binho

애플리케이션 노트

Binho Pulsar를 사용한 SPI 기반 CMIS 광 모듈 브링업

AN0010

문서 정보

키워드	AN0010, Pulsar, Supernova, CMIS, OIF, SPIMCI, SPI, 코패키지드 옵틱스, CPO, 흐름 제어, ACK, NACK, 칩 셀렉트, 광 모듈
요약	SPIMCI는 코패키지드 옵틱스(CPO)를 위해 CMIS 개정 5.3에서 추가된 SPI 관리 인터페이스로, 응답(acknowledgement)을 00h로 인코딩합니다. 따라서 유휴 버스, 전원이 꺼진 모듈, 또는 한 번도 설정되지 않은 타깃은 완벽한 응답과 그 뒤를 잇는 0 데이터를 반환하며, 성공한 것처럼 보이는 브링업은 아무것도 증명하지 못합니다. 두 번째 장애물이 문제를 더 복잡하게 만듭니다. 흐름 제어 바이트 수는 와이어에 나타나지 않으며, 그 값을 알기 전에는 읽을 수 없는 레지스터에서 도출됩니다. 이 애플리케이션 노트는 Binho Pulsar 또는 Supernova에서 무응답 버스에 속지 않는 방식으로 SPIMCI 링크를 브링업하고, 모듈 자체의 체크섬으로 두 미지수를 확정하는 명령을 제공합니다.

Binho Inc.
42 Dore Street, Unit A, San Francisco, CA 94103

개정 1.0
6 NOVEMBER 2026

> **검증 범위**. 이 문서의 사양 내용은 출처를 확인하고 검증했습니다. 섹션 6의 절차는 아직 SPIMCI 모듈에서 실행되지 않았습니다. SPIMCI 펌웨어가 없는 보드에서 실행되었으며, 이것이 섹션 8.2에서 보고하는 내용이자 섹션 6.2의 체크섬 가드를 수정하게 된 계기입니다. 섹션 8에서는 미완료 사항을 정확히 설명합니다.

1 소개

SPIMCI 모듈을 연결하고 읽기를 발행하면 잘 동작합니다. 트랜잭션이 완료되고, 응답이 있으며, 데이터도 돌아옵니다.

그런데 모듈에는 전원이 공급되지 않은 상태입니다.

SPIMCI는 ACK를 **00h**로 인코딩하므로(B.3.7.1.2), 모두 0인 값은 유효한 응답과 그 뒤를 잇는 0 데이터입니다. Binho는 벤치에서 바로 이 상황을 재현했습니다. 읽기가 완료되고, 응답이 확인되었으며, 데이터는 `00 00`이었습니다. 완료된 SPIMCI 트랜잭션은 그 자체만으로는 결코 모듈이 존재한다는 증거가 아닙니다.

이 노트는 무응답으로는 위장할 수 없는 방식으로 SPIMCI 링크를 브링업하며, 그 과정에서 SPIMCI가 호스트에 안겨 주는 두 번째 문제도 해결합니다.

1.1 범위

이 문서를 다 읽으면 하드웨어를 갖춘 독자는 모듈이 구현하는 흐름 제어 바이트 수와 R/Wn 극성을 확정하고, 식별 정보를 읽고, Page 00h 체크섬을 검증할 수 있게 됩니다. 그 과정에서 존재하지 않는 모듈이 만들어 낼 수 있는 중간 결과는 하나도 받아들이지 않습니다.

이 문서는 CMIS 부록 B.3의 SPI 관리 인터페이스를 다룹니다. 이는 규범적이며 개정 5.3에서 도입되었습니다. AN0008은 레지스터 모델을 다루며 이 문서의 전제 조건입니다. 코패키지드 옵틱스 시스템 설계는 범위에서 제외됩니다.

1.2 적용 디바이스

표 1. 적용 디바이스

항목	역할	근거
Binho Pulsar	호스트 어댑터, SPI 컨트롤러	이 노트에서는 아직 검증되지 않음
Binho Supernova	호스트 어댑터, SPI 컨트롤러	컨트롤러로 검증됨, 섹션 8.1. SPIMCI 타깃은 없었음
SPIMCI 인터페이스를 갖춘 모든 CMIS 모듈	타깃	**이 노트에서는 아직 검증되지 않음**

> **개정 1.0 시점에는 SPIMCI 모듈이 연결된 적이 없습니다.** 프레이밍, 흐름 제어 도출, 극성 모순은 OIF-CMIS-05.3 부록 B.3에서 가져왔으며 05.4와 교차 확인했습니다. 검증된 것은 호스트 측과 고장 모드입니다. 위조된 응답은 2026년 8월 15일 모듈이 연결되지 않은 버스에서 관찰되었고, 2026년 9월 2일에는 존재하지만 응답하지 않는 보드에서 다시 관찰되었으며, 이때는 체크섬 가드까지 무력화했습니다. 모듈의 응답에 의존하는 모든 것은 미완료 상태입니다.

1.3 관련 문서

표 2. 관련 문서

문서	주제
OIF-CMIS-05.3	부록 B.3, SPI-Based Management Communication Interface, 374~383페이지
OIF-CMIS-05.4	부록 B.3, 416~425페이지. 바닥글을 제외하면 본문은 5.3과 동일합니다
AN0008	Binho Supernova를 사용한 I2C 기반 CMIS 광 모듈 브링업
AN0011	Binho Supernova를 사용한 CMIS 펌웨어 업데이트 시간 측정

2 필요한 장비 및 소프트웨어

2.1 하드웨어

표 3. 필요한 장비

항목
Binho Pulsar 또는 Binho Supernova USB 호스트 어댑터
SPIMCI 관리 인터페이스를 갖춘 CMIS 모듈
USB-C 케이블
Saleae Logic 캡처 하드웨어, 선택 사항

2.2 연결

CMIS는 기존 SPI 신호의 이름을 바꿉니다. 아무도 혼동하지 않도록 사양의 이름을 사용하고 익숙한 이름을 함께 표기하십시오.

표 4. 어댑터와 모듈 간 연결

어댑터	사양 이름	기존 이름	필수 여부
CS	`CSn`	CS, SS 또는 ModSeln	예
SCK	`CLK`	SCLK	예
MOSI	`IOTI`, Initiator Out Target In	MOSI, PICO	예
MISO	`IITO`, Initiator In Target Out	MISO, POCI	예
3V3	전원		예, 보드가 자체 전원을 공급하는 경우 제외
GND	리턴		예

> **주의:** `CSn` 은 선택 사항이 아니라 필수입니다. 액티브 로우는 규범적 요구 사항이며(B.3.3), 각 타깃에는 전용 `CSn` 이 하나씩 있고 한 번에 최대 하나만 어서트됩니다(B.3.2). 이것이 없으면 프레이밍 기준도, 오류 감지도, 재동기화도 없습니다. 클록 에지가 하나만 더 들어가도 세션의 나머지 부분이 영구적으로 어긋납니다. 일반 SPI 디코더는 칩 셀렉트가 없어도 이를 허용하며 그럴듯한 쓰레기 값을 반환합니다.

2.3 소프트웨어

```
pip install binhopulsar          # Pulsar
pip install binhosupernova       # Supernova
python cmis_mci.py selfcheck
```

`--transport spi` 로 SPI 바인딩을 선택합니다. 유틸리티는 AN0008, AN0009, AN0011과 함께 제공되는 것과 같은 파일입니다.

3 SPIMCI란 무엇이며 왜 이해할 가치가 있는가

SPIMCI는 코패키지드 옵틱스를 계기로 CMIS 5.3에서 도입되었으며, 개정 이력에는 "SPI-based SPIMCI in support of CPO"라고 적혀 있습니다. CMIS 5.2에는 `SPI` 라는 문자열이 어디에도 나타나지 않습니다.

이는 향후 모든 바인딩에 중요한 선례이기도 합니다. SPIMCI는 레지스터 액세스 계층이나 Common Data Block을 전혀 건드리지 않고 추가되었으며, 그와 동시에 주소 지정 방식도 바로잡았습니다. I2CMCI는 페이지 내 바이트 주소만 전달하므로 00h:126과 00h:127의 페이지 선택 레지스터가 필요하지만, SPIMCI 트랜잭션은 뱅크, 페이지, 바이트로 이루어진 완전한 3차원 주소를 전달하고, 암시적 페이지 전환을 수행하며, 최대 2048바이트를 전송합니다.

호스트에게 실질적인 결과는 **써야 할 페이지 선택 레지스터도 없고, 잘못될 수 있는 트랜잭션 간 주소 상태도 없다**는 것입니다.

표 5. 사양으로 고정된 파라미터

파라미터	값	출처
워드 크기	8비트, MSB 우선	B.3.7.2.3
샘플링 에지	CLK 상승 에지	B.3.7.1
칩 셀렉트 극성	액티브 로우	B.3.3
최대 전송량	2048바이트	B.3.7.1

> **참고:** 사양은 샘플링 에지를 고정하지만 CLK 유휴 레벨은 어디에도 명시하지 않으므로, "SPIMCI는 SPI 모드 0이다"라는 것은 인용이 아니라 추론입니다. 유틸리티는 기본적으로 모드 0을 사용하며 모드를 선택할 수 있게 합니다.

4 트랜잭션 구조

모든 SPIMCI 트랜잭션은 세 단계로 이루어진 **4 + N + M**바이트입니다.

표 6. 트랜잭션 제어 단계, 4바이트, 이니시에이터에서 타깃 방향만

비트	필드	참고
0	Command Type, R/Wn	섹션 5.2 참조
1:4	Bank Index	4비트. 사양은 이것이 "15보다 큰 임의의 뱅크 인덱스 값을 허용하지 않는다"고 명시합니다
5:15	EncodedSize	11비트. 전송 바이트 수 = EncodedSize + 1, 최대 2048
16:23	Page Index	8비트
24:31	Byte Index	8비트, 시작 주소. 이후 순차 증가

제어 단계 전체에 걸쳐 `IITO` 는 정의되지 않습니다.

흐름 제어 단계, N바이트, N은 2 이상. 이니시에이터는 0 값의 더미 바이트를 보내고, 타깃은 이를 무시합니다. 타깃은 이 단계의 **마지막** 바이트에서 `IITO` 로 **ACK = 00h** 또는 **NACK = FFh**를 반환합니다.

데이터 전송 단계, M = EncodedSize + 1바이트. 읽기에서는 데이터가 `IITO` 에 있고 `IOTI` 로는 00h가 전송됩니다. 쓰기에서는 데이터가 `IOTI` 에 있고 `IITO` 는 정의되지 않습니다.

5 두 미지수와 이를 해결하는 방법

5.1 N은 와이어에 없으며 부트스트랩 공백이 있다

`N` 은 `00h:213` 비트 6-0의 `MciFlowControlDuration` 과 `00h:213` 비트 7의 `MciFlowControlDurationEncoding` 에서 도출됩니다.

표 7. 흐름 제어 바이트 수

인코딩 비트	도출 방법
0, 정적	N = 2 + 광고된 값
1, 속도 의존	N = max(2, ceil(duration x SpiSpeedInMHz / 8)), 속도는 `00h:27.3-0` 의 `MciSpeedConfiguration` 에서 가져옴

B.3.7.1.2는 양쪽이 `N` 을 동일하게 도출할 것을 요구합니다. 그러나 `00h:213` 은 `N` 을 알기 전에는 읽을 수 없으며, 사양은 안전한 부트스트랩 값을 명시하지 않습니다. 수동 버스 관찰자도 이를 알 수 없습니다.

해결책은 `N` 을 스윕하여 모듈 자체의 체크섬이 어떤 값이 옳은지 판단하게 하는 것이며, 이것이 `spi-probe` 가 하는 일입니다.

5.2 R/Wn 극성은 자기모순이다

표 8. CMIS 5.3의 R/Wn 극성

출처	내용
B.3.7.1.1	"이 비트가 1이면 읽기가 요청된 것이다"
B.3.8.1~B.3.8.5, 그림 B-14~B-17	읽기는 0

둘 다 같은 문서에 있습니다. 알아 둘 만한 판단 근거가 있습니다. B.3.8.5는 **TEST**를 바이트 주소 0에서의 1바이트 읽기로, **32개 제어 비트가 모두 0**인 것으로 정의하는데, 이는 0이 읽기를 의미할 때만 자기 일관성이 있습니다. 유틸리티는 기본적으로 그림을 따르고, 요청 시 어느 극성이든 사용하며, 지정되지 않은 경우 두 극성을 모두 탐색합니다.

6 절차

6.1 오프라인에서 도구 확인

```
python cmis_mci.py selfcheck
```

6.2 N과 극성 확정

```
python cmis_mci.py spi-probe --transport spi
```

이 명령은 각 극성에 대해 `N` 을 2부터 16까지 스윕하며, Page 00h 체크섬이 검증될 때**만** 조합을 받아들입니다. 즉, 94바이트의 모듈 고유 내용이 모듈 자신이 기록한 바이트와 일치해야 합니다. 완료된 트랜잭션, 응답, 심지어 그럴듯해 보이는 데이터까지도 모두 전원이 꺼진 보드에서 만들어질 수 있습니다.

> **주의:** 일치하는 체크섬 역시 전원이 꺼진 보드에서 만들어질 수 있습니다. 그 방법은 정확히 하나이며, 실제로 일어나는 것이 바로 그 방법입니다. 무응답 SPI 버스는 모든 바이트에 대해 `00h` 를 반환합니다. 94바이트의 `00h` 를 합하면 `00h` 이고, 같은 버스가 저장된 체크섬으로 보고하는 값도 `00h` 이므로 둘이 일치합니다. 따라서 스윕은 체크섬이 검증되고 **동시에** Page 00h에 서로 다른 값이 둘 이상 있을 때만 조합을 받아들입니다. 섹션 8.2에는 이 조건을 추가하기 전에 명령이 보고한 내용이 기록되어 있습니다.

값을 이미 알고 있다면 이를 전달하고 스윕을 건너뛰십시오.

```
python cmis_mci.py spi-probe --transport spi --n 4 --read-bit 0
```

실패하면 명령은 확인할 사항을 보고하며, "모듈이 응답했다"는 것을 위안으로 제시하지 않습니다. SPIMCI에서는 그것이 아무런 정보가 되지 않기 때문입니다.

6.3 모듈 읽기

```
python cmis_mci.py bringup  --transport spi --n 4 --read-bit 0
python cmis_mci.py identify --transport spi --n 4 --read-bit 0
python cmis_mci.py read --transport spi --page 01h --byte 251 --n 4 --read-bit 0
```

마지막 명령의 페이지 읽기에는 페이지 선택 쓰기가 필요하지 않다는 점에 유의하십시오. 페이지는 트랜잭션 헤더에 담겨 전달됩니다.

7 분석기가 더해 주는 것

CMIS SPIMCI 분석기로 디코딩한 `CSn` , `CLK` , `IOTI` , `IITO` 캡처는 4 + N + M 프레이밍, 응답 바이트, 그리고 트랜잭션이 주소 지정한 레지스터를 보여 줍니다.

그 외에도 사양이 실제로 정의하는 유일한 오류 조건을 보고하는데, 일반 디코더는 구조적으로 이를 표현할 수 없습니다.

ABORT. 정상적인 STOP은 예상 시점에 8비트 경계에 놓입니다. 8비트 경계가 아니거나 예상보다 이른 불규칙한 STOP은 트랜잭션을 종료시키며, "불완전하게 수신된 바이트는 수신 측에서 폐기하거나 무시해야 한다"고 규정되어 있습니다. 표준 SPI 디코더는 칩 셀렉트 디어서트로 중단된 워드를 조용히 폐기합니다. 프레임도, 플래그도, 마커도 없습니다. 바로 그 경우가 SPIMCI ABORT 조건입니다.

분석기는 또한 `N` 을 숨기지 않고 드러냅니다. `N` 은 명시적으로 설정하거나, 지속 시간과 속도로 지정하거나, `IITO` 에서 바이트 4부터 앞으로 스캔하여 첫 번째 00h 또는 FFh를 찾아 자동 감지할 수 있으며, 모두 0인 데

이터 바이트가 혼동을 일으킬 수 있으므로 신뢰도 표시가 함께 제공됩니다.

> **CMIS SPIMCI 분석기는 개발 중이며 아직 출시되지 않았습니다.** 이 섹션은 분석기가 생성하는 디코딩을 설명하는 것이며, 독자가 지금 구매할 수 있는 제품을 설명하는 것이 아닙니다.

8 측정 결과

8.1 테스트 구성

표 9. 테스트 구성

항목	값
어댑터	Binho Supernova, 시리얼 `4F970418…1111`, 펌웨어 4.4.0, 하드웨어 개정 C
SDK	`binhosupernova`, Python 3.13
모듈	없음. 모듈 자리에 SPIMCI 펌웨어가 없는 NXP FRDM-MCXN947을 연결
버스	3.3 V, 1 MHz, 모드 0, Binho 어댑터 보드의 칩 셀렉트 S0

8.2 결과

관찰, 2026년 8월 15일. 모듈이 연결되지 않은 버스에 발행한 SPIMCI 읽기가 완료되었고, 유효한 응답을 반환했으며, 데이터로 `00 00`을 반환했습니다. 이것이 섹션 1의 근거가 되는 관찰이며, 특정 모듈이 아니라 인코딩의 특성입니다.

관찰, 2026년 9월 2일. 위의 경우보다 더 나쁩니다. 같은 프로브를 존재하고, 전원이 공급되고, 올바르게 배선되었지만 SPIMCI 펌웨어를 실행하지 않는 보드를 대상으로 실행했으므로, 반환된 모든 바이트는 `00h`였습니다. 명령은 모듈이 없다고 보고하지 않았습니다. 정상 모듈이라고 보고했습니다.

```
N        : 2 flow control bytes, transaction is 4 + 2 + M
R/Wn     : 0 means READ on this module
00h:222  : checksum 0x00 matches the stored byte

PASS     : the module is present and self consistent.
```

체크섬이 일치한 이유는 비교의 양쪽이 모두 0이었기 때문입니다. 94바이트의 `00h`를 합하면 `00h`이고, 같은 무응답 버스가 저장된 체크섬으로 보고한 값도 `00h`였습니다. 섹션 6.2가 무응답 버스를 거부하기 위해 의존하던 검사 자체가 그 무응답에 의해 무력화되었으므로, 유틸리티는 체크섬 일치를 인정하기 전에 Page 00h에 서로 다른 값이 둘 이상 있을 것을 요구하도록 수정되었습니다. 같은 하드웨어에 대해 이제는 다음과 같이 보고합니다.

```
FAIL     : no combination of N and R/Wn produced a valid Page 00h checksum.
           Either the module is absent, or CSn, CLK, IOTI or IITO are
           miswired. On SPIMCI a silent bus still returns a perfect ACK.
```

이는 이 노트의 주장이 이 노트 자신에게 적용된 것입니다. 성공한 것처럼 보이는 브링업이 실제로는 아무것도 읽지 않고 있을 수 있습니다. 부록 B.3을 구현하는 호스트도 같은 경우에 대비해야 합니다.

오프라인. 이 노트가 구동하는 레지스터 액세스 계층은 AN0008이 자체 검사하는 것과 같으며, 시뮬레이션 모듈을 대상으로 8개 중 8개를 통과합니다. 여덟 번째 경우가 이 섹션에서 추가한 것입니다. 모두 0인 Page 00h는 일치하는 체크섬으로 읽히는 대신 거부되어야 합니다.

하드웨어에서, 모듈을 대상으로. 아직 없습니다. 개정 0.2에서 미완료인 사항은 SPIMCI 모듈을 대상으로 한 `N` 스윕, 극성 판정, 식별 정보 읽기, Page 00h 체크섬입니다. 섹션 8.2가 확립한 것은 고장 모드와 그에 대한 가드이며, 성공 경로가 아닙니다.

9 문제 해결

표 10. 문제 해결

증상	가능한 원인 및 조치
모든 트랜잭션이 성공하고 모든 데이터가 0임	전형적인 위조 응답입니다. 모듈이 없거나, 전원이 꺼졌거나, 설정되지 않았습니다. 94개의 0은 0인 저장 체크섬과 일치하므로, Page 00h에 서로 다른 값이 둘 이상 있을 것도 요구해야만 체크섬으로 이를 구분할 수 있습니다
`spi-probe` 가 유효한 조합을 찾지 못함	`N` 을 의심하기 전에 `CSn` , `CLK` , `IOTI` , `IITO` 배선을 확인하십시오. 무응답 버스도 응답을 반환합니다
흐름 제어 바이트가 00h도 FFh도 아님	`N` 이 잘못되어 읽고 있는 바이트가 응답이 아니라 데이터 또는 제어 바이트입니다
데이터가 하나 이상의 바이트만큼 밀려 있음	`N` 이 그만큼 어긋나 있습니다. 수동으로 조정하지 말고 스윕을 다시 실행하십시오
읽기는 데이터를 반환하지만 쓰기가 무시되는 것처럼 보임	R/Wn 극성이 반대입니다. 다른 `--read-bit` 을 시도하십시오
트랜잭션이 워드 중간에서 잘림	불규칙한 STOP, 즉 ABORT 조건입니다. 다른 무언가가 `CSn` 을 구동하고 있지 않은지 확인하십시오
어댑터가 짧은 전송을 보고함	전송 길이와 페이로드 길이가 일치하지 않습니다. 이는 버스 결함이 아니라 호스트 버그입니다

10 약어

표 11. 약어

용어	의미
ACK	Acknowledgement(응답), SPIMCI에서는 00h
CPO	Co-packaged optics(코패키지드 옵틱스)
CMIS	Common Management Interface Specification
CSn	Chip select(칩 셀렉트), 액티브 로우
IITO	Initiator In, Target Out
IOTI	Initiator Out, Target In
MCI	Management Communication Interface
NACK	Negative acknowledgement(부정 응답), SPIMCI에서는 FFh
OIF	Optical Internetworking Forum
RAL	Register Access Layer(레지스터 액세스 계층)
SPIMCI	SPI Management Communication Interface

11 문서의 소스 코드에 관한 참고 사항

이 문서에 표시되고 함께 제공되는 아카이브에 포함된 예제 코드는 평가 및 참고용으로 제공됩니다. MIT 라이선스로 배포되며, 라이선스 전문은 아카이브에 포함되어 있습니다.

표 12. 에셋 아카이브의 내용

파일	설명
`AN0010.md`	이 문서의 Markdown 소스
`assets/cmis_mci.py`	참조 유틸리티, AN0008부터 AN0011까지 공유
`assets/LICENSE`	예제 코드에 적용되는 MIT 라이선스

12 개정 이력

표 13. 개정 이력

개정	날짜	설명
0.1	2026년 9월 1일	내부 검토용 초안. 사양 내용은 검증됨. 절차는 아직 하드웨어에서 실행되지 않음.
0.2	2026년 9월 2일	AN0009에서 번호 변경. SPIMCI 펌웨어가 없는 보드를 대상으로 `spi-probe` 를 실행함: 모두 0인 위조 응답이 Page 00h 체크섬을 무력화하여 명령이 PASS를 보고함. 이제 유틸리티는 하나의 값이 반복되는 페이지를 거부하며, 섹션 8.2에 두 출력을 모두 기록함. SPIMCI 모듈은 아직 연결되지 않음.
1.0	2026년 11월 6일	공개.

법적 고지(영문)

Definitions

Draft

A document marked draft is under internal review and subject to formal approval. Binho Inc. makes no representation or warranty as to its accuracy or completeness and accepts no liability arising from its use.

Disclaimers

Limited warranty and liability

Information in this document is believed to be accurate and reliable. Binho Inc. makes no representation or warranty, express or implied, as to its accuracy or completeness, accepts no liability for the consequences of its use, and takes no responsibility for content originating outside Binho Inc.

In no event shall Binho Inc. be liable for any indirect, incidental, punitive, special or consequential damages, including without limitation lost profits, lost savings, business interruption, or costs of removing or replacing any product, whether or not based on tort, warranty, breach of contract or any other legal theory. Notwithstanding any damages a customer might incur, the aggregate and cumulative liability of Binho Inc. is limited in accordance with its terms and conditions of commercial sale.

Right to make changes

Binho Inc. reserves the right to change the information in this document, including specifications and product descriptions, at any time and without notice. This document supersedes all information supplied prior to its publication.

Suitability for use

Binho Inc. products are not designed, authorized or warranted for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure of a Binho Inc. product can reasonably be expected to result in personal injury, death, or severe property or environmental damage. Any such inclusion or use is at the customer's own risk and Binho Inc. accepts no liability for it.

Applications

Applications described here are illustrative only. Binho Inc. makes no warranty that they are suitable for the specified use without further testing or modification. Customers are responsible for the design and operation of their own applications and products, for appropriate design and operating safeguards, and for determining whether a Binho Inc. product is suitable and fit for their purpose. Binho Inc. accepts no liability for assistance with applications or customer product design.

Third-party products and specifications

This document refers to products, boards, specifications and documentation supplied by third parties, which remain the responsibility of their respective owners. Binho Inc. makes no warranty regarding them and their behavior may change without notice. Register maps, protocol details and device identifiers reproduced here were observed on the specific hardware and firmware versions stated in this document and must be confirmed against the vendor's own documentation for the reader's part.

Example code

Source code supplied with this document is provided as an example, for evaluation and reference. It is not qualified for production use, has not been developed under a functional safety or security process, and is licensed under the terms accompanying it in the distribution archive.

Terms and conditions of commercial sale

Binho Inc. products are sold subject to the general terms and conditions of commercial sale published by Binho Inc., unless otherwise agreed in a valid written individual agreement, in which case only the terms of that agreement apply.

Export control

This document and the items it describes may be subject to export control regulations. Export may require prior authorization from the competent authorities.

Security

All products may be subject to unidentified vulnerabilities or may support security standards with known limitations. The customer is responsible for the design and operation of its applications and products throughout their lifecycle to reduce the effect of such vulnerabilities, and for compliance with all legal, regulatory and security requirements concerning its products.

Trademarks

All referenced brands, product names, service names and trademarks are the property of their respective owners. **Binho** and the Binho logo are trademarks of Binho Inc. **CMIS** 및 **OIF**는 Optical Internetworking Forum의 상표 또는 등록 상표입니다. **Saleae** 및 **Logic**은 Saleae, Inc.의 상표입니다.