A 상만의 2 체배 카운트

A 상만의 엔코더 카운트는 가산만 실행합니다.

A 상, B 상을 사용한 4 체배 카운트 타이밍은 아래와 같습니다.

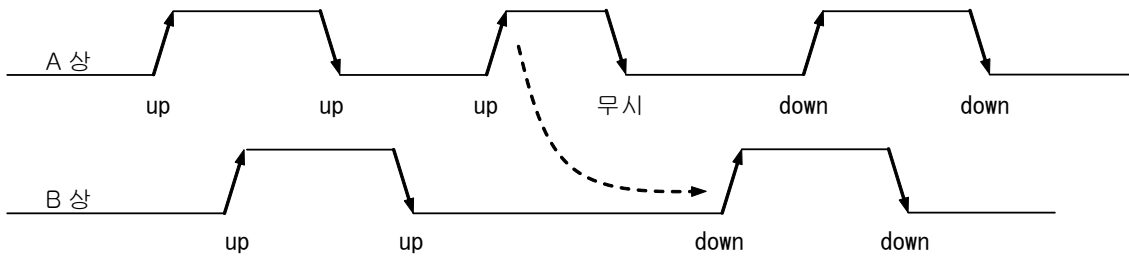


Figure 6-28 : AB 상의 4 체배 카운트

4 체배 카운트는 A 상이 B 상에 대해서 90도 위상이 진행되고 있을 때 가산됩니다. B 상이 A 상에 대해 90도 위상이 진행되고 있을 때 감산이 됩니다.

또한, 노이즈에 의한 잘못된 카운트를 방지하기 위해서 각 엣지 검출시에 위상을 체크하고, 다음 검출해야 할 엣지를 특정합니다.

위와 같이 가산 카운트 동작중 A 상이 High 가 된 후 위상이 변화한 경우는 다음에 B 상이 High 엣지를 대기합니다. 그외 엣지가 입력되어도 무시합니다.

가산, 감산 타이밍을 레지스터로 입력 할 수 있습니다.

6.10 엔코더 카운트 모드

본 제품은 3 종류의 엔코더 카운트 모드가 있습니다. 각각의 모드에 대해 설명합니다.

6.10.1 상대 카운트 모드

상대 카운트는 라인카메라를 사용한 Web 검사용도등에 적합한 모드입니다.

엔코더가 일정 거리를 진행할 때 1 라인 스캔을 실행하도록 동작합니다.

32bit 의 카운트와 32bit 의 비교 레지스터 1 (지연 펄스설정) 16bit 의 비교 레지스터 2 (간격펄스설정) 을 사용하여 동일한 펄스를 생성합니다. 동일펄스는 내부 회로에서 트리거 발생원인으로써 사용됩니다. 펄스수와 카운트 값, 동일펄스는 아래와 같은 타이밍에 생성합니다.

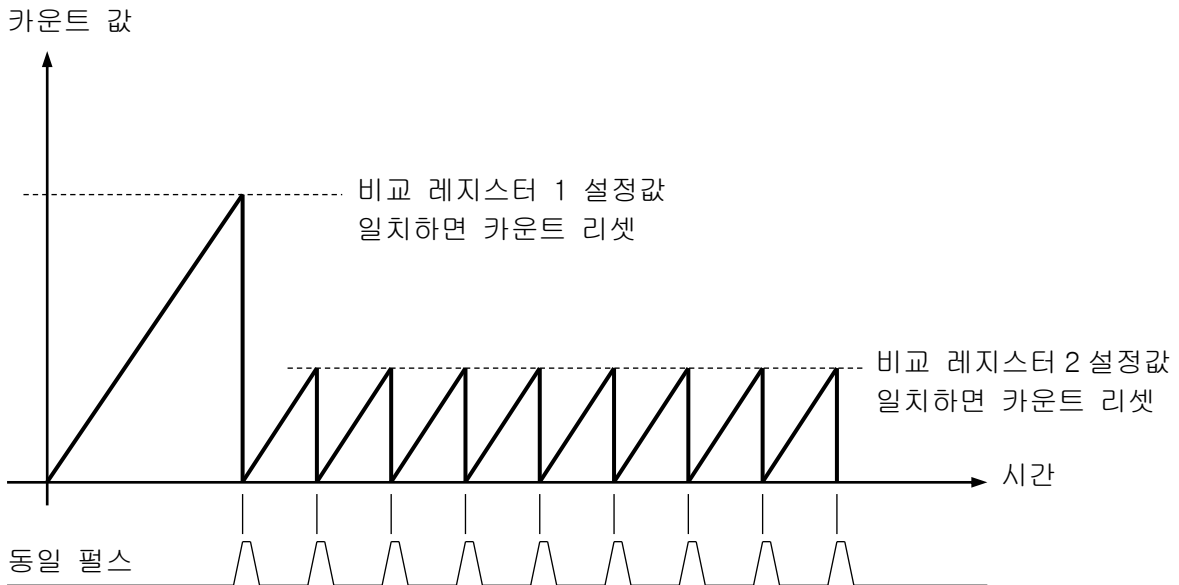


Figure 6-29 : 상대 카운트

취득 개시에서 최초 스캔까지의 지연펄스수를 비교레지스터 1 로 설정합니다.

라인스캔 간격을 비교 레지스터 2 로 설정합니다.

카운트 개시 트리거를 설정합니다. (외부 트리거로 개시, Z 상으로 개시, CPU 에 의한 개시)

상기 설정 완료 후 엔코더 카운트를 개시 합니다.

■ 소프트웨어 설정예

아래에 상대 카운트 모드 및 엔코더 스캔모드에서 촬영할 경우 샘플 코드를 설명합니다.

또한 엔코더 설정은 아래와 같습니다.

- 엔코더 카운트 모드 : 상대 카운트 모드
- 엔코더 기동방법 : 외부 트리거 미사용
- 엔코더 동작모드 : 엔코더 스캔 모드
- 엔코더 입력 펄스 선택 : AB 상 사용
- 엔코더 입력 펄스 방향 : CW 방향
- 엔코더 Z 상 입력 : Z 상 미사용
- 엔코더 비교 레지스터 1 : 10
- 엔코더 비교 레지스터 2 : 4

<샘플 코드>

```

//상대 카운트 모드
AcapSetInfo(hDev, INPUT_CH, ACL_ENC_EN, -1, 1);
//외부 트리거 미사용
AcapSetInfo(hDev, INPUT_CH, ACL_ENC_START, -1, 0);
//엔코더 스캔 모드
AcapSetInfo(hDev, INPUT_CH, ACL_ENC_MODE, -1, 0);
//AB 상 사용
AcapSetInfo(hDev, INPUT_CH, ACL_ENC_PHASE, -1, 0);
//CW 방향
AcapSetInfo(hDev, INPUT_CH, ACL_ENC_DIRECTION, -1, 0);
//Z 상 미사용
AcapSetInfo(hDev, INPUT_CH, ACL_ENC_ZPHASE_EN, -1, 0);
//비교 레지스터 1 을 10 으로 설정
AcapSetInfo(hDev, INPUT_CH, ACL_ENC_COMPARE_1, -1, 10);
//비교 레지스터 2 를 4 로 설정
AcapSetInfo(hDev, INPUT_CH, ACL_ENC_COMPARE_2, -1, 4);
//설정값 반영
AcapReflectParam(hDev, INPUT_CH);

```

6.10.2 절대 카운트 · 싱글 포인트 모드

절대 카운트 · 싱글 포인트 모드는 스테이지를 사용한 영상취득 용도에 적합한 모드입니다.

상대 카운트 모드에서는 펄스가 일치할 때 마다 카운트를 리셋해 버리지만, 이 모드에서는 원점에서 이동거리 (카운트 값) 을 항상 유지함으로, 진행과 복귀가 반복되는 스테이지 등에서 영상 취득 개시 위치를 일정하게 하는 것이 가능합니다.

32bit 의 절대 카운트와 16bit 의 간격 카운트가 있습니다.

간격 카운트를 기동하는 절대 카운트 값을 설정하는 것이 비교 레지스터 1 입니다.

간격 카운트는 동일 펄스를 발생시키는 펄스 수를 카운트 합니다. 비교 레지스터 2 에서 설정합니다.

비교 레지스터 1 및 비교 레지스터 2 를 설정후, 엔코더 카운트 개시조건을 설정하고 카운트를 개시합니다.

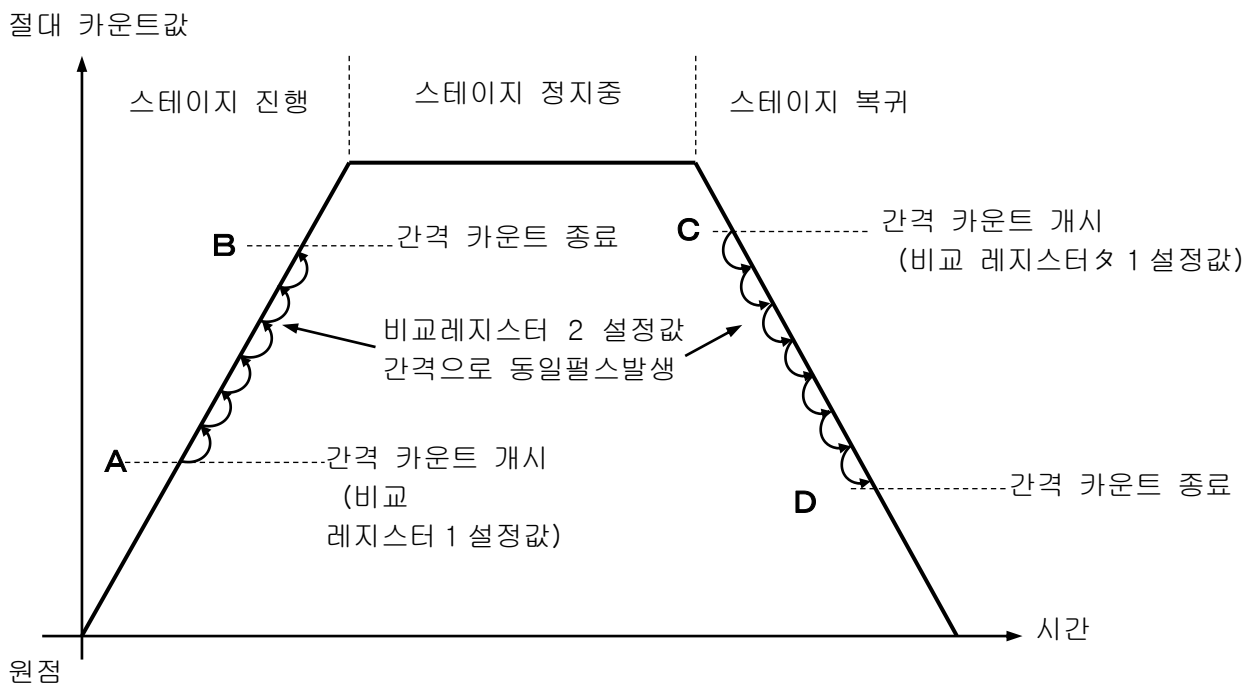


Figure 6-11 : 절대 카운트 · 싱글 포인트 모드

- 절대 카운트는 비교 레지스터 1 의 설정값과 일치하여도 리셋되지 않습니다. 항상 값을 유지합니다.
- 간격 카운트는 비교 레지스터 2 의 설정값과 일치하면 리셋 됩니다.
- 동일 펄스는 비교 레지스터 1, 비교 레지스터 2 와 각 카운트 값이 일치할 때 출력 합니다.

상기 그림에서는 진행방향의 취득개시 위치 A 를 비교 레지스터 1 로 설정하고 동일 펄스 발생간격을 비교 레지스터 2 로 설정하여 카운트를 개시합니다. 취득 개시후 절대 카운트가 B 보다 큰값이 된 시점에 간격 카운트를 정지 시킵니다.

다음으로 복귀방향의 취득 개시위치 C 를 비교 레지스터 1 로 설정하고 동일펄스 발생간격을 비교 레지스터 2 로 설정하고 간격 카운트를 다시 유효로 설정합니다. 복귀방향의 취득이 끝나고 절대 카운트가 D 보다 적은 값이 된 시점에서 간격 카운트, 절대 카운트를 정지하고 완료 합니다.

■ 소프트웨어 설정예

아래에 절대 카운트 · 싱글 포인트 모드 및 엔코더 스캔 모드로써 왕복 각각 50 라인 촬영할 경우의 샘플을 설명합니다. 또한 엔코더의 각 설정 및 「Figure 6-11 : 절대 카운트 · 싱글 포인트 모드」로 표기한 지점의 좌표는 아래와 같습니다.

[엔코더 설정]

엔코더 카운트 모드 : 절대 카운트 · 싱글 포인트 모드
 엔코더 기동방법 : 외부 트리거 미사용
 엔코더 동작 모드 : 엔코더 스캔 모드
 엔코더 입력 펄스 선택 : AB 상 사용
 엔코더 입력 펄스방향 : CW 방향
 엔코더 Z 상 입력 : Z 상 미사용
 엔코더 비교 레지스터 1 : 5000
 엔코더 비교 레지스터 2 : 100

[각지점의 좌표]

지점 A : 5000 카운트
 지점 B : 9900 카운트
 지점 C : 9900 카운트
 지점 D : 5000 카운트

〈샘플 코드〉

```

////////////////////////////////////
//                               원점 : 왕복 설정 및 활상 개시                               //
////////////////////////////////////
//왕복, 절대위치 엔코더 카운트「5000」로부터 활상을 개시
//그 후「100」카운트마다 1 라인 촬영하여, 「50」라인 촬영을 실행 합니다.

// 엔코더 설정 (왕복)
AcapSetEncoder( hDev, INPUT_CH, 2, 0, 0, 0, 0, 0, 5000, 100 );
//엔코더 카운트 개시
AcapSetInfo(hDev, INPUT_CH, ACL_ENC_ABS_START, 0, 1);
//촬영 개시(Snap)
AcapGrabStart(hDev, INPUT_CH, 1)
    
```

```

////////////////////////////////////
//                               A 지점 : 왕복 촬영 개시                               //
////////////////////////////////////
//절대값 카운트가 5000 을 달성한 지점에서 동일 펄스가 발생하고 촬영개시가 됩니다.
//「AcapGetEncoder」의「제 11 인수」 또는「ACL_ENC_ABC_COUNT」에서 현재의 카운트 값을 인식할수
//있습니다.

// 절대위치 엔코더 카운트 취득
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);
    
```



```

////////////////////////////////////
//                                     B 지점 : 왕복 촬영 종료                                     //
////////////////////////////////////
// 왕복[50]라인의 촬영이 종료. 그 후 입력을 정지하고 왕복설정을 실행합니다.
//절대위치 엔코더 카운트[9900]부터 활상 개시, 「카운트 감산(CCW 방향)」으로 설정합니다.

//입력정지(「50」라인 입력 대기)
AcapGrabStop( hDev, INPUT_CH );
// 엔코더 설정 (왕복)
AcapSetEncoder( hDev, INPUT_CH, 2, 0, 0, 0, 1, 0, 9900, 100 );

//---엔코더 카운트는 이미 개시되고 있음으로 여기에서는 어떤것도 실행하지 않음---

//촬영개시(Snap)
AcapGrabStart(hDev, INPUT_CH, 1);

```

```

////////////////////////////////////
//                                     C 지점 : 왕복 촬영 개시                                     //
////////////////////////////////////
//절대 위치 엔코더 카운트가[9900]에 도달한 지점에서 동일 펄스가 발생, 촬영이 개시됩니다.
//「AcapGetEncoder」의「제 11 인수」 또는「ACL_ENC_ABC_COUNT」에서 현재의 카운트 값을 인식할수
있습니다.
//카운트 값 취득
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);

```

```

////////////////////////////////////
//                                     D 지점 : 왕복 촬영 종료                                     //
////////////////////////////////////
//왕복[50]라인의 촬영 종료. 그 이후 입력을 정지하고 왕복 설정을 실행 합니다.
//절대 위치 엔코더 카운트[5000]부터 활상을 개시 「카운트 가산(CW 방향)」으로 설정합니다.

//입력 정지(「50」라인 입력 대기)
AcapGrabStop(hDev, INPUT_CH);
//CW 방향(왕복 설정)
AcapSetEncoder( hDev, INPUT_CH, 2, 0, 0, 0, 0, 0, 5000, 100 );

//---엔코더 카운트는 이미 개시되어 있음으로 여기에서는 아무것도 실행하지 않음---

//활상 개시(Snap)
AcapGrabStart(hDev, INPUT_CH, 1);

```

6.10.3 절대 카운트 · 멀티 포인트 모드

절대 카운트 · 멀티 포인트 모드는 절대 카운트 · 싱글 포인트 모드의 확장입니다.
32bit의 절대 카운트 중 256 포인트까지 임의의 위치에서 동일 펄스를 발생 시킬 수 있습니다.
스테이지를 사용한 에어리어 카메라의 취득 용도, 취득 간격이 일정하지 않는 경우등에 적합합니다.

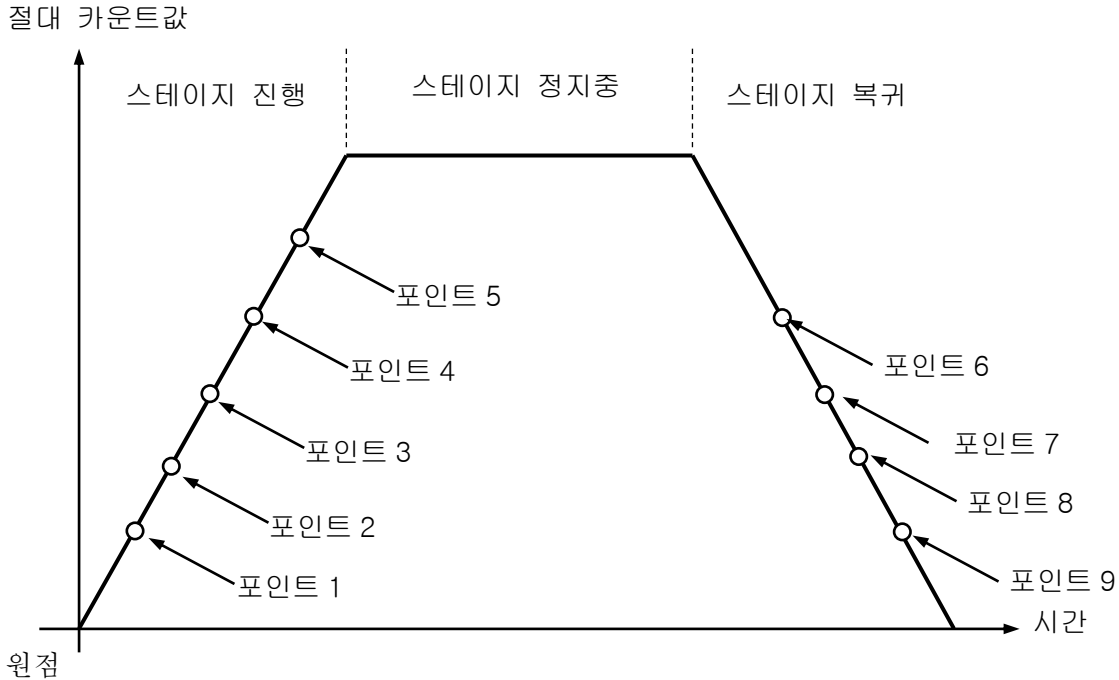


Figure 6-31 : 절대 카운트 · 멀티 포인트 모드

본 모드에서는 비교 레지스터는 존재 하지 않습니다.

비교 레지스터 대신에 내부 레지스터 영역에 맵핑된 256 개의 비교 메모리를 탑재되어 있어, 임의의 카운트 값에서 동일 펄스를 발생 시키는 것이 가능합니다.

■소프트웨어 설정예

아래의 절대 카운트 · 멀티 포인트 모드에서 포인트 1 에서 포인트 9 각각으로 활상하는 경우의 샘플

코드를 설명합니다. 또한 엔코더의 각 설정 및 「[Figure 6-31 : 절대 카운트 · 멀티 포인트 모드](#)」로 표기한 각 포인트의 좌표는 아래와 같습니다.

[엔코더 설정]

엔코더 카운트 코드	: 절대 카운트 · 멀티 포인트 모드
엔코더 기동방법	: 외부 트리거 미사용
엔코더 동작 모드	: 엔코더 스캔 모드
엔코더 입력 펄스 선택	: AB 상 사용
엔코더 입력 펄스방향	: CW 방향
엔코더 Z 상 입력	: Z 상 미사용

[각지점의 좌표]

포인트 1	: 1000 카운트
포인트 2	: 2000 카운트
포인트 3	: 3000 카운트
포인트 4	: 4000 카운트
포인트 5	: 5000 카운트
포인트 6	: 4000 카운트
포인트 7	: 3000 카운트
포인트 8	: 2000 카운트
포인트 9	: 1000 카운트

[취득 개시/종료 포인트]

왕복 개시 포인트	: 포인트 1(1000 카운트)
왕복 종료 포인트	: 포인트 5(5000 카운트)
왕복 개시 포인트	: 포인트 6(4000 카운트)
왕복 종료 포인트	: 포인트 9(1000 카운트)

<샘플코드>

```

////////////////////////////////////
//                               원점 : 비교 메모리설정(왕복)                               //
////////////////////////////////////
//왕복, 절대위치 인코더 카운트「1000」부터 촬영을 개시
//그 후 카운트가「2000」、「3000」、「4000」、「5000」이 도달할때마다 촬영하며, 합계 5회 촬영을 실행

#define INPUT_CH 1
#define CH_OFFSET 0x1000
DWORD dwRegOffset      = 0;    //레지스터 번지
DWORD dwRegVal         = 0;    //레지스터 설정값
int nStartPos          = 0;    //개시 설정
int nEndPos            = 0;    //종료 설정
int i                  = 0;    //루프용 변수

//왕복 비교 메모리를 설정합니다.
// 이 처리는AcapLib2표준함수 에서는 지원하지 않기 때문에 보드의 레지스터를
// 직접 설정해야 합니다,
// CH1의 경우:레지스터5000h번지부터 비교 메모리를 설정 함.
// CH2의 경우:레지스터6000h번지부터 비교 메모리를 설정 함.
for ( i = 0; i < 5; i++ ){
    //취득을 실행하는 카운트 값을 지정
    dwRegVal = 1000 + (i * 1000);
    //비교 메모리를 지정
    dwRegOffset = 0x5000 + (INPUT_CH - 1) * CH_OFFSET + (i * 4);
    //지정한 비교 메모리에 카운트 설정
    AcapSetReg(ghDev, INPUT_CH, 0, dwRegOffset, dwRegVal);
}

//왕복 비교 메모리를 설정합니다.
for(i = 5; i < 9; i++){
    //취득을 실행하는 카운트 값을 지정
    dwRegVal = 4000 - ((i - 5) * 1000);
    //비교 메모리 지정
    dwRegOffset = 0x5000 + (INPUT_CH - 1) * CH_OFFSET + (i * 4);
    //지정한 비교 메모리에 카운트 설정
    AcapSetReg(ghDev, INPUT_CH, 0, dwRegOffset, dwRegVal);
}
    
```

비교 메모리의 레지스터 메모리맵은 채널에 따라 아래와 같습니다.

※본 샘플 코드는 ch1 를 사용합니다.

채널	Address
ch1	5000h ~ 53FFh
ch2	6000h ~ 63FFh

Table 6-3 : 비교 메모리의 레지스터 맵

또한 상기 샘플 프로그램의 경우, 아래와 같이 설정 됩니다.

비교 메모리 No	Address	카운트 설정값
비교 메모리 0	5000h	1000
비교 메모리 1	5004h	2000
비교 메모리 2	5008h	3000
비교 메모리 3	500Ch	4000
비교 메모리 4	5010h	5000
비교 메모리 5	5014h	4000
비교 메모리 6	5018h	3000
비교 메모리 7	501Ch	2000
비교 메모리 8	5020h	1000

Table 6-4 : 비교 메모리 설정예

```
//왕복 비교 메모리 개시위치/종료 위치를 지정
nStartPos = 0;
nEndPos = 4;
dwRegVal = (0xFF & nStartPos) << 8;
dwRegVal |= 0xFF & nEndPos;
//비교 메모리 개시위치/종료 위치 설정
AcapSetInfo( ghDev, INPUT_CH, ACL_ENC_COMPARE_2, 0, dwRegVal );
```

비교 메모리 개시위치와 종료위치를 설정할 경우, AcapSetInfo 함수를 사용하여 비교 레지스터 2 에 각각의 위치를 설정합니다. 아래에 개시위치, 종료위치의 대응 Bit 를 표시합니다.

비교 메모리	대응 Bit
개시위치	Bit15 ~ Bit8
종료 위치	Bit7 ~ Bit0

Table 6-5 : 대응 Bit

예를들어 개시위치를 비교 메모리 0 카운트 설정값, 종료위치를 비교 메모리 4 카운트 설정값으로 할 경우는 비교 레지스터 2 에「0004h」를 설정합니다.

	비교 메모리 No	Address	카운트 설정값
왕복 개시위치 →	비교 메모리 0	5000h	1000
	비교 메모리 1	5004h	2000
	비교 메모리 2	5008h	3000
	비교 메모리 3	500Ch	4000
왕복 종료위치 →	비교 메모리 4	5010h	5000
	비교 메모리 5	5014h	4000
	비교 메모리 6	5018h	3000
	비교 메모리 7	501Ch	2000
	비교 메모리 8	5020h	1000

Table 6-6 : 왕복 개시위치와 종료위치

```

//멀티 포인트 모드를 유효화 합니다.
// 이 처리는AcapLib2표준함수에서는 지원하지 않기 때문에
// 보드 레지스터를 직접설정해야 합니다.

//CH1의 경우:레지스터1028h번지의 bit30을1로 설정함.
//CH2의 경우:레지스터2028h번지의 bit30을1로 설정함

//엔코더 컨트롤 레지스터를 취득
dwRegOffset = 0x1028 + (INPUT_CH - 1) * CH_OFFSET;
AcapGetReg( ghDev, INPUT_CH, 0, dwRegOffset, &dwRegVal);

//멀치 포인트 모드로 변환(bit30에1을 설정)
dwRegVal |= 0x40000000;

//엔코더 컨트롤 레지스터를 설정
AcapSetReg( ghDev, INPUT_CH, 0, dwRegOffset, dwRegVal);
    
```

절대위치 카운트 멀티 포인트 모드로 변경합니다. AcapSetReg 함수를 사용하여 본 제품의 엔코더 컨트롤러 레지스터의 bit30 을 1 로 설정합니다. 그외 bit 는 변경하지 마십시오.
 엔코더 컨트롤러 레지스터의 번지는 채널마다 아래와 같습니다.

채널	Address
ch1	1028h
ch2	2028h

Table 6-7 : 채널에 따른 엔코더 컨트롤러 레지스터

bit	기능
30	절대 카운트 모드 선택을 실행합니다. 0 : 싱글 포인트 모드 1 : 멀티 포인트 모드 전원 입력/리셋시는"0 : 싱글 포인트 모드 입니다.

Table 6-8 : 엔코더 컨트롤 레지스터(bit30)

```

//멀티 포인트 모드를 유효화 합니다.
// 이 처리는AcapLib2표준함수에서는 지원하지 않기 때문에
// 보드 레지스터를 직접설정해야 합니다.

//CH1의 경우:레지스터1028h번지의 bit30을1로 설정함.
//CH2의 경우:레지스터2028h번지의 bit30을1로 설정함

//엔코더 컨트롤러 레지스터를 취득
dwRegOffset = 0x1028 + (INPUT_CH - 1) * CH_OFFSET;
AcapGetReg( ghDev, INPUT_CH, 0, dwRegOffset, &dwRegVal);

//멀치 포인트 모드 카운트 개시(D17에1을 설정)
dwRegVal |= 0x20000;

//엔코더 컨트롤러 레지스터(1028번지)에 설정
AcapSetReg( ghDev, INPUT_CH, REGTYPE_LOCAL, dwRegOffset, dwRegVal);

//왕복 지정매수 취득 개시(5회 촬영을 실시함)
AcapGrabStart( ghDev, INPUT_CH, 5 );
    
```

절대 위치 카운트 멀티 포인트 모드의 카운트를 개시합니다. AcapSetReg 함수를 사용하여 본 제품의 엔코더 컨트롤러 레지스터의 bit17 을 1 로 설정합니다. 이외의 bit 는 변경하지 마십시오.

bit	기능
17	절대 카운트 · 멀티 포인트 모드의 카운트 개시/정지를 설정합니다. 0 : 멀티 포인트 모드 카운트 정지 1 : 멀티 포인트 모드 카운트 개시 또한 동작중 본 bit 에"0"을 쓰면 카운트를 강제정지 합니다. 전원입력/리셋시에는"0"이 됩니다.

Table 6-9 : 엔코더 컨트롤러 레지스터(bit17)

```

//////////////////////////////////////
//                               포인트 1 ~ 포인트 4 : 왕복 촬영개시                               //
//////////////////////////////////////
//절대값 카운트가 1000 에 도달한 지점에서 동일 펄스가 발생하여 1 회 촬영을 실시합니다.
//그 카운트가「2000」、「3000」、「4000」、「5000」에 도달할때 마다 촬영하며, 합계 5 회 촬영합니다.
//「AcapGetEncoder」의「제 10 인수」에서 비교 메모리의 최신 값이 돌아 옵니다.
//또한「ACL_ENC_ABC_COUNT」에서 현재 카운트 값을 확인 할 수 있습니다.
ULONG ulCompMem, ulAbsCount;
//비교 메모리의 최신 값을 취득
AcapGetEncoder( hDev, INPUT_CH, NULL, NULL, NULL, NULL, NULL, NULL, NULL, &ulCompMem, NULL );
// 절대 위치 엔코더 카운트 취득
AcapGetInfo(hDev, INPUT_CH, ACL_ENC_ABS_COUNT, 0, &ulAbsCount);
    
```

```

////////////////////////////////////
//                               포인트 5 : 왕복 촬영 종료                               //
////////////////////////////////////
//왕복 5 회 촬영이 종료. 그 후 입력을 정지하고 촬영설정을 실시합니다.
//왕복 비교메모리 개시위치/종료 위치를 지정

//입력정지
AcapGrabStop( ghDev, INPUT_CH);

//왕복 비교 메모 개시위치/종료 위치를 지정
nStartPos = 5;
nEndPos = 8;
dwRegVal = (0xFF & nStartPos) << 8;
dwRegVal |= 0xFF & nEndPos;

//비교 메모리 개시위치/종료 위치를 설정
AcapSetInfo( ghDev, INPUT_CH, ACL_ENC_COMPARE_2, 0, dwRegVal );

//---여기에서CCW로 변경하면 왕복으로 카운트맵을 하기위해 전환은 실행하지 않음---
//---엔코더 카운트는 이미 개시되어 있음으로 여기에서는 아무것도 실행하지 않음---

//왕복 지정매수 취득 개시(4회 촬영을 실시함)
AcapGrabStart( ahDev, INPUT_CH, 4 );

```

개시위치를 비교 메모리 5의 카운트 설정값 종료위치를 비교 메모리 8의 카운트 설정값으로 할 경우, 비교 레지스터에 2로「0508h」을 설정합니다.

비교 메모리 No	Address	카운트 설정값
비교 메모리 0	5000h	1000
비교 메모리 1	5004h	2000
비교 메모리 2	5008h	3000
비교 메모리 3	500Ch	4000
비교 메모리 4	5010h	5000
왕복개시위치 → 비교 메모리 5	5014h	4000
비교 메모리 6	5018h	3000
비교 메모리 7	501Ch	2000
왕복종료위치 → 비교 메모리 8	5020h	1000

Table 6-10 : 왕복 개시위치와 종료위치

```

////////////////////////////////////
//                               포인트 6 ~ 포인트 8 : 왕복 촬영개시                               //
////////////////////////////////////
//절대값 카운트가 4000 에 도달한 지점에서 동일 펄스가 발생하여 1 회 촬영을 실시합니다.
//그 카운트가 「3000」、 「2000」、 「1000」에 도달할때 마다 촬영하며, 합계 4 회 촬영합니다.
//「AcapGetEncoder」의「제 10 인수」에서 비교 메모리의 최신값이 돌아 옵니다.
//또한「ACL_ENC_ABC_COUNT」에서 현재 카운트 값을 확인 할 수 있습니다 .
ULONG ulCompMem, ulAbsCount;
//비교 메모리의 촬영 값을 취득
AcapGetEncoder( hDev, INPUT_CH, NULL, NULL, NULL, NULL, NULL, NULL, NULL, &ulCompMem, NULL );
// 절대위치 엔코더 카운트 취득
AcapGetInfo(hDev, INPUT_CH, ACL_ENC_ABS_COUNT, 0, &ulAbsCount);

```

```

////////////////////////////////////
//                               포인트 9 : 왕복 촬영종료                               //
////////////////////////////////////
//왕복 4 회 촬영이 종료. 그 후 입력을 정지하고 왕복설정을 실시합니다.
//왕복 비교 메모리 개시위치/종료 위치를 설정

//입력정지
AcapGrabStop( ghDev, INPUT_CH);

//왕복 메모리 개시위치/종료위치를 지정
nStartPos = 0;
nEndPos = 4;
dwRegVal = (0xFF & nStartPos) << 8;
dwRegVal |= 0xFF & nEndPos;

//비교 메모리 개시위치/종료위치 설정
AcapSetInfo( ghDev, INPUT_CH, ACL_ENC_COMPARE_2, 0, dwRegVal );

//---엔코더 카운트는 이미 개시 되어 있음으로 여기서는 아무것도 실행하지 않음---

//왕복 지정매수 취득 개시(5회 촬영을 실행)
AcapGrabStart( ghDev, INPUT_CH, 5 );

```

6.11 엔코더 동작 모드

본 제품은 2 종류의 엔코더 동작모드가 있습니다. 각각의 모드에 대해 설명합니다.

6.11.1 엔코더 스캔 모드

엔코더에 동일한 펄스로써 CC 신호를 출력하는 모드입니다.

카메라에 제공되는 셔터 트리거(CC 신호)를 엔코더에서 제어하려고 할 때 유효합니다.

아래 엔코더 스캔 모드의 타이밍에 대해 설명합니다.

또한, 보드의 엔코더 설정 및 카메라의 노광설정은 아래와 같습니다.

[보드 설정예]

- 엔코더 카운트 모드 : 상대 카운트 모드
- 엔코더 동작 모드 : 엔코더 스캔 모드
- 엔코더 펄스 선택 : AB 상 사용
- 엔코더 펄스 방향 : CW 방향
- 엔코더 비교 레지스터 1 : 8
- 엔코더 비교 레지스터 2 : 4

[카메라 설정]

노광개시 타이밍 : CC 신호에 동기(Active High 엷지에 동기)

노광시간 : 카메라측에서 설정

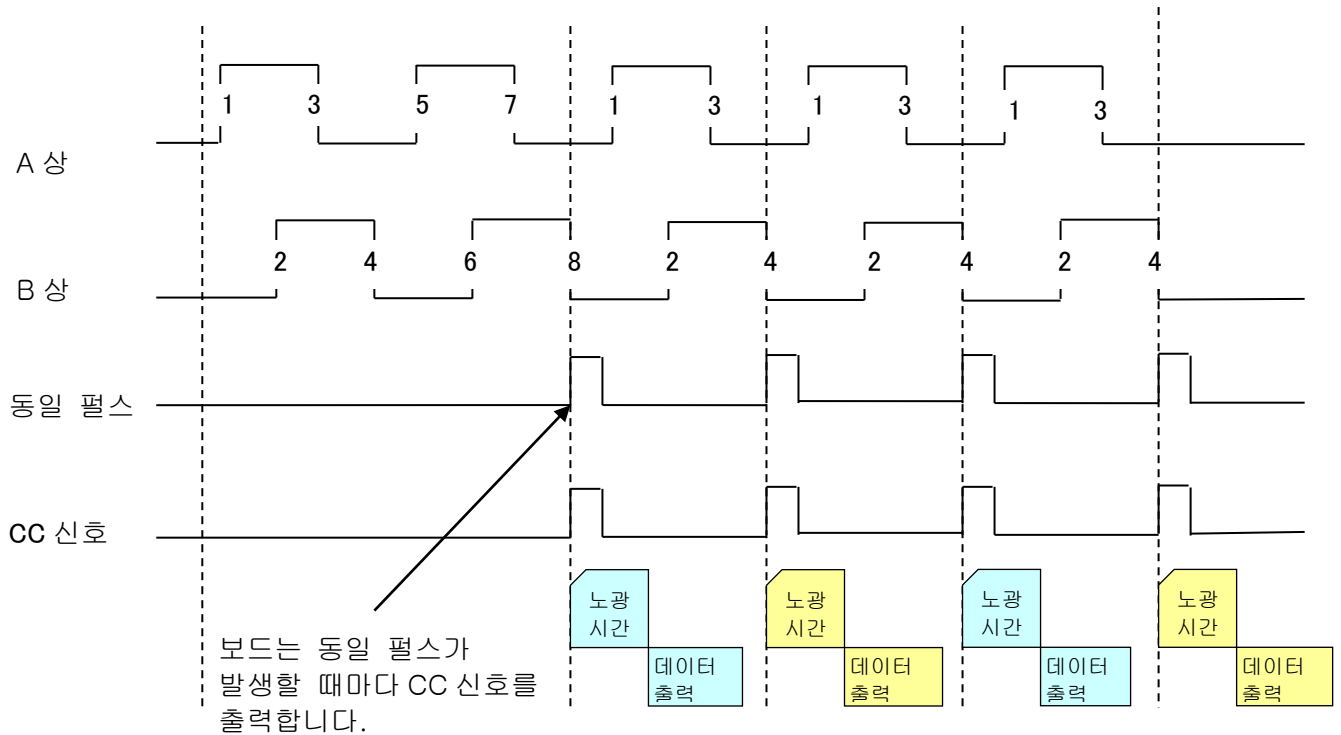


Figure 6-32 : 엔코더 스캔

【설정시 주의점】

○CC 신호가 출력 가능한 설정으로 해야만 합니다.

※CC 신호의 출력주기, 출력폭을 설정함 ([ACL_EXP_CYCLE](#), [ACL_EXPOSURE](#))

※「CC 출력주기 > CC 출력폭 > 0」을 만족하지 않는 경우에는 CC 신호가 출력되지 않습니다.

○CC 신호 출력을 유효로함 ([ACL_EXP_EN](#))

○아래의 설명에 따라 엔코더 신호를 입력해 주세요

※「동일 펄스 주기 > CC 신호 출력 주기」를 만족해야만 합니다.

아래와 같이 CC 신호출력 주기중에 동일 펄스가 발생할 경우 CC 신호를 출력 할 수 없습니다.

[보드 설정예]

- 엔코더 카운트 모드 : 상대 카운트 모드
- 엔코더 동작 모드 : 엔코더 스캔 모드
- 엔코더 펄스 선택 : AB 상 사용
- 엔코더 펄스 방향 : CW 방향
- CC 출력주기 : 100us
- CC 출력폭 : 10us
- 엔코더 비교 레지스터 1 : 8
- 엔코더 비교 레지스터 2 : 4

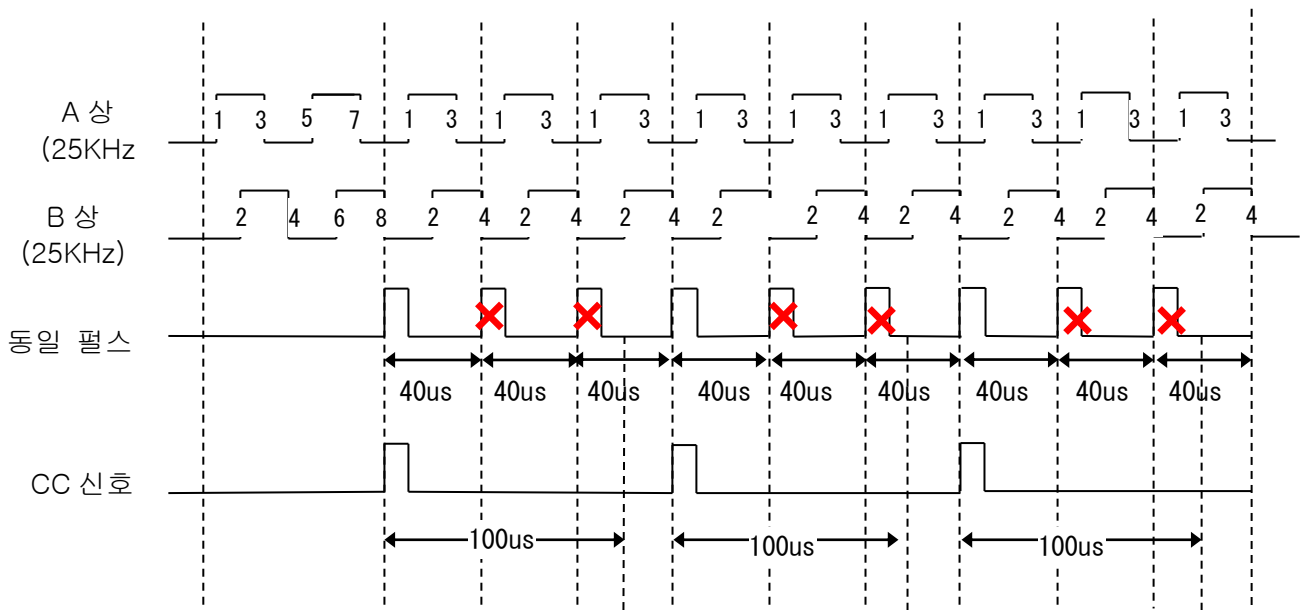


Figure 6-33 : 동일 펄스가 무시되는 경우

이러한 현상이 발생할 경우, 촬영 영상은 예상되는 영상보다도 Y 방향으로 압축된 영상이 됨으로 주의 바랍니다.

○CC 신호 출력주기를 노광시간으로 설정하는 카메라를 엔코더 스캔 모드에서 영상을 취득할 경우

엔코더 속도에 따라 노광시간이 변동 됩니다.

이 때문에 이러한 엔코더가 느릴경우는 밝고, 고속일 경우는 어두워 집니다.

[보드 설정예]

- 엔코더 카운트 모드 : 상대 카운트 모드
- 엔코더 동작 모드 : 엔코더 스캔 모드
- 엔코더 펄스 선택 : AB 상 사용
- 엔코더 펄스 방향 : CW 방향
- CC 출력주기 : 40us
- CC 출력폭 : 10us
- 비교레지스터 1 : 8
- 비교레지스터 2 : 4

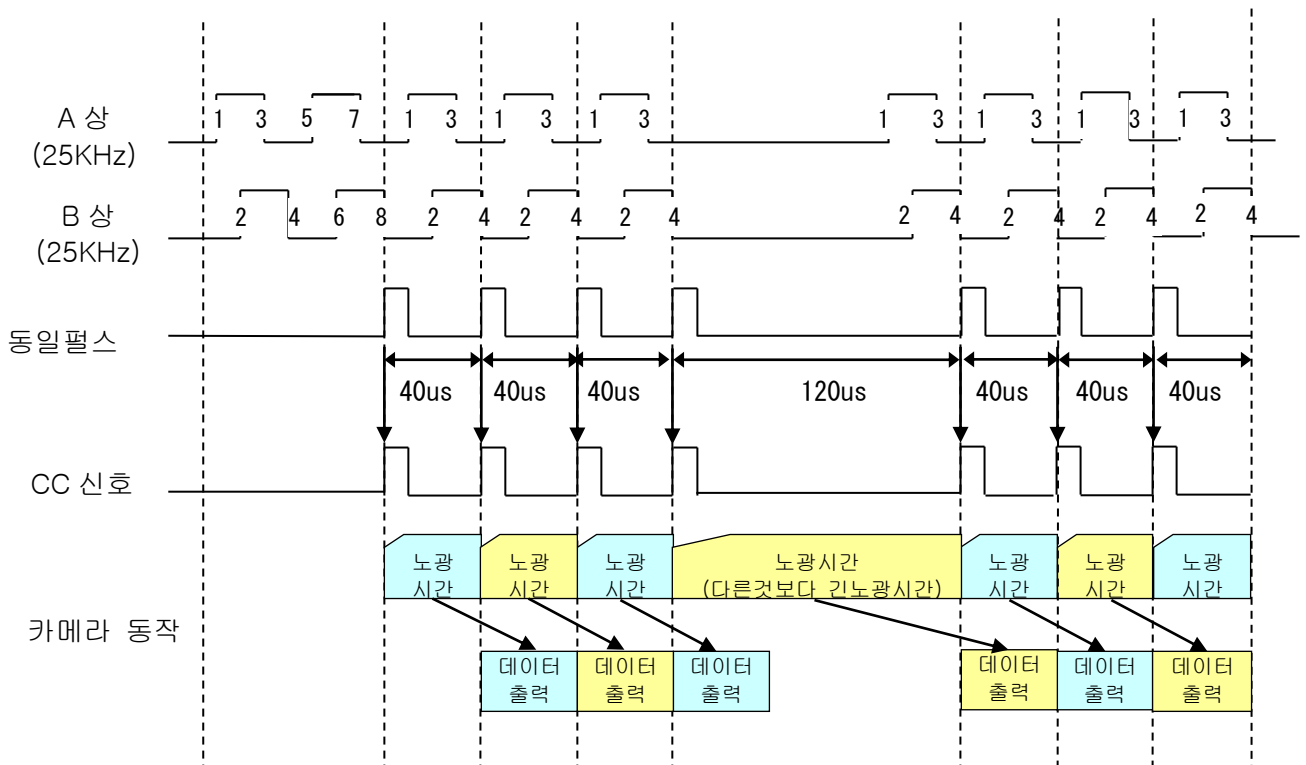


Figure 6-34 : 노광시간이 변동할 경우

상기와 같이 엔코더 속도에 따라 노광시간이 변경하는 경우에는「[6.11.2 엔코더 라인선택 모드](#)」항목의

엔코더 라인선택 모드를 사용해 주세요

6.11.2 엔코더 라인선택 모드

엔코더의 동일 펄스 검출 후, 최초 카메라의 데이터(LVAL)을 입력하는 모드입니다.

엔코더와 관계없는 방법으로 노광제어를 실행하고 데이터를 입력하는 타이밍에만 엔코더를 제어 할 경우에 유효합니다.

【카메라가 Free run 일 경우】

아래에 엔코더 라인 선택 모드 및 카메라 Free run 일 경우의 타이밍 예를 설명합니다.

또한 보드의 엔코더 설정 및 카메라의 노광설정은 아래와 같습니다.

[보드 설정예]

- 엔코더 카운트 모드 : 상대 카운트 모드
- 엔코더 동작 모드 : 엔코더 라인선택 모드
- 엔코더 펄스 선택 : AB 상 사용
- 엔코더 펄스 방향 : CW 방향
- 엔코더 비교 레지스터 1 : 8
- 엔코더 비교 레지스터 2 : 12

[카메라 설정]

- 노광개시 타이밍 : Free run
- 노광시간 : 카메라측에서 설정

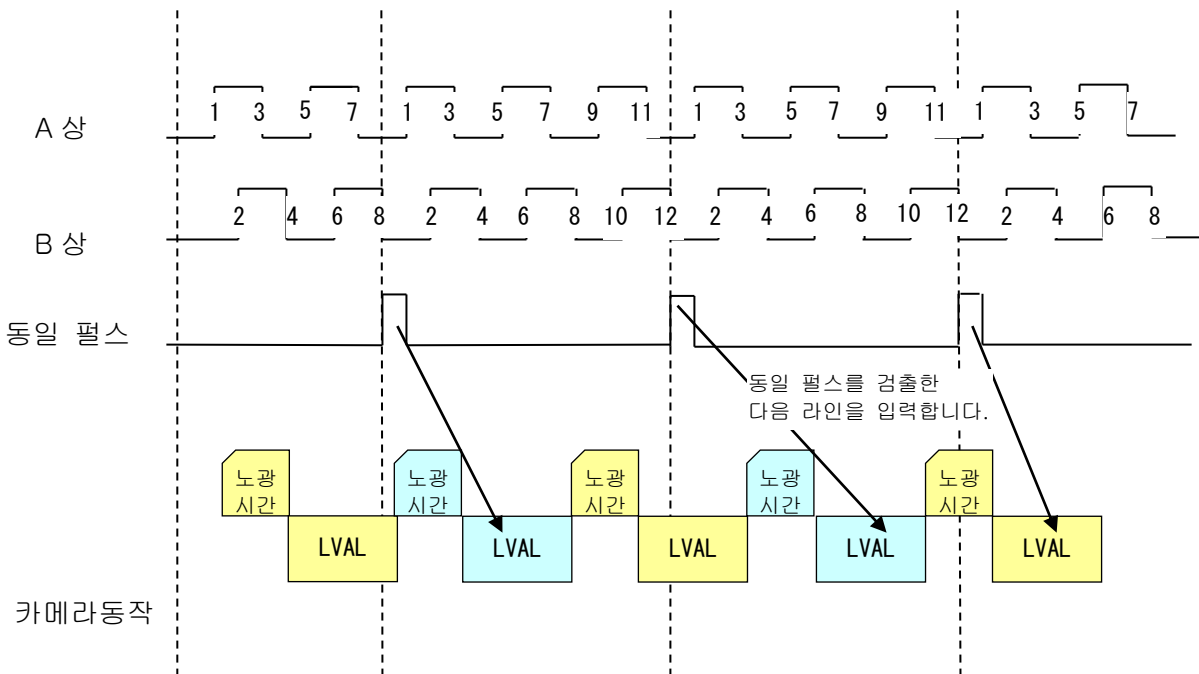


Figure 6-35 : 엔코더 라인 선택 모드(Free run)

【카메라가 CC 신호에 동기되는 경우】

아래에 엔코더 라인 선택모드 및 카메라가 CC 신호에 동기 할 경우의 타이밍의 예를 설명합니다.
또한 보드의 엔코더 설정 및 카메라 노광설정은 아래와 같습니다.

[보드 설정예]

- 엔코더 카운트 모드 : 상대 카운트 모드
- 엔코더 동작 모드 : 엔코더 라인선택 모드
- 엔코더 펄스 선택 : AB 상 사용
- 엔코더 펄스 방향 : CW 방향
- CC 출력주기 : 100us
- CC 출력폭 : 10us
- 엔코더 비교 레지스터 1 : 8
- 엔코더 비교 레지스터 2 : 12

[카메라 설정]

- 노광개시 타이밍 : CC 신호에 동기(Active high 엣지에 동기)
- 노광시간 : 카메라측에서 설정

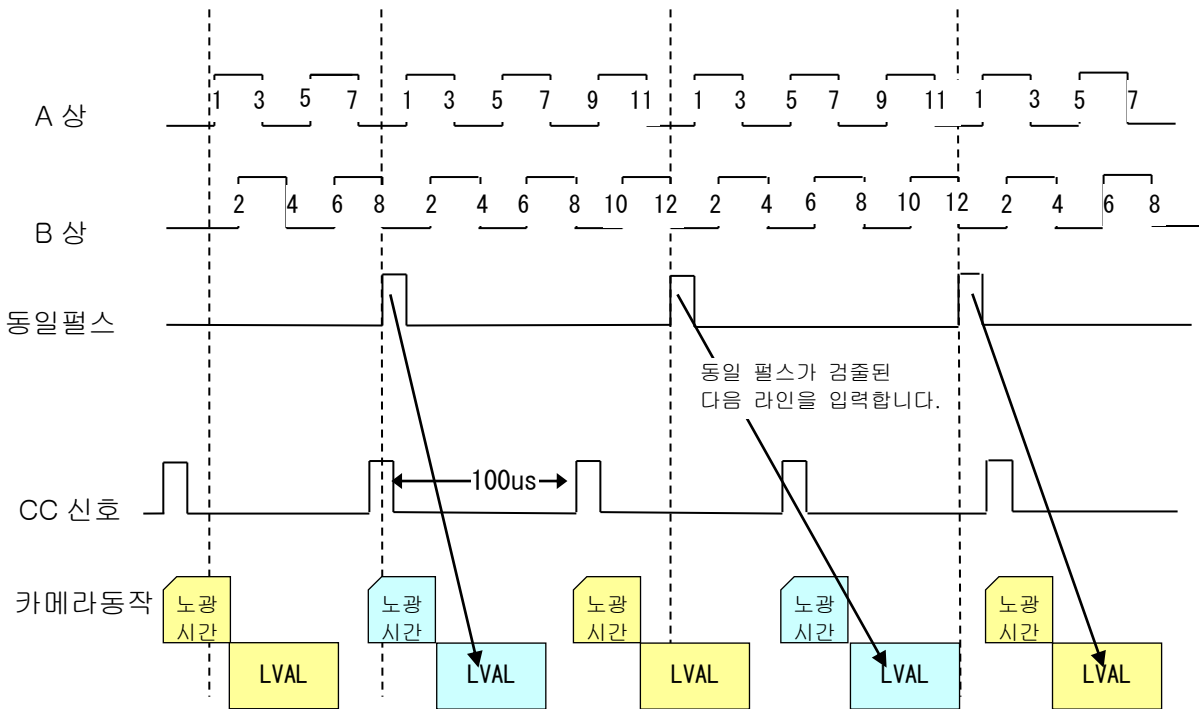


Figure 6-36 : 엔코더 라인 선택 모드(CC 신호에 동기)

【설정 시 주의점(카메라가 Free run 일 경우)】

○동일 라인에서 복수의 동일 펄스가 검출된 경우, 아래와 같이 2 회째 이후의 동일펄스는 무시됩니다.

[보드 설정예]

- 엔코더 카운트 모드 : 상대 카운트 모드
- 엔코더 동작 모드 : 엔코더 라인선택 모드
- 엔코더 펄스 선택 : AB 상 사용
- 엔코더 펄스 방향 : CW 방향
- 엔코더 비교 레지스터 1 : 4
- 엔코더 비교 레지스터 2 : 4

[카메라 설정]

노광개시 타이밍 : CC 신호의 Active high 엣지
 노광시간 : 카메라측에서 설정

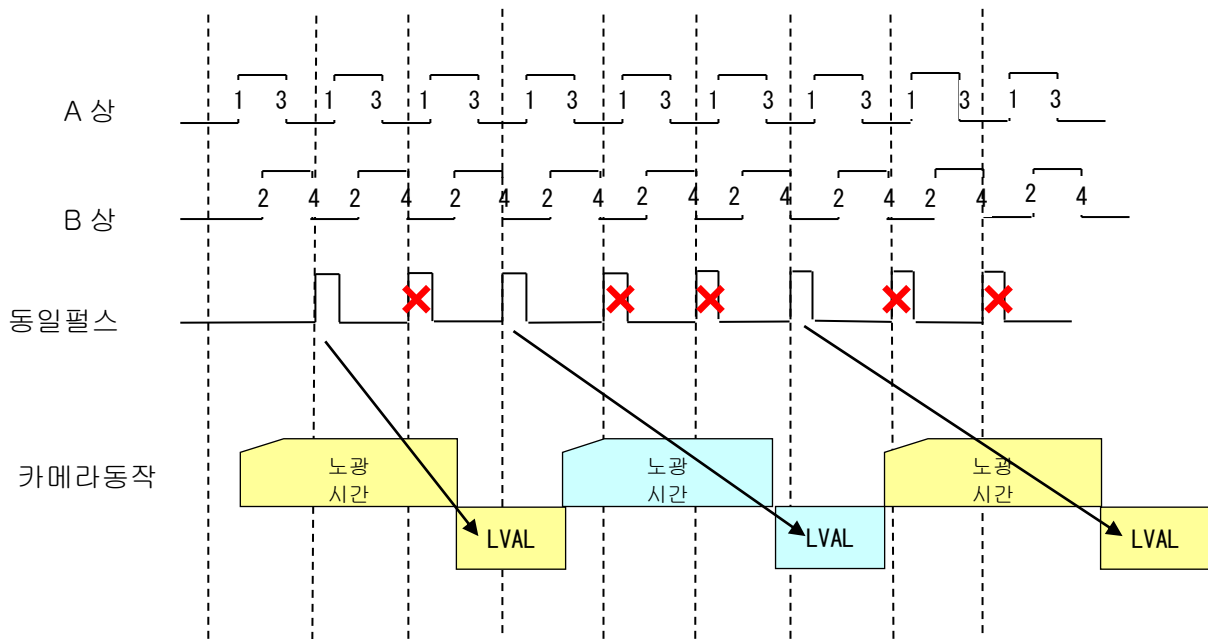


Figure 6-37 : 동일 펄스가 무시되는 경우(Free run)

이러한 현상이 발생할 경우는 촬영영상은 예상보다도 Y 방향으로 압축된 영상이 됨으로 주의바랍니다.

【설정시의 주의점(카메라가 CC 신호에 동기 될 경우)】

○CC 신호를 출력할 수 있는 설정으로 되어 있어야 합니다.

※CC 신호의 출력주기, 출력폭을 설정함(ACL_EXP_CYCLE, ACL_EXPOSURE)

※「CC 출력주기>CC 출력폭>0」을 만족하지 않는 경우 CC 신호가 출력되지 않습니다.

○CC 신호 출력을 유효로함.(ACL_EXP_EN)

○아래의 제한사항을 지켜서 엔코더 신호를 입력하여야 합니다.

※「동일 펄스의 주기>CC 신호 출력주기」를 만족해야만 합니다.

아래와 같이 CC신호 출력 주기중에 동일 펄스가 발생한 경우 2회째 이후의 동일 펄스는 무시되며 CC신호를 출력 할 수 없습니다.

[보드 설정예]

- 엔코더 카운트 모드 : 상대 카운트 모드
- 엔코더 동작 모드 : 엔코더 라인선택 모드
- 엔코더 펄스 선택 : AB 상 사용
- 엔코더 펄스 방향 : CW 방향
- CC 출력주기 : 100us
- CC 출력폭 : 10us
- 엔코더 비교 레지스터 1 : 4
- 엔코더 비교 레지스터 2 : 4

[카메라 설정]

노광개시 타이밍 : CC 신호의 Active high 엣지

노광시간 : 카메라측에서 설정

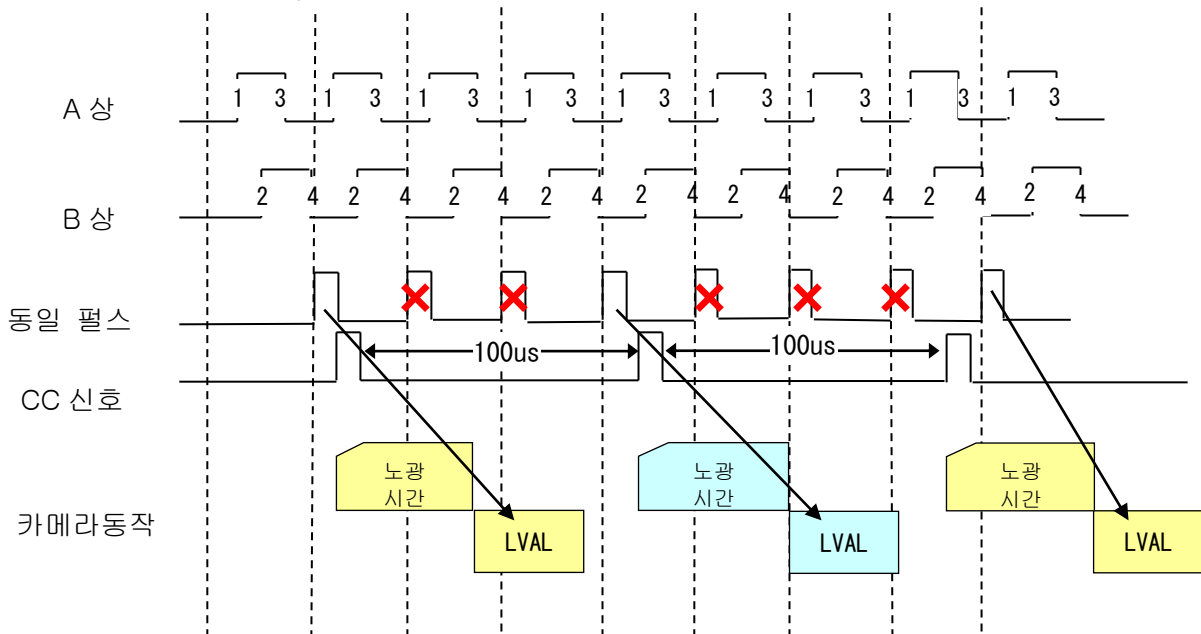


Figure 6-38 : 동일 펄스가 무시되는 경우(CC 신호에 동기)

이러한 현상이 발생할 경우는 촬영영상은 예상보다도 Y 방향으로 압축된 영상이 됨으로 주의바랍니다

6.12 MEGA-FIFO

본 제품은 카메라에서의 유효화수에서 일부분을 잘라내어 영상버퍼에 입력할 수 있습니다.

본 제품의 영상전송기능은 카메라에서의 영상을 대용량 영상입력 버퍼에 일시적으로 저장하여, 버퍼를 경유하여 PC 호스트 메모리에 DMA 전송합니다. 대용량 버퍼에서의 DMA 전송기능은 카메라의 최고 동작속도에서의 영상입력을 보증합니다. 이러한 논리구조를 MEGA-FIFO 아키텍처라고 부릅니다.

FIFO 로 구성하는 것으로 영상입력개시와 동시에 DMA 개시가 가능하며, 텍 타임을 단축할 수 있습니다.

또한 PCI Express 의 대역이 일시적으로 카메라 Frame rate 를 하회하는 경우에도 영상입력 버퍼가 Full 이 될 때까지 영상입력이 가능합니다.

영상버퍼에 저장하는 것은 아래와 같이 FVAL 에서 영상취득까지의 지연행수 TRG_DELAY, 실제 취득행수 Y_TOTAL 에 의한 행수 설정, LVAL 에서 영상취득 개시까지의 지연 화소수 X_DELAY, 실제 취득화소수 X_SIZE 에 의한 화소수 설정이 가능합니다.

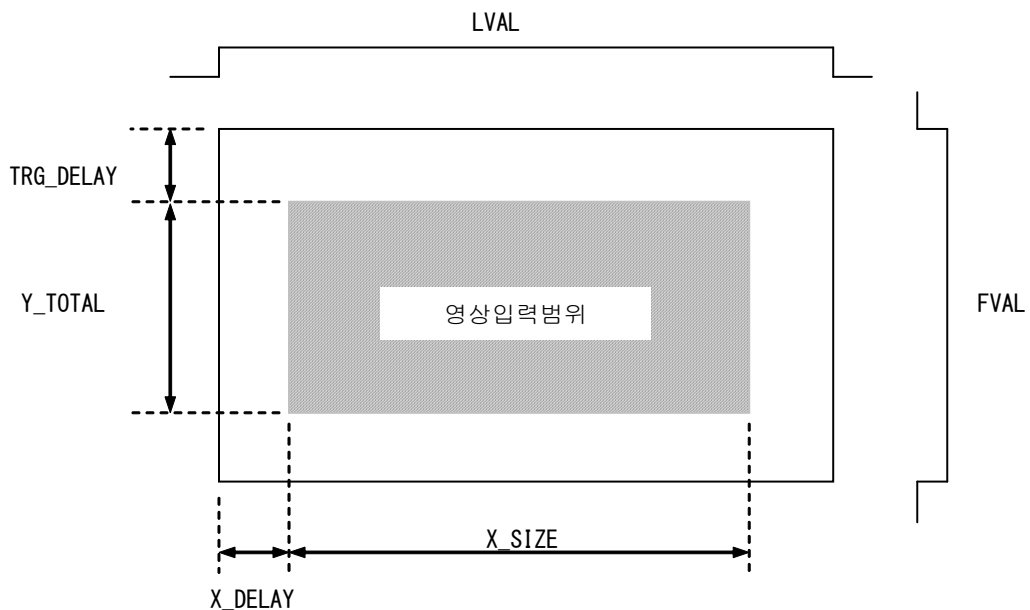


Figure 6-39 : 영상입력 설정

영상버퍼에 저장된 영상데이터는 On board DMA 기능을 사용하여 호스트에 전송합니다.

DMA 는 128bit 단위로 전송제어 됩니다.
64bit/32bit 단위로 DMA 전송에 대해서는 향후 지원 예정입니다.

영상버퍼는 채널당 64MByte 탑재하고 있습니다. 버퍼가 Full 상태가 되면「FIFO Over flow 에러」를 출력하여 입력을 정지합니다

■소프트웨어 설정예

아래에 입력 사이즈를 설정하는 샘플 코드를 설명합니다. 또한 영상입력범위 및 각 부호의 타이밍은 아래와 같습니다.

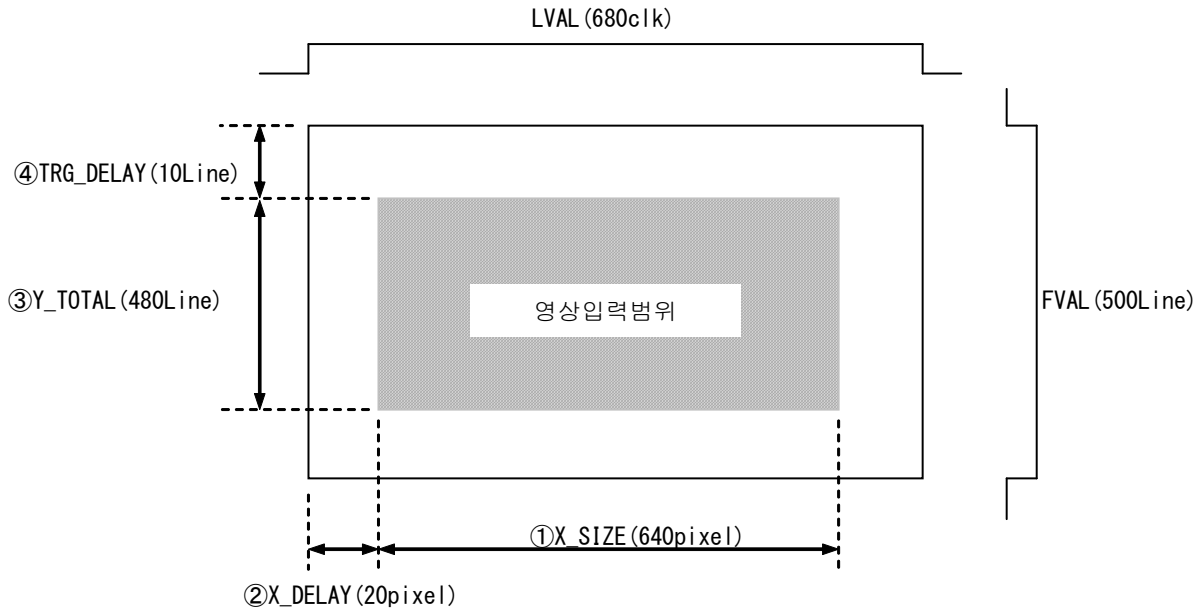


Figure 6-40 : 영상입력 설정

<샘플 코드>

```
// ①X_SIZE: 사용ID는ACL_X_SIZE 를 사용합니다.
AcapSetInfo( hDev, INPUT_CH, ACL_X_SIZE, -1, 640);

// ②X_DELAY: APX-3324 의경 우 설정 ID 는 ACL_X_DELAY 가 아니라 < ACL_LVAL_DELAY 를 사용하세요.
// 또한, 카메라가 1TAP 일 경우에는 X_DELAY 의 값을 그대로 설정합니다만
// 2TAP 의 경우는 X_DELAY 의 절반 값, 3TAP 의 경우에는 X_DELAY 의 1/3 값으로 설정해 주세요
// ACL_LVAL_DELAY 을 설정
AcapSetInfo( hDev, INPUT_CH, ACL_LVAL_DELAY, -1, 20);

// ③Y_TOTAL: 무한대 취득을 실행하지 않는 한 Y 사이즈와 Y Total 은 같은 값으로 설정해 주세요
// 사용 ID 는 ACL_Y_SIZE 와 ACL_Y_TOTAL 입니다.
// ACL_Y_SIZE 를 설정
AcapSetInfo( hDev, INPUT_CH, ACL_Y_SIZE, -1, 480);
// ACL_Y_TOTAL 을 설정
AcapSetInfo( hDev, INPUT_CH, ACL_Y_TOTAL, -1, 480);

// ④TRG_DELAY: 사용 ID 는 ACL_Y_DELAY 입니다.
AcapSetInfo( hDev, INPUT_CH, ACL_Y_DELAY, -1, 10);

//설정값 반영
AcapReflectParam(hDev, INPUT_CH);
```

■ 설정범위와 제한에 대해

ACL_X_SIZE, ACL_Y_SIZE, ACL_Y_TOTAL, ACL_LVAL_DELAY, ACL_Y_DELAY 에는 각각, 아래와 같은 제한사항이 있습니다. 이를 위해 각 설정 ID 를 사용할 때에는 주의 바랍니다.

設定 ID	설정범위와 제한	
ACL_X_SIZE	8bit/1tap	16 ~ 8,192(16 의배수)
	8bit/2tap	16 ~ 16,384(16 의배수)
	8bit/3tap	48 ~ 24,576(48 의배수)
	8bit/4tap	32 ~ 32,768(32 의배수)
	8bit/8tap	64 ~ 65,472(64 의배수)
	8bit/10tap	80 ~ 65,520(80 의배수)
	10bit ~ 16bit/1tap	8 ~ 8,192(8 의배수)
	10bit ~ 12bit/2tap	8 ~ 16,384(8 의배수)
	10bit ~ 12bit/4tap	32 ~ 32,768(32 의배수)
	10bit/8tap	64 ~ 65,472(64 의배수)
	24bitRGB/1tap	16 ~ 8,192(16 의배수)
	30bitRGB/1tap	8 ~ 8,192(8 의배수)
	32bitRGB/1tap	8 ~ 8,192(8 의배수)
	32bitRGB/2tap	8 ~ 16,384(8 의배수)
ACL_Y_SIZE	1 ~ 16,777,215	
ACL_Y_TOTAL	0 ~ 16,777,215	
ACL_LVAL_DELAY	0 ~ 65,535	
ACL_Y_DELAY	0 ~ 1,023	

Table 6-11 : 설정범위와 제한

6.13 끼워넣기

본 제품은 PCI Express 에 대해 1 레벨의 끼워넣기 출력이 가능합니다.

끼워넣기 요인은 아래 4 종류에서 복수 선택 가능합니다.

- DMA 종료 통지
- 영상입력 개시 통지
- Over flow 통지
- Time out 통지

이들 끼워넣기는 이벤트 또는 콜백함수로 통지됩니다.

- 이벤트 통지(AcapSetEvent/AcapWaitEvent)
- 콜백 함수(AcapRegistCallback/AcapRegistCallbackEx)

또한, 각각의 끼워넣기에 대한 정의는 아래와 같습니다.

끼워넣기	정의	의미
입력개시 통지	ACL_INT_GRABSTART	1 프레임의 데이터 입력이 개시된 것을 통지합니다.。
1 프레임 입력완료 통지	ACL_INT_FRAMEEND	1 프레임의 데이터를 버퍼에 전송이 완료된 것을 통지합니다.
입력종료 통지	ACL_INT_GRABEND	AcapGrabStart 함수의 InputNum 에서 입력 프레임수를 지정한 경우, 지정 프레임수의 취득이 완료된 것을 통지합니다. InputNum=0 의 연속 입력시는 통지되지 않습니다.
Over flow 통지	ACL_INT_OVERFLOW	지정된 버퍼수 이상의 이벤트의 콜백이 지연된 경우에 통지됩니다.
Time out 통지	ACL_INT_TIMEOUT	1 프레임이 입력이 지정시간 동안 반응하지 않은 경우통지합니다.

Table 6-12 : 끼워넣기 정의

6.13.1 이벤트 사용방법

이벤트 통지에 의한 끼워넣기의 대기방법을 설명합니다.

■설정

AcapSetEvent 함수에서 대기를 원하는 끼워넣기를 설정합니다.

복수의 통지 설정을 할 수 있습니다만, 통지 이벤트는 1개씩 등록해 주세요.

■대기

AcapSetEvent 함수에서 설정한 통지설정은 AcapWaitEvent 함수에서 대기 가능합니다.

■소프트웨어 설정예

아래에 연속 입력「입력 개시 통지」와「1 프레임 입력완료 통지」를 대기 시킬 경우의 샘플 코드를 설명합니다.

〈샘플 코드〉

```
int rtn;
HANDLE hDevice;

// APX-3324 을 오픈
hDevice = AcapOpen("APX-3324", 0, INPUT_CH);

// ini 파일로 초기화
rtn = AcapSelectFile(hDevice, INPUT_CH, "camera.ini", 0);

// 「입력개시 통지」와 「1 프레임 입력완료 통지」를 설정
rtn = AcapSetEvent(hDevice, INPUT_CH, ACL_INT_GRABSTART, TRUE);
rtn = AcapSetEvent(hDevice, INPUT_CH, ACL_INT_FRAMEEND, TRUE);

// 취득 개시(연속 입력)
rtn = AcapGrabStart(hDevice, INPUT_CH, 0);

// 영상입력 개시를 대기
int GrabStartTimeout_ms = 1000;
rtn = AcapWaitEvent(hDevice, INPUT_CH, ACL_INT_GRABSTART, GrabStartTimeout_ms);

// 1 프레임 입력완료를 대기
int FrameEndTimeout_ms = 1000;
rtn = AcapWaitEvent(hDevice, INPUT_CH, ACL_INT_FRAMEEND, FrameEndTimeout_ms);
```

■대기의 중지(Abort)

AcapWaitEvent 함수는 끼워넣기를 수신하던지 Time out 이 발생할 때까지 처리를 끝내지 않습니다.

다른 Thread 에서 ACL_INT_ABORT 그래프를 사용하여 AcapSetEvent 함수를 실행하면 AcapWaitEvent 함수의 대기처리를 강제종료 할 수 있습니다.

■소프트웨어 설정예

아래에 ACL_INT_FRAMEEND 대기중의 Thread(Thread_WaitFrameEnd)를 강제종료 시킬 경우의 샘플 코드를 설명 합니다.

〈샘플코드〉

```
// 강제 종료하는 이벤트와 ACL_INT_ABORT 의 OR 를 지정
rtn = AcapSetEvent(hDevice, INPUT_CH, ACL_INT_FRAMEEND | ACL_INT_ABORT, TRUE);

// 「1 프레임 입력완료 통지」를 대기하는 Thread
void Thread_WaitEvent( void* pParam )
{
    // 생략
    // 1 프레임 입력완료를 대기
    int FrameEndTimeout_ms = 10 * 1000;
    rtn = AcapWaitEvent(hDevice, INPUT_CH, ACL_INT_FRAMEEND, FrameEndTimeout_ms);
    if(rtn == ACL_RTN_ERROR) {
        ACAPERRORINFO AcapErrInfo;
        AcapGetLastErrorCode(&AcapErrInfo, 0);
        if(AcapErrInfo.dwBoardErrorCode==ACL_ABORT_EVENT_WAIT) {
            // Abort 되면 에러로 함수가 돌아옵니다.
            printf("AcapWaitEvent was aborted. %n");
        }
    }
}
```

■해제

설정한 이벤트를 대기할 필요가 없을 경우 설정을 해제할 필요가 있습니다.

설정의 해제는 AcapSetEvent 함수의 제 4 인수 bEventEnable 를 FALSE 에서 실행합니다.

■소프트웨어 설정예

아래에 이벤트 설정을 해제 시킬경우의 샘플 코드를 설명합니다.

〈샘플 코드〉

```
// 이벤트 설정을 해제
rtn = AcapSetEvent(hDevice, INPUT_CH, ACL_INT_FRAMEEND, FALSE);
```

AcapWaitEvent 함수의 실행 중(대기 중)에는 해제할 수 없으므로 에러가 발생합니다.

이 경우「대기 중지」를 실행하고 나서 해제해 주세요.

6.13.2 콜백함수의 상용방법

콜백 함수에 의한 통지의 대기방법을 설명합니다.

등록함수는 AcapRegistCallback 함수와 AcapRegistCallbackEx 함수 2 종류가 있습니다.



AcapRegistCallback 함수와 AcapRegistCallbackEx 함수는 둘중 하나만 사용 가능합니다.

■콜백 함수의 Protptype

AcapRegistCallback 함수에 등록하는 콜백 함수는 아래와 같은 Prototype 이어야만 합니다.

```
void CALLBACK CallbackFunc ( int nChannel, DWORD dwEvent, int nFrameNo, int nMemoryNo);
nChannel : 통지를 검출한 채널
dwEvent : 검출된 통지(ACL_INT_FRAMEEND 등)
nFrameNo : 현재의 「입력완료된 프레임수」(취득 정지중 최후에 취득한 결과)
nMemoryNo : 현재의 「최후에 입력이 완료된 메모리 번호 (1 ~)」(취득정지 시에는 최후에 취득한 결과)
```

AcapRegistCallbackEx 함수에 등록하는 콜백 함수는 아래와 같은 Prototype 이어야만 합니다.

```
void CALLBACK CallbackExFunc ( int nChannel, DWORD dwEvent, int nFrameNo, int nMemoryNo, void*
pParam);
nChannel : 통지를 검출한 채널
dwEvent : 검출된 통지(ACL_INT_FRAMEEND 등)
nFrameNo : 현재의 「입력완료된 프레임수」(취득 정지중 최후에 취득한 결과)
nMemoryNo : 현재의 「최후에 입력이 완료된 메모리 번호 (1 ~)」(취득정지 시에는 최후에 취득한 결과)
pParam : AcapRegistCallbackEx 함수에서 제 5 인수에 걸친 Address
※주의 : pParam 은 사용자 버퍼의 포인터가 됩니다. 등록시에 전달된 메모리가 오픈되어 버리면
세그멘테이션 위반의 원인이 됩니다.
```

■ 콜백 함수의 등록

AcapRegistCallback 함수/AcapRegistCallbackEx 함수에서 끼워넣기 통지용 콜백함수를 등록합니다.

콜백 함수는 1 개만 등록 가능하지만, 복수의 끼워넣기를 지정할 수 있습니다.

■ 소프트웨어 설정예

1ch 에서「1 프레임 입력 완료 통지」발생시에 실장된 콜백 함수를 등록할 경우의 샘플코드를 설명합니다.

〈샘플 코드〉

```
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);
    // 1 프레임 입력 완료시의 처리
}

// Main routine 으로 콜백 함수의 등록
int rtn;
rtn = AcapRegistCallback(hDevice, INPUT_CH, ACL_INT_FRAMEEND, (EVENT_FUNC)&CallbackFunc);

// 연속 입력 개시
rtn = AcapGrabStart(hDevice, INPUT_CH, 0);

// 1 프레임이 입력될 때마다 CallbackFunc 를 불러들입니다.

<< 출력결과 >> ※전송 메모리 수=3 의 경우
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]
```

■ 소프트웨어 설정예

1ch 로「입력개시 통지」「1 프레임 입력완료 통지」「입력종료 통지」발생시에 실장되는 확장 콜백

함수를 등록 할 경우의 샘플 코드를 설명합니다.

〈샘플 코드〉

```
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);
}
```

// Main routine 으로 확장 콜백함수의 등록

int rtn;

DWORD dwEvent = ACL_INT_GRABSTART | ACL_INT_FRAMEEND | ACL_INT_GRABEND; **// OR 에서 복수지정**

DWORD Param = 123456; **// 제 5 인수에 걸쳐서 파라미터**

rtn = AcapRegistCallbackEx(hDevice, INPUT_CH, dwEvent, (EVENT_FUNC_EX)&CallbackFuncEx, (void*)&Param);

// 5 프레임 입력

rtn = AcapGrabStart(hDevice, INPUT_CH, 5);

// 각 프레임의 입력개시 시에 ACL_INT_GRABSTART

// 각 프레임의 입력완료 시에 ACL_INT_FRAMEEND

// 5 프레임 입력완료 시에 ACL_GRABEND 의 콜백함수를 불러들입니다.

<< 출력결과 >> ※전송 메모리 수=3 의 경우

```
CallbackFuncEx called. nChannel[1] dwEvent[1] nFrameNo[0] nMemoryNo[3] *pParam[123456] ※주의
CallbackFuncEx called. nChannel[1] dwEvent[2] nFrameNo[1] nMemoryNo[1] *pParam[123456]
CallbackFuncEx called. nChannel[1] dwEvent[1] nFrameNo[1] nMemoryNo[1] *pParam[123456]
CallbackFuncEx called. nChannel[1] dwEvent[2] nFrameNo[2] nMemoryNo[2] *pParam[123456]
CallbackFuncEx called. nChannel[1] dwEvent[1] nFrameNo[2] nMemoryNo[2] *pParam[123456]
CallbackFuncEx called. nChannel[1] dwEvent[2] nFrameNo[3] nMemoryNo[3] *pParam[123456]
CallbackFuncEx called. nChannel[1] dwEvent[1] nFrameNo[3] nMemoryNo[3] *pParam[123456]
CallbackFuncEx called. nChannel[1] dwEvent[2] nFrameNo[4] nMemoryNo[1] *pParam[123456]
CallbackFuncEx called. nChannel[1] dwEvent[1] nFrameNo[4] nMemoryNo[1] *pParam[123456]
CallbackFuncEx called. nChannel[1] dwEvent[2] nFrameNo[5] nMemoryNo[2] *pParam[123456]
CallbackFuncEx called. nChannel[1] dwEvent[4] nFrameNo[5] nMemoryNo[2] *pParam[123456]
```

**※注 : 최초의 GRABSTART 통지는 1 프레임도 입력이 완료되지 않기 때문에
nFrameNo 은 0, nMemoryNo 는 등록 메모리 수 (3) 이 취득 됩니다.**

■ 콜백 함수의 등록해제

AcapRegistCallback 함수/AcapRegistCallbackEx 함수의 제 4 인수를 NULL 로 실행하면 등록되어 있는 콜백함수를 해제 할 수 있습니다.

■ 소프트웨어 설정예

콜백 함수를 다른 통지로 등록변경을 할 경우의 샘플 코드를 설명합니다.

<샘플 코드>

```
// 「입력개시 통지」를 등록
int rtn;
rtn = AcapRegistCallback(hDevice, INPUT_CH, ACL_INT_GRABSTART, (EVENT_FUNC)&CallbackFunc);

// 「입력개시 통지」를 등록
rtn = AcapRegistCallback(hDevice, INPUT_CH, ACL_INT_GRABSTART, NULL);

// 1 프레임 입력완료를 등록
rtn = AcapRegistCallback(hDevice, INPUT_CH, ACL_INT_FRAMEEND, (EVENT_FUNC)&CallbackFunc);
```

6.14 끼워넣기 이벤트 · 콜백 함수의 주의점

【Ver.6.2.0 이후 개선점】

이전의 콜백처리에서 아래 문제가 개선되었습니다.

- 콜백 함수가 실행중에 동일 이벤트가 복수로 통지된 경우에 콜백의 실행 회수가 줄어듦 (이벤트의 손실)
- 입력개시와 프레임 End 간에 아주 짧은 경우는 콜백을 부르는 순서가 반대가 됨.

[통지의 개념]

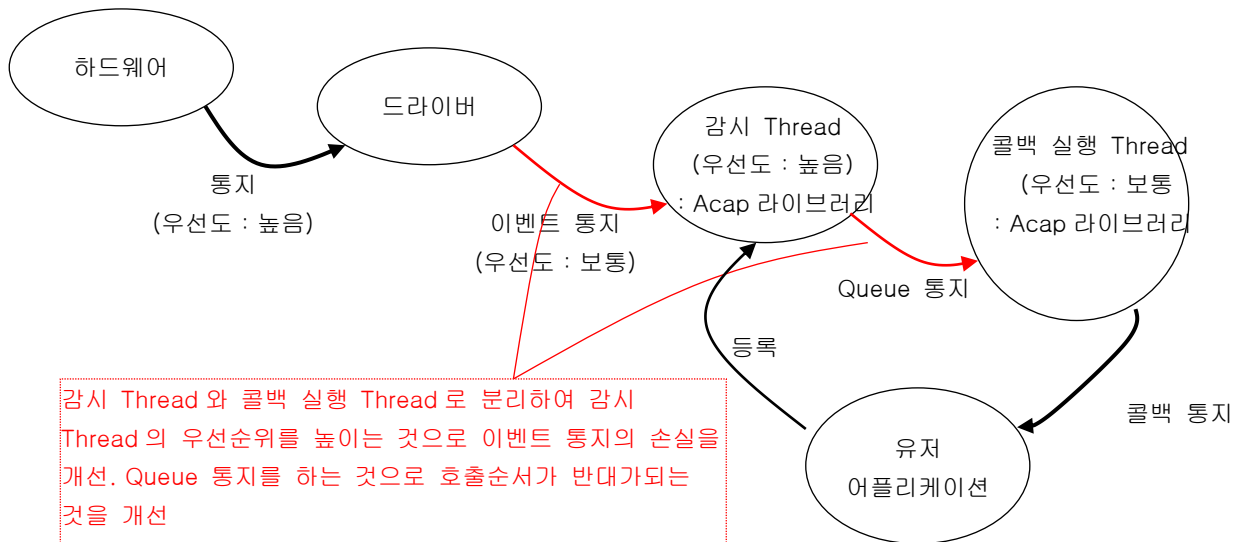


Figure 6-41 : 통지의 개념

[주의점]

콜백 실행은 순차처리로 되어 있기 때문에 콜백의 처리자체가 취득시간에 비해 아주 긴 경우와 버퍼수가 적은 경우는 콜백 회수의 손실이 발생할 수 있습니다.

이러한 상황에서는 아래와 같은 처리가 필요합니다.

- 전송 메모리 매수를 복수로 설정함.
- 콜백처리시에 Thread를 일으켜서, 실제처리를 여기서 실행함.
- CPU 부하를 경감
- PC 성능을 높임
- Frame rate를 낮춤
- 1 화면의 전송량(해상도)를 낮춤

콜백의 손실이 발생할 경우, 새롭게 준비된 Over flow(ACL_INT_OVERFLOW)의 이벤트가 발생함으로 필요에 따라서는 실장하여 주십시오

第 7 章 설정 ID 일람

AcapSetInfo 함수 AcapGetInfo 함수의 동작을 지정하는「설정 ID」를 설명합니다.

설정 ID 는 AcapSetInfo 함수, AcapGetInfo 함수의 제 3 인수로 설정합니다.

상세내용은 SDK-AcapLib2 라이브러리 메뉴얼(SDK-AcapLib2 Library Manual(J2OM080155*))을 참조 해주세요.

<마크 개요>

- : 대응
- : 미대응

설정내용	정의	Set	Get	AcapSetInfo 의 설정값/AcapGetInfo 의 복귀값	
X 사이즈	ACL_X_SIZE	○	○	8bit/1tap	16 ~ 65,520(16 의배수)
				8bit/2tap	16 ~ 65,520(16 의배수)
				8bit/3tap	48 ~ 65,520(48 의배수)
				8bit/4tap	32 ~ 32,768(32 의배수)
				8bit/8tap	64 ~ 65,472(64 의배수)
				8bit/10tap	80 ~ 65,520(80 의배수)
				10bit ~ 16bit/1tap	8 ~ 8,192(8 의배수)
				10bit ~ 12bit/2tap	8 ~ 16,384(8 의배수)
				10bit ~ 12bit/4tap	32 ~ 32,768(32 의배수)
				10bit/8tap	64 ~ 65,472(64 의배수)
				24bitRGB/1tap	16 ~ 8,192(16 의배수)
				30bitRGB/1tap	8 ~ 8,192(8 의배수)
				32bitRGB/1tap	8 ~ 8,192(8 의배수)
				32bitRGB/2tap	8 ~ 16,384(8 의배수)
Y 사이즈	ACL_Y_SIZE	○	○	1 ~ 16,777,215	
X 지연	ACL_X_DELAY	○	○	0 ~ 65,535	
Y 지연	ACL_Y_DELAY	○	○	0 ~ 1,023	
YTotal 사이즈	ACL_Y_TOTAL	○	○	0 ~ 16,777,215	
카메라 입력 Bit 수	ACL_CAM_BIT	○	○	8/10/12/14/16/24/32	
보드 입력 Bit 수	ACL_BOARD_BIT	-	○	8/16/24/32	
Shift 회수	ACL_PIX_SHIFT	○	○	0/2/4/6/8	
Time out 시간[ms]	ACL_TIME_OUT	○	○	대기 없음	0
				Time out 없음	-1
				mm 秒	상기 이외
메모리 확보 수	ACL_MEM_NUM	○	○	1 ~ 16,383	
노광신호(CC)출력논리	ACL_EXP_POL	○	○	부 논리	0
				정 논리	1

노광신호(CC)출력주기 [1us 단위]	ACL_EXP_CYCLE	○	○	0 ~ 429,496,729	
노광신호(CC)출력시간 [1us 단위]	ACL_EXPOSURE	○	○	0 ~ 429,496,728	
노광신호(CC)출력주기 [100ns 단위]	ACL_EXP_CYCLE_EX	○	○	0 ~ 4,294,967,295	
노광신호(CC)출력시간 [100ns 단위]	ACL_EXPOSURE_EX	○	○	0 ~ 4,294,967,294	
노광신호(CC)선택	ACL_EXP_CC_OUT	○	○	1 ~ 4	
Rolling 셔터	ACL_ROLLING_SHUTTER	○	○	Rolling 셔터 미사용	0
				Rolling 셔터 사용	1
외부 트리거 신호 선택	ACL_EXT_EN	○	○	외부트리거 미사용	0
				TTL	1
				RS-422	2
				신규 차동(Diff)	3
				포토 커블러 입력	4
외부 트리거모드	ACL_EXT_MODE	○	○	연속 외부 트리거 모드	0
				단발 외부 트리거 모드	1
외부 트리거 검출무효시간 [1us 단위]	ACL_EXT_CHATTER	○	○	0 ~ 65,535	
외부 트리거 검출지연시간 [1us 단위]	ACL_EXT_DELAY	○	○	0 ~ 4,294,967,295	
엔코더 선택	ACL_ENC_EN	○	○	엔코더 미사용	0
				상대위치 엔코더 사용	1
				절대위치 엔코더 사용	2
엔코더 기동방법	ACL_ENC_START	○	○	외부 트리거를 엔코더로 사용하지 않음	0
				외부 트리거를 엔코더의 기동에 사용함	1
				외부 트리거를 엔코더의 동일 펄스로써 사용함.	2
엔코더 동작 모드	ACL_ENC_MODE	○	○	엔코더 스캔 모드	0
				엔코더 라인 선택 모드	1
엔코더 입력 펄스 선택	ACL_ENC_PHASE	○	○	AB 상	0
				A 상	1
엔코더 입력 펄스 방향	ACL_ENC_DIRECTION	○	○	CW	0
				CCW	1
엔코더 Z 상 입력	ACL_ENC_ZPHASE_EN	○	○	엔코더 기동에 Z 상을 사용하지 않음	0
				엔코더 기동시에 Z 상을 사용합니다.	1
엔코더 비교 레지스터 1	ACL_ENC_COMPARE_1	○	○	0 ~ 2,147,483,647	
엔코더 비교 레지스터 2	ACL_ENC_COMPARE_2	○	○	0 ~ 2,147,483,647	

스트로브 유효/무효 선택	ACL_STROBE_EN	○	○	스트로브 무효	0
				스트로브 유효	1
스트로브신호 출력지연시간 [1us 단위]	ACL_STROBE_DELAY	○	○	0 ~ 65,535	
스트로브 신호출력 폭 [1us 단위]	ACL_STROBE_TIME	○	○	0 ~ 65,535	
CC1 레벨(위상)	ACL_CC1_LEVEL	○	○	Low	0
				High	1
CC2 레벨(위상)	ACL_CC2_LEVEL	○	○	Low	0
				High	1
CC3 레벨(위상))	ACL_CC3_LEVEL	○	○	Low	0
				High	1
CC4 레벨(위상)	ACL_CC4_LEVEL	○	○	Low	0
				High	1
카메라 종류 선택	ACL_SCAN_SYSTEM	○	○	에어리어 센서	0
				라인센서	1
Y 방향 Reverse DMA	ACL_REVERSE_DMA	○	○	Y 방향 반전 DMA 무효	0
				Y 방향 반전 DMA 유효	1
DVAL 유효/무효서정	ACL_DVAL_EN	○	○	DVAL 무효	0
				DVAL 유효	1
입력 탭수	ACL_TAP_NUM	○	○	1/2/3	
탭 재배열	ACL_TAP_ARRANGE	○	○	재배열 무효	0
				전 탭 독립	1
				2 탭 통합	2
SyncLT 유효/무효설정	ACL_SYNC_LT	-	-	미대응	
GPIN 레벨 취득	ACL_GPIN_POL	-	-	미대응	
GP_OUT 핀 출력 신호 선택	ACL_GPOUT_SEL	○	○	캡처 플러그	미대응(향후대응예정)
				범용출력	1
GP_OUT 신호출력 논리(※1)	ACL_GPOUT_POL	○	○	Low 출력	0
				High 출력	1
입력라인 수 카운트 간격	ACL_INTR_LINE	○	○	입력라인 수를 카운트 합니다.	1 ~ Y 사이즈
				입력라인 수를 카운트 하지 않습니다.	상기 이외
시리얼 채널 설정	ACL_SERIAL_CHANNEL	-	-	미대응	
시리얼 포트 설정	ACL_SERIAL_PORT	-	-	미대응	
노광신호(CC)출력 유효/무효 설정	ACL_EXP_EN	○	○	노광신호 무효	0
				노광신호 유효	1
절대위치 엔코더 개시	ACL_ENC_ABS_START	○	○	엔코더 정지	0
				엔코더 개시	1
절대위치 엔코더 카운트 값	ACL_ENC_ABS_COUNT	-	○	-2,147,483,647 ~ 2,147,483,647	

입력 데이터 마스터 (하위 32bit)	ACL_DATA_MASK_LOWE R	○	○	0 ~ 0xFFFF_FFFF	
입력 데이터 마스터 (하위 32bit)	ACL_DATA_MASK_UPPE R	○	○	0 ~ 0xFFFF_FFFF	
상대위치 엔코더 카운트값	ACL_ENC_RLT_COUNT	-	○	-2,147,483,647 ~ 2,147,483,647	
상대위치 엔코더 총 카운트 값	ACL_ENC_RLT_ALL_COU NT	-	○	-2,147,483,647 ~ 2,147,483,647	
상대위치 엔코더 동일펄스 수	ACL_ENC_AGR_COUNT	-	○	0 ~ 4,294,967,295	
외부 트리거 무효기간 개별설정	ACL_EXT_CHATTER_SEPA RATE	○	○	High 무효기간과 Low 무효기간을 동일한 값으로 설정함	0
				High 무효기간과 Low 무효기간을 각각 개별적으로 설정함	1
외부 트리거 신호선택 (확장)	ACL_EXT_PIN_SEL	-	-	미대응	
GPIN 끼워넣기 핀 선택	ACL_GPIN_PIN_SEL	-	-	미대응	
동기 취득설정	ACL_SYNC_CH	○	○	동기설정 없음	0
				동기되는 ch 을 설정합니다	1/2
베이어 설정	ACL_BAYER_SETUP	-	-	미대응	
베이어 개시위치 패턴 설정	ACL_BAYER_GRID	-	-	미대응	
베이어 LUT 편집 개시/정지	ACL_BAYER_LUT_EDIT	-	-	미대응	
베이어 LUT 데이터 설정	ACL_BAYER_LUT_DATA	-	-	미대응	
카메라 전원 ON/OFF 제어	ACL_POWER_SUPPLY	○	○	전원 OFF	0
				전원 ON	1
카메라 전원에러 제어	ACL_POWER_STATE	○	○	카메라전원 에러상태를 클리어 합니다. [AcapSetInfo 사용시]	설정 값은 의미가 없습니다.
				에러 미 발생 [AcapGetInfo 사용시]	0
				에러 발생 [AcapGetInfo 사용시]	1 ※해당하는 에러 Bit 가 발생합니다.
스트로브 신호 출력극성	ACL_STROBE_POL	-	-	미대응	
VerticalRemap 대응 DMA	ACL_VERTICAL_REMAP	○	○	무효	0
				유효	1
CC 출력 지연설정	ACL_CC_DELAY	○	○	0 ~ 65,535	
High clp 유효/무효설정	ACL_HIGH_CLIP	-	-	미대응	
PciExpress Link 폭 취득	ACL_EXPRESS_LINK	-	○	1/4	
FPGA 버전 취득	ACL_FPGA_VERSION	-	○	현재의 FPGA 버전	

탭 방향	ACL_TAP_DIRECTION	○	○	순방향(→)	0
				역방향(←)	1
재배열 총 X 사이즈	ACL_ARRANGE_XSIZE	○	○	ACL_X_SIZE 참조	
LVAL 지연	ACL_LVAL_DELAY	○	○	0 ~ 65,535	
라인 반전	ACL_LINE_REVERSE	○	○	좌우반전 하지 않습니다.	0
				좌우 반전합니다.	1
카메라 접속상태 확인	ACL_CAMERA_STATE	-	○	카메라 미 접속	0
				카메라 접속 중	1
GPIN 레벨 취득	ACL_GPIN_POL	-	○	미대응	
보드 에러상태	ACL_BOARD_ERROR	○	○	보드 에러 상태를 클리어 합니다. ※AcapSetInfo 사용시	설정 값은 의미가 없습니다.
				에러 미 발생 ※AcapGetInfo 사용시	0
				에러 발생 ※AcapGetInfo 사용시	1
입력개시 프레임 번호설정	ACL_START_FRAME_NO	○	○	입력 개시 프레임 번호	1~
				에러	상기 이외
초기화 캔슬	ACL_CANCEL_INITIALIZE	○	○	초기화 캔슬을 하지 않습니다.	0
				초기화의 캔슬을 실행합니다.	1
노광신호(CC)출력주기 [100ns 단위]	ACL_EXP_CYCLE_EX	○	○	0 ~ 4,294,967,295	
노광신호(CC)출력시간 [100ns 단위]	ACL_EXPOSURE_EX	○	○	0 ~ 4,294,967,294	
CXP 상태 취득/ 에러 클리어	ACL_CXP_STATE	-	-	미대응	

Table 7-1 : 설정 ID 일람

※1 ACL_GPOUT_POL 사용시는 돌아오는 값의 각 bit 에 GP_OUT 신호가 할당 되어 있습니다.

bit	GP_OUT 신호의 할당
31-8	미사용
7	GP_OUT8
6	GP_OUT7
5	GP_OUT6
4	GP_OUT5
3	GP_OUT4
2	GP_OUT3
1	GP_OUT2
0	GP_OUT1

Table 7-2 : 할당 일람

第 8 章 에러 코드

AcapGetLastErrorCode 함수에서 얻을 수 있는 에러코드 값입니다.

- 공통 에러코드(dwCommonErrorCode) : AcapLib2 라이브러리 함수와 관련한 에러코드
- 보드 에러코드(dwBoardErrorCode) : 보드 고유의 에러코드
- 확장 에러코드(dwExtendErrorCode) : 드라이버 요인의 에러코드 등

8.1 공통 에러코드

ACAPERRORINFO 의「dwCommonErrorCode」에 저장됩니다.

Define 값	16 진수값	의미
ACL_NO_ERROR	0	에러는 없습니다.
ACL_BOARD_NOT_FOUND_ERROR	0xF101	보드를 찾을 수 없습니다.
ACL_BOARD_NAME_ERROR	0xF102	보드명이 틀립니다.
ACL_BOARD_NUMBER_ERROR	0xF103	지정한 보드번호가 없음
ACL_SAME_BOARDNO_ERROR	0xF104	동일한 보드번호가 여러 개 존재함
ACL_INVALID_HANDLE_ERROR	0xF105	핸들이 부정
ACL_SELECT_CHANNEL_ERROR	0xF106	지정한 채널이 부정
ACL_MEM_ALLOCATE_ERROR	0xF107	메모리 확보에 실패
ACL_INVALID_MUTEX_ERROR	0xF108	배타용 핸들이 부정
ACL_MUTEX_LOCKED_ERROR	0xF109	함수가 배타상태로 되어 있음

ACL_FILE_NOT_FOUND_ERROR	0xF10A	지정한 파일을 찾을 수 없음
ACL_EVENT_TIMEOUT_ERROR	0xF10B	이벤트 대기에서 Time out
ACL_OPEN_ERROR	0xF10C	디바이스 open 에 실패
ACL_CLOSE_ERROR	0xF10D	디바이스 Close 에 실패
ACL_INITIALIZE_ERROR	0xF10E	보드의 초기화에 실패
ACL_REFLECT_PARAM_ERROR	0xF10F	설정값 반영에 실패
ACL_SNAP_ERROR	0xF110	한 화면입력에 실패
ACL_SNAP_WAIT_ERROR	0xF111	한 화면 입력 대기에 실패
ACL_GRAB_START_ERROR	0xF112	연속입력에 실패
ACL_GRAB_STOP_ERROR	0xF113	연속입력 정지에 실패
ACL_GRAB_ABORT_ERROR	0xF114	강제정지 처리에 실패
ACL_GET_FRAMENO_ERROR	0xF115	프레임수의 취득에 실패
ACL_SET_EVENT_ERROR	0xF116	이벤트 등록에 실패
ACL_WAIT_EVENT_ERROR	0xF117	이벤트 대기에 실패
ACL_REGIST_CALLBACK_ERROR	0xF118	콜백함수 등록에 실패
ACL_SET_ENCODER_ERROR	0xF119	엔코더 설정 실패
ACL_GET_ENCODER_ERROR	0xF11A	엔코더 취득 실패
ACL_SET_SHUTTER_TRIGGER_ERROR	0xF11B	트리거 설정 실패
ACL_SET_LINE_TRIGGER_ERROR	0xF11C	라인트리거 설정 실패
ACL_SET_EXTERNAL_TRIGGER_ERROR	0xF11D	외부 트리거 설정 실패
ACL_ABORT_EXTERNAL_WAIT	0xF11E	외부 트리거 대기 중지함
ACL_SET_STROBE_ERROR	0xF11F	스트로브 설정 실패
ACL_SET_GPIO_ERROR	0xF120	GPIO 설정 실패

ACL_IMAGE_CONVERT_ERROR	0xF121	영상 Bit 변환 실패
ACL_SET_CAMERALINK_ERROR	0xF122	CameraLink 설정 실패
ACL_SET_DMA_ERROR	0xF123	DMA 설정 실패
ACL_SERIAL_OPEN_ERROR	0xF124	시리얼 Open 에 실패
ACL_SERIAL_CLOSE_ERROR	0xF125	시리얼 Close 에 실패
ACL_SERIAL_WRITE_ERROR	0xF126	시리얼 송신(쓰기)에 실패
ACL_SERIAL_READ_ERROR	0xF127	시리얼 수신(읽기)에 실패
ACL_SERIAL_SET_PARAM_ERROR	0xF128	시리얼 파라미터 설정 실패
ACL_SERIAL_GET_PARAM_ERROR	0xF129	시리얼 파라미터의 취득 실패
ACL_SERIAL_HANDLE_ERROR	0xF12A	시리얼 핸들 부정
ACL_SERIAL_SELECT_PORT_ERROR	0xF12B	지정된 포트는 사용할 수 없음
ACL_SET_BUFFER_ADRS_ERROR	0xF12C	버퍼 어드레스의 설정에 실패
ACL_GET_BUFFER_ADRS_ERROR	0xF12D	버퍼 어드레스 취득에 실패
ACL_SET_CAMERALINK_EX_ERROR	0xF12E	CameraLink 설정(확장)에 실패
ACL_ABORT_EVENT_WAIT	0xF12F	이벤트 대기 중지함
ACL_GET_BOARDINFO_ERROR	0xF130	보드정보 취득에 실패
ACL_SET_INFO_ERROR	0xF131	값 설정에 실패
ACL_GET_INFO_ERROR	0xF132	값 취득에 실패
ACL_SELECT_FILE_ERROR	0xF133	ini 파일 불러들임 실패
ACL_GET_FILE_VERSION_ERROR	0xF134	파일 버전 취득 실패
ACL_REGIST_CALLBACK_EX_ERROR	0xF135	콜백함수 등록(확장)에 실패
ACL_ROI_CONVERT_ERRPR	0xF136	ROI 영상변환에 실패
ACL_GET_CAMERALINK_ERROR	0xF137	CameraLink 설정 취득에 실패

ACL_GET_DMA_ERROR	0xF138	DMA 설정 취득 실패
ACL_GET_LINE_TRIGGER_ERROR	0xF139	라인 트리거 취득에 실패
ACL_GET_EXTERNAL_TRIGGER2_ERROR	0xF13A	외부 트리거 설정 실패
ACL_SET_DMA_OPTION_EX_ERROR	0xF13B	비선형 DMA 설정 실패
ACL_GET_DMA_OPTION_EX_ERROR	0xF13C	비선형 DMA 설정취득 실패
ACL_GET_GPOUT_ERROR	0xF13D	GPOUT 상태취득 실패
ACL_PARAM_1_ERROR	0xF201	파라미터 에러 (제 1 인수)
ACL_PARAM_2_ERROR	0xF202	파라미터 에러 (제 2 인수)
ACL_PARAM_3_ERROR	0xF203	파라미터 에러 (제 3 인수)
ACL_PARAM_4_ERROR	0xF204	파라미터 에러 (제 4 인수)
ACL_PARAM_5_ERROR	0xF205	파라미터 에러 (제 5 인수)
ACL_PARAM_6_ERROR	0xF206	파라미터 에러 (제 6 인수)
ACL_PARAM_7_ERROR	0xF207	파라미터 에러 (제 7 인수)
ACL_PARAM_8_ERROR	0xF208	파라미터 에러 (제 8 인수)
ACL_PARAM_9_ERROR	0xF209	파라미터 에러 (제 9 인수)
ACL_PARAM_10_ERROR	0xF20A	파라미터 에러 (제 10 인수)
ACL_PARAM_11_ERROR	0xF20B	파라미터 에러 (제 11 인수)

Table 8-1 : 공통 에러코드

8.2 보드 에러 코드

ACAPERRORINFO 의 「dwBoardErrorCode」에 저장되어 있습니다.

또한 *에는 보드 번호(APX-3364의 경우는 C)가 입력됩니다.

예) Time out 이 발생한 경우에는 「0xFB08」가 표시됩니다

공통에러 define 값	16 진수값	의미
ACL_3300_NO_ERROR	0	에러 없음
ACL_3300_DRIVER_ERROR	0xF*01	드라이버 에러
ACL_3300_INVALID_PARAM	0xF*02	파라미터가 부정
ACL_3300_NO_DEVICE	0xF*03	디바이스를 찾을 수 없음
ACL_3300_EVENT_ERROR	0xF*04	이벤트 등록에러
ACL_3300_INVALID_HANDLE	0xF*05	핸들 부정
ACL_3300_INVALID_CAPTURE	0xF*06	영상 취득부정
ACL_3300_CAPTURE_RUNNING	0xF*07	영상 취득 중
ACL_3300_TIME_OUT	0xF*08	Time out
ACL_3300_INSUFFICIENT_CONFIG	0xF*09	설정이 부족함
ACL_3300_NOT_CONNECTED_CAMERA	0xF*0A	카메라가 접속되어 있지 않음
ACL_3300_NULL_POINTER_ERROR	0xF*0B	지정된 포인터가 NULL 입니다.
ACL_3300_X_SIZE_ERROR	0xF*0C	잘못된 X 사이즈가 설정되어 있음
ACL_3300_Y_SIZE_ERROR	0xF*0D	잘못된 Y 사이즈가 설정되어 있음
ACL_3300_BITSHIFT_ERROR	0xF*0E	잘못된 Bit shift 가 설정되어 있음
ACL_3300_TAPNUM_ERROR	0xF*0F	잘못된 탭수가 설정되어 있음

ACL_3300_SET_BUFFER_ERROR	0xF*10	버퍼 등록에 실패
ACL_3300_CAMERA_KIND_ERROR	0xF*11	카메라 종류 (Area, Line) 이 잘못되어있음
ACL_3300_GRAB_ABORT	0xF*12	입력을 강제로 정지함
ACL_3300_CH_NOT_OPENED	0xF*13	설정된 채널은 오픈되어 있지 않음
ACL_3300_FIFO_ERROR	0xF*14	FIFO 에러가 발생함
ACL_3300_DMA_ERROR	0xF*15	DMA 에러가 발생함
ACL_3300_GET_REGISTRY_KEY_ERROR	0xF*16	레지스터 키 취득 실패
ACL_3300_DLL_DRIVER_ERROR	0xF*17	DLL 에러
ACL_3300_INTERNAL_LIBRARY_ERROR	0xF*18	라이브러리 에러
ACL_3300_UNKNOWN_ERROR	0xF*19	알 수 없는 에러
ACL_3300_INVALID_CAMBIT	0xF*20	카메라 입력 Bit 가 부정
ACL_3300_OPEN_ALREADY	0xF*21	다른 프로세스가 열려 있음
ACL_3300_INVALID_EVENT	0xF*24	이벤트가 부정
ACL_3300_SETEVENT_ALREADY	0xF*25	이미 이벤트가 등록되어 있음
ACL_3300_CREATEEVENT_FAILED	0xF*26	CreateEvent 에 실패
ACL_3300_WAITEVENT_ALREADY	0xF*27	이미 대기상태 또는 ABORT 처리 중
ACL_3300_EVENT_NOT_IDLE	0xF*28	대기중임으로 해제 할 수 없습니다.
ACL_3300_CALLBACK_ALREADY	0xF*29	이미 콜백이 등록되어 있습니다.
ACL_3300_INVALID_CC	0xF*2A	CC 주기 및 폭의 관계가 부정
ACL_3300_INVALID_BUFFER	0xF*2B	버퍼 부정
ACL_3300_INVALID_XDELAY	0xF*2C	Xdelay 가 부정
ACL_3300_INVALID_YDELAY	0xF*2D	Ydelay 가 부정
ACL_3300_INVALID_POWER_STATE	0xF*2E	카메라 전원상태가 이상

ACL_3300_POWER_STATE_ERROR	0xF*2F	카메라 전원상태 에러
ACL_3300_POWER_ON_TIMEOUT	0xF*30	카메라 전원을 ON 하였지만 지정시간 내에 카메라 클럭을 검출할 수 없었음
ACL_3300_POWER_OFF_TIMEOUT	0xF*31	카메라 전원을 OFF 하였지만 지정시간경과후에 도 카메라 클럭을 검출함
ACL_3300_TAP_SEL_ERROR	0xF*32	Tap/Bit/Arrange 의 관계가 부정
ACL_3300_ARRANGE_X_ERROR	0xF*33	ArrangeXSize 가 부정
ACL_3300_NOT_SUPPORT	0xF*FF	지원 되고 있지 않음

Table 8-2 : 보드 에러 코드

8.3 확장 에러코드 (Windows)

ACAPERRORINFO 의「dwExtendErrorCode」에 저장되어 있습니다.

에러 발생상태에 따라 입력된 값의 의미가 다릅니다.

발생상태	상세	확장에러 코드값의 의미
내부 라이브러리 에러	보드 에러코드가 아래인 경우 ACL_3300_INTERNAL_LIBRARY_ERROR	확장 에러코드는 DWORD 값으로 표시됩니다. 주로 각 드라이버가 출력된 에러코드가 저장됩니다. 「드라이버가 출력하는 확장 에러코드」를 참조해 주세요.
버퍼 관련에러	보드 에러코드가 아래인 경우 ACL_3300_SET_BUFFER_ERROR	확장 에러코드는 int 형의 부수로 표시됩니다. 주로 버퍼등록시의 설정이상을 표시하는 에러 코드가 저장됩니다. 「버퍼관련 확장 에러코드」를 참조해 주세요.
인수관련 에러	보드 에러코드가 아래인 경우 ACL_3300_INVALID_PARAM	확장 에러코드는 DWORD 값으로 표시됩니다. 함수의 인수에서 지정된 값이 부정인 경우에 부정인수의 번호가 저장됩니다. ACL_PARAM_n_ERROR (n=부정인수 번호)
기타 에러	분류 등이 아닌, 확장정보가 필요한 경우	각 함수 설명의「Extend」에 적혀있는 값이 들어옵니다. 그 외의 확장에러코드가 발생한 경우에, 상세 정보가 필요한 경우에는 당사로 연락 바랍니다

Table 8-3 : 확장 에러코드 발생 상황

8.3.1 드라이버가 출력하는 확장 에러코드

「치명적인 에러」라고 기재된 에러에 대해서는 물리적인 변경(PC 의 변경, 메모리의 증설 등)이 필요한 경우가 있습니다. 이런 에러가 발생한 경우에는, 재부팅후에 어플리케이션을 다시 실행해 주세요. 동일한 에러가 발생한 경우에는 당사로 연락 바랍니다.

8.3.2 버퍼관련 확장 에러코드

버퍼 등록처리에서 문제가 발생한 경우에 저장되는 에러 코드입니다.

AcapSelectFile 함수와 AcapReflectParam 함수를 실행한 경우와 AcapSetBufferAddress 함수 AcapSetInfo 함수 등에서 설정반영(초기화)을 실행한 경우등에 주로 발생합니다.

정의 값	값	내용
ACL_BUFFINFO_ERR_NUMOFMEM	-1	설정하려고 하는 버퍼번호와 Acap 라이브러리가 확보하고 있는 버퍼수에 불일치가 발생하였습니다. 예 : ACL_MEM_NUM 가 4 인데 5 번째 버퍼를 등록함
ACL_BUFFINFO_ERR_ALLOCATE	-2	Acap 라이브러리 내부메모리 확보에 실패 1 프레임 사이즈(X 사이즈 x 라인 수)와 메모리 수 (ACL_MEM_NUM)가 너무 커서 발생함
ACL_BUFFINFO_ERR_EXIST	-3	등록하려고 하는 버퍼번호가 이미 등록된 경우에 발생함 또한「버퍼 분할 DMA」을 실행한 경우 Region 구형에 공백이 존재하고 있는 경우에도 이런 에러가 발생합니다.
ACL_BUFFINFO_ERR_XSIZE	-4	등록한 버퍼 X 사이즈 값이 부정한 경우에 발생
ACL_BUFFINFO_ERR_YSIZE	-5	등록한 버퍼 Y 사이즈 값이 부정한 경우에 발생
ACL_BUFFINFO_ERR_BITWIDTH	-6	등록하는 버퍼 Bit 폭이 부정한 경우에 발생.
ACL_BUFFINFO_ERR_PTRNULL	-7	등록하는 버퍼의 포인터가 NULL 인 경우에 발생
ACL_BUFFINFO_ERR_NO_BUFFER	-8	지정한 버퍼가 존재하지 않는 경우에 발생 또한, 버퍼 어드레스 취득시에 포인터 저장용 버퍼가 존재하지 않는 경우에 발생함.
ACL_BUFFINFO_ERR_REGION_SIZE	-9	「비선형 DMA(ROI 전송)」과「버퍼 분할 DMA」등으로 지정한 Region 구조체의 구형사이즈가 부정한 경우에 발생함
ACL_BUFFINFO_ERR_REGION_PTRNULL	-10	「비선형 DMA(ROI 전송)」과「버퍼분할 DMA」등으로 지정한 Region 구조체의 버퍼 포인트가 NULL 인 경우에 발생함
ACL_BUFFINFO_ERR_REGION_NUM	-11	「버퍼 분할 DMA」에서 Region 구조체의 수가 2 이하인 경우 발생합니다.
ACL_BUFFINFO_ERR_INITDMA	-12	드라이버가 초기화되지 않는 상태의 버퍼가 등록되어 있음 아래 원인이 예상됨
ACL_BUFFINFO_ERR_SETBUFFER	-13	원인 1 : 버퍼 선두어드레스의 얼라인먼트 제한이 지켜지지 않음 원인 2 : X 사이즈의 얼라인먼트 제한이 지켜지지 않음 원인 3 : 복수의 버퍼 사이즈가 불일치 원인 4 : 사이즈정보와 실제 버퍼 사이즈가 불일치
ACL_BUFFINFO_ERR_SIZEOF	-14	AcapSetDmaOptionEx 함수의 제 4 인수에서 지정한 구조체 멤버 SizeOf 값이 부정임.

Table 8-4 : 버퍼 관련 확장 에러 코드

펌웨어 갱신이력

버전	개정일	개정내용
0x11	2013.4.19	初版

개정이력




판	개정일	개정내용
A	2013.3.27	初版

お問い合わせ

お問い合わせは下記までお願いいたします。

AVAL DATA CORPORATION

本社 町田事業所 営業部 〒194-0023 東京都町田市旭町 1 丁目 25 番 10 号		042-732-1030
		042-732-1032

技術的なお問い合わせ ハードウェアやソフトウェア等、製品に関する技術的なご質問		support@avaldata.co.jp
価格・仕様などのお問い合わせ 製品の価格や基本的な仕様、在庫等、製品に関する一般的なご質問		sales@avaldata.co.jp
製品のご案内		http://www.avaldata.co.jp
お問い合わせ受付時間：月曜日～金曜日（祝祭日を除く）/ 9:00～17:00		

APX-3326 User's Manual

2013年4月19日

初版

編集 株式会社アバールデータ

発行 株式会社アバールデータ

OM13000004A

© 2013 AVALDATA CORPORATION