EzProg-I EzTemplate Tutorial

(Version 1.1)

ICP DAS CO., LTD.

泓格科技股份有限公司

Warranty

All products manufactured by ICPDAS Inc. are warranted against defective materials for a period of one year from the date of delivery to the original purchaser.

Warning

ICPDAS Inc. assumes no liability for damages consequent to the use of this product. ICPDAS Inc. reserves the right to change this manual at any time without notice. The information furnished by ICPDAS Inc. is believed to be accurate and reliable. However, no responsibility is assumed by ICPDAS Inc. for its use, or for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Trademark

The names used for identification only maybe registered trademarks of their respective companies.

License

The user can use, modify and backup this software on a single machine. The user may not reproduce, transfer or distribute this software, or any copy, in whole or in part.

Technical Support
If you have problems about using the product, please contact ICP DAS Product Support.

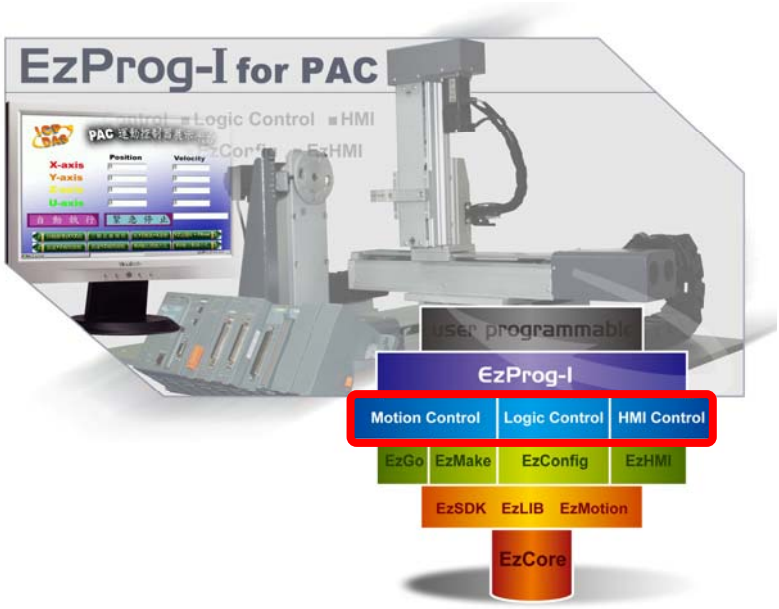
Email: Service@icpdas.com

目錄

1 EzProg-I 介紹

近年來工業控制器 PAC 的發展逐漸成熟，PAC 應用也越來越廣，使用者迅速增加所需功能，且自動化設備需求亦日益蓬勃，為協助新使用者對 PAC 控制器能輕鬆入門，因此 ICP DAS 泓格科技，設計此套 EzCore 軟體引擎，主要設計目標，是為了精簡工業自動化系統的開發時程。

EzCore 引擎主要工作方法目為在 eVC++ ,VS2008 平台輔以 EzProg-I 設計，來架構控制系統。整組套件包括了設定測試工具，HMI 輔助設計人機介面，我們架設了一個 Framework 包括了 IO、暫存器、斷電保持暫存器，利用 EzConfig 工具協助了系統規劃與設定與 IO 測試。Motion control 部分有 EzGo、EzMake 為協助測試與設計伺服馬達運動控制的工具。工程師可以更專注在控制邏輯的設計，並有效的縮短設計時程。



PAC 系統路徑: 因 EzProg-I 在各平臺安裝位置略為不同，為說明方便，我們會將 EzProg-I 的系統路徑統一稱為 **EzProg_Path**，在文中看見時請在此查閱。

WinCon W-8x8x-GM1: EzProg_Path = CompactFlash
MPAC: EzProg_Path = System_Disk

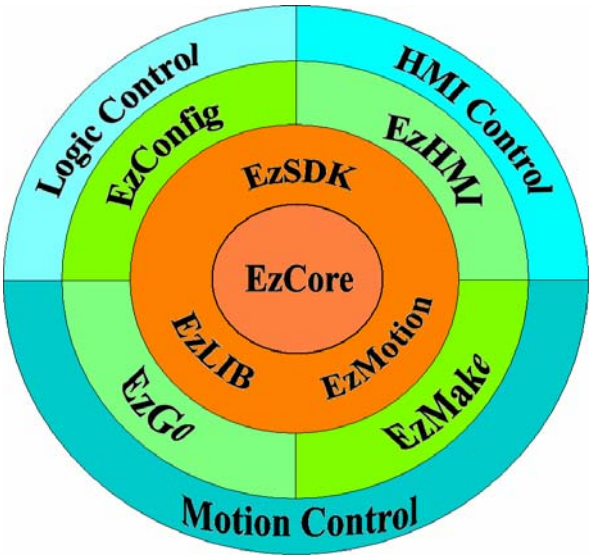
開發環境: 因 EzProg-I 在各平臺安裝的 WinCE OS 版本略為不同，為說明方便，我們會將微軟提供的開發環境的統一稱為 MS VC++，在文中看見時請在此查閱。

WinCon W-8x8x-GM1: MS VC++ = eVC++ 4.0

MPAC: **MS VC++ = VS.net 2008** 或更新的版本

1.1 EzProg-I 內容

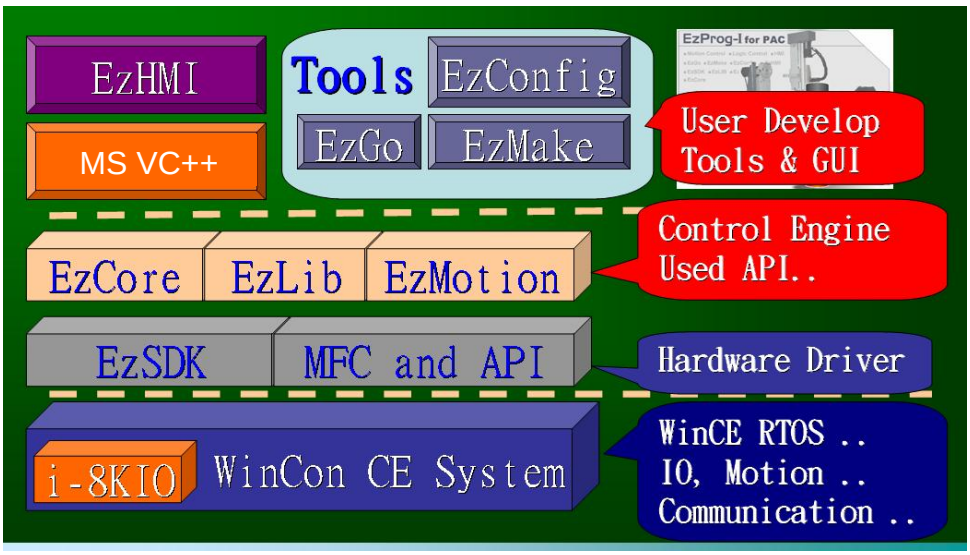
EzCore 與 eVC++4.0:



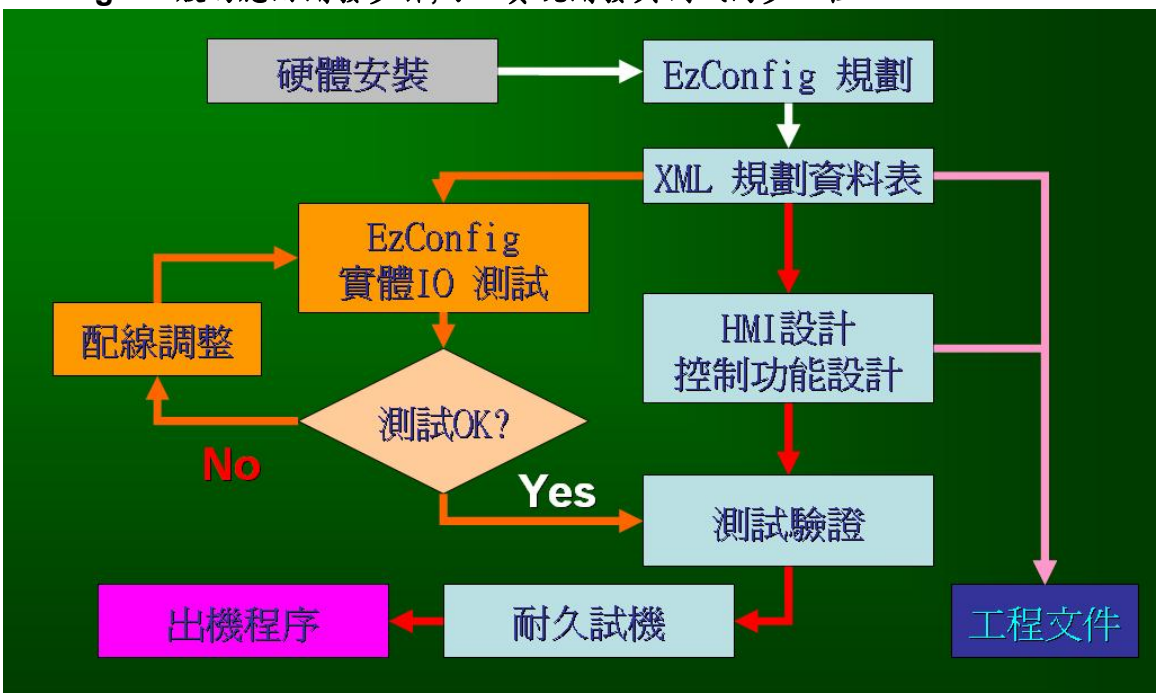
EzProg-I 是基於 **MS VC++** 開發出來的一套開放型開發套件, 其套件核心為 EzCore(DLL), EzCore 提供基本系統應用的 API, 並緊密結合硬體 DIO, AIO, Motion, EzHMI, 等應用.
以 **MS VC++** 開發編譯程式的主要工具, 除 EzProg-I 提供的開發資源外, 您更可以利用微軟 **MS VC++** 達成更多的應用功能

EzProg-I 發展基本結構:

最底層的是 WinCE OS 與硬體基本驅動程序,再上一層是 IO API 與 MFC API, 再上一層是您主要使用的 EzCore 與 Ezlib 及 Motion API,而最上層是用 MS VC++ 設計控制程序,人機操作界面(EzHMI)Motion,與其他工具的整合運用.



EzProg-I 一般的應用開發步驟,可以實現開發與測試同步工程:

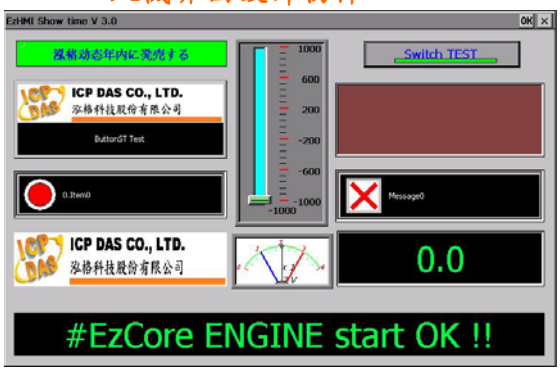


EzCore 主要功能:

EzCore 除提供類似 PLC 控制邏輯的多工即時 SCAN 執行引擎與中斷程序,還有一般所有基本 DIO(X.Y) /AIO /M/D/F/DB/C/T/MSG…等等的 API,而這些 API 就提供與所有其他開發資源結合的界面,例如 EzHMI,Motion,EzLIB 等等。

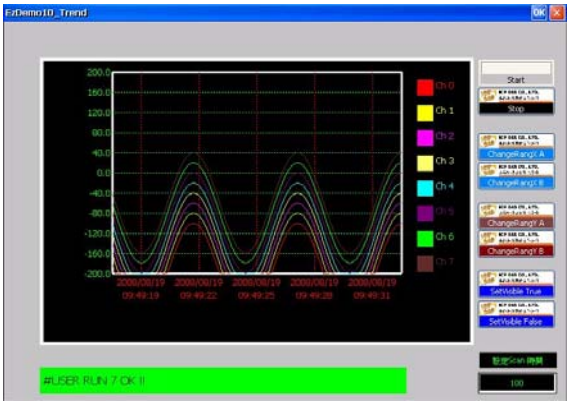
EzCore 提供八個定時執行的 RSTR ,八個單執行的 USER THREAD,與 DI(i8048), Motion(i8092,i8094 系列)中斷服務程序,在合理系統執行負荷下,可以得到預期的即時控制效果。

EzHMI 人機界面設計物件:



EzProg-I 人機界面物件
EzHMI(ActiveX)
直接顯示與控制 IO 狀態
直接顯示 EzProg-I 內部暫存器狀態
直接輸入到內部暫存器狀態
UNICODE 多國語系顯示支援
支援大型輸入介面
支援動態警告介面
支援動態更換圖片介面
物件可支援不同 Windows 字型
物件可支援不同顏色設定
物件可以設定為 Enable 或 Disable

EzLIB 其他的應用函式庫:



EzLIB 提供方便的應用程式庫
.各種資料轉換功能
.讀取日期時間功能
.方便的本文(CDC)繪圖功能
.BMP 開檔存檔繪圖功能
.簡易 TCP/IP 連線功能
.簡易的 FTP 傳檔功能
.Trend 趨勢圖功能
.生產配方檔與檔案 R/W 功能

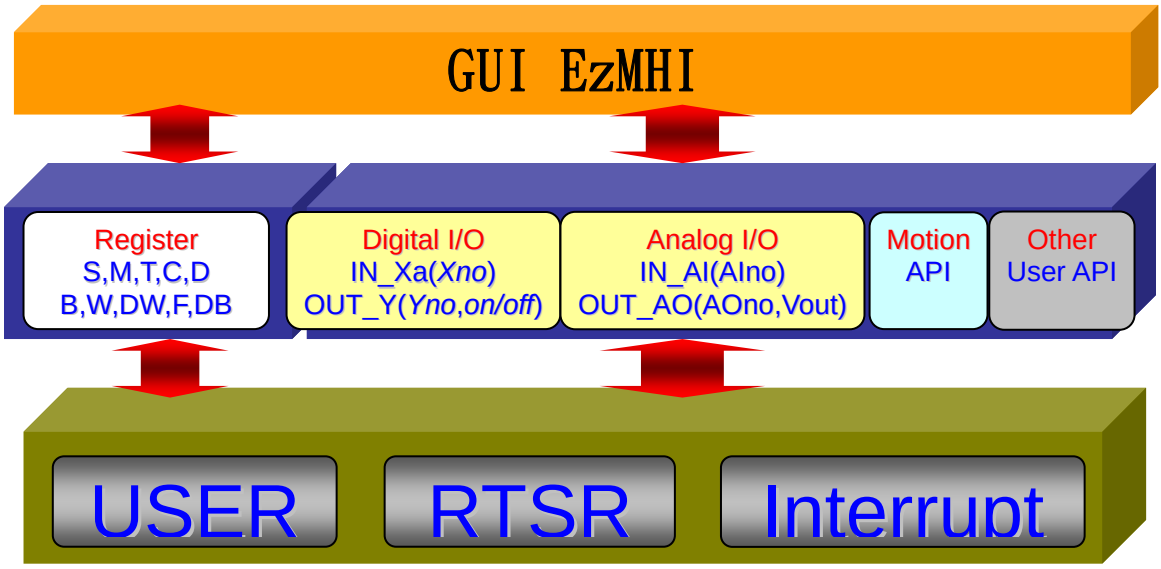
馬達運動控制開發資源:

EzProg-I 可以支援 i8092F、i8094、i8094F、 i8094A、i8094H 系列 2 軸與 4 軸馬達運動控制卡,配合 EzProg-I 與 EzGo 及 EzMake 使用時可以達到相關的控制功能需求要。

設計與使用詳情請參閱 i8092、i8094、i8094H 相關使用手冊。

1.2 EzProg-I 使用架構說明

主要架構如下圖:



整個系統由三個部份組成:

1. 上層 EzHMI:控制系統設計者可以透過 EzHMI ActiveX 物件的屬性表設定，就可以清鬆的設計人機操作介面，而這些 ActiveX 物件直接使用 **EzCore** 的 **IO** 與暫存器(變數) ，直接存取輸入與顯示狀態等等，其中還包括 **UNICODE** 多國語系支援。

2. 中間層 API: 軟硬體經過 **EzConfig** 的規劃後，EzProg-I 提供一系列統一的 API，系統開發設計者,很快就能熟悉與運用這些 API，大幅簡化以前客戶需針對不同的 **DIO/AIO** 模組使用不同的底層 **API** 的程續，很容易就可以存取到 EzCore 的變數與實體 IO 輸入輸出動作，例如 **DO Y** 的 **on/off** ，輸出 **AO** 的值，設定 **D** 暫存器的值等等。

3. 下層控制邏輯設計: 控制軟體實際運作程序，提供三種不同的控制軟體設計方法:

3.1 使用者自定程序:一般的執行程序，啟動後只執行一次。

3.2 定時執行程序: 類似 **PLC** 的執行方法，啟動後系統會按使用者設定時間定，定時執行。

3.3 硬體中斷程序:**EzProg-I** 經過適當的設定後，可以接受 **DI** 信號的中斷與 **Motion** 的中斷，並在此設計中斷的服務程序。

而這三種運作程序，一樣可以透過中間層的 **API**，很容易就可以存取到 EzCore 的變數與實體 IO 輸入輸出動作，例如 **DO Y** 的 **on/off** ，輸出 **AO** 的值，設定 **D** 暫存器的值等等，而達成控制系統的控制功能，期望系統開發者能達成同步與快速開發出易於管理與一定穩定品質的控制應用軟體。

1.3 EzProg-I IO 與暫存器列表

EzProg-I IO 與暫存器的使用詳細說明，請參考 EzProg-I_Tool 手冊 3.7 章節。

目前 EzProg-定義的暫存器範圍如下表:

Specification		Register	Register number		Data type	Size	Range
Slot Device Register	Digital Input	X	Local DI:	0 ~ 777	bit	1 bit	true / false
			FRNet DI:	1000 ~ 7777			
	Digital Output	Y	Local DO:	0 ~ 777	bit	1 bit	true / false
			FRNet DO:	1000 ~ 7777			
	Analog Output	AO	Local AO:	0 ~ 511	float	4 bytes	3.4E +/- 38
	Analog Input	AI	Local AI:	0 ~ 511	float	4 bytes	3.4E +/- 38
Software Register	Timer	T	None Retain:	1 ~ 299	bit	1 bit	true / false
	Counter	C	None Retain:	1 ~ 511	bit	1 bit	true / false
			Retain:	512 ~ 1023			
	Flag	M	None Retain:	1 ~ 6999	bit	1 bit	true / false
			Retain:	8192 ~ 15999			
	Step	S	None Retain:	1 ~ 8191	bit	1 bit	true / false
	long integer	D	None Retain:	1 ~ 3599	long integer	4 bytes	–2,147,483,648 to 2,147,483,647
			Retain:	4096 ~ 7999			
	BYTE	B	None Retain:	1 ~ 699	unsigned char	1 byte	0 to 255
			Retain:	1024 ~ 2047			
	WORD	W	None Retain:	1 ~ 1023	unsigned short	2 bytes	0 to 65,535
			Retain:	1024 ~ 1999			
	DWORD	DW	None Retain:	1 ~ 4095	unsigned long	4 bytes	0 to 4,294,967,295
			Retain:	4096 ~ 8191			
	Float	F	None Retain:	1 ~ 1899	float	4 bytes	3.4E +/- 38
			Retain:	2048 ~ 3999			
	Special Type	DB	None Retain:	1 ~ 49			
	Message	MSG	Retain:	1 ~ 249	30 wchar_t	60 bytes	30 unicode char

下表將列出 EzProg 中已指定用途的暫存器，請按照用途使用，而保留給系統特定用途的暫存器請勿任意使用以免造成不可預知的結果。

下表是 EzProg 中已指定用途的暫存器，請按照用途使用:

Register Number	Note
D8000	Muti-language
M16000	HMI:ColorEdit Input Control

1.4 EzProg-I 專案開發範本

為了強化 EzProg-I 應用專案開發與訓練課程的學習效率，ICP DAS 設計 EzTemplat 的開發範本，利用 EzTemplat 的開發範本的範例程式分階段說明並實作，使每位參與的學員在最短時間內，獲得利用 EzProg-I software package 撰寫程式的基本能力，本手冊範例會以 EzTemplate 為啟始範本專案。

1.4.1 基本 EzTemplat 架構簡介

EzTemplat 的開發範本，包括:

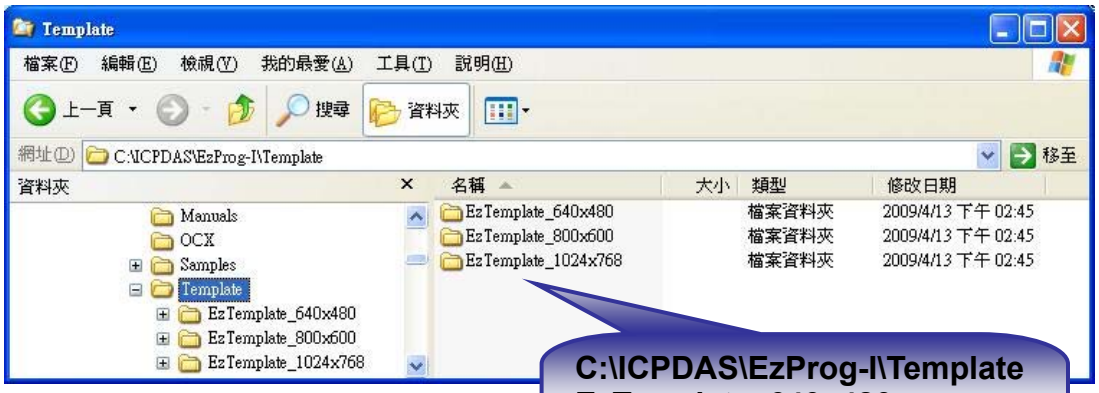
1.預建 50 個 HMI 的頁面,透過定義，可以修改使用數量。

2.基本的 EzPorg-I 設定與啟動已經建製完成。

3.HMI 的頁面切換已經建製完成，利用設定 ButtonST 的 M(接點)屬性，即可有換頁功能。

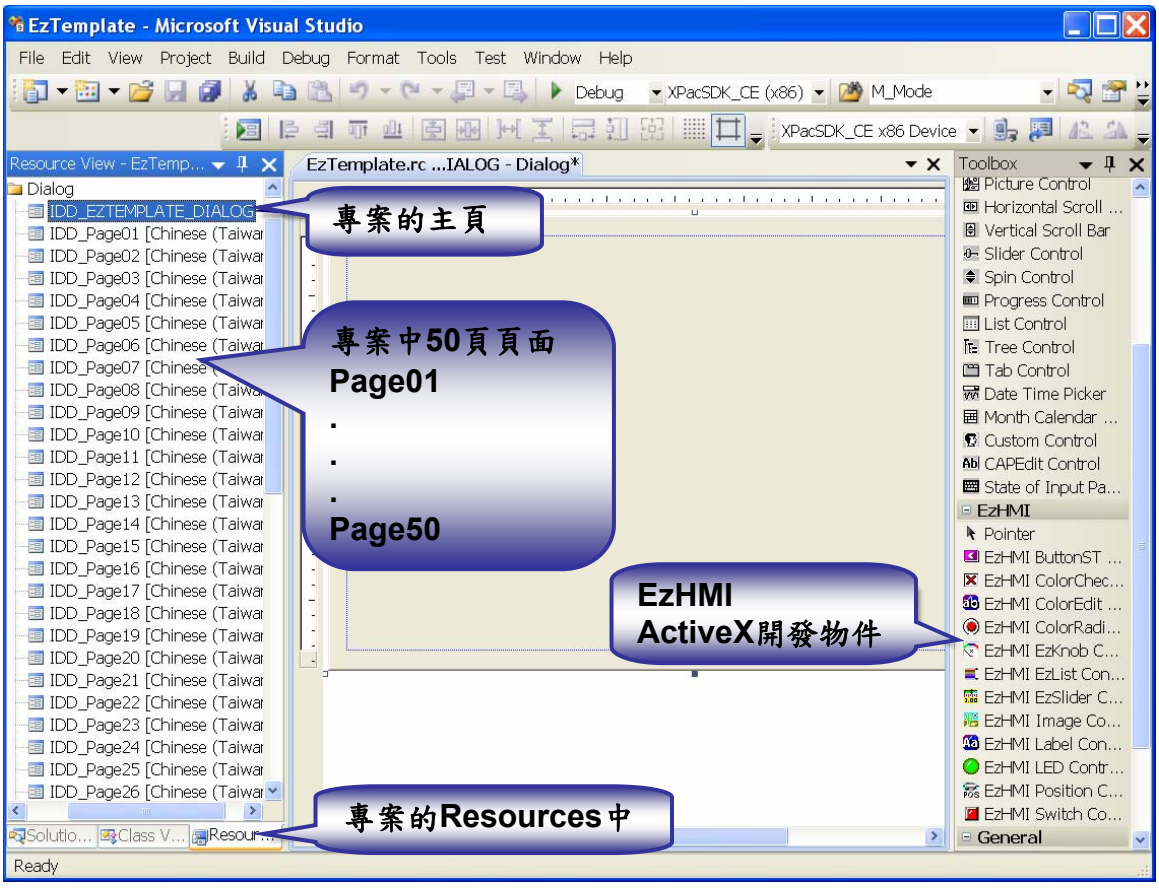
4.User Thread 與 RTSR 框架已經建置，使用者輕易的就可以開始設計控制程式。

5.提供 640x480，800x600，1024x768 不同的畫面解析度範本，如下圖請複製您選擇的畫面解析度範本，到其他工作目錄，開始使用開發範本。



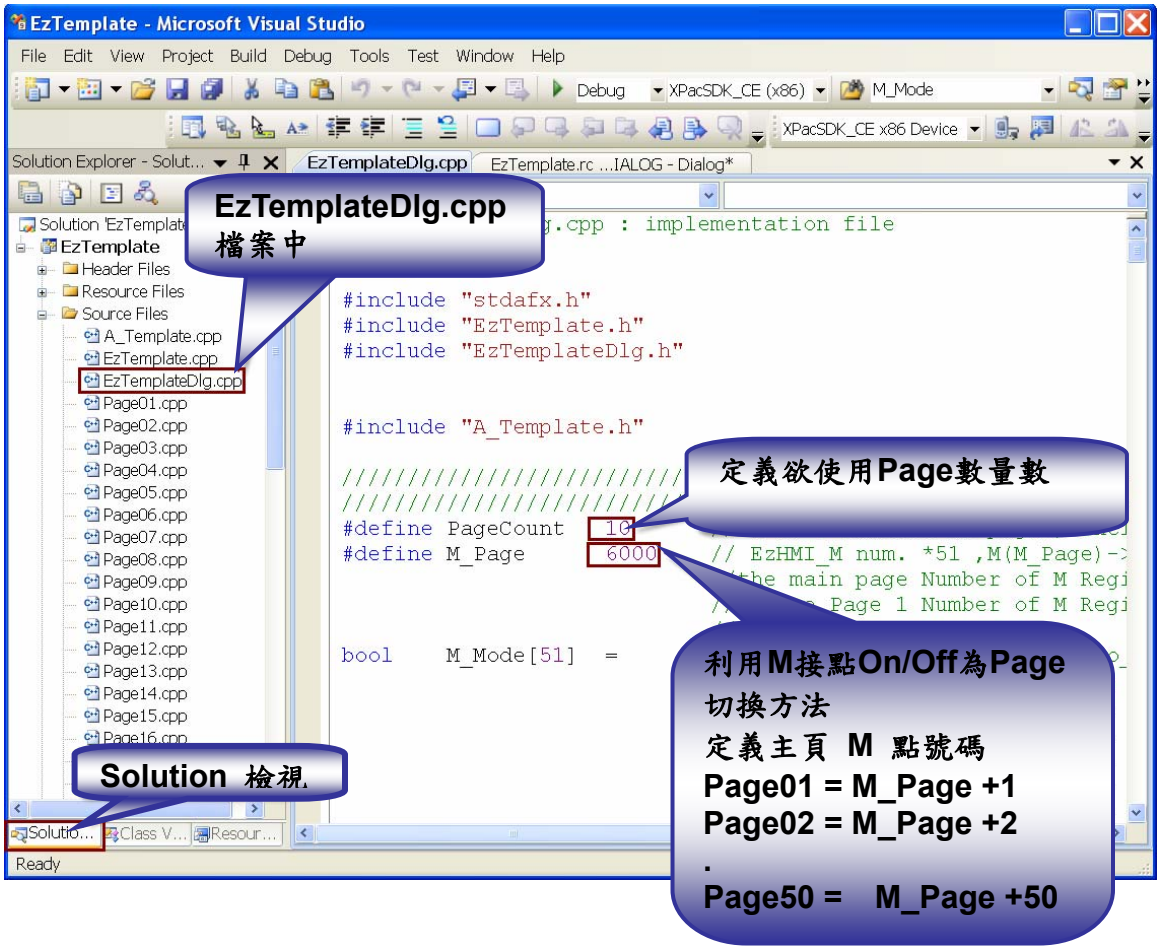
1.4.2 EzTemplate 50 個頁面

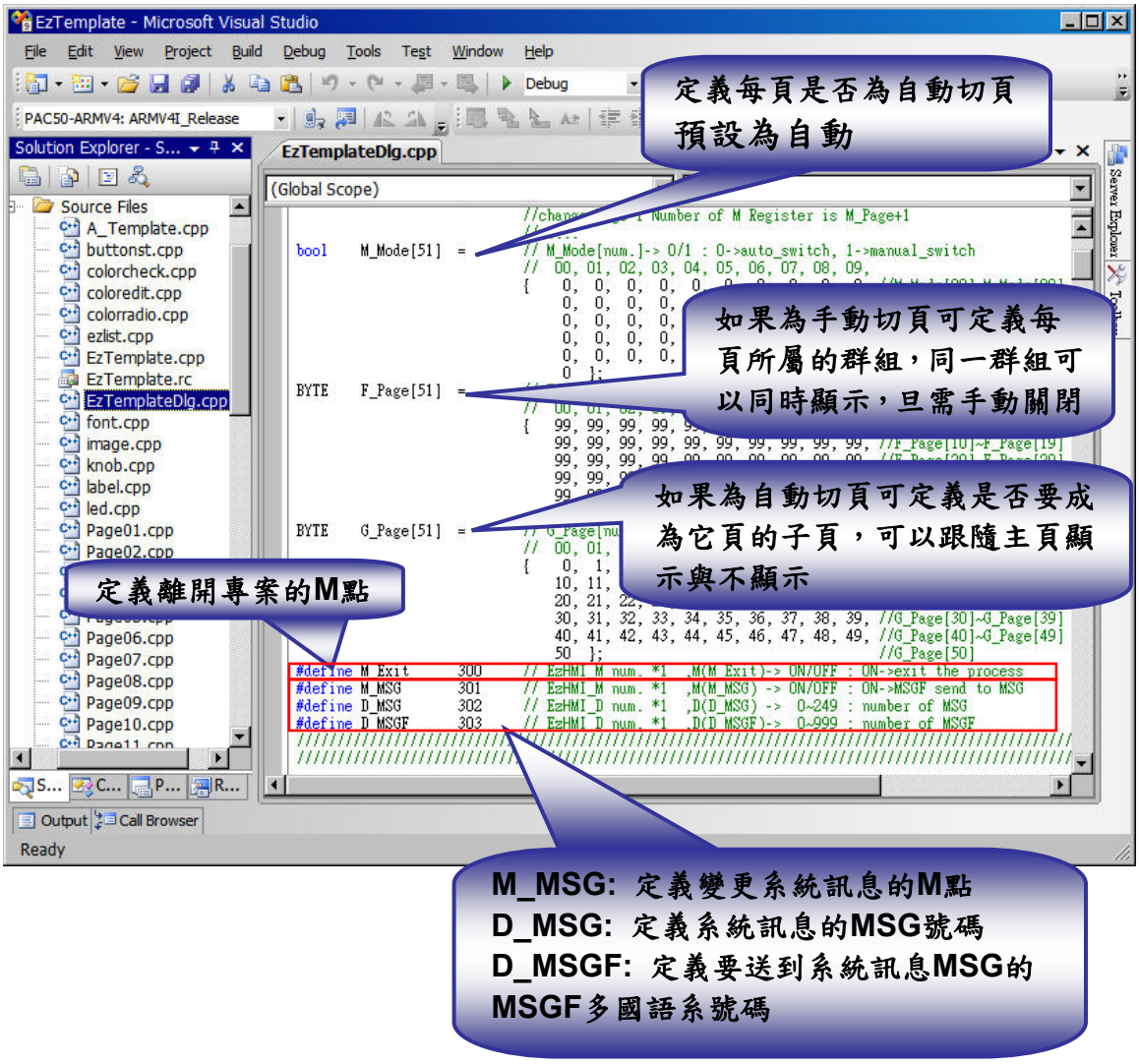
EzTemplat 的開發範本，包括主頁與 50 個頁面共 51 頁，基本的專案架構都已經先建置完成。



1.4.3 EzTemplate 的定義

一般畫面與系統定義:





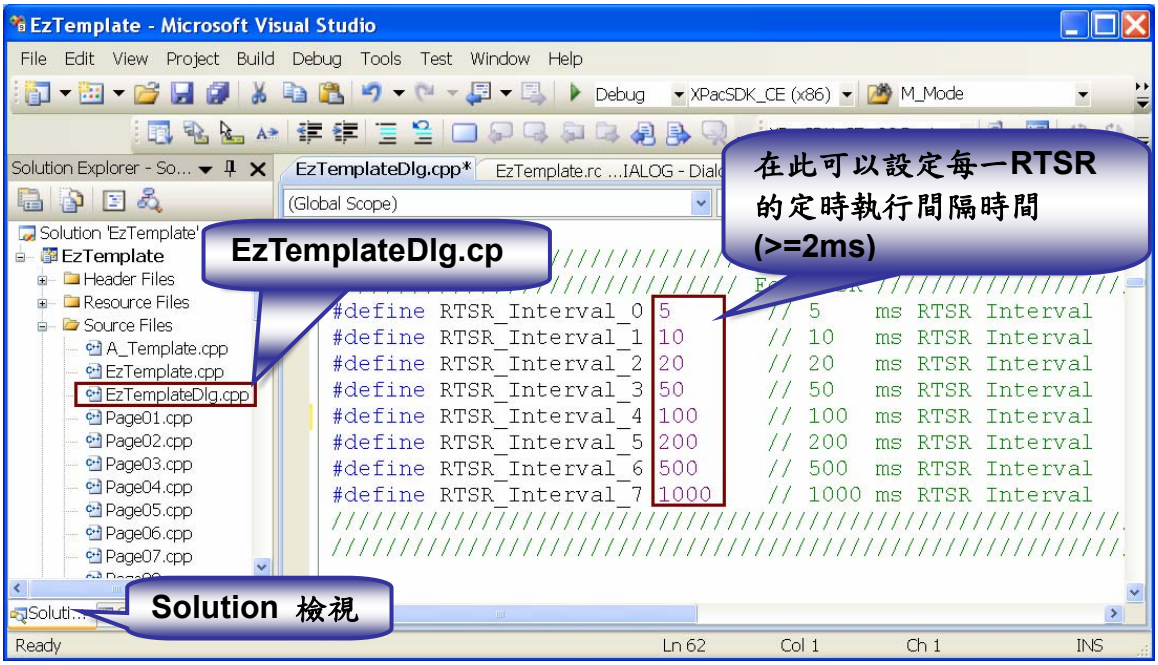
a.專案定義主要在EzTemplateDlg.cpp
檔案中

b.PageCount 50 **=>定義本專案欲使用Page數量數 0~50 可以依需要修改**

c. M_Pape 100 **=>定義本專案主頁的代表M 號碼可以依需要修改，**
當代表M 號碼 = true 時即表示要顯示代表的那一頁

d. M_Exit **=>定義要關閉本專案的代表M 號碼可以依需要修改，**
當代表M 號碼 = true 時即會離開本程式。

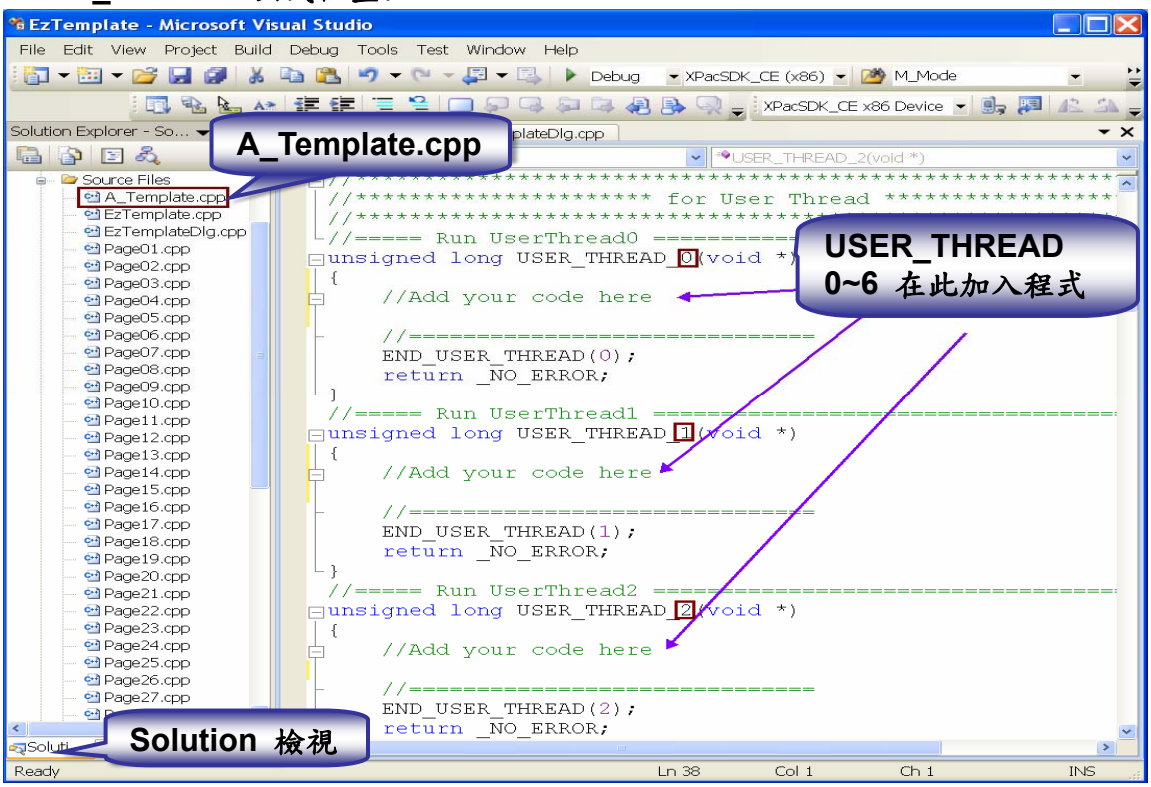
RTSR 系統定義：



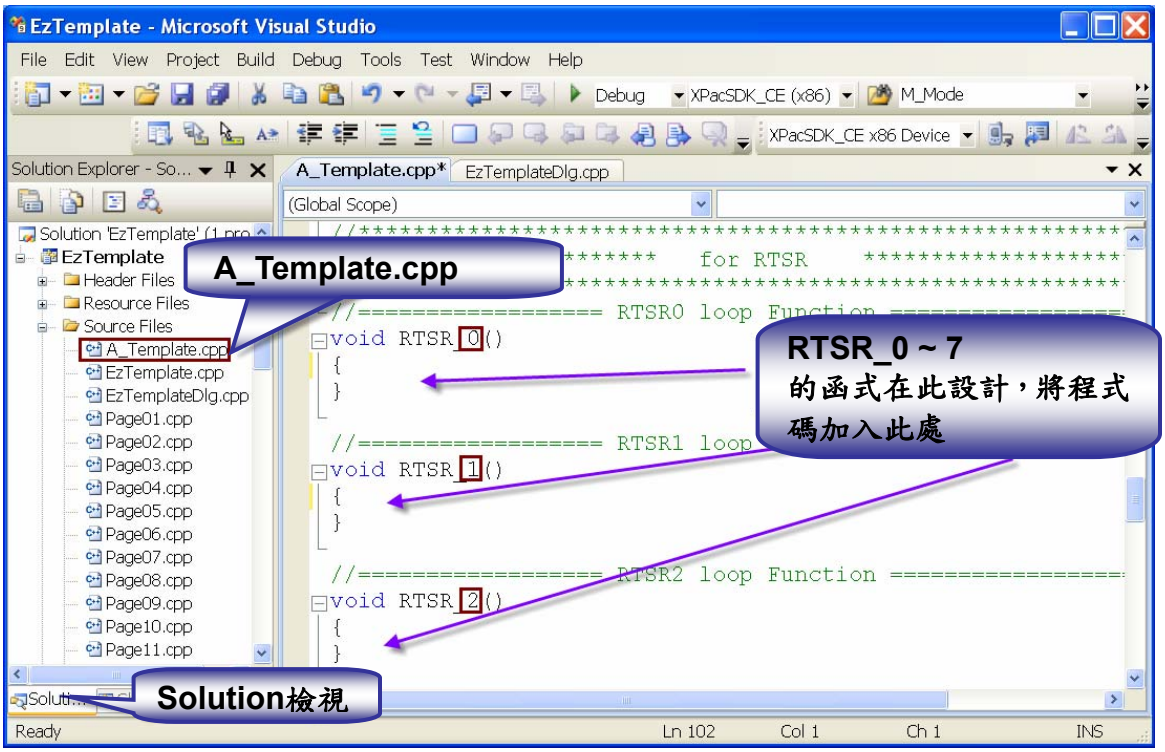
1.4.4 EzProg-I Function 的定義

為方便程式寫作，在 A_Template.cpp 中已定義 USER_THREAD 與 RTSR 函式方便直接在此寫控制程序。

USER_THREAD 函式位置：

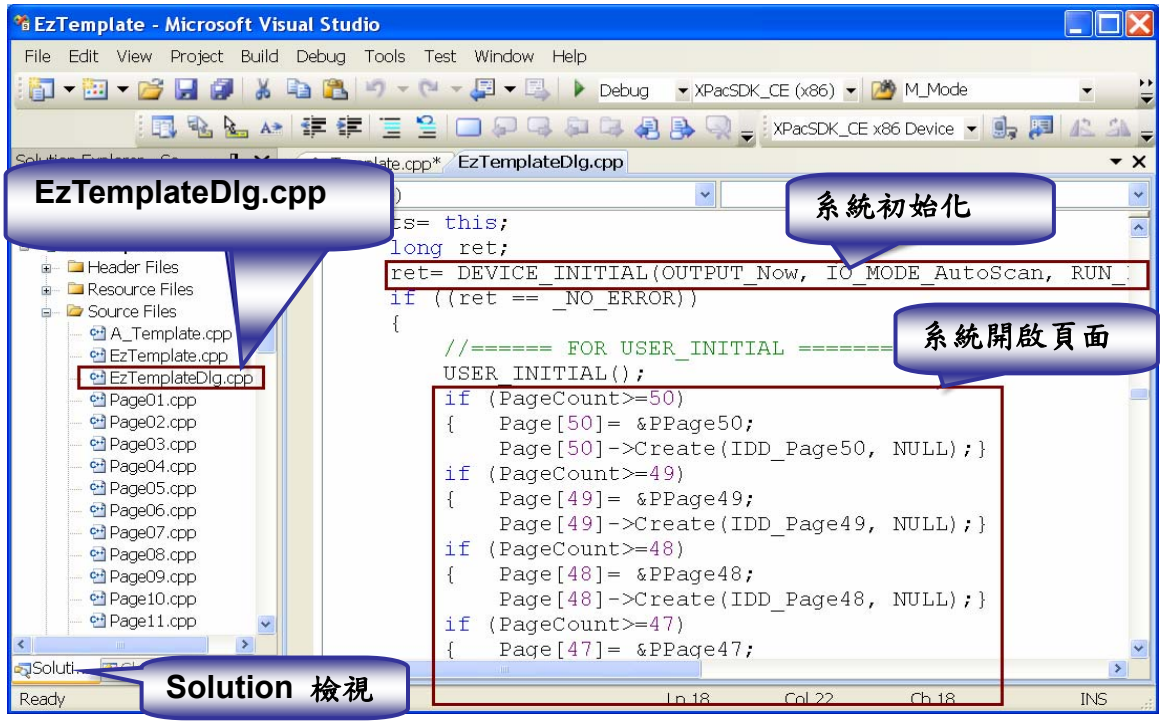


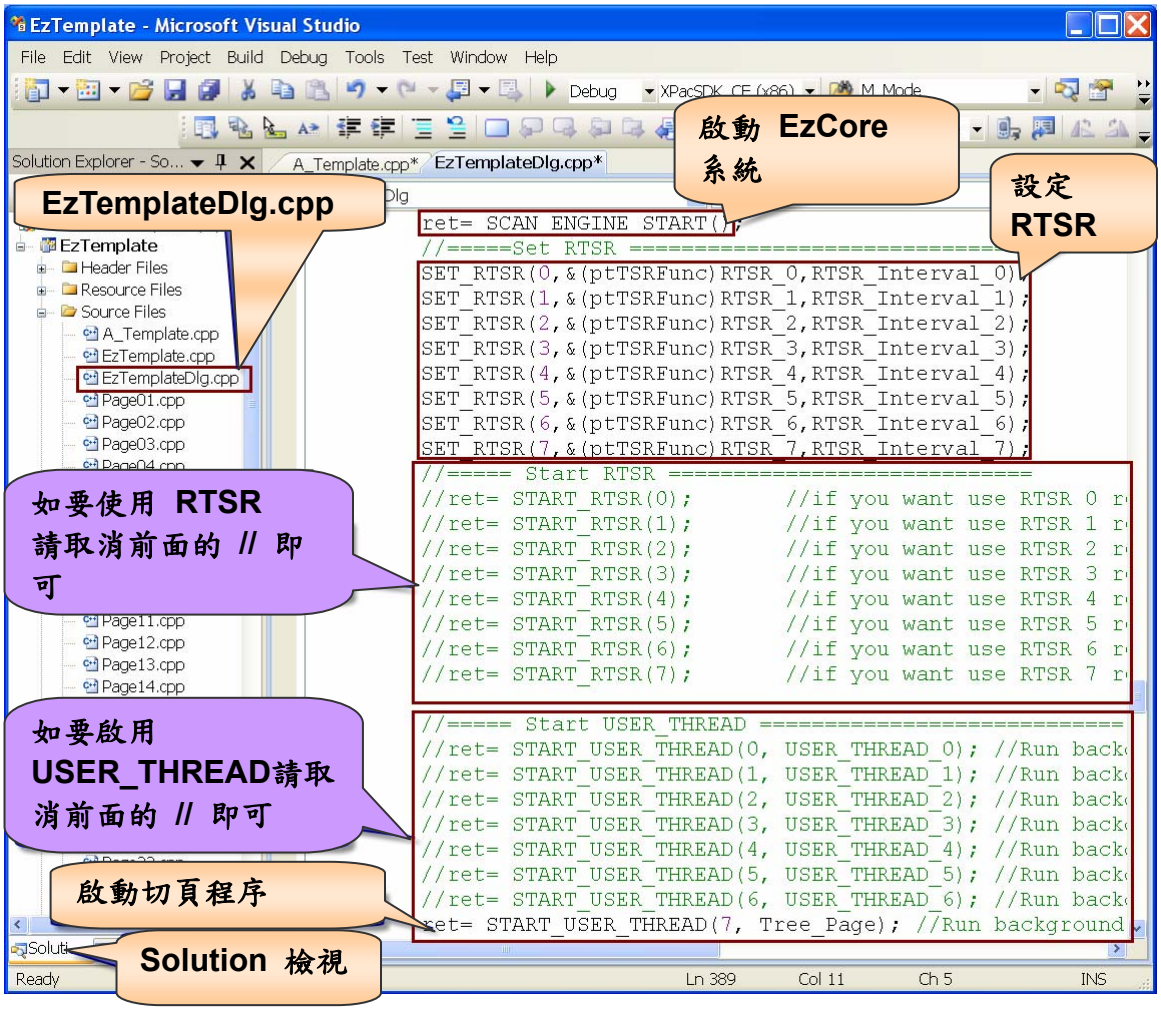
RTSR 函式位置:



1.4.5 EzProg-I 的啟動

範本中已經建好架構，再此只是說明讓您了解。




EzTemplate - Microsoft Visual Studio
File Edit View Project Build Debug Tools Test Window Help
Debug
啟動 EzCore 系統
設定 RTSR
EzTemplateDlg.cpp
Solution Explorer - So...
A_Template.cpp*
EzTemplateDlg.cpp*
EzTemplate
Header Files
Resource Files
Source Files
A_Template.cpp
EzTemplate.cpp
EzTemplateDlg.cpp
Page01.cpp
Page02.cpp
Page03.cpp
Page11.cpp
Page12.cpp
Page13.cpp
Page14.cpp
ret= SCAN_ENGINE_START();
//=====Set RTSR ============================
SET_RTSR(0,&(ptTSRFunc)RTSR_0,RTSR_Interval_0);
SET_RTSR(1,&(ptTSRFunc)RTSR_1,RTSR_Interval_1);
SET_RTSR(2,&(ptTSRFunc)RTSR_2,RTSR_Interval_2);
SET_RTSR(3,&(ptTSRFunc)RTSR_3,RTSR_Interval_3);
SET_RTSR(4,&(ptTSRFunc)RTSR_4,RTSR_Interval_4);
SET_RTSR(5,&(ptTSRFunc)RTSR_5,RTSR_Interval_5);
SET_RTSR(6,&(ptTSRFunc)RTSR_6,RTSR_Interval_6);
SET_RTSR(7,&(ptTSRFunc)RTSR_7,RTSR_Interval_7);
如要使用 RTSR 請取消前面的 // 即可
//===== Start RTSR ============================
//ret= START_RTSR(0); //if you want use RTSR 0
//ret= START_RTSR(1); //if you want use RTSR 1
//ret= START_RTSR(2); //if you want use RTSR 2
//ret= START_RTSR(3); //if you want use RTSR 3
//ret= START_RTSR(4); //if you want use RTSR 4
//ret= START_RTSR(5); //if you want use RTSR 5
//ret= START_RTSR(6); //if you want use RTSR 6
//ret= START_RTSR(7); //if you want use RTSR 7
如要啟用 USER_THREAD請取消前面的 // 即可
//===== Start USER_THREAD ============================
//ret= START_USER_THREAD(0, USER_THREAD_0); //Run back
//ret= START_USER_THREAD(1, USER_THREAD_1); //Run back
//ret= START_USER_THREAD(2, USER_THREAD_2); //Run back
//ret= START_USER_THREAD(3, USER_THREAD_3); //Run back
//ret= START_USER_THREAD(4, USER_THREAD_4); //Run back
//ret= START_USER_THREAD(5, USER_THREAD_5); //Run back
//ret= START_USER_THREAD(6, USER_THREAD_6); //Run back
ret= START_USER_THREAD(7, Tree_Page); //Run background
啟動切頁程序
Solution 檢視
Ready
Ln 389
Col 11
Ch 5
INS

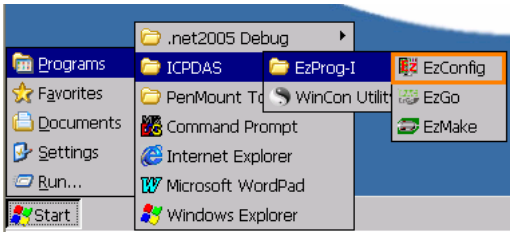
2 應用範例

在開始操作本章節的範例前，先準備好PC的開發環境，詳細安裝過程請參閱 EzProg-I Getting Started 手冊中的開發環境安裝說明。

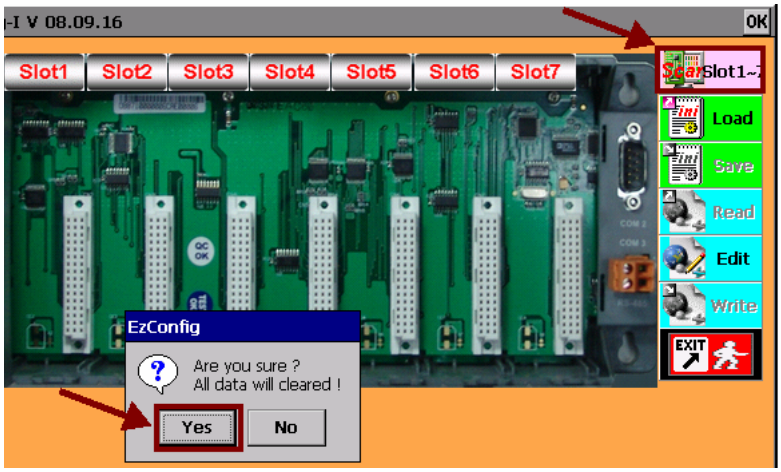
所有的專案執行前一定要先插入IO模組與做過EzConfig 的設定並儲存到控制器中，否則專案初始化時會發生錯誤，無法執行，所以新控制器或IO模組有變更時要先執行過才行。

2.1 首先做 EzConfig IO 規劃

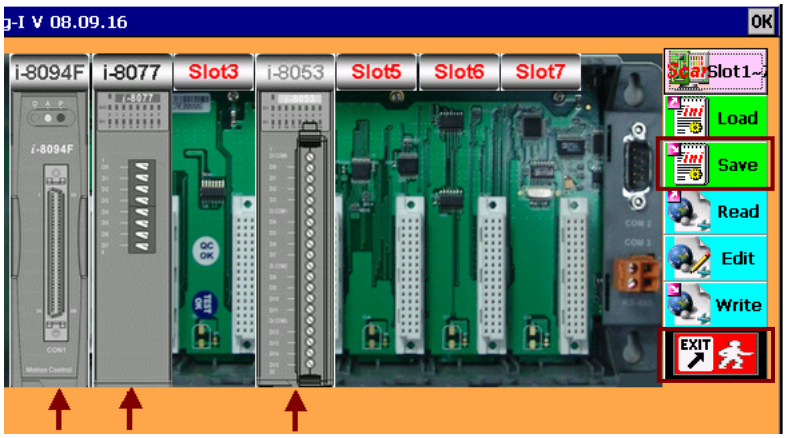
從"開始" =>"程式集"中啟動 EzConfig 如下圖:



按右上角 SCAN Slot1~7 按鈕開始自動掃瞄 IO 模組，按"是:開始 如下圖:



SCAN 完成後如下圖:



做完簡易規劃完成後按”Save”按鈕儲存規劃設定資訊。
再按 “EXIT”離開 EzConfig

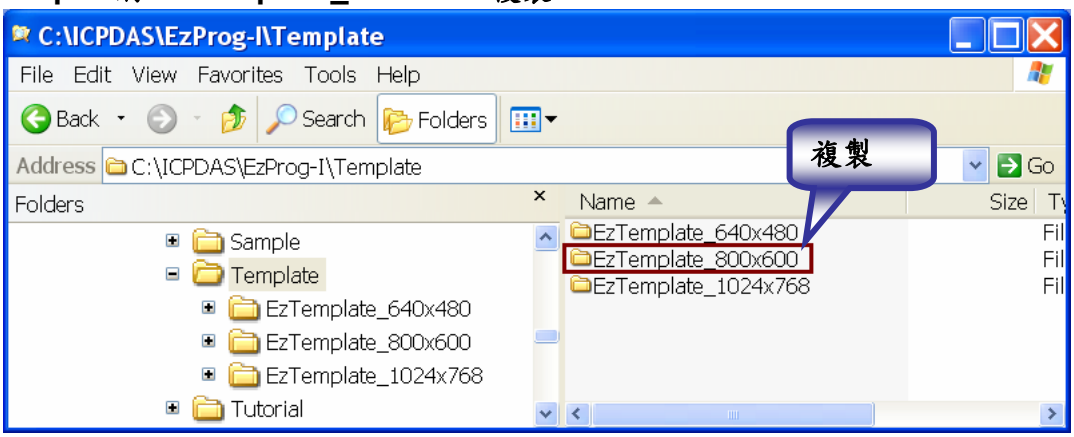
2.2 範例 Hello World

本範例是第一個範例主要再說明如何用,EzTemplate 快速建立專案。

2.2.1 複製開發樣本與修改

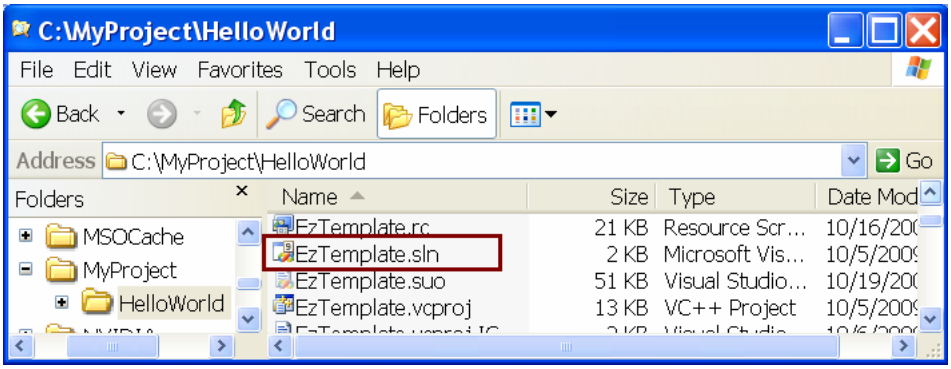
請複製 C:\ICPDAS\EzProg-I\Template\EzTemplate_800x600 到您專案要放置的路徑，並將名稱改為 HelloWorld(您專案的名稱)如下圖:

Step1. 將 EzTemplate_800x600 複製

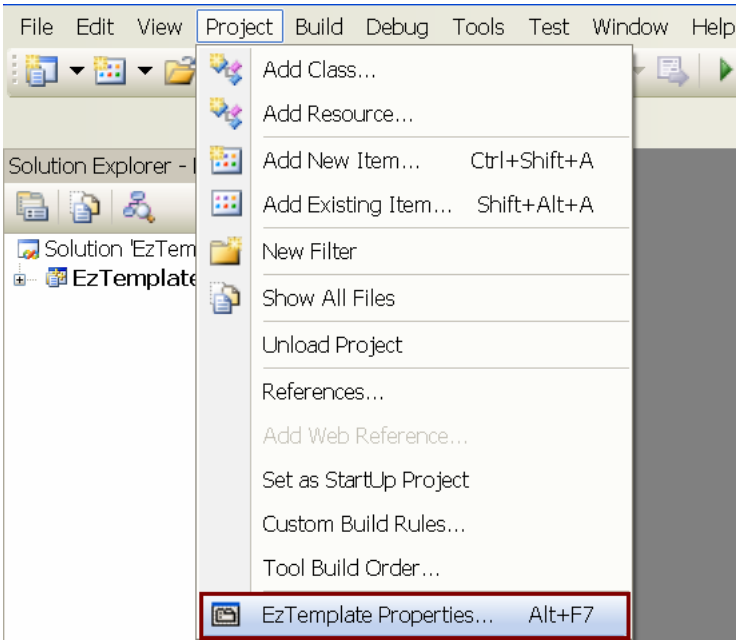


Step2.重新命名資料夾為"Hello World"

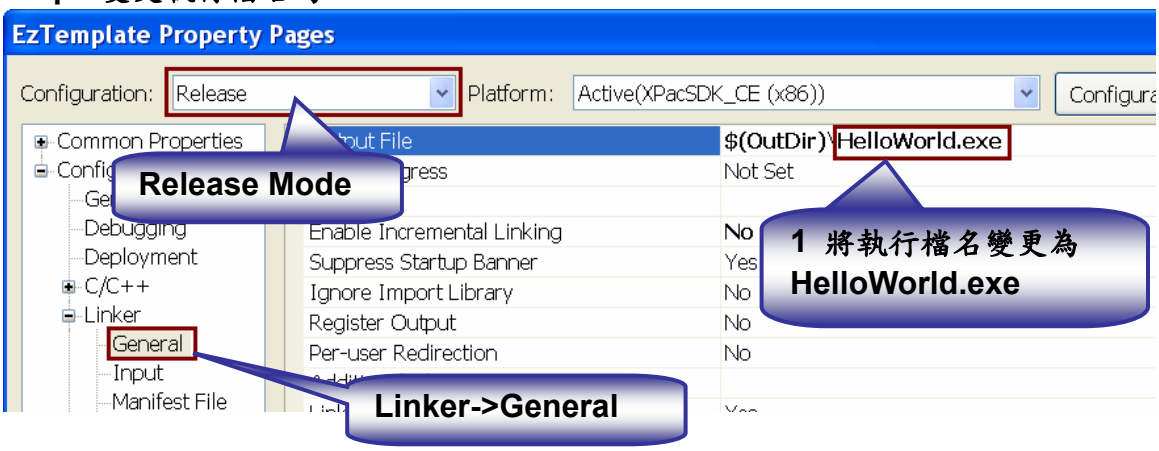
Step3.用 Visual Studio 2008 開啟 HelloWorld\Eztemplate.sln 專案檔如下圖:



Step4.專案開啟後請選擇 Project => EzTemplate Properties… 如下圖:



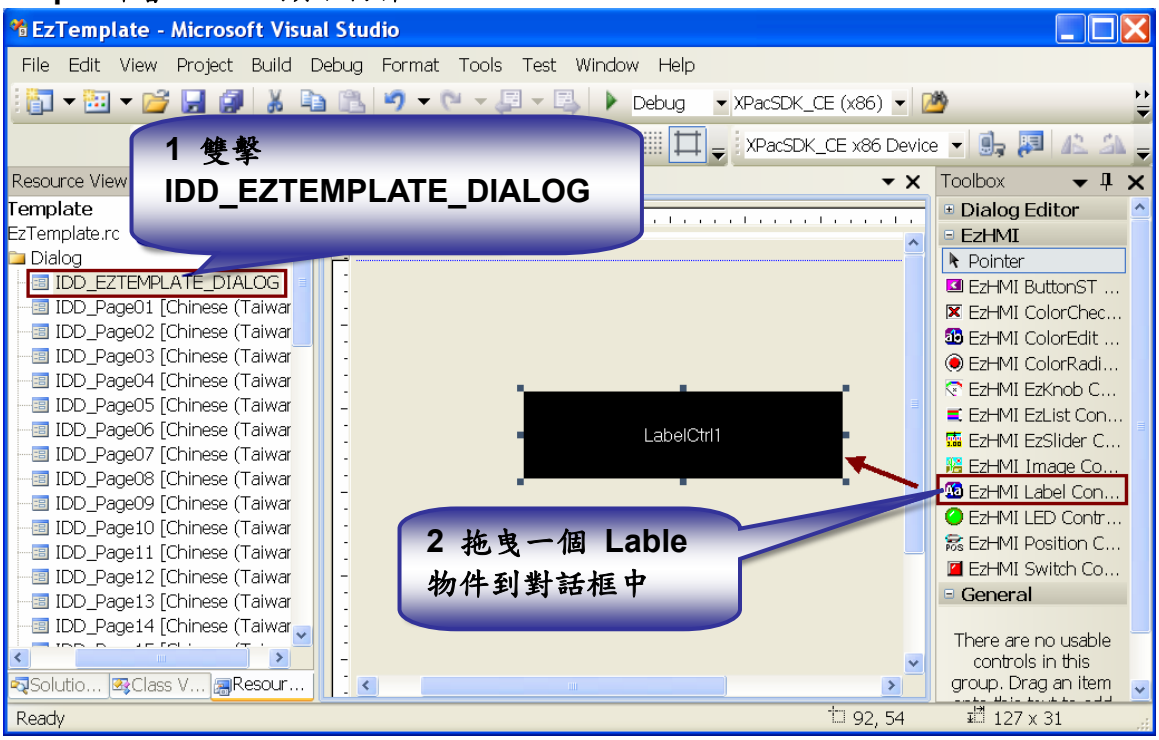
Step5.變更執行檔名為 HelloWorld.exe



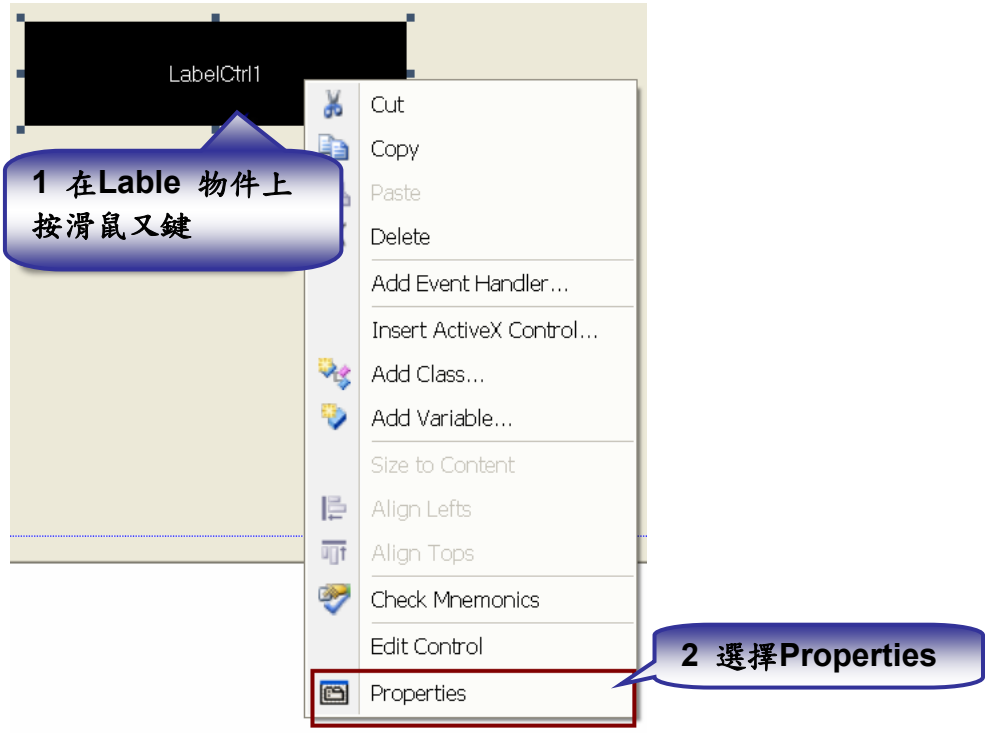
如此已完成樣本的修改，本範例為修改發行模式(Release Mode)，如要使用除錯模式(Debug Mode)請依同樣方法變更之。

2.2.2 設計與編譯

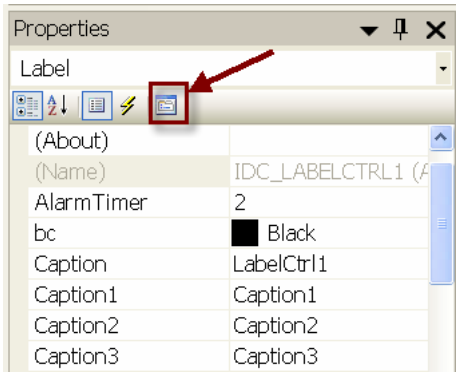
Step1.新增 Lable 顯示物件:



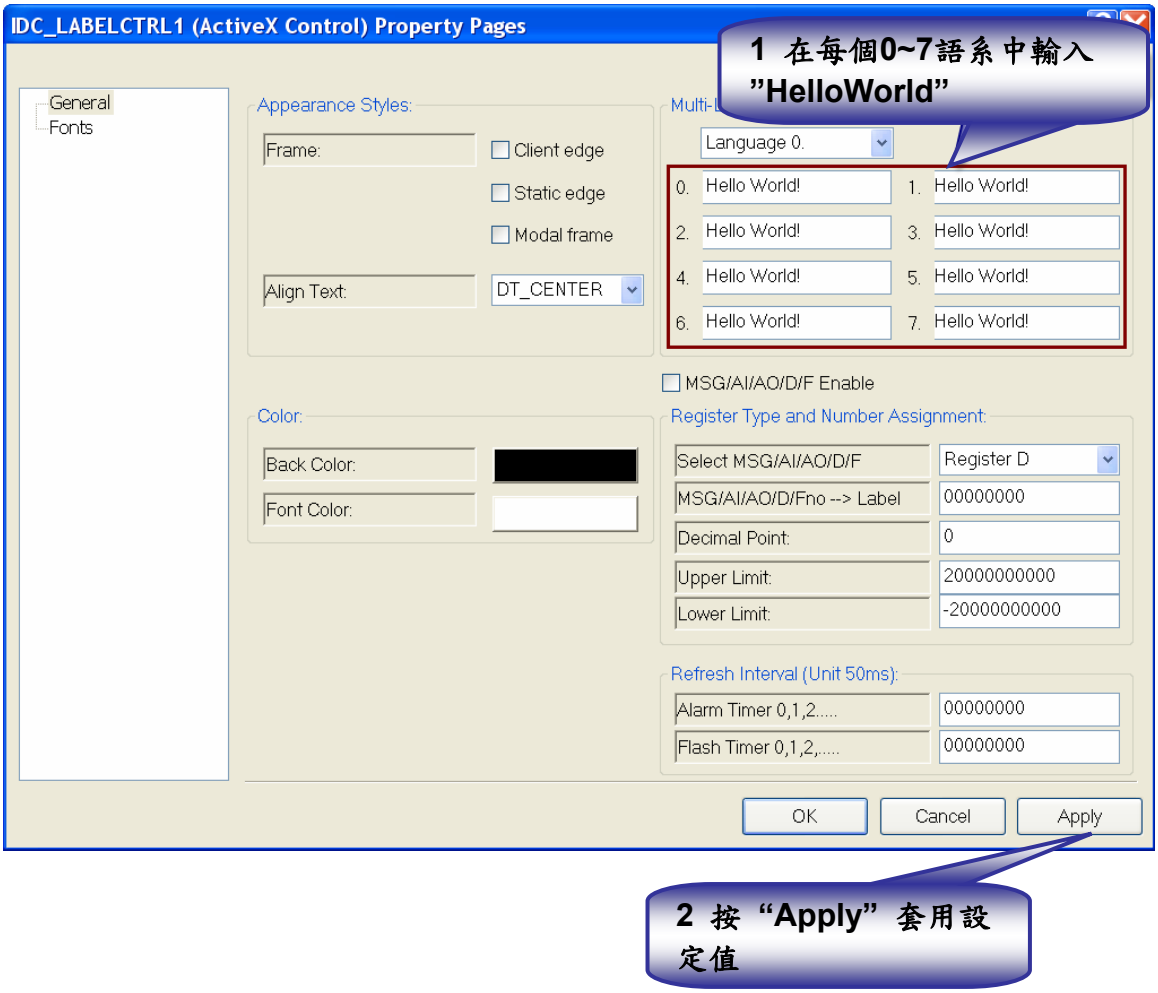
Step2.開啟 Lable 物件屬性設定頁:



Step3.點選“*Properties Pages*” 圖示



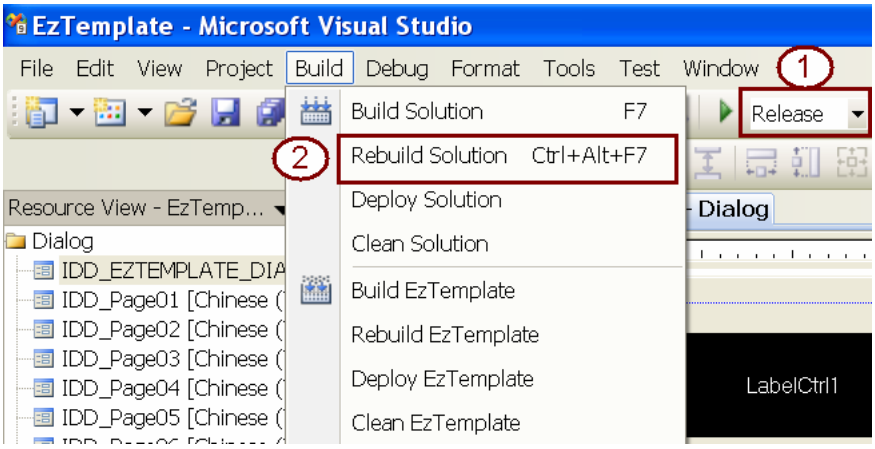
Step4.開啟 Lable 物件屬性設定頁，並設定顯示文字為”HelloWorld”：



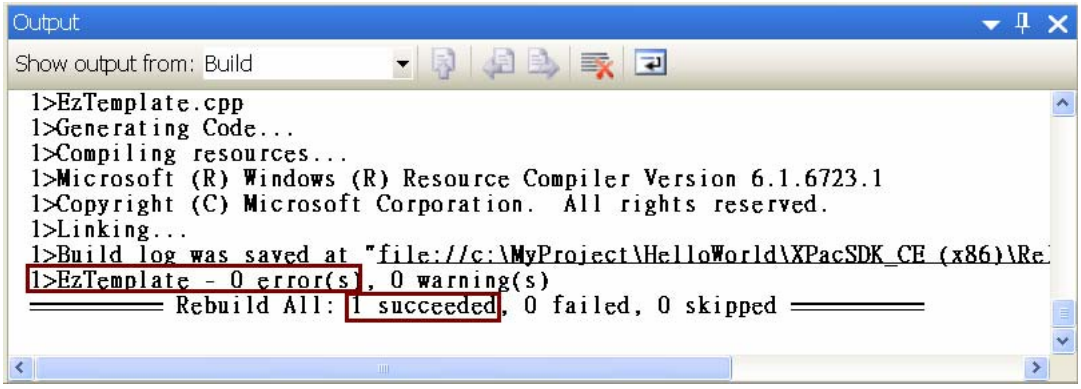
2.2.3 編譯專案為執行檔:

Step1.把編譯選項設定選為 Release 模式

Step2. 建立執行檔 Build->Rebuild Solution

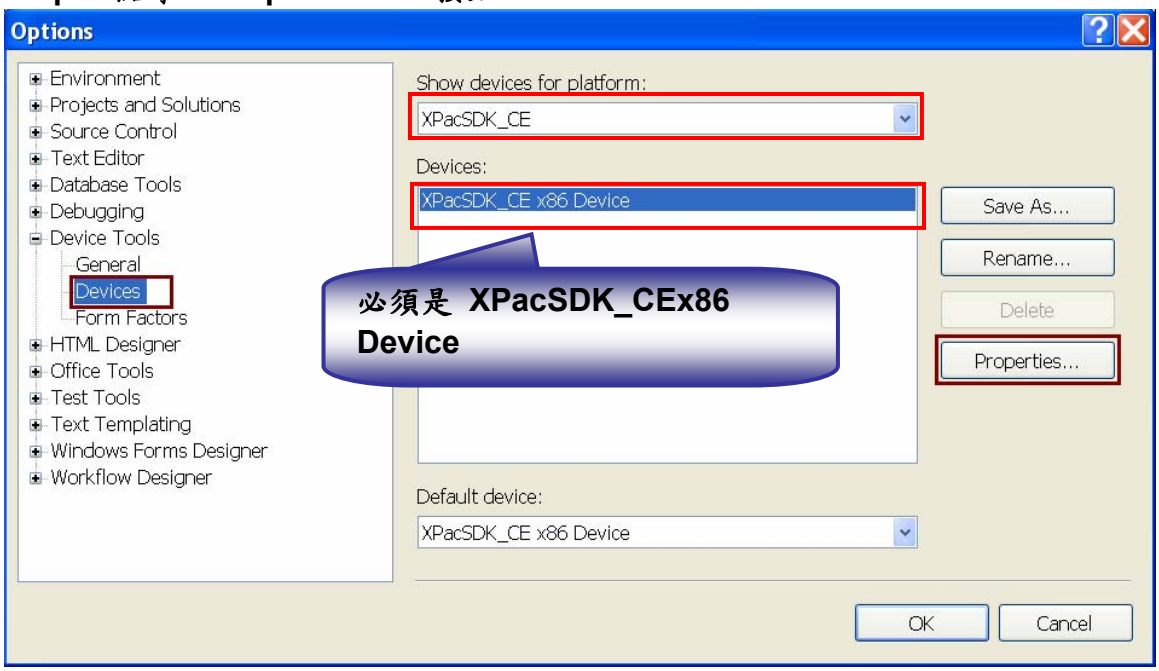


在 Ouput 視窗可以看到成功建立的資訊

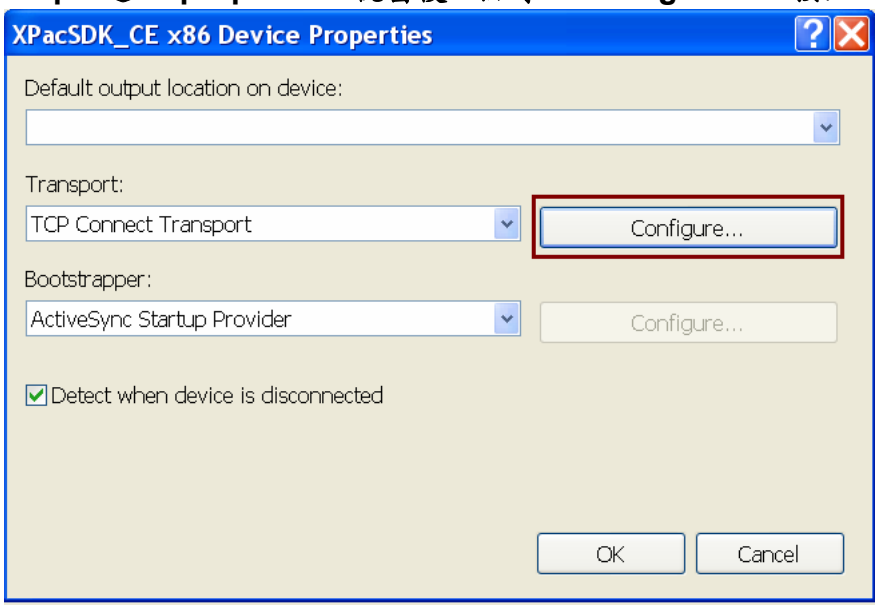


2.2.4 與 PAC 連線及下載程式

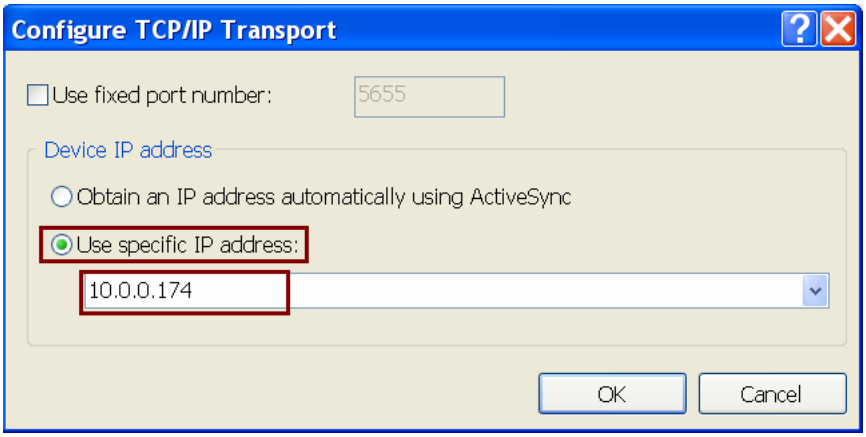
Step1. 從上方功能表點選 Tool-> Options…
Step2. 在 Options 視窗點擊展開”Device Tools”然後再點選“Devices”
Step3. 點擊 “Properties…” 按鈕



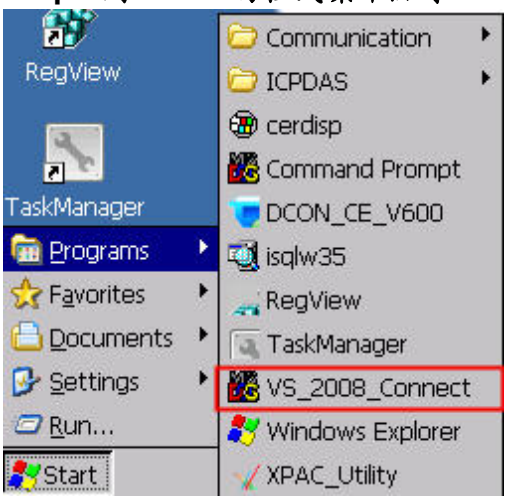
Step4.進入 properties 視窗後，點擊 “Configure…” 按鈕



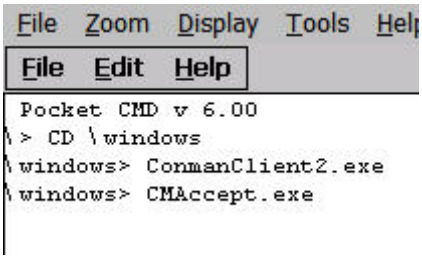
Step5. 請選擇 “Use specific IP address”，並在下面打上 PAC 的 IP 位址



Step6.到 PAC 的程式集中點擊 “VS_2008_Connect”



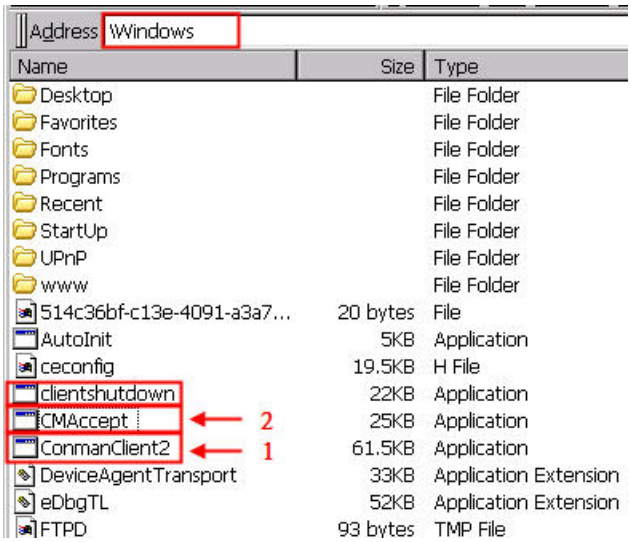
然後會出現以下畫面，表示等待 PC 連線



或是：

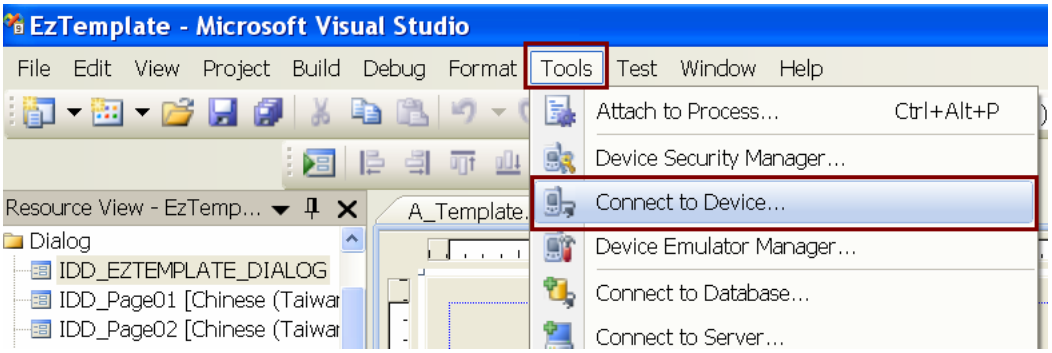
到 **PAC** 的 ***\Windows*** 資料夾內，依順序點擊下列兩個檔案

- ***conmanclient2.exe***
- ***cMaccept.exe***

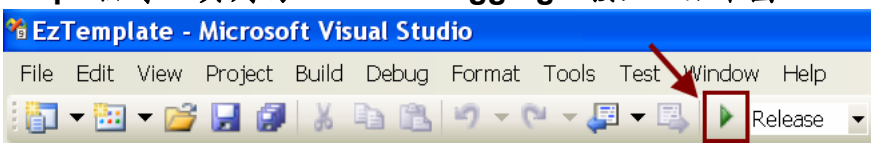


若是有 PC 無法連線的情況請先執行 “clientshutdown” 再依序執行 1，2 兩個檔案

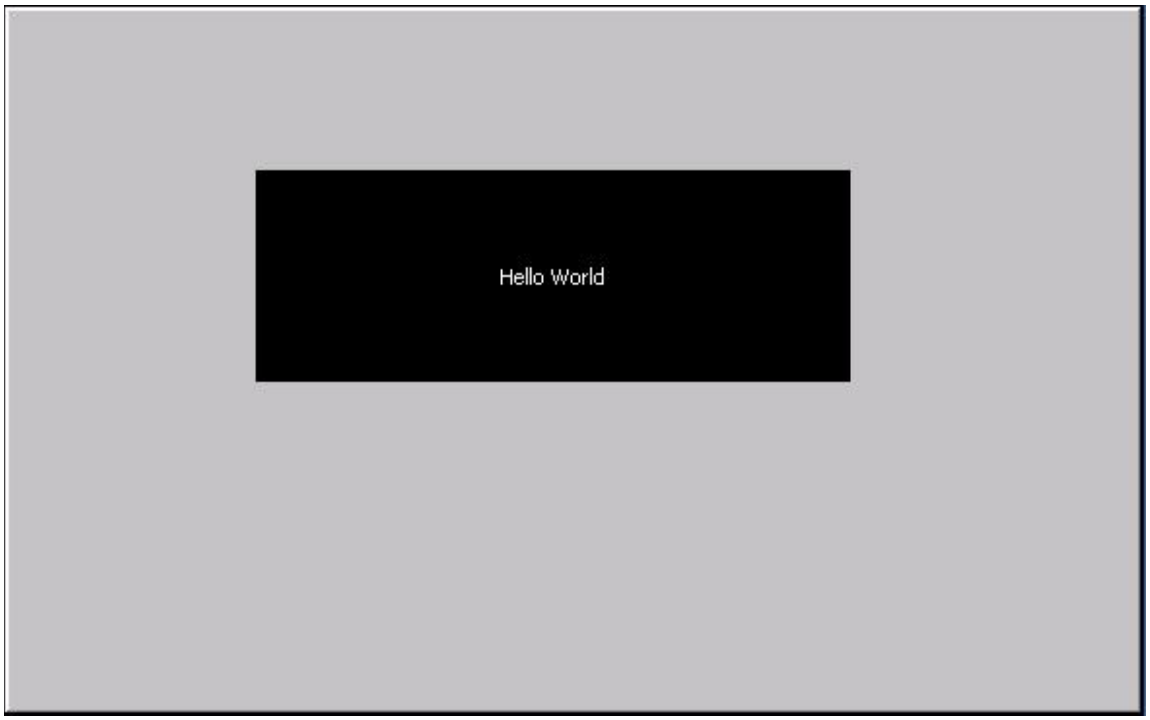
Step7. 點擊 Visual Studio 2008 功能表中的 Tools -> Connect to Device… 做連線的動作



Step8.點擊工具列的"Start Debugging" 按鈕，如下圖



如果設定與連線沒問題的話，在 PAC 端即可看到執行畫面如下圖



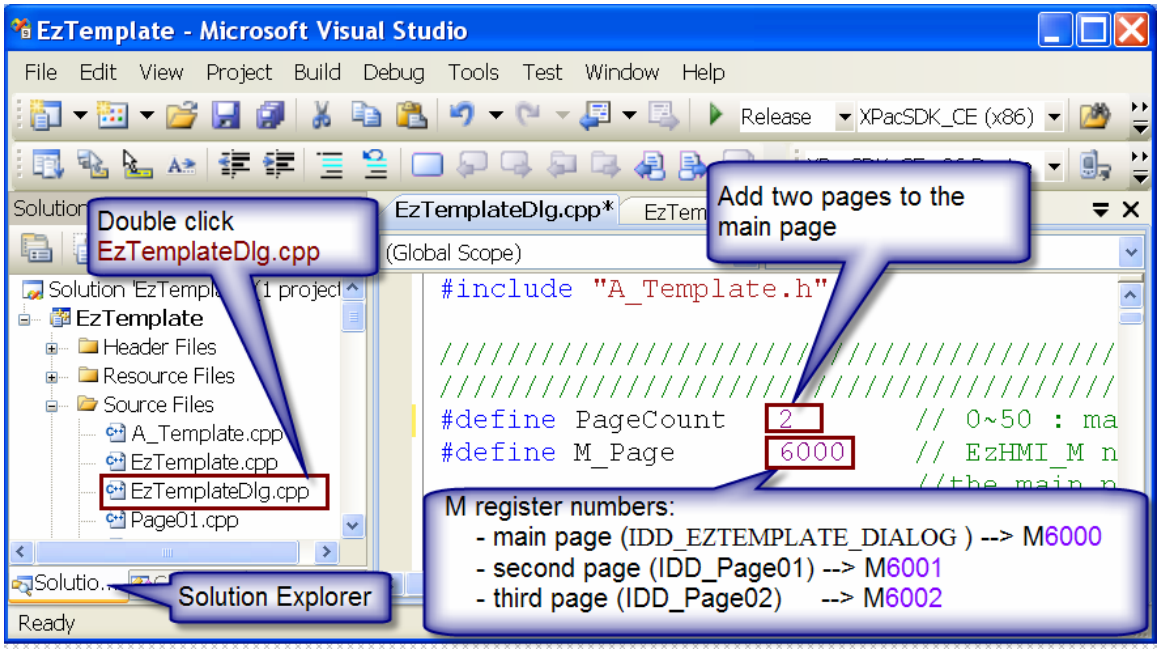
2.3HMI Page

本範例是以三個 HMI 頁面應用範例，主要在說明如何用 EzTemplate 專案範本快速設計建立多操作頁面專案，並可以在頁面間切換。

請依照 2.2.1 複製樣板專案，請名稱改為 HMI_page,執行檔為 HMI_Page.exe，並開啟之。

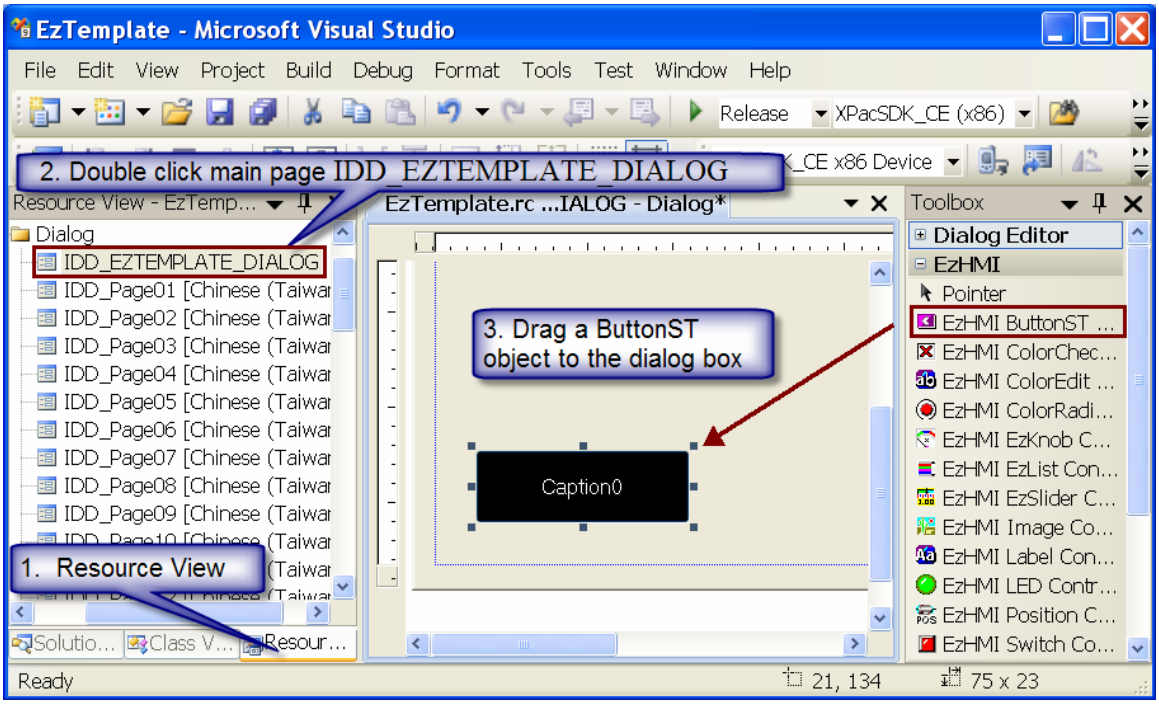
2.3.1 系統 Page 設定

請在專案管理選 Solution，並在 Source File 中雙擊 EzTemplateDlg.cpp 開啟文件編輯它的內容，如下圖。

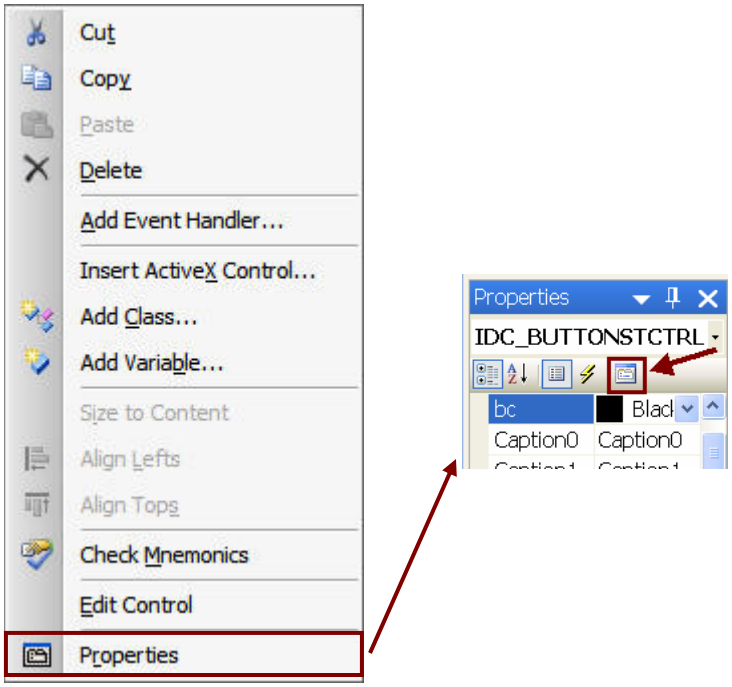


2.3.2 設計與編譯

Step1.請在專案管理選 Resources
Step2.雙擊 IDD_EZTEMPLATE_DIALOG 開啟畫面
Step3.在畫面上放一個 ButtonST 物件

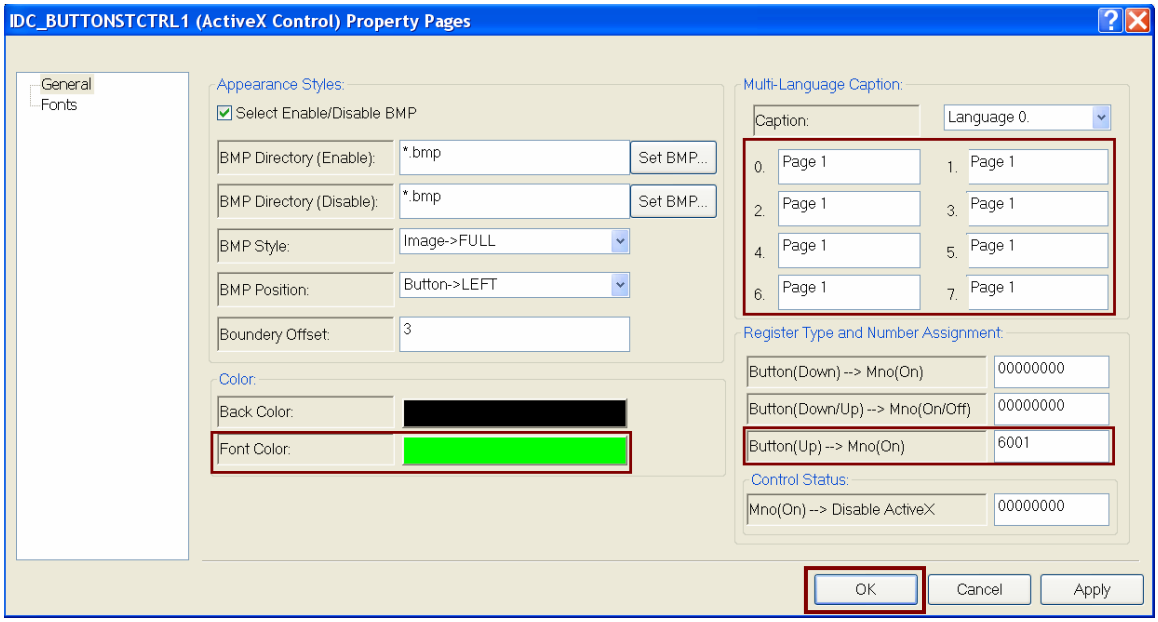


Step4.在 ButtonST 上按右鍵，點選 Properties 並開啟 “Property Pages”



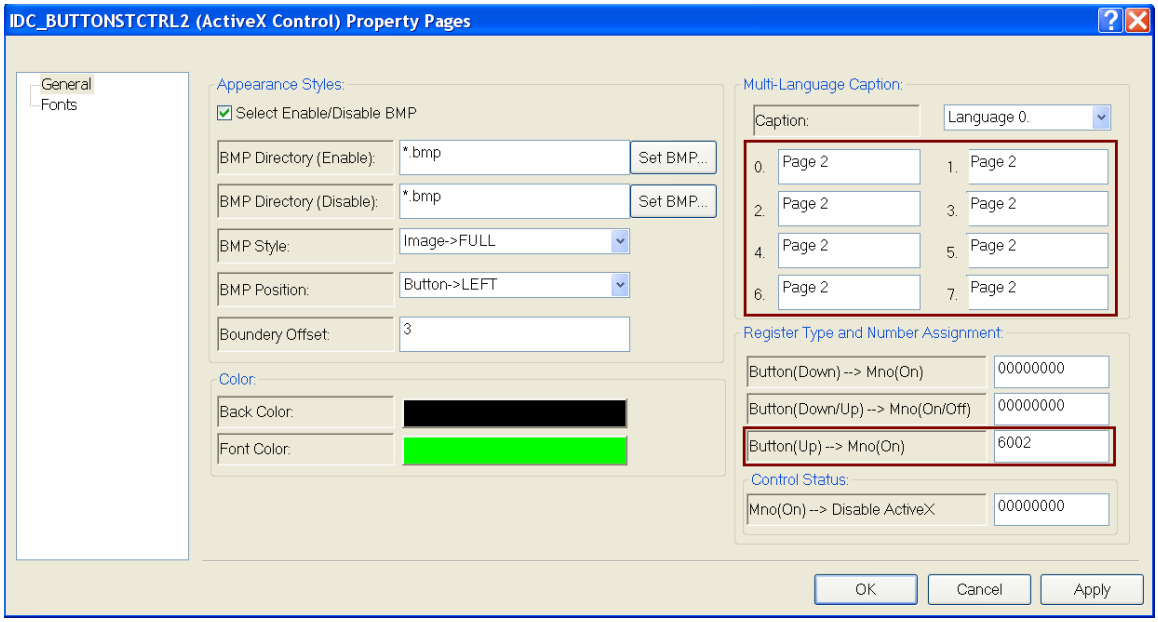
Step5.設定屬性如下，然後按 “OK” 按鈕

- **Caption : Page1**
- **Button(Up)→Mno(On) : 6001**
- **Font Color :**



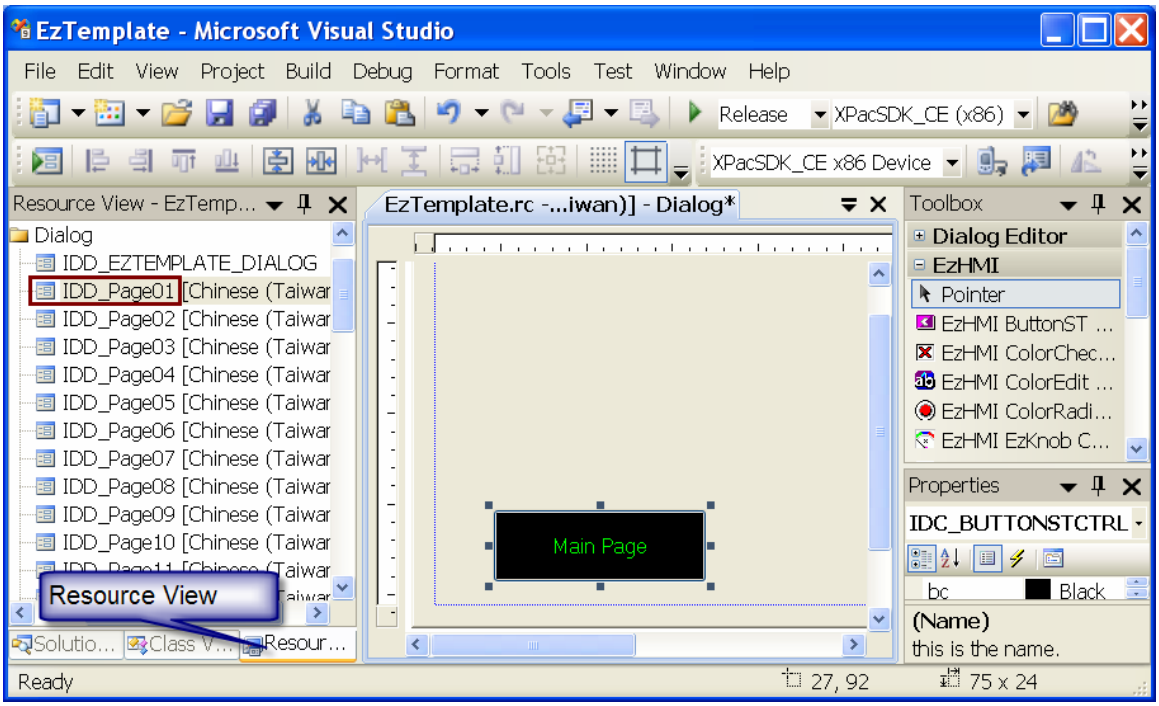
Step6.選擇第一個 ButtonST，並 copy(Ctrl+C)與 Paste(Ctrl+V)做第二個 ButtonST，如第一個方法，將其 Properties 設定如下:

- **Caption : Page2**
- **Button(Up)→Mno(On) : 6002**
- **Font Color :**



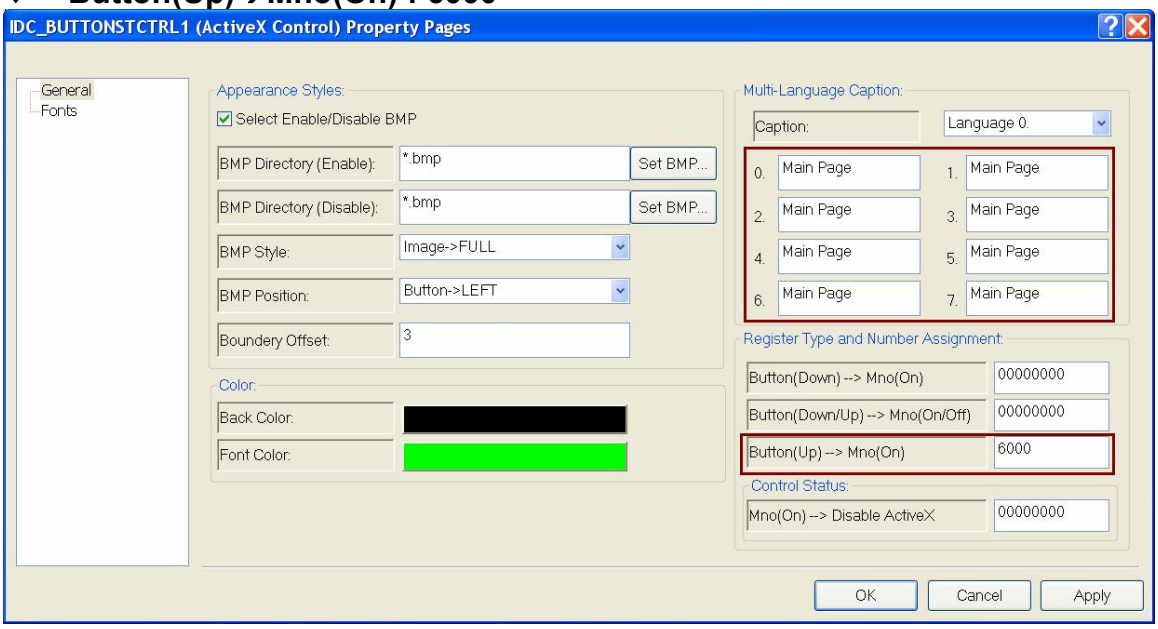
設定完成後畫面如下圖

Step7.選擇一個 ButtonST 並複製(copy) ，然後貼上(paste) 在 Page01 建立回主頁的按鈕:



將其 **Properties** 設定如下:

- **Caption : Main Page**
- **Button(Up)→Mno(On) : 6000**



Step8.將 Step7. 做好的按鈕複製(copy) 並貼上(paste)到 Page2 (IDD_Page02)

Step9.然後編譯專案為 HMI_Page.exe，或按快速功能鍵 F5 編譯下載與執行。按 Page 1，Page 2 ，main Page 就可以輕易切換頁面。

2.4 範例 Multi-language

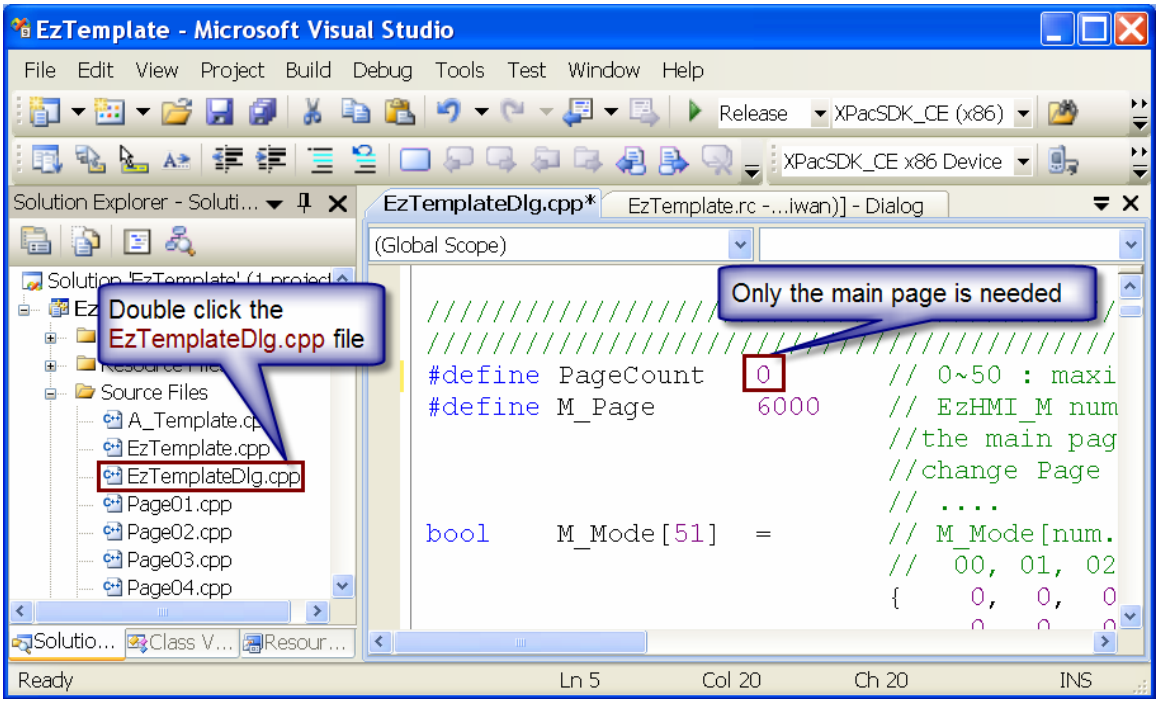
本範例是以一個 HMI 頁面應用範例，主要在說明如何用 EzTemplate 專案範本設計建立多國語系操作頁面專案，透過 ColorRadio 物件直接變更 D8000 暫存器值(0~7)，就可以變更顯示語言為 0~7，先準備文字如下:

0.English	**multi-language demonstration**
1.繁體中文	多語言的示範
2.简体中文	多语言的示范
3.日文	**multi-language** デモンストレーション
4.German	**mehrsprachige Demonstration**
5.Spanish	**demostración multilingue**
6.Р у с с к о	**multi-language демонстрация**
7. Portugese	**demonstração multilíngue**

請依照 2.2.1 複製樣板專案，請名稱改為 Multi_language,執行檔為 Multi_language.exe，並開啟之。

2.4.1 系統 Page 設定

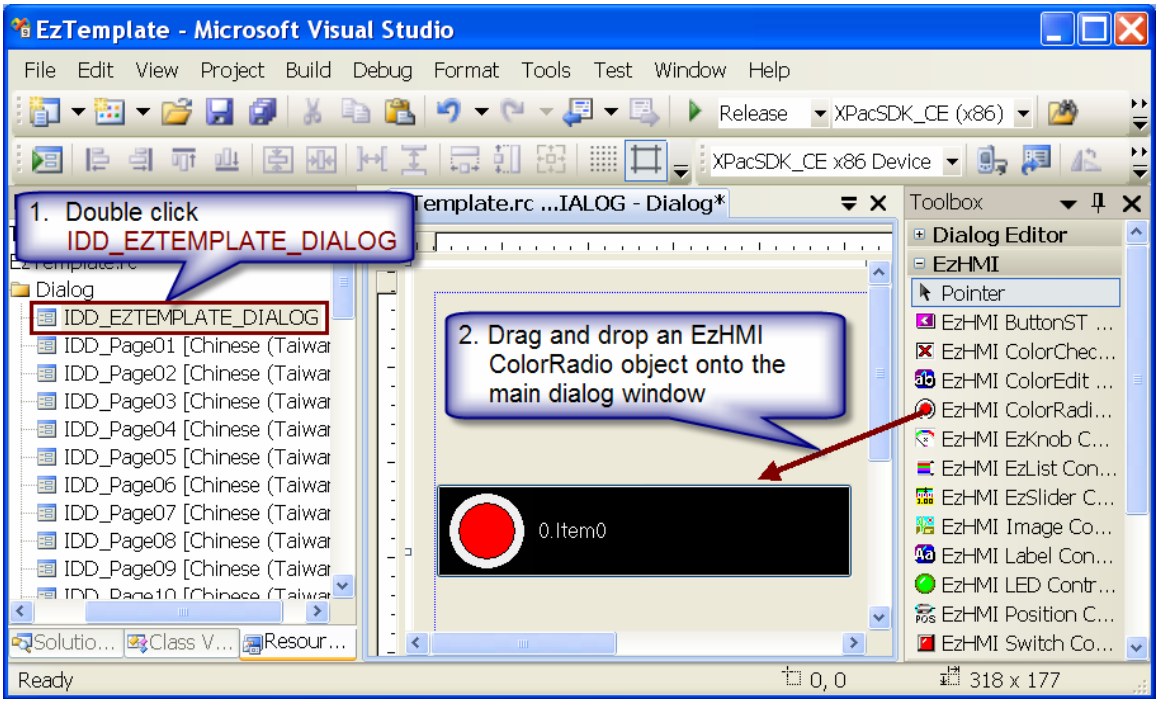
請在專案管理選 Solution，並在 File 中選 EzTemplateDlg.cpp，並做設定。



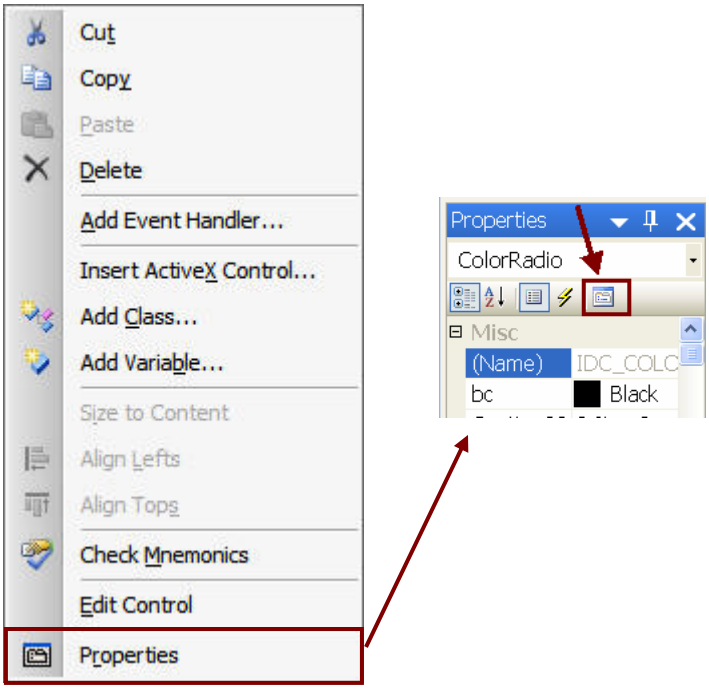
2.4.2 設計與編譯

Step1.雙擊 IDD_EZTEMPLATE_DIALOG 開啟畫面

Step2.拖曳一個 ColorRadio 物件到畫面上



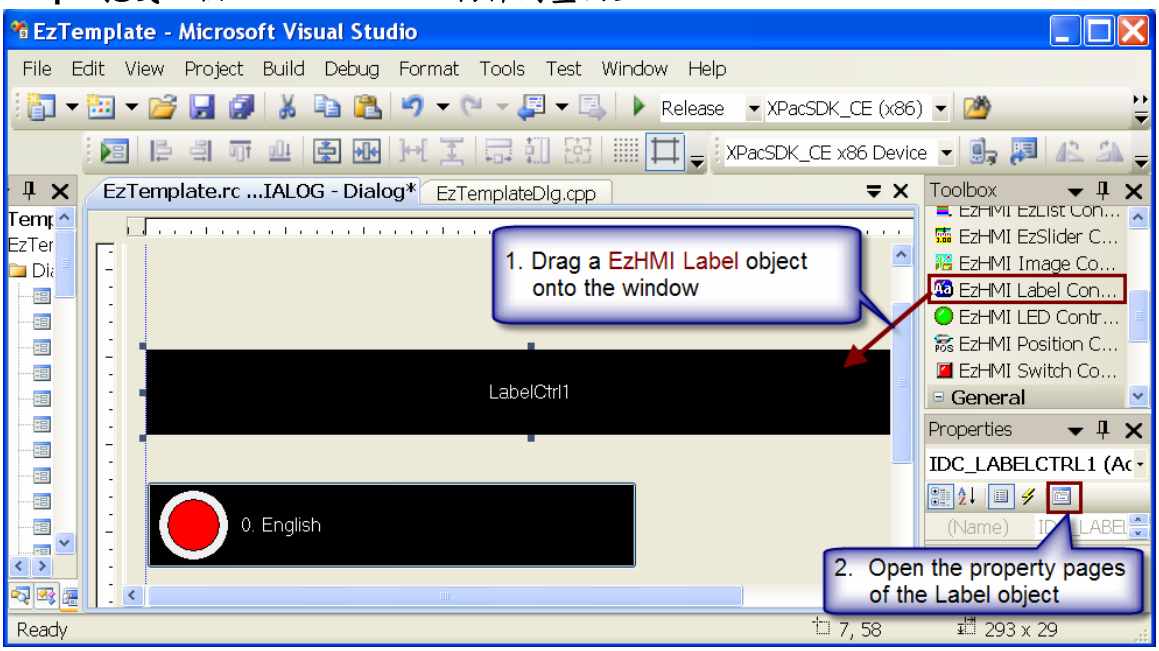
Step3. 選取 ColorRadio 並按右鍵開啟 ColorRadio 的 Properties page



Step4.設定 ColorRadio 的 Properties ，如下圖：

- **Muilt Language Caption: Language 0**
- **List Item： 照一開始準備好的文字標題依順序複製並貼上去**
- **Radio Select→Dno : 8000**

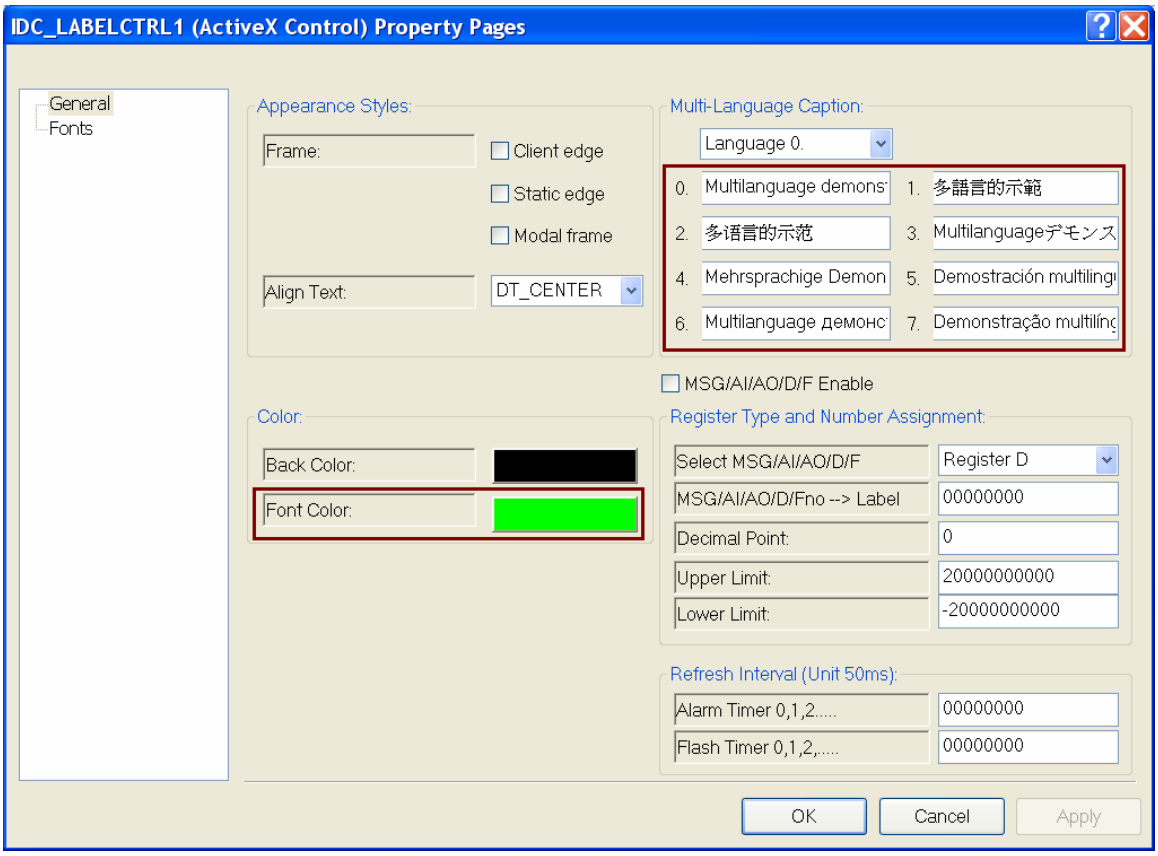
Step5.拖曳一個 EZHMI Label 物件到畫面上



Step6.設定 Label Properties

Multi-Language Caption 0~7: 照之前準備好的文字內容依順序貼上去

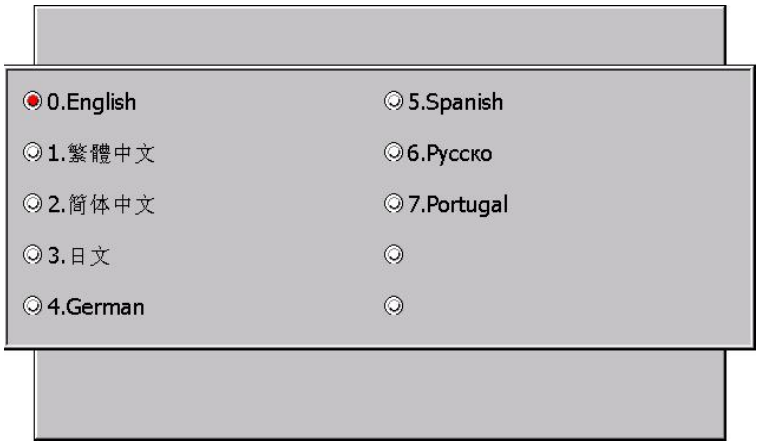
Font Color :



Step7.編譯專案為 Multi_language.exe，或按快速功能鍵 F5 編譯下載與執行。

執行如下圖:


multi-language demonstration
0.English


0.English
1.繁體中文
2.簡体中文
3.日文
4.German
5.Spanish
6.Русско
7.Portugal


多語言的示範
1.繁體中文

2.5HMI DIO

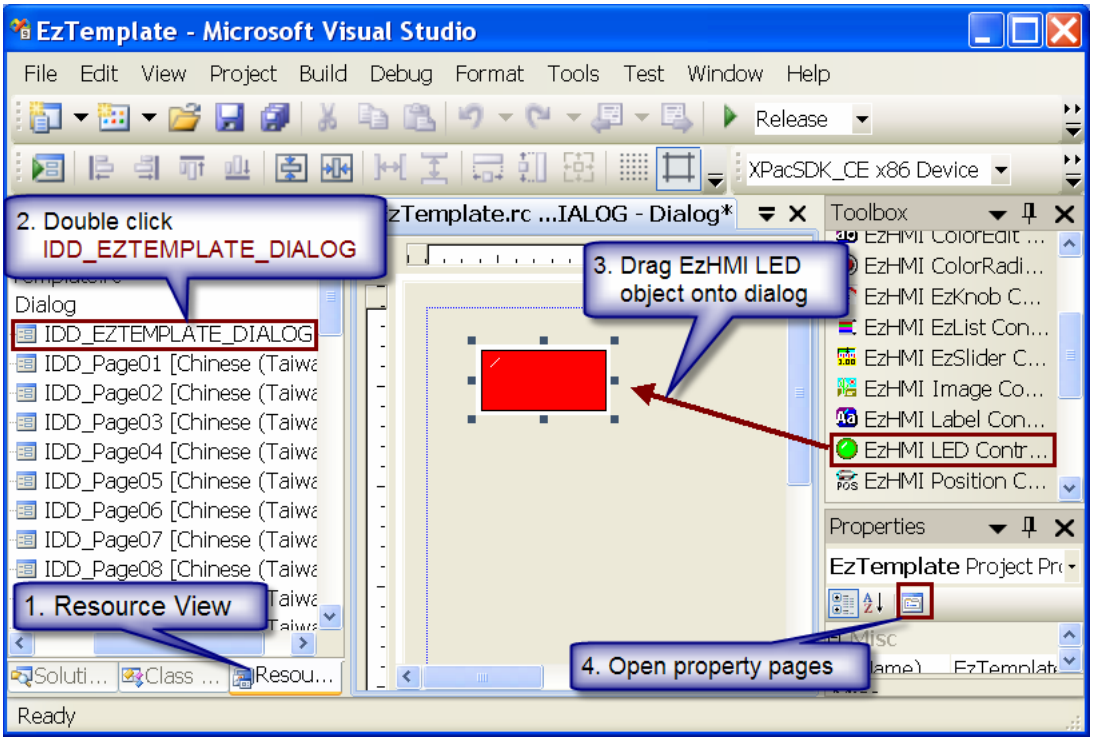
本範例是以一個 HMI 頁面應用範例，主要在說明如何用 EzTemplate 專案範本設計建立 DIO 操作專案。

請依照 2.2.1 複製樣板專案，請名稱改為 HMI_DIO,執行檔為 HMI_DIO.exe，並開啟之。

2.5.1 HMI 物件畫面控制設計

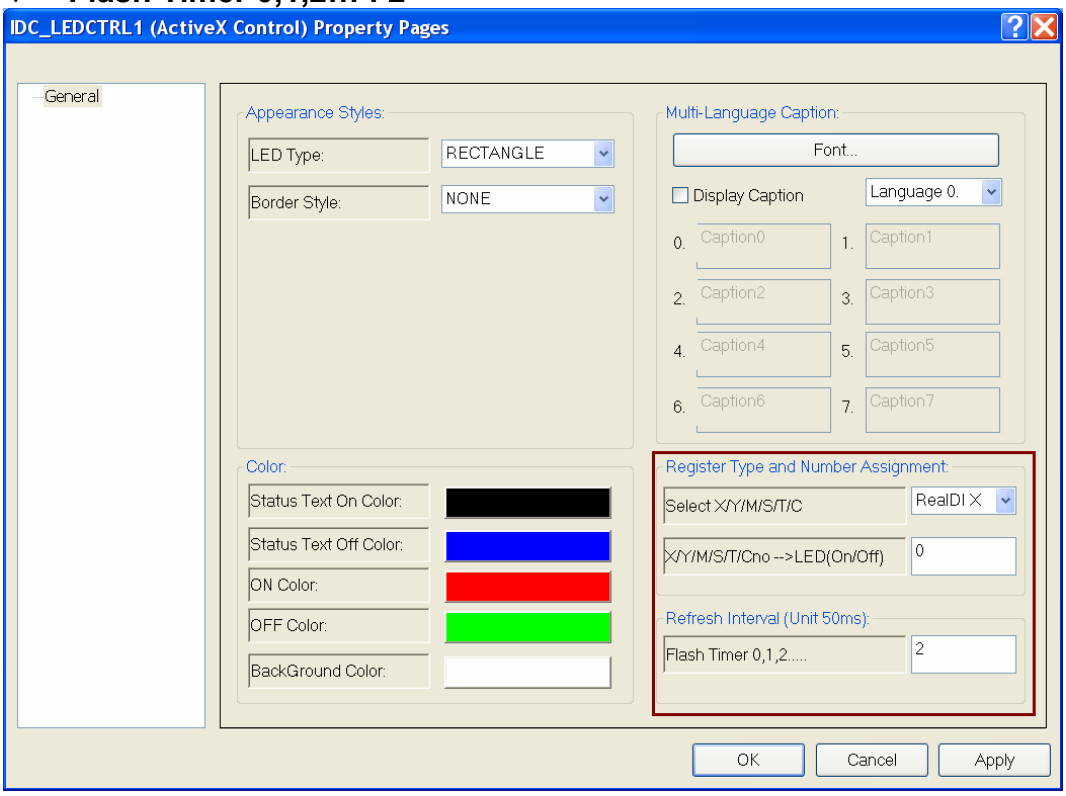
Step1.雙擊 IDD_EZTEMPLATE_DIALOG 開啟畫面

Step2.拖曳一個 LED 物件到畫面上，並開啟 Properties page

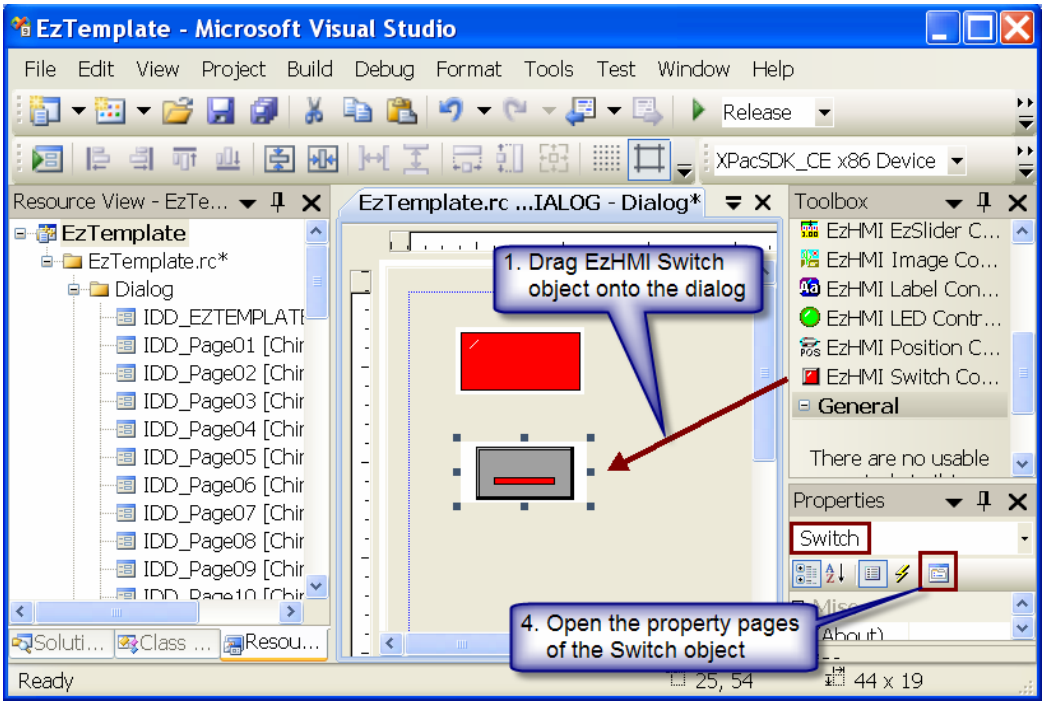


Step3.設定 LED 屬性

- **Select X/Y/M/S/T/C : RealDI X**
- **X/Y/M/S/T/Cno→LED(On/Off) : 0**
- **Flash Timer 0,1,2… : 2**

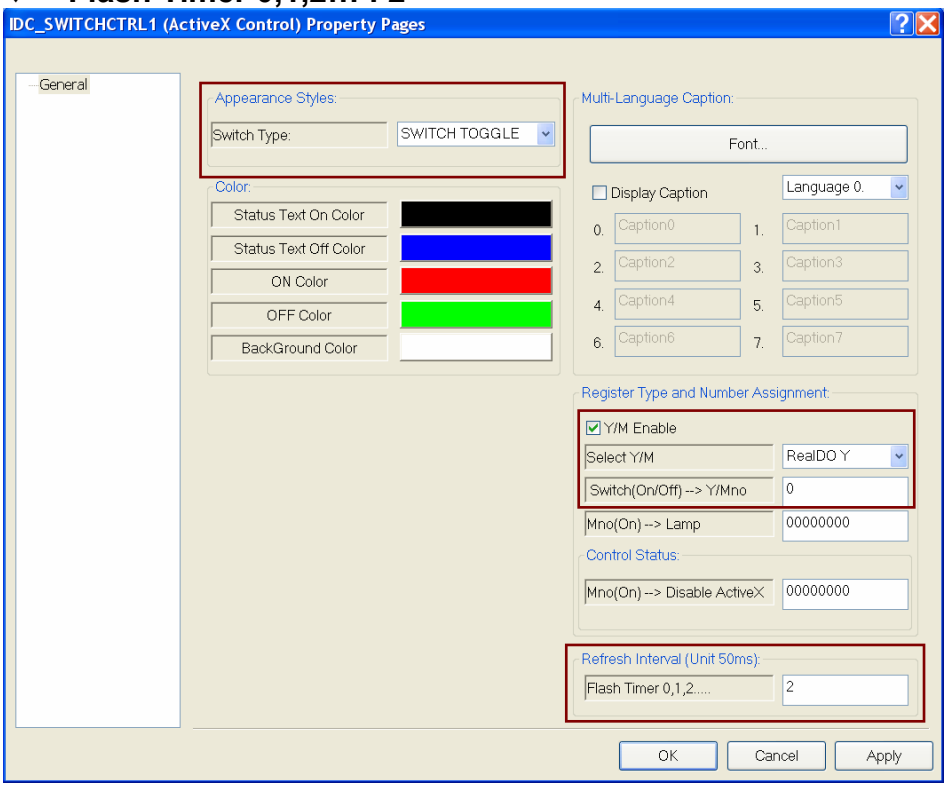


Step4. 拖曳一個 Switch



設定屬性:

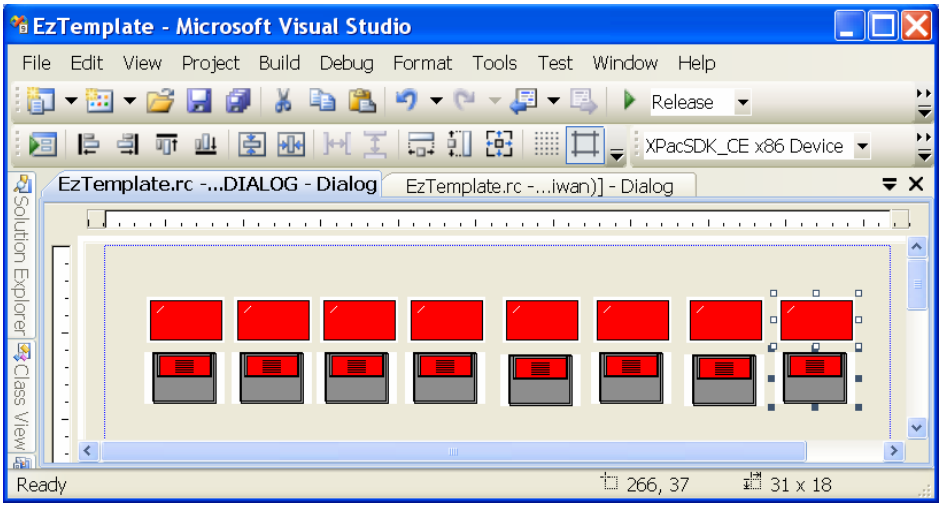
- **Switch Type : SWITCH TOGGLE**
- **Y/M Enable : 選取**
- **Select Y/M : RealDO Y**
- **Switch(On/Off)→Y/Mno : 0**
- **Flash Timer 0,1,2… : 2**



Step5.再利用滑鼠圈選第一個 LED 與 SWITCH 按快速鍵 Ctrl+C(copy) 再利用 Ctrl+V(Paste) 作出其他 7 組共 8 組:

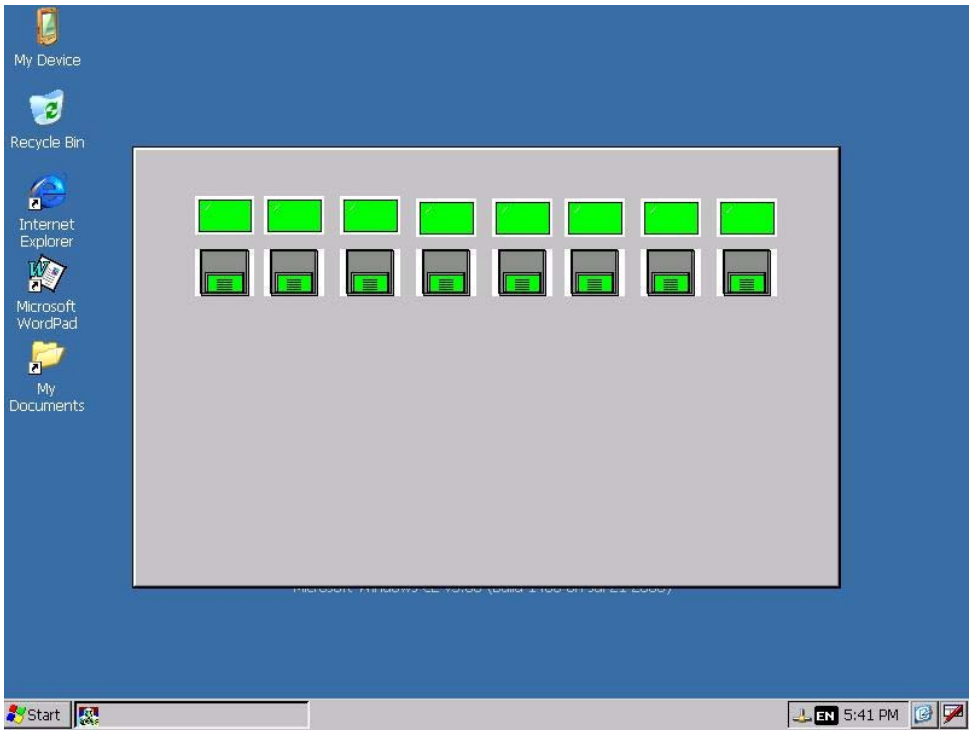
Xno 輸入 ”01” ，”02” ，”03” ，”04” ，”05” ，”06” ，”07”

Yno 輸入 ”01” ，”02” ，”03” ，”04” ，”05” ，”06” ，”07”



2.5.2 專案啟始與測試

按熱鍵 F5 連線下載執行:(請先確認網路設定是正確的)，連線除錯方法請參閱附錄連線除錯方法,如果沒有其他異常，在 PAC 畫面出現如下對執行畫面，你可以看到 LED 輸入 X 0 狀態，你也可以按 Switch 改變 Y 0 輸出狀態。



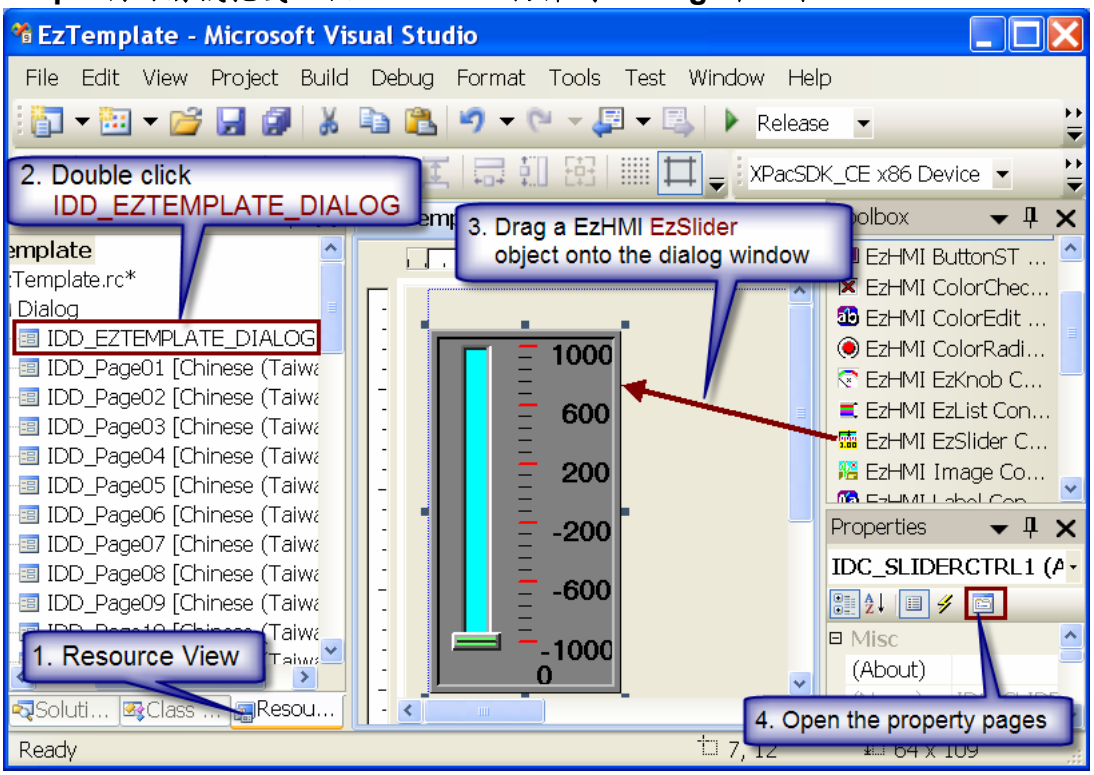
2.6HMI AIO

本範例是以一個 HMI 頁面應用範例，主要在說明如何用 EzTemplate 專案範本快速設計建立 AIO 操作專案。

請依照 2.2.1 複製樣板專案，請名稱改為 HMI_AIO,執行檔為 HMI_AIO.exe，並開啟之。

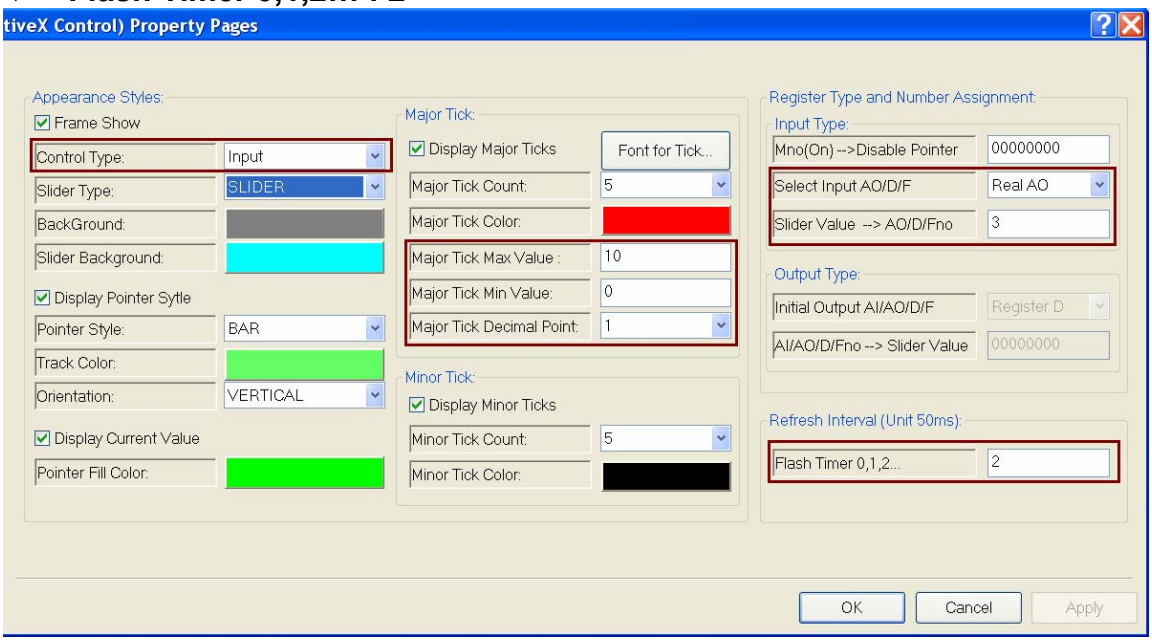
2.6.1 HMI 物件畫面控制設計

Step1.利用滑鼠拖曳一個 EzSlider 物件到 Dialog 中如下

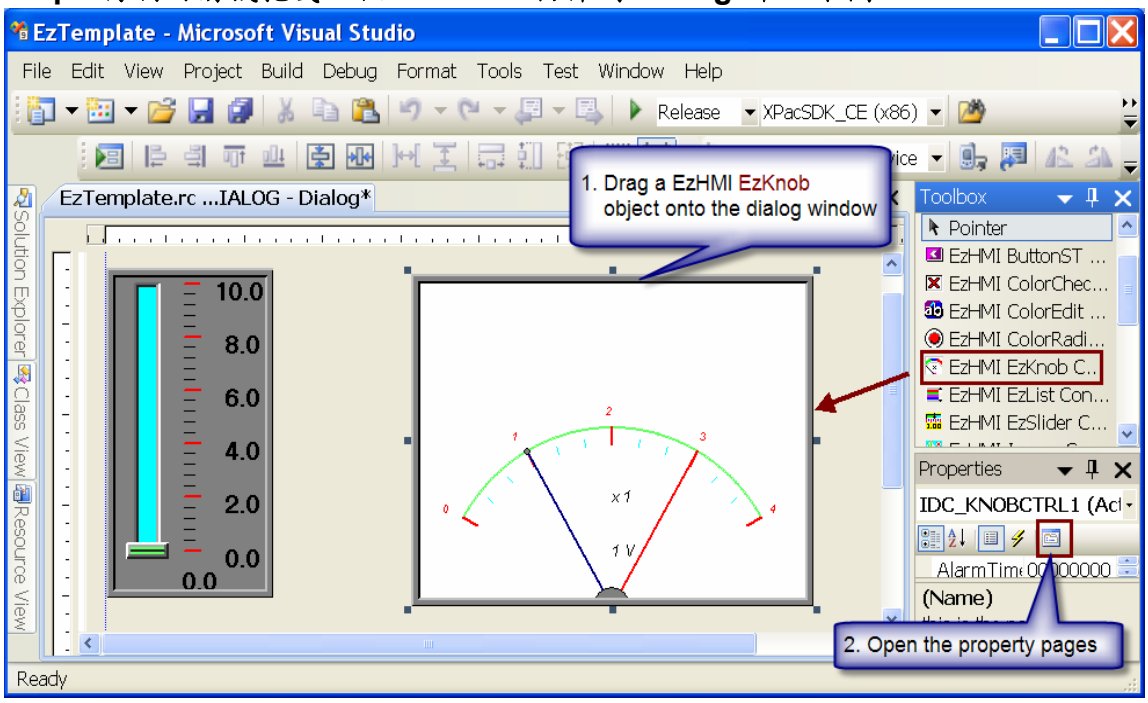


設定屬性:

- **Control Type : Input**
- **Major Tick Max Value : 10**
- **Major Tick Min Value : 0**
- **Major Tick Decimal Point : 1**
- **Select Input AO/D/Fno : 2**
- **Slider Value→AO/D/Fno : 3**
- **Flash Timer 0,1,2… : 2**

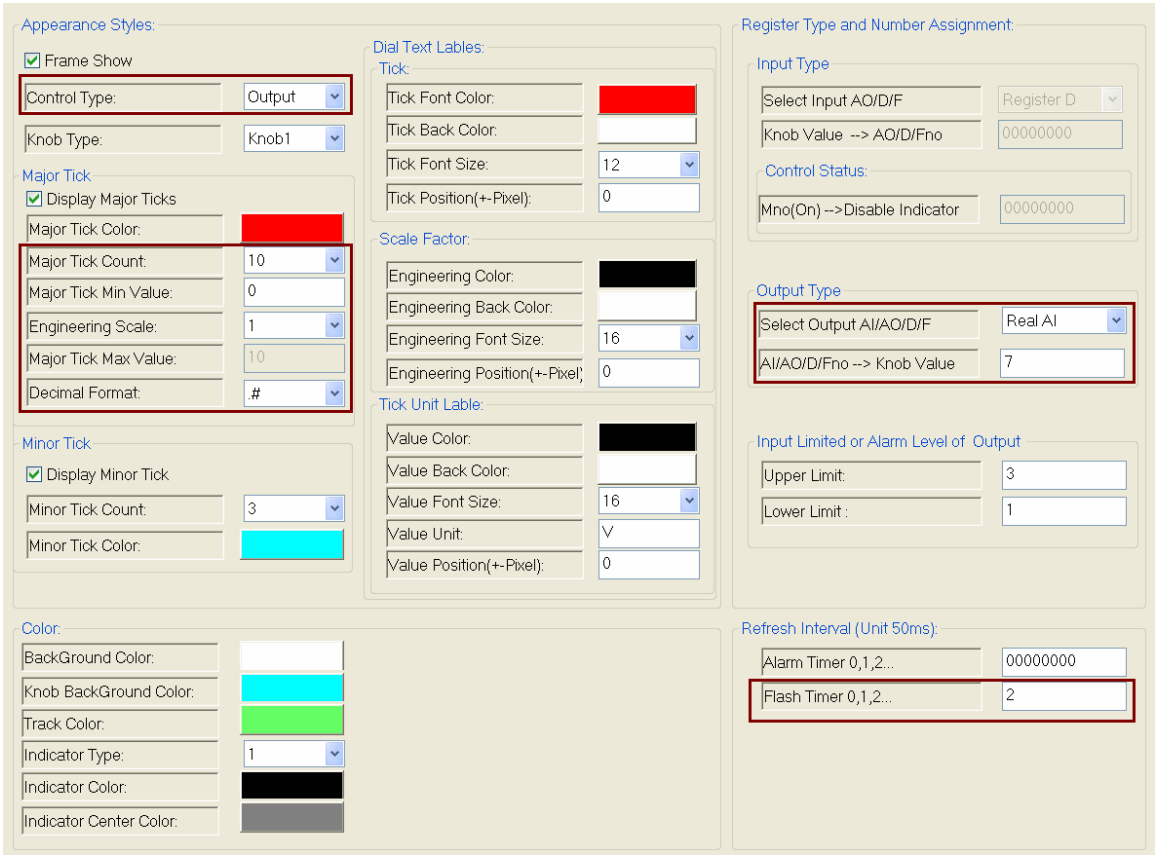


Step2.再利用滑鼠拖曳一個 EzKnob 物件到 Dialog 中如下圖



設定屬性:

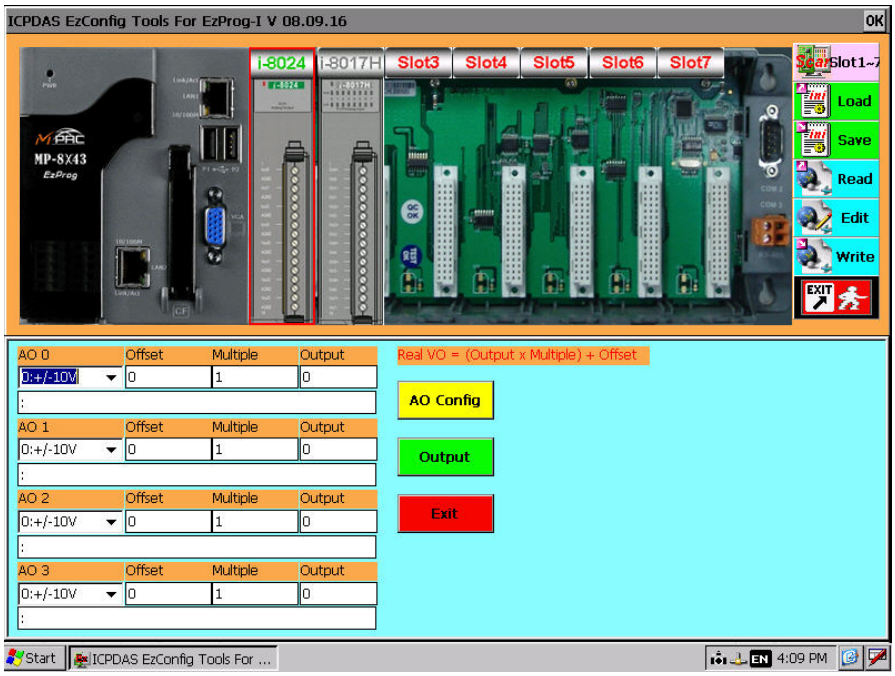
- **Control Type : Output**
- **Major Tick Count : 10**
- **Major Tick Min Value : 0**
- **Engneering Scale : 1**
- **Decimal Format : #**
- **Select Output AI/AO/D/F : Real AI**
- **AI/AO/D/Fno : 7**



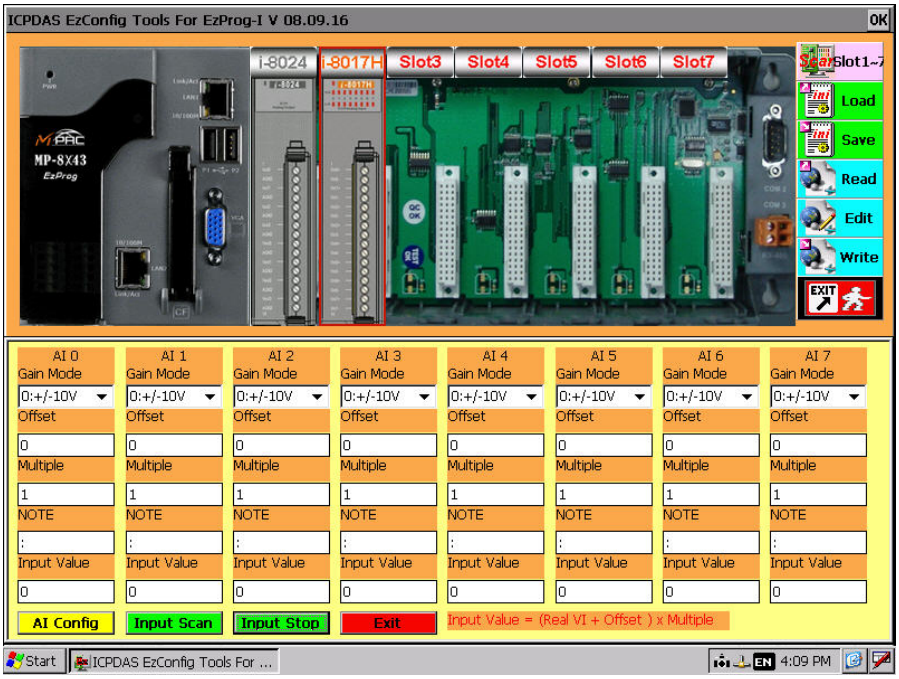
2.6.2 專案啟始與測試

在測試前請先將 I-8024 的 Vout3 與 I-8017H 模組的 Vin7 對接並使用 EzConfig 規劃並儲存，此範例 I-8024 與 I-8017H 模組照預設值儲存即可。

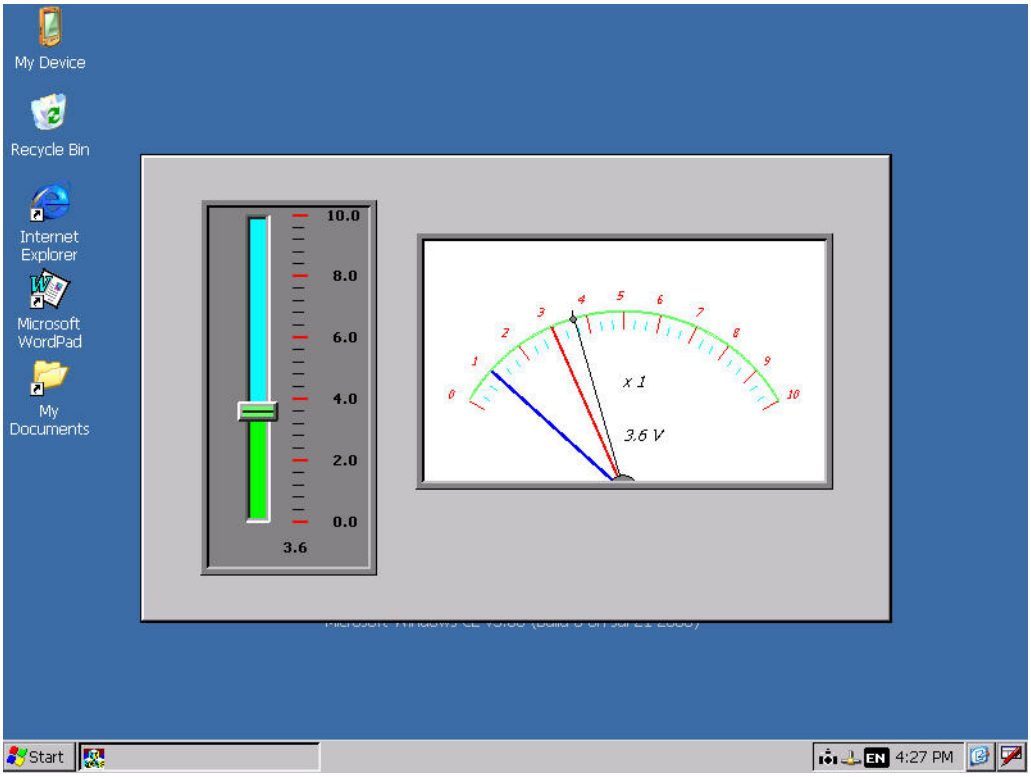
I-8024



I-8017H



按熱鍵 F5 連線下載執行:(請先確認網路設定是正確的)，請參閱 EzProg-I Getting Started 手冊中附錄的連線除錯方法說明。如果沒有其他異常，在 PAC 畫面出現如下對執行畫面，你可以看到當左邊的 EzSlider 控制 I-8024 模組的 channel Vout3 輸出電壓時，右邊的 EzKnob 也因讀取 I-8017H 的 channel Vin7 電壓值改變狀態。



2.7 範例 UserThread

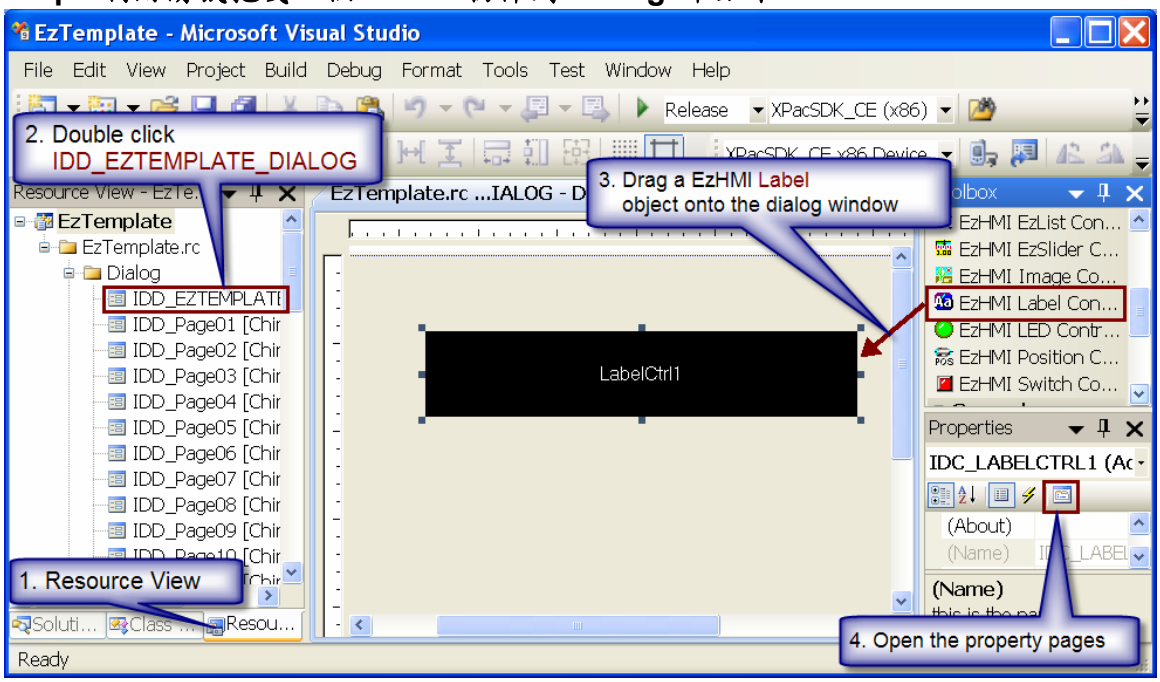
本範例說明如何使用”USER 自定執行緒”功能”，設計與使用時要特別注意，由於自定執行程序只執行一次，如果要設計為迴圈或無窮迴圈，請在其中加入暫停指令”Sleep(ms)” 及最好要設計停止迴圈的機制，以方便控制執行整個程序結束時間。

這將展示如何利用 USER 自定執行緒，設計 HMI 操作與程序處理方法，當使用者在 HMI 按下離開按鈕後如何完成對映操作的控制程序(顯示 Hello)。

請依照 2.2.1 複製樣板專案，請名稱改為 UserThread,執行檔為 UserThread.exe，並開啟之。

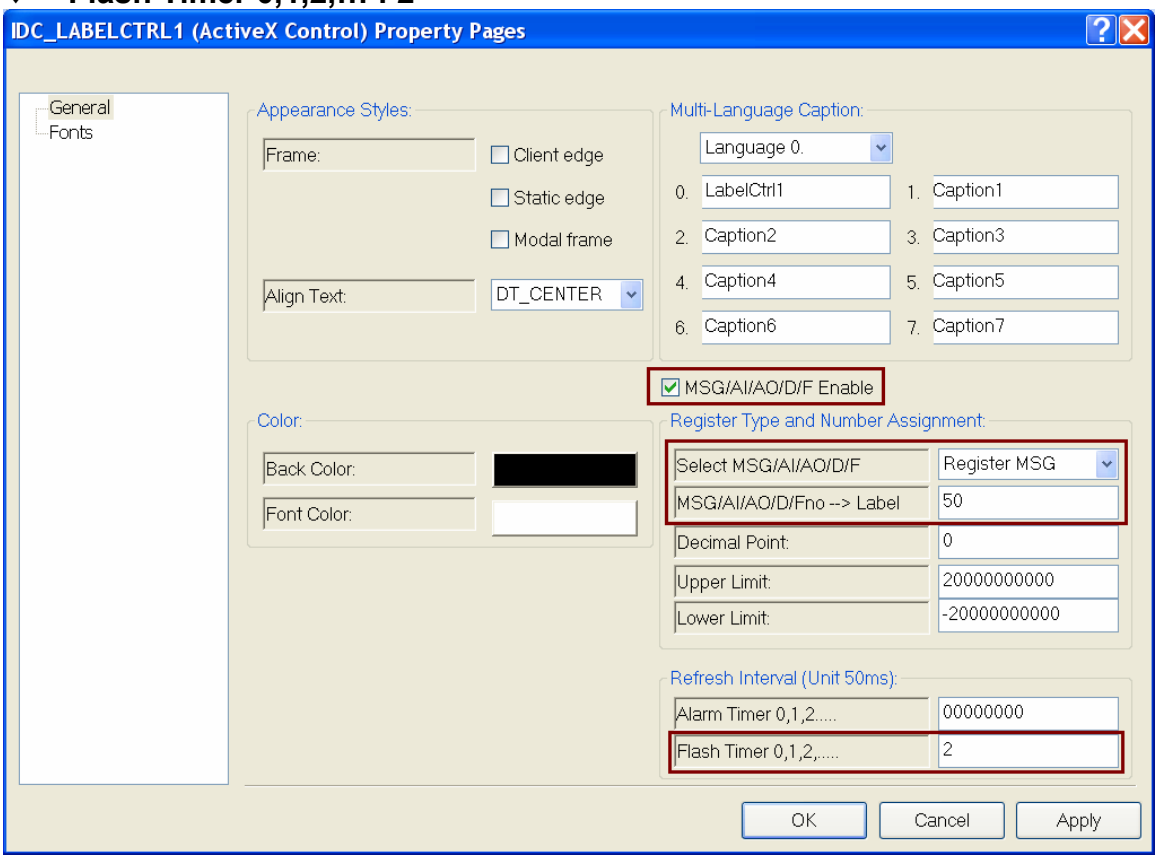
2.7.1 HMI 物件畫面控制設計

Step1.利用滑鼠拖曳一個 Label 物件到 Dialog 中如下.

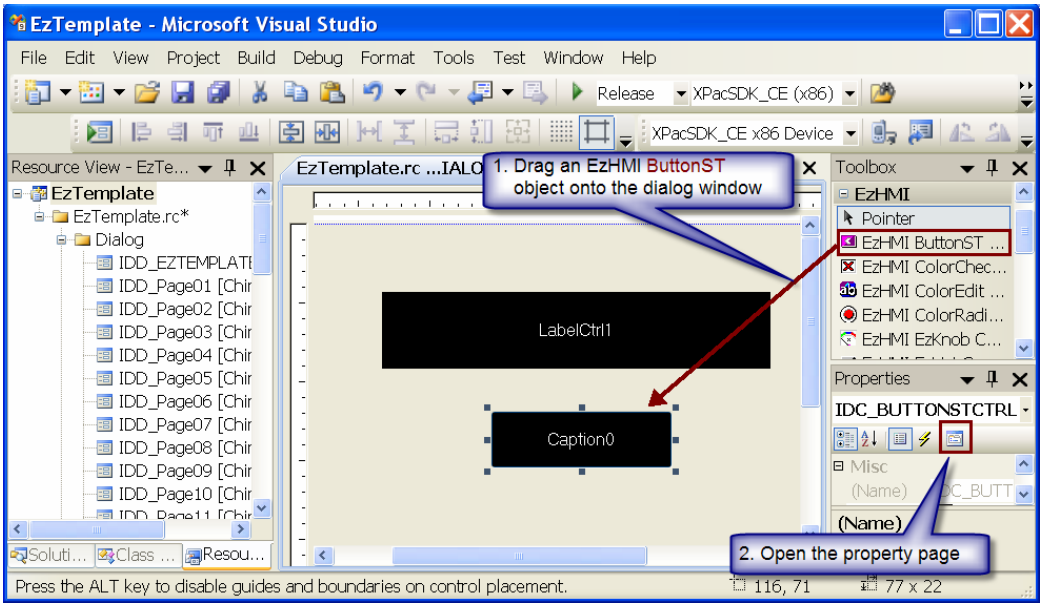


設定屬性:

- **MSG/AI/AO/D/F Enable : 選取**
- **Select MSG/AI/AO/D/F : Register MSG**
- **MSG/AI/AO/D/Fno : 50**
- **Flash Timer 0,1,2,… : 2**

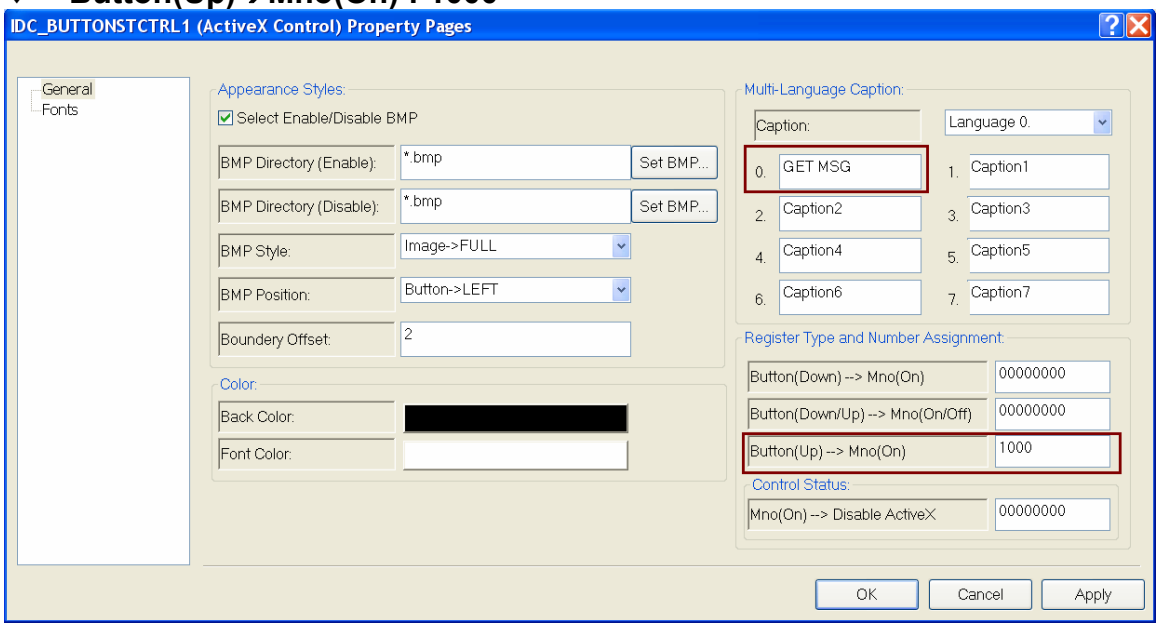


Step2.再利用滑鼠拖拉一個 ButtonST 物件到 Dialog 中如下



設定屬性:

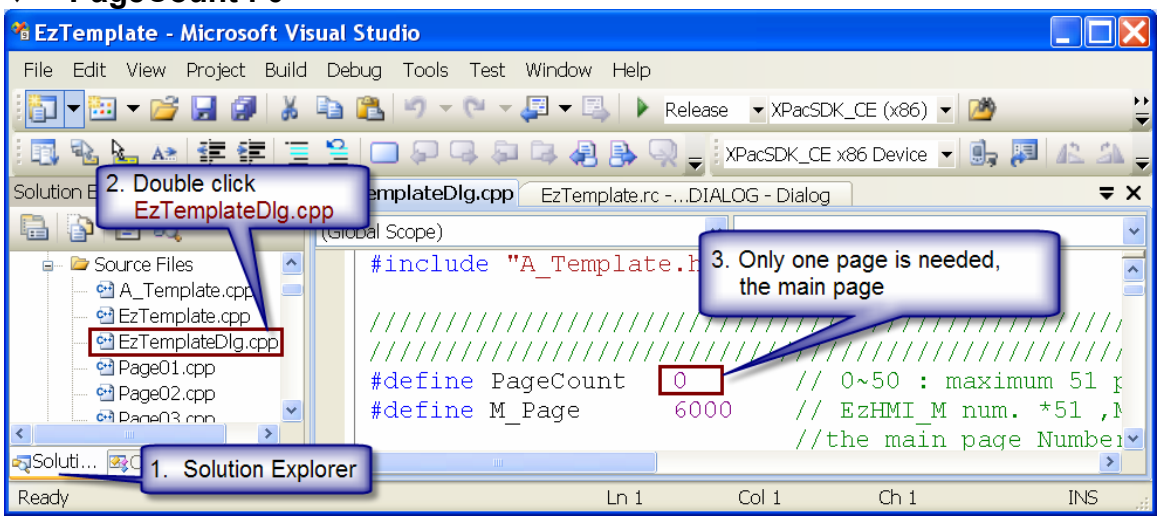
- **Caption 0 : GET_MSG**
- **Button(Up)→Mno(On) : 1000**



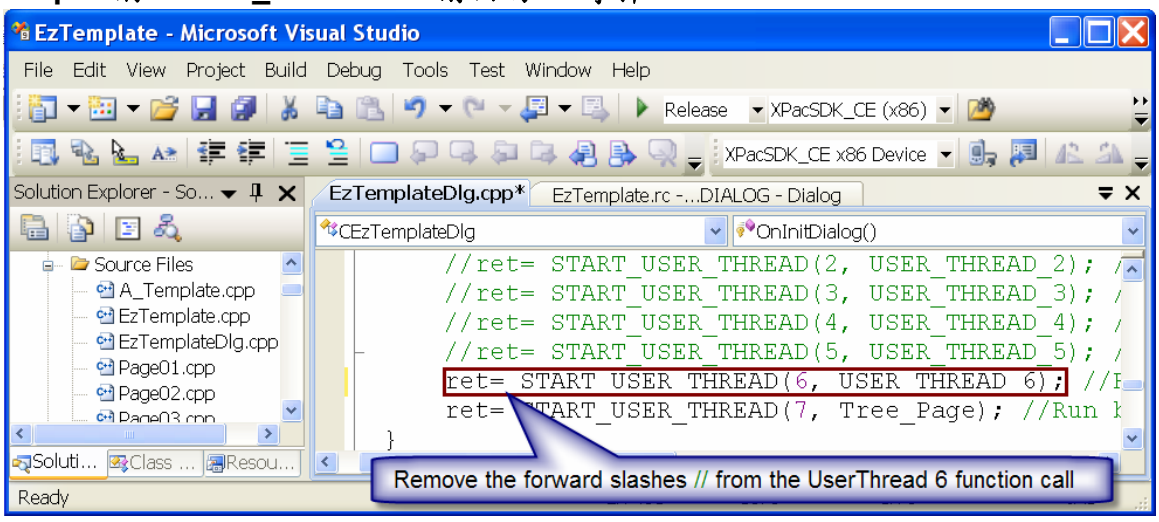
2.7.2 設計與編譯

Step1.在專案管理選 Solution 檢視，並雙擊 EzTemplateDlg.cpp 做如下設定。

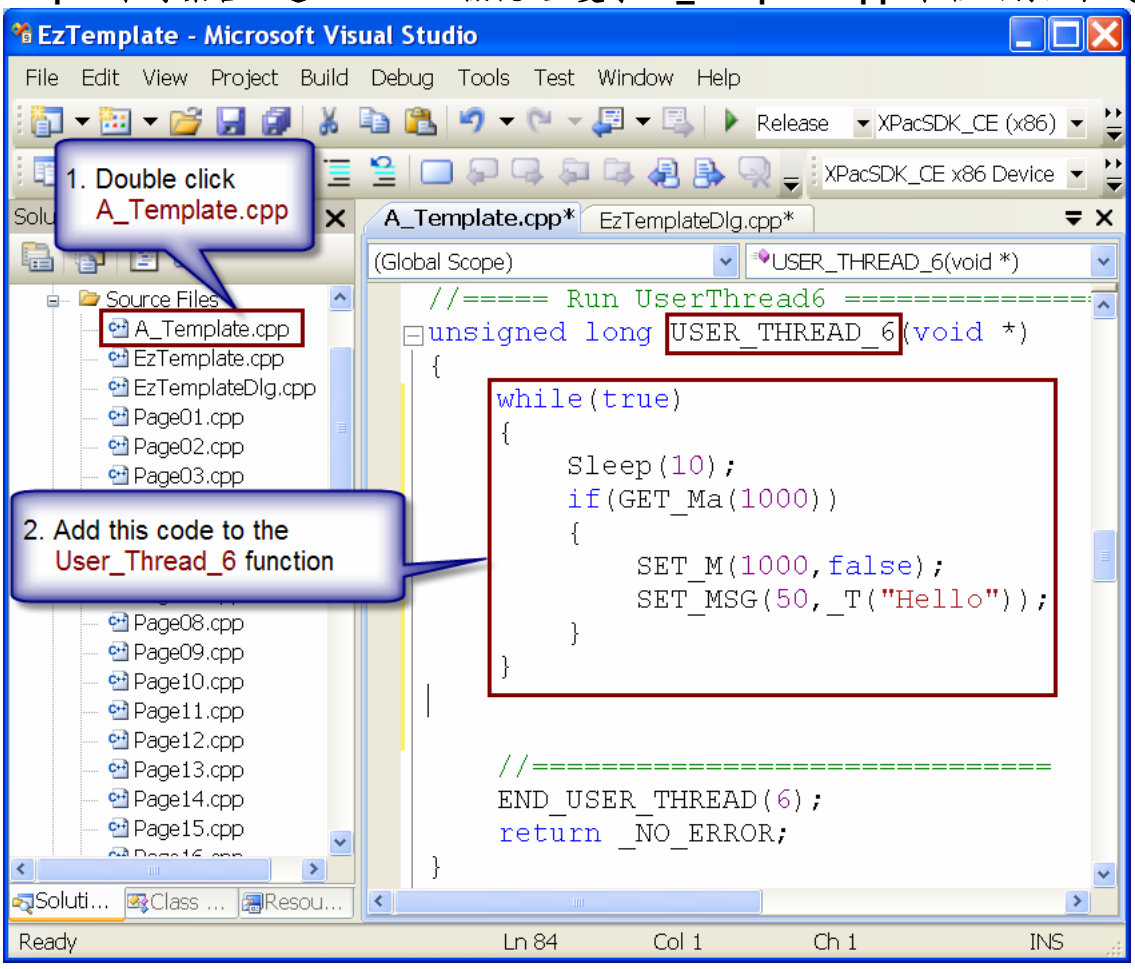
- **PageCount : 0**



Step2. 將 USER_THREAD6 前面的 // 拿掉



Step3. 在專案管理選 Solution 檢視，並雙擊 A_Template.cpp 在裡面打上程式。



程式內容：

```
while(true)
{
    Sleep(10);
    if(GET_Ma(1000))
    {
        SET_M(1000,false);
        SET_MSG(50,_T("Hello"));
    }
}
```

迴圈內每次暫停 10 ms

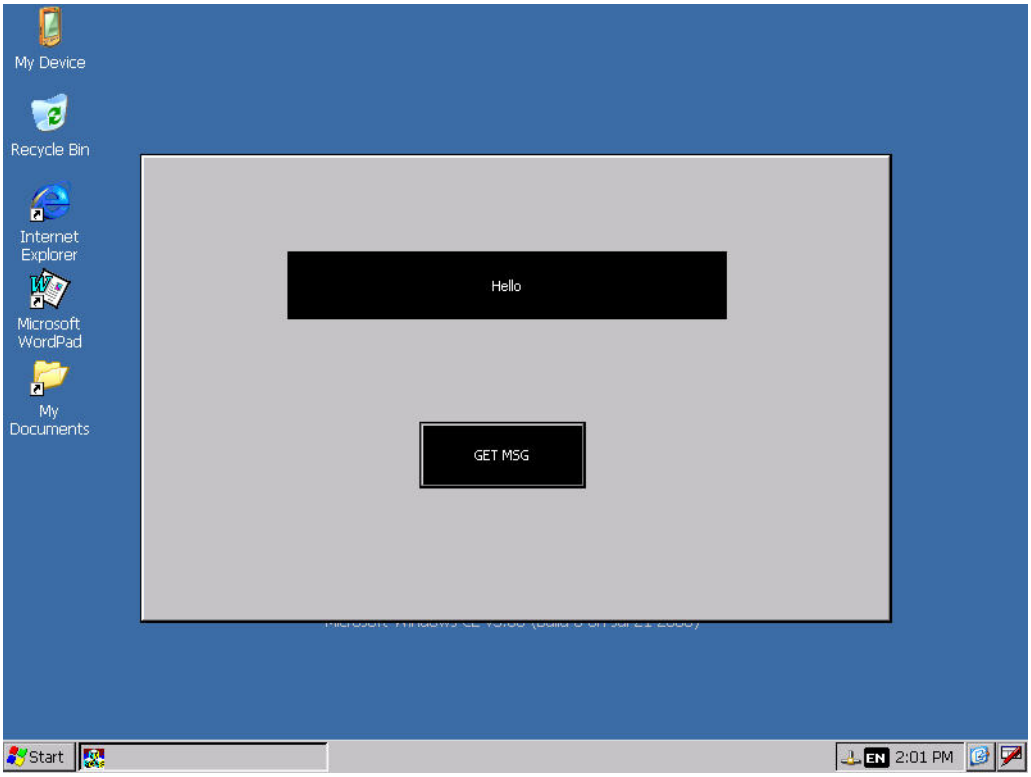
判斷是否有按下按鈕M 1000

如果有按下按鈕則須將 M 1000 點還原為 false ，下次才能再判斷是否有被按下

如果按下按鈕就把 MSG50 暫存器的內容設定為 “Hello”

2.7.3 專案啟始與測試

按熱鍵 F5 連線下載執行:(請先確認網路設定是正確的)，請參閱 EzProg-I Getting Started 手冊中附錄的連線除錯方法說明。如果沒有其他異常，在 PAC 畫面出現如下對執行畫面，當按下 ButtonST 按鈕，上面的 Label 會顯示出 “Hello”



2.8 範例 RTSR

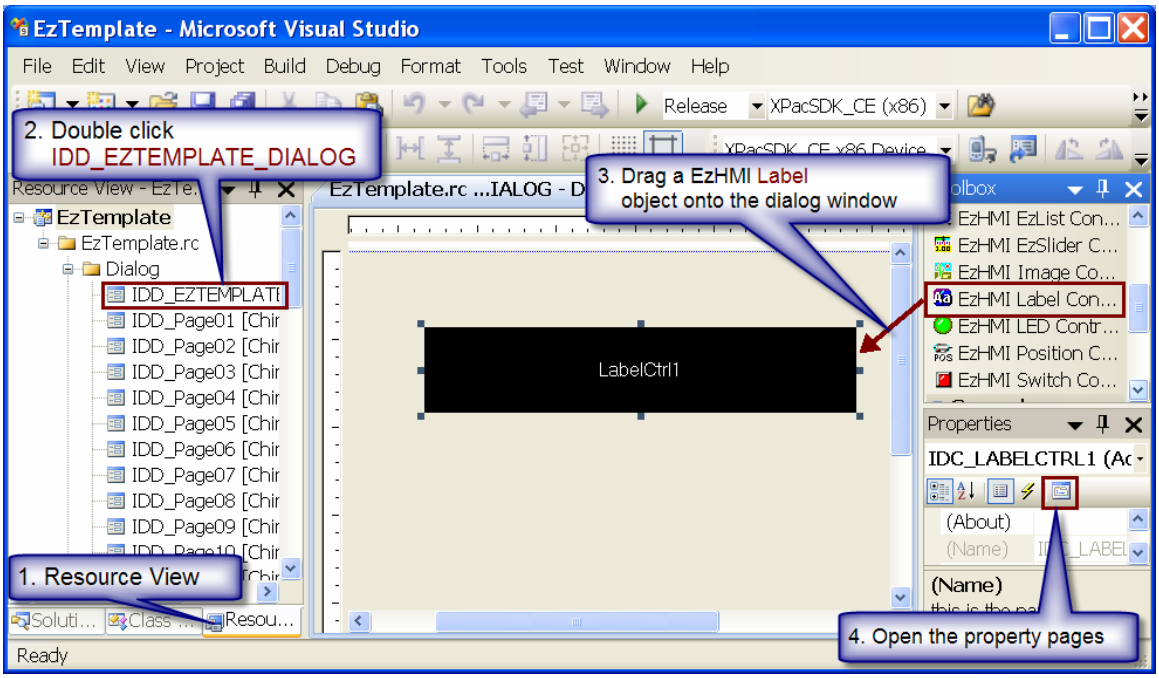
本範例說明如何使用”RTSR 自訂定時執行”功能，系統會定時來自動執行本程序，設計與使用時要特別注意，由於定時執行程序優先權很高，所以請盡量避免在 RTSR 處理程序中撰寫執行時間很長的迴圈或無窮迴圈或程式碼與暫停指令如 Sleep(ms) 執行整個程序超過設定定時時間，將會使下一次執行時間自動延後。

請依照 2.2.1 複製樣板專案，請名稱改為 RTSR 執行檔為 RTSR.exe，並開啟之。

本範例將展示如何利用 RTSR 定時執行的特性，並使用 EzLIB 提供的 API，將系統時間顯示出來。

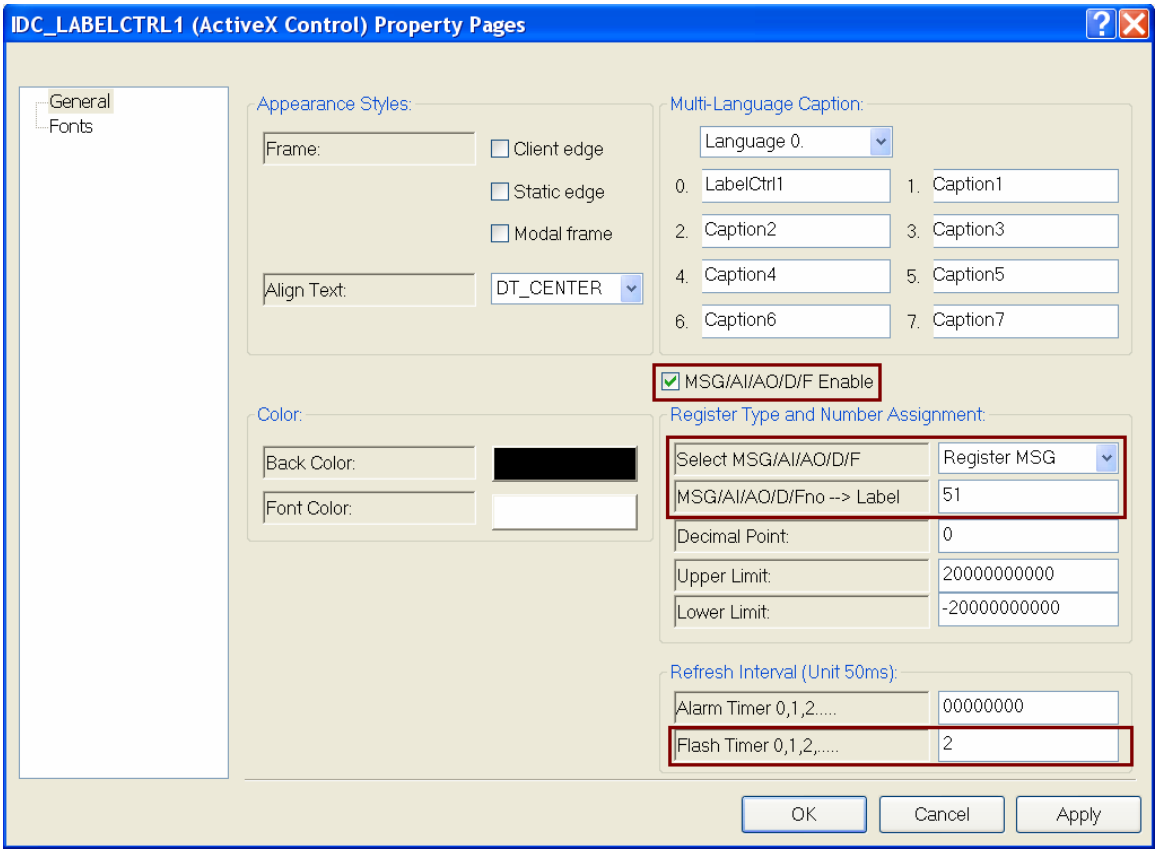
2.8.1 HMI 物件畫面控制設計

Step1. 選擇 Resource View 檢視，雙擊 IDD_EZTEMPLATE_DIALOG 開啟畫面，拖曳一個 Label 放在畫面上



設定屬性:

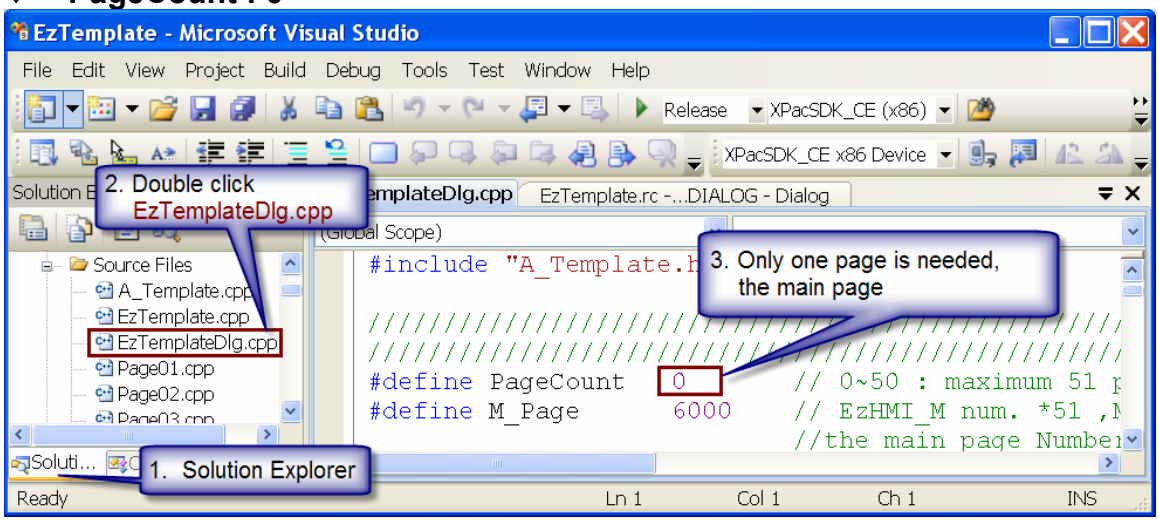
- **MSG/AI/AO/D/F Enable : 選取**
- **Select MSG/AI/AO/D/F : Register MSG**
- **MSG/AI/AO/D/Fno : 51**
- **Flash Timer 0,1,2,… : 2**



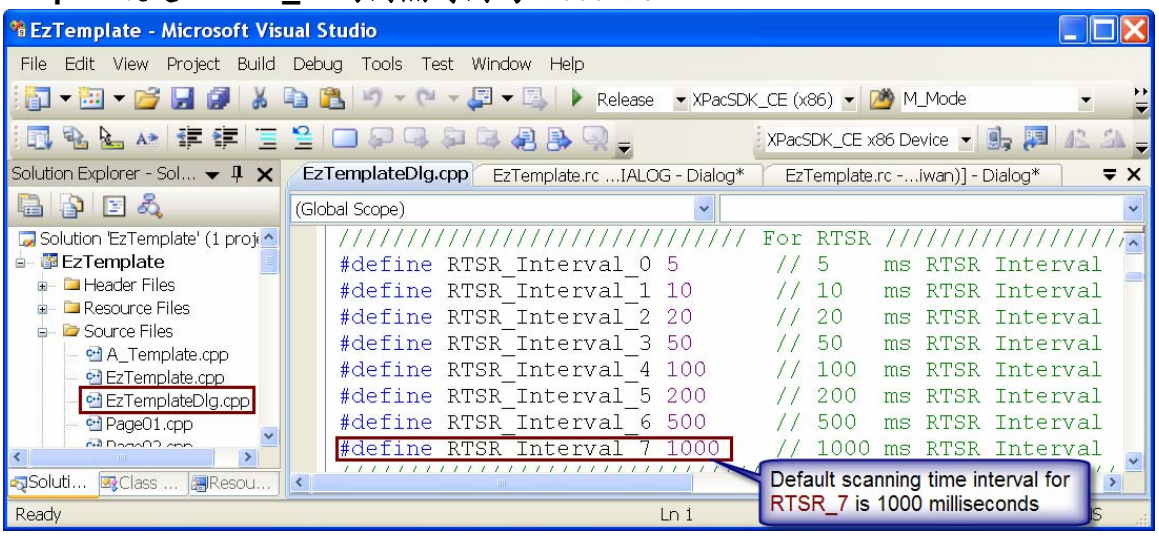
2.8.2 設計與編譯

Step1.在專案管理選 Solution 檢視，並雙擊 EzTemplateDlg.cpp 做如下設定。

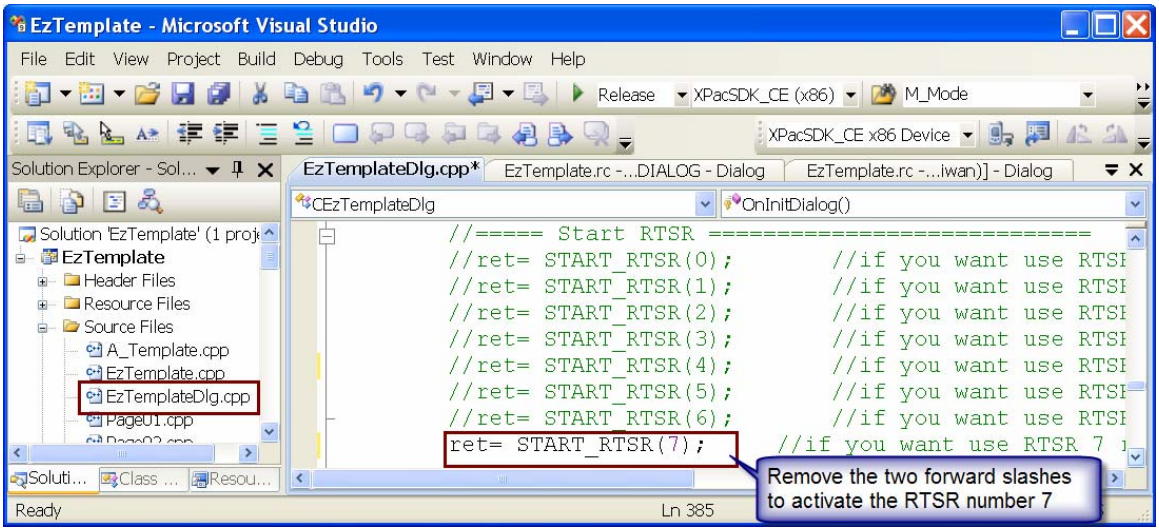
- **PageCount : 0**



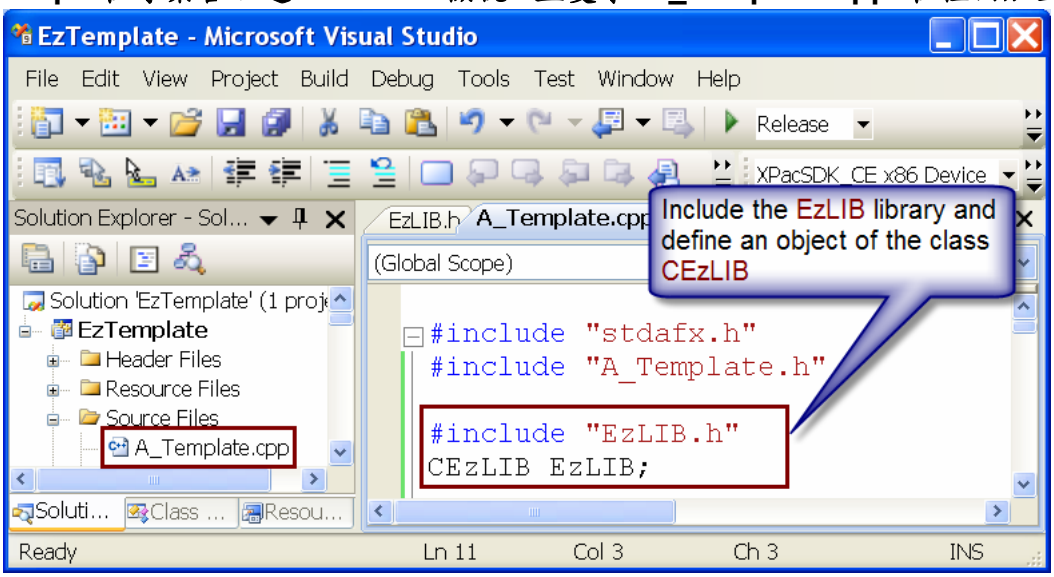
Step2. 設定 RTSR_7 的間隔時間為 1000 ms



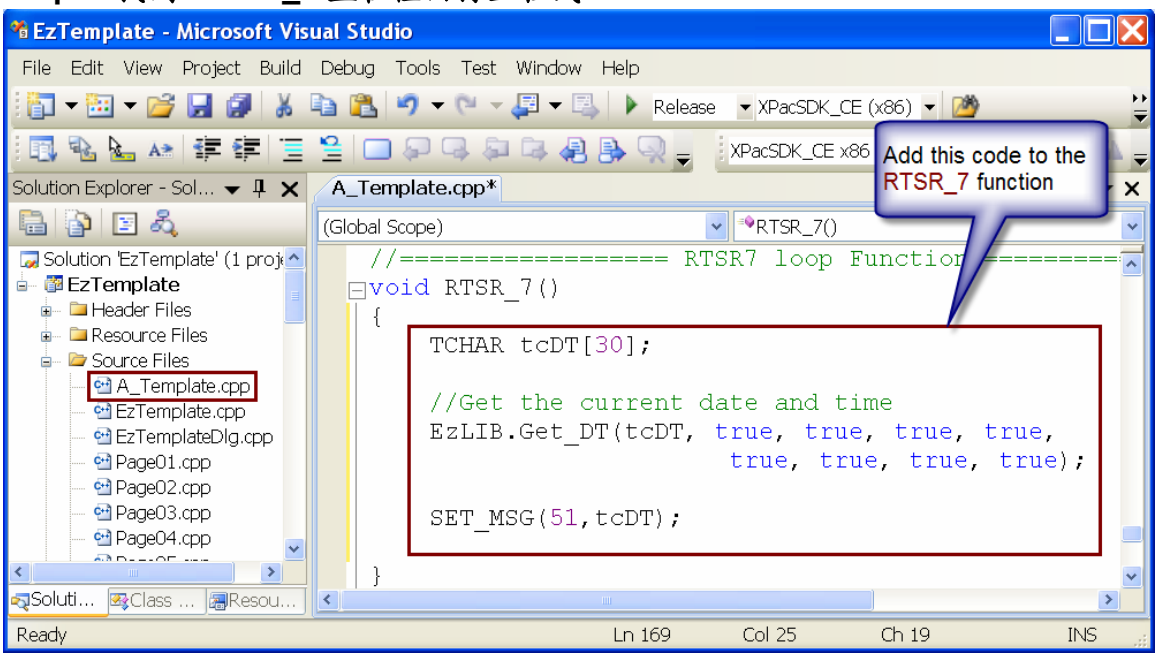
Step3. 將 START_RTSR(7) 前的 // 刪掉以啟用它



Step4.在專案管理選 Solution 檢視，並雙擊 A_Template.cpp 在裡面加上程式。



Step5. 找到 RTSR_7 並在裡面打上程式



程式內容：

加入 EzLIB 的宣告:

在 A_Template.cpp 檔案宣告位置輸入

將EzLib 包含到 A_Template.cpp

```
#include "EzLib.h"
CEzLib EzLib;
```

以 EzLib 去繼承 CEzLib 類別

宣告一個長度30的 TCHAR 陣列

利用 EzLib 抓取日期與時間放到 tcDT

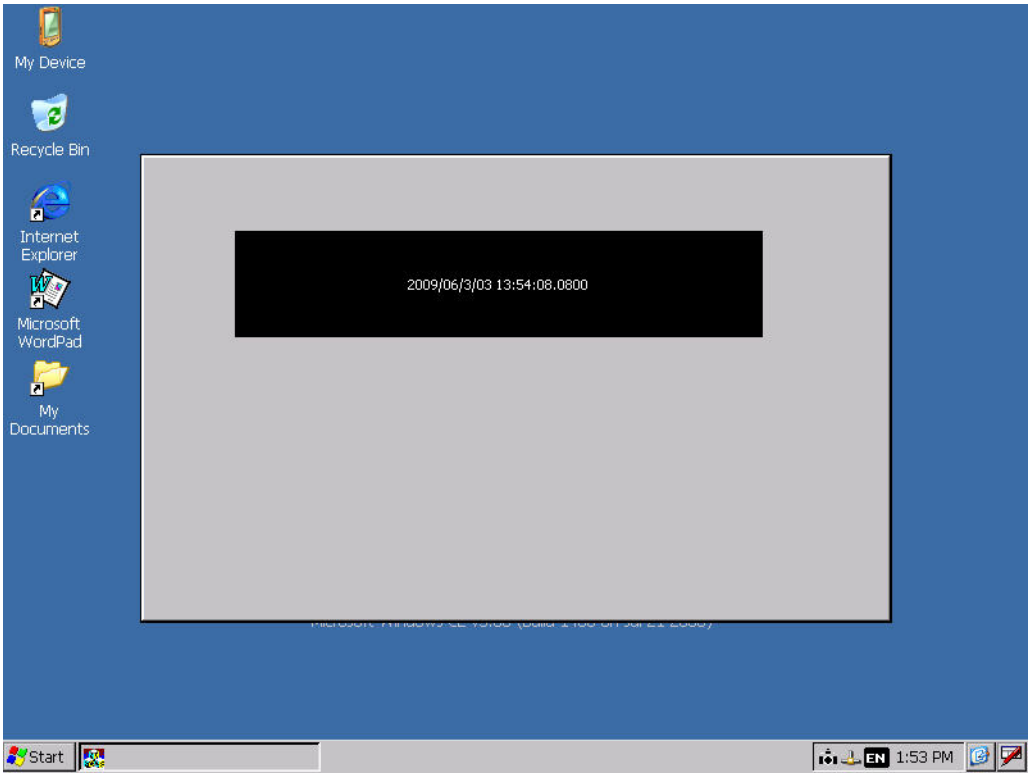
```
TCHAR tcDT[30];
EzLIB.Get_DT(tcDT, true, true, true, true, true, true, true, true);
SET_MSG(51,tcDT);
```

把 tcDT 放到 MSG51

2.8.3 專案啟始與測試

按熱鍵 F5 連線下載執行:(請先確認網路設定是正確的) ，請參閱 EzProg-I Getting Started 手冊中附錄的連線除錯方法說明。如果沒有其他異常，在 PAC 畫面出現如下對執行畫面。

RTSR 固定週期去抓取系統日期、時間再由 Label 顯示出來。



2.9 範例 Timer

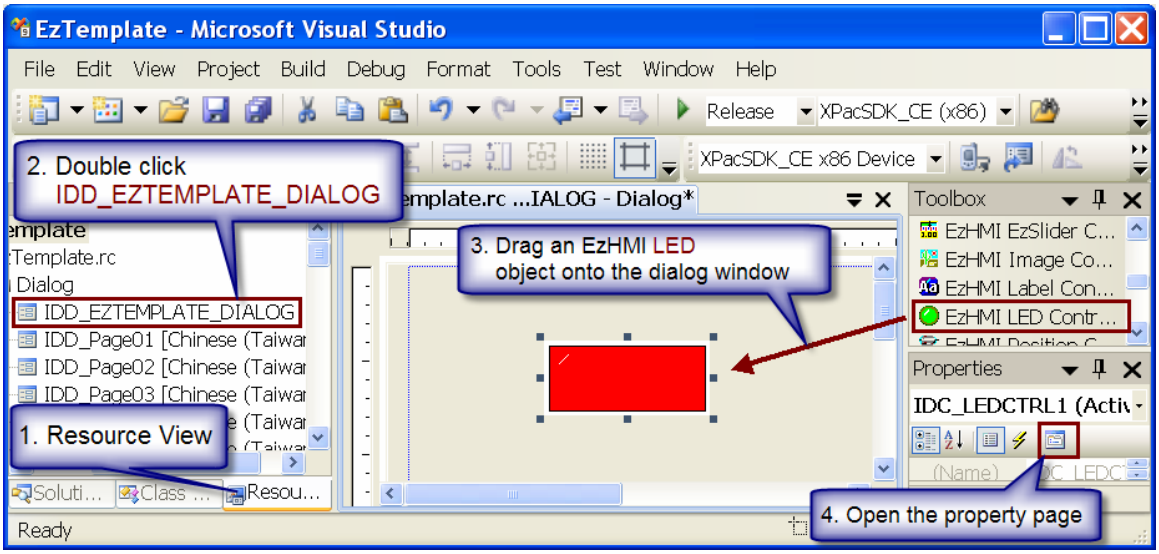
本範例說明如何使用”計時器”功能，使用計時器 Timer 可設定一個倒數計時的時間，時間單位為 ms。詳細應用程式庫可參照 “EzProg-I_Tool” 手冊。

請依照 2.2.1 複製樣板專案，請名稱改為 Timer 執行檔為 Timer.exe，並開啟之。

本範例將展示使用 EzCore 中提供的 Timer 功能，實現包括啟動，停止，顯示等功能。

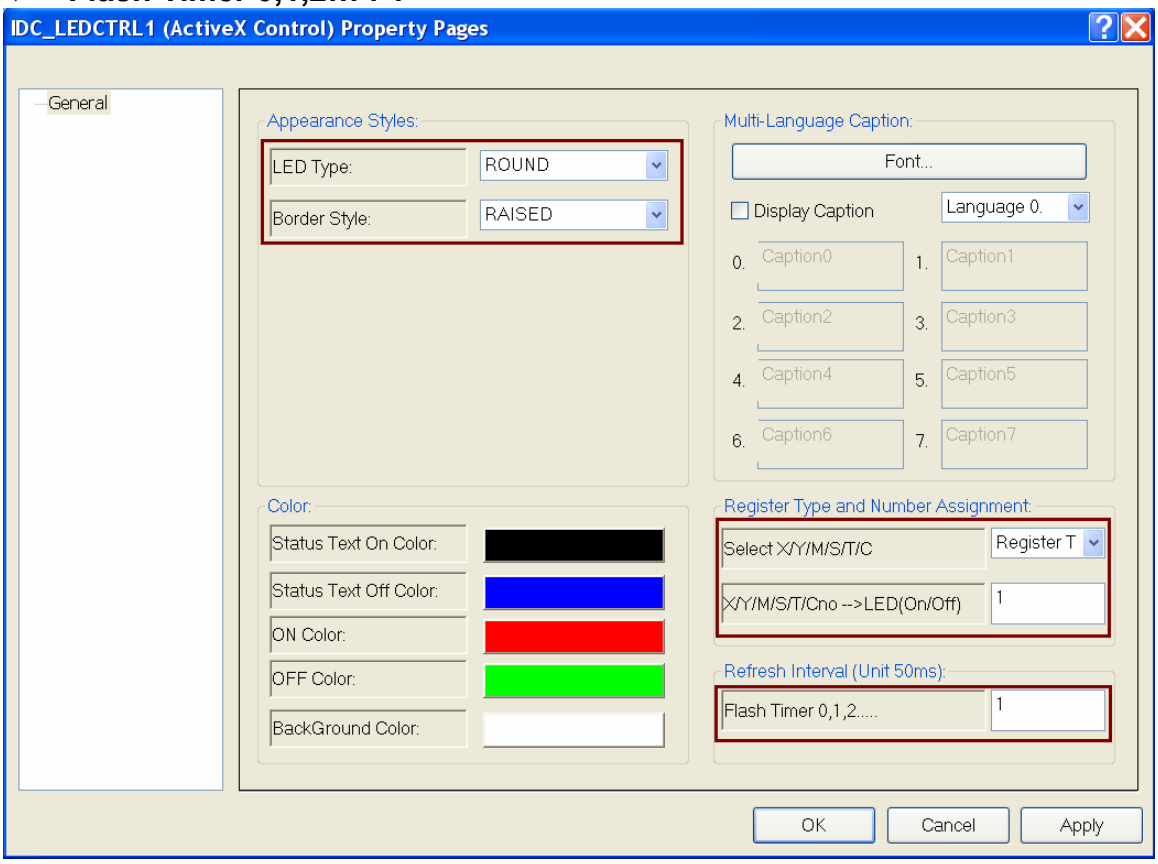
2.9.1 HMI 物件畫面控制設計

Step1.選擇 Resource View 檢視，雙擊 IDD_EZTEMPLATE_DIALOG 開啟畫面，拖曳一個 LED 放在畫面上

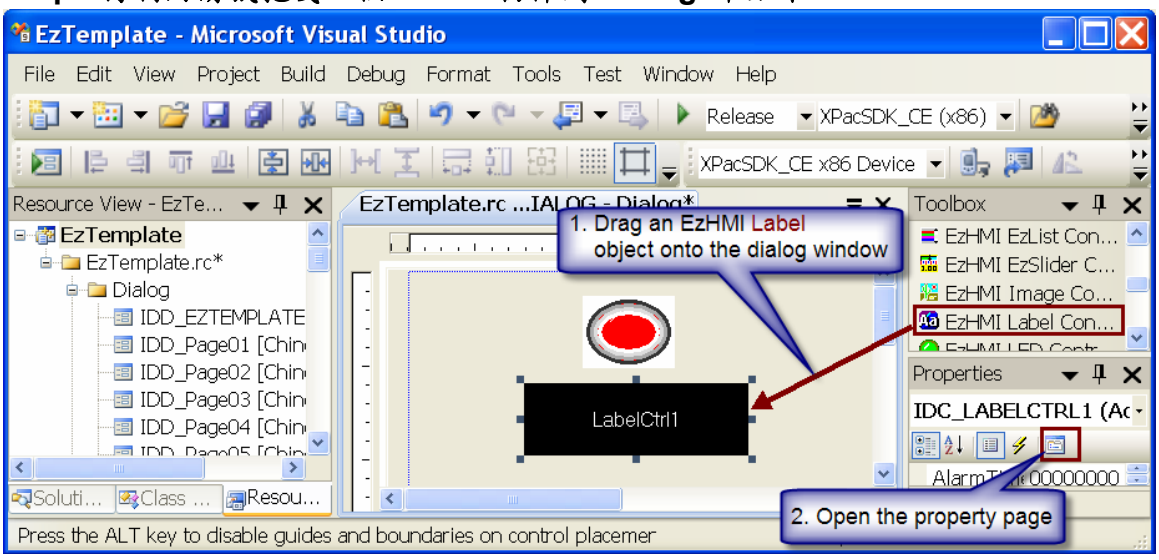


設定屬性:

- **LED Type : ROUND**
- **Border Style : RAISED**
- **Select X/Y/M/S/T/C : Register T**
- **X/Y/M/S/T/Cno→LED(On/Off) : 1**
- **Flash Timer 0,1,2… : 1**

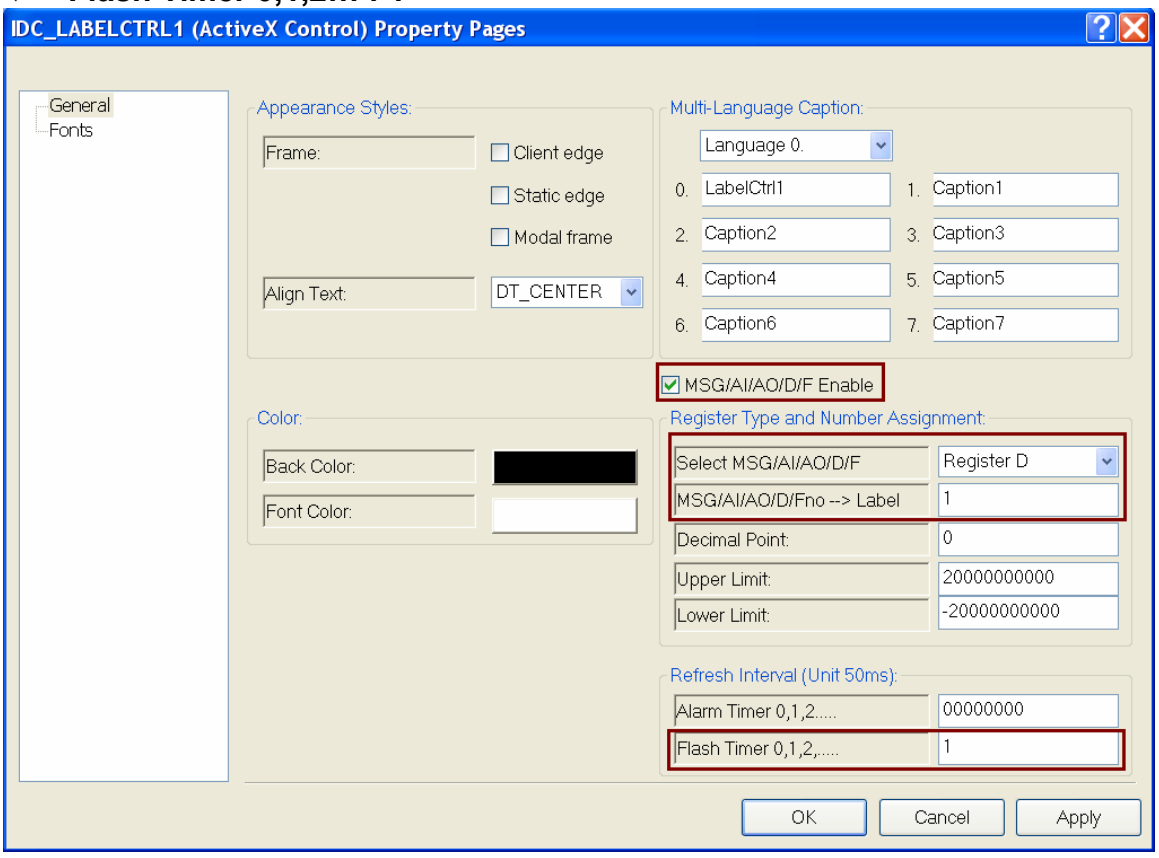


Step2.再利用滑鼠拖曳一個 Label 物件到 Dialog 中如下

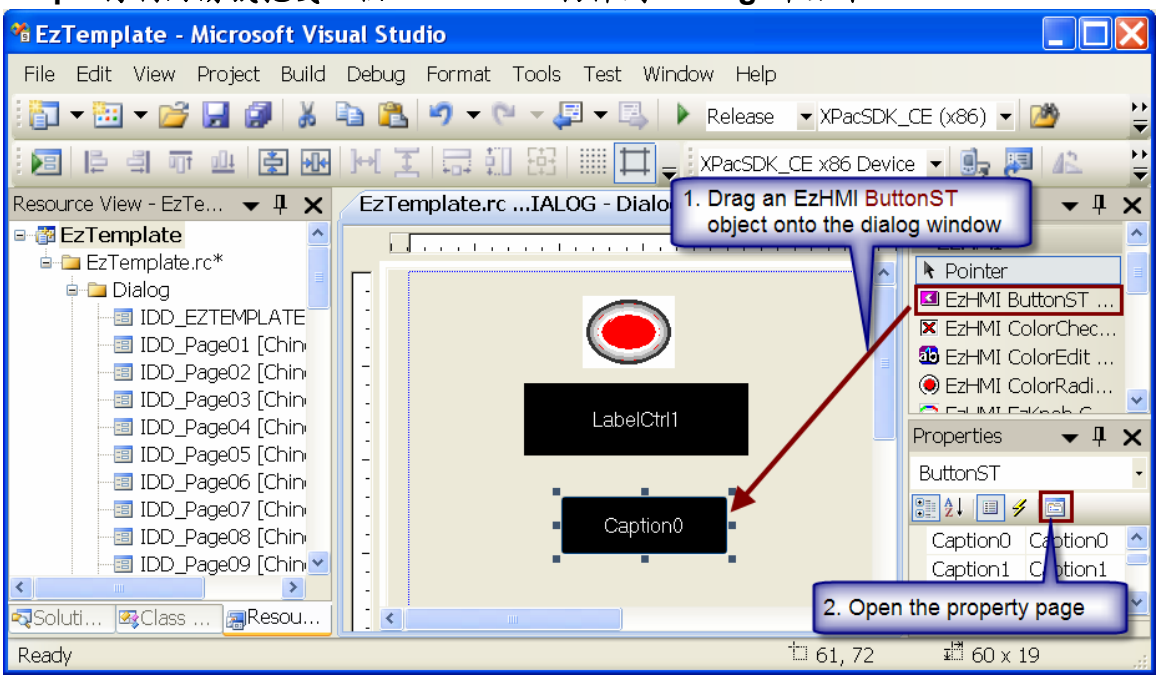


設定屬性:

- **MSG/AI/AO/D/F Enable : 選取**
- **Select MSG/AI/AO/D/F : Register D**
- **MSG/AI/AO/D/F→Label :1**
- **Flash Timer 0,1,2… : 1**

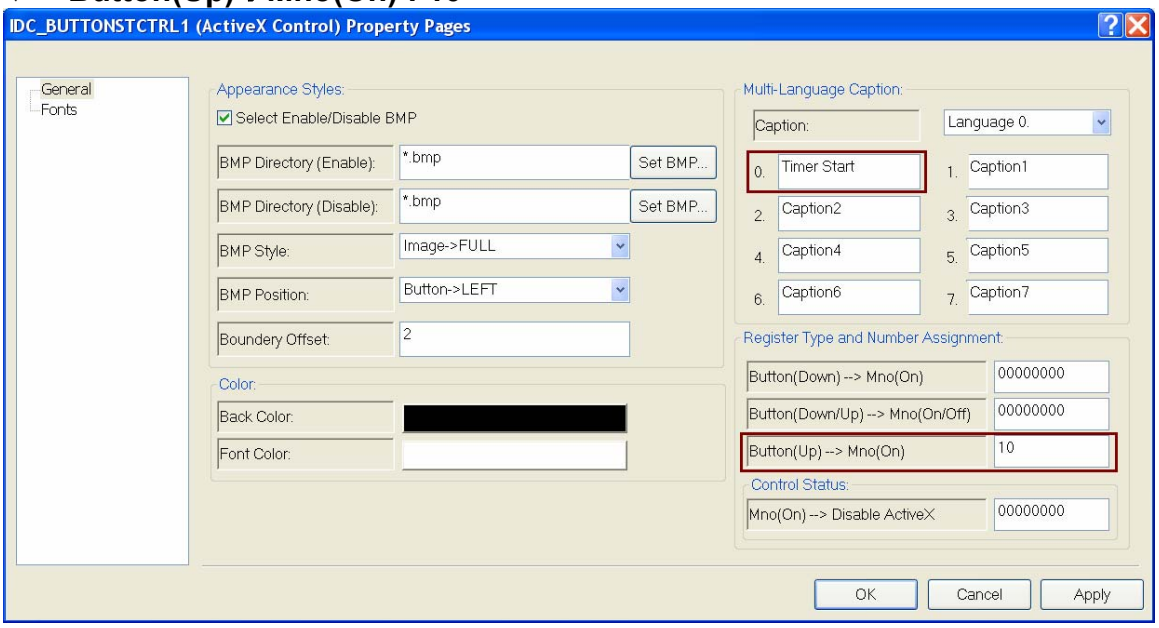


Step3.再利用滑鼠拖曳一個 ButtonST 物件到 Dialog 中如下

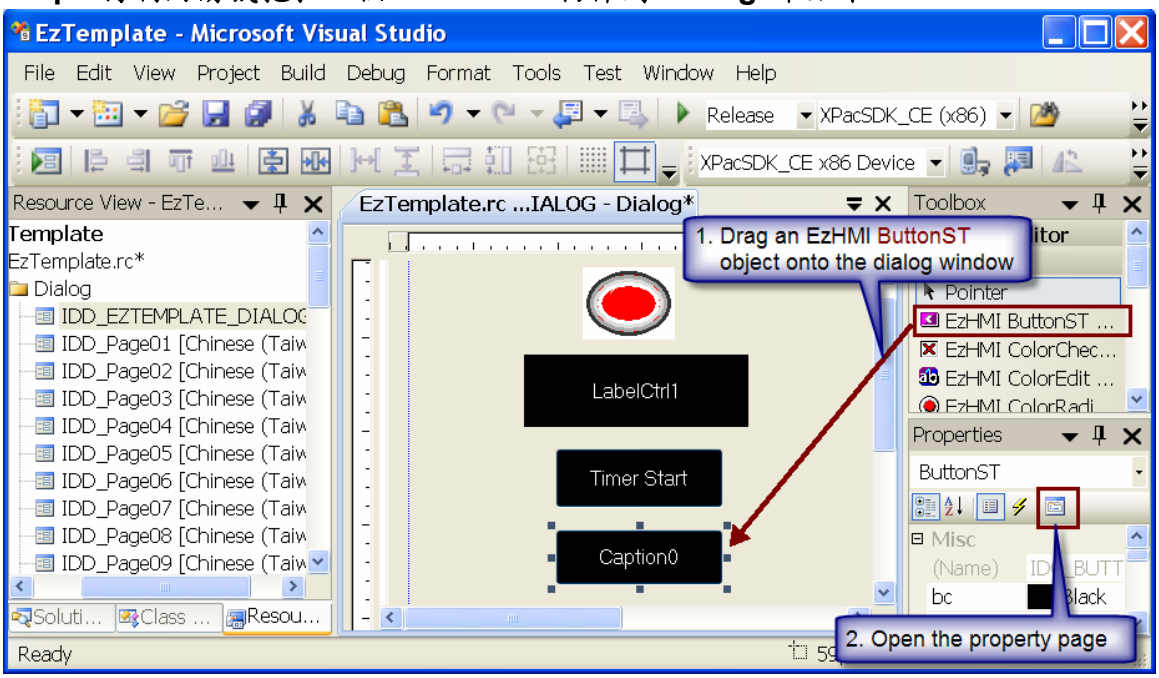


設定屬性:

- **Caption : Timer Start**
- **Button(Up)→Mno(On) : 10**

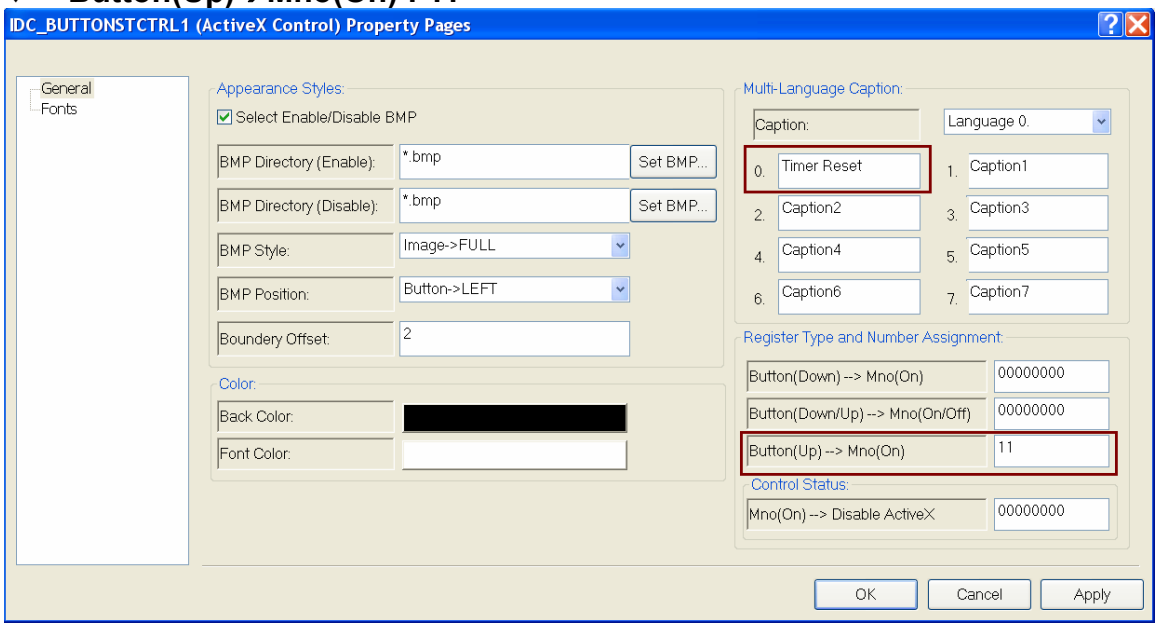


Step4.再利用滑鼠拖拉一個 ButtonST 物件到 Dialog 中如下



設定屬性:

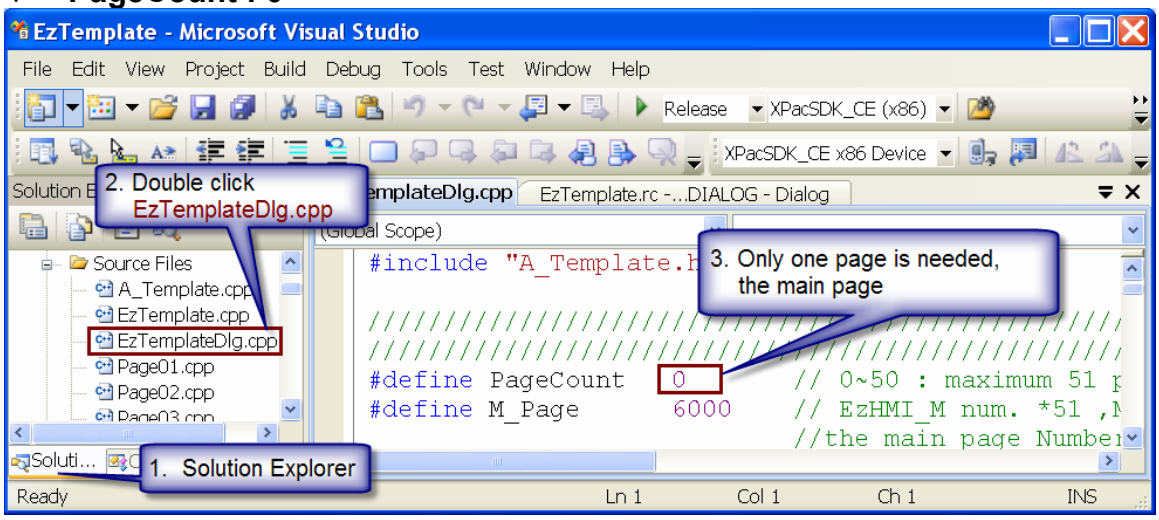
- **Caption : “Timer Reset”**
- **Button(Up)→Mno(On) : 11**



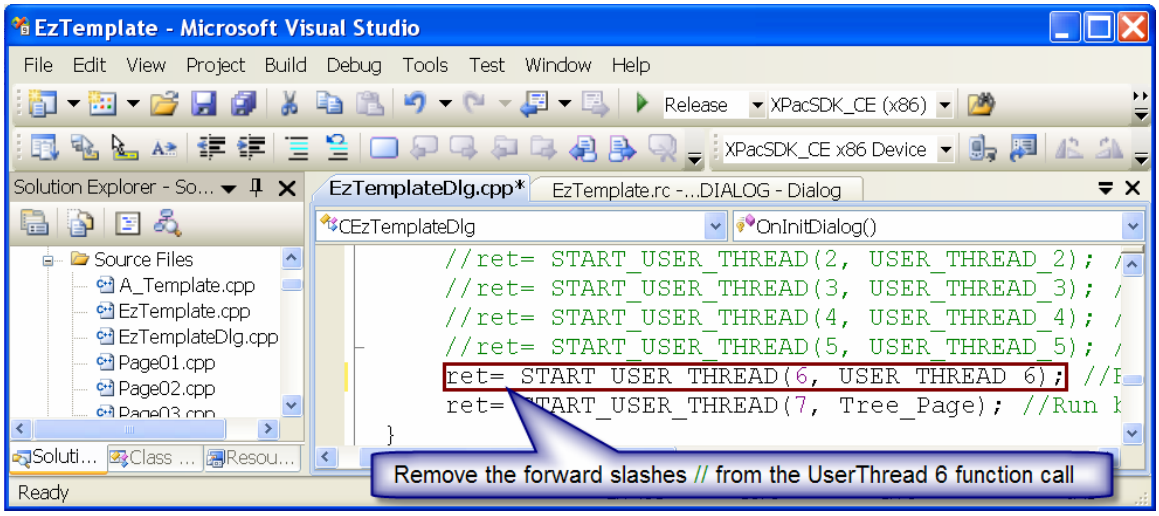
2.9.2 設計與編譯

Step1.在專案管理選 Solution 檢視，並雙擊 EzTemplateDlg.cpp 做如下設定。

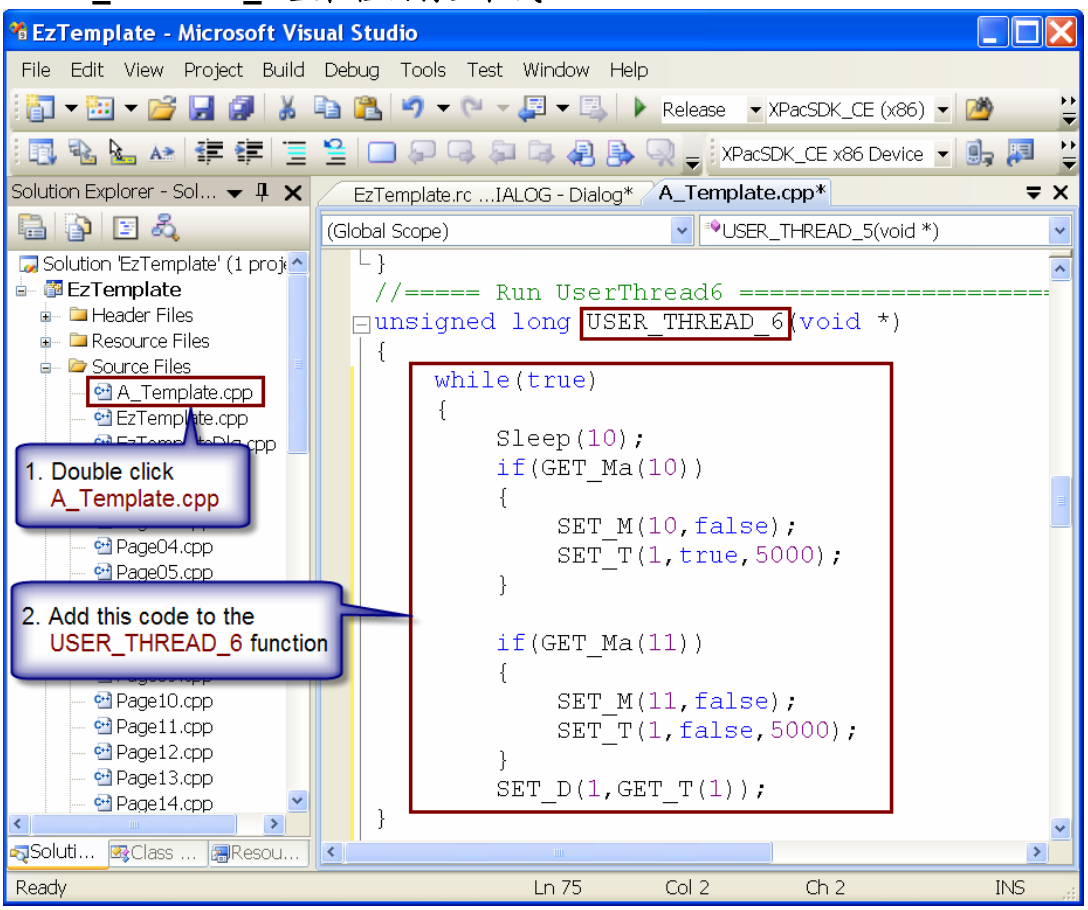
- **PageCount : 0**



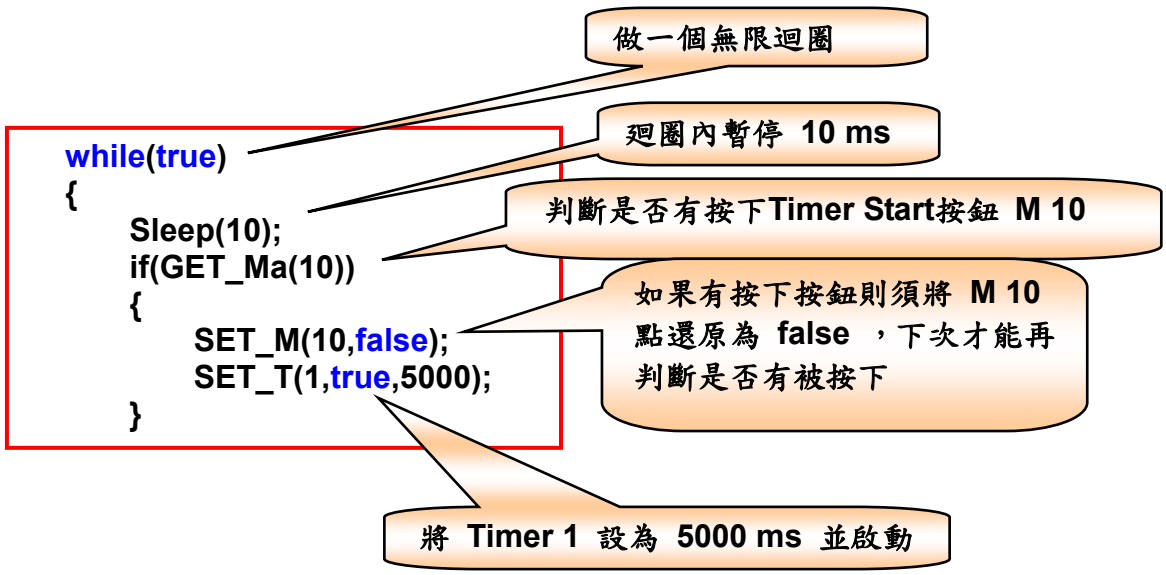
Step2. 將 USER_THREAD_6 前的 // 刪掉以啟用它

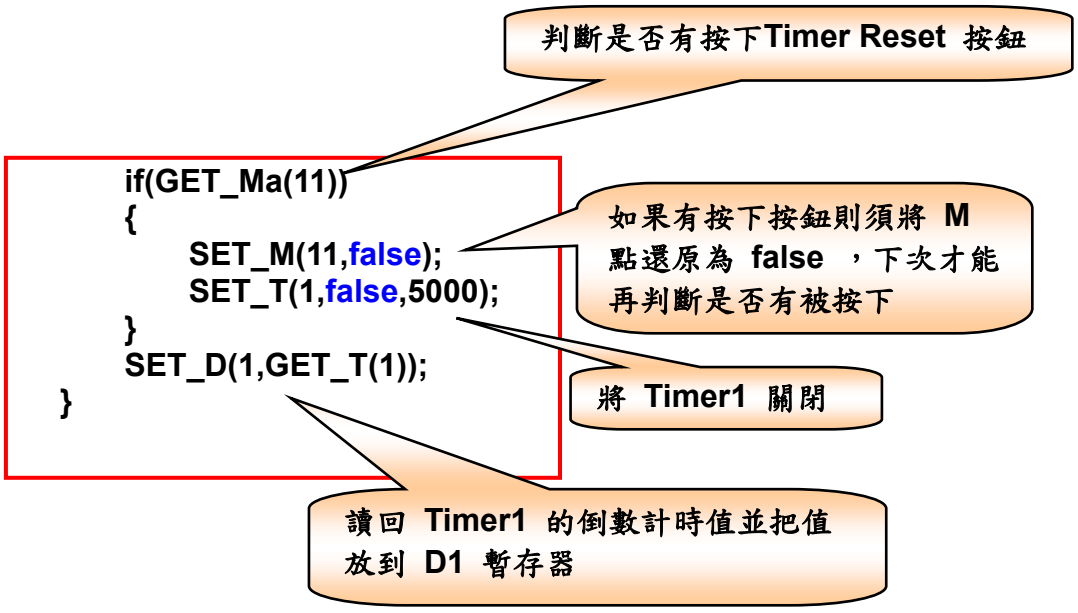


Step4.在專案管理選 Solution 檢視，並雙擊 A_Template.cpp 找到 USER_THREAD_6 並在裡面打上程式



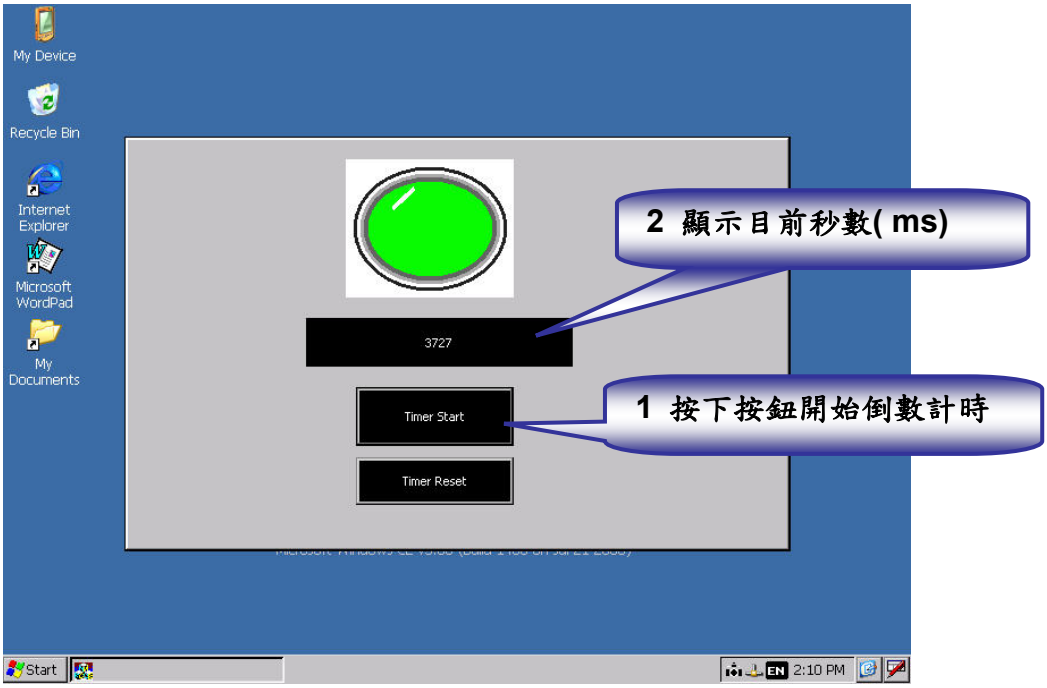
程式內容：

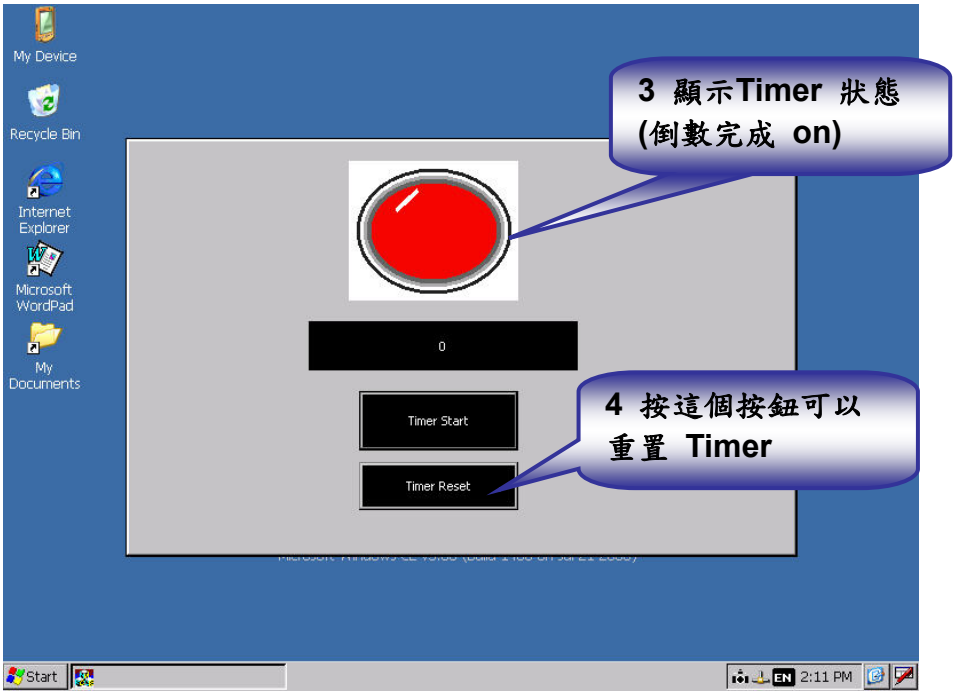



判斷是否有按下Timer Reset 按鈕
if(GET_Ma(11))
{
SET_M(11,false);
SET_T(1,false,5000);
}
SET_D(1,GET_T(1));
}
如果有按下按鈕則須將 M 點還原為 false ，下次才能再判斷是否有被按下
將 Timer1 關閉
讀回 Timer1 的倒數計時值並把值放到 D1 暫存器

2.9.3 專案啟始與測試

按熱鍵 F5 連線下載執行:(請先確認網路設定是正確的) ，請參閱 EzProg-I Getting Started 手冊中附錄的連線除錯方法說明。如果沒有其他異常，在 PAC 畫面出現如下對執行畫面。





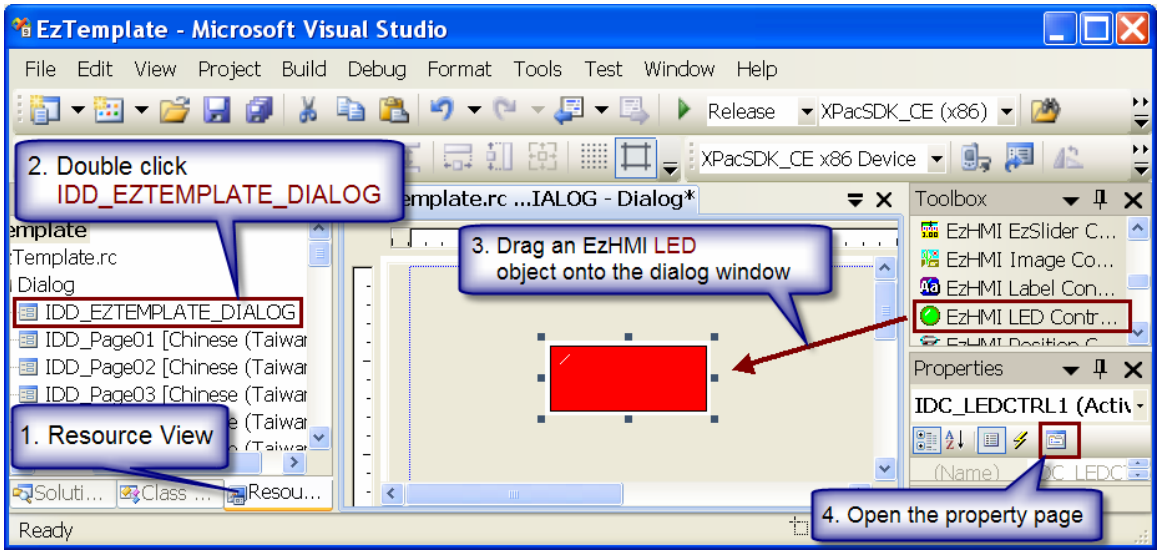
2.10 範例 Counter

本範例說明如何使用”計數器”功能，使用計時器 Counter 可設定一個倒數的數值。詳細應用程式庫可參照 “EzProg-I_Tool” 手冊。

請依照 2.2.1 複製樣板專案，請名稱改為 Counter 執行檔為 Counter.exe，並開啟之。

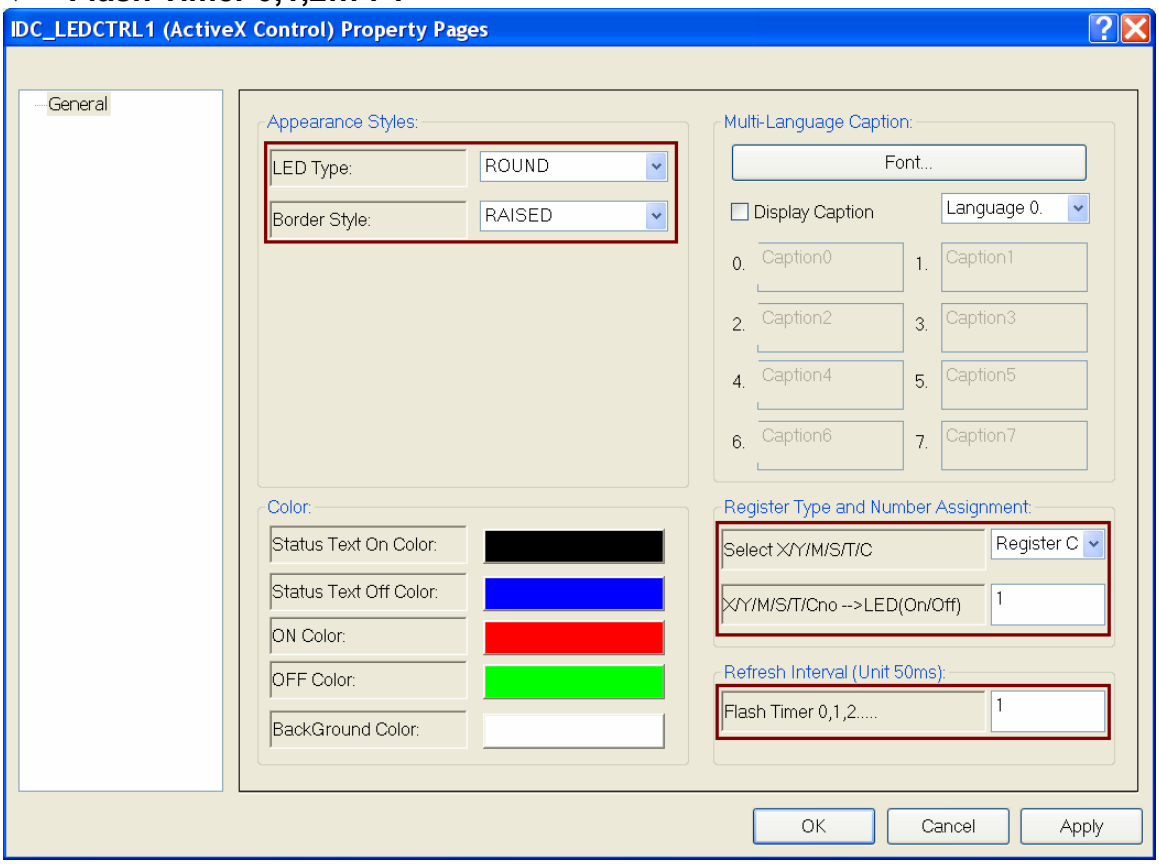
2.10.1 HMI 物件畫面控制設計

Step1.選擇 Resource View 檢視，雙擊 IDD_EZTEMPLATE_DIALOG 開啟畫面，拖曳一個 LED 放在畫面上

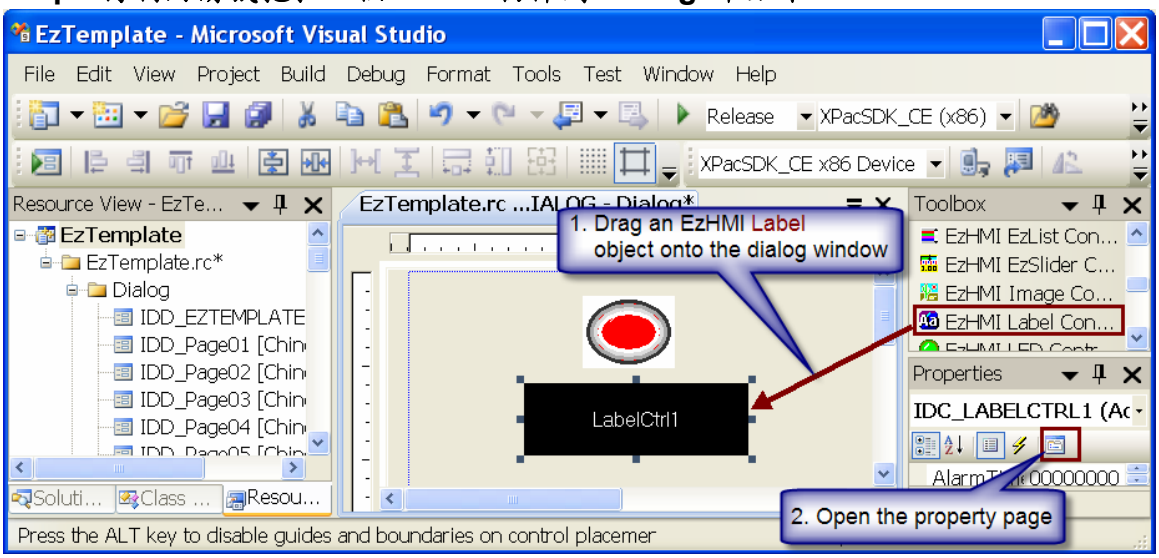


設定屬性:

- **LED Type : ROUND**
- **Border Style : RAISED**
- **Select X/Y/M/S/T/C : Register C**
- **X/Y/M/S/T/Cno→LED(On/Off) : 1**
- **Flash Timer 0,1,2… : 1**



Step2.再利用滑鼠拖拉一個 Label 物件到 Dialog 中如下

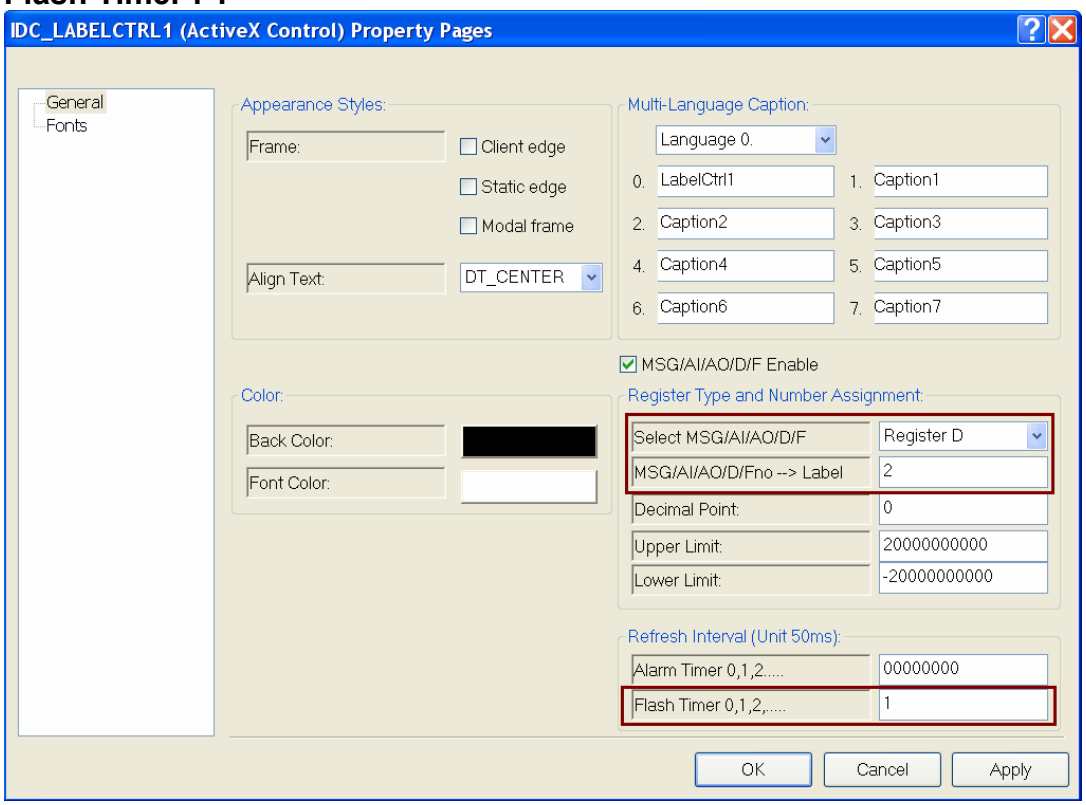


設定屬性:

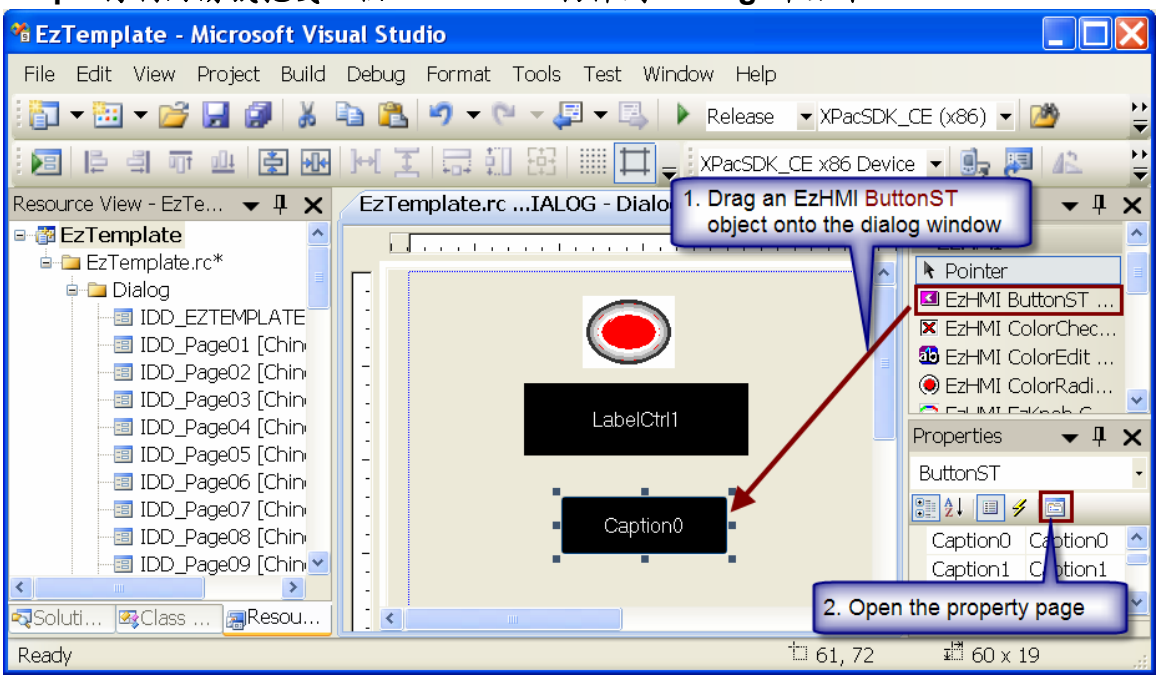
Select MSG/AI/AO/D/F : Register D

MSG/AI/AO/D/Fno→ Label : 2

Flash Timer : 1



Step3.再利用滑鼠拖曳一個 ButtonST 物件到 Dialog 中如下

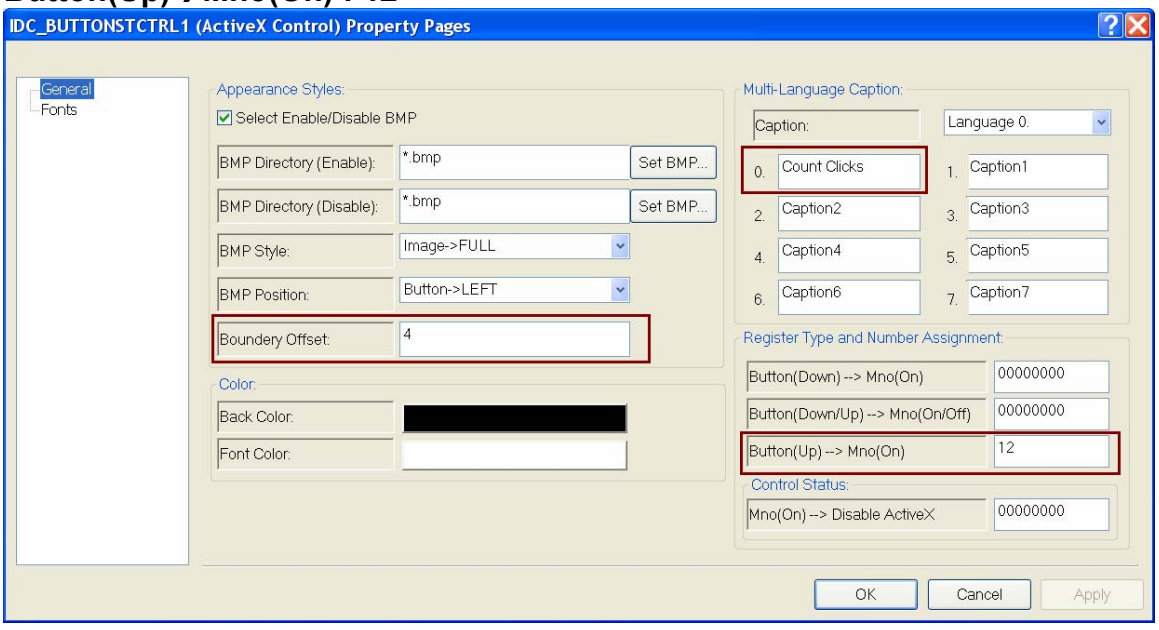


設定屬性:

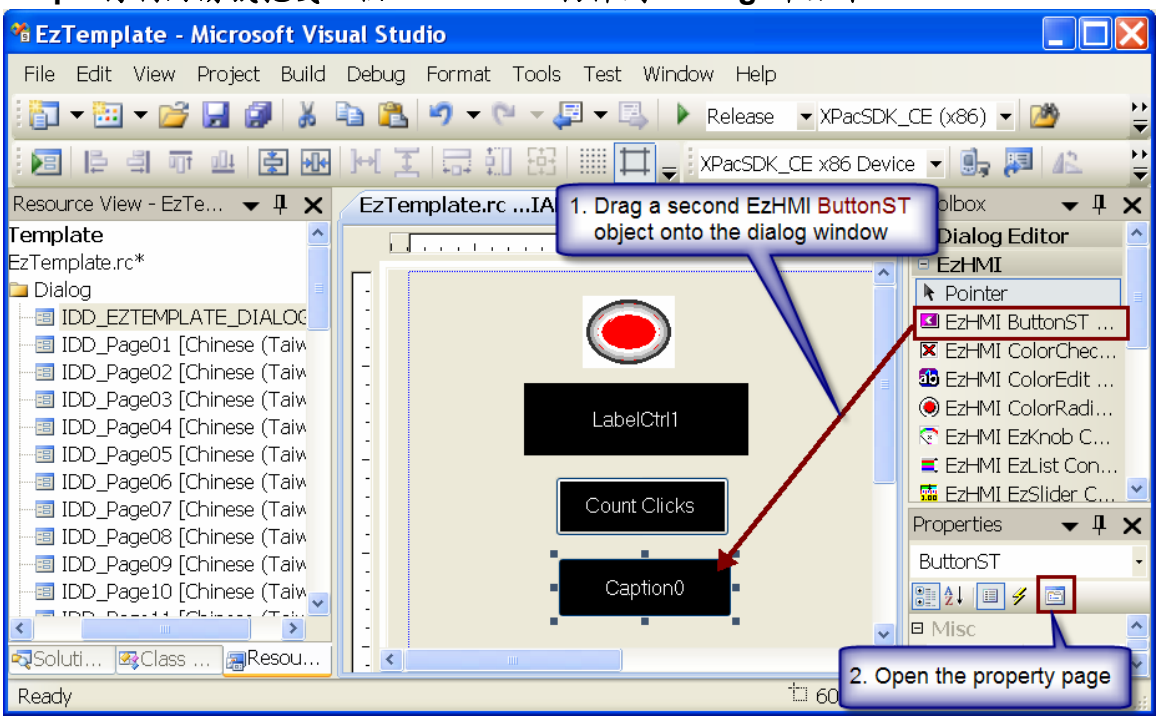
Boundery Offset : 4

Caption : Count Clicks

Button(Up)→Mno(On) : 12

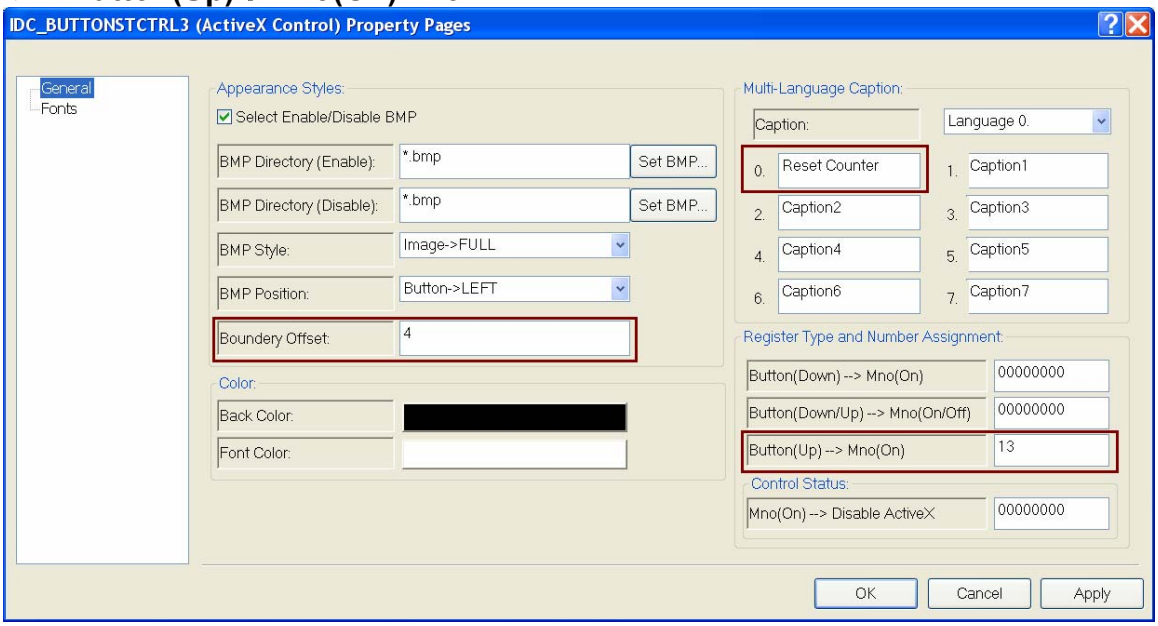


Step4.再利用滑鼠拖曳一個 ButtonST 物件到 Dialog 中如下



設定屬性:

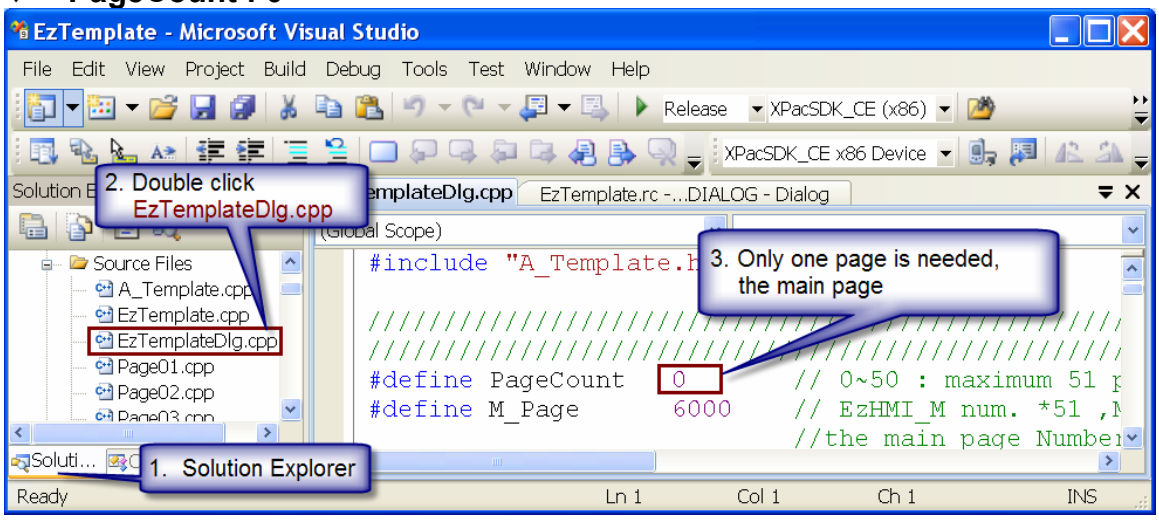
- **Boundery Offset : 4**
- **Caption : Reset Counter**
- **Button(Up)→Mno(On) : 13**



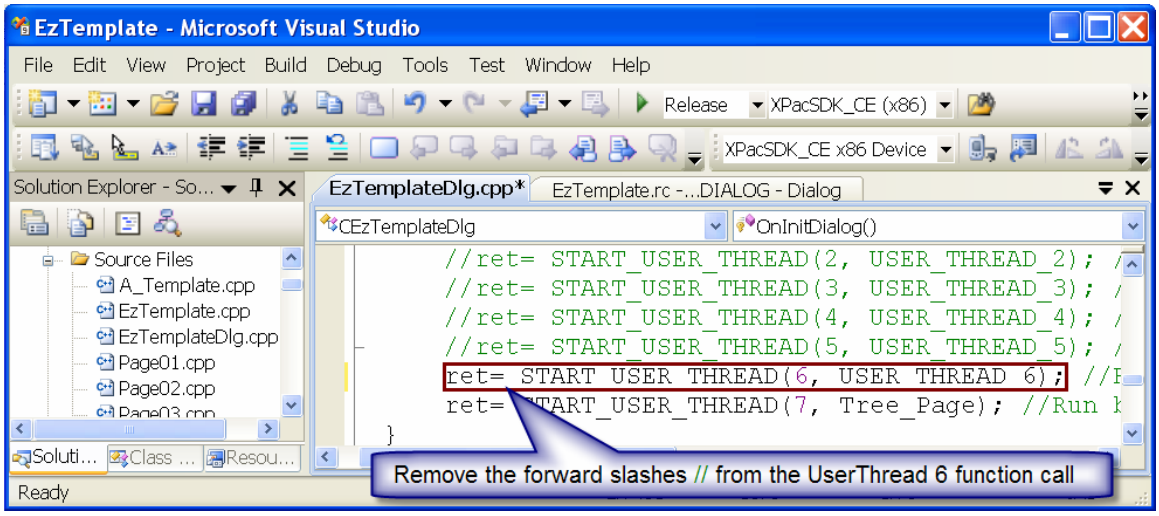
2.10.2 設計與編譯

Step1.在專案管理選 Solution 檢視，並雙擊 EzTemplateDlg.cpp 做如下設定。

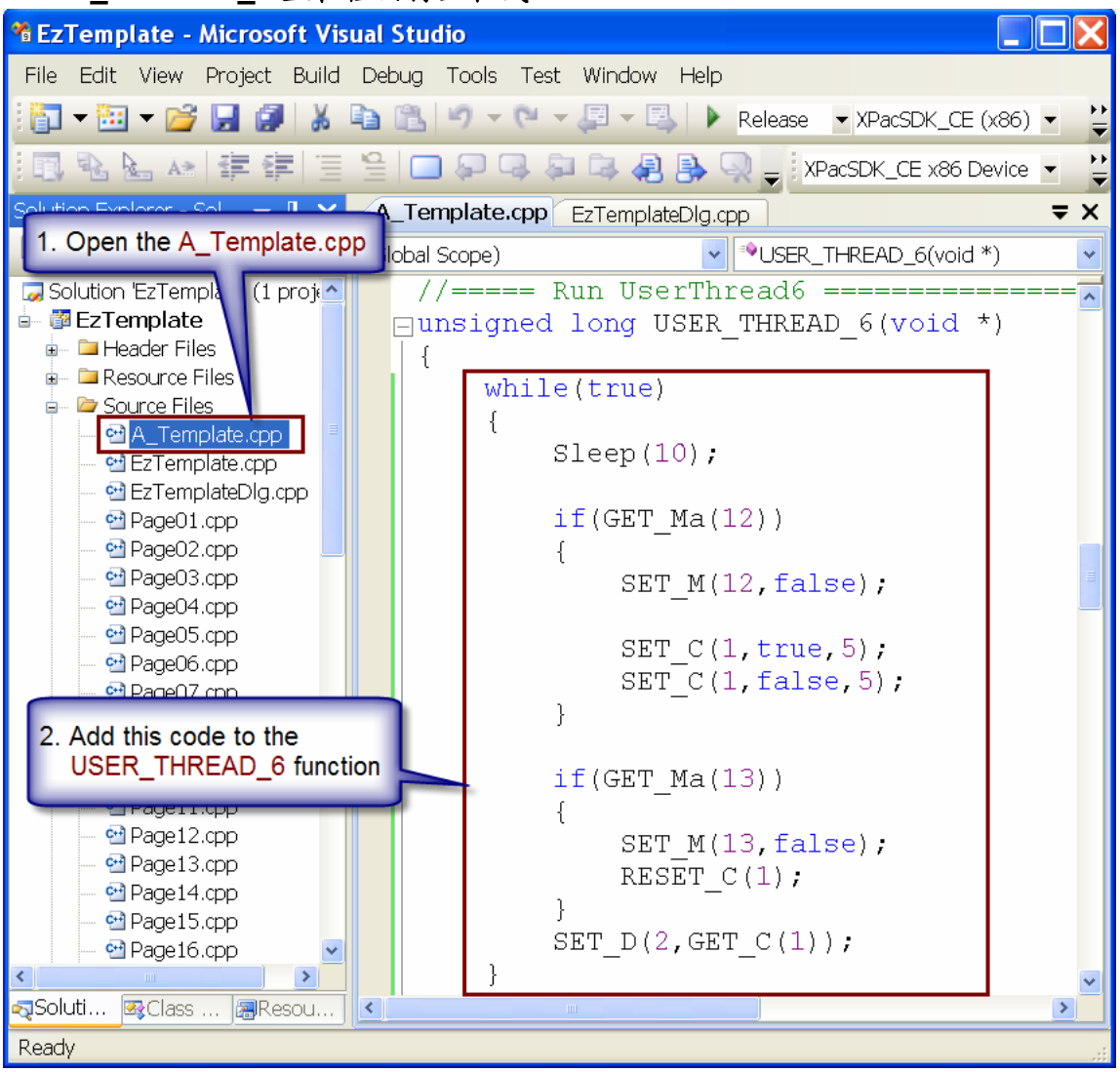
- **PageCount : 0**



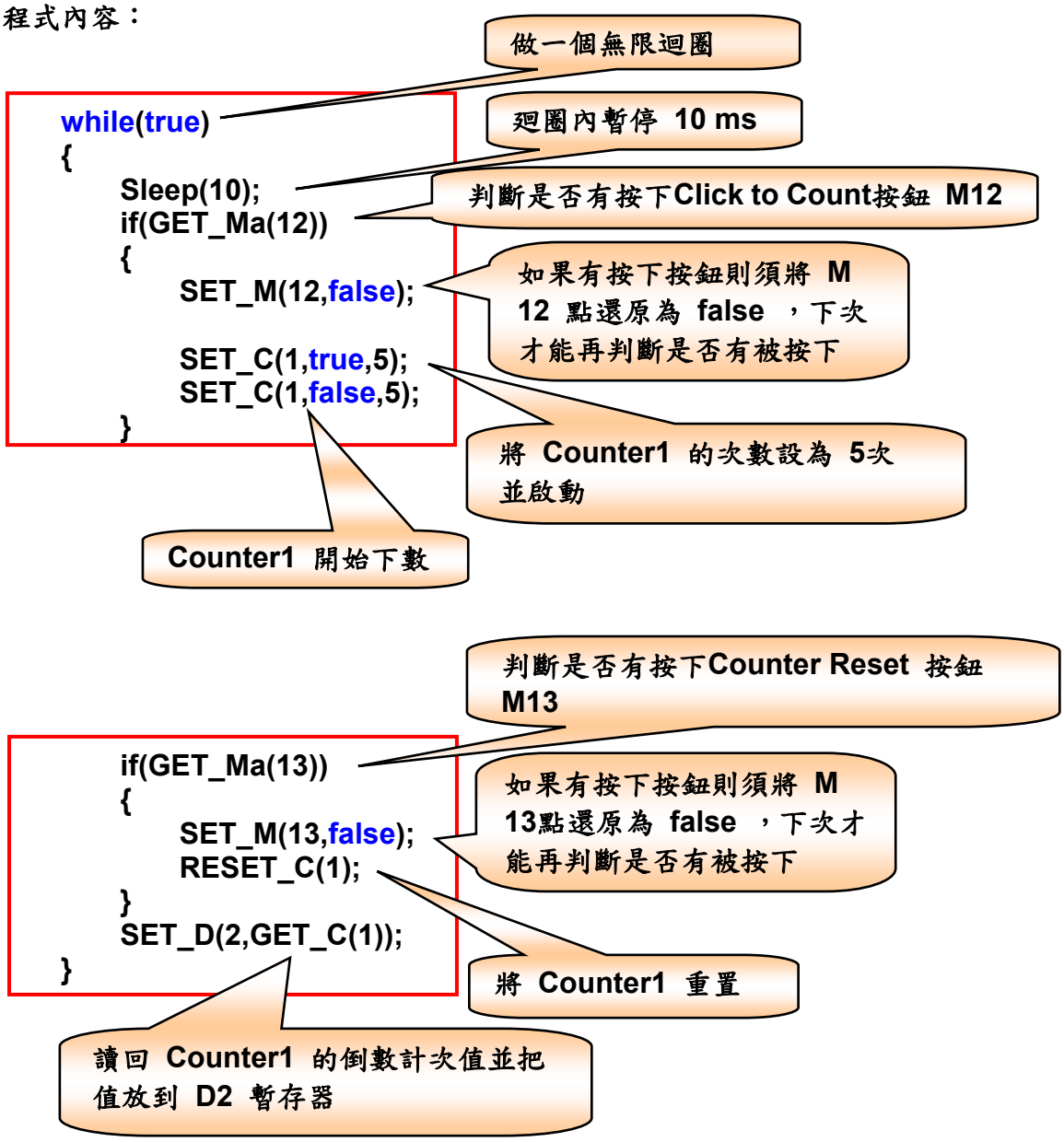
Step2.將 USER_THREAD_6 前的 // 刪掉以啟用它



Step3.在專案管理選 Solution 檢視，並雙擊 A_Template.cpp 找到 USER_THREAD_6 並在裡面打上程式



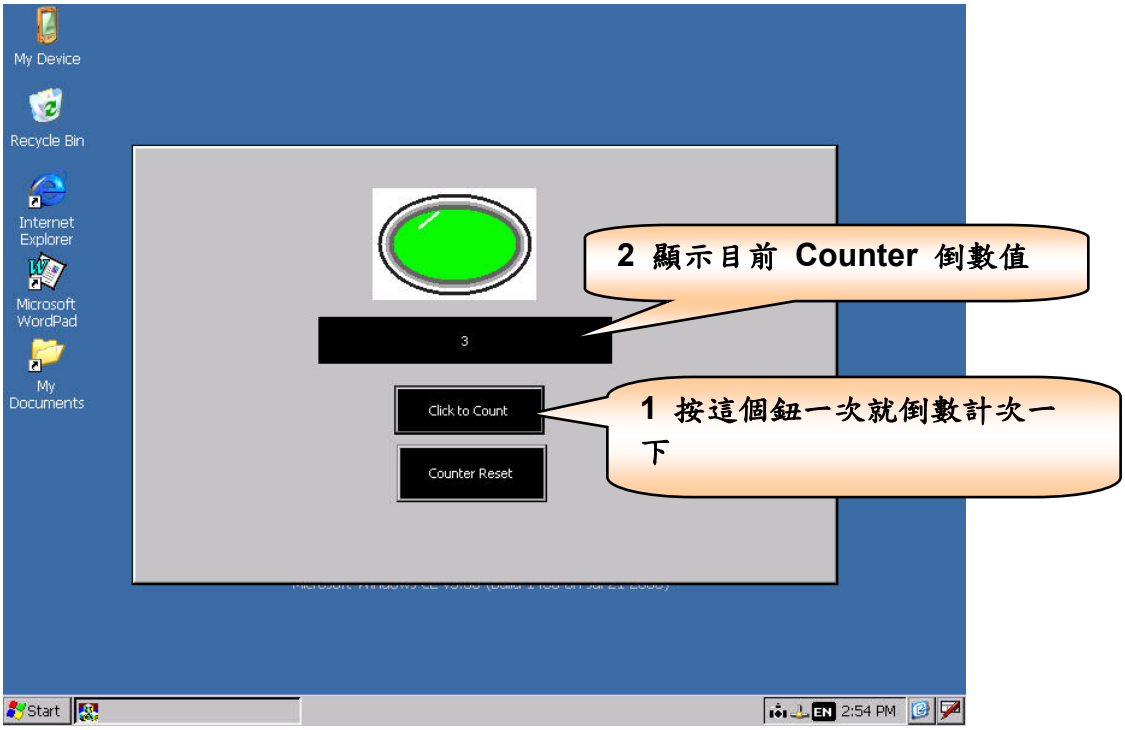
程式內容：

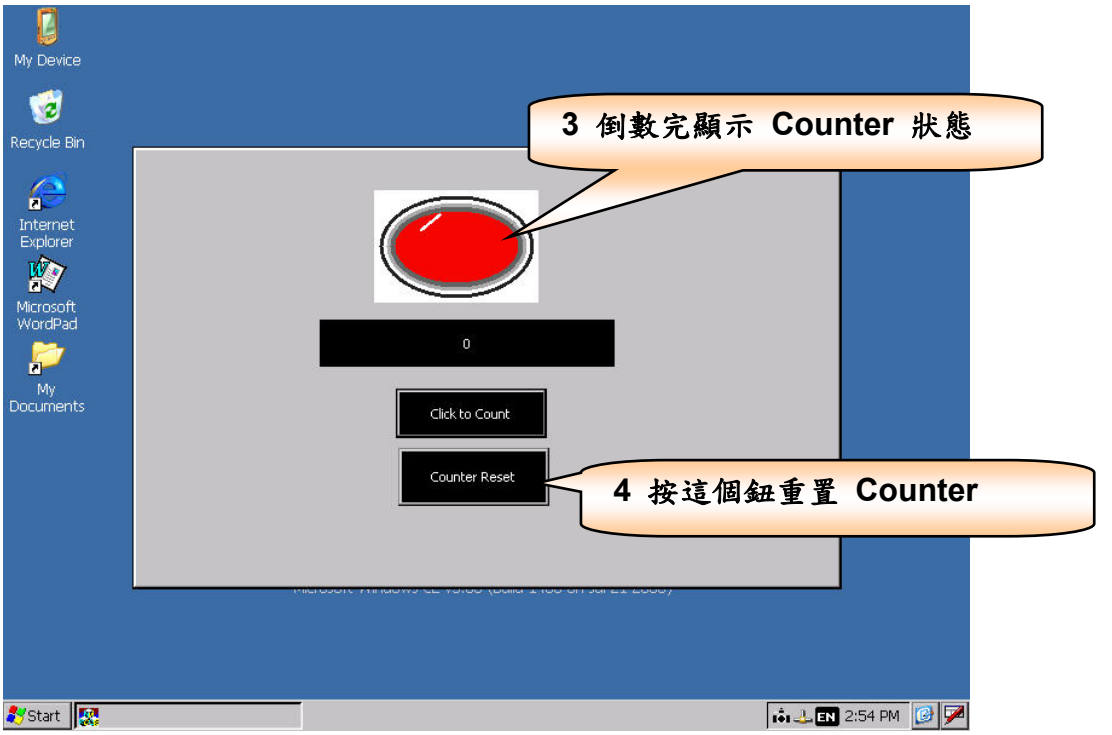



```
while(true)
{
    Sleep(10);
    if(GET_Ma(12))
    {
        SET_M(12,false);

        SET_C(1,true,5);
        SET_C(1,false,5);
    }

    if(GET_Ma(13))
    {
        SET_M(13,false);
        RESET_C(1);
    }
    SET_D(2,GET_C(1));
}
```

2.10.3 專案啟始與測試

按熱鍵 F5 連線下載執行:(請先確認網路設定是正確的) ，請參閱 EzProg-I Getting Started 手冊中附錄的連線除錯方法說明。如果沒有其他異常，在 PAC 畫面出現如下對執行畫面。





2.11 範例 AES

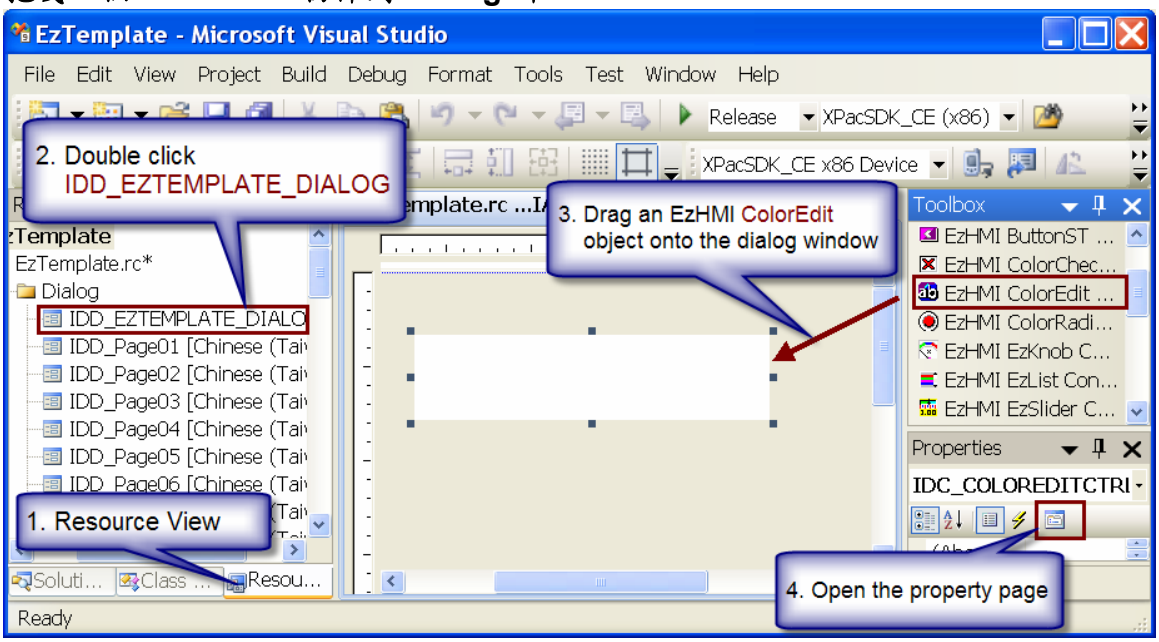
EzCore 內建一方便的 AES 加密機制功能，軟體開發者可以輕易的保護合法軟體，而此機制是利用軟體開發者指定的金鑰(AES_KEY)作加解密，並結合 PAC 的唯一硬體序號做認證。

EzCore 提供兩種簡易的方法自由選擇使用，即可操作 EzCore AES 的認證，並在 EzConfig 中提供方便的操作介面(依輸入 PAC 的唯一硬體序號及加密金鑰產生註冊碼 全功能保護著作財產權。

請依照 2.2.1 複製樣板專案，請名稱改為 AES 執行檔為 AES.exe，並開啟之。

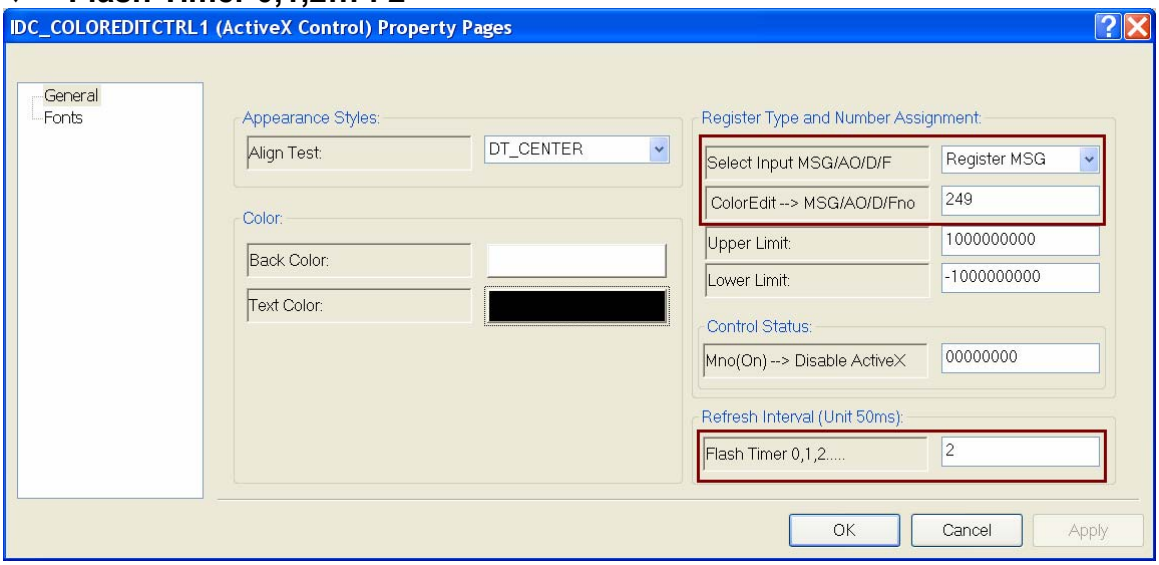
2.11.1 HMI 物件畫面控制設計

Step1.選擇 Resource View 檢視，雙擊 IDD_EZTEMPLATE_DIALOG 開啟畫面拖曳一個 ColorEdit 物件到 Dialog 中

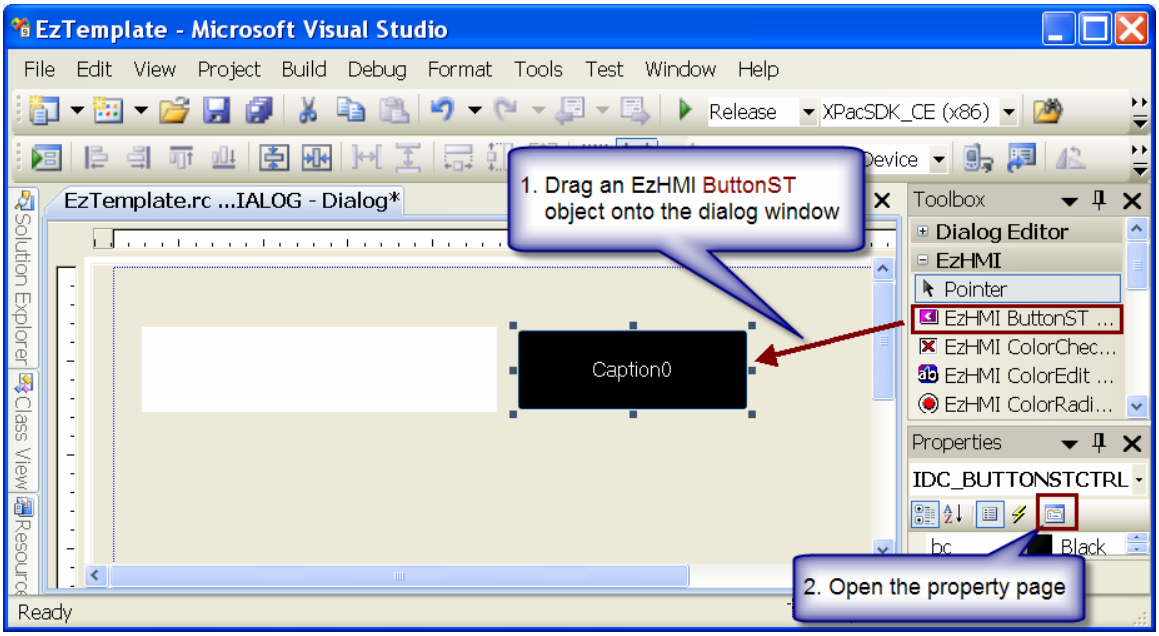


設定屬性:

- **Select Input MSG/AO/D/F : Register MSG**
- **ColorEdit→ MSG/AO/D/F : 249**
- **Flash Timer 0,1,2… : 2**

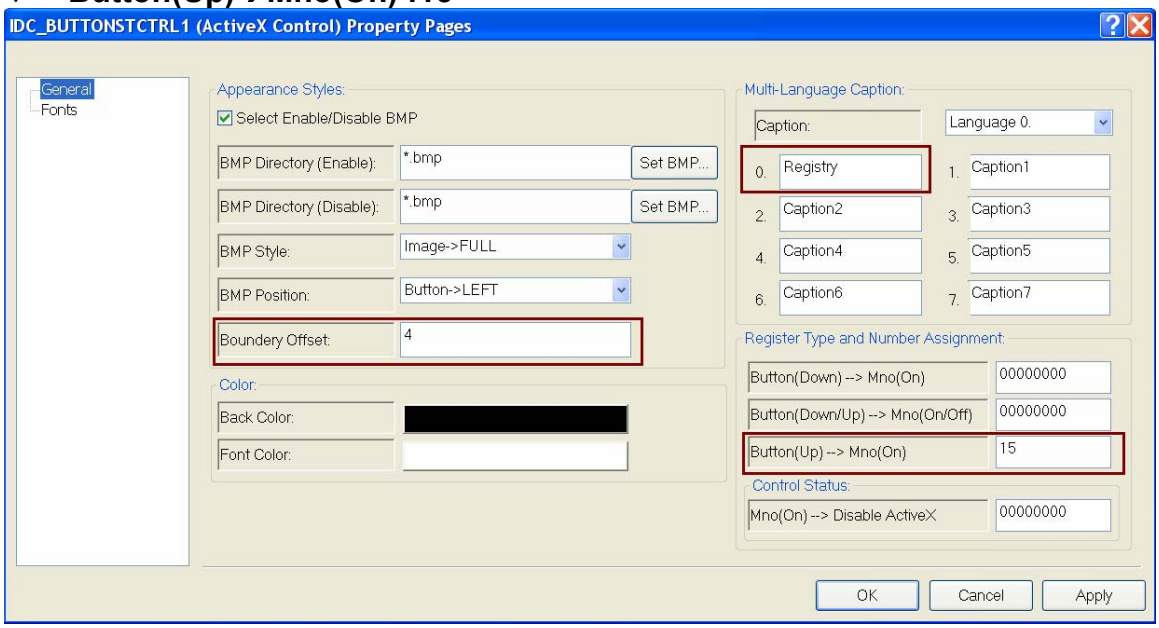


Step2.再利用滑鼠拖曳一個 ButtonST 物件到 Dialog 中如下

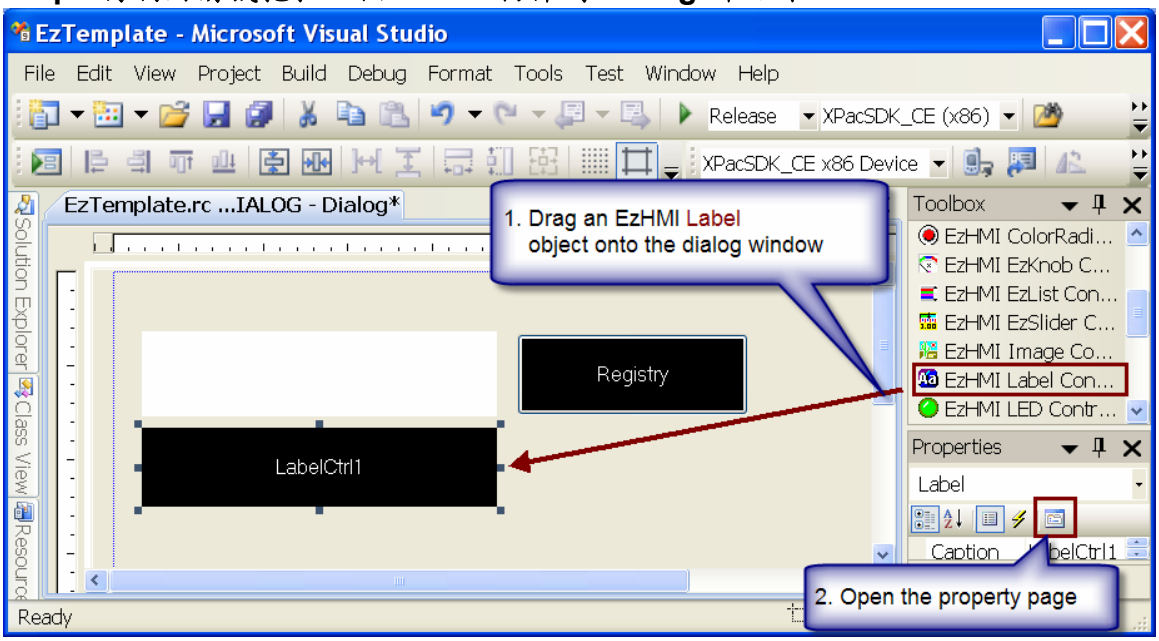


設定屬性:

- **Boundery Offset : 4**
- **Caption : Registry**
- **Button(Up)→Mno(On) :15**

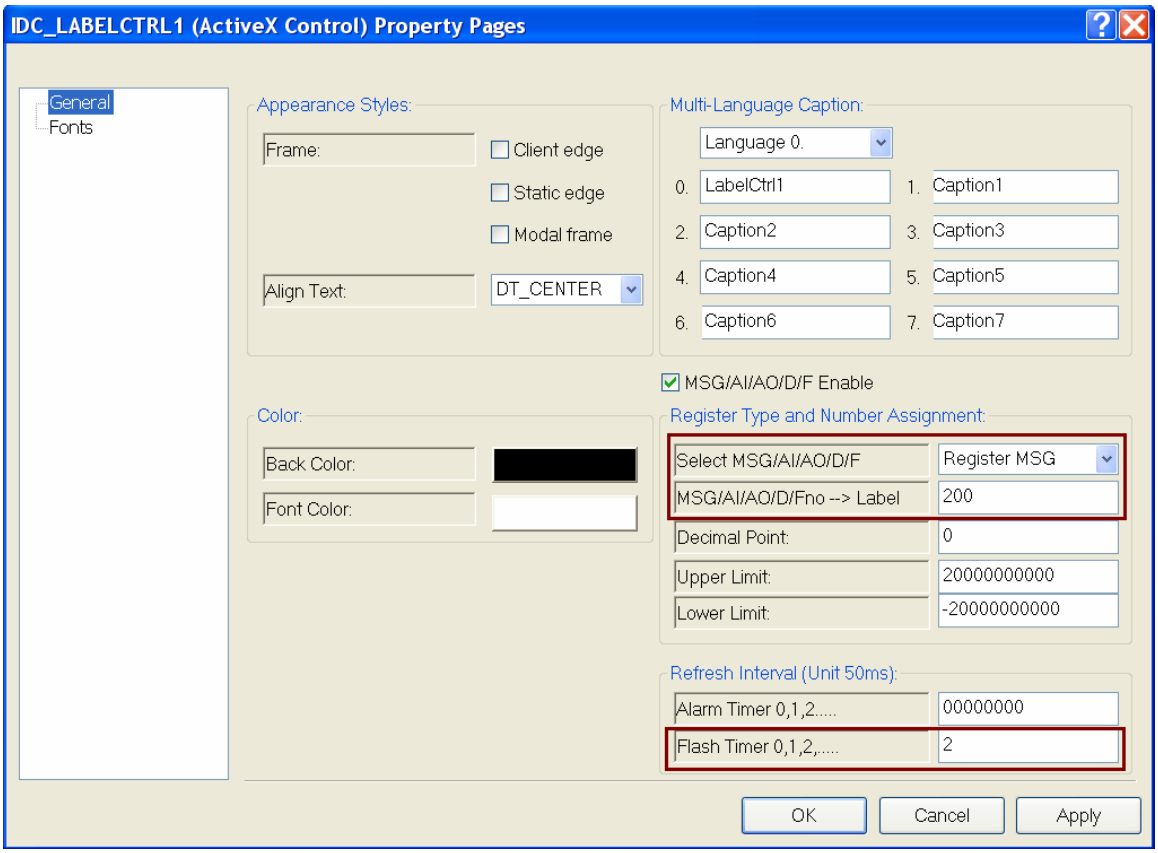


Step3.再利用滑鼠拖拉一個 Label 物件到 Dialog 中如下



設定屬性:

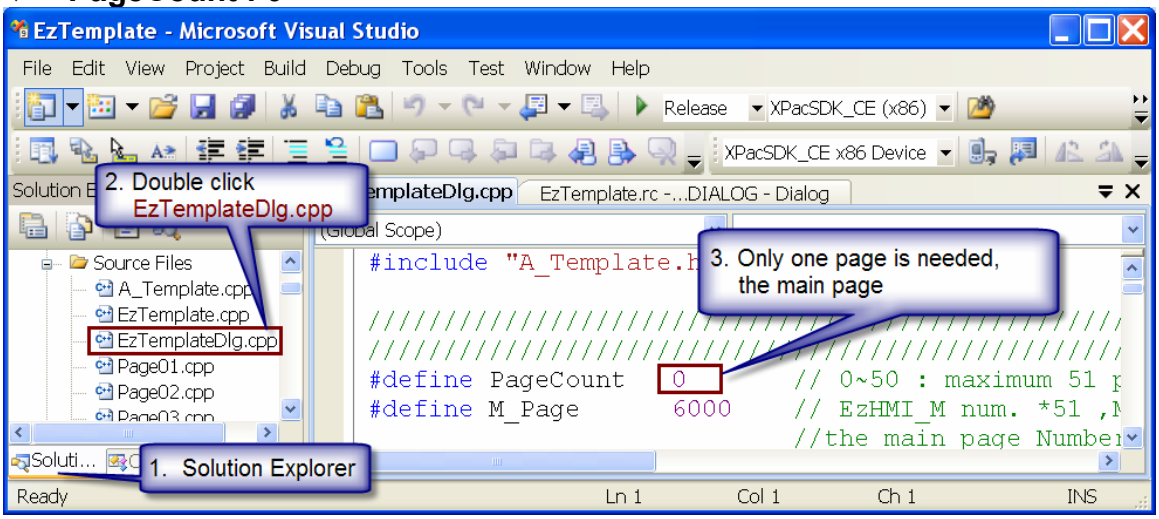
- **Select MSG/AI/AO/D/F : Register MSG**
- **MSG/AI/AO/D/Fno→Label : 200**



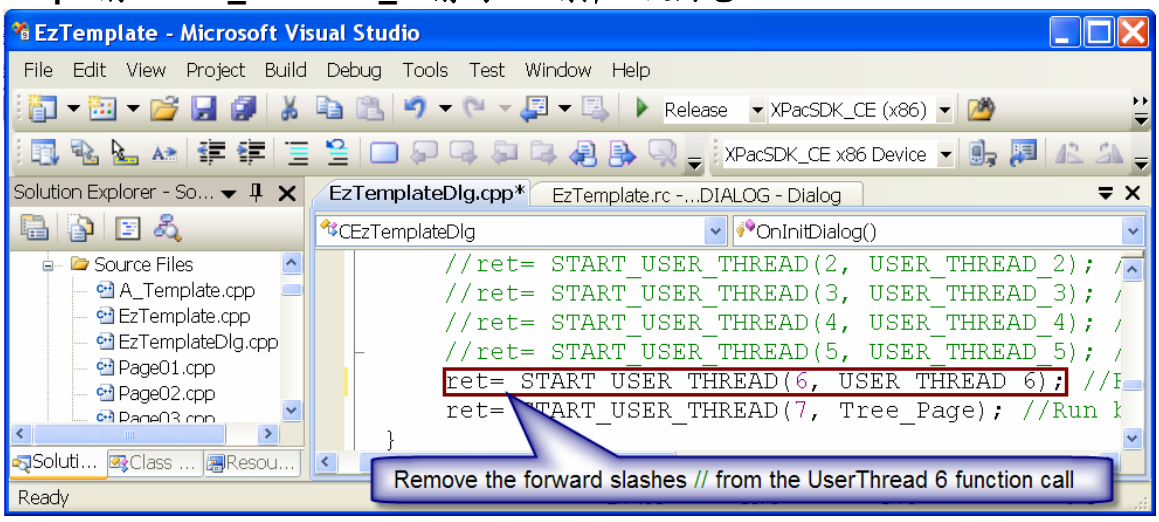
2.11.2 設計與編譯

Step1.在專案管理選 Solution 檢視，並雙擊 EzTemplateDlg.cpp 做如下設定。

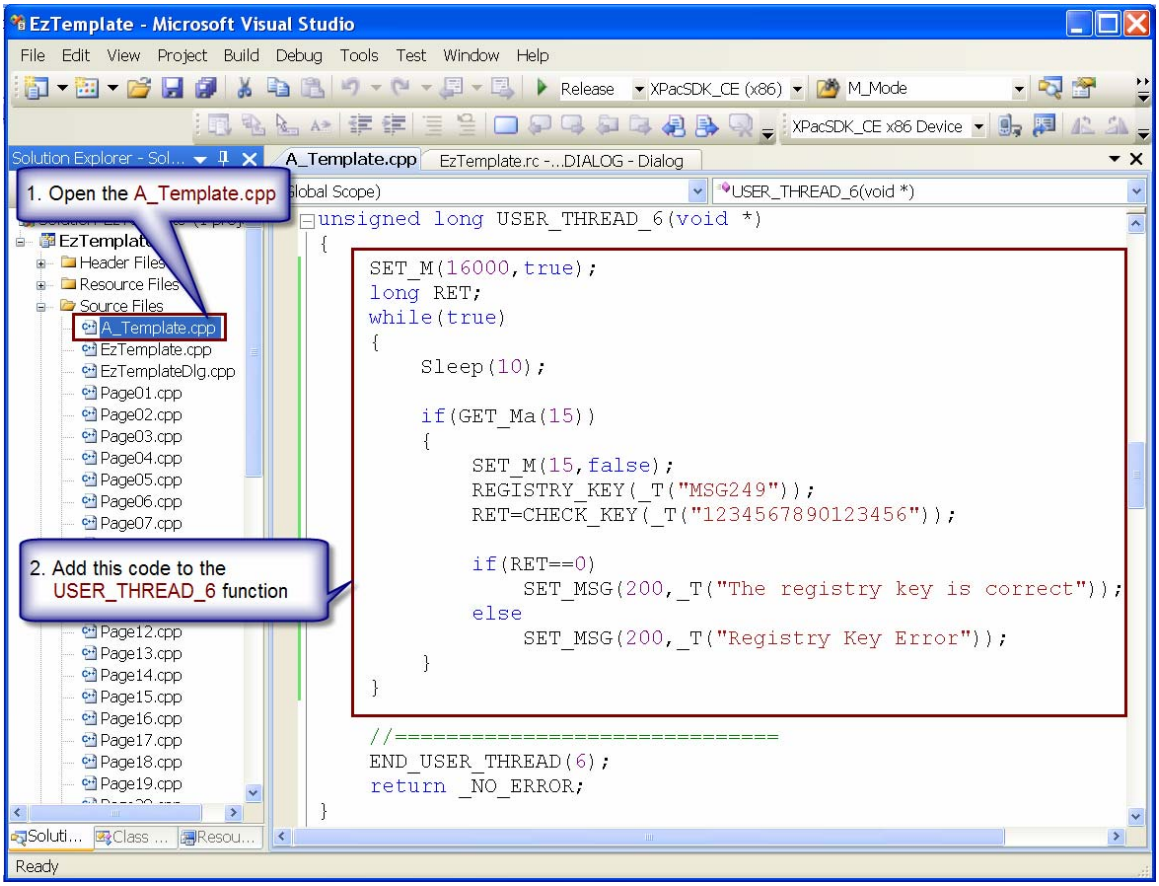
- **PageCount : 0**



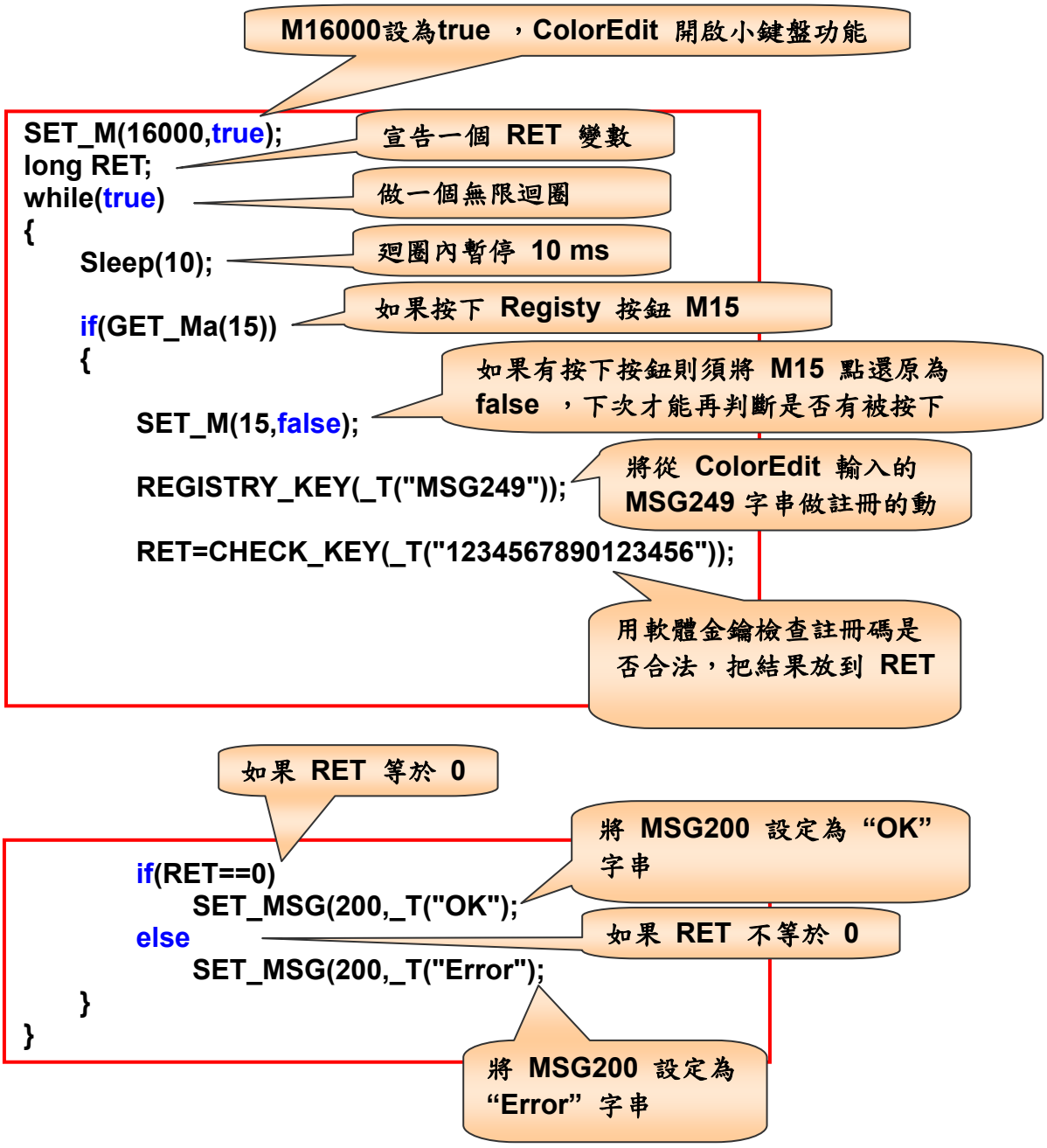
Step2.將 USER_THREAD_6 前的 // 刪掉以啟用它



Step3.在專案管理選 Solution 檢視，並雙擊 A_Template.cpp 找到 USER_THREAD_6 並在裡面打上程式

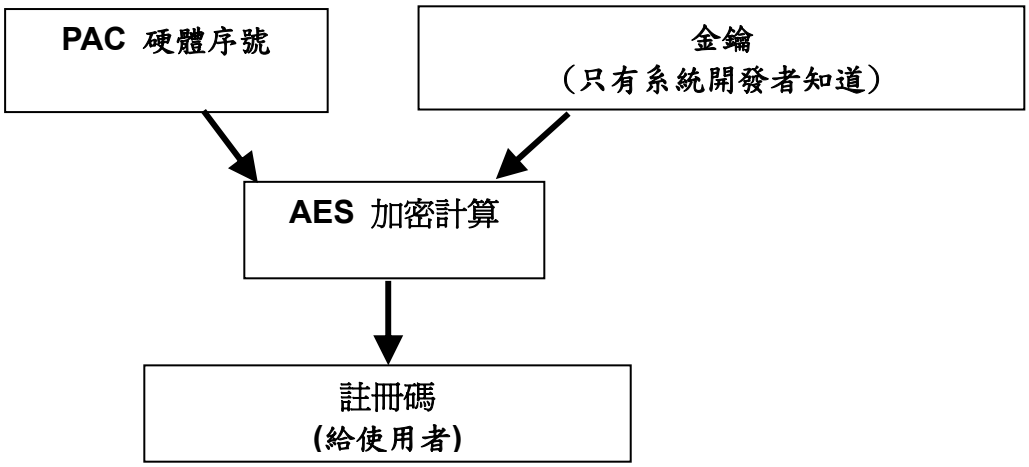


程式內容：

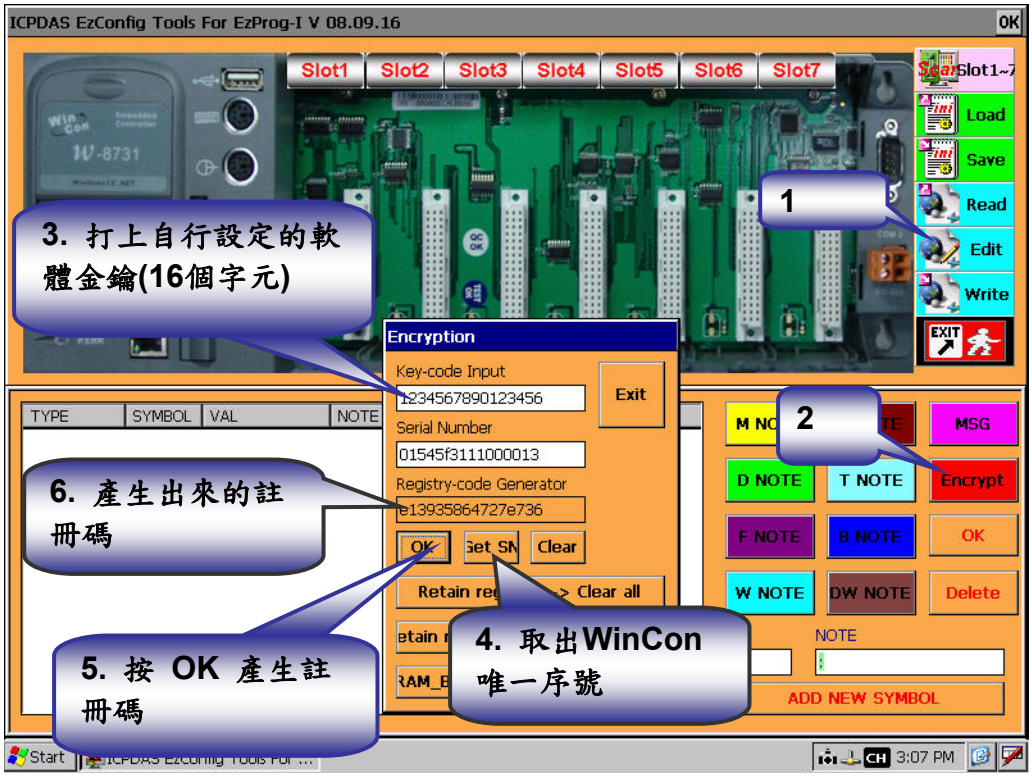

M16000設為true ，ColorEdit 開啟小鍵盤功能
SET_M(16000,true);
long RET;
while(true)
{
Sleep(10);
if(GET_Ma(15))
{
SET_M(15,false);
REGISTRY_KEY(_T("MSG249"));
RET=CHECK_KEY(_T("1234567890123456"));
宣告一個 RET 變數
做一個無限迴圈
迴圈內暫停 10 ms
如果按下 Registy 按鈕 M15
如果有按下按鈕則須將 M15 點還原為 false ，下次才能再判斷是否有被按下
將從 ColorEdit 輸入的 MSG249 字串做註冊的動
用軟體金鑰檢查註冊碼是否合法，把結果放到 RET
如果 RET 等於 0
if(RET==0)
SET_MSG(200,_T("OK");
else
SET_MSG(200,_T("Error");
}
}
將 MSG200 設定為 “OK” 字串
如果 RET 不等於 0
將 MSG200 設定為 “Error” 字串

2.11.3 專案啟始與測試

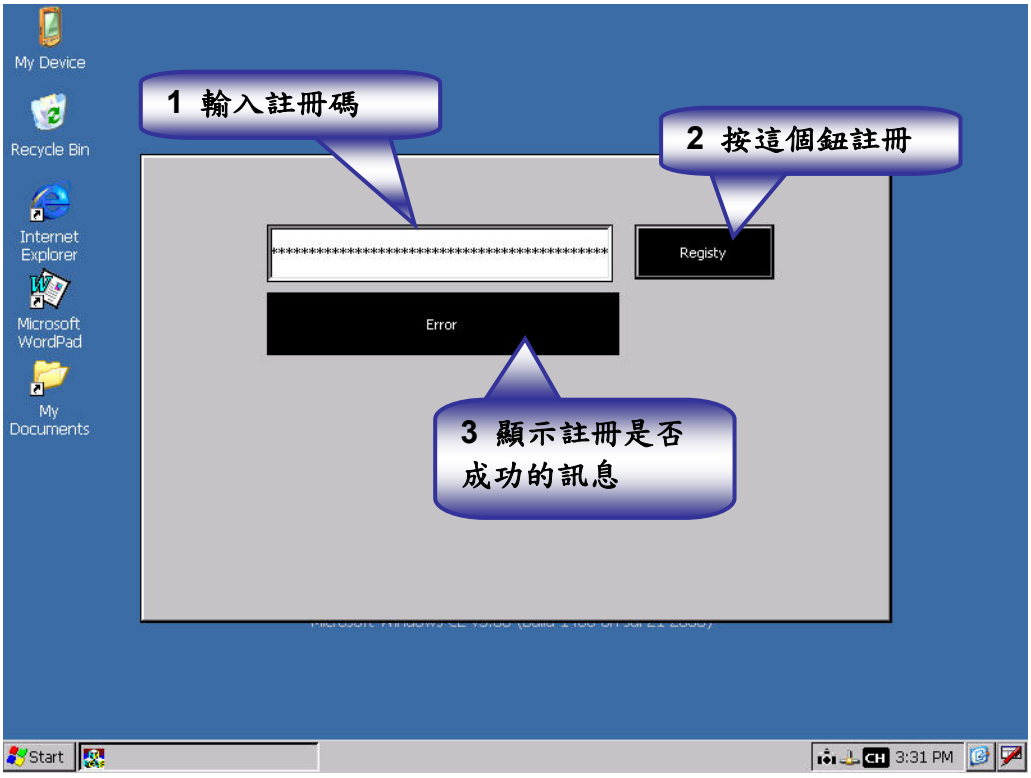
註冊碼是由 PAC 的唯一硬體序號及由系統開發者自行設定的金鑰兩者經過 AES 計算而產生的



在開始測試之前請先利用 EzConfig 產生註冊碼。



按熱鍵 F5 連線下載執行:(請先確認網路設定是正確的)，請參閱 EzProg-I Getting Started 手冊中附錄的連線除錯方法說明。如果沒有其他異常，在 PAC 畫面出現如下對執行畫面。



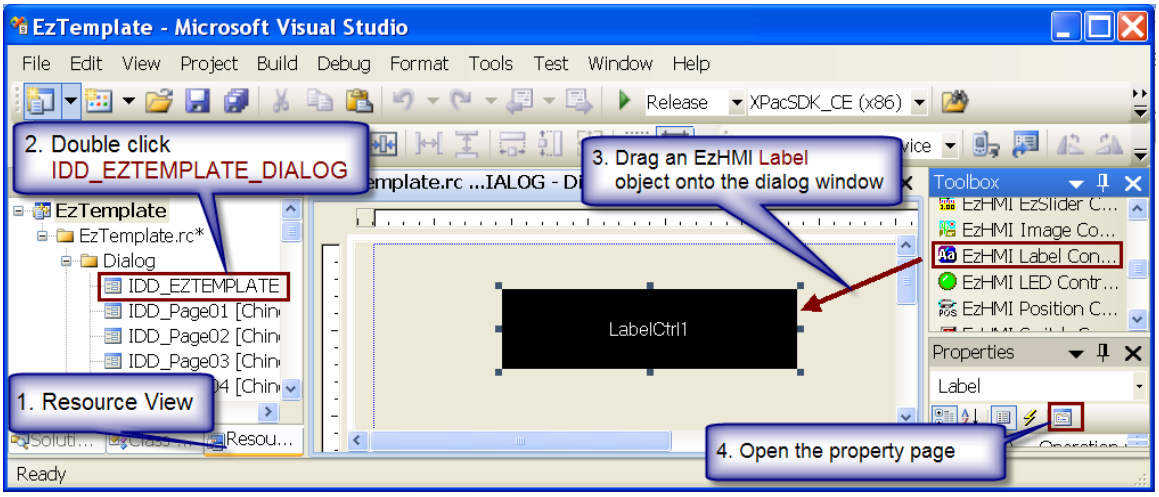
2.12 範例 MSG MSGF

EzCore 提供多國語系訊息，可以讓系統依不同語系使用不同訊息，可以用於應用程序與人機界面多國語系切換用，當 D8000 被修改時，稍後立即自行變更。而多國語系檔案用 Unicode 分別存為 ML0.txt，ML1.txt ~ML7.txt，分別對應 Multi-Language 0~7 八個語系文字檔。

請依照 2.2.1 複製樣板專案，請名稱改為 MSGF 執行檔為 MSGF.exe，並開啟之。

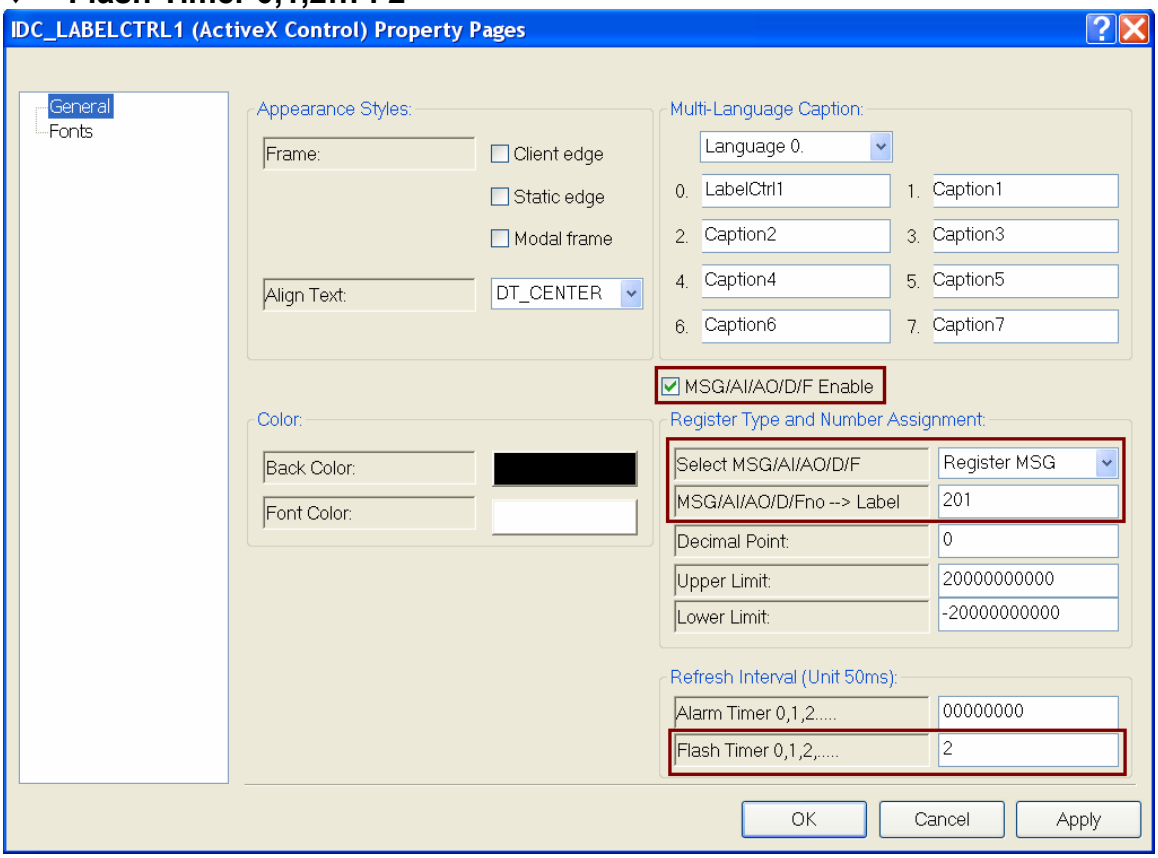
2.12.1 HMI 物件畫面控制設計

Step1.選擇 Resource View 檢視，雙擊 IDD_EZTEMPLATE_DIALOG 開啟畫面拖曳一個 Label 物件到 Dialog 中

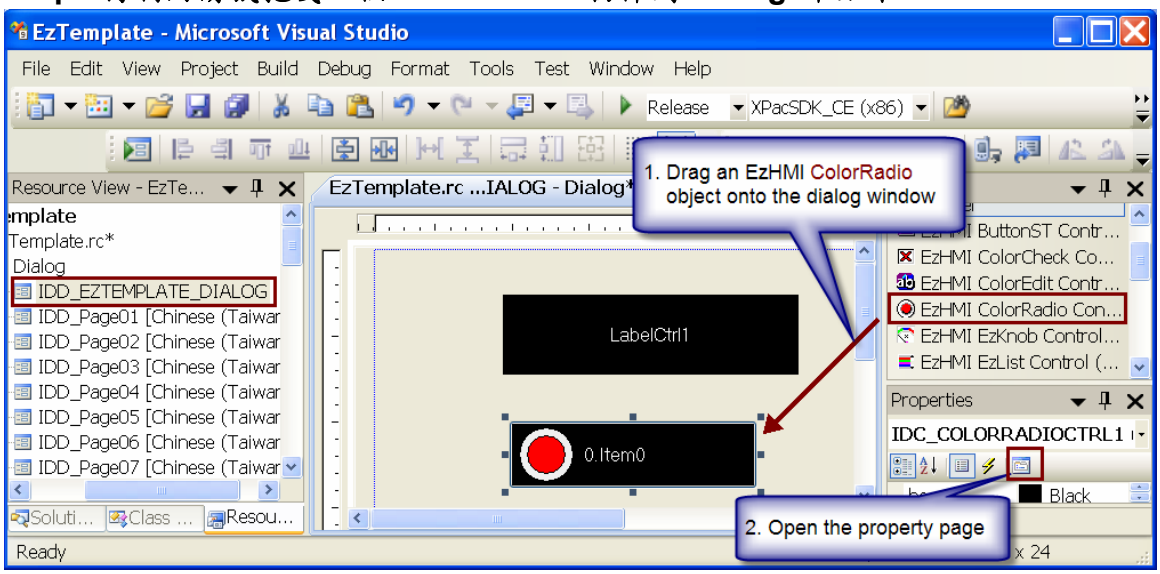


設定屬性:

- **MSG/AI/AO/D/F Enable : 選取**
- **Select MSG/AI/AO/D/F : Register MSG**
- **MSG/AI/AO/D/Fno→Label : 201**
- **Flash Timer 0,1,2… : 2**

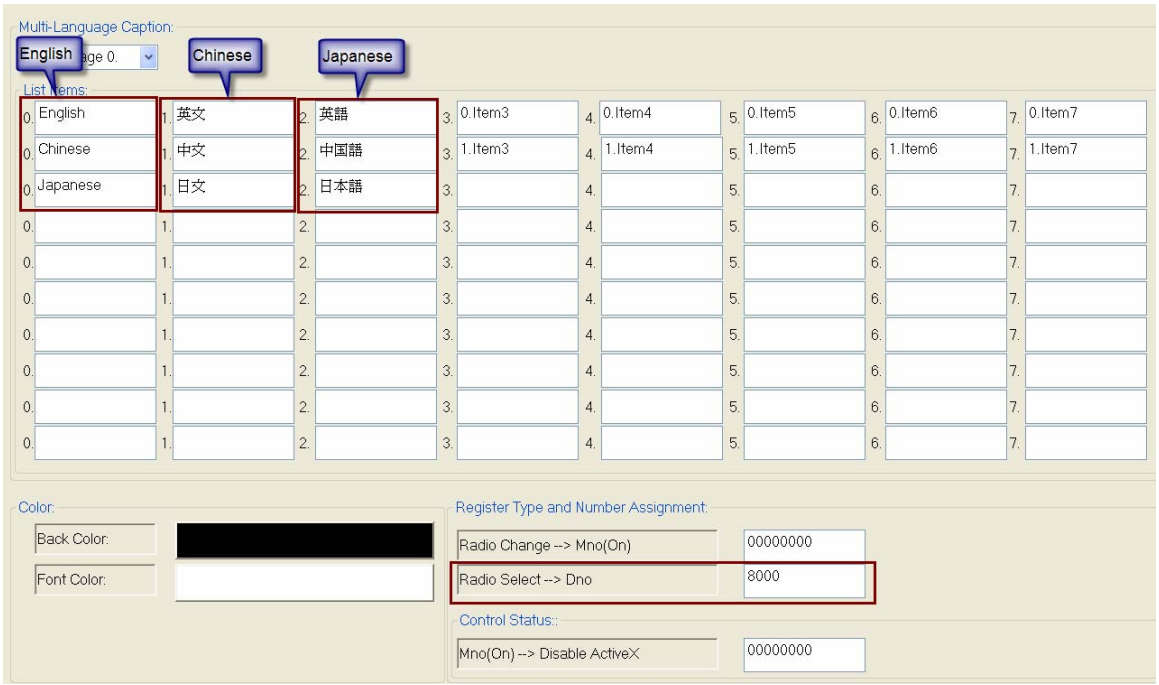


Step2.再利用滑鼠拖曳一個 ColorRadio 物件到 Dialog 中如下

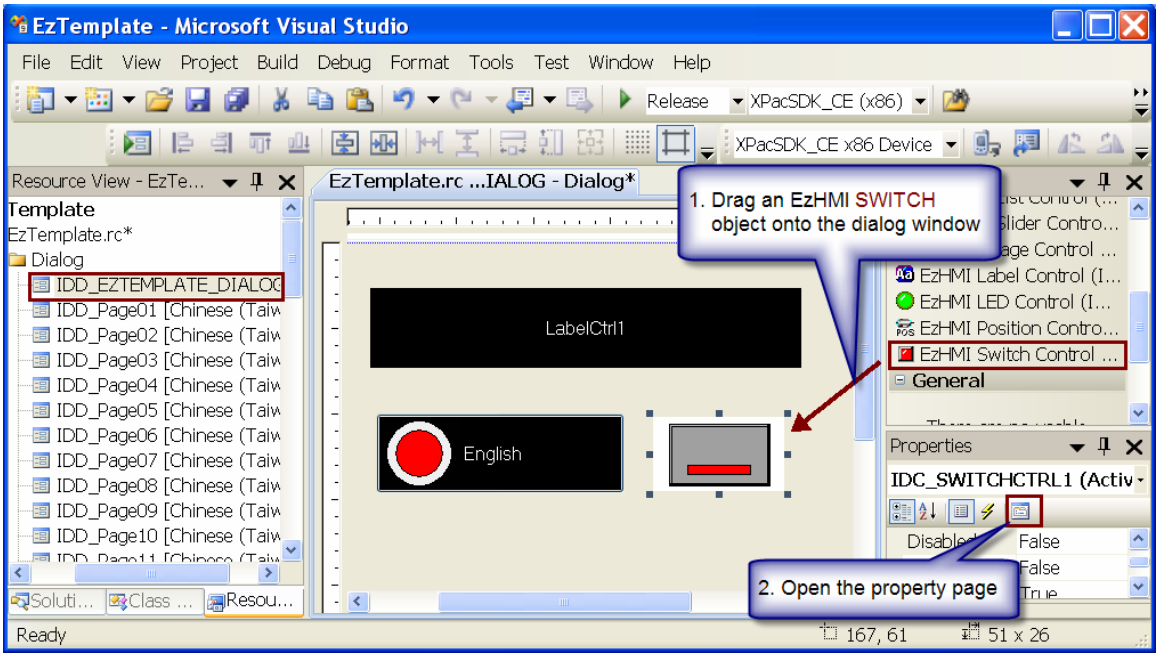


設定屬性:

- **Item 0 : “English” , “Chinese” , “Japnese” (貼上英文選項)**
- **Item 1 : “英文”, “中文” , “日文” (貼上中選項)**
- **Item 2 : “英語”, “中国語”, “日本語” (貼上日文選項)**
- **Radio Select→Dno : 8000**

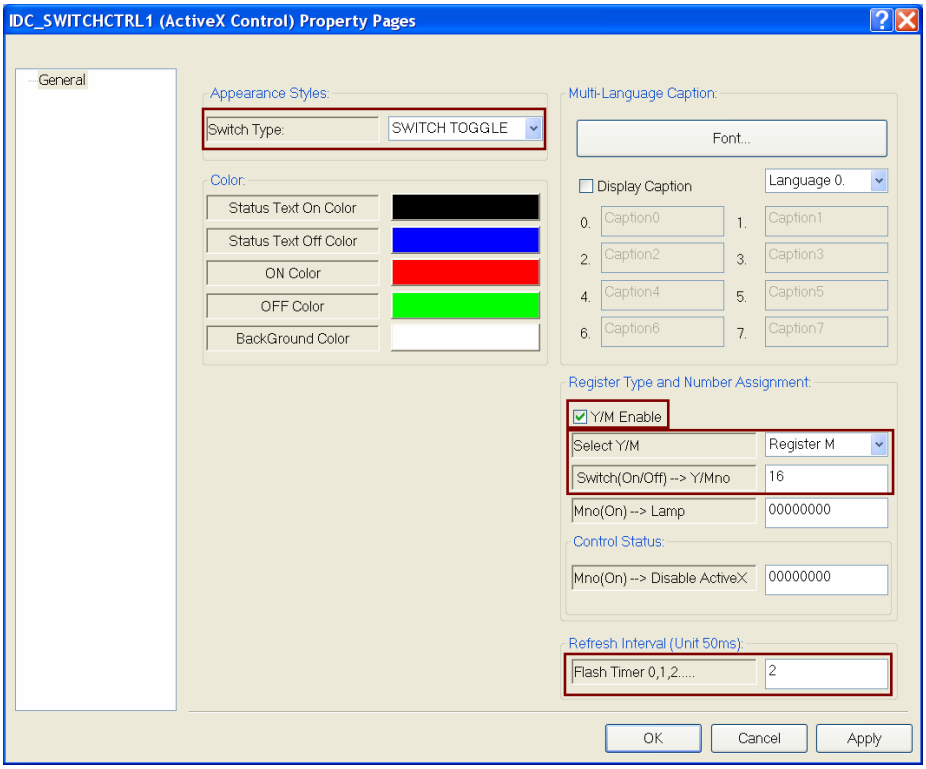


Step3.再利用滑鼠拖曳一個 Switch 物件到 Dialog 中如下



設定屬性:

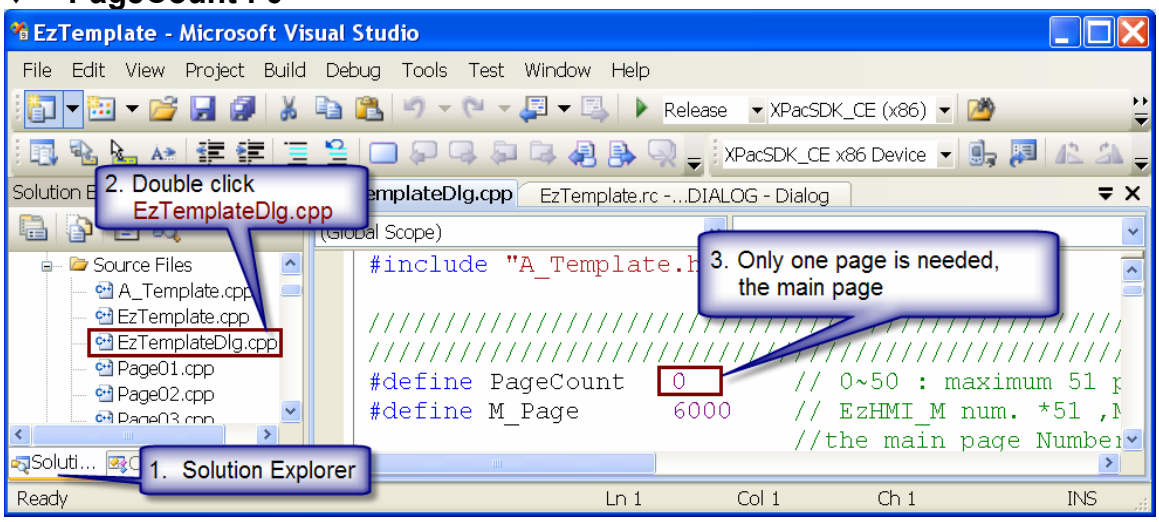
- **Switch Type : SWITCH TOGGLE**
- **Y/M Enable : 選取**
- **Select Y/M : Register M**
- **Switch(On/Off)→Y/Mno : 16**
- **Flahs Timer : 2**



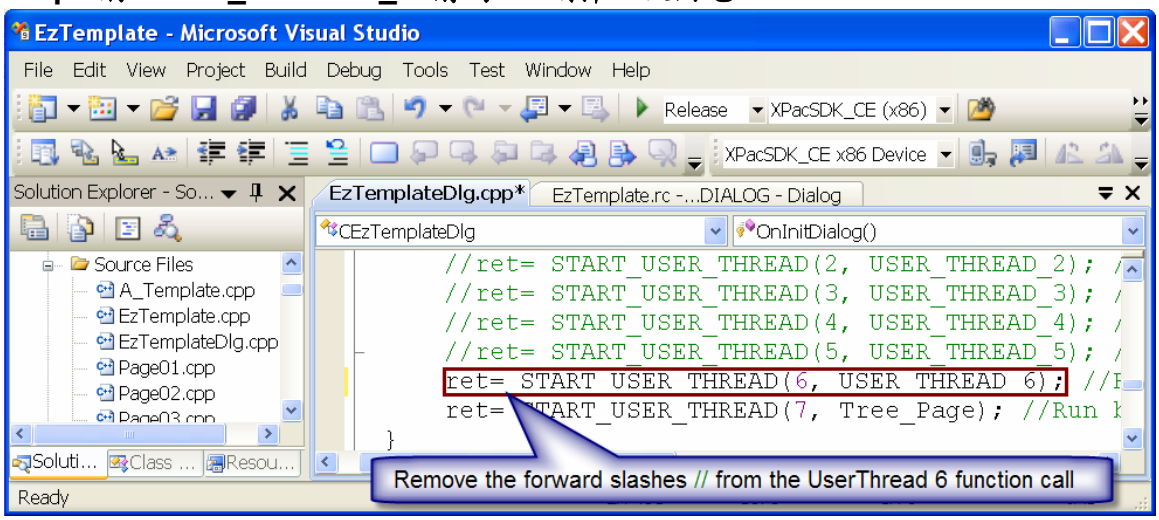
2.12.2 設計與編譯

Step1.在專案管理選 Solution 檢視，並雙擊 EzTemplateDlg.cpp 做如下設定。

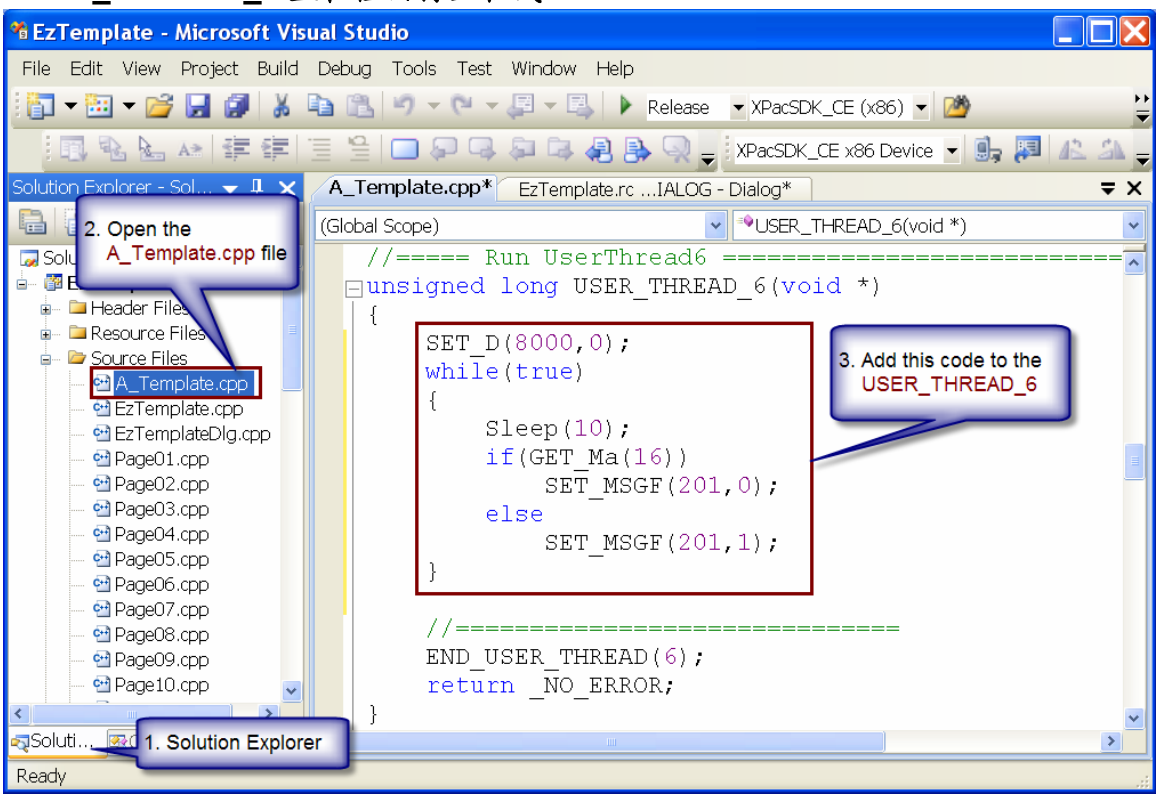
- **PageCount : 0**



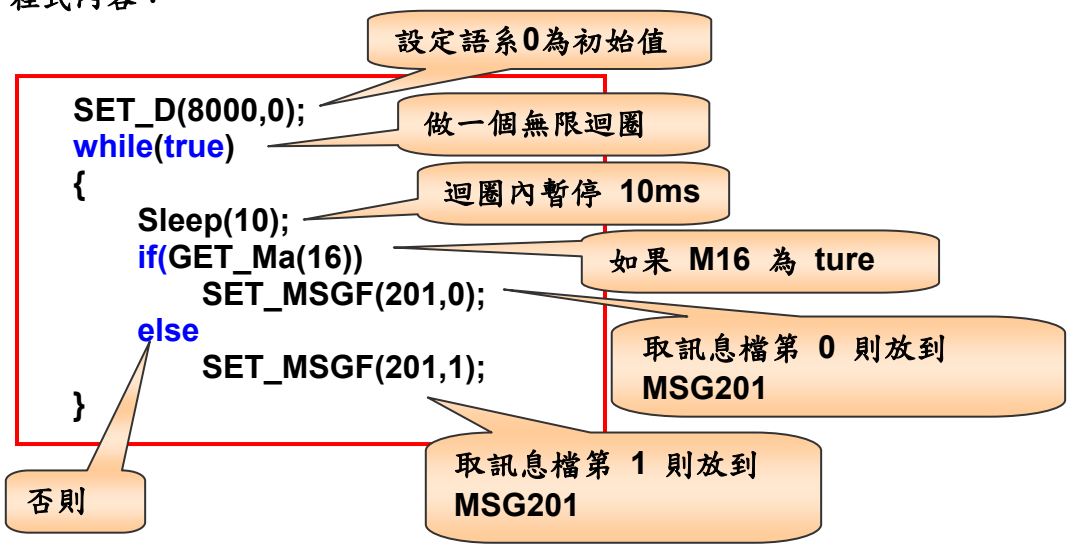
Step2.將 USER_THREAD_6 前的 // 刪掉以啟用它



Step3.在專案管理選 Solution 檢視，並雙擊 A_Template.cpp 找到 USER_THREAD_6 並在裡面打上程式



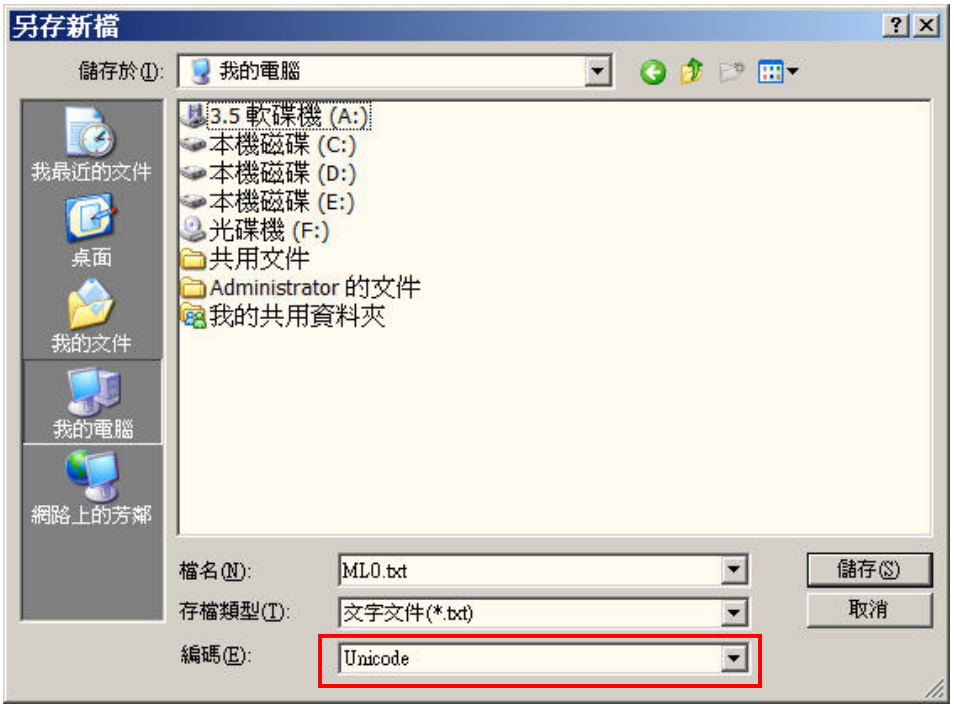
程式內容：



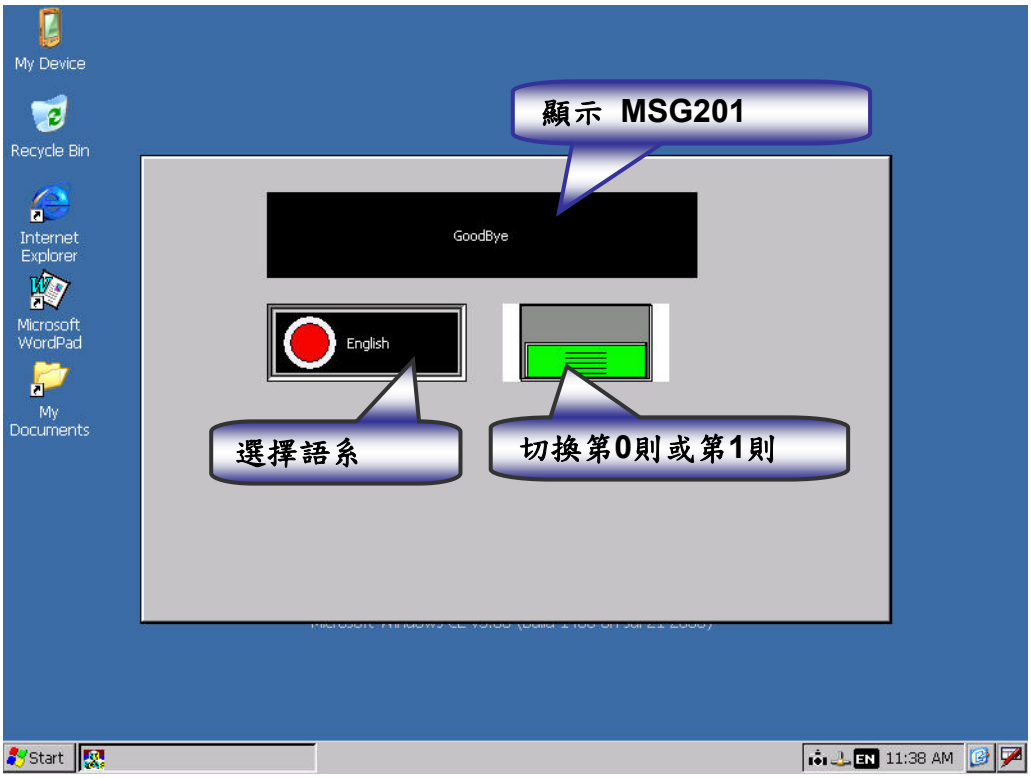
2.12.3 專案啟始與測試

先建立好訊息的文字檔，並存放到 PAC 中的 EzProg_Path\EzProg-I\EzHMI\ML。
使用記事本打上三種語系的文字

依序將檔名存為 ML0.txt ,ML1.txt ,ML2.txt，編碼必須選 Unicode，



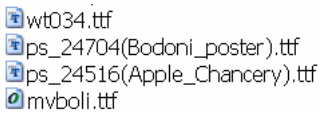
按熱鍵 F5 連線下載執行:(請先確認網路設定是正確的)，連線除錯方法請參閱附錄連線除錯方法,如果沒有其他異常，在 PAC 畫面出現如下對執行畫面。



2.13 範例 字型

本範例是以一個 HMI 頁面應用範例，，主要在說明如何用 EzTemplate 專案範本設計建立含多種字型操作專案。

WinCE 支援字型並不完全與 Windows 一樣，所以使用時要測是否可以正常顯示，本範例有使用三種 Windows(位於 C:\WINDOWS\Fonts)字型檔分別是:

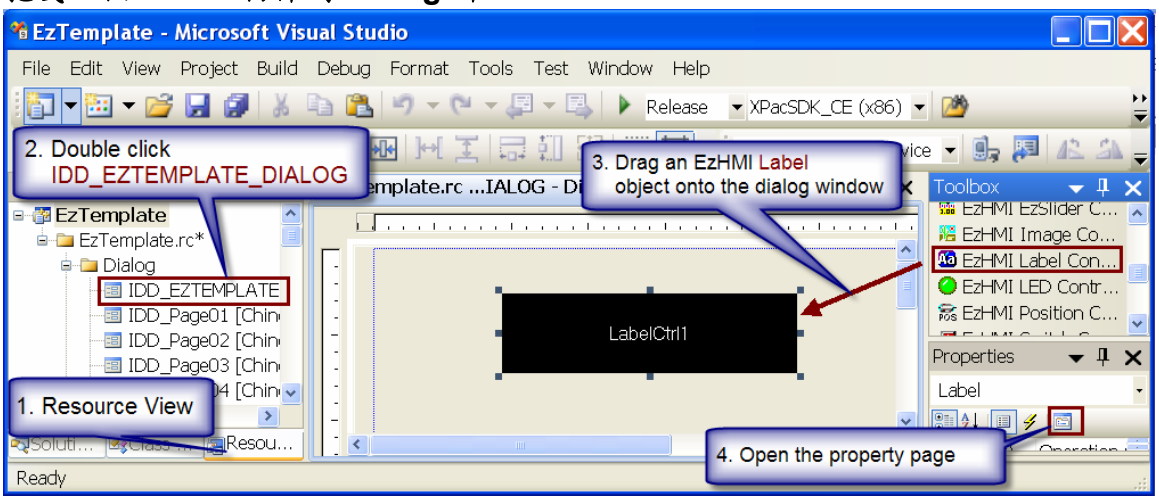


mvboli.ttf ，ps_24516(Apple_Chancery).ttf，ps_24704(Bodoni_poster).ttf，使用時清請先將字形檔複製到 EzProg_Path\ICPDAS\MAinit\. 中並重新開機

請依照 2.2.1 複製樣板專案，請名稱改為 EzTemplate_FONT,執行檔為 EzTemplate_FONT.exe，並開啟之。

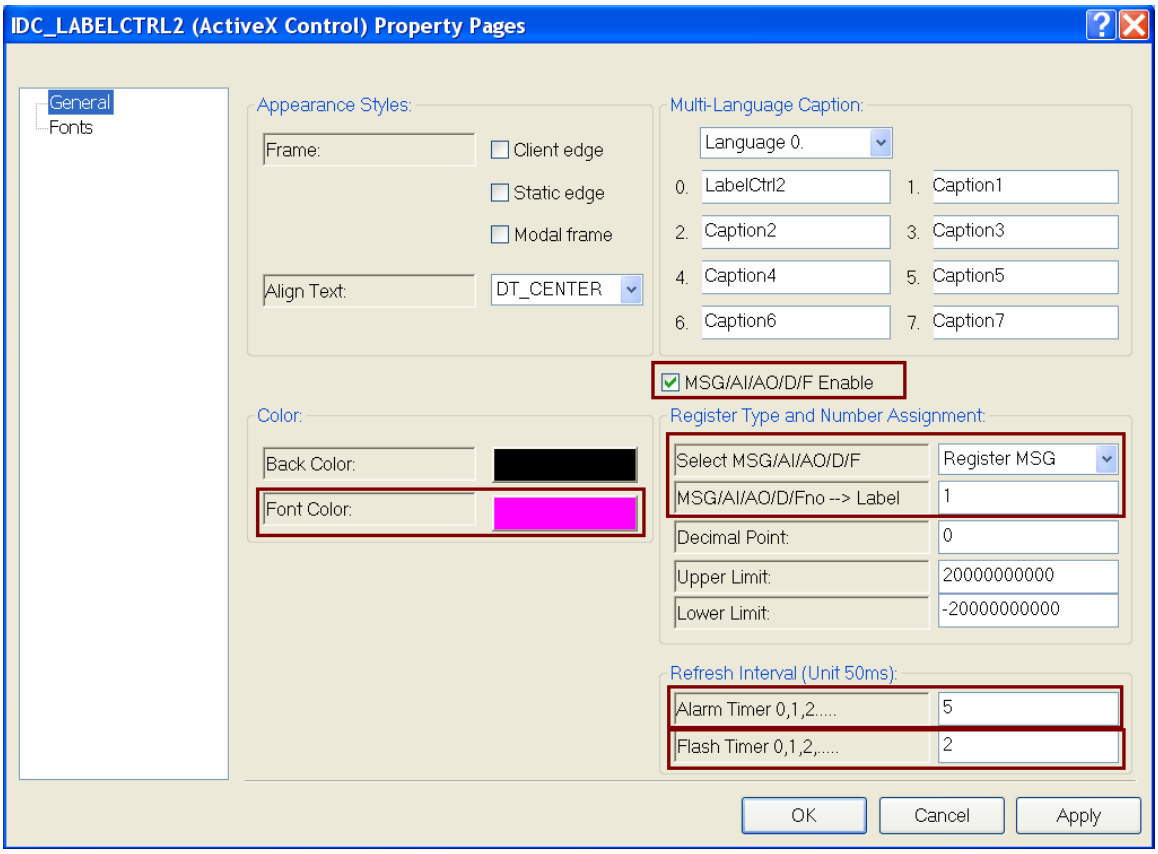
2.13.1 HMI 物件畫面控制設計

Step1.選擇 Resource View 檢視，雙擊 IDD_EZTEMPLATE_DIALOG 開啟畫面拖曳一個 Label 物件到 Dialog 中



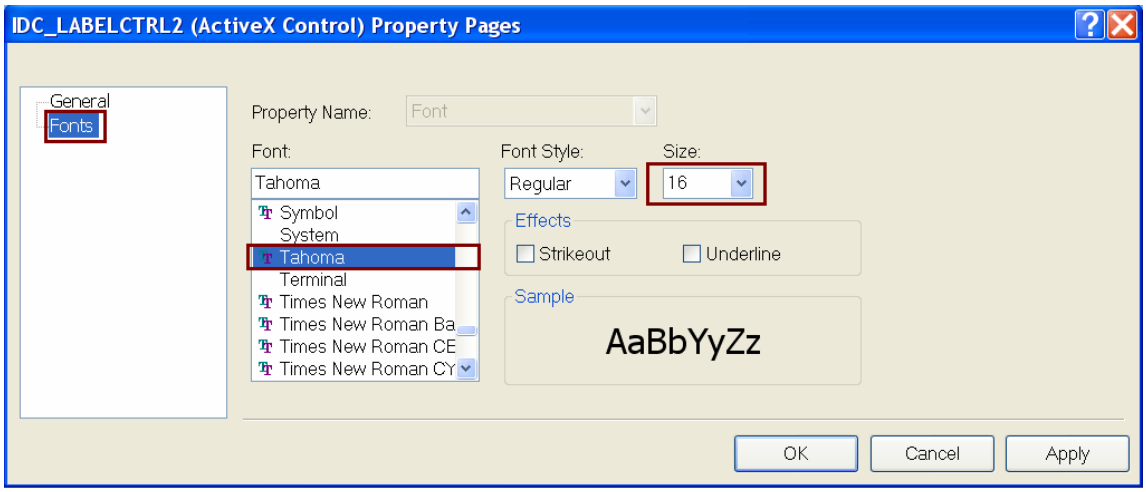
設定屬性

- **MSG/AI/AO/D/F Enable :選取**
- **Select MSG/AI/AO/D/F : Register MSG**
- **MSG/AI/AO/D/F→Label : 1**
- **Alamr timer 0,1,2… : 5**
- **Flash Timer 0,1,2… : 2**
- **Font color :**

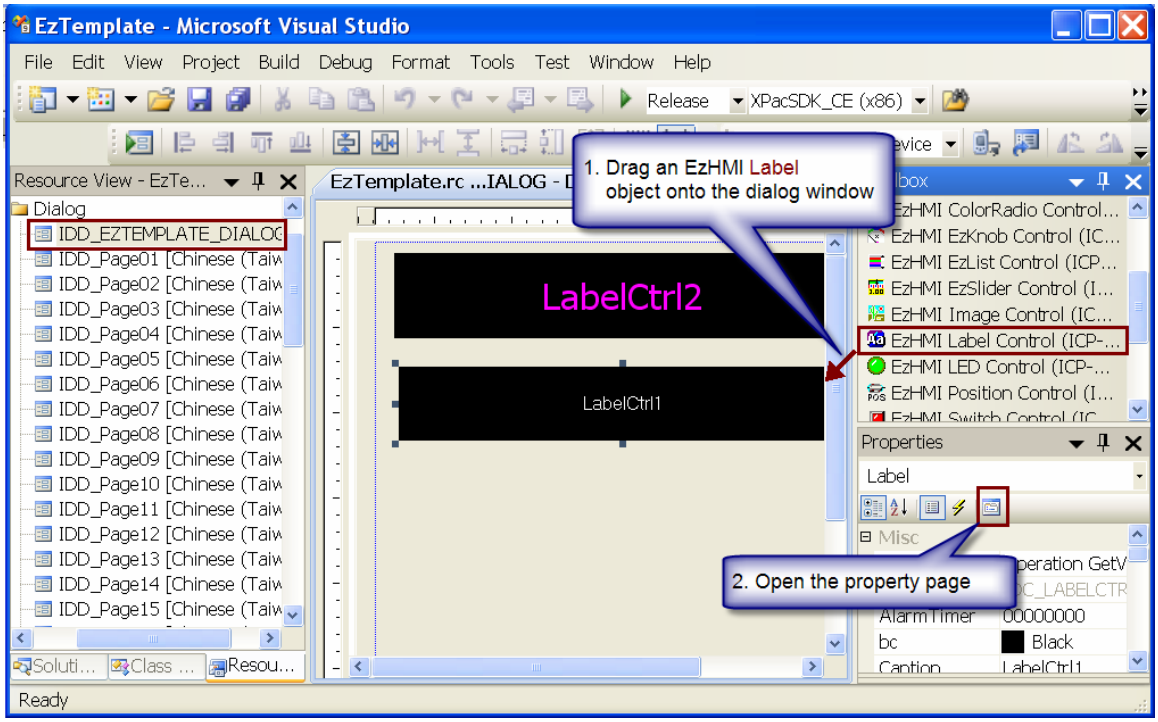


點擊左邊 “Fonts” 切換到字型設定

- **Font : Tahoma**
- **Size : 16**

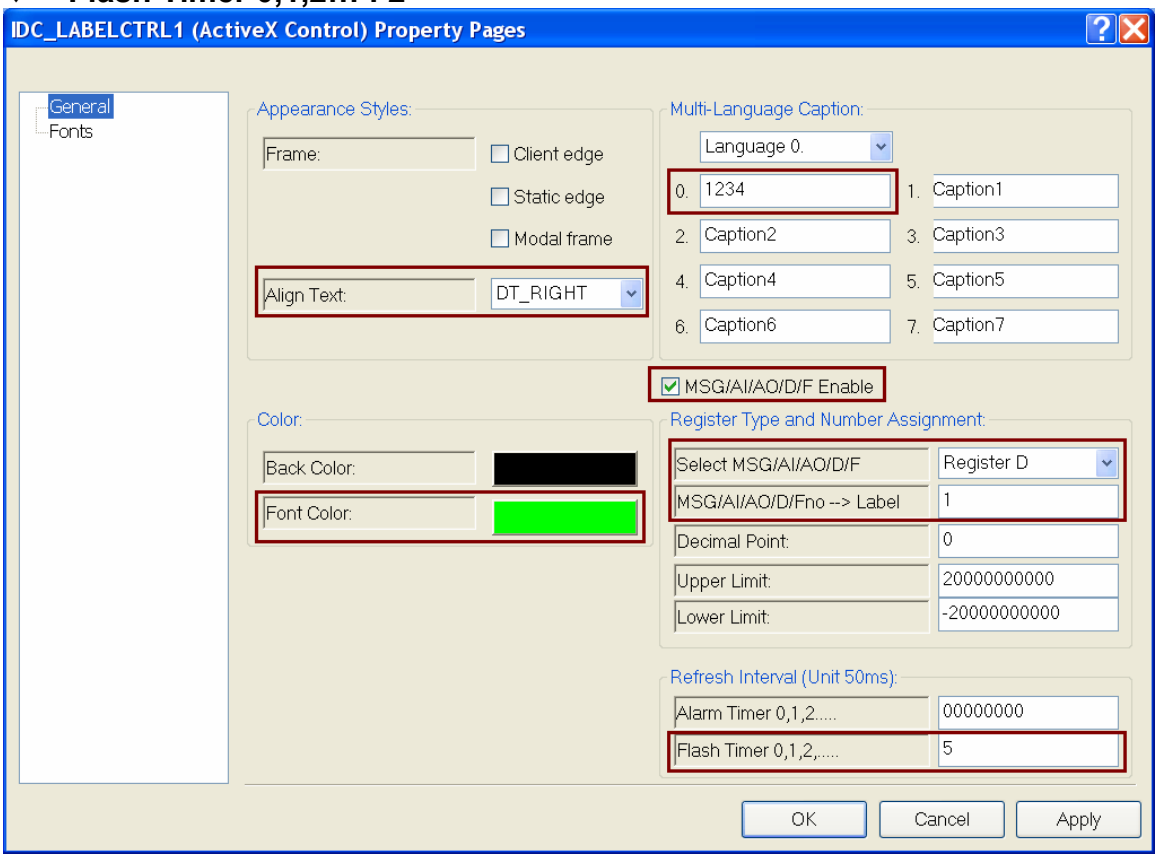


Step2.再利用滑鼠拖曳一個 Lable 物件到 Dialog



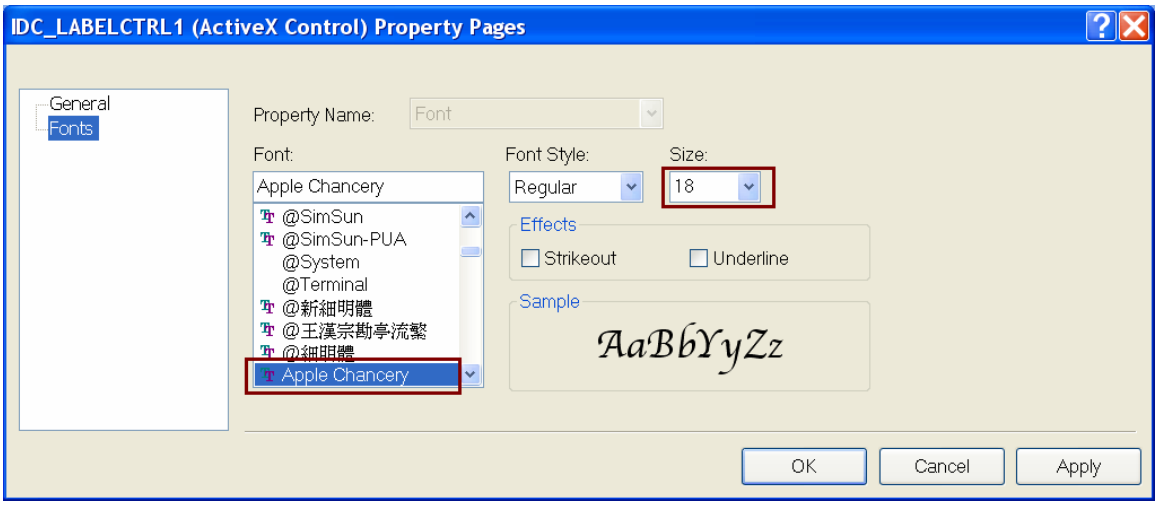
設定屬性:

- **Align Text : DT_RIGHT**
- **Font Color :**
- **Caption : 1234**
- **MSG/AI/AO/D/F Enable :選取**
- **Select MSG/AI/AO/D/F : Register D**
- **MSG/AI/AO/D/F→Label : 1**
- **Flash Timer 0,1,2… : 2**

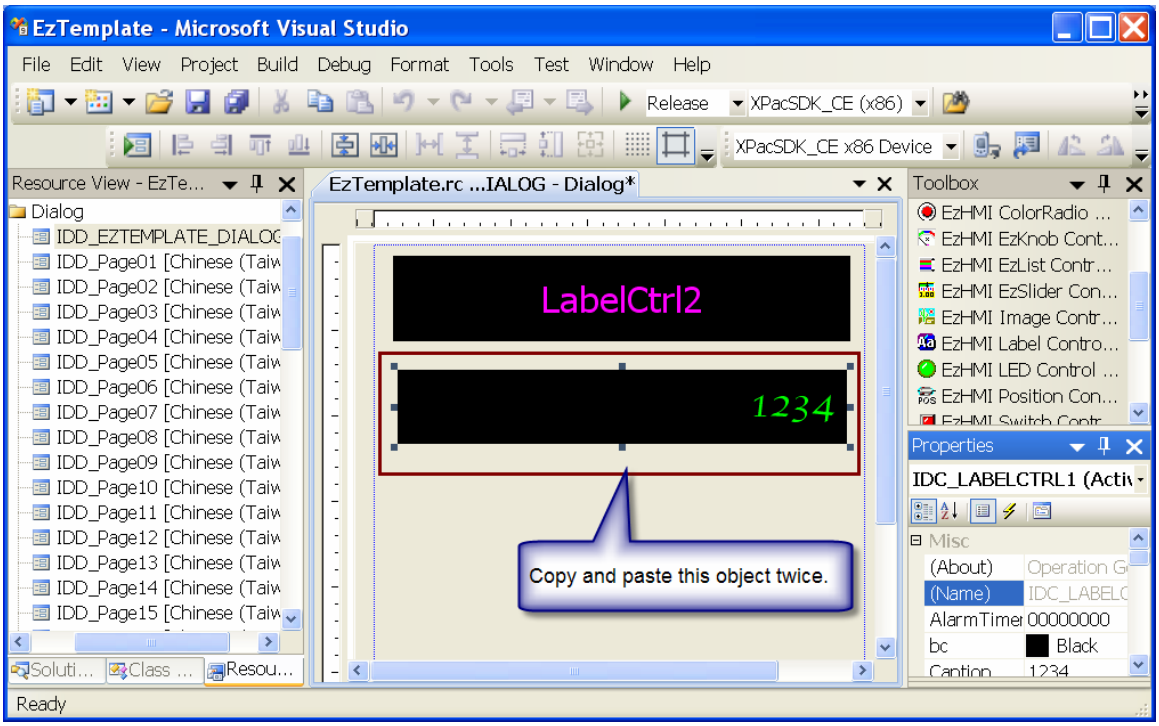


點擊左邊“Fonts”切換到字型設定

- **Font : Apple Chancery**
- **Size : 18**

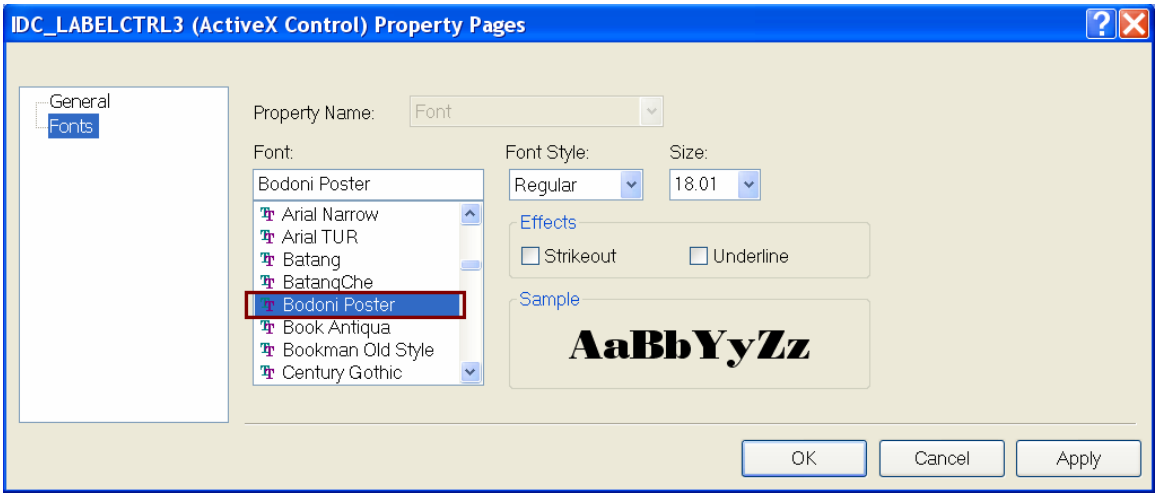


完成後如下圖：

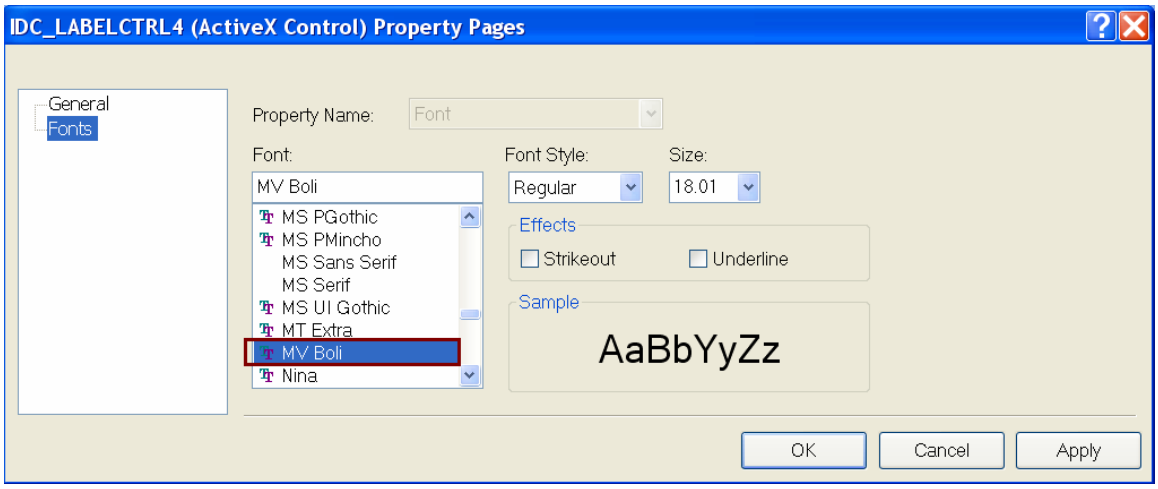


再複製這個物件兩個並將字形分別設定如下:

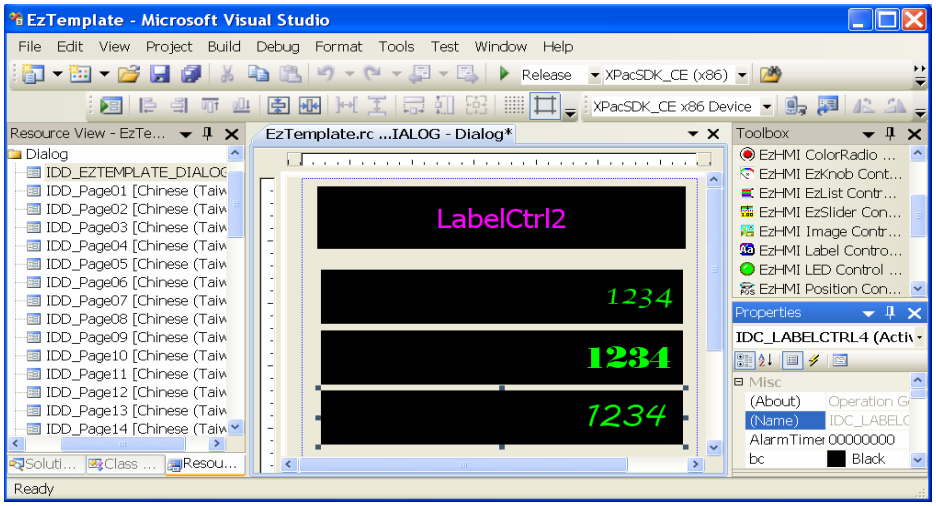
Font : Bodoni Poster



Font : MV Boli



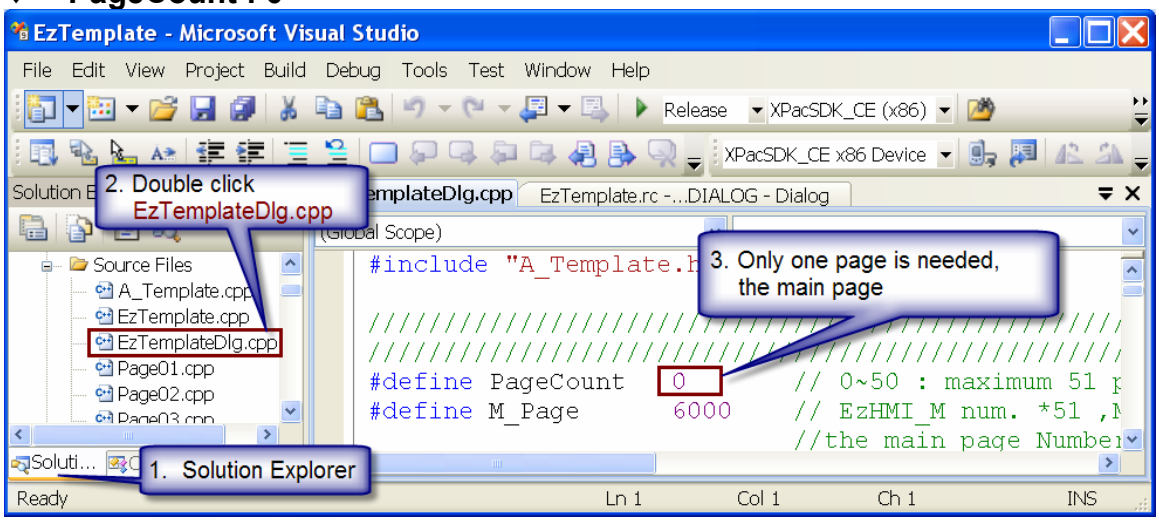
完成後如下圖:



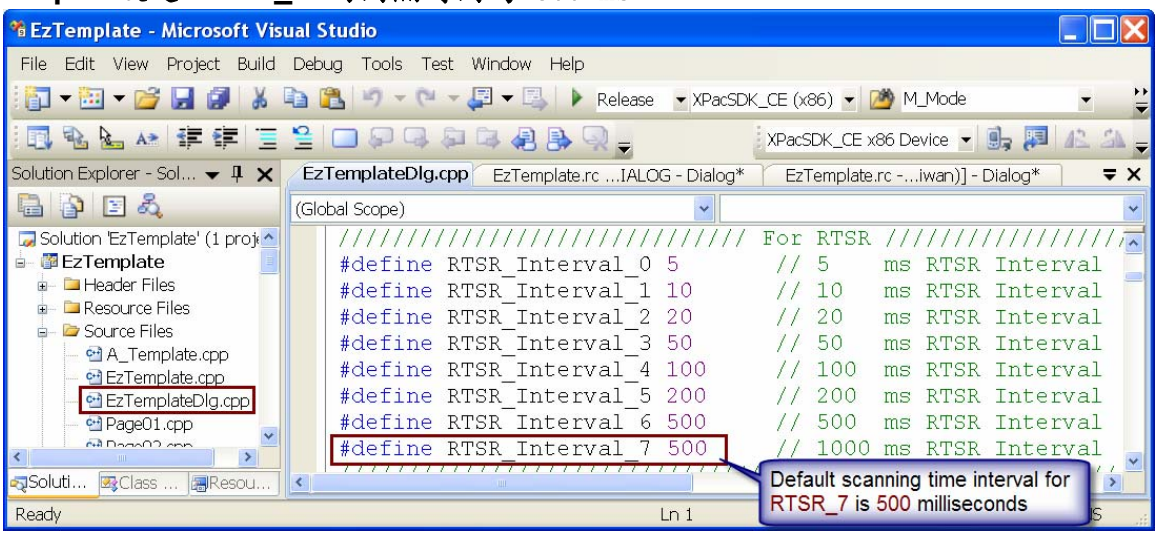
2.13.2 設計與編譯

Step1.在專案管理選 Solution 檢視，並雙擊 EzTemplateDlg.cpp 做如下設定。

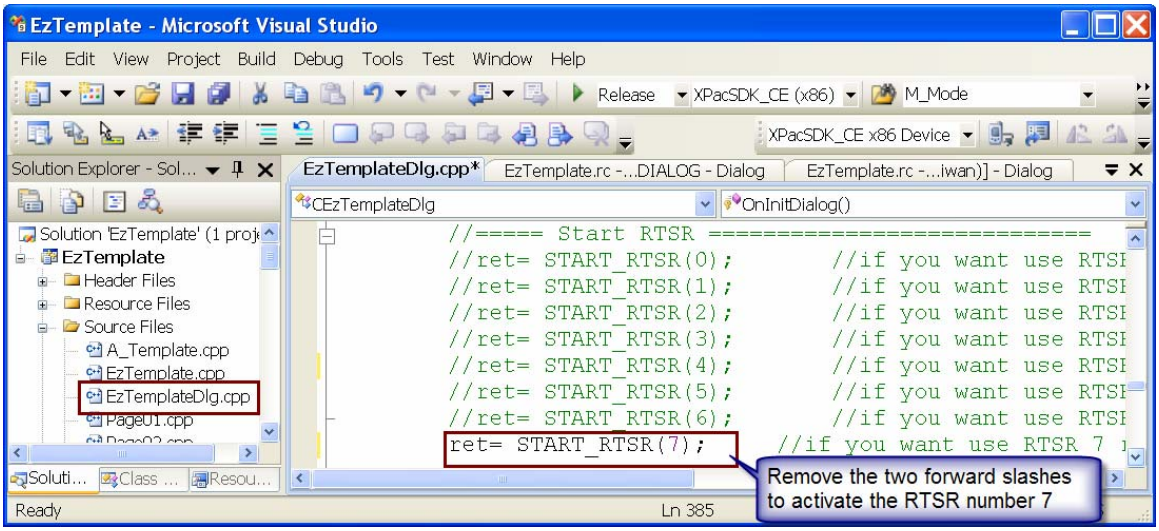
- **PageCount : 0**



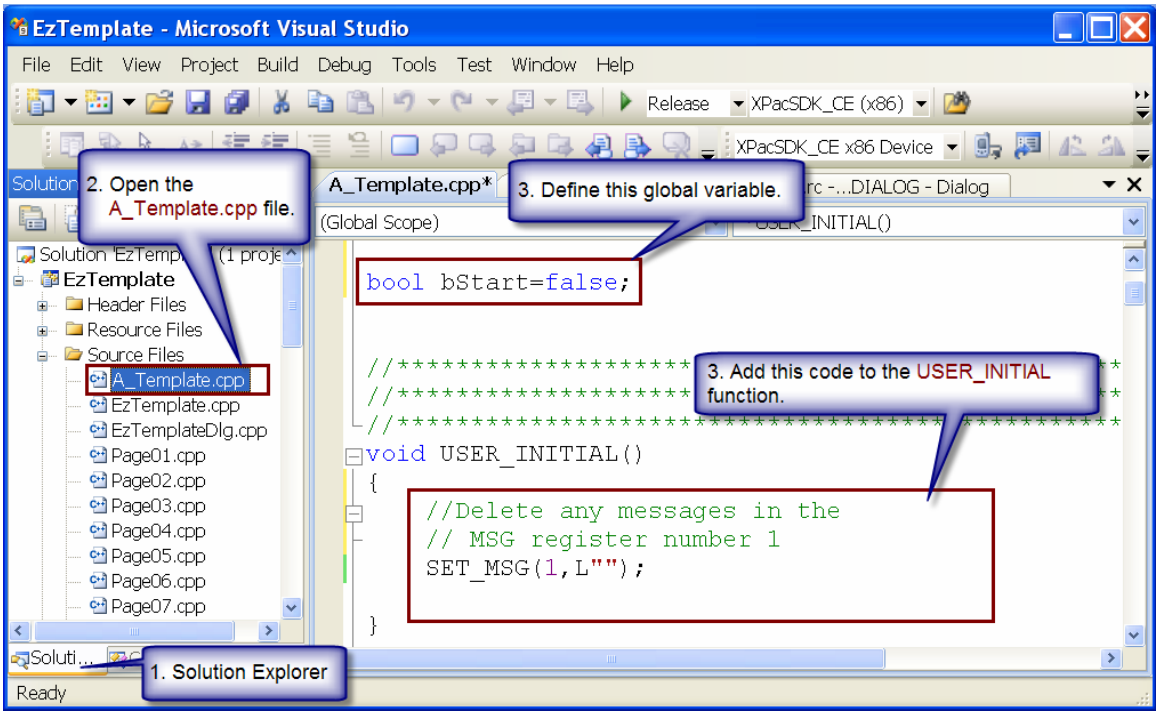
Step2. 設定 RTSR_7 的間隔時間為 500 ms



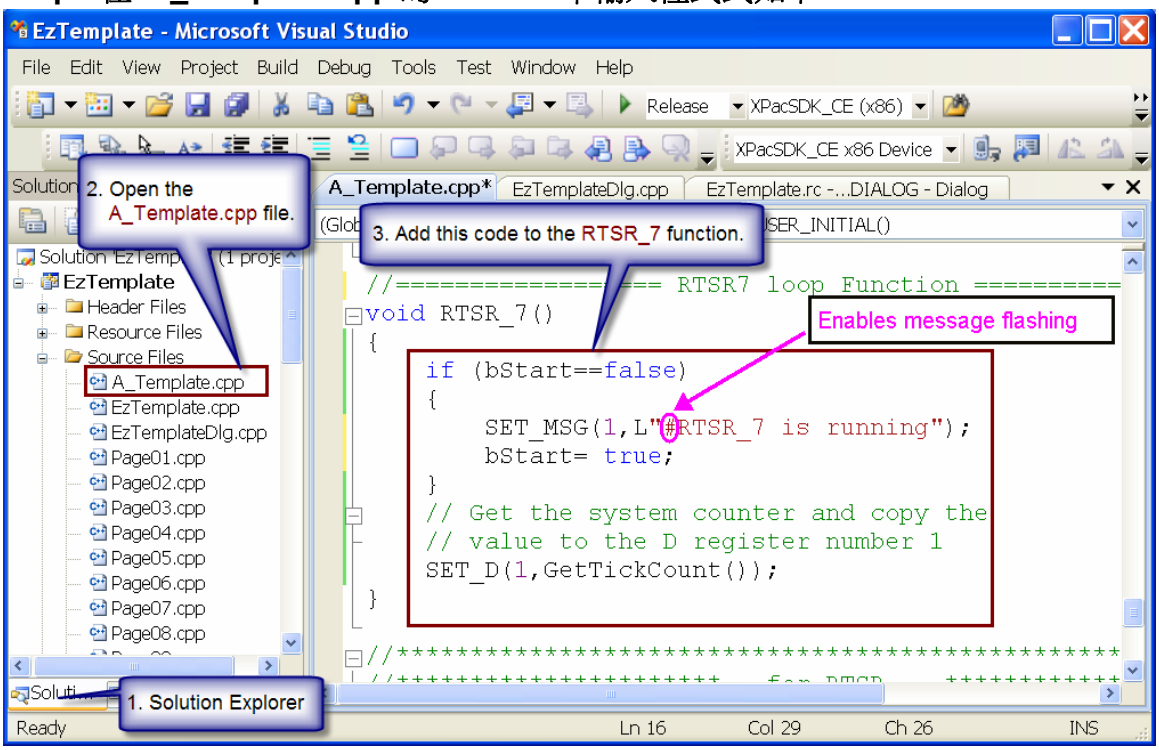
Step3. 將 START_RTSR(7) 前的 // 刪掉以啟用它



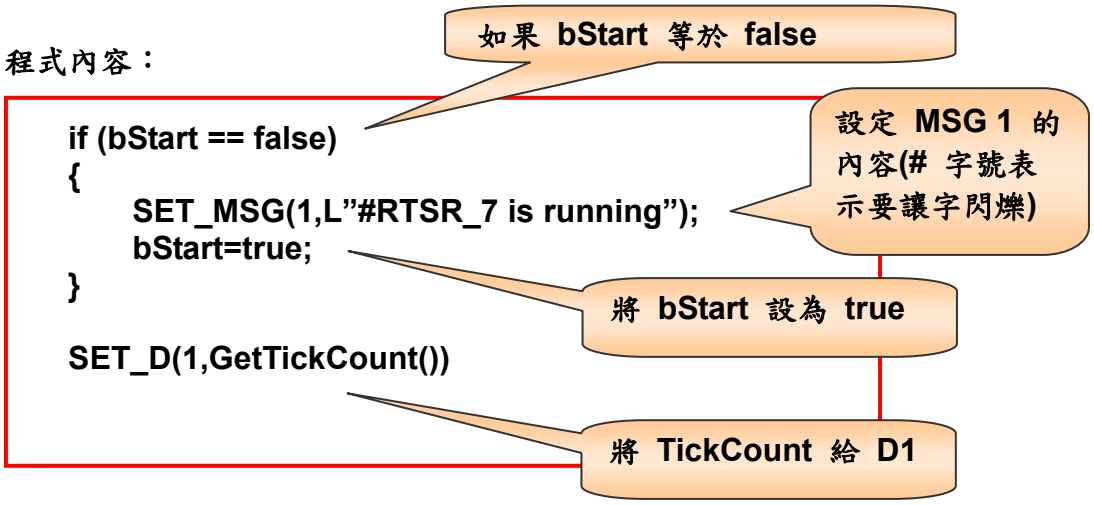
Step4.在專案管理選 Solution 檢視，並雙擊 A_Template.cpp 加入 bStart , USER_INITIAL() 設定初始值



Step5.在 A_Template.cpp 的 RTSR7 中輸入程式式如下

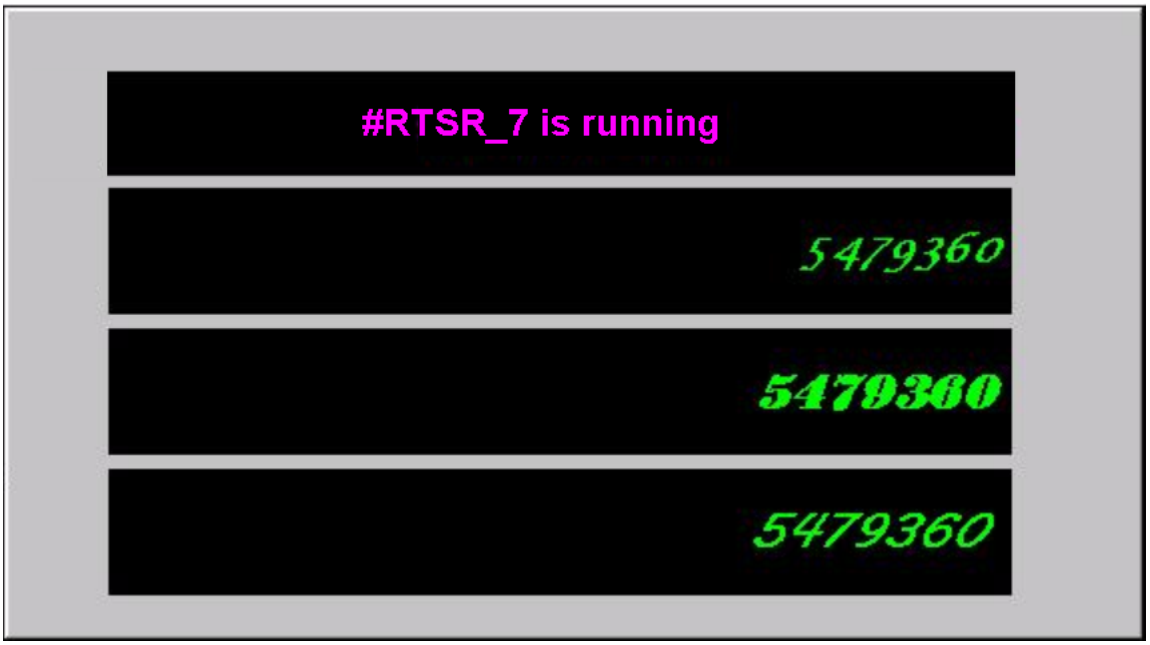


程式內容：



如此即完成專案的建構與執行，如果連線有問題，請參閱 **EzProg-I Getting Started** 手冊中附錄的連線除錯方法說明。

完成後執行如下圖：



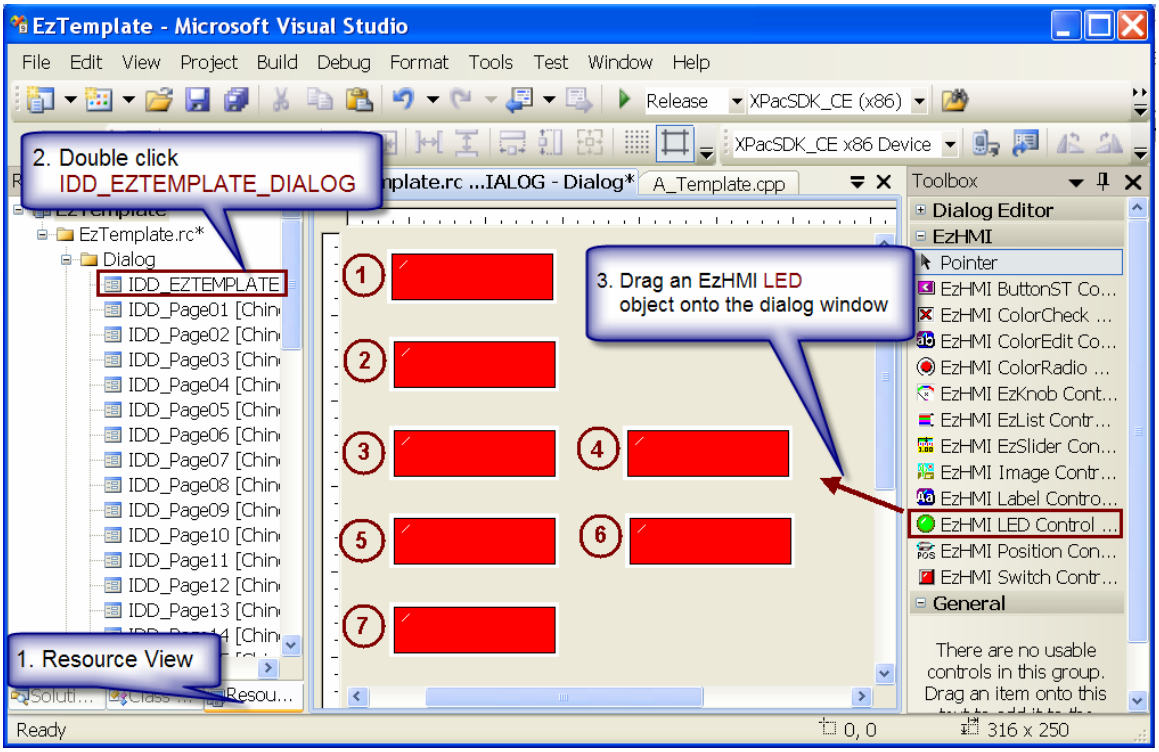
2.14 範例 STEP

步進階梯的程序設計（STEP LADDER），它是 PLC 由邏輯描述語進階到圖式化階梯圖，再進一步進化到結構化軟體設計的重要指標。它使原先的 LADDER 可讀性更高、維護更新變容易，減少軟體設計人員發生錯誤的機率。所以這種結合流程控制與階梯圖語言的設計方式，也是我們 EzCore 控制框架相似的結構，所以稱之為步進（Step）程序，亦可以做為類似狀態機(Status Machine) 的應用功能。

本範例以 EzProg-I 來實現 Step 程式設計概念，請依照 2.2.1 複製樣板專案，請名稱改為 Step ,執行檔為 Step.exe，並開啟之。

2.14.1 HMI 物件畫面控制設計

Step1.拖曳七個 LED 物件到 Dialog 中如下



設定屬性:

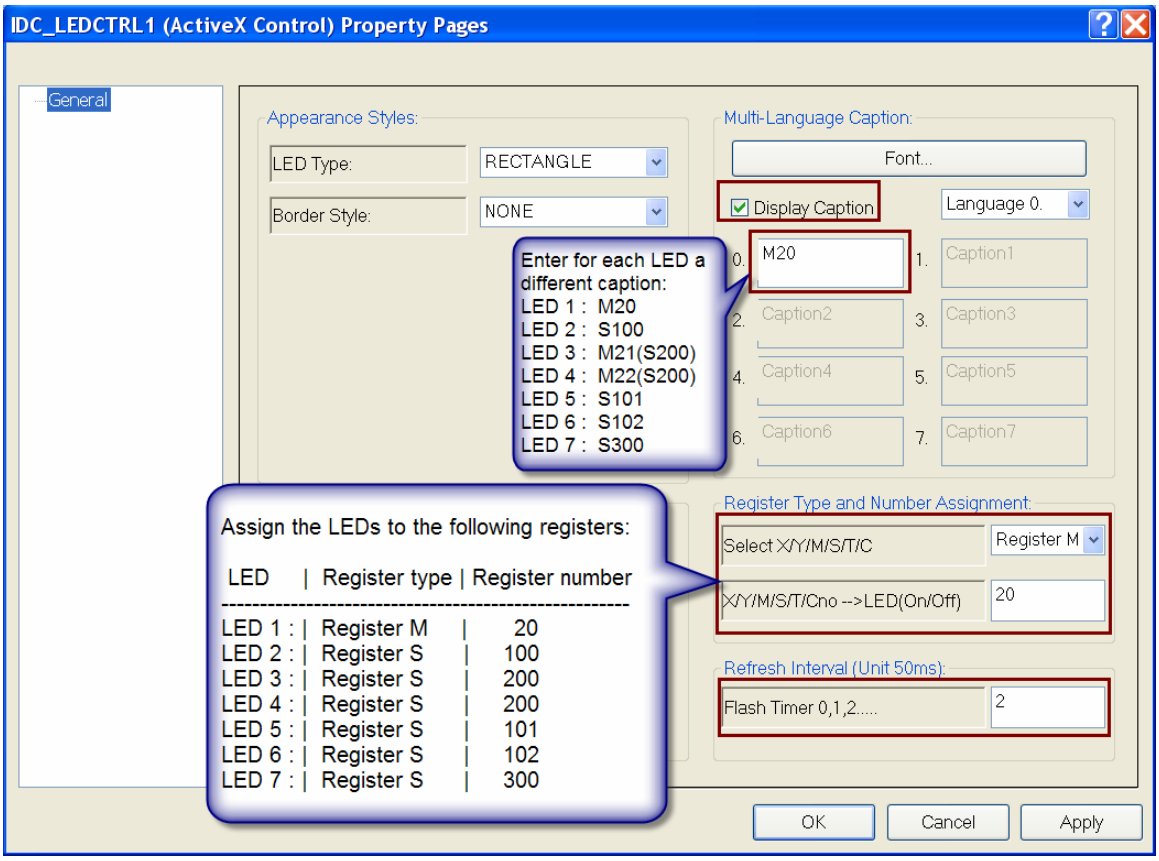
請按照上頁 **Step1.**圖中每一個 **LED** 的編號逐一設定

- **Display Caption : LED1~7** 每一個都要選取起來
- **Caption :**
  LED 1 : M20
  LED 2 : S100
  LED 3 : M21(S200)
  LED 4 : M22(S200)
  LED 5 : S101
  LED 6 : S102
  LED 7 : S300

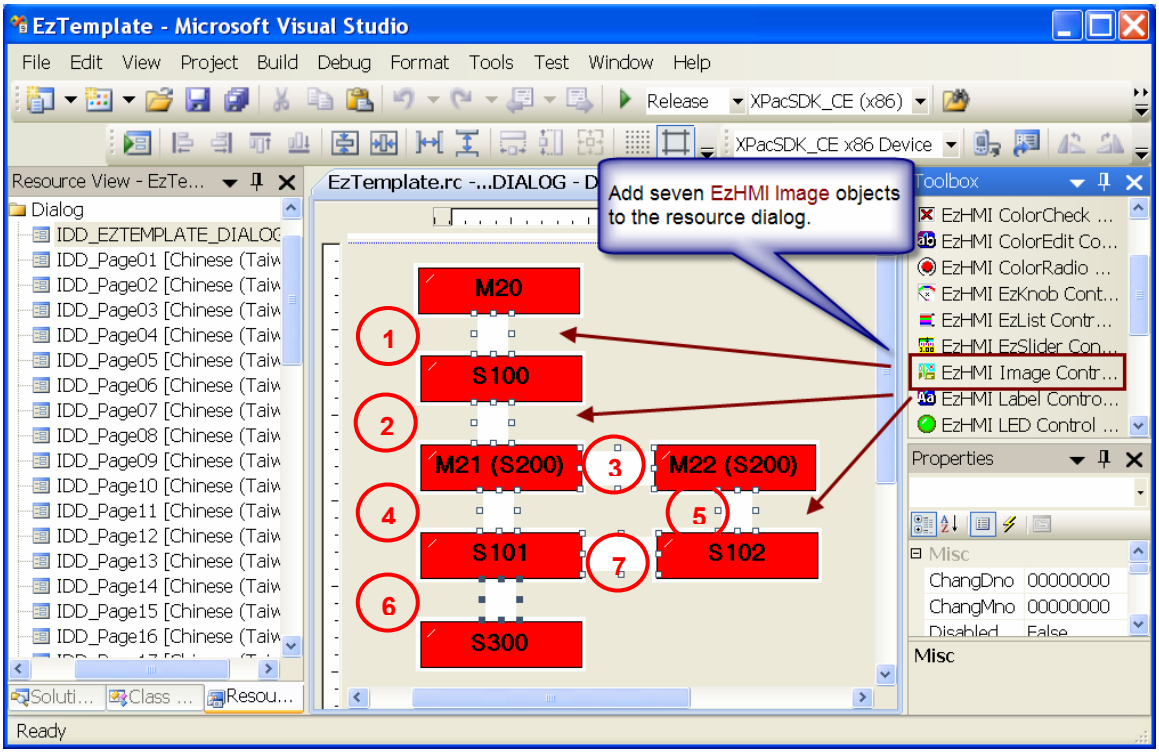
- **Register Type and Number Assignment :**

LED	Register type	Register number
LED 1 :	Register M	20
LED 2 :	Register S	100
LED 3 :	Register S	200
LED 4 :	Register S	200
LED 5 :	Register S	101
LED 6 :	Register S	102
LED 7 :	Register S	300

- **Flash Timer 0,1,2… : LED1~7** 每一個都設定 **2**



Step2. 再利拖曳七個 Image 物件到 Dialog 中如下



設定屬性:

請按照上頁 **Image** 的編號逐一設定

Image 圖片請選擇 **C:\ICPDAS\EzProg-I\Tutorial\BMP** 資料夾下的圖片

- **Directory:**

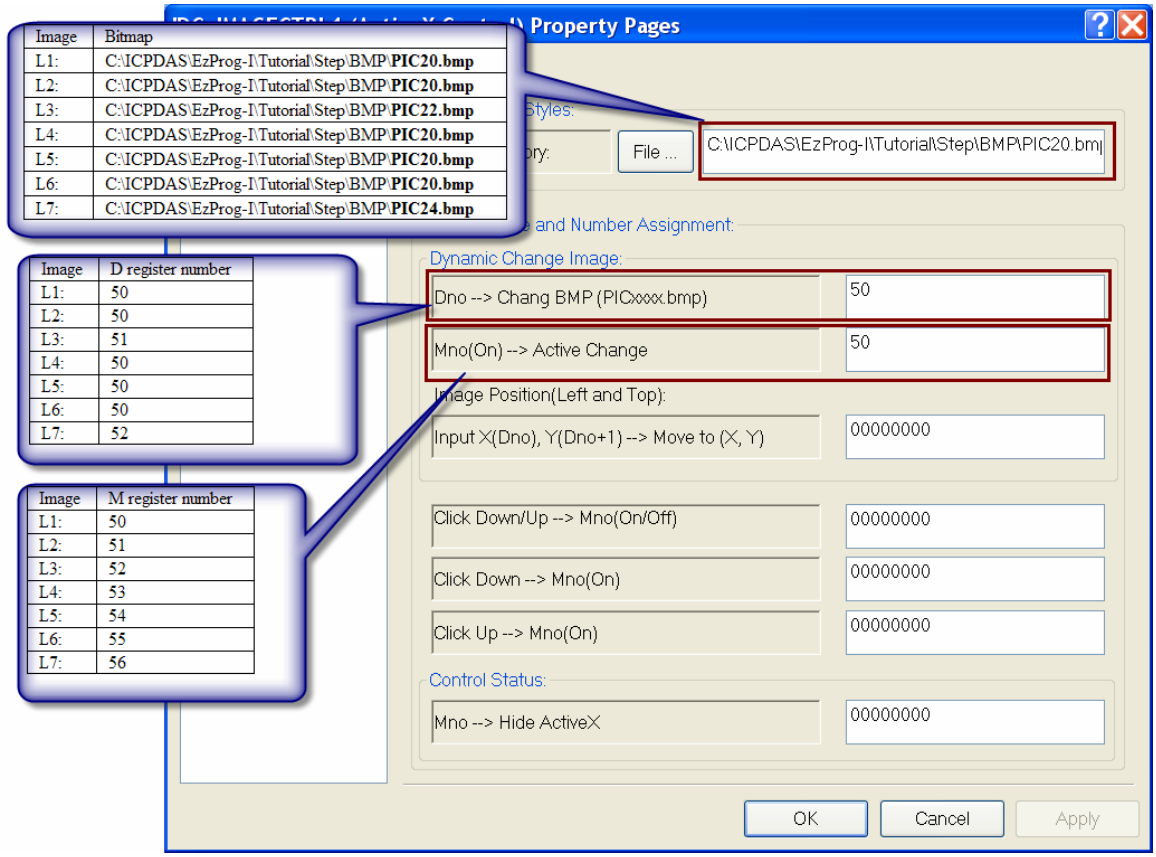
Image	Bitmap
L1:	C:\ICPDAS\EzProg-I\Tutorial\Step\BMP**PIC20.bmp**
L2:	C:\ICPDAS\EzProg-I\Tutorial\Step\BMP**PIC20.bmp**
L3:	C:\ICPDAS\EzProg-I\Tutorial\Step\BMP**PIC22.bmp**
L4:	C:\ICPDAS\EzProg-I\Tutorial\Step\BMP**PIC20.bmp**
L5:	C:\ICPDAS\EzProg-I\Tutorial\Step\BMP**PIC20.bmp**
L6:	C:\ICPDAS\EzProg-I\Tutorial\Step\BMP**PIC20.bmp**
L7:	C:\ICPDAS\EzProg-I\Tutorial\Step\BMP**PIC24.bmp**

- **Dno→Change BMP(PICxxxx.bmp) :**

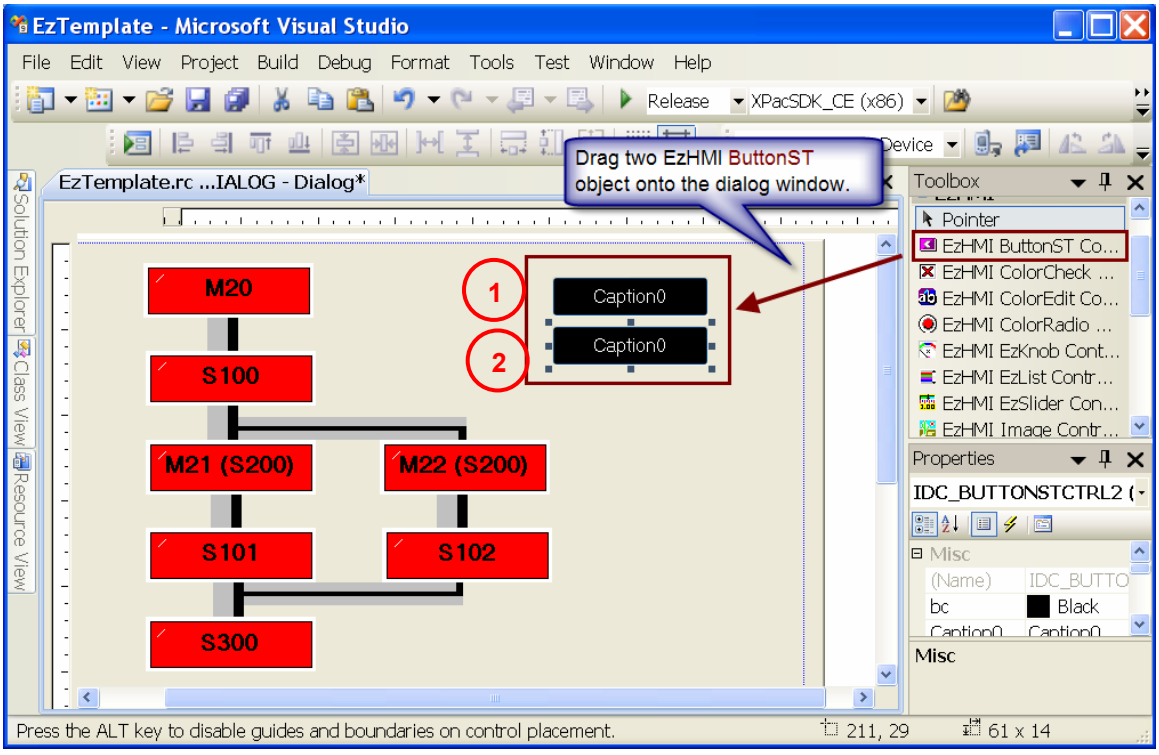
Image	D register number
L1:	50
L2:	50
L3:	51
L4:	50
L5:	50
L6:	50
L7:	52

- **Mno(On)→Active Change :**

Image	M register number
L1:	50
L2:	51
L3:	52
L4:	53
L5:	54
L6:	55
L7:	56



Step3.再拖曳兩個 ButtonST 物件到 Dialog 中如下



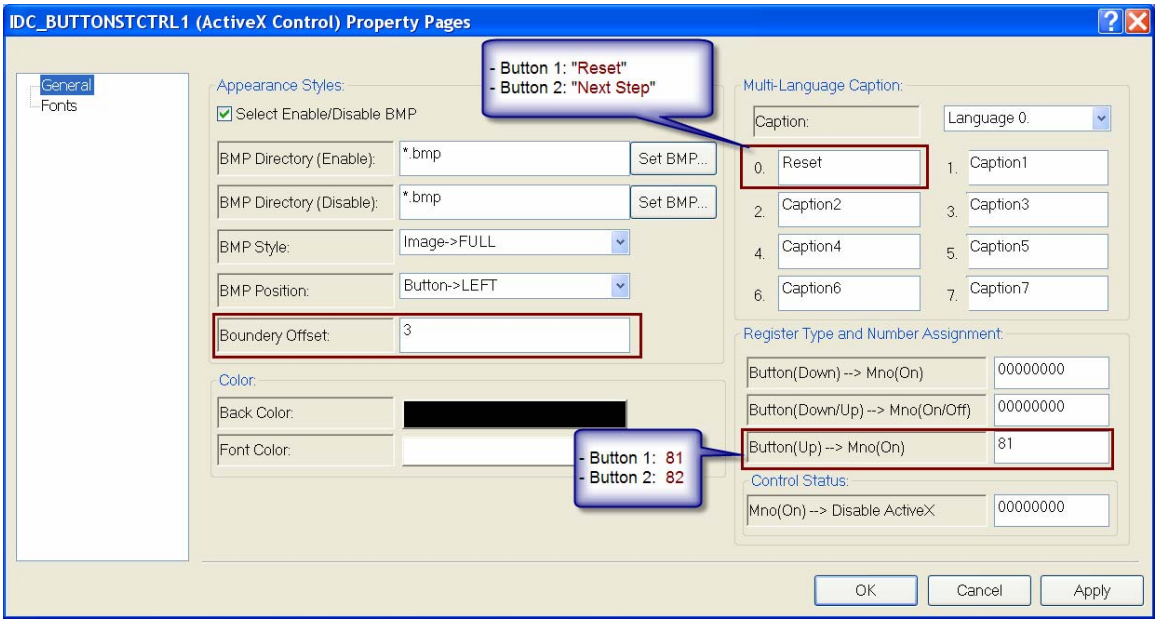
設定屬性:

請按照上頁 **ButtonST** 的編號逐一設定

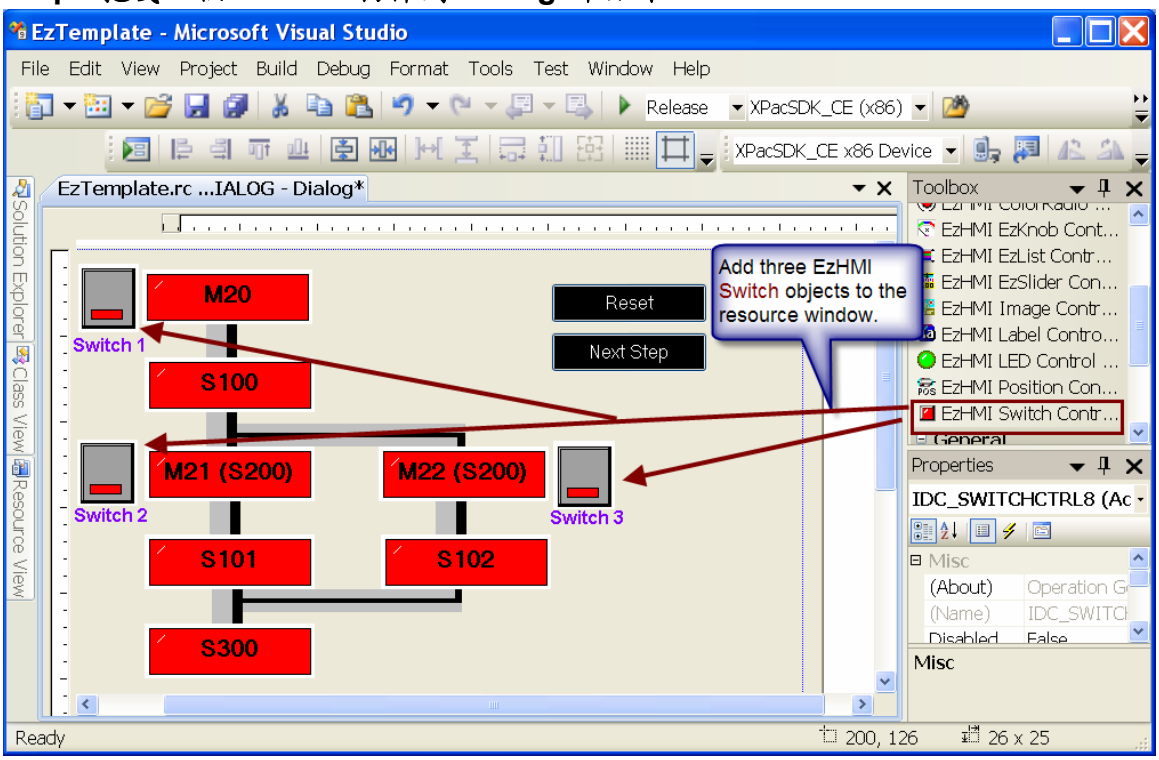
- **Boundery Offset : Button 1,2 都是設定 3**

- **Caption :**
  Button 1: Reset
  Button 2: Next Step

- **Button(Up)→Mno(On) :**
  Button 1: 81
  Button 2: 82



Step4.拖曳三個 Switch 物件到 Dialog 中如下



設定屬性:

請按照上頁 Switch 的編號逐一設定

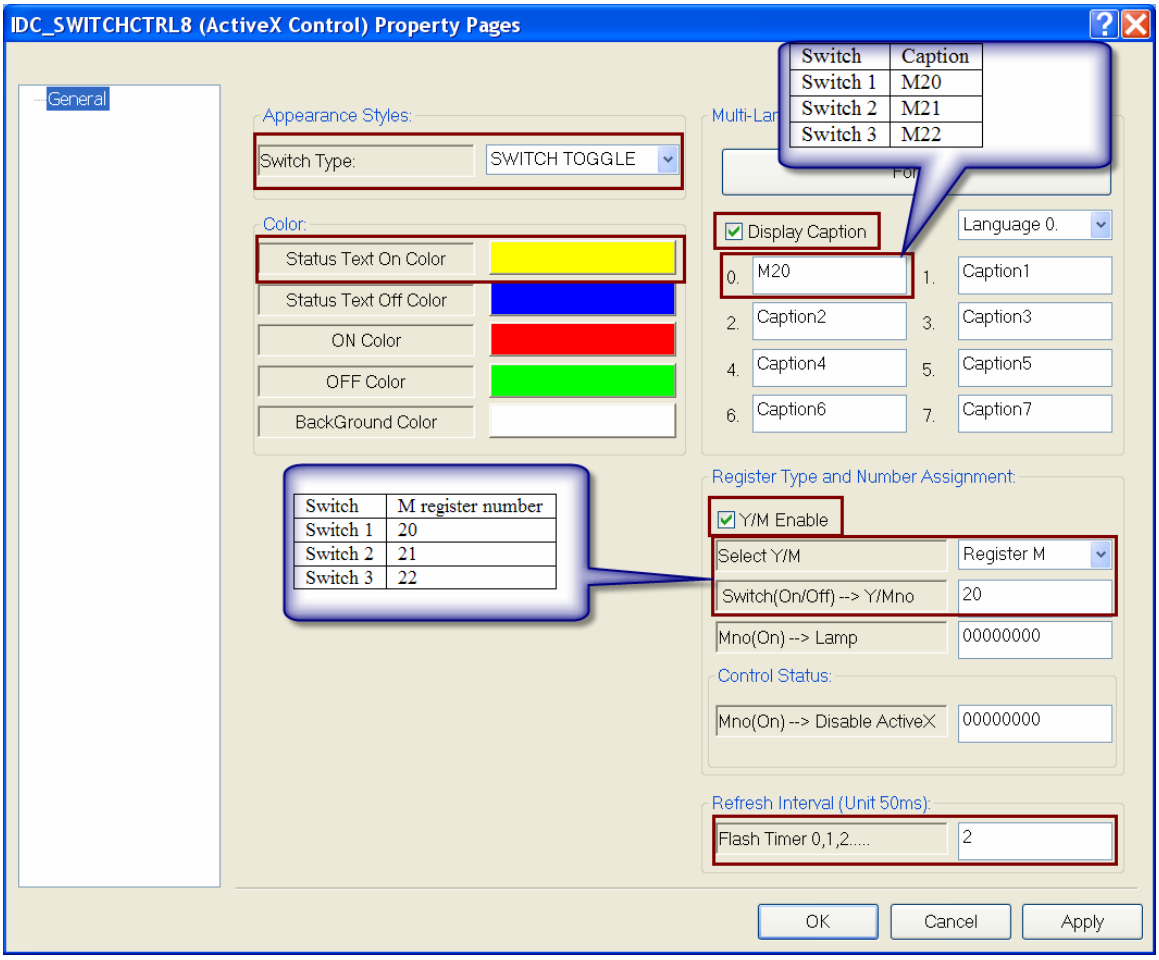
- **Swtich Type : Switch1~3 都設定”SWITCH TOGGLE”**
- **Status Text On Color : Switch1~3 都設定**
- **Display Caption : Switch1~3 都選取**
- **Caption0 :**

Switch	Caption
Switch 1	M20
Switch 2	M21
Switch 3	M22

- **Y/M Enable : Switch1~3 都選取**
- **Select Y/M : Switch1~3 都設定 Register M**
- **Switch (On/Off) :**

Switch	M register number
Switch 1	20
Switch 2	21
Switch 3	22

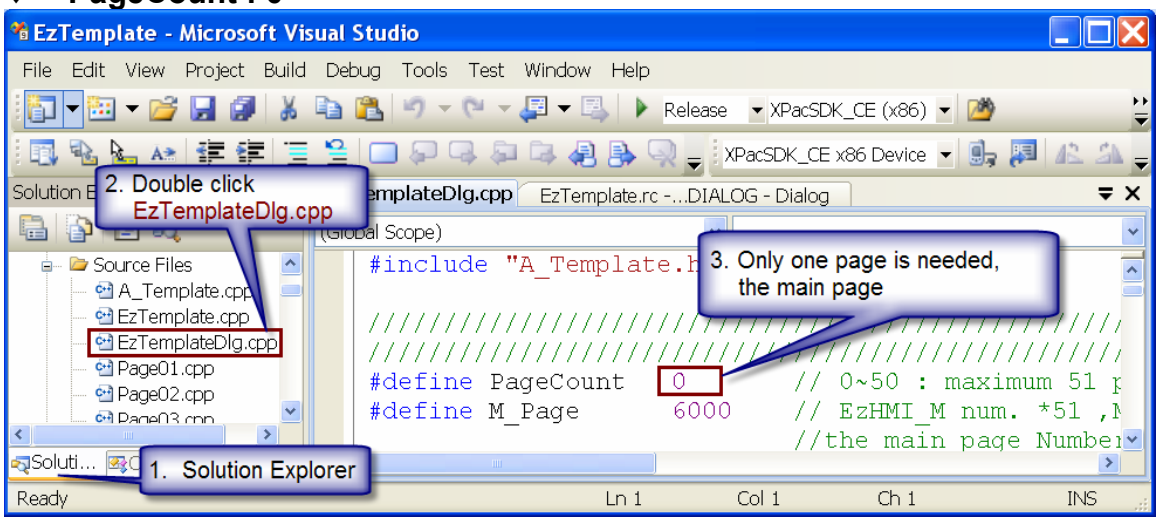
- **Flash Timer 0,1,2… : Switch1~3 都設定 2**


IDC_SWITCHCTRL8 (ActiveX Control) Property Pages
General
Appearance Styles:
Switch Type:
SWITCH TOGGLE
Color:
Status Text On Color
Status Text Off Color
ON Color
OFF Color
BackGround Color
Switch | Caption
Switch 1 | M20
Switch 2 | M21
Switch 3 | M22
Display Caption
Language 0.
0. M20
1. Caption1
2. Caption2
3. Caption3
4. Caption4
5. Caption5
6. Caption6
7. Caption7
Switch | M register number
Switch 1 | 20
Switch 2 | 21
Switch 3 | 22
Register Type and Number Assignment:
Y/M Enable
Select Y/M
Register M
Switch(On/Off) --> Y/Mno
20
Mno(On) --> Lamp
00000000
Control Status:
Mno(On) --> Disable ActiveX
00000000
Refresh Interval (Unit 50ms):
Flash Timer 0,1,2.....
2
OK
Cancel
Apply

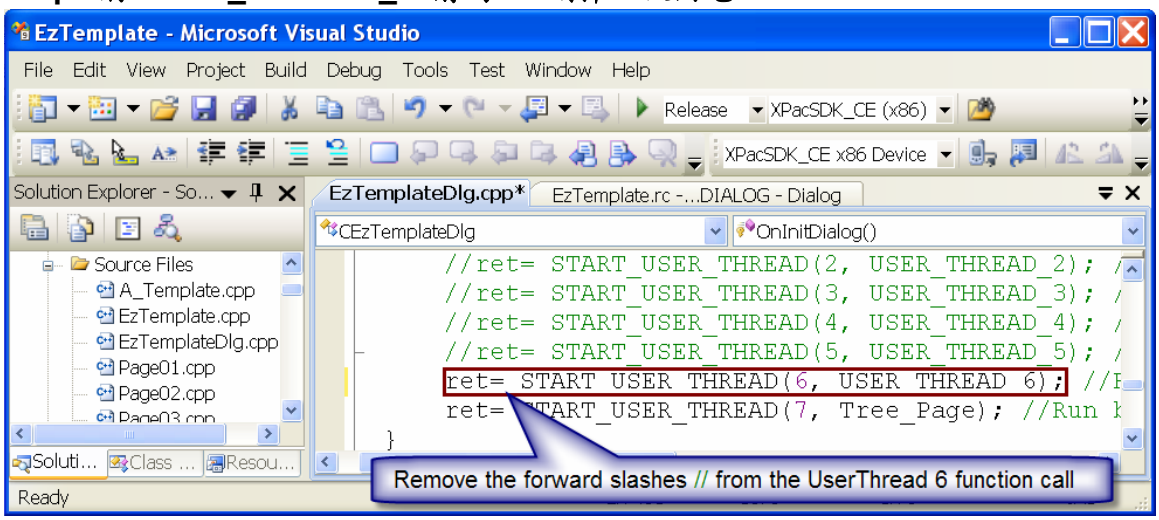
2.14.2 設計與編譯

Step1.在專案管理選 Solution 檢視，並雙擊 EzTemplateDlg.cpp 做如下設定。

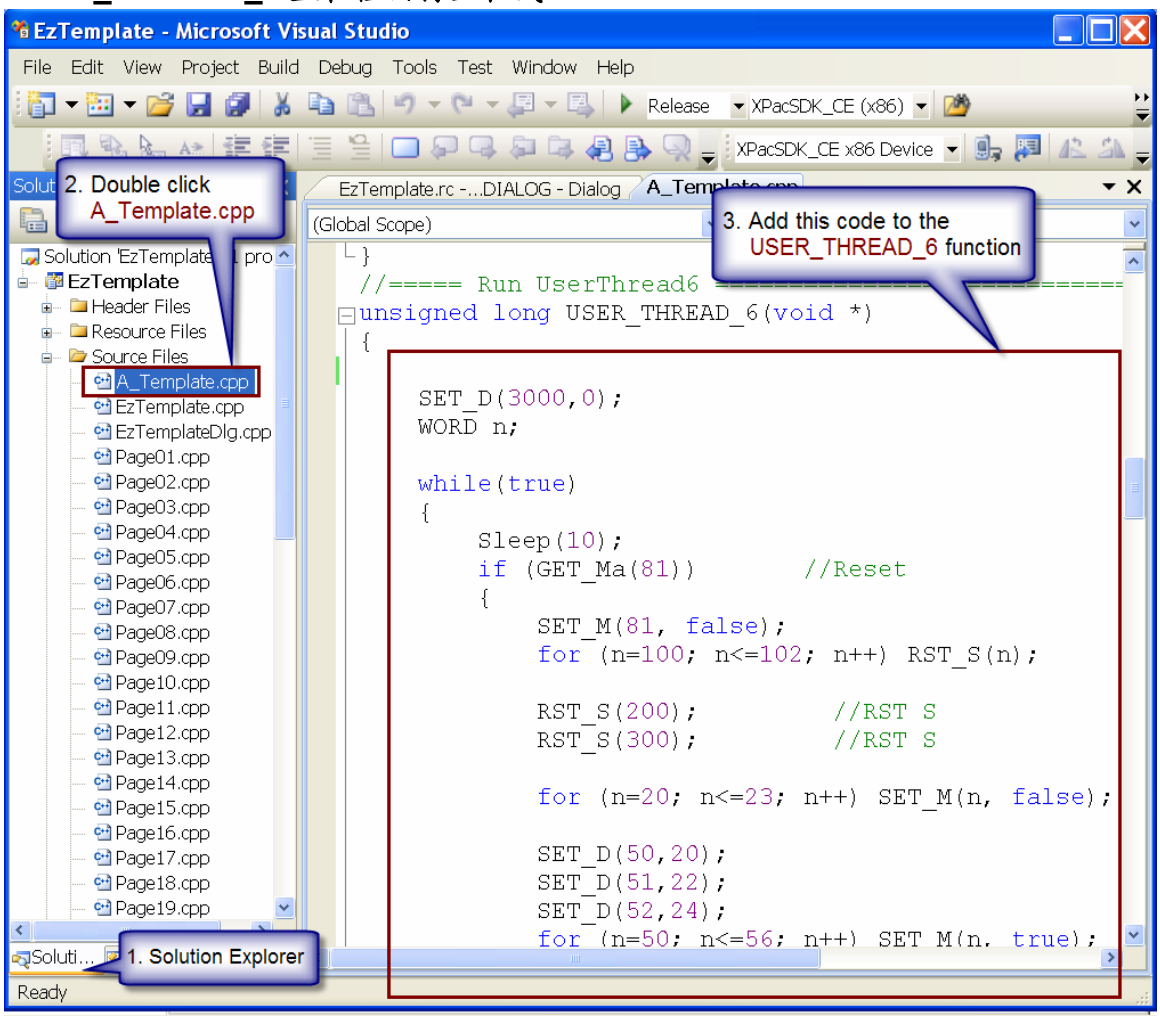
- **PageCount : 0**



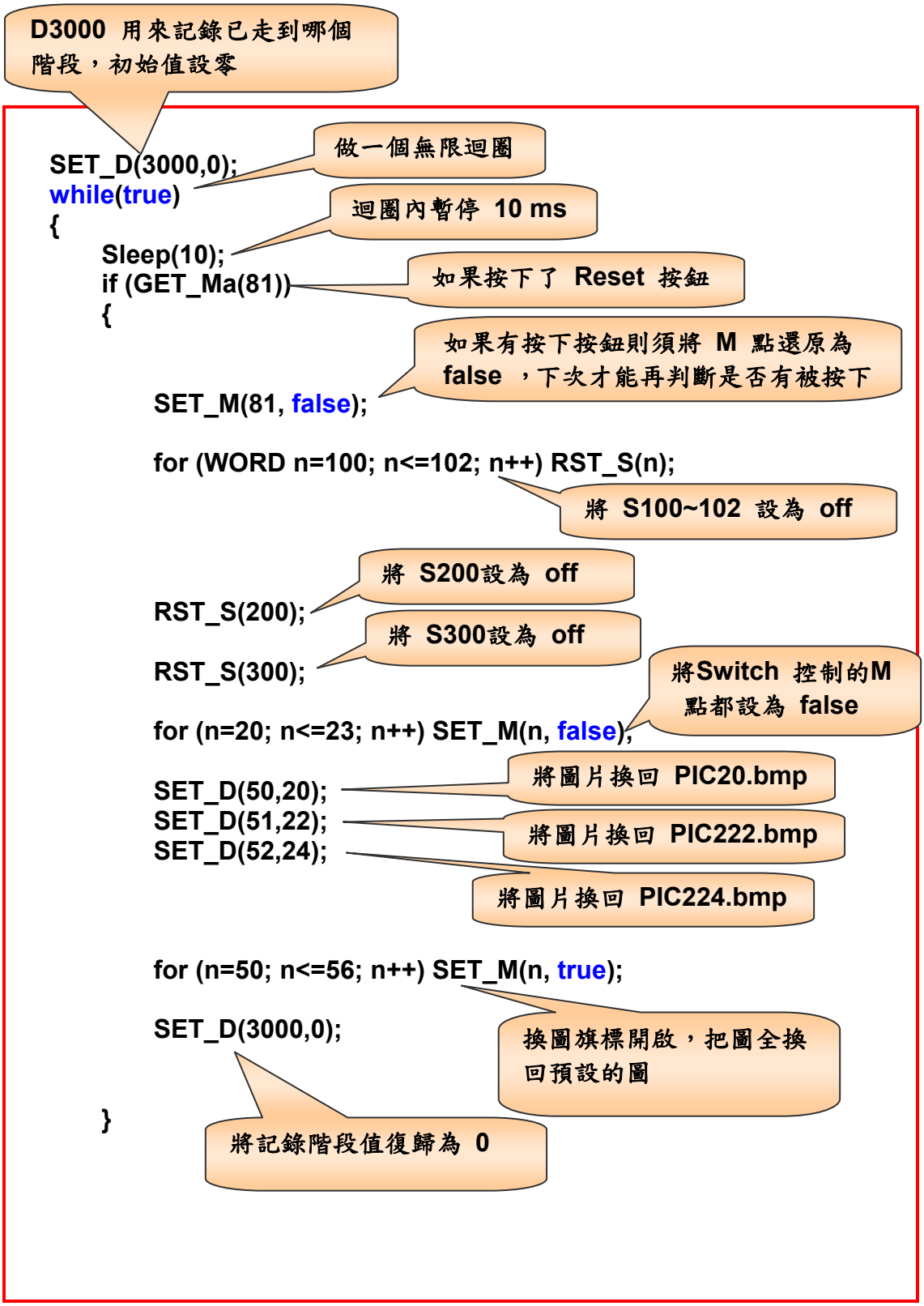
Step2.將 USER_THREAD_6 前的 // 刪掉以啟用它

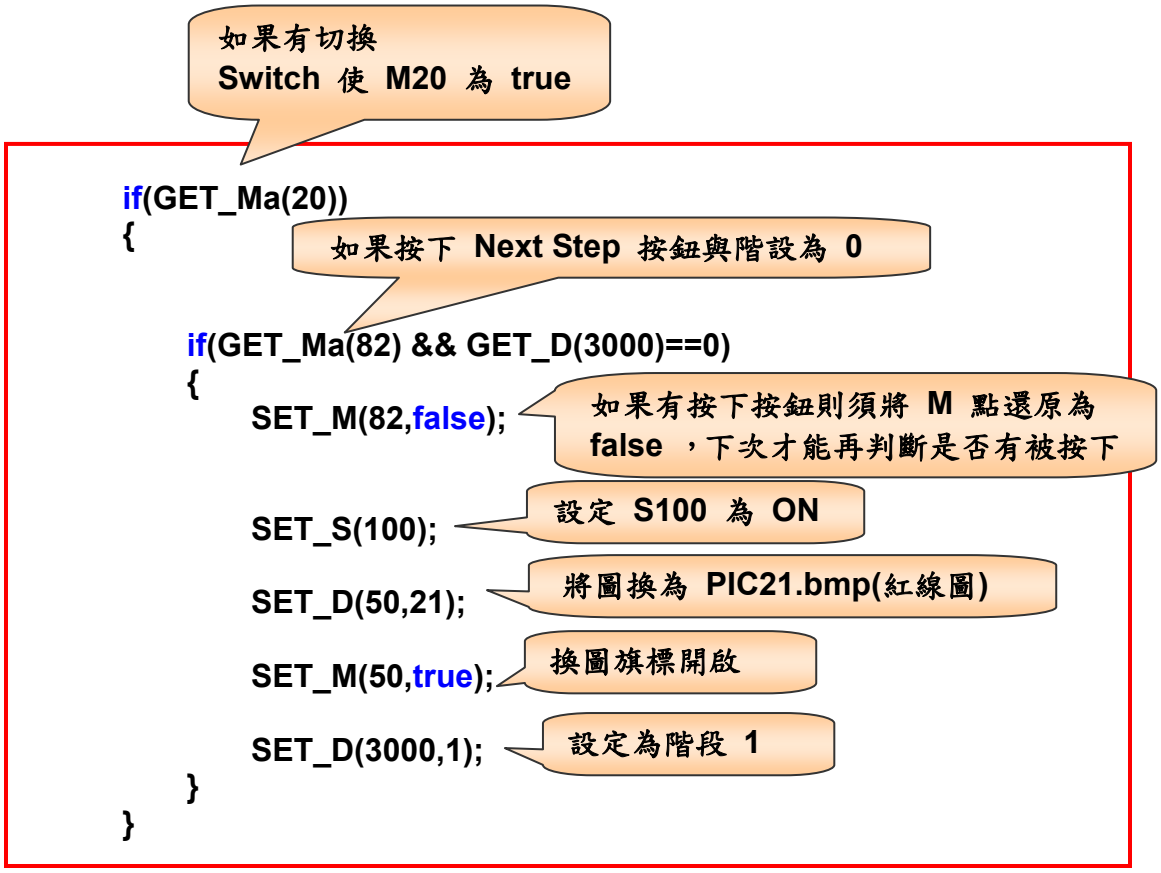


Step3.在專案管理選 Solution 檢視，並雙擊 A_Template.cpp 找到 USER_THREAD_6 並在裡面打上程式

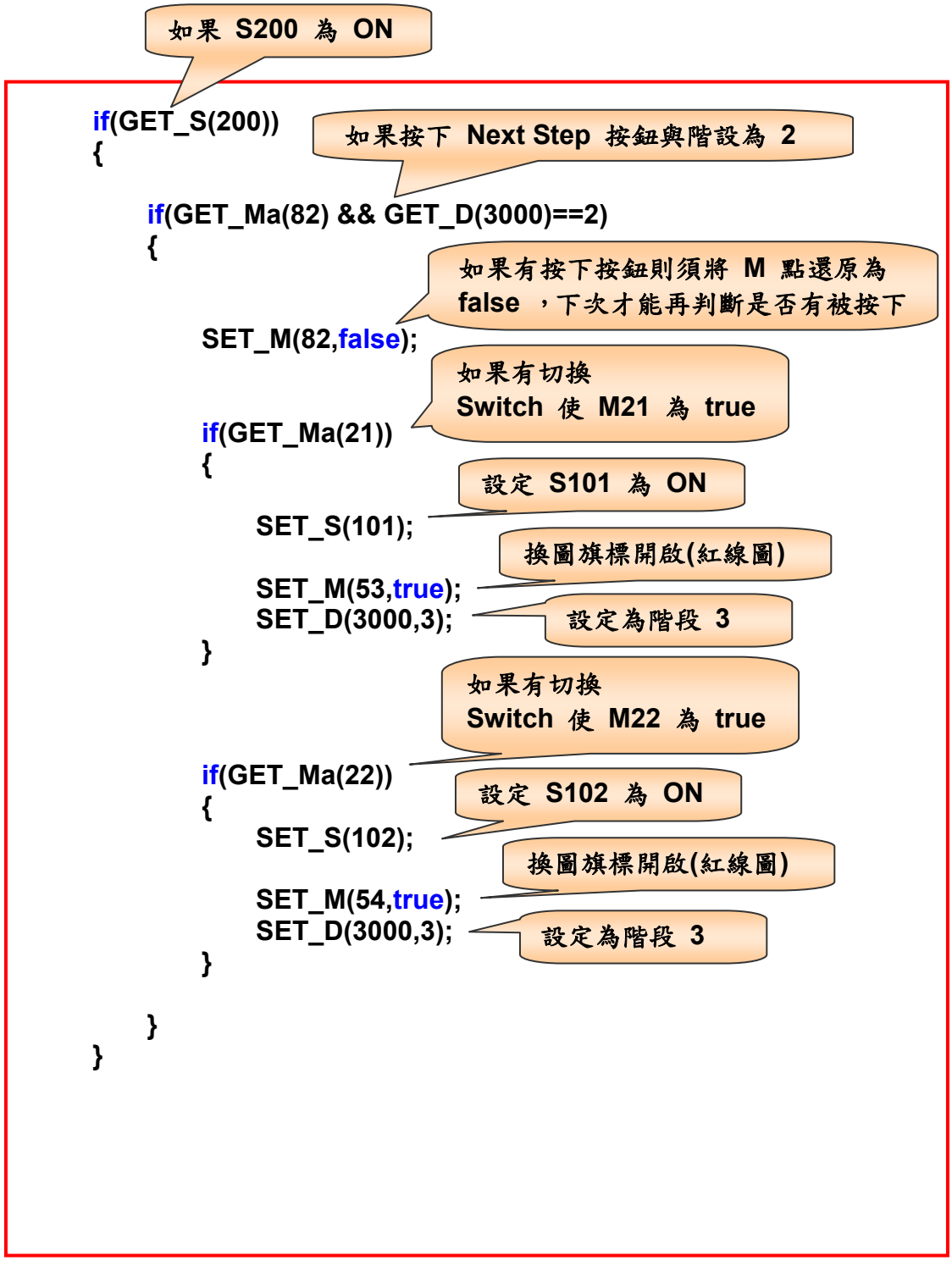


程式內容：


D3000 用來記錄已走到哪個階段，初始值設零
做一個無限迴圈
迴圈內暫停 10 ms
如果按下了 Reset 按鈕
如果有按下按鈕則須將 M 點還原為 false ，下次才能再判斷是否有被按下
將 S100~102 設為 off
將 S200設為 off
將 S300設為 off
將Switch 控制的M點都設為 false
將圖片換回 PIC20.bmp
將圖片換回 PIC222.bmp
將圖片換回 PIC224.bmp
換圖旗標開啟，把圖全換回預設的圖
將記錄階段值復歸為 0
SET_D(3000,0);
while(true)
{
Sleep(10);
if (GET_Ma(81))
{
SET_M(81, false);
for (WORD n=100; n<=102; n++) RST_S(n);
RST_S(200);
RST_S(300);
for (n=20; n<=23; n++) SET_M(n, false);
SET_D(50,20);
SET_D(51,22);
SET_D(52,24);
for (n=50; n<=56; n++) SET_M(n, true);
SET_D(3000,0);
}


如果有切換
Switch 使 M20 為 true
if(GET_Ma(20))
{
如果按下 Next Step 按鈕與階設為 0
if(GET_Ma(82) && GET_D(3000)==0)
{
SET_M(82,false);
如果有按下按鈕則須將 M 點還原為
false ，下次才能再判斷是否有被按下
SET_S(100);
設定 S100 為 ON
SET_D(50,21);
將圖換為 PIC21.bmp(紅線圖)
SET_M(50,true);
換圖旗標開啟
SET_D(3000,1);
設定為階段 1
}
}


如果 S100 為 ON
if(GET_S(100))
{
如果按下 Next Step 按鈕與階設為 1
if(GET_Ma(82) && GET_D(3000)==1)
{
SET_M(82,false);
如果有按下按鈕則須將 M 點還原為 false ，下次才能再判斷是否有被按下
SET_S(200);
設定 S200 為 ON
SET_D(51,23);
將圖換為 PIC23.bmp(紅線圖)
SET_M(51,true);
換圖旗標開啟
SET_M(52,true);
換圖旗標開啟
SET_D(3000,2);
設定為階段 2
}
}


如果 S200 為 ON
if(GET_S(200))
{
如果按下 Next Step 按鈕與階設為 2
if(GET_Ma(82) && GET_D(3000)==2)
{
如果有按下按鈕則須將 M 點還原為 false ，下次才能再判斷是否有被按下
SET_M(82,false);
如果有切換 Switch 使 M21 為 true
if(GET_Ma(21))
{
設定 S101 為 ON
SET_S(101);
換圖旗標開啟(紅線圖)
SET_M(53,true);
設定為階段 3
SET_D(3000,3);
}
如果有切換 Switch 使 M22 為 true
if(GET_Ma(22))
{
設定 S102 為 ON
SET_S(102);
換圖旗標開啟(紅線圖)
SET_M(54,true);
設定為階段 3
SET_D(3000,3);
}
}
}

```
        if(GET_S(101))
        {
            if(GET_Ma(82) && GET_D(3000)==3)
            {
                SET_M(82,false);
                SET_S(300);
                SET_M(55,true);
                SET_D(3000,4);
            }
        }

        if(GET_S(102))
        {
            if(GET_Ma(82) && GET_D(3000)==3)
            {
                SET_M(82,false);
                SET_S(300);
                SET_D(50,28);
                SET_M(55,true);
                SET_D(52,25);
                SET_M(56,true);
                SET_D(3000,4);
            }
        }

}
```

如果 S101 為 ON

如果按下 Next Step 按鈕與階設為 3

如果有按下按鈕則須將 M 點還原為 false ，下次才能再判斷是否有被按下

設定 S300 為 ON

換圖旗標開啟(紅線圖)

設定為階段 4

如果 S102 為 ON

如果按下 Next Step 按鈕與階設為 3

如果有按下按鈕則須將 M 點還原為 false ，下次才能再判斷是否有被按下

設定 S300 為 ON

將圖換為 PIC28.bmp(紅線圖)

換圖旗標開啟(紅線圖)

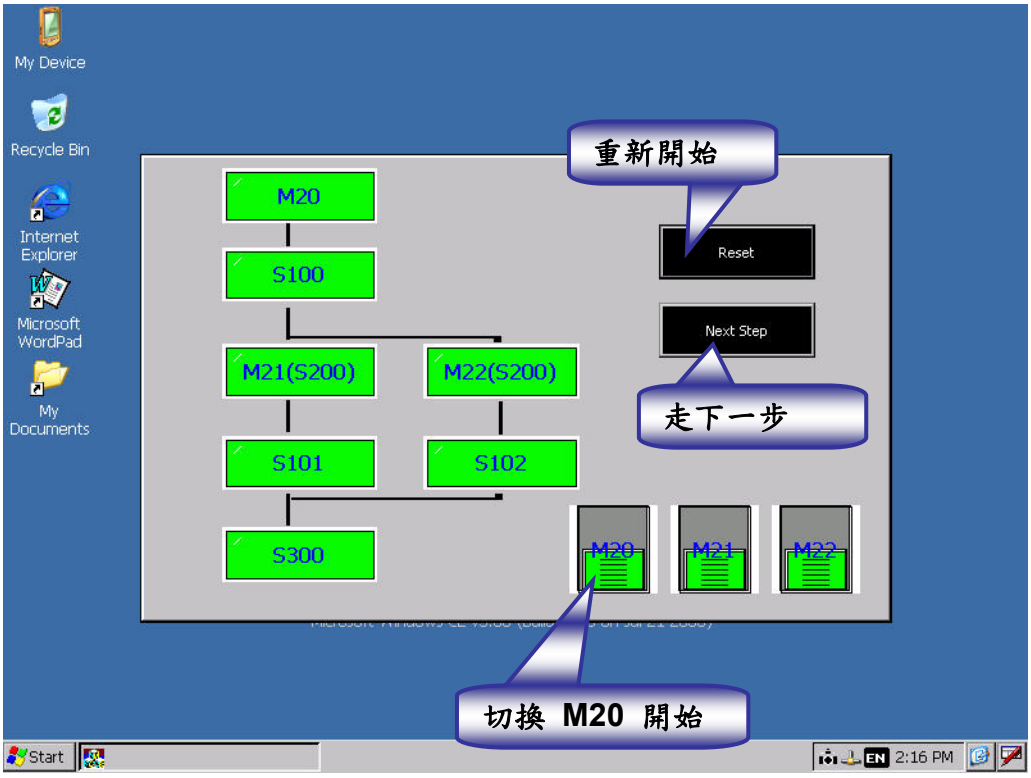
將圖換為 PIC25.bmp(紅線圖)

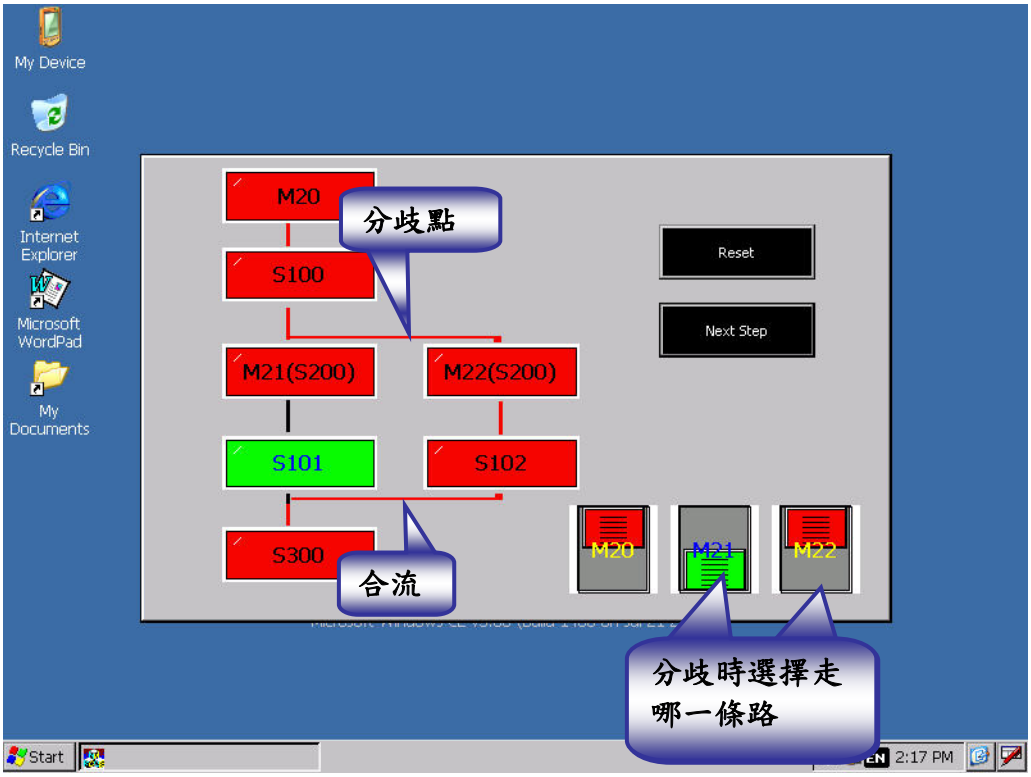
換圖旗標開啟(紅線圖)

設定為階段 4

2.14.3 專案啟始與測試

首先要將圖檔存放到 PAC 中的 EzProg_Path\EzProg-I\EzHMI\BMP
按熱鍵 F5 連線下載執行:(請先確認網路設定是正確的)，連線除錯方法請參閱附錄 連線除錯方法,如果沒有其他異常，在 PAC 畫面出現如下對執行畫面。




My Device
Recycle Bin
Internet Explorer
Microsoft WordPad
My Documents
M20
分歧點
S100
M21(S200)
M22(S200)
S101
S102
S300
合流
Reset
Next Step
M20
M21
M22
分歧時選擇走哪一條路
Start
2:17 PM