binho

應用筆記

在 Microchip dsPIC33AK 上建構 CMIS I3C 目標裝置並以 Binho Supernova 進行測試

AN0019

文件資訊

關鍵詞	AN0019, Supernova, CMIS, OIF, I3C, MIPI I3C Basic, SDR, I3C 目標裝置, dsPIC33AK, dsPIC33AK256MPS306, MCC Melody, DMA, ENTDAA, IBI, GETSTATUS, 光模組
摘要	CMIS 光模組以 I3C 目標裝置的身分回應其主機，而主機讀取它的方式與 I2C 主機相同：寫入位元組位址、重複 START、讀取。 在 dsPIC33AK256MPS306 的 I3C 周邊上，該寫入的中斷在讀取標頭已被回應之後才到達，因此由該中斷備妥回應的目標裝置 會對每一次此類讀取回以 NACK。本應用筆記說明位元組位址在何處能及時取得、韌體如何改由接收 DMA 進行備妥， 以及在 Binho Supernova 上量測到的數據：4 到 128 位元組的讀取 60 次中 60 次成功，開啟帶內中斷時 300 次中 300 次成功， 重設後亦然。本筆記提供主機端測試指令碼，並列出移植時必須預先規劃的周邊與驅動程式行為。

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

修訂版 1.0
6 NOVEMBER 2026

> **測試範圍**。第 8 節中的每項結果均在第 3 節所述的測試台上量測。目標裝置韌體不在資產壓縮檔中，如需取得請聯絡 techsupport@binho.io。

1 簡介

CMIS 模組提供一個 256 位元組的視窗，由低端記憶區（Lower Memory）加上高端記憶區（Upper Memory）的一個 128 位元組頁組成，頁由 `00h:127` 處的頁選擇位元組選定。透過 I2C，主機先寫入位元組位址，然後以重複 START 進行讀取。 正在為 CMIS 定義的 I3C 管理介面保留了這種存取方式，因此以 I3C 目標裝置周邊建構的模組必須回應這樣一個讀取： 其位元組位址在同一次傳輸中僅早幾個位元時間到達。

如果周邊在每個位元組到達時就將其交給韌體，這很容易做到。而 dsPIC33AK256MPS306 的 I3C 周邊則是以命令與回應為基礎： 接收到的傳輸只以回應的形式回報一次，讀取由一個傳送命令回應，而該命令必須在讀取標頭到達時就已經有效。 本應用筆記探討的就是如何讓這兩者銜接起來。

1.1 範圍

- dsPIC33AK 上的 CMIS 暫存器模型：分頁、第 8.2.15 節的不支援頁覆寫、附錄 B.1.2 規定的 128 位元組區段內環繞， 以及攜帶 Flags Summary 位元組 `00h:4` 的 IBI。
- 在 2.5 MHz 推挽模式下回應寫入 + Sr + 讀取（組合讀取）以及寫入、STOP、讀取（分離讀取）。
- 動態位址分配（DAA）、RSTDAA 和 Direct GET CCC。
- 一個從 Supernova 執行本筆記中所有檢查的主機指令碼。

不涵蓋：CDB、韌體下載、HDR 模式，以及 CMIS 控制暫存器在模組端的意義。除頁選擇與主機寫入的位元組外， 暫存器映像是靜態的。

1.2 適用裝置

表 1. 適用裝置

系列	範例元件	識別碼	依據
具 I3C 的 dsPIC33AK MPS	dsPIC33AK256MPS306（EV02E57A DIM）	PID `02 36 43 4D 49 53`	已測試
不具 I3C 的 dsPIC33AK MP	dsPIC33AK512MPS506 及其系列	-	不適用：無 I3C 周邊
具有相同 I3C 周邊的其他元件	-	-	未測試

1.3 相關文件

表 2. 相關文件

文件	標題
OIF-CMIS-05.3	Common Management Interface Specification，修訂版 5.3
MIPI I3C Basic v1.1.1	MIPI I3C Basic Specification，版本 1.1.1
DS70005539	dsPIC33AK MPS 系列資料手冊，第 24 節（I3C）
AN0009	使用 Binho Supernova 透過 I3C 管理 CMIS 光模組

2 系統概觀

Supernova 是 I3C 控制器，也是匯流排上唯一的控制器。dsPIC33AK 是目標裝置，靜態位址為 `0x50`，即 CMIS 模組位址。控制器透過 ENTDAA 為其分配動態位址（測試台分配 `0x08`），然後以 SDR 對其進行讀寫。

決定韌體設計的唯一事實是：組合讀取的位元組位址何時變得可見。在關閉 I3C 中斷的情況下，以 100 MHz 計時器為基準 輪詢周邊，並與邏輯分析儀對齊，量測結果如下：

表 3. 2.5 MHz 推挽下的組合讀取，自 START 起算的時間

事件	時間
位元組位址的 T 位元	22.3 µs
位元組位址進入接收 FIFO	23.8 µs
重複 START	24.1 µs
讀取標頭的 ACK 時槽	~27.4 µs
寫入的回應中斷	ACK 時槽之後

由寫入的回應中斷備妥（arm）的傳送命令總是太遲，目標裝置會對讀取回以 NACK。位元組位址本身在 ACK 時槽之前 3.6 µs 即可取得，而備妥一個命令需要 1.15 到 1.26 µs。因此韌體由接收 DMA 進行備妥：將接收到的字組搬移到 RAM 的通道每個字組產生一次中斷，其優先順序高於 I3C 中斷，該中斷依據第一個位元組備妥傳送命令。

其餘一切都源自該周邊及其 MCC Melody 驅動程式的三項特性：

- 一個傳送命令回應一次讀取。韌體備妥從位元組位址到區段結尾的 128 位元組資料串；控制器會提前結束讀取。
- 傳送緩衝區可容納 8 個字組（32 位元組）。資料串的其餘部分由主迴圈供給。任何使主迴圈離開的時間超過傳送 32 位元組 所需時間（約 115 µs）的動作，都會使讀取在 32 位元組處被截斷。
- 重新備妥接收會重設接收 FIFO，同時也會捨棄已備妥的傳送命令。接收僅在讀取之後才重新備妥。

3 硬體設定

3.1 所需設備

- Binho Supernova，韌體 4.6.2，I3C HV 連接埠。
- Microchip EV74H48A Curiosity 平台，搭配 EV02E57A dsPIC33AK256MPS306 DIM。其 PKoB4 用於燒錄元件並承載 UART。
- 選用：連接 SCL 和 SDA 的邏輯分析儀。本筆記中的擷取使用 Saleae Logic 以 12 MS/s 取得。

3.2 連接

表 4. Supernova 與目標裝置之間的連接

Supernova I3C 連接埠	dsPIC33AK DIM	是否必要
SCL	I3CSCL1（RC5）	是
SDA	I3CSDA1（RC4）	是
GND	GND	是
I3C 電源軌，3.3 V	不連接，DIM 由 Curiosity 板供電	-
-	UART1 TX（RD0），經 PKoB4 虛擬 COM 連接埠輸出	選用

3.3 匯流排注意事項

匯流排工作於 3.3 V。推挽 2.5 MHz 是 Supernova 提供的最低速率，也是本文唯一量測過的速率。較快的速率會縮短表 3 中的 3.6 µs 視窗：在 12.5 MHz 時，該視窗短於備妥一個命令所需的時間。

4 軟體設定

目標裝置韌體：

- MPLAB X v6.35 及 MPLAB IPE、XC-DSC v4.00、dsPIC33AK-MP DFP 1.4.260。
- MCC Melody I3C 目標裝置驅動程式，含第 5.6 節所述的本地修改。

主機：

- Python 3 及 pycosmicsdk（最新版本），以及本筆記資源中的 `cmis_target_check.py`。

```
pip install --upgrade pycosmicsdk
```

5 目標裝置韌體設計

5.1 暫存器模型

低端記憶區與高端記憶區頁 `00h` 取自 CMIS 5.3 附錄 C 的範例映像。`00h:127` 處的頁選擇接受 `00h` 和 `01h`； 任何其他值都會被替換為 `00h`（第 8.2.15 節）。每個區段首尾相接儲存兩次，另加偏移一到三個位元組的副本， 因此從任何位元組位址開始的 128 位元組資料串都是一個對齊的緩衝區，區段內的 B.1.2 環繞不需要任何程式碼。

5.2 識別資訊

表 5. 控制器看到的識別資訊

項目	值
靜態位址	`0x50`
PID	`02 36 43 4D 49 53`（MIPI 製造商 `0x11B`，元件 `"CMIS"`）
BCR	`0x17`：支援 IBI，IBI 攜帶一個位元組，有最大資料速率限制
DCR	`0x00`

5.3 備妥讀取

1. 接收 DMA 通道將每個接收到的字組搬移到 `rxbuf`，其完成中斷已啟用，優先順序為 5，高於優先順序為 4 的 I3C 中斷。
2. 在寫入的第一個字組到達時，DMA 中斷將 `rxbuf[0]` 作為位元組位址，並備妥從該位址開始之資料串的傳送命令。
3. 當寫入的回應到達且其中只攜帶了位址時，命令保持備妥狀態，接收在讀取完成後重新備妥。
4. 當寫入攜帶了資料時，透過重設延伸傳送緩衝區來取消該命令，並立即重新備妥接收。
5. 如果讀取標頭到達時仍未找到已備妥的命令，周邊會發出讀取請求。韌體捨棄任何遲到備妥的命令，並透過 `I3C1_Target_ReceiveCancel()` 重新備妥接收，因為在這種情況下寫入的回應永遠不會到來。

5.4 供給長讀取

傳送緩衝區預先填入 8 個字組。`I3C1_Target_TxRefill()` 從主迴圈供給其餘部分，在其執行的少數指令期間遮罩 I3C 與 DMA 中斷。開發調試韌體的 UART 狀態列預設為關閉：其在 9600 鮑率下的阻塞式 `printf` 每秒佔用主迴圈約 110 ms， 使 128 位元組的讀取在 32 位元組處被截斷。

5.5 事件

在 IBI 啟用期間，IBI 每秒發出一次，以 `00h:4` 作為其必要的資料位元組。周邊重設後 IBI 處於啟用狀態，這符合 I3C Basic 的規定，而且周邊不會保留寫入該啟用位元的 0：DISEC 才是關閉 IBI 的方法。

5.6 對 Melody 驅動程式的修改

Table: `mcc/` 中的本地修改，以 `binho:` 標記

修改	原因
`BufferTransmit` 預先填入 8 個字組；`I3C1_Target_TxRefill()` 供給其餘部分	超過 8 個字組時，DMA 路徑將第 7 個字組寫了兩次且不再繼續，因此命令遭到拒絕
`I3C1_Target_ReceiveCancel()`	在寫入 + Sr + 被 NACK 的讀取標頭之後，寫入不會有回應；驅動程式一直將其接收標記為進行中，並拒絕之後的每一次接收
在完成回呼之前清除 `rxInProgress`	使回呼能夠重新備妥接收
靜態位址 `0x50`	CMIS 模組位址

6 操作程序

步驟 1：在韌體資料夾中建置目標裝置韌體並燒錄：

```
make flash
```

建置以 MPLAB IPE 輸出的 `Program Succeeded.` 結束。

步驟 2：依表 4 將 Supernova I3C 連接埠連接到 DIM。

步驟 3：以寫入 + Sr + 讀取方式讀取 128 位元組，共 60 次：

```
python cmis_target_check.py reads --length 128 --count 60
```

```
init bus: FW_I3C_BUS_INIT_NACK_RSTDAA
target DA 0x08, PID 02 36 43 4d 49 53
............................................................
reads 60/60  (combined, 128 B, IBI off)
```

`init bus` 這一行表示重設後的第一次 RSTDAA 被 NACK（第 8.3 節）；指令碼會再次初始化。

步驟 4：在啟用 IBI 的情況下執行相同動作（讀取之前送出 ENEC）：

```
python cmis_target_check.py reads --length 128 --count 300 --ibi
```

```
reads 299/300  (combined, 128 B, IBI on)
```

點列中的 `F` 表示發生錯誤的讀取，`B` 表示傳回錯誤資料的讀取。

步驟 5：資料寫入與讀取混合：每一輪寫入 `7Fh=00h` 和 `7Eh=00h`，然後讀取 `7Fh`（1 位元組）和 `80h`（16 位元組），並與映像進行比對：

```
python cmis_target_check.py mixed --count 60
```

```
rounds 60/60
```

步驟 6：位址重設與重新分配：

```
python cmis_target_check.py daa --cycles 10
```

```
0: RSTDAA SUCCESS, GETSTATUS at old DA FW_I3C_NACK_ADDRESS, read after ENTDAA ok
...
cycles 10/10
```

步驟 7：識別資訊與 Direct GET CCC：

```
python cmis_target_check.py ccc --getstatus 100
```

```
GETBCR   SUCCESS                17
GETDCR   SUCCESS                00
GETPID   SUCCESS                02 36 43 4d 49 53
GETMXDS  SUCCESS                00 00 00 00 01
GETMRL   FW_I3C_NACK_ADDRESS
GETMWL   SUCCESS                ff
GETSTATUS 100/100  {'SUCCESS': 100}
```

Direct GET 可能因 `7Eh` 標頭被 NACK 而失敗（第 8.3 節）；此處 GETMRL 即失敗了。請再執行一次。

7 參考

Table: `cmis_target_check.py` 命令

命令	功能
`reads --length N --count N [--split] [--ibi]`	依序在 `00h`、`10h`、`7Ch`、`80h`、`A0h`、`FCh` 處讀取並與映像比對。除非指定 `--split`，否則為組合讀取；`--ibi` 先送出 ENEC，否則送出 DISEC
`mixed --count N`	向 `7Fh` 和 `7Eh` 寫入資料，然後對 `7Fh` 和 `80h` 進行組合讀取並驗證

命令	功能
`ccc [--getstatus N]`	GETBCR、GETDCR、GETPID、GETMXDS、GETMRL、GETMWL，然後執行 GETSTATUS N 次
`daa --cycles N`	RSTDAA，檢查舊位址是否 NACK，ENTDAA，檢查一次讀取
`i2c`	在任何位址分配之前，以 I2C 模式在 `0x50` 處讀取 `00h:0-3` 和 `00h:46-48`

通用選項：`--serial`、`--push-pull`（預設 `PUSH_PULL_2_5_MHZ_DC_25`）、`--voltage-mv`（預設 3300）、`--image`（預設為指令碼旁的 `cmis_image.h`）。每個命令都以 RSTDAA 和 ENTDAA 開始，並依 PID 中的 `"CMIS"` 找到目標裝置。

8 量測結果

8.1 測試組態

表 6. 測試組態

項目	版本
Supernova	序號 4F970418，韌體 4.6.2
目標裝置	EV74H48A 上的 EV02E57A DIM
工具鏈	XC-DSC v4.00、DFP 1.4.260、MPLAB X v6.35
主機	Ubuntu 26.04；工作階段 1 和 2：Python 3.13、binhosupernova 4.2.0；工作階段 3：Python 3.14.4、pycosmicsdk 1.9.1
匯流排	3.3 V，推挽 2.5 MHz，開汲極 1 MHz

8.2 結果

共三個工作階段。工作階段 2 在燒錄（一次重設）之後開始；工作階段 3（2026 年 11 月 6 日）在每項測試前都燒錄目標裝置， 主機使用 pycosmicsdk：

表 7. 結果

測試	工作階段 1	工作階段 2	工作階段 3
組合讀取，4 / 16 / 32 / 128 B，IBI 關閉	各 60/60	60/60（4 和 128 B）	各 60/60
組合讀取，128 B，IBI 開啟	300/300	300/300	299/300
分離讀取，4 / 128 B	各 60/60	60/60（128 B）	60/60（128 B）
資料寫入與組合讀取，資料經驗證	60/60 輪	60/60 輪	60/60 輪
RSTDAA，舊位址 NACK，ENTDAA，讀取	10/10	10/10	10/10
GETSTATUS	97/100	94/100	100/100

在採用接收 DMA 備妥之前，同樣的組合讀取為 60 次中 0 次成功。

8.3 主機應預期的行為

- **目標裝置重設後的第一次傳輸會被 NACK**。在 5 次重設的每一次之後，第一個 `7Eh` 標頭都被 NACK，第二個被 ACK。 DS70005539 第 24.4.5 節描述該模組只有在偵測到 SCL 時脈後才離開重設。只探測一次並將 NACK 視為 「無 I3C 目標裝置」的主機將會漏掉此模組；請重試探測。
- **部分 Direct GET CCC 會因 `7Eh` 被 NACK 而失敗**。在一個工作階段中，Direct GET 之後的 `7Eh` 標頭約每 7 個中有 1 個被 NACK（1200 次 GET 中有 173 次失敗，全部 173 次在線路上都可見為 `7E/W NACK`）；在另一個工作階段中， 500 次中 500 次通過。私有讀取不受影響。請重試因 NACK 而失敗的 GET。
- **GETMRL 回報 255，且 SETMRL 不會生效**。韌體設定為 128；周邊自行以 255 回應 GETMRL。執行 SETMRL 16 之後， 32 位元組的讀取仍傳回 32 位元組。

8.4 位址分配之前的 I2C 讀取

在 ENTDAA 之前，目標裝置在 `0x50` 處回應 I2C。從 Supernova 的 I2C 連接埠（一個一般 I2C 控制器，3.3 V 下使用 2.2 kOhm 上拉電阻）進行讀取，每次執行前都燒錄目標裝置，在 `00h`、`10h`、`7Ch`、`80h`、`A0h` 和 `FCh` 處讀取 3、4、16 和 128 位元組，在 100 kHz 與 400 kHz 下均為各 30/30，資料與映像比對一致。從 I2C 模式下的 I3C 連接埠（每次傳輸都以 `7Eh` 標頭開始）進行讀取時，工作階段 1 中 30 次有 8 次正確；在工作階段 3 中，重設後的第一次 讀取被 NACK，第二次從 `00h:46-48` 傳回 `80 00 00`。

9 疑難排解

表 8. 疑難排解

症狀	可能原因與處置
每次組合讀取都以 `FW_I3C_NACK_ADDRESS` 失敗	命令備妥太遲：檢查 DMA 中斷優先順序是否高於 I3C 中斷，以及其中是否執行了耗時的動作
128 位元組讀取失敗，分析儀顯示 308 個 SCL 時脈	讀取在 32 位元組處被截斷：主迴圈沒有供給傳送緩衝區。移除阻塞式 `printf` 及類似動作
讀取失敗後目標裝置停止回應，直到重設	接收一直被標記為進行中；韌體需要 `I3C1_Target_ReceiveCancel()`
上電後的第一次初始化以 `FW_I3C_BUS_INIT_NACK_RSTDAA` 失敗	重設後的第一次傳輸被 NACK；請再次初始化
MPLAB IPE 回報 "Connection Failed" 或 `Transmission on endpoint 2 failed (err = -150)`	需要重新插拔 PKoB4 的 USB

10 縮寫

表 9. 縮寫

術語	意義
BCR	Bus Characteristics Register（匯流排特性暫存器）
CCC	Common Command Code（通用命令碼）
CMIS	Common Management Interface Specification（通用管理介面規範）
DA	Dynamic Address（動態位址）
DCR	Device Characteristics Register（裝置特性暫存器）
DMA	Direct Memory Access（直接記憶體存取）
ENTDAA	Enter Dynamic Address Assignment（進入動態位址分配）
IBI	In-Band Interrupt（帶內中斷）
MRL	Max Read Length（最大讀取長度）
PID	Provisioned ID（預置 ID）
RSTDAA	Reset Dynamic Address Assignment（重設動態位址分配）
SDR	Single Data Rate（單資料速率）
Sr	Repeated START（重複 START）

11 關於本文件中原始碼的說明

本文件中展示並隨附封存檔中提供的範例程式碼僅供評估與參考。其依據 MIT 授權條款發布，授權條款全文包含在封存檔中。

封存檔 `AN0019-assets.zip` 包含 `cmis_target_check.py` 和 `cmis_image.h`。

12 修訂歷史

表 10. 修訂歷史

修訂版	日期	說明
0.1	29 October 2026	供內部審閱的草稿。
0.2	6 November 2026	主機指令碼改用 pycosmicsdk；工作階段 3 的結果；位址分配之前的 I2C 讀取改以一般 I2C 控制器量測。
1.0	6 November 2026	正式發布。

法律資訊（英文）

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. **I3C** 和 **MIPI** 是 MIPI Alliance, Inc. 的商標或註冊商標。 **CMIS** 和 **OIF** 是 Optical Internetworking Forum 的商標或註冊商標。 **Microchip**、**dsPIC**、**MPLAB** 和 **Curiosity** 是 Microchip Technology Incorporated 的商標或註冊商標。**Saleae** 和 **Logic** 是 Saleae, Inc. 的商標。