binho

應用筆記

# 使用 Binho Supernova 讀取 DDR5 SPD Hub 並透過 DIMM 邊帶驅動 Corsair RGB

AN0016

## 文件資訊

| | |
|---|---|
| **關鍵詞** | AN0016, Supernova, DDR5, SPD, SPD5118, SPD Hub, JESD300-5, 邊帶, I2C, I3C, SETAASA, RSTDAA, PMIC, 溫度感測器, Corsair, Vengeance, RGB, CRC-8 |
| **摘要** | DDR5 模組在其邊帶匯流排上配有一個 SPD5118 Hub：包含 SPD NVM、一個溫度感測器，以及通往模組上其他裝置的閘道。 本應用筆記使用 Binho Supernova 在 I2C 模式與 I3C 模式下讀取該 Hub，並驅動位於 Hub 後方的 Corsair DDR5 模組的 RGB 燈光控制器。文中提供暫存器存取序列、在一個測試平台上傳回正確資料的 I3C 速率、 一個在 I3C 暫存器寫入中出現並使匯流排停滯直到模組重新上電的故障，以及兩個基於 pycosmicsdk 的 Python 指令碼。 |

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

修訂版 1.0
6 NOVEMBER 2026

> **測試範圍**。本文件中的每個數字都來自 2026 年 10 月 6 日的一次測試平台作業， 使用一個模組。隨附指令碼由測試平台上使用的指令碼整合而成，之後在同一測試平台上執行（第 12 節）。

# 1 簡介

DDR5 DIMM 提供一條獨立於記憶體匯流排的兩線邊帶（HSCL、HSDA）。在該匯流排上最先回應的是 SPD5118 Hub；它保存模組的 SPD 資料、回報溫度，並將流量轉送到模組本地匯流排上的其他裝置，例如 PMIC，以及在具燈光的模組上的 RGB 控制器。Hub 以 I2C 模式啟動，可以切換到 I3C。

本筆記說明如何在沒有主機板的情況下，從主機轉接器讀取 Hub 及上述裝置。

## 1.1 範圍

本文件涵蓋：

- 識別 SPD5118 Hub、讀取其溫度感測器，以及讀取 1 KB 的 SPD NVM
- 讀取 PMIC 狀態，僅讀取
- 使用 SETAASA 將 Hub 切換到 I3C、以 SDR 讀取，並使用 RSTDAA 返回 I2C
- 在本測試平台上傳回正確資料的 I3C 速率，以及未傳回正確資料的那個速率
- 位於 Hub 後方的 Corsair RGB 控制器：使其離開韌體更新模式、逐顆 LED 設定顏色，以及啟動硬體效果

本文件不涵蓋：寫入 SPD NVM、透過 I3C 寫入 Hub 暫存器（第 5.3 節說明原因）、來自 Hub 的 帶內中斷（IBI）（已嘗試，未收到）、PEC，以及在 Hub 處於 I3C 模式時存取 RGB 控制器（未嘗試）。

## 1.2 適用裝置

**表 1.** 適用裝置

| 角色 | 元件 | 識別資訊 | 依據 |
|---|---|---|---|
| 主機轉接器 | Binho Supernova | 硬體版本 C，韌體 4.6.1 | 實測，本文件 |
| SPD Hub | 下述模組上的 SPD5118 | MR0-1 `51 18`，MR2 版本 `0x14` | 實測，本文件 |
| PMIC | 下述模組上的 PMIC | 位址 `0x48` | 唯讀狀態，實測 |
| RGB 控制器 | Corsair Vengeance RGB RS DDR5，6 顆 LED | VID `0x1B1C`；PID 在韌體更新模式下為 `0x0B00`，在一般模式下為 `0x0B01` | 實測，本文件 |
| 模組 | Corsair `CMG16GX5M2E6000Z36`（套組中的一條 DIMM） | SPD 位元組 2 `0x12`（DDR5），位元組 3 `0x02`（UDIMM） | 實測，本文件 |
| 其他 SPD5118 模組 | 任何 DDR5 UDIMM 或 SODIMM | MR0-1 `51 18` | 第 5.1 與 5.2 節遵循 JESD300-5；除一個模組外未經測試 |
| 其他 Corsair DDR5 RGB 模組 | | 其他 PID | 未測試 |

## 1.3 相關文件

**表 2.** 相關文件

| 文件 | 標題 |
|---|---|
| JESD300-5 | SPD5118 Hub and Serial Presence Detect Device Standard（JEDEC） |
| JESD400-5 | DDR5 Serial Presence Detect (SPD) Contents（JEDEC） |
| MIPI I3C Basic | Improved Inter Integrated Circuit Basic 規範 |
| OpenRGB | Corsair DRAM 控制器原始碼，第 5.4 節中 RGB 暫存器對應表的來源 |

# 2 系統概述

**表 3.** 邊帶上看到的裝置，Hub 處於 I2C 模式，100 kHz

| 位址 | 裝置 | 回應方式 |
|---|---|---|
| `0x50` | SPD5118 Hub | 暫存器指標 + 讀取；單純讀取也會回應 |
| `0x48` | PMIC | 暫存器指標 + 讀取 |
| `0x18` | Corsair RGB 控制器 | **僅限暫存器指標 + 讀取**；單純讀取不會被確認（ACK） |

該模組的 HSA 被設定為 0，使 Hub 位於 `0x50`。Hub 將傳往模組本地匯流排上裝置的傳輸轉送到 主機端；JESD300-5 將這兩個位址欄位稱為本地 ID（local ID）與主機 ID（host ID）。在 I2C 模式下，這種轉送不需要任何設定。PMIC 與 RGB 控制器都透過它存取。

以下兩個事實決定了下文的所有內容：

- **單純讀取的掃描找不到 RGB 控制器。** 以一位元組讀取掃描 `0x08` 到 `0x77`，只傳回 `0x48` 與 `0x50`。先寫入暫存器指標再讀取，才能在 `0x18` 找到它。
- **在本測試平台上，Hub 是一個不會被 ENTDAA 發現的 I3C 目標裝置。** 收到 SETAASA 時，它以其靜態位址作為動態位址；收到 RSTDAA 時返回 I2C。

# 3 硬體設定

## 3.1 所需設備

**表 4.** 所需設備

| 項目 | 說明 |
|---|---|
| Binho Supernova | 此處使用韌體 4.6.1 |
| DDR5 模組 | 第 1.2 節中的模組 |
| DIMM 轉接板 | 引出 HSCL、HSDA 與 GND，設定 HSA 腳位，並從其自身的 5 V 輸入提供 VIN_BULK |
| 轉接板的 5 V 電源 | 此處使用轉接板上的 USB-C |
| 4 線纜線 | 連接 Supernova I3C LV 連接器與轉接板 |

### 3.2 連接

**表 5.** Supernova 與 DIMM 轉接板之間的連接

| Supernova I3C LV 連接器 | DIMM 轉接板 | 必要 |
|---|---|---|
| SCL | HSCL | 是 |
| SDA | HSDA | 是 |
| GND | GND | 是 |

### 3.3 匯流排注意事項

匯流排運作於 **1.1 V**（設定 1100 mV，回讀 1106 mV）。Supernova 僅依據電壓選擇 I3C 連接器：

**表 6.** Supernova I3C 連接器選擇

| 要求電壓 | 連接器 |
|---|---|
| 800 到 1199 mV | LV |
| 1200 到 3300 mV | HV |

> **注意：** 要求 1200 mV 會將匯流排切換到 HV 連接器。模組接在 LV 連接器上時，將沒有任何 裝置回應。請要求 1100 mV。
>
> **注意：** 模組需要在 VIN_BULK 上提供 5 V。該電壓由轉接板提供，轉接器不提供。請先為模組 上電，再啟動匯流排。要清除第 5.3 節中的停滯，請將模組重新上電，而不是轉接器。

## 4 軟體設定

需求：Python 3 與 pycosmicsdk（最新版本）。

```
pip install --upgrade pycosmicsdk
python ddr5_spd.py --help
python ddr5_rgb.py --help
```

兩個指令碼只使用 pycosmicsdk 的公開呼叫：`Device.open`、`I3cController.set_voltage`、`bring_up`、`legacy_i2c_read`、`legacy_i2c_write`、`sdr_read`、`ccc.setaasa` 與 `ccc.rstdaa`。兩個指令碼都接受 `--serial` 用於選擇轉接器，以及 `--voltage-mv`（預設 1100）。測試平台使用的是 Ubuntu 22.04 主機，pycosmicsdk 1.6.0 與 Python 3.10。

# 5 暫存器與協定

## 5.1 I2C 模式下的 SPD5118 Hub

所有存取都是一個一位元組暫存器指標，後接一次讀取。

**表 7.** 使用的 Hub 暫存器

| 暫存器 | 本模組上的讀值 | 意義 |
|---|---|---|
| MR0-1 | `51 18` | 裝置類型，SPD5118 |
| MR2 | `0x14` | 裝置版本 |
| MR5 | `0x03` | 能力：Hub 與溫度感測器 |
| MR11 | `0x00` | I2C 傳統模式：位元 2:0 選擇 NVM 頁，位元 3 選擇 2 位元組定址 |
| MR12-13 | `FF 7F` | NVM 寫入保護，每個 64 位元組區塊一個位元 |
| MR18 | I2C 下為 `0x00`，I3C 下為 `0x20` | 位元 5 = I3C 模式 |
| MR49-50 | 兩個位元組 | 溫度 |

**溫度。** MR50 位元 4:0 與 MR49 位元 7:0 組成一個 13 位元欄位，其中位元 1:0 未使用。其餘 11 位元為二補數，每 LSB 為 0.25 °C：

```
raw = (((mr50 & 0x1F) << 8) | mr49) >> 2
if raw & 0x400:
    raw -= 0x800
celsius = raw * 0.25
```

在測試平台上，該模組讀值為 27 到 29 °C。其限值暫存器讀值為 55 °C（高）與 85 °C（臨界）。

**NVM。** 1024 位元組的 NVM 由八個 128 位元組的頁組成。在 1 位元組定址下，位元 7 設定的指標 定址 NVM，位元 6:0 為頁內偏移，MR11 位元 2:0 為頁號。讀取全部內容的方法：保存 MR11，依序將頁 0 到 7 寫入其中，每次從指標 `0x80` 讀取 128 位元組，然後寫回保存的值。MR11 是唯一被寫入的暫存器。該模組的傾印解碼為位元組 2 = `0x12`（DDR5）與位元組 3 = `0x02`（UDIMM），且位元組 0 到 509 上的 CRC-16（多項式 `0x1021`，初始值 0）與位元組 510 及 511 中儲存的值相符。

**PMIC。** 暫存器 R32 讀值為 `0x80`（穩壓器已啟用），且未設定任何故障位元。PMIC 僅被讀取。

## 5.2 I3C 模式下的 SPD5118 Hub

1. 以 I3C 模式啟動控制器。
2. 廣播 SETAASA，裝置表中包含 `0x50`。MR18 從 `0x00` 變為 `0x20`。
3. 使用暫存器指標以 SDR 讀取：`sdr_read(0x50, n, subaddress=bytes([reg]))`。

4. 廣播 RSTDAA。透過 I2C 讀取時，MR18 再次為 `0x00`。

`init_bus()` 使用 ENTDAA 發現目標裝置，在本測試平台上未能啟動 Hub。請使用 SETAASA。

**表 8.** 1.1 V 下從 Hub 進行的 I3C SDR 讀取，依推挽速率

| 推挽速率 | 結果 |
|---|---|
| 1 MHz | 20/20 |
| 5 MHz | 50/50（首次執行為 20/20） |
| 6.25 MHz | 50/50 |
| 7.5 MHz | 50/50 |
| 10 MHz | 50/50 |
| 12.5 MHz | 9/50 正確；其餘 41 次在成功狀態下傳回錯誤資料 |

12.5 MHz 下的失敗並未被回報為錯誤：讀取已完成，但資料錯誤。轉接板佈線在 1.1 V 下可能是限制因素。**在較長的邊帶佈線上請使用 4 MHz 或更低。** 已量測 1 MHz（20/20）以及 5 到 10 MHz；兩者之間的速率未量測。`ddr5_spd.py --i3c` 預設使用 1 MHz。

## 5.3 透過 I3C 寫入暫存器

> **注意：** 在 I3C SDR 模式下對 Hub 的暫存器寫入未依寫入的值生效，隨後匯流排停滯。寫入為 26.5 °C 的溫度限值回讀為 48 °C；寫入 MR27 的 `0x01` 回讀為 `0x10`。此後，每個 I3C 操作都傳回 `FW_I3C_ENGINE_BUSY`。重新開機 Supernova 無法清除該狀態，將 LV 連接器重新上電也不行。只有將 模組重新上電才能清除。原因（Hub 的 I3C 寫入格式或主機端）尚未確定。在 I3C 模式下僅進行讀取。 若需要寫入，請在 I2C 模式下進行，或先寫入一個無害的暫存器並以回讀確認。

## 5.4 Corsair RGB 控制器

以下暫存器對應表與封包格式來自 OpenRGB 的 Corsair DRAM 控制器（GPL-2.0）。本文件中的指令碼 依據該描述重新實作；未複製任何 OpenRGB 程式碼。在 Hub 處於 I2C 模式時，透過 `0x18` 存取該控制器。

每次存取都是寫入 `[register, value]`，或暫存器指標後接一位元組讀取。**控制器閒置一段時間後的第一次存取可能被拒絕一次；請重試。**

**表 9.** 使用的 RGB 控制器暫存器

| 暫存器 | 存取 | 用途 |
|---|---|---|
| `0x0B`, `0x21` | 寫入 `0x00` | 重設封包緩衝區及其 CRC |
| `0x20` | 寫入，每次存取一個位元組 | 封包緩衝區 |
| `0x42` | 讀取 | 自重設以來載入之位元組的 CRC-8 |
| `0x82` | 寫入 | 套用：`0x01` 效果封包，`0x02` 顏色緩衝區 |
| `0x30` | 讀取 | 套用過程中位元 3 設定 |
| `0x61` 然後 `0x21` | 寫入 `0x00` | 選擇裝置資訊區塊 |
| `0x40` | 讀取，32 次 | 裝置資訊區塊 |
| `0x23` | 寫入 `0x01` | 離開韌體更新模式 |

CRC 為 CRC-8，多項式 `0x07`，初始值 0。套用之前請將其與 `0x42` 比較；不相符表示有位元組遺失或重複（重試寫入可能造成此情況），此時不得套用該封包。

**裝置資訊區塊。** 位元組 0-1 為 VID，小端序；位元組 2-3 為 PID；位元組 28 為協定版本。

**韌體更新模式。** 兩端的 LED 都亮紅色，是 Corsair 文件中所述的韌體更新（開機載入程式）模式 指示。在該狀態下，模組回報 PID `0x0B00` 與協定 1，並接受效果封包（CRC 相符），但 LED 不變化。向 `0x23` 寫入一次 `0x01` 後，它變為 PID `0x0B01`、協定 4，原廠彩虹效果隨之亮起（圖 1）。模組 如何進入該模式尚未確定；它在上電後不久曾被探測過。

> **注意：** 切勿向暫存器 `0x23` 寫入 `0x00`。

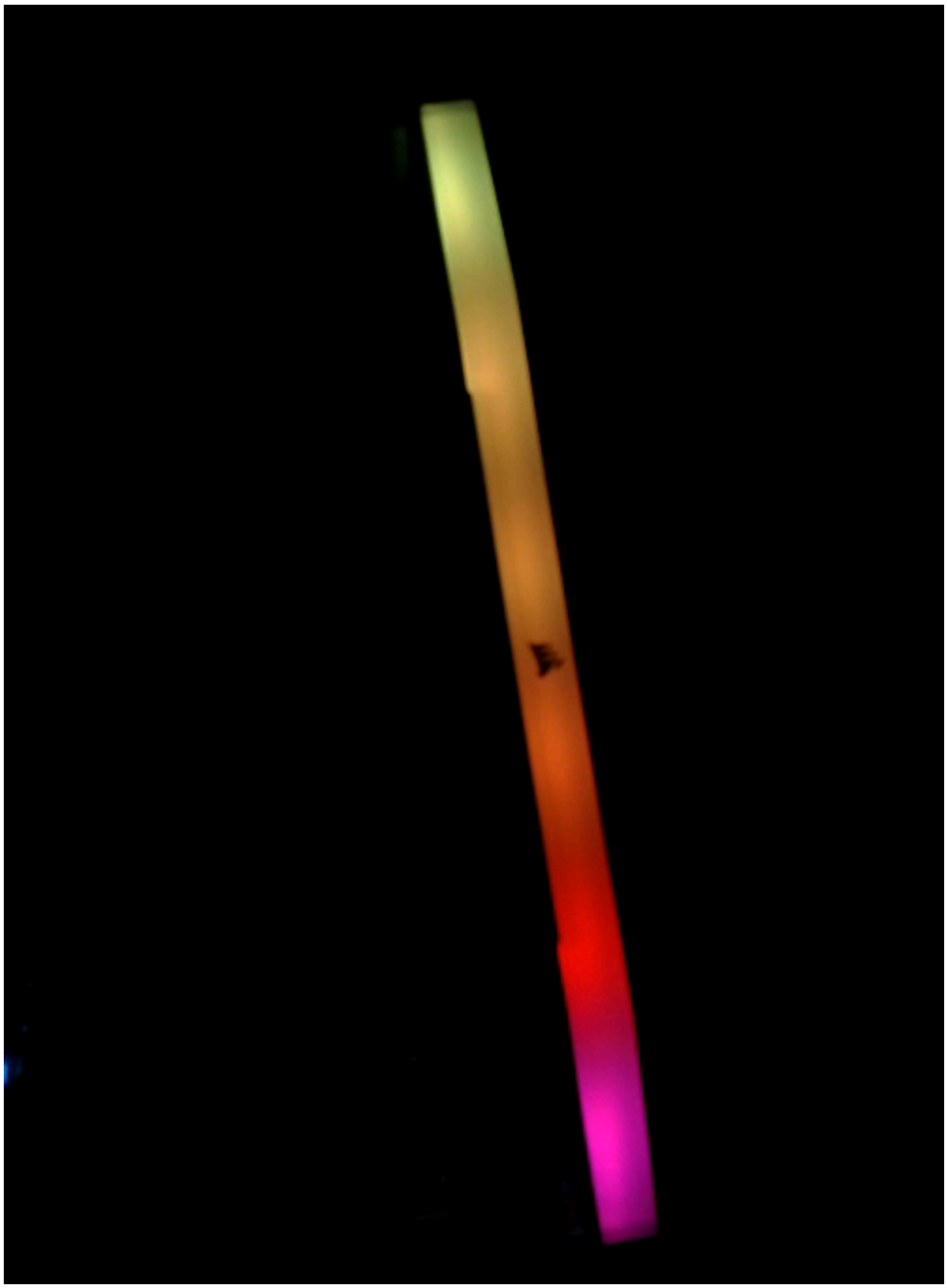


**圖 1.** 離開韌體更新模式後的模組：原廠彩虹效果。

**顏色緩衝區。** 六個 `R, G, B, 0xFF` 項目，每顆 LED 一個，共 24 位元組，在重設後載入到 `0x20`，檢查 CRC，並以 `0x82 = 0x02` 套用。由此可設定單一顏色與逐顆 LED 顏色（圖 2）。該緩衝區 僅在靜態模式下顯示：在動畫效果下，它會被接受（包括 CRC），但會被忽略。因此 `ddr5_rgb.py color` 會先送出靜態效果（模式 `0x10`）。

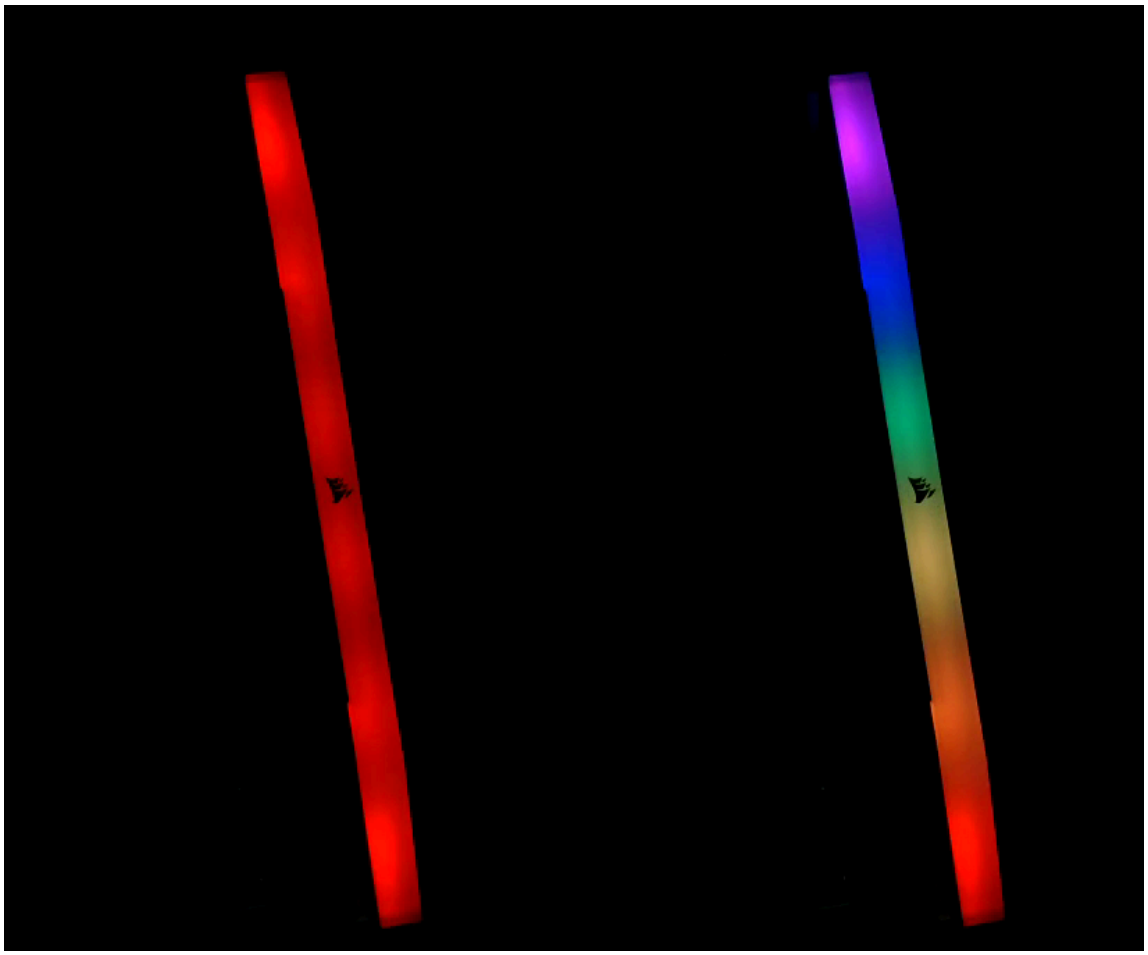


**圖 2.** 顏色緩衝區：六顆 LED 全部為紅色（左），每顆 LED 一種顏色（右）。

**效果封包。** 20 位元組，以 `0x82` = `0x01` 套用：

```
mode, speed, 0x01, direction, R1, G1, B1, brightness, R2, G2, B2, brightness, 0 x 8
```

**表 10.** 效果模式

| 模式 | 代碼 | 在本模組上觀察到 |
|---|---|---|
| 彩虹波（rainbow wave） | `0x03` | 是 |
| 靜態（static） | `0x10` | 已套用，但維持先前的顏色；請改用顏色緩衝區 |
| 彩虹（rainbow） | `0x08` | 未觀察到 |
| 脈衝（pulse） | `0x01` | 未觀察到 |

# 6 操作程序

## 6.1 上電與連接

透過轉接板為模組供電。將纜線連接到 Supernova I3C LV 連接器。

## 6.2 識別 Hub 並讀取其溫度

```
python ddr5_spd.py --serial <serial>
```

測試平台上的輸出：

```
MR0-1 5118  (SPD5118 reads 5118)
MR2 revision 0x14  MR18 0x00  (bit 5 = I3C mode)
MR49-50 CC 01 = 28.75 C
```

## 6.3 保存 NVM（選用）

將 NVM 保存到您選擇的檔案中。請將該檔案視為可識別該模組的資訊：它包含模組的序號。

```
python ddr5_spd.py --serial <serial> --nvm my_module_spd.bin
```

指令碼會還原 MR11，並回報 SPD CRC 是否相符：

```
MR0-1 5118  (SPD5118 reads 5118)
MR2 revision 0x14  MR18 0x00  (bit 5 = I3C mode)
MR49-50 CC 01 = 28.75 C
NVM: 1024 bytes to my_module_spd.bin, MR11 restored to 0x00
SPD byte 2 0x12 (0x12 = DDR5), byte 3 0x02, CRC 0-509 calc 0x45BF stored 0x45BF OK
```

## 6.4 以 I3C 模式讀取 Hub（選用）

指令碼使用 SETAASA 進入 I3C 模式，以 SDR 讀取，然後使用 RSTDAA 將 Hub 返回 I2C：

```
python ddr5_spd.py --serial <serial> --i3c --i3c-mhz 1
```

在 I3C 讀取期間 MR18 讀值為 `0x20`，RSTDAA 之後為 `0x00`：

```
MR0-1 5118  (SPD5118 reads 5118)
MR2 revision 0x14  MR18 0x00  (bit 5 = I3C mode)
MR49-50 CC 01 = 28.75 C
I3C SDR: MR18 0x20  MR0-1 5118  temperature 29.00 C
after RSTDAA (I2C): MR18 0x00
```

在轉接器韌體 4.6.1 上，每次轉接器重設後此操作可執行 11 次；參見第 9 節。

## 6.5 讀取 RGB 控制器的裝置資訊

```
python ddr5_rgb.py --serial <serial> info
```

測試平台上的輸出，控制器處於一般模式：

```
VID 0x1B1C PID 0x0B01 protocol 4
```

## 6.6 離開韌體更新模式

若 PID 為 `0x0B00`（兩端為紅色），請離開韌體更新模式。指令碼僅在 PID 為 `0x0B00` 時才寫入。

```
python ddr5_rgb.py --serial <serial> leave-bootloader
```

控制器處於韌體更新模式時的輸出：

```
before: PID 0x0B00 protocol 1
after:  PID 0x0B01 protocol 4
```

以及控制器已處於一般模式時的輸出：

```
before: PID 0x0B01 protocol 4
not in firmware-update mode; nothing written
```

## 6.7 設定顏色或效果

```
python ddr5_rgb.py --serial <serial> color 255,0,0
python ddr5_rgb.py --serial <serial> color 255,0,0 255,80,0 255,220,0 0,255,0
0,0,255 160,0,255
python ddr5_rgb.py --serial <serial> effect wave 255,0,0 0,0,255 --speed 2
python ddr5_rgb.py --serial <serial> demo
```

這些命令成功時不輸出任何內容；若控制器的 CRC 不相符，則以錯誤結束。

示範序列見 `figures/an0016-video1-rgb-demo.mp4`（42 s），由該指令碼所整合自的測試平台指令碼錄製。

## 7 參考

**表 11.** ddr5_spd.py 選項

| 選項 | 作用 |
|---|---|
| `--serial` | 轉接器序號；預設為找到的第一個轉接器 |
| `--voltage-mv` | 匯流排電壓，預設 1100（LV 連接器） |
| `--nvm FILE` | 透過 MR11 將 1024 位元組 NVM 讀入 FILE，然後還原 MR11 |
| `--i3c` | SETAASA，以 SDR 讀取 MR18、MR0-1 與溫度，RSTDAA |
| `--i3c-mhz` | `--i3c` 的推挽速率：1、3.125 或 5；預設 1 |

**表 12.** ddr5_rgb.py 命令

| 命令 | 作用 |
|---|---|
| `info` | 檢查資訊區塊的 CRC 後，印出 VID、PID 與協定 |
| `leave-bootloader` | 若 PID 為 `0x0B00`，寫入 `0x23 = 0x01`；印出前後的 PID |
| `color R,G,B` | 透過顏色緩衝區將六顆 LED 設為同一種顏色 |
| 帶六個 `R,G,B` 的 `color` | 每顆 LED 一種顏色 |
| `effect MODE C1 [C2]` | 效果封包；MODE 為 static、rainbow、wave 或 pulse；`--speed` 0 到 2 |
| `demo` | 紅、綠、藍、白，逐顆 LED 彩虹，追逐效果，然後彩虹波 |

兩個指令碼對每次存取最多重試三次（`ddr5_spd.py`）或五次（`ddr5_rgb.py`）。

# 8 量測結果

## 8.1 測試配置

**表 13.** 測試配置

| 項目 | 值 |
|---|---|
| 轉接器 | Binho Supernova，硬體版本 C，韌體 4.6.1 |
| 匯流排 | I3C LV 連接器，要求 1100 mV，回讀 1106 mV |
| I2C 模式 | 100 kHz 開汲極 |
| I3C 模式 | SETAASA，SDR，推挽速率見第 5.2 節 |
| 模組 | Corsair `CMG16GX5M2E6000Z36`，一條 DIMM，安裝在自備 5 V 電源的 DIMM 轉接板上 |
| 主機 | Ubuntu 22.04，pycosmicsdk 1.6.0，Python 3.10 |
| 日期 | 2026 年 10 月 6 日 |

## 8.2 結果

**表 14.** 結果

| 量測項目 | 結果 |
|---|---|
| Hub 識別，MR0-1 | `51 18` |
| 溫度，MR49-50 | 27 到 29 °C |
| 透過 MR11 讀取 NVM | 1024 位元組，位元組 0-509 上的 SPD CRC 相符；3/3 逐位元組相同 |
| SETAASA，MR18 | `0x00` 變為 `0x20` ；RSTDAA 後回到 `0x00` |
| I3C SDR 讀取，1 到 10 MHz | 1 MHz 下 20/20；5、6.25、7.5 與 10 MHz 下各 50/50 |
| I3C SDR 讀取，12.5 MHz | 9/50 正確，41 次在成功狀態下傳回錯誤資料 |
| I3C 暫存器寫入 | 未依寫入的值生效；匯流排停滯，直到模組重新上電 |
| 來自 Hub 的帶內中斷 | 已嘗試，未收到 |
| 在 `0x18` 找到 RGB 控制器 | 僅限暫存器指標 + 讀取 |
| 離開韌體更新模式 | PID `0x0B00` 變為 `0x0B01` ，協定 1 變為 4；在兩天中共 2 次（一次使用測試平台指令碼，一次使用 `ddr5_rgb.py` ），每次都發現控制器處於韌體更新模式 |
| 顏色緩衝區，單一顏色與逐顆 LED | 各 10/10，CRC 相符；從執行中的效果切換到單一顏色，攝影機確認 9/9 |
| 效果封包，彩虹波 | 10/10 |

這些是一個模組在一個測試平台上的結果，不是規格。隨附指令碼的計數見第 12 節。尚未進行重新上電 後的開機載入程式離開測試。

# 9 疑難排解

**表 15.** 疑難排解

| 症狀 | 原因 | 處理 |
|---|---|---|
| 掃描找到 `0x48` 與 `0x50`，但 `0x18` 處沒有裝置 | RGB 控制器不確認單純讀取 | 寫入暫存器指標後讀取，如 `ddr5_rgb.py info` 所做 |
| 對 `0x18` 的第一次存取被拒絕，下一次成功 | 控制器處於閒置狀態 | 重試；指令碼會自動重試 |
| 兩端 LED 為紅色，效果封包被接受但沒有任何變化 | 控制器處於韌體更新模式，PID `0x0B00` | `ddr5_rgb.py leave-bootloader` |
| 之後控制器在未重新上電的情況下又回到韌體更新模式 | 在廣播 CCC 流量（RSTDAA/ENTDAA）與轉接器重設之後出現；原因未查明 | 執行 `ddr5_rgb.py info`；若 PID 為 `0x0B00`，再次執行 `leave-bootloader` |
| 對 Hub 的每次 I2C 讀取都傳回 `51 FF`，無論暫存器為何 | Hub 被之前的 SETAASA 留在 I3C 模式（MR18 位元 5 設定） | 送出 RSTDAA，或將模組重新上電 |
| 多次 I3C 作業後 SETAASA 被拒絕：目標裝置表已滿（見下文） | 轉接器韌體 4.6.1 在 RSTDAA 之後的每次 SETAASA 都新增一個目標裝置表項目；第 12 次失敗（3/3 次執行） | 重設轉接器（`reboot` 或重新插拔）；已在轉接器韌體中追蹤 |
| 顏色被接受（CRC 相符），但 LED 繼續播放動畫 | 控制器正在執行動畫效果；顏色緩衝區僅在靜態模式下顯示 | `ddr5_rgb.py color` 會先選擇靜態模式；使用原始寫入時，請先送出靜態效果，再送出顏色 |
| `0x42` 處 CRC 不相符 | 載入時有位元組遺失或重複 | 重新送出整個封包；不要套用 |
| `init_bus()` 找不到 Hub | 在本測試平台上 Hub 不會被 ENTDAA 發現 | 使用 `ccc.setaasa([0x50])` |
| 完全沒有裝置回應 | 1200 mV 或更高會選擇 HV 連接器 | 要求 1100 mV |
| I3C 模式下資料錯誤但狀態為成功 | 推挽速率相對於佈線過高 | 4 MHz 或更低 |
| I3C 寫入後每個 I3C 操作都傳回 `FW_I3C_ENGINE_BUSY` | Hub 停滯（第 5.3 節） | 將模組重新上電；重新開機 Supernova 無法清除 |
| NVM 傾印 CRC 不相符 | MR11 被變更，或傾印過程中某次讀取失敗 | 重新執行；檢查 MR11 位元 3 為 0 |

目標裝置表已滿導致的拒絕回報為 `FW_NOT_ENOUGH_SPACE_IN_TARGET_DEVICES_TABLE`。

# 10 縮寫

**表 16.** 縮寫

| 術語 | 意義 |
|---|---|
| CCC | 通用命令代碼（Common Command Code，I3C） |
| CRC | 循環冗餘檢查（Cyclic redundancy check） |
| DIMM | 雙列直插式記憶體模組（Dual in-line memory module） |
| ENTDAA | 進入動態位址分配（Enter Dynamic Address Assignment，I3C CCC） |
| HSA | 主機邊帶位址設定腳位（Host sideband address strap） |
| HSCL, HSDA | 主機邊帶時脈與資料（Host sideband clock and data） |
| LED | 發光二極體（Light-emitting diode） |
| LV, HV | Supernova 的低電壓與高電壓 I3C 連接器 |
| MR | SPD5118 的模式暫存器（Mode register） |
| NVM | 非揮發性記憶體（Non-volatile memory） |
| PEC | 封包錯誤檢查（Packet error check） |
| PID, VID | 產品 ID，廠商 ID（Product ID, vendor ID） |
| PMIC | 電源管理積體電路（Power management integrated circuit） |
| RSTDAA | 重設動態位址分配（Reset Dynamic Address Assignment，I3C CCC） |
| SDR | 單倍資料速率（Single data rate，I3C） |
| SETAASA | 將所有位址設為靜態位址（Set All Addresses to Static Address，I3C CCC） |
| SPD | 序列存在偵測（Serial presence detect） |
| UDIMM | 無緩衝 DIMM（Unbuffered DIMM） |
| VIN_BULK | DIMM 的 5 V 輸入電源 |

# 11 關於文件中原始碼的說明

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

**表 17.** 資源封存檔的內容

| 檔案 | 說明 |
|---|---|
| `AN0016.md` | 本文件的 Markdown 原始檔 |
| `assets/ddr5_spd.py` | Hub 識別、溫度、NVM 讀取、I3C 讀取檢查 |
| `assets/ddr5_rgb.py` | RGB 控制器：裝置資訊、離開韌體更新模式、顏色、效果、示範 |
| `assets/LICENSE` | 涵蓋範例程式碼的 MIT 授權條款 |
| `figures/an0016-video1-rgb-demo.mp4` | 示範序列的 42 s 影片 |

不提供 SPD 傾印：一份傾印可識別一個具體的實體模組。

## 12 修訂歷史

**表 18.** 修訂歷史

| 修訂版 | 日期 | 說明 |
|---|---|---|
| 0.1 | 2026 年 10 月 7 日 | 初始草稿。第 8 節的首批量測使用了較早的、獨立的指令碼。隨附指令碼在同一測試平台上執行：`ddr5_spd.py` 在 I2C 模式下 10/10；`--nvm` 3/3，逐位元組相同，SPD CRC 相符；`--i3c` 在 1 MHz 下 10/10、在 5 MHz 下 10/10（批次之間重設轉接器，見第 9 節），每次 Hub 都回到 I2C 模式；`ddr5_rgb.py` `info` 10/10，`color` 單一顏色 10/10、逐顆 LED 10/10，`effect wave` 10/10，`demo` 1/1，`leave-bootloader` 1/1（控制器曾有一次處於韌體更新模式）。從執行中的效果送出單一顏色：攝影機確認 9/9 相符。記錄的輸出：`_shared/tools/documented/ddr5-*.txt`。 |
| 1.0 | 2026 年 11 月 6 日 | 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** 與 **MIPI** 是 MIPI Alliance, Inc. 的商標或註冊商標。 **JEDEC** 是 JEDEC Solid State Technology Association 的商標。 **Corsair** 與 **Vengeance** 是 CORSAIR MEMORY, Inc. 的商標或註冊商標。 **OpenRGB** 是一個開源專案的名稱，此處僅用於識別該專案。