

Relay Control Guide

Daniel 2017/4/17 Initial Version

New PEB Relay 結構

每一片 PEB 主板上共有 64 顆 Channel Relay、32 個 UR Relay Control，皆由 Main FPGA 所控制。每一個 Channel 都有獨立的 Relay 開關，每一片 PEB Board 有 64 個 Channel 故有 64 個 Relay。另外有 32 個 UR Relay 控制(僅控制，未實際裝 Relay)藉由面板 Connector 輸出，這些 Relay 控制可以用作 Diag. Board 或是 User 的 DUT Board 控制需求使用。

Main FPGA 透過串列傳輸介面控制 Relay Controller，每一筆 command 實際需要 3us 處理，故需要軟體 delay 或是 polling flag 來確認指令完成。另外由於採用 shift register 介面，relay 狀態回讀時是回讀“上一次”command 所帶的 address。因為 Relay 控制通常沒有速度要求，故回讀時可以固定發送兩次 Dummy Write。

Main FPGA Relay Control Register MAP:

W 0xE0	CH_RLY [7:0]	Bit[0] = CH_RLY[0], Bit[1] = Protection Bit of Bit[0] Bit[2] = CH_RLY[1], Bit[2] = Protection Bit of Bit[2] ...
W 0xE1	CH_RLY [15:8]	
W 0xE2	CH_RLY [23:16]	
W 0xE3	CH_RLY [31:24]	
W 0xE4	CH_RLY [39:32]	
W 0xE5	CH_RLY [47:40]	
W 0xE6	CH_RLY [55:48]	
W 0xE7	CH_RLY [63:56]	
W 0xE8	USR_RLY [7:0]	
W 0xE9	USR_RLY [15:8]	
W 0xEA	USR_RLY [23:16]	
W 0xEB	USR_RLY [31:24]	
W 0xEF	Reset All Relays to OFF (0) // Reset 指令並不會發出 Protocol，所以後面不需要 Polling 0xEF	
R 0xE0	Relay Read Back Data	if Relay Control Address = 0: Bit[0] = CH_RLY[0], Bit[1] = always 0 Bit[2] = CH_RLY[1], Bit[2] = always 0 ... If Relay Control Address = A: Bit[0] = USR_RLY[16], Bit[1] = always 0 Bit[2] = USR_RLY[17], Bit[2] = always 0 ... // Read 指令並不會發出 Protocol，所以後面不需要 Polling 0xEF
R 0xEF	Relay Controller Working (Bit[0]) // 每次 Write Relay 指令後都應該 Polling Working Bit[0]	

更改 CH_RLY[23] = OFF, CH_RLY[16] = ON 範例:

```
R  0xEF    Polling_Data  // Polling 是否工作中，直到回讀 Data Bit[0] = 0 才可繼續下面指令
W  0xE2    16'b1100000000000011  // Protection Bit 為"1"才可以寫入，避免影響既有的其
                                     //他 Relay 設定。而 Bit[0]表示 CH_RLY[16]、Bit[14]表示
                                     // CH_RLY[23]
```

回讀 USR_RLY[7:0] 狀態範例:

```
R  0xEF    Polling_Data[0] = 0
W  0xE8    16'b0000000000000000  // Dummy Write (Protection Bit 皆為 0)，
                                     //目的為送入 Address (USR_RLY[7:0]為 0xE8)

R  0xEF    Polling_Data[0] = 0
W  0xE8    16'b0000000000000000  // Dummy Write (Protection Bit 皆為 0)，
                                     //再送一次 Dummy Write 把 Data Shift 出來

R  0xEF    Polling_Data[0] = 0  //確認 Command 完成
R  0xE0    ReadBack_Data        //回讀 USR_RLY[7:0]的 ON OFF 狀態
```

快速掃描所有 Relay 狀態:

```
R  0xEF    Polling_Data[0] = 0
W  0xE0    16'b0000000000000000  // 預備讀取 CH_RLY [7:0]
R  0xEF    Polling_Data[0] = 0
W  0xE1    16'b0000000000000000  // 預備讀取 CH_RLY [15:8]
R  0xEF    Polling_Data[0] = 0
R  0xE0    ReadBack_Data          // 回讀 CH_RLY[7:0]的 ON OFF 狀態
W  0xE2    16'b0000000000000000  // 預備讀取 CH_RLY [23:16]
R  0xEF    Polling_Data[0] = 0
R  0xE0    ReadBack_Data          // 回讀 CH_RLY[15:8]的 ON OFF 狀態
W  0xE3    16'b0000000000000000  // 預備讀取 CH_RLY [31:24]
R  0xEF    Polling_Data[0] = 0
R  0xE0    ReadBack_Data          // 回讀 CH_RLY[23:16]的 ON OFF 狀態
...
```