

NX1 使用 ICE 仿真的注意事項

- 內容：** NX1 使用 ICE 仿真時，當 NX1 停下時的週邊動作說明。
- 原因：** 使用 NX1_FDB Ver.A/B 或 NX1_ICE Ver.A + NX1_COB VerA/B 的開發環境來偵錯，當 ICE 停下來時，有些週邊功能的動作並不會隨著停止，以致於所取得的相關暫存器數值非當時停止時的數值。
- 方法：** 當使用這二個工具，建議使用者將需要觀察的資料透過 UART 介面上傳終端機進行觀察動作，盡量不要使用 ICE Debug Mode 的停止功能。

[Example 1]

NX1_FDB Ver.A/B 開發環境，使用 ADC with Timer Trigger 功能，則當 NX1 停下時，此時若使用者觀察 ADC 暫存器的狀態，則會發現 ADC 的 FIFO 已經滿了及中斷旗標會一直被舉著，這是由於 Timer 依然持續動作，且觸發 ADC 轉換，造成短時間就填滿了 FIFO。

上述條件，若再使用 Auto Scan Mode with FIFO 的條件下，則當 NX1 停下時，將可能使得 Channel 的控制順序錯亂。

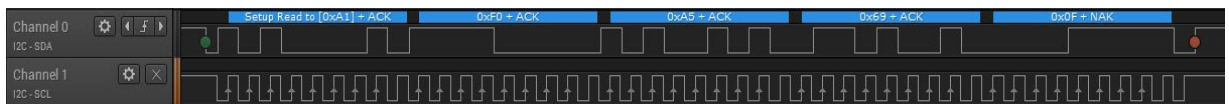
NX1_FDB Ver.C 和 NX1_ICE Ver.A + NX1_COB Ver.C 有針對 ICE Debug Mode 重新定義了週邊功能的動作行為，讓 NX1 週邊也停下來然後再繼續執行，以盡量和 Free Run 行為一致。

此版本的修改，使用者需注意在應用於 I²C Slave Mode 時，當 NX1 停止後，此時 Master 發出 Read 指令，NX1 還是會持續執行 I²C 控制流程，不論 Master 要求多少筆資訊，NX1 皆只會回覆 FIFO 的第一筆資料，且因 FIFO Count 不會減少，後續再執行 Free Run 後，會因為 FIFO 的資料移位而造成錯誤。

建議使用者在 I²C 完成階段傳輸後，再設置斷點，若仿真必要於傳送過程中設定斷點時，使用者在停下來看完狀態後，建議讓程序重新執行，以避免後續執行程式錯誤，此點必須注意。

[Example 2]

Free Run 實際狀況: I²C FIFO 狀態 Data[4] = { 0xF0 ,0xA5 ,0x69 ,0x0F }



ICE Debug Mode: 由於 EV chip 已停下來，此時 Master 來跟 NX1 (Slave)要資料，都只會得到 FIFO 的第一筆資料，如下圖



Data 只傳送目前 FIFO 當前第一筆資料，造成傳送錯誤。