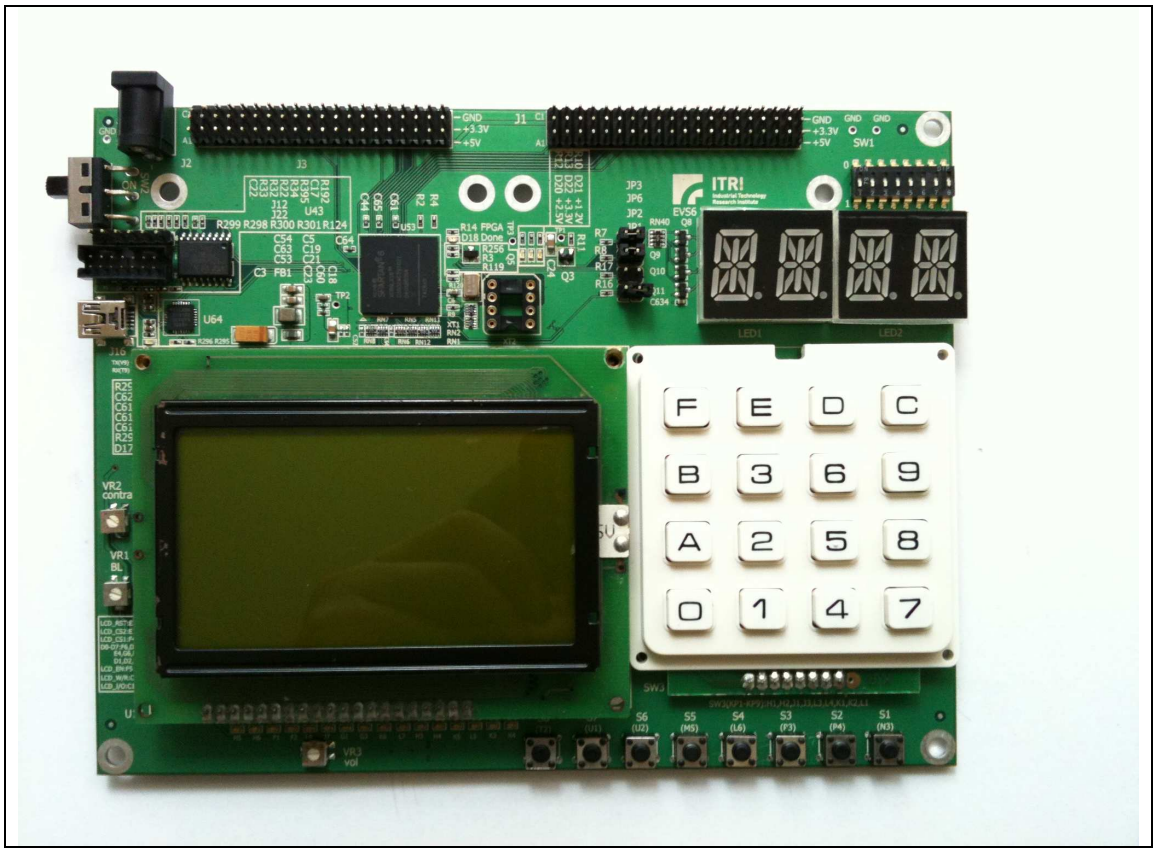
EVS6
使用手冊

Version : 1.0



＊＊＊＊＊＊＊＊〔目　　錄〕＊＊＊＊＊＊＊＊

第一章 概論

EVS6開發套件,採用Xilinx最新一代Spartan6系列所設計的FPGA開發板,搭配相對應的週邊介面,可使使用者方便驗證程式設計碼,板子上預留使用者可擴充的輸入輸出接腳,將可提供FPGA使用者一個完整且容易上手的實驗平台.這份手冊將會含括這片實驗板上所有元件的設定及特性說明.

主要元件及功能

EVS6 開發套件包含由 EVS6 主板與電源供應器,內附使用手冊與使用範例,組成一個完整的實驗板,使用者需自備 JTAG 下載線與個人電腦與 Xilinx ISE 開發軟體.

1-1. EVS6 套件主要功能

EVS6 板包含了以下主要元件:

- Xilinx Spartan-6 XC6SLX9 或 XC6SLX16 FPGA 包裝爲 CS324
  . 內含 16/32 個 18K-bit 的 Block RAMs (共 216K bits)
  . 內含 32 個 DSP48A1 處理單元
  . 內含 2 個 Clock Managers

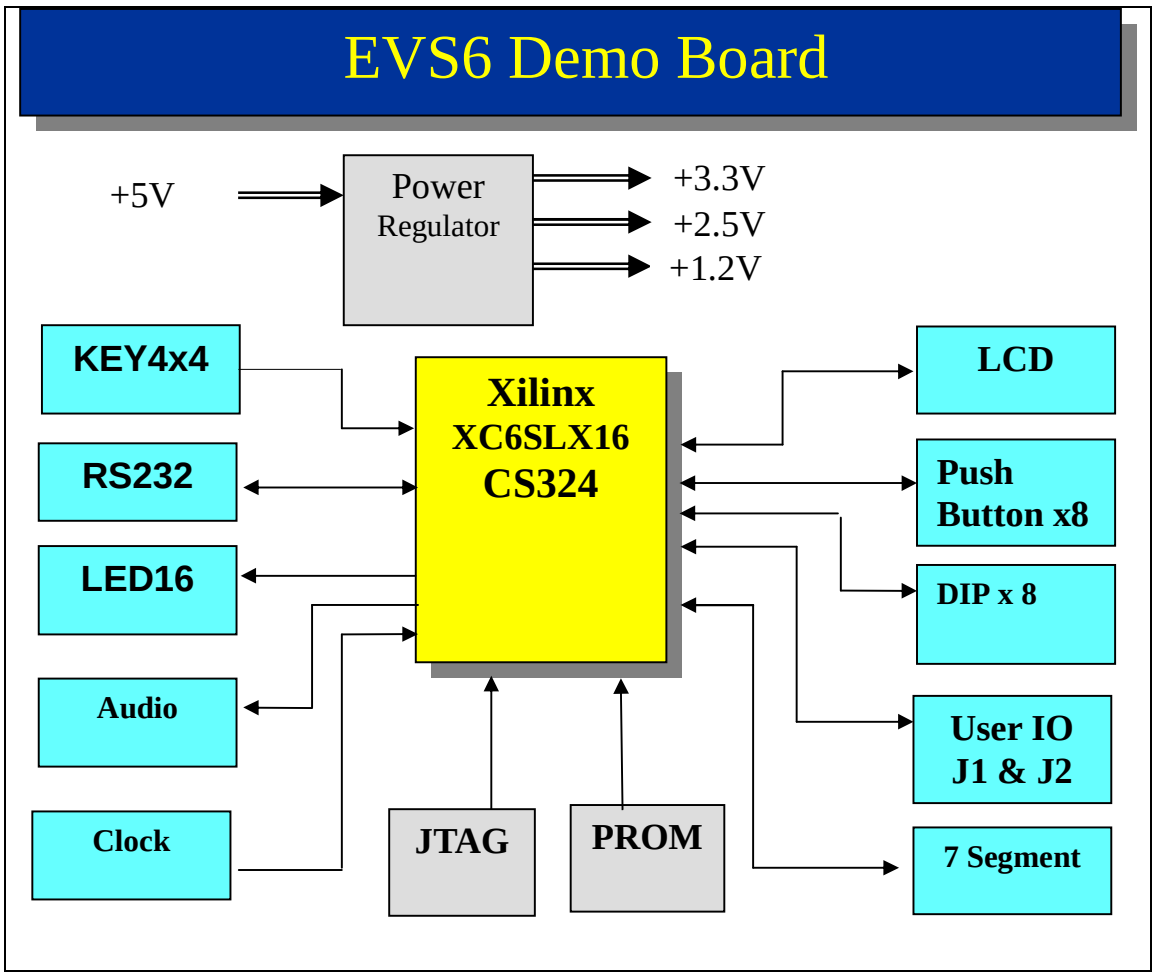
Table 1: **Spartan-6 FPGA Feature Summary by Device**

Device	Logic Cells[1]	Configurable Logic Blocks (CLBs)			DSP48A1 Slices[3]	Block RAM Blocks		CMTs[5]	Memory Controller Blocks	Endpoint Blocks for PCI Express	Maximum GTP Transceivers	Total I/O Banks	Max User I/O
		Slices[2]	Flip-Flops	Max Distributed RAM (Kb)		18 Kb[4]	Max (Kb)						
XC6SLX4	3,840	600	4,800	75	8	12	216	2	0	0	0	4	120
XC6SLX9	9,152	1,430	11,440	90	16	32	576	2	2	0	0	4	200
XC6SLX16	14,579	2,278	18,224	136	32	32	576	2	2	0	0	4	232

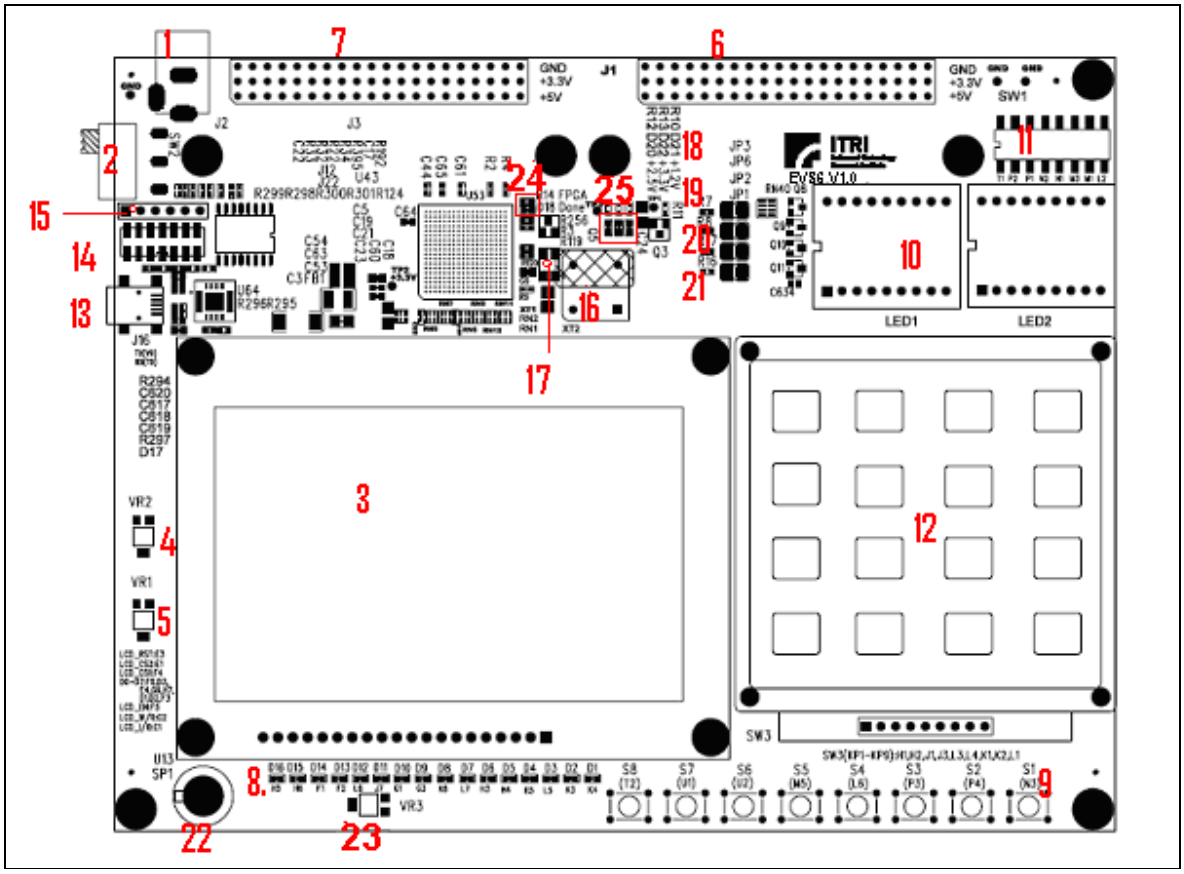
- RS-232 界面
- PROM SPI_FLASH
- 40MHz 的振盪晶體.
- 一個使用者振盪晶體的 Socket (Half Size)
- 108 點 User IO,可連接使用者設計的板子.
- 1 個 8 位 DIP switch
- 16 個 LED 輸出界面
- 8 個 按鍵輸入
- 128 x 64 點矩陣液晶顯示屏
- 2 個米字型七節燈管
- 4x4 按鍵輸入
- 單一電源輸入(+5V/ 3A)
  內部有電源穩壓器,.提供板子上電源 (+1.2 V ,+2.5V,+ 3.3 V)
- JTAG 界面,提供 FPGA 程式下載
- SPI Flash 燒錄介面
- 聲音輸出介面.

1-2. 系統方塊圖

如圖所示, EVS6 擁有豐富的週邊元件,並預留使用者可以自行擴充輸入輸出埠 (User IO,透過 J1,J2),形成一個多用途的應用平台.



1-3. EVS6 週邊元件的規格



編號	週邊名稱	功能描述
1	J2 電源輸入	DC+5V/3A (中間爲正).
2	電源開關 SW2	往下開啓電源
3	LCD 液晶顯示器	爲點矩陣型的液晶顯示器.
4	LCD 字體亮度調整鈕(VR2)	VR2 逆時鐘旋轉,字體亮度加大
5	LCD 背光調整鈕(VR1)	VR1 順時鐘旋轉,背光亮度加大
6	J1:使用者擴充用 I/O	J1 可以留給使用者擴充 I/O 用
7	J3:使用者擴充用 I/O	J3 可以留給使用者擴充 I/O 用
8	LED16	16 個 LED 輸出
9	使用者按鍵輸入	S1 ~ S8
10	米字型七節燈管	可輸出數字與字型
11	Dip switch	DIP switch
12	4x4 鍵盤介面	可以輸入 1,2,3,4,5,6,7,8,9,0,A,B,C,D,E,F 共 16 個字
13	RS232 串列介面	USB 轉 UART 介面
14	JTAG 介面	FPGA 程式下載接頭
15	SPI_FLASH 介面	SPI Flash 燒錄接頭

編號	週邊名稱	功能描述
16	User Clock 輸入	提供使用者輸入 clock 的插槽
17	System Clock 輸入	Clock 輸入爲 40MHz 的振盪晶體
18	JP3 Jumper (CLOSED)	HSWAP_EN 設爲 High(OPEN)或 LOW(CLOSED)
19	JP6 Jumper (CLOSED)	FPGA_MODE1 設爲 Master(CLOED)
20	JP2 Jumper (OPEN)	FPGA_SUSPEND 設爲 High(OPEN)或 LOW(CLOSED)
21	JP1 Jumper (OPEN)	FPGA_PROG_B,當 CLOSED 時,會使 FPGA 重新下載
22	喇叭輸出音量調整鈕	聲音輸出調整 VR3
23	Audio 喇叭	聲音輸出.(位於板子後面).
24	D18 : FPGA_DONE	FPGA_DONE(D18)亮起,代表 FPGA 下載成功.
25	D20 : +2.5V 電源燈號 D22 : +3.3V 電源燈號 D21 : +1.2V 電源燈號	D20 電源燈號亮起,代表+2.5V 電源正常 D22 電源燈號亮起,代表+3.3V 電源正常 D21 電源燈號亮起,代表+1.2V 電源正常

第二章 ： EVS6 模擬板

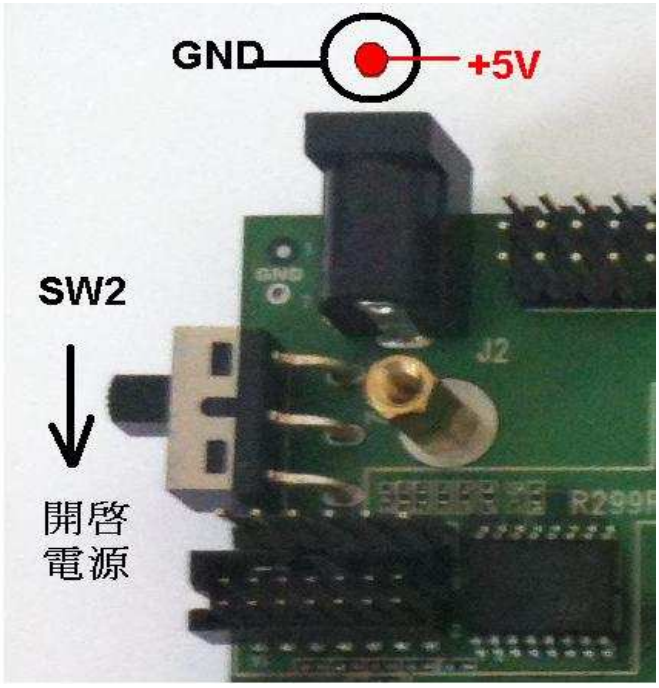
EVS6 板,包含電源,FPGA,LED…等週邊介面,再利用 J1,J3 連接使用者 I/O 板,本章節主要是敘述 EVS6 主板相關週邊電路的相關說明.

2-1. 電源接頭 J2.

電源插頭 J2 爲外部 5 V 輸入,內正外負的接頭,提供板子的電源輸入,在 J2 的下方的 SW2 爲電源的切換開關, SW2 切到下邊時,電源開啓;SW2 切到上邊時,電源關閉.

EVS6 以內部的穩壓器,將+5V 穩壓成+3.3V,+2.5V,+1.2V 等二種電源,其中 +3.3V,+2.5V,+1.2V 則提供 EVS6 板子使用,這兩種電源則有 D20,D21,D22 綠色 LED 顯示電源正常.

板子上+2.5V 的電壓正常時 ; D20 橘色 LED 爲亮 板子上+3.3V 的電壓正常時 ; D22 橘色 LED 爲亮 板子上+1.2V 的電壓正常時 ; D21 橘色 LED 爲亮

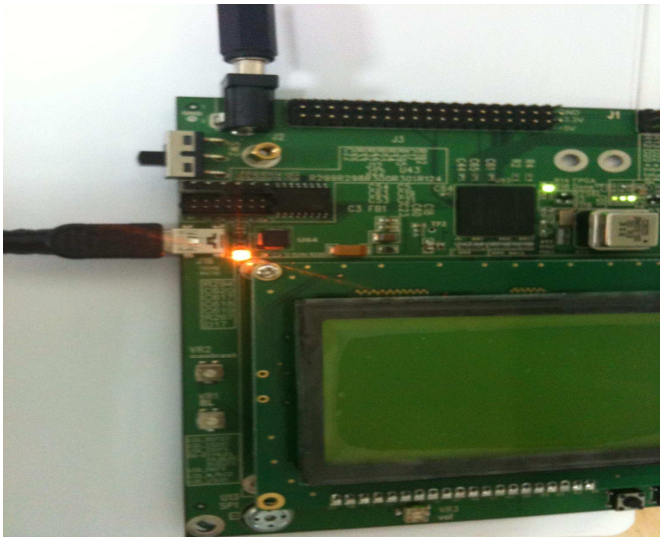


2-2. RS232 介面

RS232 介面提供 EVS6 板子與 PC 個人電腦之間串列傳輸的介面,板子上的 J16 就是 RS232 接頭.

```
NET     "UART_TX"    LOC = "V9"    ;     ## J16
NET     "UART_RX"    LOC = "T9"    ;     ## J16
```

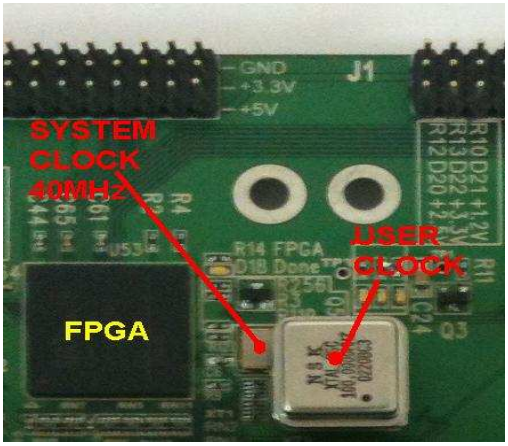
當 RS232 與電腦連接之後,D17 會亮起.使用 RS232 須先安裝 CP2102 的驅動程式



2-3. CLOCK 時脈週期

實驗板上有兩個 CLOCK 輸入,板子上的 CLK 爲 40MHz 的振盪晶體,接到 FPGA 的 R10,而板子另則提供使用者自行提供使用者所需的振盪晶體,此接腳接到 FPGA 的 T10

```
NET    "SYS_CLK"       LOC = "R10"    ;       ## SMD 40MHz
NET    "USER_CLK"      LOC = "T10"    ;       ## Socket
```

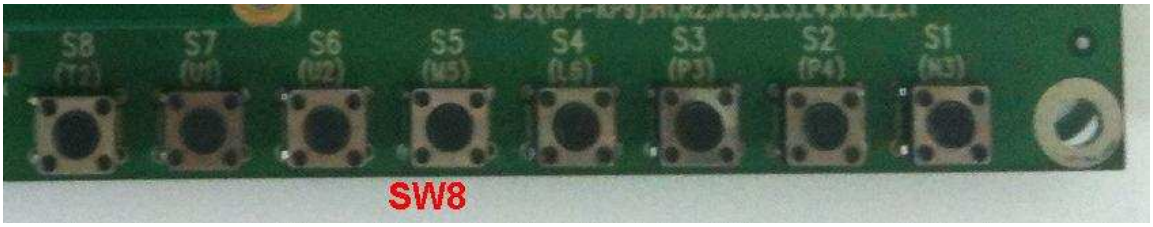


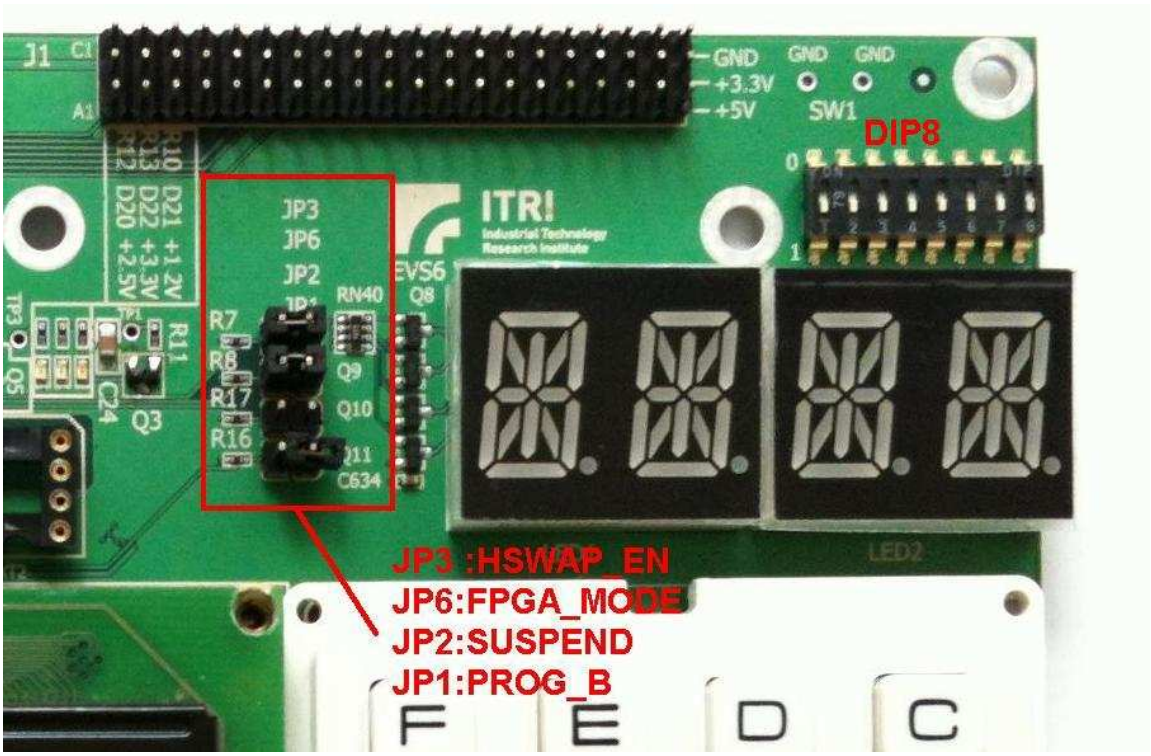
2-4. PUSH_BUTTON 與 DIP_SWITCH 輸入

實驗板上有 8 個 PUSH_BUTTON 與 8 個 DIP_SWICTH 與 FPGA_SUSPEND 設定跳線座與 1 個 FPGA_PROG_B 的最下方.

*當 PUSH_BUTTON 按下時,輸入的電壓準位爲低電位'0';當不按時爲'1'
*當 DIP_SWITCH 往上切爲[ON]時,輸入電壓準位爲低電位'0';而下切時爲[OFF].
*當 PROGRAM CLOSED,FPGA 重新執行 Configuration Download

```
NET "FPGA_SUSPEND"      LOC = "K5";              ##SUSPEN

#====SW & LED ============
Net     "Push_Button<7>"    LOC = "T2";
Net     "Push_Button<6>"    LOC = "U1";
Net     "Push_Button<5>"    LOC = "U2";
Net     "Push_Button<4>"    LOC = "M5";
Net     "Push_Button<3>"    LOC = "L6";
Net     "Push_Button<2>"    LOC = "P3";
Net     "Push_Button<1>"    LOC = "P4";
Net     "Push_Button<0>"    LOC = "N3";

Net     "DIP<7>"    LOC = "L2";
Net     "DIP<6>"    LOC = "M1";
Net     "DIP<5>"    LOC = "M3";
Net     "DIP<4>"    LOC = "N1";
Net     "DIP<3>"    LOC = "N2";
Net     "DIP<2>"    LOC = "P1";
Net     "DIP<1>"    LOC = "P2";
Net     "DIP<0>"    LOC = "T1";
```





FPGA PIN	SYMBOL	LABEL NAME	LOGIC	COMMENT
N3	S1	Push_Button<0>	按下時,輸入爲'0'	
P4	S2	Push_Button<1>	按下時,輸入爲'0'	
P3	S3	Push_Button<2>	按下時,輸入爲'0'	
L6	S4	Push_Button<3>	按下時,輸入爲'0'	
M5	S5	Push_Button<4>	按下時,輸入爲'0'	
U2	S6	Push_Button<5>	按下時,輸入爲'0'	
U1	S7	Push_Button<6>	按下時,輸入爲'0'	
T2	S8	Push_Button<7>	按下時,輸入爲'0'	
T1	SW1-1	DIP<0>	上方爲'0',下方爲'1'	
P2	SW1-2	DIP<1>	上方爲'0',下方爲'1'	
P1	SW1-3	DIP<2>	上方爲'0',下方爲'1'	
N2	SW1-4	DIP<3>	上方爲'0',下方爲'1'	
N1	SW1-5	DIP<0>	上方爲'0',下方爲'1'	
M3	SW1-6	DIP<1>	上方爲'0',下方爲'1'	
M1	SW1-7	DIP<2>	上方爲'0',下方爲'1'	
L2	SW1-8	DIP<3>	上方爲'0',下方爲'1'	

2-5. LED 輸出

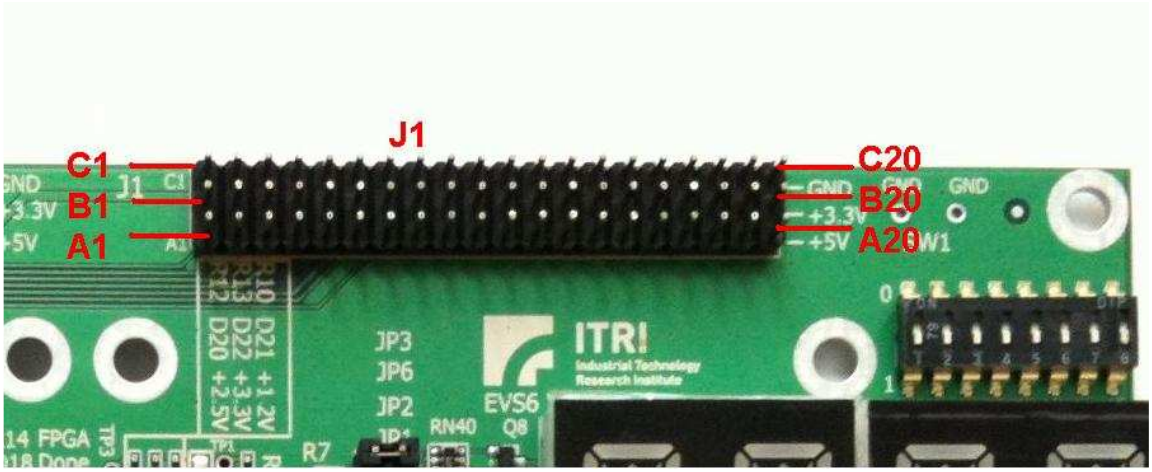
實驗板上有 16 個 LED,當輸出高電位'1'時,LED 亮,當輸出低電位'0'時,LED 不亮

```
NET "LEDS<0>"    LOC = "K4"   ;
NET "LEDS<1>"    LOC = "K3"   ;
NET "LEDS<2>"    LOC = "L5"   ;
NET "LEDS<3>"    LOC = "K5"   ;
NET "LEDS<4>"    LOC = "H4"   ;
NET "LEDS<5>"    LOC = "H3"   ;
NET "LEDS<6>"    LOC = "L7"   ;
NET "LEDS<7>"    LOC = "K6"   ;
NET "LEDS<8>"    LOC = "G3"   ;
NET "LEDS<9>"    LOC = "G1"   ;
NET "LEDS<10>"   LOC = "J7"   ;
NET "LEDS<11>"   LOC = "J6"   ;
NET "LEDS<12>"   LOC = "F2"   ;
NET "LEDS<13>"   LOC = "F1"   ;
NET "LEDS<14>"   LOC = "H6"   ;
NET "LEDS<15>"   LOC = "H5"   ;
```

FPGA PIN	SYMBOL	LABEL NAME	LOGIC	COMMENT
K4	D1	LEDS<0>	輸出'1'➔D1 亮起	
K3	D2	LEDS<1>	輸出'1'➔D2 亮起	
L5	D3	LEDS<2>	輸出'1'➔D3 亮起	
K5	D4	LEDS<3>	輸出'1'➔D4 亮起	
H4	D5	LEDS<4>	輸出'1'➔D5 亮起	
H3	D6	LEDS<5>	輸出'1'➔D6 亮起	
L7	D7	LEDS<6>	輸出'1'➔D7 亮起	
K6	D8	LEDS<7>	輸出'1'➔D8 亮起	
G3	D9	LEDS<8>	輸出'1'➔D9 亮起	
G1	D10	LEDS<9>	輸出'1'➔D10 亮起	
J7	D11	LEDS<10>	輸出'1'➔D11 亮起	
J6	D12	LEDS<11>	輸出'1'➔D12 亮起	
F2	D13	LEDS<12>	輸出'1'➔D13 亮起	
F1	D14	LEDS<13>	輸出'1'➔D14 亮起	
H6	D15	LEDS<14>	輸出'1'➔D15 亮起	
H5	D16	LEDS<15>	輸出'1'➔D16 亮起	

2-6. J1 CONNECTOR

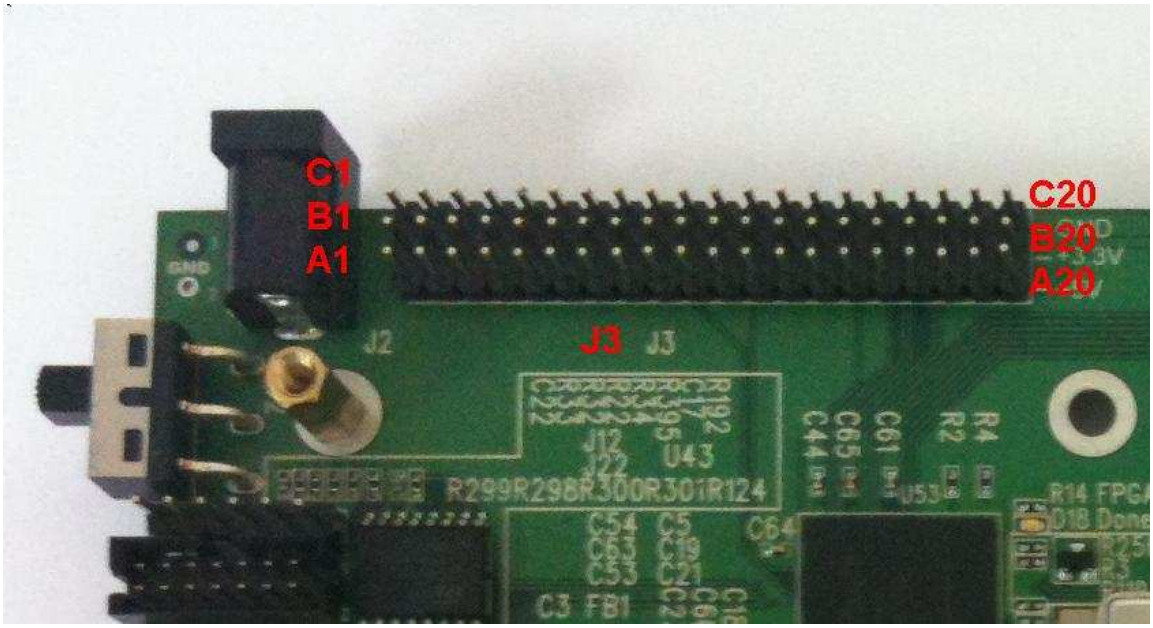
J1 提供 54 點輸出/輸入介面,提供板子擴充 IO 之用,其輸出準位爲標準的+3.3V 介面.



J1								
符號名稱	FPGA接腳	J3接腳	符號名稱	FPGA接腳	J3接腳	符號名稱	FPGA接腳	J3接腳
IO_L33N_1	E18	J1.A1	IO_L34P_1	K12	J1.B1	IO_L34N_1	K13	J1.C1
IO_L35P_1	F17	J1.A2	IO_L35N_1	F18	J1.B2	IO_L36P_1	H13	J1.C2
IO_L36N_1	H14	J1.A3	IO_L37P_1	H15	J1.B3	IO_L37N_1	H16	J1.C3
IO_L38P_1	G16	J1.A4	IO_L38N_1	G18	J1.B4	IO_L39P_1	J13	J1.C4
IO_L39N_1	K14	J1.A5	IO_L40P_GCLK11_1	L12	J1.B5	IO_L40N_GCLK10_1	L13	J1.C5
IO_L44P_1	J16	J1.A6	IO_L44N_1	J18	J1.B6	IO_L45P_1	K17	J1.C6
IO_L45N_1	K18	J1.A7	IO_L46P_1	L17	J1.B7	IO_L46N_1	L18	J1.C7
IO_L47P_1	M16	J1.A8	IO_L47N_1	M18	J1.B8	IO_L48P_1	N17	J1.C8
IO_L48N_1	N18	J1.A9	IO_L49P_1	P17	J1.B9	IO_L49N_1	P18	J1.C9
IO_L50P_1	N15	J1.A10	IO_L50N_1	N16	J1.B10	IO_L51P_1	T17	J1.C10
IO_L51N_1	T18	J1.A11	IO_L52P_1	U17	J1.B11	IO_L52N_1	U18	J1.C11
IO_L53P_1	M14	J1.A12	IO_L53N_1	N14	J1.B12	IO_L61P_1	L14	J1.C12
IO_L61N_1	M13	J1.A13	IO_L74P_1	P15	J1.B13	IO_L74N_1	P16	J1.C13
IO_L2P_CMPCLK_2	U16	J1.A14	IO_L2N_CMPMOSI_2	V16	J1.B14	IO_L12P_D1_MISO2_2	T14	J1.C14
IO_L12P_D2_MISO3_2	V14	J1.A15	IO_L13N_D10_2	P12	J1.B15	IO_L14P_D11_2	U13	J1.C15
IO_L23P_2	U11	J1.A16	IO_L14N_D12_2	V13	J1.B16	IO_L23N_2	V11	J1.C16
IO_L16P_2	R11	J1.A17	IO_L16N_VREF_2	T11	J1.B17	IO_L30P_GCLK1_D13_2	U10	J1.C17
IO_L30N_GCLK0_USERCL	V10	J1.A18	IO_L31P_GCLK1_D13_2	R8	J1.B18	IO_L31N_GCLK31_D14_2	T8	J1.C18
VOLT_5V		J1.A19	VOLT_3.3V		J1.B19	GND		J1.C19
VOLT_5V		J1.A20	VOLT_3.3V		J1.B20	GND		J1.C20

2-7. J3 CONNECTOR

J3 提供 54 點輸出/輸入介面,提供板子擴充 IO 之用,其輸出準位爲標準的+3.3V 介面.



J3								
符號名稱	FPGA接腳	J3接腳	符號名稱	FPGA接腳	J3接腳	符號名稱	FPGA接腳	J3接腳
IO_L1N_VREF_0	C4	J3.A1	IO_L2P_0	B2	J3.B1	IO_L2N_0	A2	J3.C1
IO_L3P_0	D6	J3.A2	IO_L3N_0	C6	J3.B2	IO_L4P_0	B3	J3.C2
IO_L4N_0	A3	J3.A3	IO_L5P_0	B4	J3.B3	IO_L5N_0	A4	J3.C3
IO_L6P_0	C5	J3.A4	IO_L6N_0	A5	J3.B4	IO_L10P_0	C7	J3.C4
IO_L10N_0	A7	J3.A5	IO_L8P_0	B6	J3.B5	IO_L8N_VREF_0	A6	J3.C5
IO_L11P_0	D8	J3.A6	IO_L11N_0	C8	J3.B6	IO_L33P_0	B8	J3.C6
IO_L33N_0	A8	J3.A7	IO_L34P_GCLK19_0	D9	J3.B7	IO_L34N_GCLK18_0	C9	J3.C7
IO_L35P_GCLK17_0	B9	J3.A8	IO_L35N_GCLK16_0	A9	J3.B8	IO_L36P_GCLK15_0	D11	J3.C8
IO_L36N_GCLK14_0	C11	J3.A9	IO_L37P_GCLK13_0	C10	J3.B9	IO_L37N_GCLK12_0	A10	J3.C9
IO_L38P_0	G9	J3.A10	IO_L38N_VREF_0	F9	J3.B10	IO_L39P_0	B11	J3.C10
IO_L39N_0	A11	J3.A11	IO_L41P_0	B12	J3.B11	IO_L41N_0	A12	J3.C11
IO_L62P_0	B14	J3.A12	IO_L62N_VREF_0	A14	J3.B12	IO_L63P_SCP7_0	F13	J3.C12
IO_L63N_SCP6_0	E13	J3.A13	IO_L64P_SCP5_0	C15	J3.B13	IO_L64N_SCP4_0	A15	J3.C13
IO_L65P_SCP3_0	D14	J3.A14	IO_L65N_SCP2_0	C14	J3.B14	IO_L66P_SCP1_0	B16	J3.C14
IO_L66N_SCP0_0	A16	J3.A15	IO_L1P_A25_1	F15	J3.B15	IO_L1N_VREF_1	F16	J3.C15
IO_L29P_1	C17	J3.A16	IO_L29N_1	C18	J3.B16	IO_L30P_1	F14	J3.C16
IO_L30N_1	G14	J3.A17	IO_L31P_1	D17	J3.B17	IO_L31N_1	D18	J3.C17
IO_L32P_1	H12	J3.A18	IO_L32N_1	G13	J3.B18	IO_L33P_1	E16	J3.C18
VOLT_5V		J3.A19	VOLT_3.3V		J3.B19	GND		J3.C19
VOLT_5V		J3.A20	VOLT_3.3V		J3.B20	GND		J3.C20

2-8. FPGA　Configuration

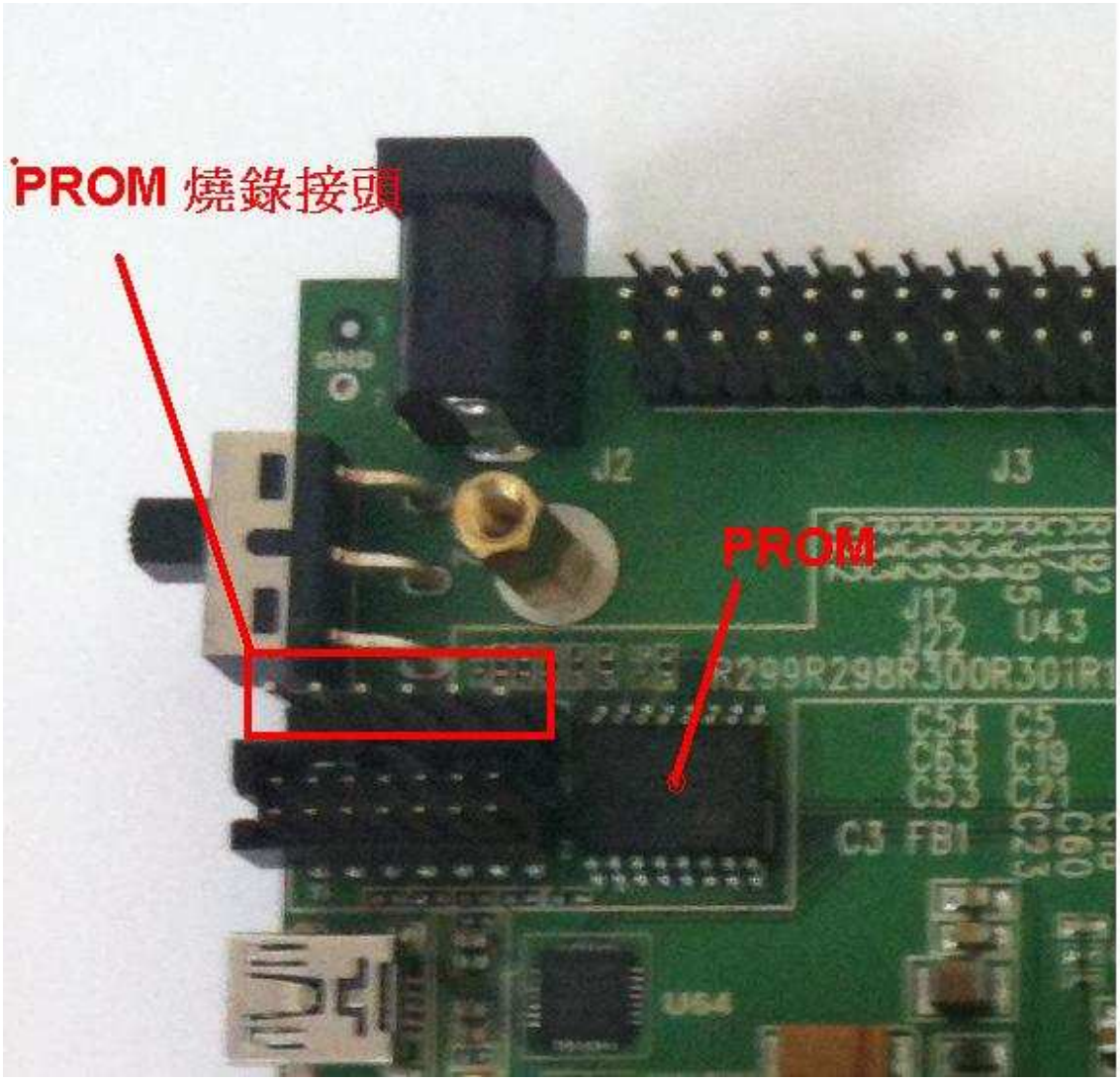
FPGA Configuration 提供使用者將自己設計的檔案輸入給 FPGA,而一般 FPGA 提供 2 種 Download 的方式,其一爲 JTAG,其一爲 PROM 兩種.

[第一種]: JTAG Programming

透過 JTAG 與 ISE 軟體的 Impact,可以將使用者程式 Download 到 FPGA,這一種方式,你需要一個 USB 下載線,連接電腦的 USB Port 與 EVS6 的 J22,就可以利用 Impact 來 Download.,如圖所示

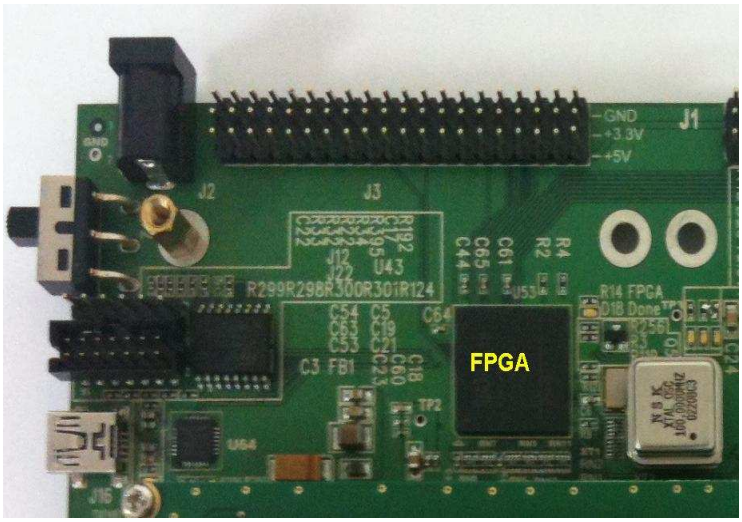
[第二種方式]: PROM

這種方式,需將板子的 JP6 以短路跳線座,將 JP6 短路,開機就可以從 FLASH_SPI 在 Download 時,需將檔案 Download 到 FPGA.



2-9. FPGA

EVS6 爲 FPGA,使用 Xilinx 的 Sparatn6 系列的 XC6SLX16 元件,其包裝爲 CSG324, 其相關接腳如表所示.



CSG324 Package—LX16

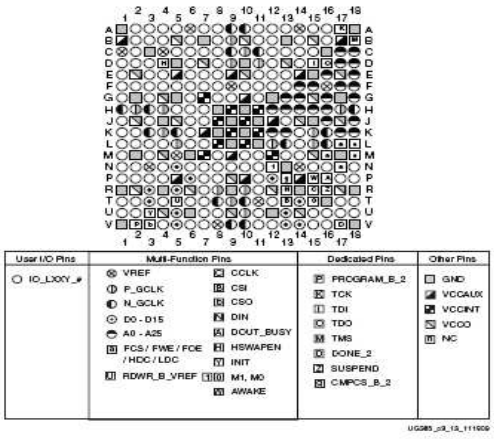


Figure 3-13: **CSG324 Package—LX16 Pinout Diagram**

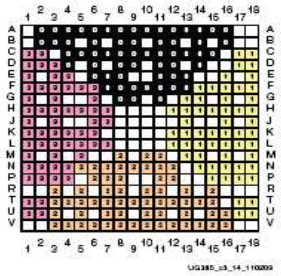


Figure 3-14: **CSG324 Package—LX16 I/O Bank Diagram**

Signal	FPGA_PIN	Peripherial	Signal	FPGA_PIN	Peripherial
GND	U53.A1		VCC_AUX	U53.B1	
IO_L2N_0	U53.A2	J3.C1	IO_L2P_0	U53.B2	J3.B1
IO_L4N_0	U53.A3	J3.A3	IO_L4P_0	U53.B3	J3.C2
IO_L5N_0	U53.A4	J3.C3	IO_L5P_0	U53.B4	J3.B3
IO_L6N_0	U53.A5	J3.B4	VOLT_3.3V	U53.B5	
IO_L8N_VREF_0	U53.A6	J3.C5	IO_L8P_0	U53.B6	J3.B5
IO_L10N_0	U53.A7	J3.A5	GND	U53.B7	
IO_L33N_0	U53.A8	J3.A7	IO_L33P_0	U53.B8	J3.C6
IO_L35N_GCLK16_0	U53.A9	J3.B8	IO_L35P_GCLK17_0	U53.B9	J3.A8
IO_L37N_GCLK12_0	U53.A10	J3.C9	VOLT_3.3V	U53.B10	
IO_L39N_0	U53.A11	J3.A11	IO_L39P_0	U53.B11	J3.C10
IO_L41N_0	U53.A12	J3.C11	IO_L41P_0	U53.B12	J3.B11
NC	U53.A13		GND	U53.B13	
IO_L62N_VREF_0	U53.A14	J3.B12	IO_L62P_0	U53.B14	J3.A12
IO_L64N_SCP4_0	U53.A15	J3.C13	VOLT_3.3V	U53.B15	
IO_L66N_SCP0_0	U53.A16	J3.A15	IO_L66P_SCP1_0	U53.B16	J3.C14
FPGA_TCK	U53.A17	J22.6	GND	U53.B17	
GND	U53.A18		FPGA_TMS	U53.B18	J22.4

Signal	FPGA_PIN	Peripherial	Signal	FPGA_PIN	Peripherial
LCD_I/O	U53.C1	U13.4	LCD_D5	U53.D1	U13.12
LCD_W/R	U53.C2	U13.5	LCD_D6	U53.D2	U13.13
GND	U53.C3		LCD_D1	U53.D3	U13.8
IO_L1N_VREF_0	U53.C4	J3.A1	HSWAP_EN	U53.D4	JP3.1
IO_L6P_0	U53.C5	J3.A4	GND	U53.D5	
IO_L3N_0	U53.C6	J3.B2	IO_L3P_0	U53.D6	J3.A2
IO_L10P_0	U53.C7	J3.C4	VOLT_3.3V	U52.D7	
IO_L11N_0	U53.C8	J3.B6	IO_L11P_0	U53.D8	J3.A6
IO_L34N_GCLK18_0	U53.C9	J3.C7	IO_L34P_GCLK19_0	U53.D9	J3.B7
IO_L37P_GCLK13_0	U53.C10	J3.B9	GND	U53.D10	
IO_L36N_GCLK14_0	U53.C11	J3.A9	IO_L36P_GCLK15_0	U53.D11	J3.C8
NC	U53.C12		NC	U53/D12	
GND	U53.C13		VOLT_3.3V	U53.D13	
IO_L65N_SCP2_0	U53.C14	J3.B14	IO_L65P_SCP3_0	U53.D14	J3.A14
IO_L64P_SCP5_0	U53.C15	J3.B13	FPGA_TDI	U53.D15	R299.1
GND	U53.C16		FPGA_TDO	U53.D16	J22.8

IO_L29P_1	U53.C17	J3.A16	IO_L31P_1	U53.D17	J3.B17
IO_L29N_1	U53.C18	J3.B16	IO_L31N_1	U53.D18	J3.C17

Signal	FPGA_PIN	Peripherial	Signal	FPGA_PIN	Peripherial
LCD_CS2	U53.E1	U13.16	LED13	U53.F1	
VOLT_3.3V	U53.E2		LED12	U53.F2	
LCD_RST	U53.E3	U13.17	LCD_D7	U53.F3	U13.14
LCD_D2	U53.E4	U13.19	LCD_CS1	U53.F4	U13.15
VCC_AUX	U53.E5		LCD_EN	U53.F5	U13.6
NC	U53.E6		LCD_D0	U53.F6	U13.7
NC	U53.E7		NC	U53.F7	
NC	U53.E8		NC	U53.F8	
VCC_AUX	U53.E9		IO_L38N_VREF_0	U53.F9	J3.B10
VOLT_3.3V	U53.E10		NC	U53.F10	
NC	U53.E11		NC	U53.F11	
NC	U53.E12		NC	U53.F12	
IO_L63N_SCP6_0	U53.E13	J3.A13	IO_L63P_SCP7_0	U53.F13	J3.C12
VCC_AUX	U53.E14		IO_L30P_1	U53.F14	J3.C16
GND	U53.E15		IO_L1P_A25_1	U53.F15	J3.B15
IO_L33P_1	U53.E16	J3.C18	IO_L1N_VREF_1	U53.F16	J3.C15
VOLT_3.3V	U53.E17		IO_L35P_1	U53.F17	J1.A2
IO_L33N_1	U53.E18	J1.A1	IO_L35N_1	U53.F18	J1.B2

Signal	FPGA_PIN	Peripherial	Signal	FPGA_PIN	Peripherial
LED9	U53.G1		KP1	U53.H1	
GND	U53.G2		KP2	U53.H2	
LED8	U53.G3		LED5	U53.H3	
VOLT_3.3V	U53.G4		LED4	U53.H4	
GND	U53.G5		LED15	U53.H5	
LCD_D3	U53.G6	U13.10	LED14	U53.H6	
VOLT_1.2V	U53.G7		LCD_D4	U53.H7	U13.11
NC	U53.G8		GND	U53.H8	
IO_L38P_0	U53.G9	J3.A10	VOLT_1.2V	U53.H9	
VCCAUX	U53.G10		GND	U53.H10	
NC	U53.G11		VOLT_1.2V	U53.H11	
GND	U53.G12		IO_L32P_1	U53.H12	J3.A18

IO_L32N_1	U53.G13	J3.B18	IO_L36P_1	U53.H13	J1.C2
IO_L30N_1	U53.G14	J3.A17	IO_L36N_1	U53.H14	J1.A3
VOLT_3.3V	U53.G15		IO_L37P_1	U53.H15	J1.B3
IO_L38P_1	U53.G16	J1.A4	IO_L37N_1	U53.H16	J1.C3
GND	U53.G17		AF_OUT_CLK	U53.H17	U14.6
IO_L38N_1	U53.G18	J1.B4	APPSEL	U53.H18	U14.7

Signal	FPGA_PIN	Peripherial	Signal	FPGA_PIN	Peripherial
KP3	U53.J1		KP7	U53.K1	
VOLT_3.3V	U53.J2		KP8	U53.K2	
KP4	U53.J3		LED1	U53.K3	
GND	U53.J4		LED0	U53.K4	
VOLT_3.3V	U53.J5		LED3	U53.K5	
LED11	U53.J6		LED7	U53.K6	
LED10	U53.J7		VCCAUX	U53.K7	
VOLT_1.2V	U53.J8		GND	U53.K8	
GND	U53.J9		VOLT_1.2V	U53.K9	
VOLT_1.2V	U53.J10		GND	U53.K10	
GND	U53.J11		VOLT_1.2V	U53.K11	
VCCAUX	U53.J12		IO_L34P_1	U53.K12	J1.B1
IO_L39P_1	U53.J13	J1.C4	IO_L34N_1	U53.K13	J1.C1
VOLT_3.3V	U53.J14		IO_L39N_1	U53.K14	J1.A5
GND	U53.J15		NC	U53.K15	
IO_L44P_1	U53.J16	J1.A6	AF_OUT_BCK	U53.K16	U14.1
VOLT_3.3V	U53.J17		IO_L45P_1	U53.K17	J1.C6
IO_L44N_1	U53.J18	J1.B6	IO_L45N_1	U53.K18	J1.A7

Signal	FPGA_PIN	Peripherial	Signal	FPGA_PIN	Peripherial
KP9	U53.L1		DIP_06	U53.M1	
DIP_07	U53.L2		GND	U53.M2	
KP5	U53.L3		DIP_05	U53.M3	
KP6	U53.L4		VOLT_3.3V	U53.M4	
LED2	U53.L5		PB4	U53.M5	
PB3	U53.L6		GND	U53.M6	
LED6	U53.L7		VOLT_1.2V	U53.M7	
VOLT_1.2V	U53.L8		NC	U53.M8	

GND	U53.L9		VCCAUX	U53.M9	
VOLT_1.2V	U53.L10		NC	U53.M10	
GND	U53.L11		NC	U53.M11	
IO_L40P_GCLK11_1	U53.L12	J1.B5	VOLT_1.2V	U53.M12	
IO_L40N_GCLK10_1	U53.L13	J1.C5	IO_L61N_1	U53.M13	J1.A13
IO_L61P_1	U53.L14	J1.C12	IO_L53P_1	U53.M14	J1.A12
AF_OUT_WS	U53.L15	U14.2	VOLT_3.3V	U53.M15	
AF_OUT_DATA	U53.L16	U14.3	IO_L47P_1	U53.M16	J1.A8
IO_L46P_1	U53.L17	J1.B7	GND	U53.M17	
IO_L46N_1	U53.L18	J1.C7	IO_L47N_1	U53.M18	J1.B8

Signal	FPGA_PIN	Peripherial	Signal	FPGA_PIN	Peripherial
DIP_04	U53.N1		DIP_02	U53.P1	
DIP_03	U53.N2		DIP_01	U53.P2	
PB0	U53.N3		PB2	U53.P3	
LEDA_CB	U53.N4		PB1	U53.P4	
LEDA_CG1	U53.N5		VOLT_2.5V	U53.P5	
NC	U53.N6		LEDA_CA	U53.P6	
NC	U53.N7		NC	U53.P7	
NC	U53.N8		NC	U53.P8	
NC	U53.N9		VOLT_3.3V	U53.P9	
NC	U53.N10		VOLT_2.5V	U53.P10	
NC	U53.N11		NC	U53.P11	
FPGA_MODE1	U53.N12		IO_L13N_D10_2	U53.P12	J1.B15
GND	U53.N13		VOLT_3.3V	U53.P13	R2.1
IO_L53N_1	U53.N14	J1.B12	VOLT_2.5V	U53.P14	
IO_L50P_1	U53.N15	J1.A10	IO_L74P_1	U53.P15	J1.B13
IO_L50N_1	U53.N16	J1.B10	IO_L74N_1	U53.P16	J1.C13
IO_L48P_1	U53.N17	J1.C8	IO_L49P_1	U53.P17	J1.B9
IO_L48N_1	U53.N18	J1.A9	IO_L49N_1	U53.P18	J1.C9

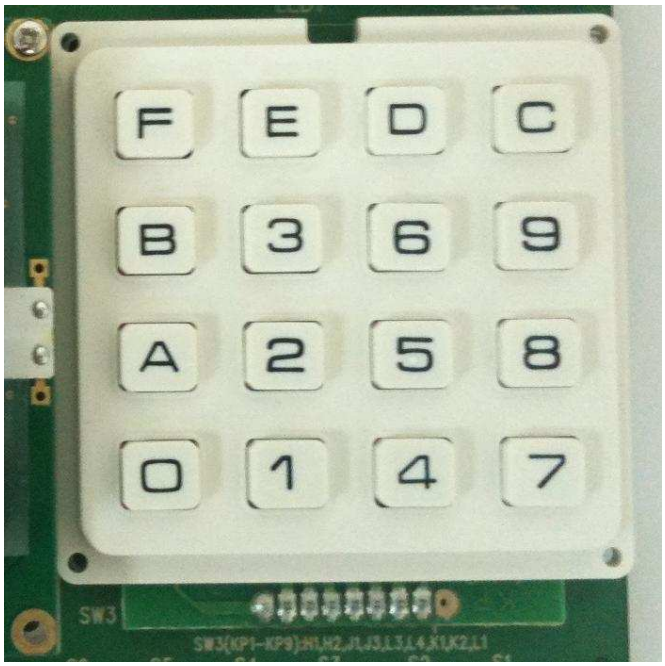
Signal	FPGA_PIN	Peripherial	Signal	FPGA_PIN	Peripherial
GND	U53.R1		DIP_00	U53.T1	
VOLT_3.3V	U53.R2		PB7	U53.T2	

LCDA_CF	U53.R3		LEDA_CH	U53.T3	
GND	U53.R4		LEDA_CJ	U53.T4	
LEDA_CG2	U53.R5		LEDA_CDE	U53.T5	
VOLT_3.3V	U53.R6		LEDA_B1	U53.T6	
LCDA_CM	U53.R7		LEDA_CN	U53.T7	
IO_L31P_GCLK1_D13_2	U53.R8	J1.B18	IO_L31N_GCLK31_D14_2	U53.T8	J1.C18
GND	U53.R9		UART_RX	U53.T9	
FPGA_CLK0	U53.R10	40MHZ	FPGA_CLK1	U53.T10	USERCLK
IO_L16P_2	U53.R11	J1.A17	IO_L16N_VREF_2	U53.T11	J1.B17
VOLT_3.3V	U53.R12		NC	U53.T12	
FPGA_DIN	U53.R13		FPGA_MOSI	U53.T13	
GND	U53.R14		IO_L12P_D1_MISO2_2	U53.T14	J1.C14
FPGA_CCLK	U53.R15		VOLT_3.3V	U53.T15	PULL UP
FPGA_SUSPEND	U53.R16	JP2.1	GND	U53.T16	
VOLT_3.3V	U53.R17		IO_L51P_1	U53.T17	J1.C10
GND	U53.R18		IO_L51N_1	U53.T18	J1.A11

Signal	FPGA_PIN	Peripherial	Signal	FPGA_PIN	Peripherial
PB6	U53.U1		GND	U53.V1	
PB5	U53.U2		FPGA_PROG_B	U53.V2	JP1.1
INIT_B	U53.U3	PULL_UP	FPGA_SPI_CS	U53.V3	
VOLT_3.3V	U53.U4		LEDA_CK	U53.V4	
LEDA_DP	U53.U5		LEDA_CC	U53.V5	
GND	U53.U6		LEDA_B2	U53.V6	
LEDA_CE	U53.U7		LEDA_CL	U53.V7	
LEDA_B3	U53.U8		LEDA_B4	U53.V8	
VOLT_3.3V	U53.U9		UART_TX	U53.V9	
IO_L30P_GCLK1_D13_2	U53.U10	J1.C17	IO_L30N_GCLK0_USERCLK_2	U53.V10	J1.A18
IO_L23P_2	U53.U11	J1.A16	IO_L23N_2	U53.V11	J1.C16
GND	U53.U12		NC	U53.V12	
IO_L14P_D11_2	U53.U13	J1.C15	IO_L14N_D12_2	U53.V13	J1.B16
VOLT_3.3V	U53.U14		IO_L12P_D2_MISO3_2	U53.V14	J1.A15
NC	U53.U15		NC	U53.V15	
IO_L2P_CMPCLK_2	U53.U16	J1.A14	IO_L2N_CMPMOSI_2	U53.V16	J1.B14
IO_L52P_1	U53.U17	J1.B11	FPGA_DONE	U53.V17	
IO_L52N_1	U53.U18	J1.C11	GND	U53.V18	

2-10. KEY4X4

EVS6 有一個 4X4 的鍵盤,共可定義 16 個按鍵,4X4 按鍵可以以掃描方式控制.

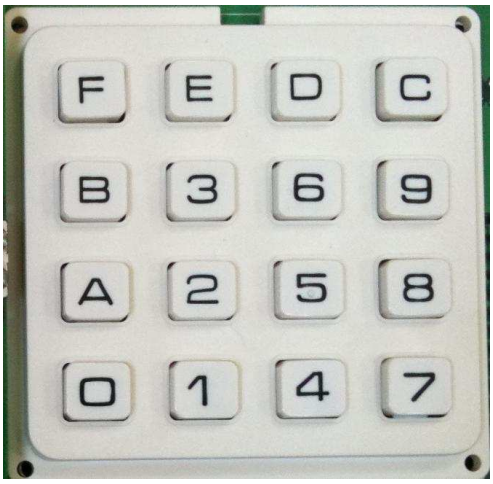



```
NET "KEY4X4_I<0>"        LOC = "L3"    ;
NET "KEY4X4_I<1>"        LOC = "L4"    ;
NET "KEY4X4_I<2>"        LOC = "K1"    ;
NET "KEY4X4_I<3>"        LOC = "K2"    ;
NET "KEY4X4_O<0>"        LOC = "H1"    ;
NET "KEY4X4_O<1>"        LOC = "H2"    ;
NET "KEY4X4_O<2>"        LOC = "J1"    ;
NET "KEY4X4_O<3>"        LOC = "J3"    ;
```

FPGA PIN	SYMBOL	LABEL NAME	LOGIC	COMMENT
L3	KEY4X4_C1	KEY4X4_C1	列 1	
L4	KEY4X4_C2	KEY4X4_C2	列 2	
K1	KEY4X4_C3	KEY4X4_C3	列 3	
K2	KEY4X4_C4	KEY4X4_C4	列 4	
H1	KEY4X4_R1	KEY4X4_R1	行 1	
H2	KEY4X4_R2	KEY4X4_R2	行 2	
J1	KEY4X4_R3	KEY4X4_R3	行 3	
J3	KEY4X4_R4	KEY4X4_R4	行 4	

EVS6 有一個 4X4 的鍵盤,是以掃描方式控制來掃描按鍵輸入是第幾行與第幾列,使用者得到掃描的第幾行與第幾列的值,再配合 4X4 的鍵盤印刷的字定義按下的鍵盤的意義,一般 4X4 的鍵盤有兩種,如下圖所示.當你拿到板子時,確認爲那一種之後,再自行定義其鍵盤的意義.

[圖左: 第一種 4x4 鍵盤]



[圖右: 第二種 4x4 鍵盤]

VHDL 程式示範如下.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity key4x4_scan is
      Port ( reset : in    STD_LOGIC;
                clk : in    STD_LOGIC;
                col : in    STD_LOGIC_VECTOR (3 downto 0);
                row : out    STD_LOGIC_VECTOR (3 downto 0);
                keyout : out    STD_LOGIC_VECTOR (15 downto 0));
end key4x4_scan;
architecture Behavioral of key4x4_scan is
signal clk_cnt : std_logic_vector(23 downto 0);
signal clk_row : std_logic;
signal clk_col : std_logic;
signal row_i      : std_logic_vector(3 downto 0);
begin
--============================
--    clock divide
--============================
process(reset,clk)
begin
      if reset = '1' then
```

```
            clk_cnt <= (others => '0');
        elsif rising_edge(clk) then
            clk_cnt <= clk_cnt + '1';
        end if;
end process;
clk_row <= clk_cnt(4);
clk_col <= '1' when clk_cnt(7 downto 0) = x"FF" else '0';
--==============================
--    key output
--==============================
row <= row_i;

with clk_cnt(1 downto 0) SELect
row_i <=    "1110" when "00",    --0
            "1101" when "01",    --1
            "1011" when "10",    --2
            "0111" when "11",    --3
            "1111" when others;      --???

with row_i SELect
keyout <= "111111111111" & col(3 downto 0)                    when "1110",    --0
            "11111111"       & col(3 downto 0) & "1111"       when "1101",    --1
            "1111"           & col(3 downto 0) & "11111111"   when "1011",    --2
              col(3 downto 0)& "111111111111"                  when "0111",    --3
              "1111111111111111" when others;      --???
end Behavioral;
```

Keyout 的數值	圖左:第一種 4x4 鍵盤	圖右:第二種 4x4 鍵盤
1111,1111,1111,1110	F	0
1111,1111,1111,1101	B	4
1111,1111,1111,1011	A	8
1111,1111,1111,0111	0	c
1111,1111,1110,1111	E	1
1111,1111,1101,1111	3	5
1111,1111,1011,1111	2	9
1111,1111,0111,1111	1	D
1111,1110,1111,1111	F	2
1111,1101,1111,1111	6	6
1111,1011,1111,1111	5	A
1111,0111,1111,1111	4	B
1110,1111,1111,1111	C	3
1101,1111,1111,1111	9	7
1011,1111,1111,1111	8	B
0111,1111,1111,1111		F

VERILOG 程式示範如下.

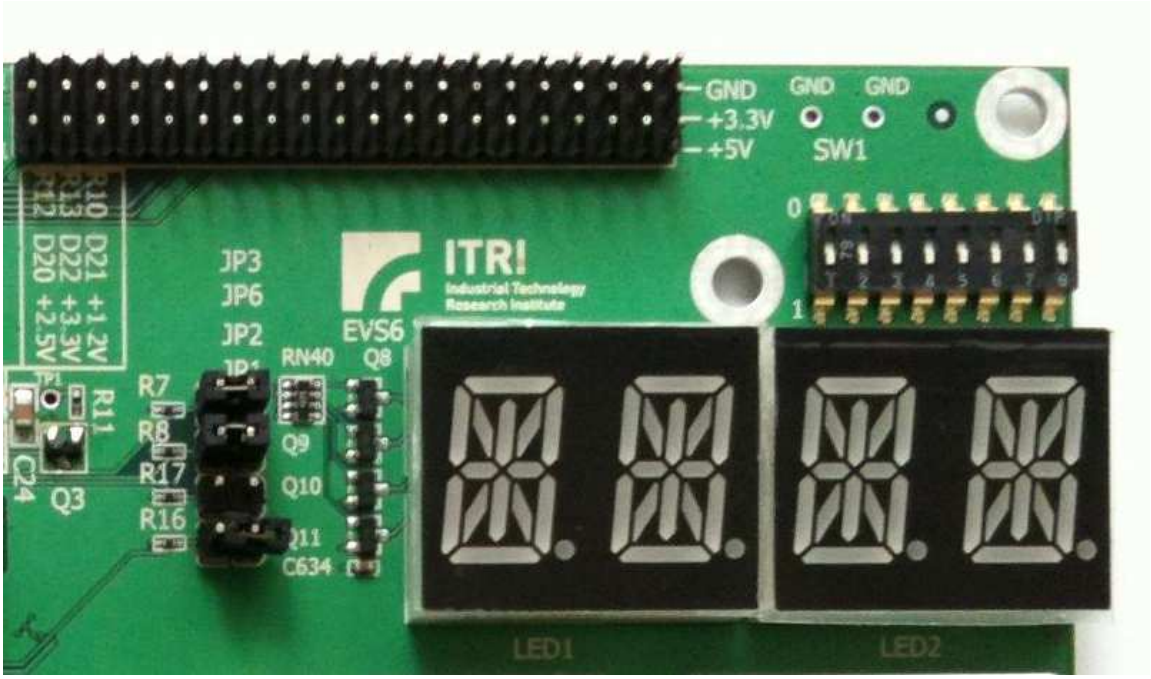
```
`ifndef KEY_SPEED
        `define KEY_SPEED 25000
`endif
`ifndef KEY_SPEED_DW
        `define KEY_SPEED_DW 25
`endif

module key4x4(clk, rst, col, row, keyout );
        input clk;
        input rst;
        input [3:0] col;
        output [3:0] row;
        output [15:0] keyout;
//      parameter speed = `KEY_SPEED;
        reg [24:0] cnt, cnt_next;
        reg clk_new, clk_next;
// internal
        reg [3:0] row, row_next;
        reg [15:0] keyout;

        always @(posedge clk or negedge rst) begin
                if (rst == 1'b0) begin
                        clk_new <= 1'b1;
                        cnt <= 25'd0;
                end else begin
                        clk_new <= clk_next;
                        cnt <= cnt_next;
                end
        end

// clock div
        always @(*) begin
                if (cnt != 25'd`KEY_SPEED) begin
                        cnt_next = cnt + 25'd1;
                        clk_next = clk_new;
                end else begin
                        cnt_next = 25'd0;
                        clk_next = ~clk_new;
                end
        end

        always @(posedge clk or negedge rst) begin
                if (rst == 1'b0) begin
                        row <= 4'b1110;
```

```
                end else begin
                        row <= {row[2:0],row[3]};
                end
        end

        always @(*) begin
                keyout = 16'b1111_1111_1111_1111;
                // synopsys parallel case
                case(row)
                4'b1110:
                        keyout[3:0] = col;
                4'b1101:
                        keyout[7:4] = col;
                4'b1011:
                        keyout[11:8] = col;
                4'b0111:
                        keyout[15:12] = col;
                default:
                        keyout = 16'b1111_1111_1111_1111;
                endcase
        end
endmodule
```

Keyout 的數值	圖左:第一種 4x4 鍵盤	圖右:第二種 4x4 鍵盤
1111,1111,1111,1110	F	0
1111,1111,1111,1101	B	4
1111,1111,1111,1011	A	8
1111,1111,1111,0111	0	c
1111,1111,1110,1111	E	1
1111,1111,1101,1111	3	5
1111,1111,1011,1111	2	9
1111,1111,0111,1111	1	D
1111,1110,1111,1111	F	2
1111,1101,1111,1111	6	6
1111,1011,1111,1111	5	A
1111,0111,1111,1111	4	B
1110,1111,1111,1111	C	3
1101,1111,1111,1111	9	7
1011,1111,1111,1111	8	B
0111,1111,1111,1111		F

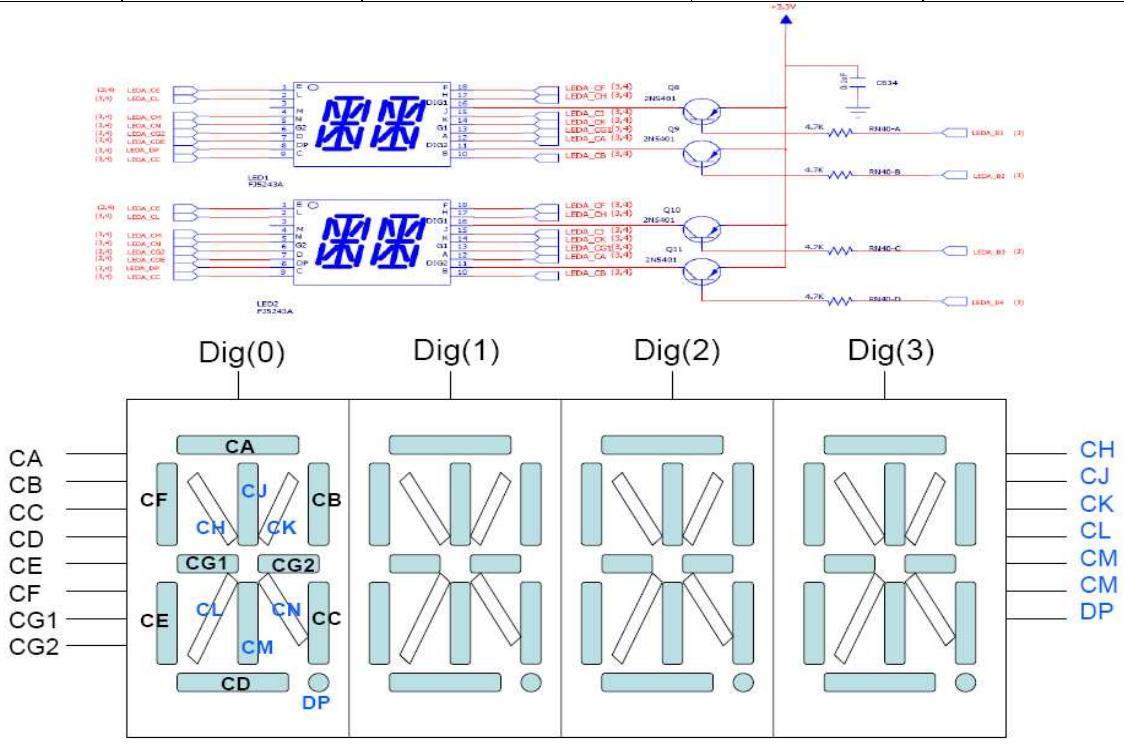
2-11. 米字型七節燈管

EVS6 提供 2 個米字型七節燈管,由 FPGA 控制.



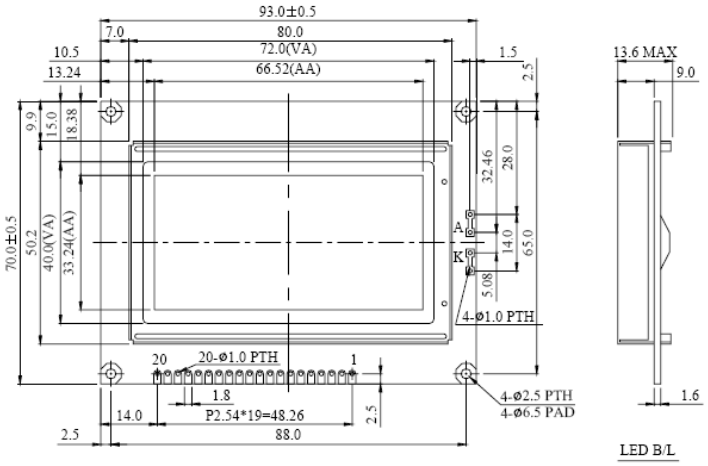

```
NET "SEGMENT<0>" LOC = "P6"; ##LEDA_CA
NET "SEGMENT<1>" LOC = "N4"; ##LEDA_CB
NET "SEGMENT<2>" LOC = "V5"; ##LEDA_CC
NET "SEGMENT<3>" LOC = "T5"; ##LEDA_CD
NET "SEGMENT<4>" LOC = "U7"; ##LEDA_CE
NET "SEGMENT<5>" LOC = "R3"; ##LEDA_CF
NET "SEGMENT<6>" LOC = "N5"; ##LEDA_G1
NET "SEGMENT<7>" LOC = "R5"; ##LEDA_G2

NET "SEGMENT<8>" LOC = "T3"; ##LEDA_CH
NET "SEGMENT<9>" LOC = "T4"; ##LEDA_CJ
NET "SEGMENT<10>" LOC = "V4"; ##LEDA_CK
NET "SEGMENT<11>" LOC = "V7"; ##LEDA_CL
NET "SEGMENT<12>" LOC = "R7"; ##LEDA_CM
NET "SEGMENT<13>" LOC = "T7"; ##LEDA_CN
NET "SEGMENT<14>" LOC = "U5"; ##LEDA_DP

NET "DIG<0>" LOC = "T6"; ##DIG1
NET "DIG<1>" LOC = "V6"; ##DIG2
NET "DIG<2>" LOC = "U8"; ##DIG3
NET "DIG<3>" LOC = "V8"; ##DIG4
```

FPGA PIN	SYMBOL	LABEL NAME	LOGIC	COMMENT
T6	LEDA_B1	DIGIT<0>	輸出'0',動作	LED1_FJ5243A
V6	LEDA_B2	DIGIT<1>	輸出'0',動作	LED1_FJ5243A
U8	LEDA_B3	DIGIT<2>	輸出'0',動作	LED2_FJ5243A
V8	LEDA_B4	DIGIT<3>	輸出'0',動作	LED2_FJ5243A
P6	LEDA_CA	SEGEMENT<0>	輸出'1',動作	CA
N4	LEDA_CB	SEGEMENT<1>	輸出'1',動作	CB
V5	LEDA_CC	SEGEMENT<2>	輸出'1',動作	CC
T5	LEDA_CD	SEGEMENT<3>	輸出'1',動作	CD
U7	LEDA_CE	SEGEMENT<4>	輸出'1',動作	CE
R3	LEDA_CF	SEGEMENT<5>	輸出'1',動作	CF
N5	LEDA_G1	SEGEMENT<6>	輸出'1',動作	CG1
R5	LEDA_G2	SEGEMENT<7>	輸出'1',動作	CG2
T3	LEDA_CH	SEGEMENT<8>	輸出'1',動作	CH
T4	LEDA_CJ	SEGEMENT<9>	輸出'1',動作	CJ
V4	LEDA_CK	SEGEMENT<10>	輸出'1',動作	CK
V7	LEDA_CL	SEGEMENT<11>	輸出'1',動作	CL
R7	LEDA_CM	SEGEMENT<12>	輸出'1',動作	CM
T7	LEDA_CN	SEGEMENT<13>	輸出'1',動作	CN
U5	LEDA_DP	SEGEMENT<14>	輸出'1',動作	DP



2-12. LCD128X64

LCD128X64 是一個每行 128 像素.每列 64 像素的液晶顯示器,可以顯示 ASCII 字型,或中文字與任意圖形介面.



PIN NO.	SYMBOL
1	Vss
2	Vdd
3	Vo
4	D/I
5	R/$\overline{W}$
6	E
7	DB0
8	DB1
9	DB2
10	DB3
11	DB4
12	DB5
13	DB6
14	DB7
15	CS1
16	CS2
17	$\overline{RES}$
18	Vout
19	A
20	K

Pin No.	Symbol	Level	Description
1	V_{SS}	0V	Ground
2	V_{DD}	3.0V	Supply voltage for logic
3	V_O	(Variable)	Operating voltage for LCD
4	D/I	H/L	H: Data , L : Instruction
5	R/W	H/L	H: Read (MPU←Module) , L: Write (MPU→Module)
6	E	H	Enable signal
7	DB0	H/L	Data bit 0
8	DB1	H/L	Data bit 1
9	DB2	H/L	Data bit 2
10	DB3	H/L	Data bit 3
11	DB4	H/L	Data bit 4
12	DB5	H/L	Data bit 5
13	DB6	H/L	Data bit 6
14	DB7	H/L	Data bit 7
15	CS1	H	Select Column 1~ Column 64
16	CS2	H	Select Column 65~ Column 128
17	RST	L	Reset signal
18	Vout	-10V	Negative Voltage
19	A	—	Power Supply for LED backlight (+)
20	K	—	Power Supply for LED backlight (-)

Display Control Instruction

The display control instructions control the internal state of the KS0108B. Instruction is received from MPU to KS0108B for the display control. The following table shows various instructions

ruction	D/I	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Function
Display ON/OFF	0	0	0	0	1	1	1	1	1	0/1	Controls the display on or off. Internal status and display RAM data are not affected. 0:OFF, 1:ON
Set Address	0	0	0	1	Y address (0~63)						Sets the Y address in the Y address counter.

Set Page (X address)	0	0	1	0	1	1	1	Page (0 ~7)			Sets the X address at the X address register.
Display Start Line	0	0	1	1	Display start line(0~63)						Indicates the display data RAM displayed at the top of the screen.
Status Read	0	1	BUSY	0	ON/OFF	RESET	0	0	0	0	Read status. BUSY 0:Ready 1:In operation ON/OFF 0:Display ON 1:Display OFF RESET 0:Normal 1:Reset
Write Display Data	1	0	Display Data								Writes data (DB0:7)into display data RAM. After writing instruction, Y address is increased by 1 automatically.
Read Display Data	1	1	Display Data								Reads data (DB0:7) from display data RAM to the data bus.

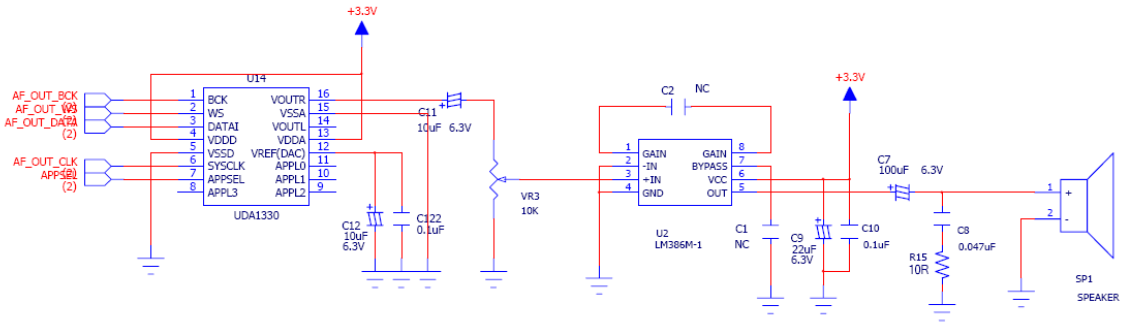
FPGA_Pin

```
NET    " LCD_D<0>"    LOC = F6;
NET    " LCD_D<1>"    LOC = D3;
NET    " LCD_D<2>"    LOC = E4;
NET    " LCD_D<3>"    LOC = G6;
NET    " LCD_D<4>"    LOC = H7;
NET    " LCD_D<5>"    LOC = D1;
NET    " LCD_D<6>"    LOC = D2;
NET    " LCD_D<7>"    LOC = F3;
NET    " LCD_RSTn "    LOC = E3;
NET    " LCD_CS1"      LOC = F4;
NET    " LCD_CS2"      LOC = E1;
NET    " LCD_EN "      LOC = F5;
NET    " LCD_WR"       LOC = C2;
NET    " LCD_IO "      LOC = C1;
```

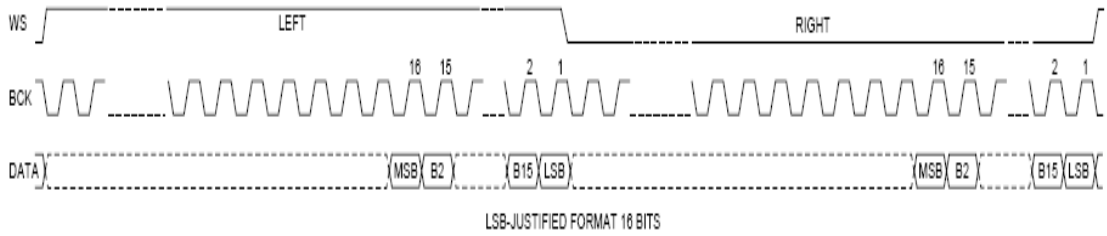
FPGA PIN	SYMBOL	LABEL NAME	LOGIC	COMMENT
F6	DB0	LCD_D<0>	LCD_Data0	
D3	DB1	LCD_D<1>	LCD_Data1	
E4	DB2	LCD_D<2>	LCD_Data2	
G6	DB3	LCD_D<3>	LCD_Data3	
H7	DB4	LCD_D<4>	LCD_Data4	
D1	DB5	LCD_D<5>	LCD_Data5	
D2	DB6	LCD_D<6>	LCD_Data6	
F3	DB7	LCD_D<7>	LCD_Data7	
F5	E	LCD_EN	LCD Enable	
C2	R/W	LCD_W/R	Read/Write	0: Write 1: Read
E3	RS	LCD_RST	Register Select	
F4	CS1	LCD_CS1	LCD Chip_Select1	
E1	CS2	LCD_CS2	LCD Chip_Select2	
C1	D/I	LCD_I/O	LCD_IO	0: Instruction 1: Data

2-13.Audio 輸出

UDA1330是一個專爲聲音輸出所設計的數位轉類比信號轉換器,配合後方的運算放大器,提供多元化的聲音輸出,其相關電路如下.



UDA1330 是一個立體聲DAC輸出電路,透過WS,BCK,DATA的串烈資料,就可以輸出立體聲,目前使用 16 位元解析度,最大值爲 32767(0x7FFF),最小值爲-32768(0x8000),可以輸出不同的高低音,其輸入波形如下圖,相關資料,可以參考 dataheet.



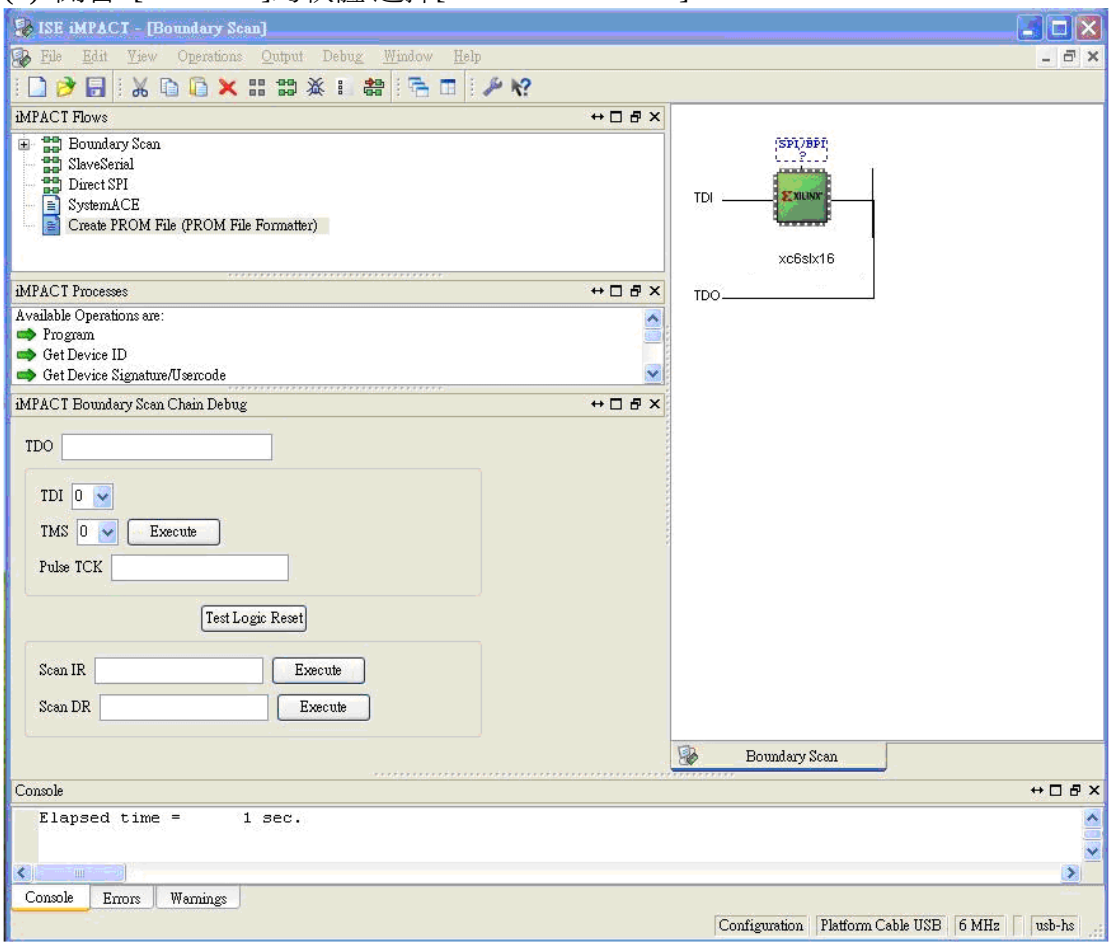

```
##### Audio#####
NET "AUDIO_BCK"        LOC = "K16";   ##BCK
NET "AUDIO_WS "        LOC = "L15";   ##WS
NET "AUDIO_DATA"       LOC = "L16";    ##DATAI
NET "AUDIO_CLK"        LOC = "H17";   ##SYSCLK
NET "AUDIO_APPSEL"     LOC = "H18";   ##APPSEL
```

FPGA PIN	SYMBOL	LABEL NAME	LOGIC	COMMENT
K16	BCK	AF_OUT_BCK	BIT CLOCK INPUT	
L15	WS	AF_OUT_WS	WORD SELECT INPUT	
L16	DATAI	AF_OUT_DATA	DATA INPUT	
H17	SYSCLK	AF_OUT_CLK	SYSTEM CLOCK	
H18	APPSEL	APPSEL	APPLICATION MODE	

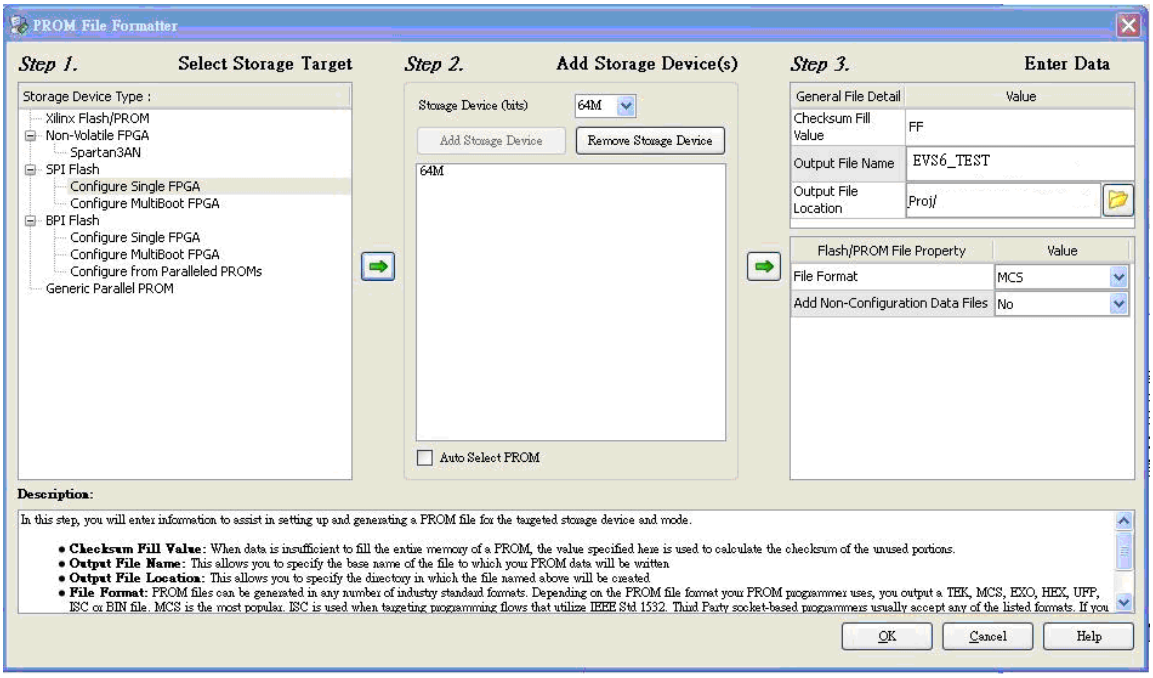
[附錄]: PROM 的燒錄與使用方法

在 ISE 發展程式完成之後,你可以將 ISE 的 BIT 檔轉成 PROM 的格式,利用 impact 轉成 PROM 的檔案,並將燒錄到 PROM,在下次開機之後,就可以由 PROM 完成 FPGA 的下載.以下是操作程序.

(1). 開啓 [IMPACT]的軟體.選擇[Create PROM File]

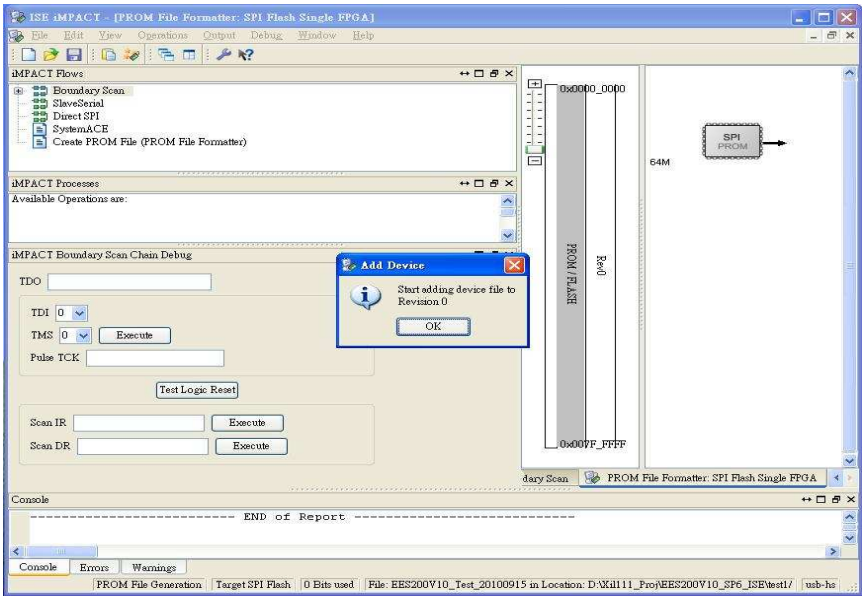


(2).在這個畫面下,

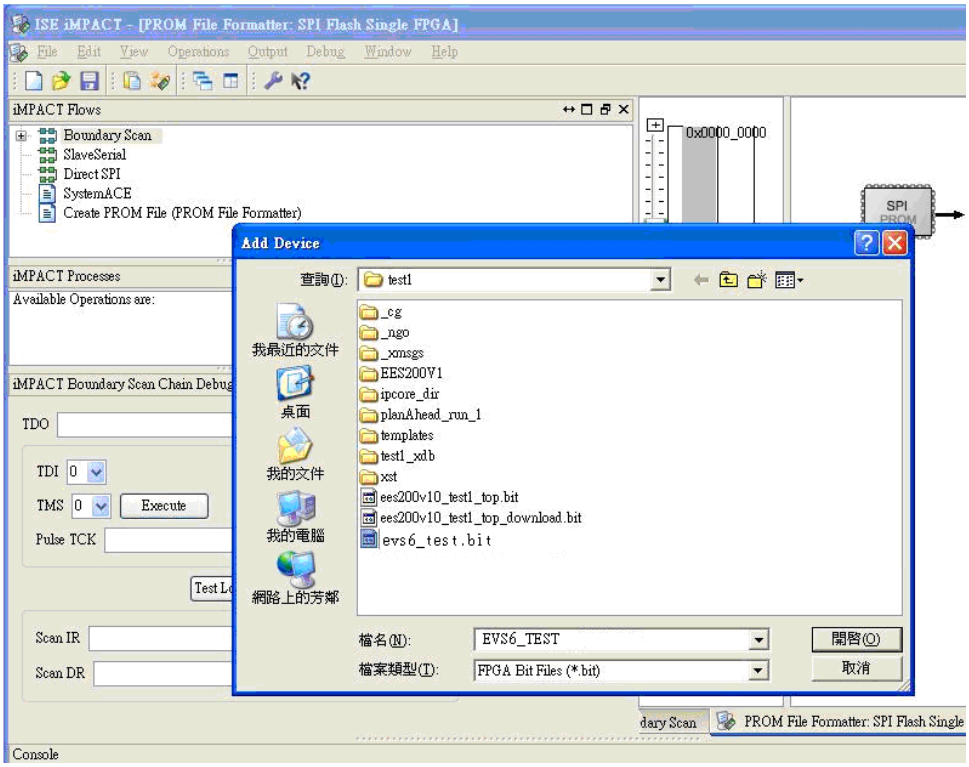


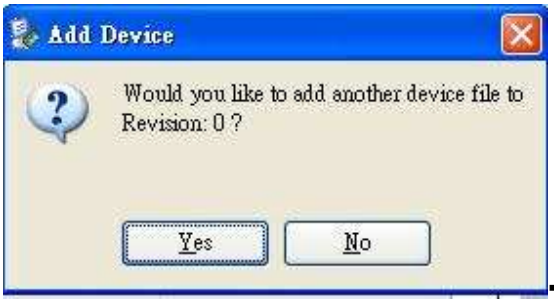
[STEP1]: SPI_FLASH ➔ Configure Single FPGA
[STEP2]: Storage=64M ➔ Add Storage Device
[STEP3]: Checksum Fill Value = FF
Output File Name : EVS6_Test (自己設定輸出檔案名稱)
Output File Location : 自選
File Format : MCS
Add Non-Configuration Data File : No

➔ 完成後,按[OK]



(3).加入使用者的 File.

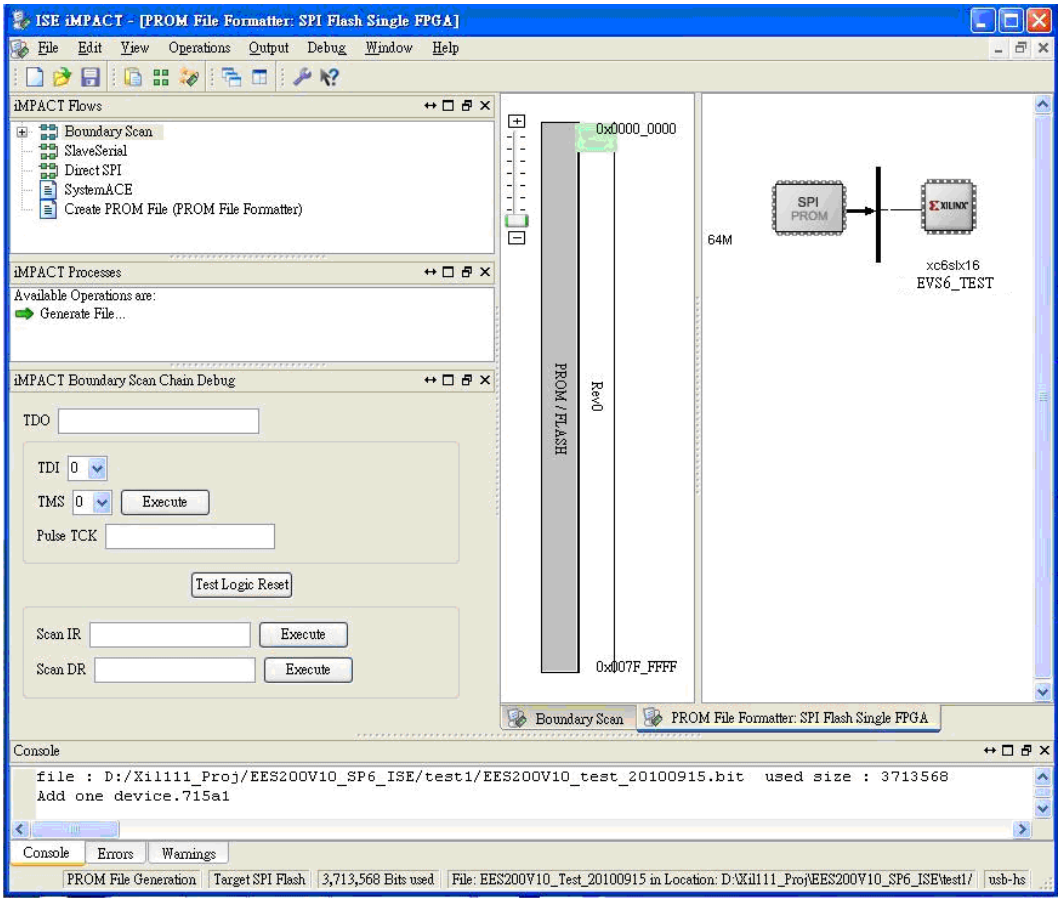




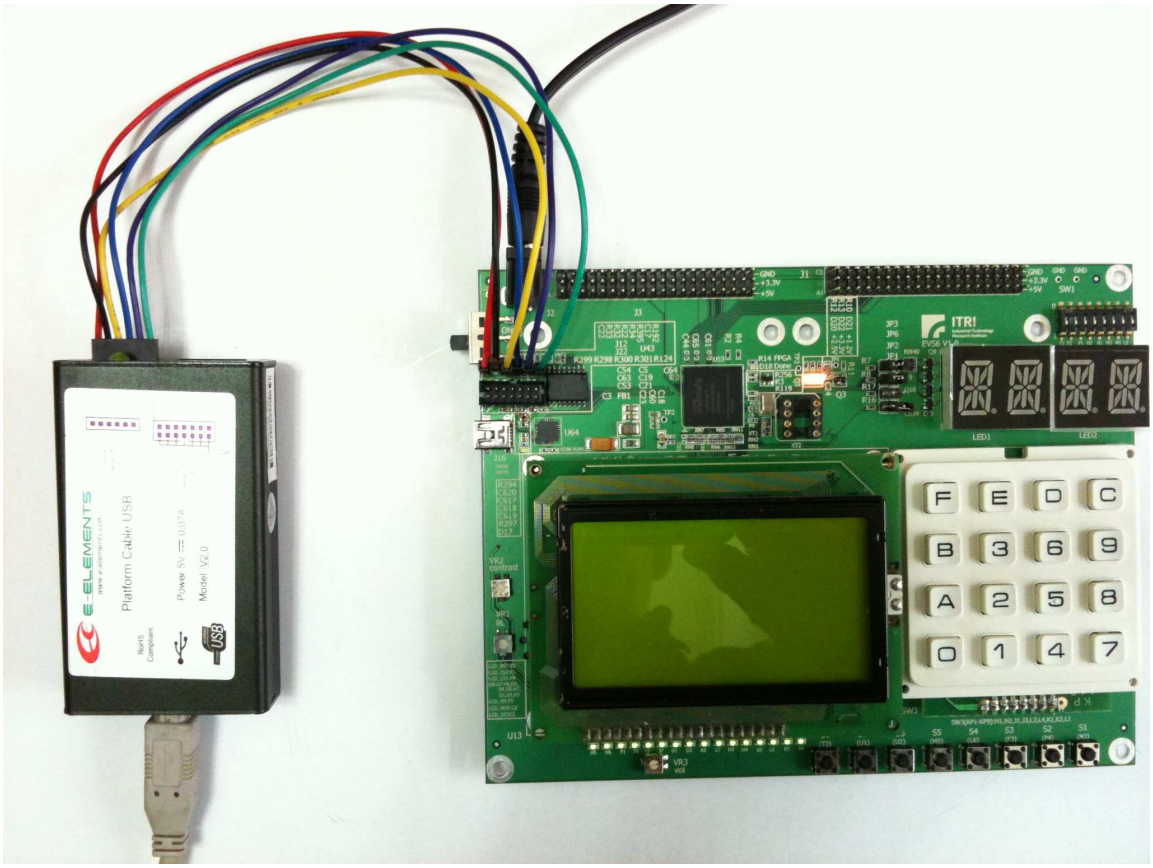
➔ 按[No]

➔ 按[OK]

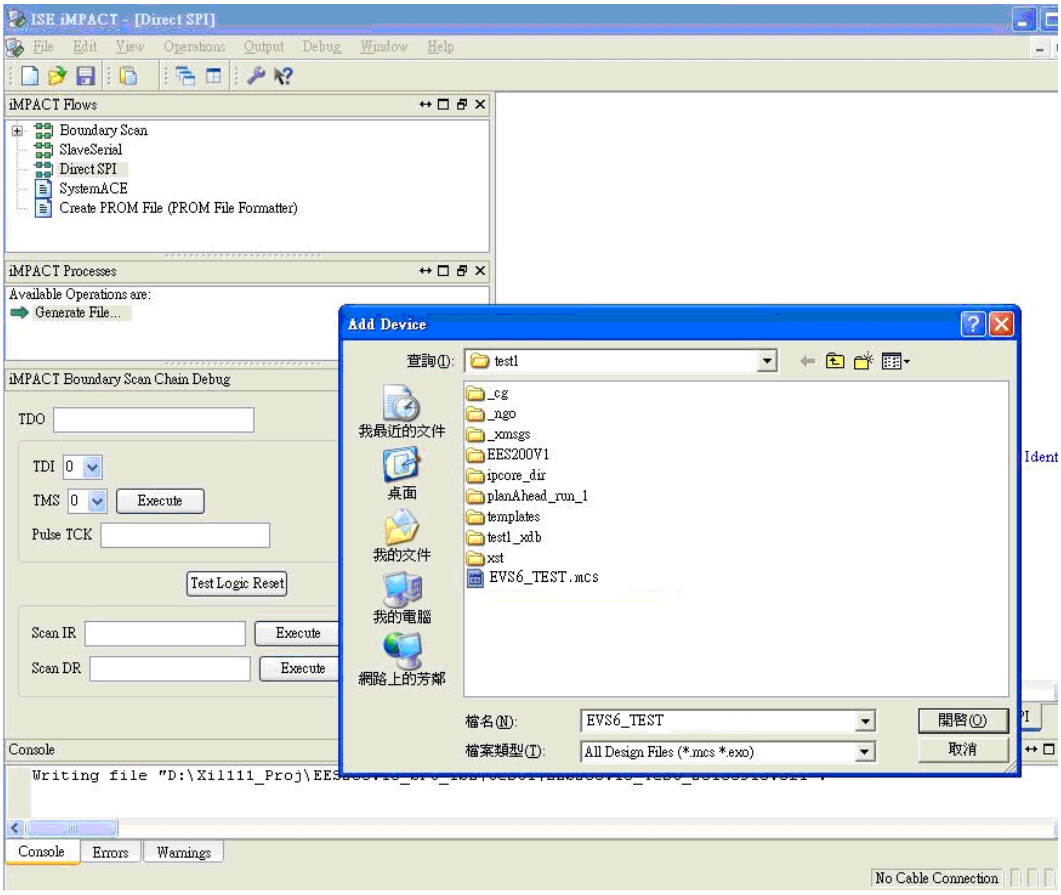
(4). 按[ok]後,出現以下畫面,按[Generate File] ➔ 出現[Generate Success]



(5).連接 JTAG 到 板子,如圖.

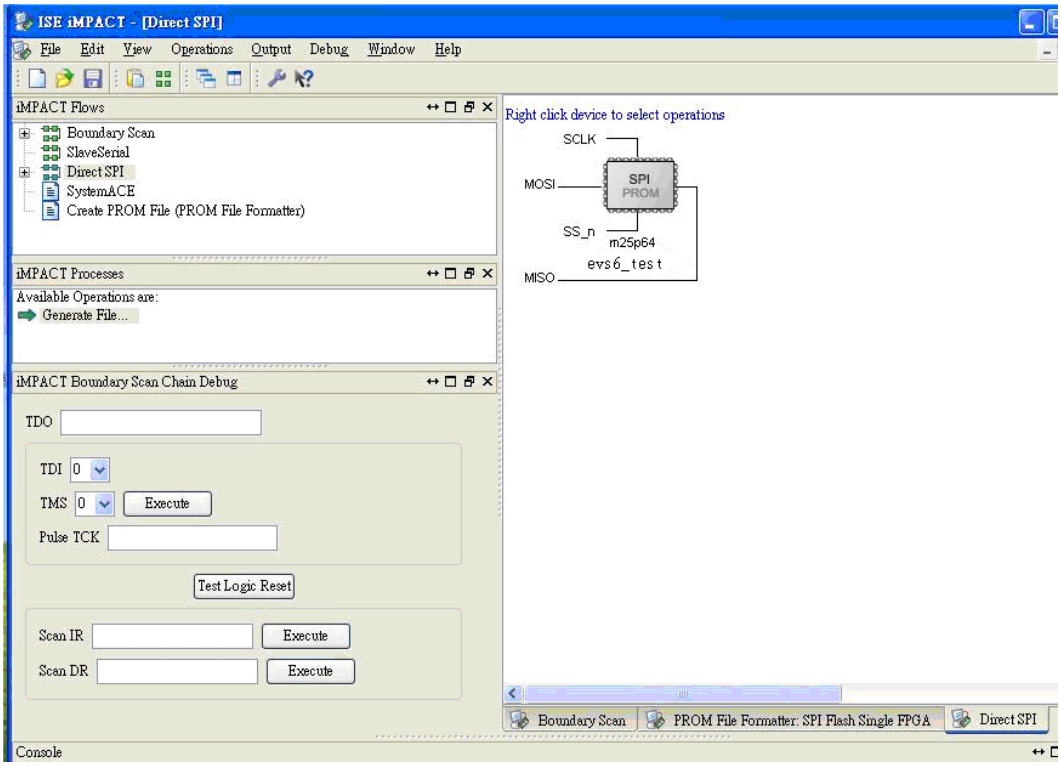


(6).按[IMPACT]軟體中的[Direct SPI] ,在右邊空白的視窗中,按滑鼠右鍵,選擇[Add Device],加入 EVS6_TEST.mcs

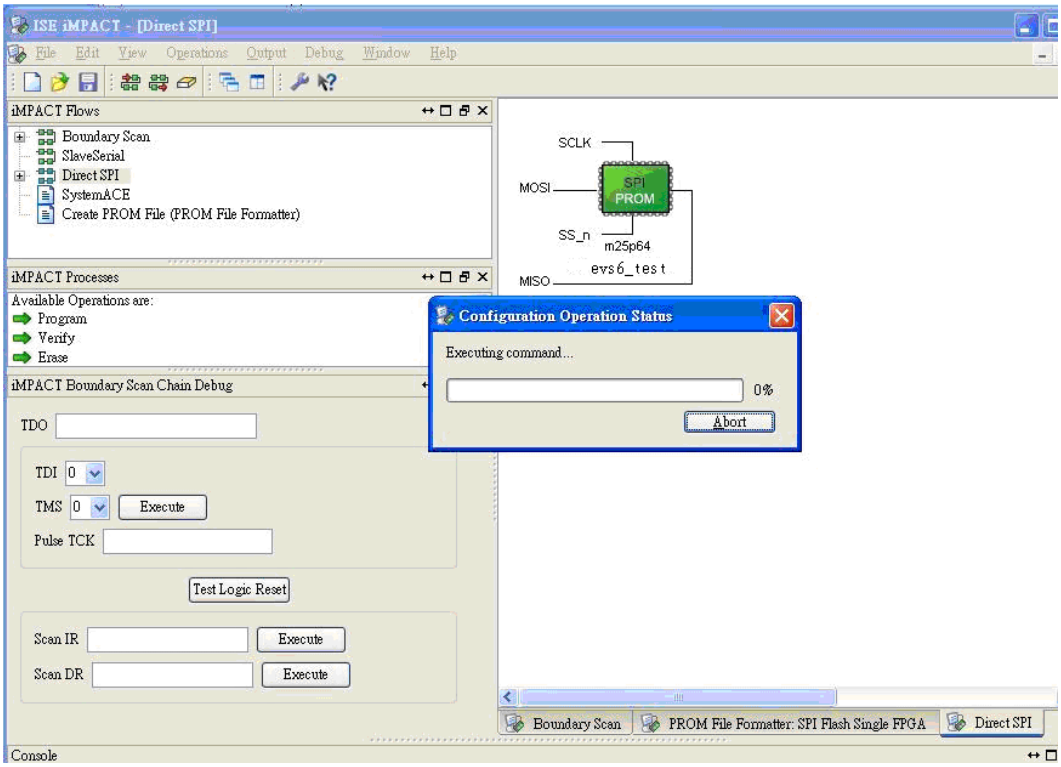


(7).初現視窗,選擇元件[M25P64]

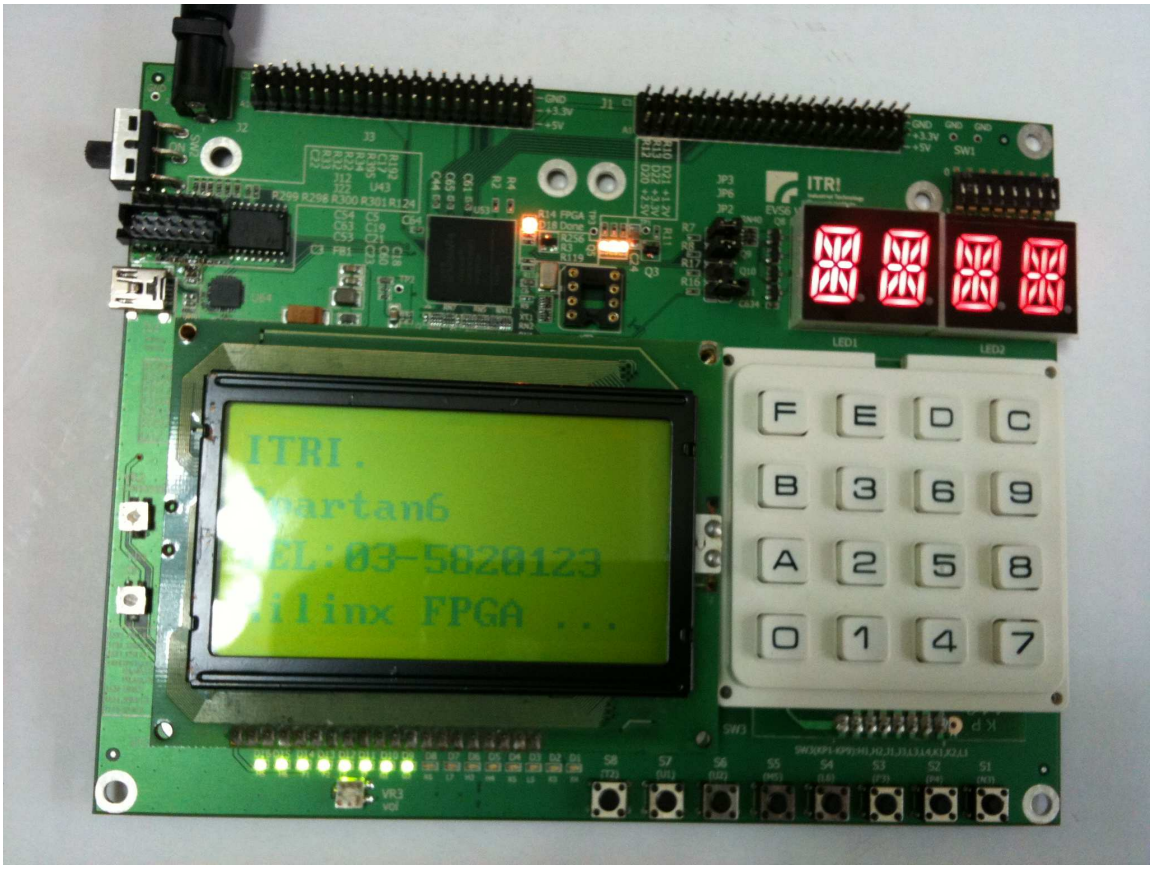
(8).出現如圖.


ISE iMPACT - [Direct SPI]
File Edit View Operations Output Debug Window Help
iMPACT Flows
Boundary Scan
SlaveSerial
Direct SPI
SystemACE
Create PROM File (PROM File Formatter)
iMPACT Processes
Available Operations are:
Generate File...
iMPACT Boundary Scan Chain Debug
TDO
TDI 0
TMS 0 Execute
Pulse TCK
Test Logic Reset
Scan IR Execute
Scan DR Execute
Right click device to select operations
SCLK
MOSI
SPI PROM
SS_n
m25p64
evs6_test
MISO
Boundary Scan
PROM File Formatter: SPI Flash Single FPGA
Direct SPI
Console

(9). 按滑鼠右鍵,選[PROGRAM],開始燒錄.



(10).設定 JP6 爲短路,重新開機即可.


ITRI.
EL:03-5820123
F E D C
B 3 6 9
A 2 5 8
0 1 4 7