組合語言與微處理機

期末專題

**Puzzle Square**



組別: 第三組

組員: 林鑫彤 黃翊 許睿玲

學號: B013040024 B013040049 B013040051

指導老師: 張雲南 老師

1. **專題簡介：**

這次組合語言與微處理機實驗課的專題，我們是利用Zedboard來開發的，結合廠商提供的API還有spec文件。我們的題目是Puzzle Square，智益方塊，一款可以兩人互動的益智類遊戲，在螢幕方框中會出現兩個方塊，紅色代表攻擊，黃色代表防守，控制左邊按鍵的人，按制鈕會先控制左上方的紅色方塊，控制右邊按鍵的人，按制鈕會先控制右下方的黃色方塊，方塊會自動前進，兩者是以相向方向前進，我們可以利用控制按鈕來控制方塊的上下，當紅色方塊成功攻擊到黃色方塊就會觸發exception，使程式停並顯示圖案，那之所以我們會稱他為益智遊戲呢，因為他有一個特色是當方塊撞到牆壁之後，顏色會作互換，所以相對的就是等於攻守交換，原本攻擊的人當碰到牆壁之後就變成防守的人，反之就是防守的人變成攻擊的人，所以不論是攻擊方或是防守方都需要考慮清楚目前位置以及轉換後的位置來作最適合的攻擊或是防守，這也是我們稱他為智益方塊的原因。

1. **使用平台、應用技術及相關知識：**

* ZedBoard簡介

ZedBoard是基於Xilinx Zynq™-7000擴展式處理平臺（EPP）的低成本

開發板。此板可以運行基於Linux，Android，Windows®或其他OS/ RTOS

的設計。此外，可擴展介面使得使用者可以方便訪問處理系統和可程式

設計邏輯。

* IDE

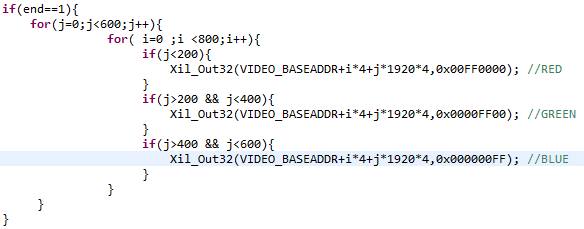
使用的是Xilinx ISE Design Tool 14.7版，介面跟Eclipse很像，所使用的語言是類C的語言，但有一些輸入輸出的函式不一樣，需要去查spec才看得懂或是去看API的使用方法，但是主要的程式邏輯是一樣的。

* 顏色控制

首先先了解RGB三種顏色16進位表示法以及格式

例如:0x00FF0000是紅色，0x0000FF00是綠色，0x000000FF是藍色

因為其他顏色都是由這三個顏色的集合所調配出來的，像我們作品中的黃色為紅色加上綠色所以為0x00FFFF00，比較的特殊的是白色為0x00FFFFFF，黑色為0x00000000



* 螢幕邊界

此次使用的螢幕大小為長800寬600的4:3螢幕，要取得這個資訊才能讓我們接下來做圖形位置的控制

* 函式控制起始位置

由數字來控制從螢幕開始繪圖的起始位置，像我們一開始的位置為0x000000



* 函式繪圖輸出



VIDEO\_BASEADDR為起始位置，i是橫向位置控制，j是縱向位置控制，最後在填出要輸出的顏色

* 函式讀取輸入的按鍵數值

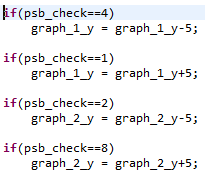


利用此函式才能得知移動方向

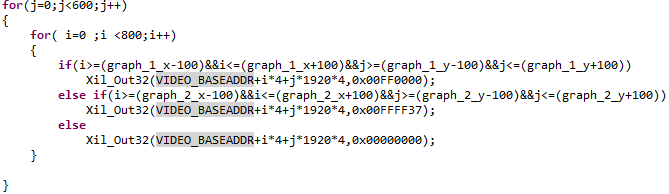
1. **實作內容**

* 畫正方形





我們畫出兩個正方形，為了好控制移動位置，先設定兩個正方形的x與y座標，接著用這座標來決定我們正方形的位置



接著以x及y的座標為中心點來畫出正方形，如上圖所示，我們的正方形是邊長為200，所以要用-100和+100來做為填色範圍

* 速度控制



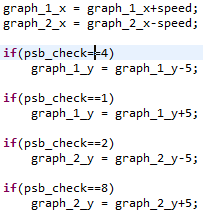
利用速度的正反來控制兩正方形的移動方向，但因為圖形會反轉所以在每次遇到螢幕邊界之後，速度會變號，正變負，負變正

* 圖形顏色反轉



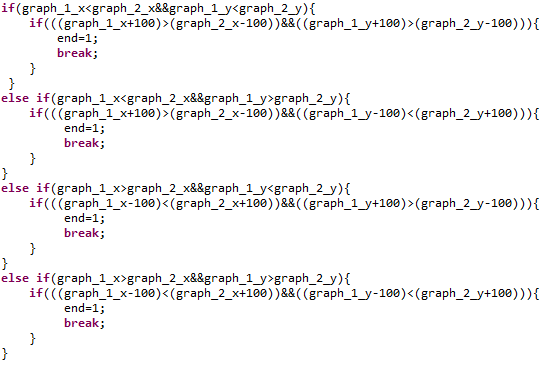
由每次的速度正負來判斷說兩正方形是否要反轉顏色，例如上圖當速度大於0的時候，graph1是紅色，graph2是黃色，而當速度小於0的時候則相反，graph1是黃色，graph2是紅色

* 按鍵輸入移動



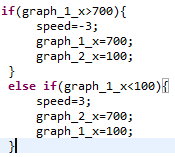
經由實驗結果，得知ZedBoard的上下左右按鍵的數值來做為正方形上下左右移動的依據，上是4，下是8，左是1，右是2

* 碰撞處理



先分為四種相對位置的碰撞，接著用兩個正方形的重疊來判斷是否碰撞，如果碰撞則end=1及跳出迴圈

* 邊界處理



用正方形中心點來判斷是否碰到邊界，當碰到之後，兩正方形位置移動方向反向以達到反彈效果

* 結束條紋圖形



由螢幕大小來分析條紋出現位置，接著嘗試各種顏色組合來畫出12種顏色圖形

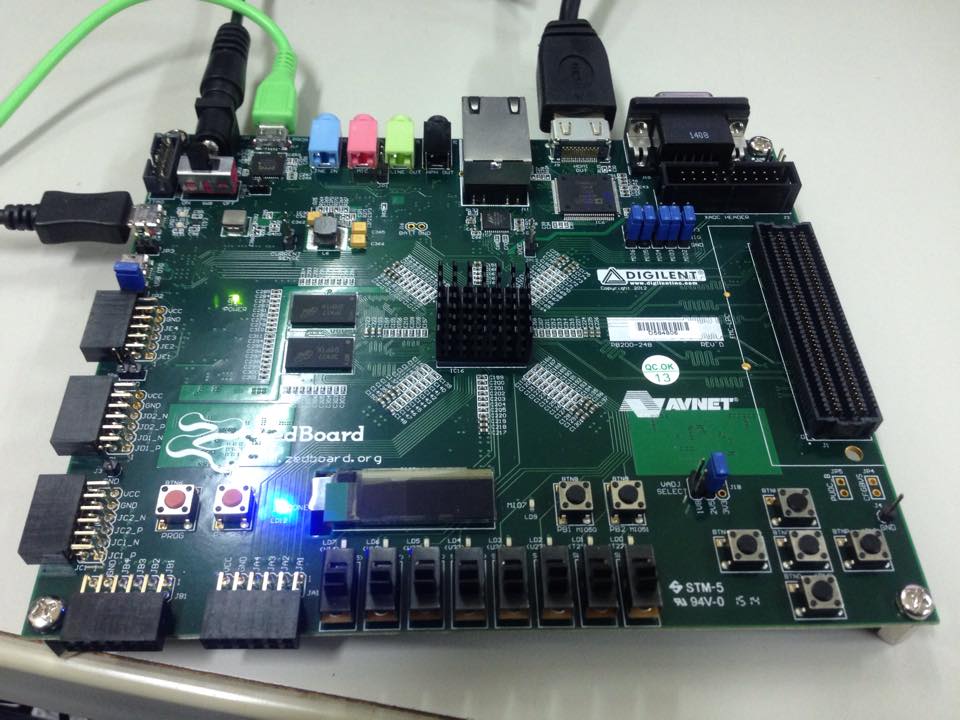
1. **實作成果(可含照片)：**

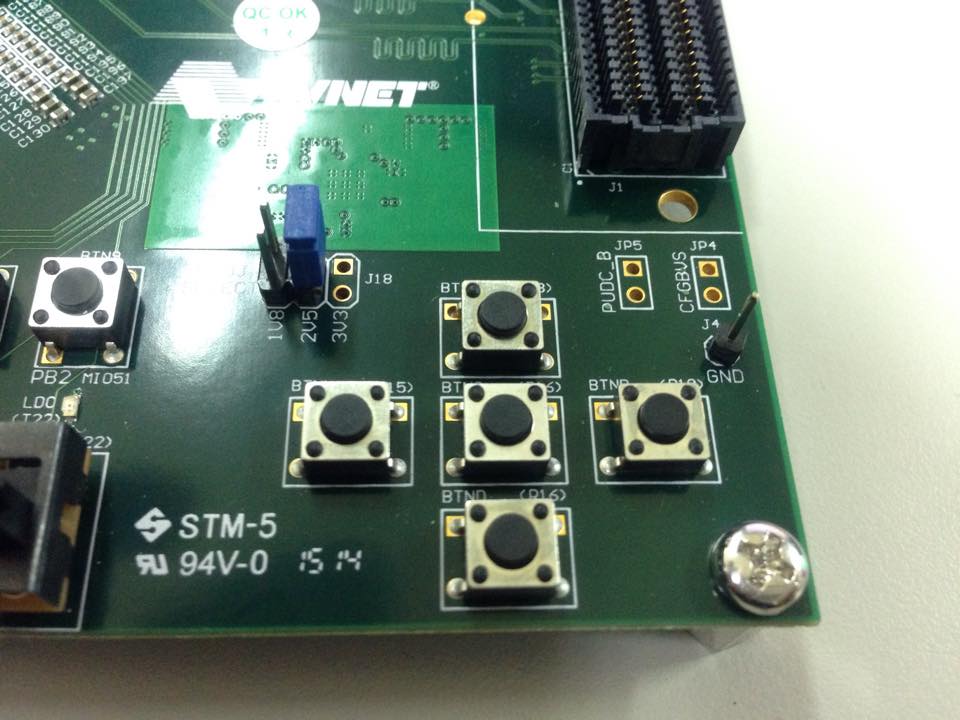
**遊戲規則:**

紅色的正方形為攻擊方，黃色的正方形為防守方，當防守方成功防禦抵達邊界，則攻守互換，紅色正方形變成黃色正方形成為防守方，黃色正方形變成紅色正方形成為攻擊方，若攻擊方成功碰撞到防守方則獲得勝利遊戲結束，獲得美麗的條紋

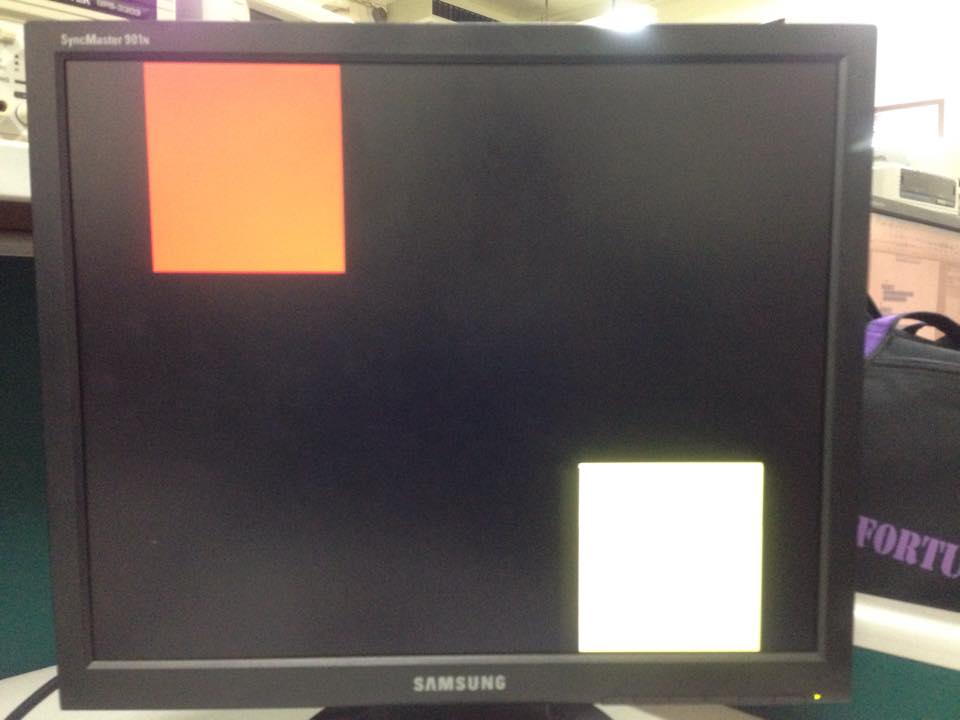
**步驟:**

Step0:利用ZedBoard的控制鍵

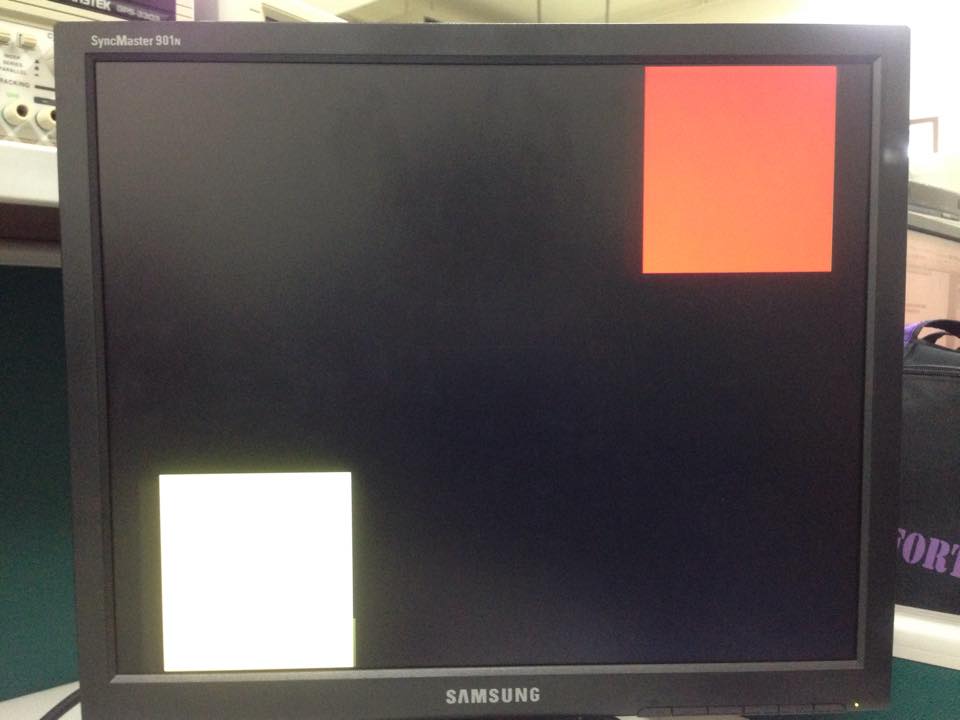




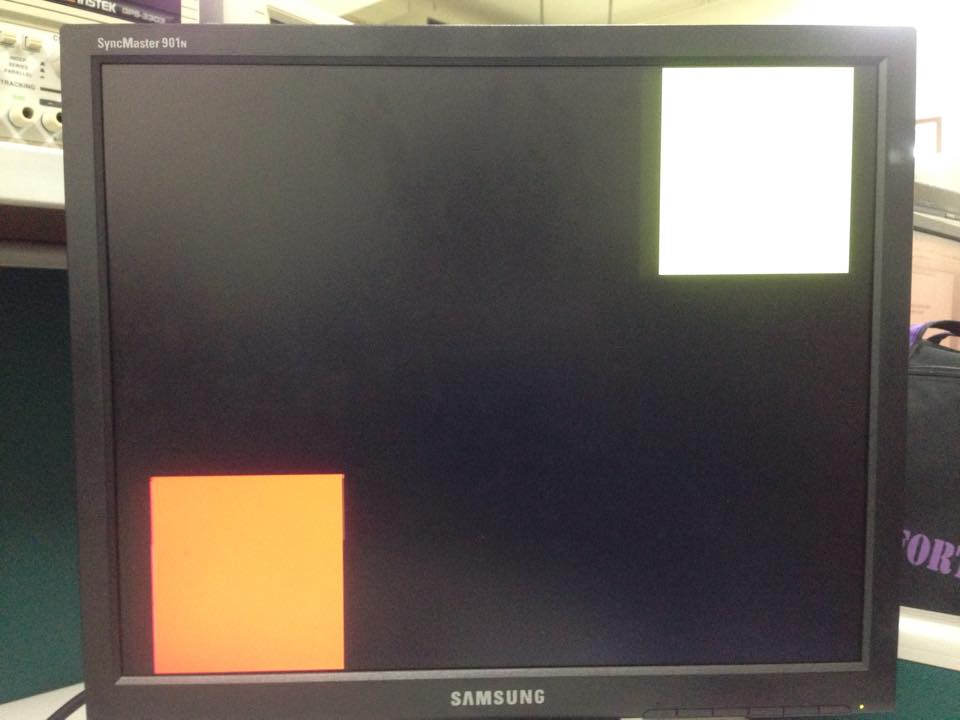
Step1:兩正方形分別從左上及右下向中間移動



Step2:兩正方形快接近邊界時



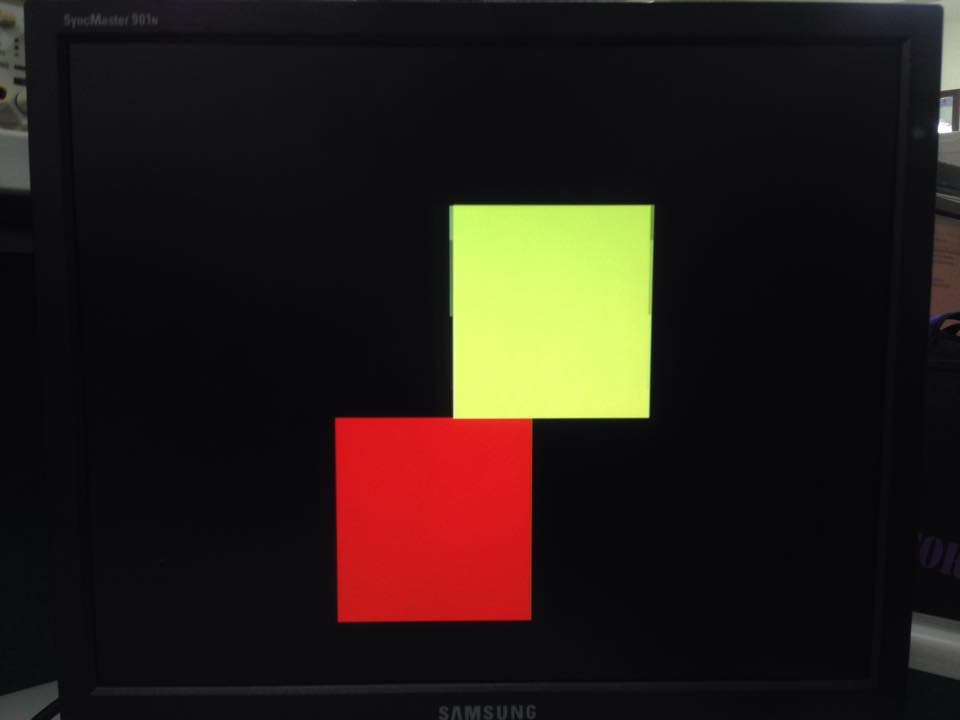
Step3:兩正方形碰到邊界且反轉後



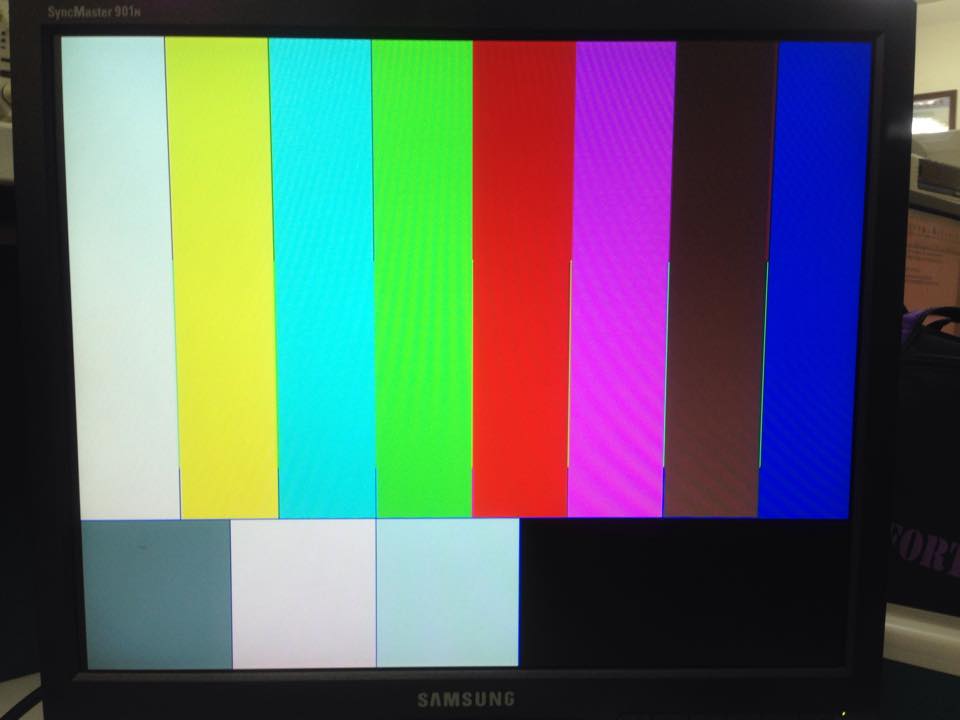
Step4:利用控制鍵使兩正方形互相靠近



Step5:碰撞發生



Step6:碰撞發生後



1. **心得 :**

這學期的LAB真的學到很多東西，最後的專題也成功的作出一個作品出來，真的很高興，第一次接觸ZedBoard的時候，還記得那天學長只是要我們照著投影片的步驟，然後能run程式最後在外接螢幕上秀出紅綠藍的橫條紋即可。後來嘗試去了解在才發現起始位置的重要，要怎麼去設定才能使我們要顯示的東西出現在我們預期的位置上，接著開始去研究顏色要怎麼更改，RGB三原色的應用，各種顏色的數值這些基本的東西，最後才去想要怎麼利用兩個for迴圈控制i,j來劃圖形。當有了這些基礎的概念，才開始思考要做什麼專題，其實一開始是想要把圖片load進來的，但無奈和學長研究API很久還是無法成功，是有下下策就是用一個二維陣列來劃圖，一個pixel一個pixel來堆圖案，但如果是複雜圖形應該就無法了。這次我們的主題，是從老師想要我們作按鍵的控制以及可以有使用上有互動性下所產生的，整個程式架構看似沒那麼複雜，但是其中確要考慮到很多細節，像你的背景底圖是要每次都刷新，或是當方塊在移動的時候馬上補顏色上去，覺得最困難的地方是在判斷怎麼樣的條件下，攻擊方成功攻擊使程式結束，因為這邊牽涉到四個位置方向的判斷，右下碰左上，右上碰左下，左上碰右下，左下碰右上，而且要作到很精準的判斷，所以需要仔細的去算方塊還有牆壁的距離等等去推判斷式該怎麼寫，還有顏色圖形反轉上也不簡單。藉由這次的專題實作，真的學到很多東西，不管是從網路上或是學長還是同學身上，在最後大家的demo時候更是見識到別組優秀的作品，都很讓人為之驚艷，但我想我們的作品也是很不錯的，需要思考力以及快速的手速來控制按鈕才能取勝，而且背後也牽涉到許多細微的控制運算和邏輯上的技巧。ZedBoard上還有許多的功能還沒有去使用到像是switch按鍵的控制，Led的控制顯示等等，而當然這只是小部分，未來如果還有機會使用開發板會嘗試去做出更多不一樣的東西，像是結合IPcam就是一個不錯的題材。

1. **參考文獻 :**
2. <http://www.xilinx.com/tools/sdk.htm>
3. <http://forums.xilinx.com/t5/Design-Tools/ct-p/DESIGN>
4. <http://narong.ee.engr.tu.ac.th/embedded/document/xilinx_drivers.pdf>
5. <http://forums.xilinx.com/>
6. <http://www.xilinx.com/support/clearexpress/websupport.htm>
7. <http://www.xilinx.com/support/programr/specs.htm>

一共有六個標題，分別為 一.專題簡介 二.使用平台、應用技術及相關知識

三.實作內容 四.實作成果(可含照片) 五.心得 六.參考文獻，

六個標題一個都不可以缺少，標題中的內容越多越好，讓我們可以清楚的了解到你們所做的專題。