

優方科技股份有限公司  
Best Solution Inc.



BS8XXXXA  
觸控軟件包使用手冊  
2015/01/20  
V4.13

## 目錄

1.0 版本異動.....	3
2.0 軟件包介紹.....	4
2.1 軟件包結構 / 命名規則 .....	4
2.2 觸控軟件包介紹 .....	4
2.2.1 功能介紹.....	4
2.2.2 佔用資源.....	6
2.2.3 參數介紹.....	7
2.2.3.1 參數列表.....	7
2.2.3.2 參數說明.....	8
2.2.4 函數介紹（請參考 *.AEX 或 *.CEX 檔的內容） .....	9
2.2.4.1 資料 RAM .....	9
2.2.4.2 標誌位 .....	9
2.2.4.3 副程序.....	10
2.3 共用軟件包（基本功能模塊）介紹.....	11
2.3.1 MAIN_PROGRAM .....	11
2.3.2 USER_PROGRAM.....	11
2.4 參數定義檔介紹（TKS_GLOBE_VARIES.INC;TKS_GLOBE_VARIES.H） ..	12
3.0 軟件包使用方法.....	13
3.1 使用流程 .....	13
3.2 設計一個專案 .....	14
3.3 使用軟件包注意事項 .....	16
4.0 附錄.....	17
4.1 軟件包適用型號 .....	17
4.2 ROM / RAM 宣告方式（更詳細的說明，請參考 HT-IDE3000 使用手冊）	17
4.3 設計一個新的軟件包（功能模塊） .....	18
4.4 常用軟件包介紹 .....	18
4.5 其它補充事項 .....	19

## 1.0 版本異動

日期	版本	適用型號
2014/03/31	V413	BS83A04A;BS83B08A;BS83B12A;BS83B16A BS84B08A;BS84C12A BS82C16A;BS82D20A
2014/01/13	V413	BS83A04A;BS83B08A;BS83B12A;BS83B16A BS84B08A;BS84C12A; BS82C16A;BS82D20A
2015/01/20	V413	BS83A04A;BS83B08A;BS83B12A;BS83B16A BS84B08A;BS84C12A; BS82C16A;BS82D20A BS66F340;BS66F350;BS66F360; BS67F340;BS67F350;BS67F360

## 2.0 軟件包介紹

新一代的軟件包，不再單指只有觸控功能的軟件包，而是將週邊的應用功能都模塊化而這些功能可藉由原廠；代理商或協力廠商的工程人員，共同來開發並包裝成軟件包的形式，再提供給有需要的開發者使用，不需重複開發性質相同的軟件功能，以加速產品開發，降低觸控應用的門檻。

## 2.1 軟件包結構 / 命名規則

為達到功能模塊化，形成不同的軟件包的目的，首先要建立的就是“規則”，包含檔案的組成結構有哪些？以及名稱要如何取？才能讓使用者一看就知道軟件包的功能是什麼！軟件包的組成結構至少要有 4 個檔案（寫程序所需用到的檔案，未來會視情況增加其它說明文件，例如：原理圖……等），且主檔名皆相同，只有副檔名不同，用以區分不同的用途，這 4 個檔案分別為：

**xxxx.ASM** 軟件包的匯編程序檔，若使用 C code 時，則.ASM 用 .C 取代即可，

若有保密需求，則用.OBJ 檔代替，例如：觸控軟件包是提供.OBJ 型式

**xxxx.INC** 程序檔若有需要參考其它軟件包的參數或名稱或功能時，需將該軟件包的外部參考檔（.AEX 或.CEX 檔）加到.INC 檔內

**xxxx.CEX** 此軟件包提供給其它軟件包參考的參考檔（C 語言型式）

**xxxx.AEX** 此軟件包提供給其它軟件包參考的參考檔（匯編型式）

除了具備上述的檔案外，為方便管理未來持續增加的軟件包，必需將這些檔案放置在相同的資料夾內，且資料夾名稱需和上述檔案的主檔名相同，例如下圖所示



資料夾名稱

資料夾內含檔案

檔案命名（資料夾名稱）越是貼近實際功能描述，越能讓使用者一看就能大致明白軟件包的功能或版別或適用性為何！

## 2.2 觸控軟件包介紹

### 2.2.1 功能介紹

#### 觸摸鍵硬件控制

設定相關的暫存器；觸摸信號的取樣時間及觸摸鍵的掃描順序

### **觸摸信號取樣及濾波**

從每組觸摸硬件的第一個鍵同時開始掃描，在預設的時間長度內取得一次觸摸鍵的信號，接著掃描第二鍵；第三鍵；第四鍵；再回到第一鍵，依此順序循環不斷讀取觸摸鍵信號，取得信號後，再判斷每次信號間的差異，取出信號的峰值及平均值，再將峰值信號以移動平均法處理後，可得到二組穩定的信號，並以穩定的峰值為主要信號，平均值為次要信號來判斷按鍵的觸發邏輯及信號受干擾程度。

### **開機平衡**

上電後，以二分逼近法的方式，調整內部平衡電容，直到與外部環境（雜散）電容一致後，即為平衡，可使抗電壓波動性能達到最佳的效果。

### **按鍵邏輯判斷**

依照預設的閾值參數，來判斷觸摸鍵按下及鬆開的邏輯，另外按下的條件有分兩種：一是所有按下的鍵都承認，二是只承認變量最大的鍵，且不能超過或等於四個鍵，否則所有按鍵均不承認。

### **去抖判斷**

觸摸鍵壓下或鬆掉的信號必需維持（超過或低於閾值）多久才的時間才承認按鍵態狀的判斷！

### **電源干擾判斷**

利用濾波功能所得到的峰值(波峰和波谷)，將這兩組值相互比對，正常情況下，此兩組值幾乎相等，一旦電源上出現信號較強的干擾源時，此兩組值會開始產生差距，軟件會判斷當差距到達一定的程度時，視為電源干擾，檢測到有干擾後，會依參數的設定值，決定是否要關閉按鍵的輸出，避免誤觸發。

### **按鍵操作時間判斷**

判斷觸摸鍵持續按住的時間是否超過參數設定的時間，若超過則立即清除該鍵的狀態，變成無按鍵！

### **省電功能處理**

在觸摸鍵一段時間不操作時，進入睡眠，再利用看門狗功能喚醒，喚醒後立即掃描並判斷觸摸鍵是否判按下，若沒按，再進入睡眠，等待下次喚醒，依此模式循環操作，以平均電流的觀念，控制觸摸晶片的耗電流。

### **自動環境校正**

依參數所設定的時間在沒按鍵的態狀下對環境做校正，校正方法採逼近法，將參考信號逼近實際環境信號，另外也可設定成在有按鍵的狀態下也會動態校正參考值，以維持穩度的觸發條件

### **平衡狀態維持**

當一段時間內都沒有任何按鍵觸發時，會重複檢查平衡態狀，若檢查到任一鍵有不平衡的現象時，會重新啟動開機平衡的功能。

## 2.2.2 佔用資源

IC	ROM	RAM	Stack	Interrupt	Other
BS83A04A	92% ( total 1K )	82% (Total 96 bytes) (60H-0AEH)	3	Time Base	MP1 ; IAR1 ; MP0 ; IAR0
BS83B08A	60% ( total 2K )	78% (Total 160 bytes) (60H-0DCH)	3	Time Base	MP1 ; IAR1 ; MP0 ; IAR0
BS83B12A	63% ( total 2K )	61% (Total 288 bytes) Bank0 (60H-0AFH) Bank1 (80H-0DFH)	3	Time Base	MP1 ; IAR1 ; MP0 ; IAR0
BS83B16A	65% ( total 2K )	77% (total 288 bytes) Bank0 (60H-0BFH) Bank1 (80H-0FFH)	3	Time Base	MP1 ; IAR1 ; MP0 ; IAR0
BS84B08A	40% ( total 3K )	43% (Total 288 bytes) Bank0 (60H-084H) Bank1 (80H-0D7H)	3	Time Base	MP1 ; IAR1 ; MP0 ; IAR0
BS84C12A	31% ( total 4K )	45% (Total 384 bytes) Bank0 (60H-08BH) Bank1 (80H-0A3H) Bank2 (80H-0DFH)	3	Time base	MP1 ; IAR1 ; MP0 ; IAR0
BS82C16A	32% ( total 4K )	43% (Total 512 bytes) Bank0 (80H-0AFH) Bank1 (80H-0AFH) Bank2 (80H-0FFH)	3	Time base0	MP1 ; IAR1 ; MP0 ; IAR0
BS82D20A	17% ( total 8K )	35% (Total 768 bytes) Bank0 (80H-0B6H) Bank1 (80H-0BBH) Bank2 (80H-0CFH) Bank3 (80H-0CFH)	3	Time base0	MP1 ; IAR1 ; MP0 ; IAR0
BS67F340	37% ( total 4K )	46% (Total 512 bytes) Bank0 (80H-0AEH) Bank1 (80H-0AFH) Bank2 (80H-0BFH) Bank3 (80H-0CDH)	4	Time base0 Touch Key_Int	MP1L ; MP1H; MP2H ; MP2L ;
BS67F350	19% ( total 8K )	37% (Total 768 bytes) Bank0 (80H-0B4H) Bank1 (80H-0BBH) Bank2 (80H-0CFH) Bank3 (80H-0DEH)	4	Time base0 Touch Key_Int	MP1L ; MP1H; MP2H ; MP2L ;
BS67F360	10% ( total 16K )	37% (Total 1024 bytes) Bank0 (80H-0BFH) Bank1 (80H-0D4H)	4	Time base0 Touch Key_Int	MP1L ; MP1H; MP2H ; MP2L ;

		Bank2 (80H-0EFH) Bank3 (80H-0FFH)			
BS66F340	36% ( total 4K )	36% (Total 512 bytes) Bank0 (80H-0AAH) Bank1 (80H-0A3H) Bank2 (80H-0AFH) Bank3(80H-0BDH)	4	Time base0 Touch Key_Int	MP1L ; MP1H; MP2H ; MP2L ;
BS66F350	19% ( total 8K )	37% (Total 768 bytes) Bank0 (80H-0B4H) Bank1 (80H-0BBH) Bank2 (80H-0CFH) Bank3 (80H-0DEH)	4	Time base0 Touch Key_Int	MP1L ; MP1H; MP2H ; MP2L ;
BS66F360	10% ( total 16K )	37% (Total 1024 bytes) Bank0 (80H-0BFH) Bank1 (80H-0D4H) Bank2 (80H-0EFH) Bank3 (80H-0FFH)	4	Time base0 Touch Key_Int	MP1L ; MP1H; MP2H ; MP2L ;

### 2.2.3 參數介紹

#### 2.2.3.1 參數列表

參數	功能	數值範圍	建議值
DebounceTimes	按鍵去抖設定	0~15	5~7
AutoCalibrationPeriod	自動校準時間設定	0~15	4~8
HighSensitive	高感度設定	0=關閉 ; 1=開啟	0
MaximumKeyHoldTime	按鍵最長反應時間	0~15	1~3
FastResponse	快速反應設定	0=關閉 ; 1=開啟	0
AutoFrequencyHopping	自動跳頻設定	0=關閉 ; 1=開啟	1
OneKeyActive	單鍵輸出設定	0=關閉 ; 1=開啟	視應用功能設定
PowerSave	省電功能設定	0=關閉 ; 1=開啟	視省電需求設定
NoiseProtect	干擾保護設定	0=關閉 ; 1=開啓	1
MovingCalibration	動態校正設定	0=正常校正 1=動態校正	1
Key Threshold	觸發閾值	範圍 8 ~ 255	16~40

\*以上參數均放在 TKS\_GLOBE\_VARIES.INC 內，可手動修改參數值

### 2.2.3.2 參數說明

#### DebounceTimes

設定去抖次數(值越大，按鍵反應越慢)以 10mS 為單位

(判斷\_SCAN\_CYCLEF 標誌位)計1數一次，從最小值 0=30mS；1=40mS；2=50 mS……15=180mS，每階增加 10 mS。當 DebounceTimes=0 時。

#### AutoCalibrationPeriod

設定校正時間，以 80mS 為單位， 0=80 mS ； 1=160 mS……15=1280 mS 當設定的校正時間一到，且無按鍵時，即校正一次環境（更新參考值）。

#### HighSensitive

設定感度高低

設為 0 時： 正常感度。

設為 1 時： 感度放大一倍。

#### MaximumKeyHoldTime

設定最長動作時間，以 4 秒為單位，計算最長動作時間限制

0=關閉此功能；1=4 秒……15=60 秒。

#### FastResponse

設定按鍵濾波速率,較快的濾波速度,可加快按速的反應速度

0=關閉

1=開啓

#### AutoFrequencyHopping

設定硬體自動跳頻開啓或關閉。

0=關閉

1=開啓

#### OneKeyActive

設定單鍵輸出功能（不支援 BS83A04A）

設為 0 時，按多個觸摸鍵都有對應的按鍵旗標輸出。

設為 1 時，按多個觸摸鍵只有一個鍵成立，變量最大的鍵成立，若超過 4 個(含)以上的鍵被按下，則視為所有觸摸鍵都無效。

#### PowerSave

設定省電模式開啓或關閉。

0=關閉

1=開啓

#### NoiseProtect

設定 AC 雜訊干擾保護功能開啓或關閉，

開啓時，檢測到干擾時，觸摸鍵強制失效

0=關閉



1=開啓

### **MovingCalibration**

設定動態更新環境功能，

0=觸摸鍵按下後不更新“參考值”

1=動態更新“參考值”，不論是否有觸摸鍵成立，都依當前環境進行校正。

### **Key Threshold**

設定觸發閾值，值越大，感度越低，值越小感度越高。

## 2.2.4 函數介紹（請參考 \*.AEX 或 \*.CEX 檔的內容）

### 2.2.4.1 資料 RAM

#### **\_DATA\_BUF[0]~\_DATA\_BUF[7]**（可讀寫）

軟件包所使用的資料暫存器，可用來做資料交換使用，但不可儲存資料

#### **\_GLOBE\_VARS**（可讀寫）

存放“觸摸鍵參數”及“觸發閾值”的暫存器

#### **\_KEY\_REF**（只能讀）

存放觸摸鍵“參考值”的暫存器

#### **\_KEY\_STATUS**（只能讀）

存放觸摸鍵信號“目前狀態”的暫存器

#### **\_TKS\_TIME\_BASE**（只能讀）

觸摸功能使用的計時器，在 TIME BASE 中斷裏計數

#### **\_STANDBY\_TIME**（可讀寫）

鬆掉所有觸摸鍵後，進入省電模式的時間計數器

#### **\_KEY\_IO\_SEL**（可讀寫）

設定 PIN 腳為“觸摸鍵”或“I/O 口”的設定暫存器

#### **\_SAMPLE\_TIMES**（只能讀）

觸摸信號取樣週期計數器

#### **\_KEY\_DATA**（只能讀）

觸摸鍵觸發狀態暫存器

### 2.2.4.2 標誌位

#### **\_SCAN\_CYCLEF**（只能讀）

所有觸摸鍵掃描一個週期標誌位，每週期約 10mS（快速模式為 5mS），也是 KeyDebounce 的計時基準

#### **\_ANY\_KEY\_PRESSF**（只能讀）

任一觸摸鍵按下標誌位

#### **\_TKS\_ACTIVEF**（只能讀）

觸摸軟件包完成初始化，並開始運行標誌位。

#### **\_TKS\_63MSF**（只能讀）

63mS 計時器標誌位。

#### **\_TKS\_250MSF**（只能讀）

250mS 計時器標誌位。

#### **\_TKS\_500MSF**（只能讀）

500mS 計時器標誌位

## \_FORCE\_CALIBRATEF (可讀寫)

強制所有觸摸鍵，對環境做重校正標誌位

### 2.2.4.3 副程序

## \_BS8xxxxA\_LIBVxx\_INITIAL ; \_LIBRARY\_RESET

功能 : 觸摸功能初始化。

輸入 : 無

輸出 : 無

Stack : 1

說明 : 主程序初始化時，需調用此程序，避免 Library 功能不穩定。

## \_BS8xxxxA\_LIBVxx

功能 : 主功能，負責信號取樣；濾波；演算法處理及內部計時。

輸入 : 無

輸出 : 無

Stack : 3

說明 : 此程序為軟件包主要功能，一定要調用，否則 “觸摸鍵” 功能無法實現。

## \_GET\_KEY\_BITMAP

功能 : 讀取 “觸摸鍵” 狀態；並以 bitmap 的方式輸出，相對應的 bit 為 0=放開；1 = 壓下。

輸入 : 無

輸出 : \_DATA\_BUF[0]；\_DATA\_BUF[1]  
\_DATA\_BUF[0] = KEY 8 (MSB) ~ KEY1 (LSB)。  
\_DATA\_BUF[1] = KEY16 (MSB) ~ KEY9 (LSB)。

## \_GET\_ENV\_VALUE

功能 : 讀取觸摸鍵 “信號” 值。

輸入 : ACC =0 (第 1 KEY) ; 1 (第 2 KEY) .....20 (第 20 KEY)

輸出 : \_DATA\_BUF[1]；\_DATA\_BUF[0]

Stack : 1

說明 : 1 筆環境值佔兩個 byte

## \_GET\_REF\_VALUE

功能 : 讀取觸摸鍵 “參考” 值。

輸入 : ACC =0 (第 1 KEY) ; 1 (第 2 KEY) .....20 (第 20 KEY)

輸出 : \_DATA\_BUF[1]；\_DATA\_BUF[0]

Stack : 1

說明 : 1 筆參考值佔兩個 byte。

## \_GET\_RCC\_VALUE

功能：讀取內部的平衡電容值。  
輸入：ACC =0（第 1 KEY）； 1（第 2 KEY）……15（第 20 KEY）  
輸出：\_DATA\_BUF[0]  
Stack：1  
說明：無

#### \_GET\_LIB\_VER

功能：讀取 觸摸軟件包 版本資訊。  
輸入：無  
輸出：無  
Stack：1  
說明：

#### \_GET\_KEY\_AMOUMT

功能：讀取 觸摸鍵 的總數。  
輸入：無  
輸出：無  
Stack：1  
說明：無

## 2.3 共用軟件包（基本功能模塊）介紹

### 2.3.1 MAIN\_PROGRAM

程序的開始，每個專案都必需使用的功能模塊，主要功能有：  
IC 硬件初始化（不同的 IC 型號，有不同的初始化程序）  
功能初始化（調用所有被加入的功能模塊的初始化程序）  
載入觸控軟件包的參數  
執行每個功能模塊的功能

### 2.3.2 USER\_PROGRAM

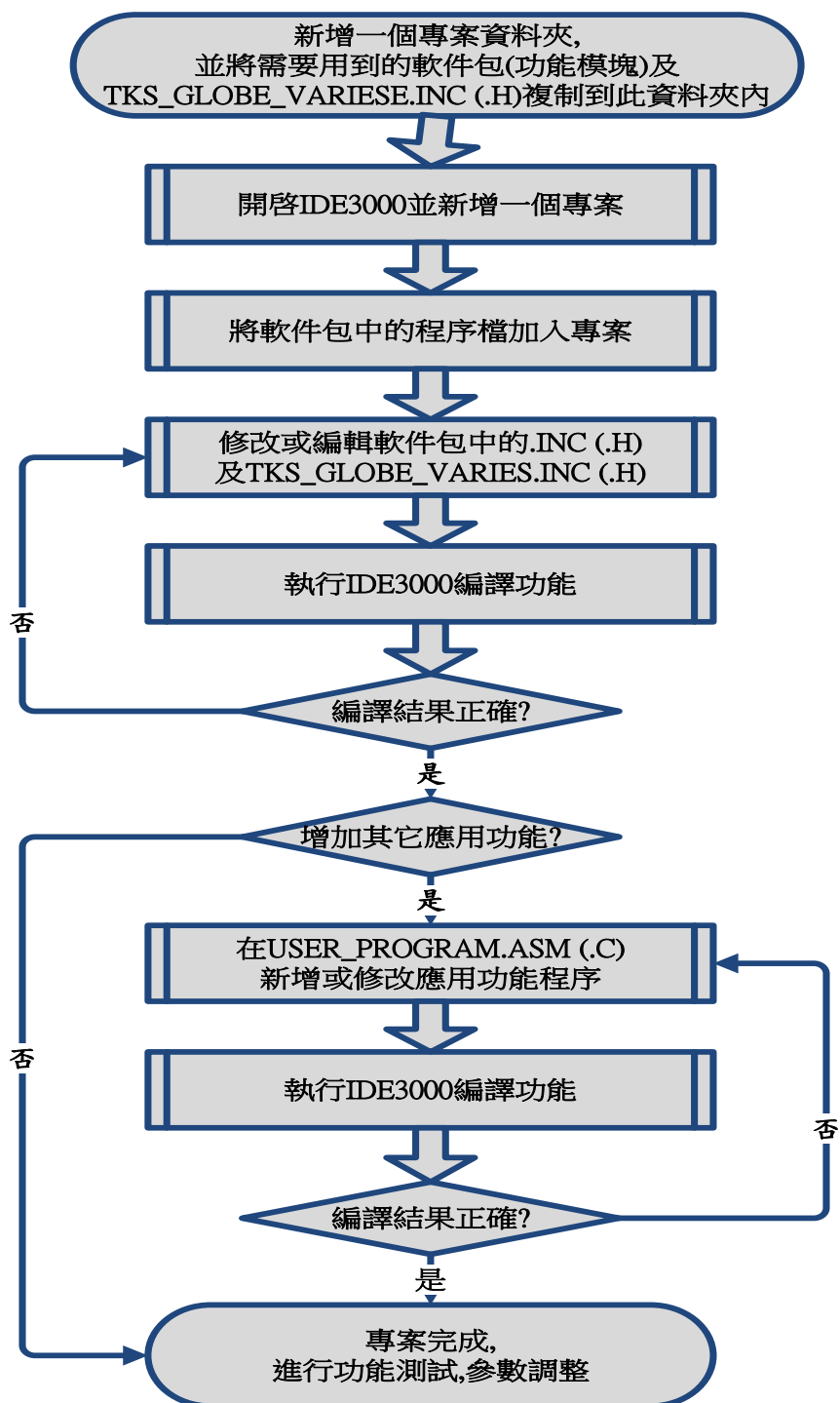
空的軟件包，提供簡單幾條指令範例及軟件包的基礎結構做為使用上的參考，使用者可在此軟件包內直接編寫程序，開發新的應用功能，也可以套用此軟件包的格式，在編寫完程序後，變更資料夾及其它檔案的主檔名，並適當修改其它檔案的內容，包裝成另一個新的軟件包。

## 2.4 參數定義檔介紹 (TKS\_GLOBE\_VARIES.INC;TKS\_GLOBE\_VARIES.H)

參數設定檔的功能是用來存放整個專案中可被設定或修改的參數，這些參數都來自於所有被選用進來使用的每個軟件包（功能模塊），例如：一個專案中套用 **UART\_TEST**；**MAIN\_PROGRAM**；**BS83B16A\_LIBV413** 三個功能模塊，則其所屬的可設定參數，都會存放在參數定義檔內！若使用匯編程序開發，則參數在 **TKS\_GLOBE\_VARIES.INC**；若是 C 程序開發，則參數是存放在 **TKS\_GLOBE\_VARIES.H**，使用者可視實際應用在此檔案內調整相關的參數，例如：感度；選項；Baudrate……等等.不用到個別程序中修改參數！增加使用上的方便性。

## 3.0 軟件包使用方法

### 3.1 使用流程



## 3.2 設計一個專案

以 BS83B16A-3 為應用，開 16 鍵，使用匯編程序開發一個有任何鍵按下時，在 PA1 輸出高電平的簡單應用功能。

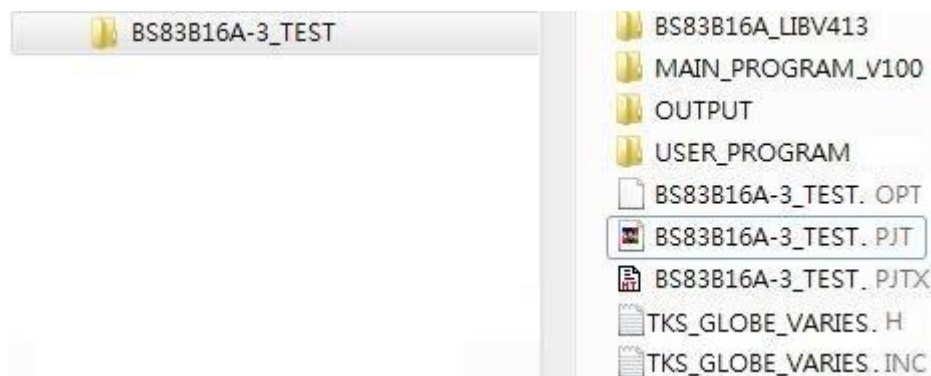
步驟：

一. 新增一個資料夾 BS83B16A-3\_TEST

並將軟件包 MAIN\_PROGRAM\_V100；BS83B16A\_LIBV413；USER\_PROGRAM 及 TKS\_GLOBE\_VARIES.INC（.H）複製到 BS83B16A-3\_TEST 資料夾內



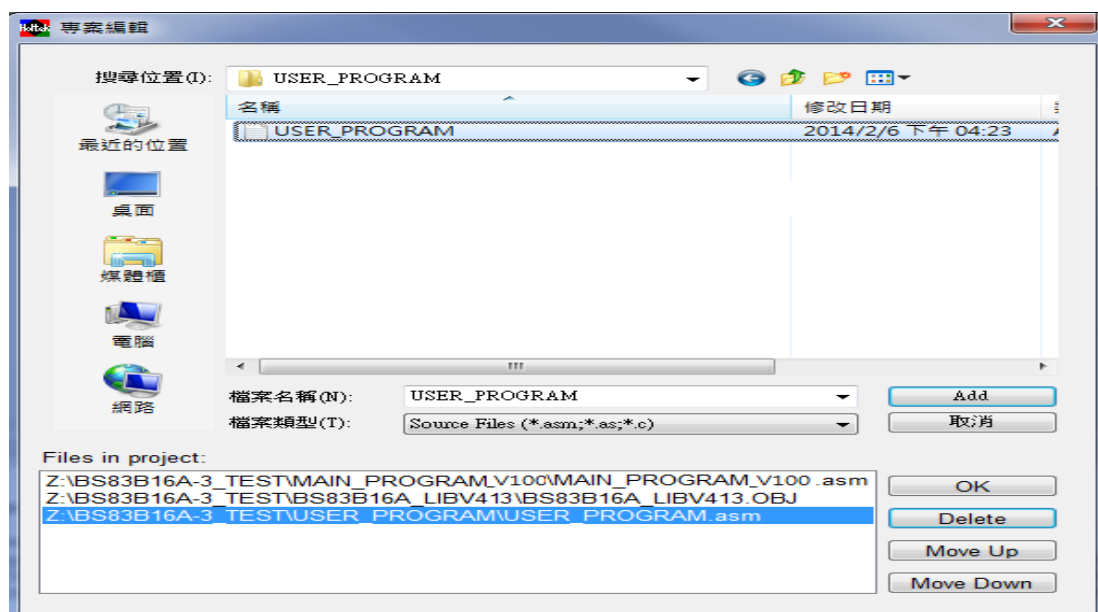
二. 開啓 HT-IDE3000，並新增一個名為 BS83B16A-3\_TEST 的專案，並將檔案的輸出指定到\OUTPUT\資料夾



三. 在“專案\編輯”的功能下，

將 MAIN\_PROGRAM\_V100.ASM ； BS83B16A\_LIBV413.OBJ ；

USER\_PROGRAM.ASM 加入專案。（註:任何被加進來的程序都會佔用 RAM/ROM 資源,例如:感度測試程序 UART\_TEST,即使程序中不調用,也會佔資源）



#### 四. 修改 MAIN\_PROGRAM\_V100\MAIN\_PROGRAM\_V100.INC 內容

```
INCLUDE "..\BS83B16A_LIBV413\BS83B16A_LIBV413.AEX"
#define EXTEND_FUNCTION_1A_INITIAL _BS83B16A_LIBV413_INITIAL
#define EXTEND_FUNCTION_1A _BS83B16A_LIBV413

INCLUDE "..\USER_PROGRAM\USER_PROGRAM.AEX"
#define EXTEND_FUNCTION_1B_INITIAL _USER_PROGRAM_INITIAL
#define EXTEND_FUNCTION_1B _USER_PROGRAM
```

#### 修改 USER\_PROGRAM\USER\_PROGRAM.INC 內容

```
INCLUDE "..\BS83B16A_LIBV413\BS83B16A_LIBV413.AEX"
```

BS83B16A\_LIB413 未參考其它軟件包的參數，所以不需修改.INC 內容

#### 修改 TKS\_GLOBE\_VARIES.INC 中的參數定義

```
;-SELECT IC BODY & INCLUDE REFERENCE FILE
#define _V413
#define _BS83B16A_
INCLUDE BS83B16A-3.INC

;-----
;-DEFINE SYSTEM CLOCK -
;-----
#define SystemClock 0 ;0=8MHZ
;1=12MHZ
;2=16MHZ

;-----
;-TOUCH KEY LIBRARY VARIES DEFINE
;-----
;-numeric operate ; range 0 ~ 15;
#define DebounceTimes 5

;-numeric operate ; range 0 ~ 15;
#define AutoCalibrationPeriod 7

;-bit operate ; range 0/1 ;
#define HighSensitive 0

;-numeric operate ; range 0 ~ 15;
#define MaximumOnTime 15

;-bit operate ; range 0/1 ;
#define FastResponse 0

;-bit operate ; range 0/1 ;
#define AutoFrequencyHopping 1

;-bit operate ; range 0/1 ;
#define OneKeyActive 0

;-bit operate ; range 0/1 ;
#define PowerSave 0

;-bit operate ; range 0/1 ;
#define NoiseProtect 0

;-bit operate ; range 0/1 ;
#define MovingCalibration 0
;-Key threshold define
#define Key1Threshold 16 ;range 8 ~ 255
#define Key2Threshold 16
#define Key3Threshold 16
#define Key4Threshold 16
;--
#define Key5Threshold 16
#define Key6Threshold 16
#define Key7Threshold 16
#define Key8Threshold 16
;--
#define Key9Threshold 16
#define Key10Threshold 16
#define Key11Threshold 16
#define Key12Threshold 16
;--
#define Key13Threshold 16
#define Key14Threshold 16
#define Key15Threshold 16
#define Key16Threshold 16
;-----
;-DEFINE PIN AS I/O OR TOUCH INPUT
;-----
#define IO_TOUCH_ATTR 000000000000000011111111111111B;0=IO ; 1=TOUCH INPUT
;KEY 3 ~ 2 ~ 2 ~ 1 ~ 1 ~ 0 ~ 0 ~ 0 ;KEY32~KEY1
; 2 4 0 6 2 8 4 1
```

五.執行“\編譯\全部重建”功能！

六.將功能編寫在 USER\_PROGRAM.ASM 內

```
INCLUDE USER_PROGRAM.INC

PUBLIC _USER_PROGRAM_INITIAL
PUBLIC _USER_PROGRAM

;=====
;=USER DATA DEFINE      =
;=====
USER_DATA .SECTION 'DATA'

;=====
;=USER PROGRAM           =
;=====
USER_PROGRAM .SECTION 'CODE'

;*****
; * USER_PROGRAM_INITIAL *
;*****
_USER_PROGRAM_INITIAL:
    CLR    PAC.1
    CLR    PA.1
    RET

;*****
; * USER PROGRAM ENTRY *
;*****
_USER_PROGRAM:
    SNZ    _SCAN_CYCLEF
    RET
    CALL   GET_KEY_BITMAP
    MOV    A,_DATA_BUF[0]
    OR     A,_DATA_BUF[1]
    SZ     Z
    CLR    PA.1
    SNZ    Z
    SET    PA.1
    RET
```

七.再次執行“\編譯\全部重建”功能！

八.專案完成，測試功能！若功能不正確則回到步驟六，進行 Debug 與修改，若感度不足，可調整 TKS\_GLOBE\_VARIES.INC 內的閾值，再回到步驟七。

### 3.3 使用軟件包注意事項

#### 指定正確的路徑

資料夾路徑需注意是否正確，在 \*.INC 或 \*.H 中的

INCLUDE “..| 軟件包名稱 | 軟件包名稱.AEX ” 有包含路徑宣告，

其中 .. 是指回到上層路徑，再指到軟件包資料夾內的 \*.AEX 檔

#### 程序進入點宣告

在 *MAIN\_PROGRAM\_V100.INC* 內，除了加入其它軟件包的參考檔外，需再指定軟件包的程序進入點（包括初始化程序及功能程序）

```
INCLUDE "..\BS83B16A_LIBV413\BS83B16A_LIBV413.AEX"
#define EXTEND_FUNCTION_1A_INITIAL _BS83B16A_LIBV413_INITIAL
#define EXTEND_FUNCTION_1A        _BS83B16A_LIBV413

INCLUDE "..\USER_PROGRAM\USER_PROGRAM.AEX"
#define EXTEND_FUNCTION_1B_INITIAL _USER_PROGRAM_INITIAL
#define EXTEND_FUNCTION_1B        _USER_PROGRAM
```



## 程序位址宣告

程序位址（含中斷位址）採絕對位址宣告，不可用 **ORG** 方式宣告

## 資料 RAM 宣告

資料 RAM 採 **DB** ； **DBIT** 的方式宣告

註：程序位址和資料 RAM 的宣告，請參考附錄或 IDE3000 使用手冊

## 4.0 附錄

### 4.1 軟件包適用型號

BS83A04A ； BS83B08A ； BS83B12A ； BS83B16A  
BS84B08A ； BS84C12A  
BS82C16A ； BS82D20A

### 4.2 ROM / RAM 宣告方式 （更詳細的說明，請參考 HT-IDE3000 使用手冊）

#### ROM 宣告方式

##### 程式段宣告方式

PROGname .SECTION 'CODE'

用法： *SAMPLE\_CODE* .SECTION 'CODE'

說明： 宣告接下來的程序都屬於 *SAMPLE\_CODE* 這個程式段的範圍

##### 中斷宣告方式

INTname .SECTION AT *INTaddress* 'CODE'

用法： *TIME\_BASE\_INT* .SECTION AT *014H* 'CODE'

*JMP TBI\_PROG*

說明： 宣告 *TIME BASE* 中斷的進入點在 *014H*，接著跳到中斷處理程序執行

##### 查表宣告方式

TABLEname .SECTION AT TABLEaddress 'CODE'

用法： *TABLE* .SECTION AT *0700H* 'CODE'

*DC 000H, 001H, 002H.....*

說明： 宣告查表 *TABLE* 的位址在 *0700H*，並定義查表的內容

#### RAM 資料段宣告方式

##### RAM BANK 宣告

RAMBANK number BANKname

BANKname .SECTION AT RAMaddress 'DATA'

用法 1 : RAMBANK 0 LIBRARY\_RAMBANK0

LIBRARY\_RAMBANK0 .SECTION AT 060H 'DATA'

說明 : 宣告 LIBRARY\_RAMBANK0 在 RAM BANK0 的 60H 為起始位址

用法 2 : RAMBANK 1 LIBRARY\_RAMBANK1

LIBRARY\_RAMBANK1 .SECTION 'DATA'

說明 : 宣告 LIBRARY\_RAMBANK1 在 RAM BANK1 但不指定起始位址

### 資料宣告

RAMname DB ?

用法 : TIMER DB ?

說明 : 宣告 TIMER 為一個 byte 的 RAM

FLAGname DBIT

用法 : FLAG DBIT

說明 : 宣告 FALG 為一個 bit 的標誌位

## 4.3 設計一個新的軟件包（功能模塊）

直接修改 USER\_PROGRAM.xxx 的名稱及內容（程序功能；宣告資料……等）

步驟：

一. 將 USER\_PRGORAM.xxx 全部更換成新的名稱.

例如 : 4KEY\_4LED\_V100.xxx

二. 修改 \*.AEX ; \*.CEX ; \*.INC 的內容以配合新的名稱

三. 新增一個專案，將 MAIN\_PROGRAM ; TOUCH\_LIBRARY；新的功能模塊及其它有用到的功能模塊加入專案中

四. 在 IDE3000 執行編譯功能，確定語法都正確.

五. 開始編寫功能

註：參考軟件包使用說明

## 4.4 常用軟件包介紹

### UART\_TEST

搭配平台使用的測試軟件包，可以觀察信號的變化，使用 UART 溝通協議提供 Baudrate 及 TX PIN 的設定（在 TKS\_GLOBE\_VARIES.INC 內）

## LIBV413\_IIC\_SALVE\_TEST

搭配平台使用的測試軟件包，可以觀察信號的變化及填入新的參數  
使用 IIC 溝通協議

註：詳細的程序內容，請直接參考 \*.ASM 檔（源程序）

## 4.5 其它補充事項

暫無補充事項