

Please keep in Confidential

Application Program Interface of PenPower Recognition System

(Preliminary version date: 01/24/2002)

(Revised Date: 03/11/2002)

(Revised Date: 04/06/2004)

(Revised Date: 03/22/2006)

增加日文平假名與片假名的設定。

增加俄文字母大小寫的設定。

增加詞庫 API。

Assumption:

The upper left corner of the Digitizer is the origin.

```
typedef struct point_type {  
    short  x;  
    short  y;  
} POINT_TYPE;
```

```
typedef POINT_TYPE* PPOINT_TYPE;
```

```
type struct tagHWRDATA{  
    signed char *pInternalDict; /* predefined database address for HWRE */  
    signed char *pExternalDict; /* supplement database for HWRE */  
    signed char *pPrivate;      /* RAM address for HWRE */  
} HWRData;  
typedef HWRData* PHWRData;
```

```
typedef struct tagHWRBOX  
{  
    short  left;  
    short  top;  
    short  right;  
    short  bottom;  
} HWRBOX;  
typedef HWRBOX* LPHWRBOX;
```

```
#ifndef BYTE
#define BYTE    unsigned char
#endif
```

```
#ifndef WORD
#define WORD    unsigned short
#endif
```

```
#ifndef DWORD
#define DWORD    unsigned long
#endif
```

返回值的定義：

/* Following are the return value definition for the mandatory API functions */

```
#define STATUS_OK                0x0000    /* NO Error */
#define STATUS_NO_DICT            0x0001    /* db addr undefined */
#define STATUS_INVALID_DICT      0x0002    /* db format error */
#define STATUS_INVALID_TYPE      0x0003    /* unsupported recognition type */
#define STATUS_INVALID_NUMBER    0x0004    /* invalid candidate count */
#define STATUS_NO_ENOUGH_MEMORY  0x0005    /* RAM size too small */
#define STATUS_INVALID_HWRDATA_PTR 0x0006    /* Invalid object pointer */
#define STATUS_INVALID_PRIVATE_PTR 0x0007    /* Invalid RAM pointer */
#define STATUS_INVALID_BOX_PTR   0x0008    /* Invalid Box pointer */
#define STATUS_INVALID_TRACE_PTR 0x0009    /* Invalid Ink Pointer */
#define STATUS_INVALID_RESULT_PTR 0x000A    /* Invalid Cand Array pointer */
#define STATUS_INVALID_DBENDIAN   0x000B    /* db Endian type mismatch */
#define STATUS_UNKNOWN           0x00FE    /* unknown error */
```

詞庫選項：

/* Following are the Phrase Module Area options definitions */

```
#define TRADITIONAL_PHRASE      0x0000    /* Search Traditional Phrase */
#define SIMPLIFIED_PHRASE       0x0001    /* Search Simplified Phrase */
```

```

/* Following are the mandatory API functions */
unsigned short PPHWRInit(HWRData* pData);
unsigned long PPHWRGetRamSize(void);
unsigned short PPHWRSetType(HWRData* pData, DWORD type);
unsigned short PPHWRSetBox(HWRData* pData, LPHWRBOX pbox);
unsigned short PPHWRSetCandidateNum(HWRData* pData, WORD number);
unsigned short PPHWRRecognize(HWRData* pData, WORD* pTrace, WORD* pResult);
int find_phrase(unsigned short code,
                unsigned short size_of_return_buffer,
                unsigned short *return_buffer,
                signed char area_flag);

```

程序調用接口：

1. unsigned long PPHWRGetRamSize(void)

輸入：無。

輸出：在調用 PPHWRInit 的時候，物件中的內存指針所標定的矩陣的大小。

描述：在調用初始化程序之前，必須透過這個程序取得適當的內存大小，以便進行動態配置內存的動作，或者是檢驗預置矩陣的大小。

2. unsigned short PPHWRInit(HWRData* pData)

輸入：識別核心物件的指針。

輸出：識別核心初始後的結果。

描述：（1）使用識別核心之前，一定要調用本程序，否則有可能會導致後續程序運行失敗，甚至死機。

（2）在調用之前，必須指定好物件之中的基本數據庫地址與內存地址。

3. unsigned short PPHWSetType(HWRData* pData, DWORD type)

輸入：識別物件指針以及識別核心必須檢查和允許返回的型別。

輸出：運行後的檢查碼。

描述：容許的型別定義如下：

型別	值	意義
LOWER_ALPHA_TYPE	1	小寫字母
UPPER_ALPHA_TYPE	2	大寫字母
NUMA_TYPE	4	數字
GENERAL_PUNCT_TYPE	8	常用標點
EXTEND_PUNCT_TYPE	16	擴展標點
GENERAL_SYMBOL_TYPE	32	常用符號
EXTEND_SYMBOL_TYPE	64	擴展標點
GESTURE_TYPE	128	手勢（特殊命令）
RADICAL_TYPE	256	偏旁部首
CHINESE_1_TYPE	512	漢字一級字庫
CHINESE_2_TYPE	1024	漢字二級字庫
CHINESE_3_TYPE	2048	GBK 三區中的漢字
CHINESE_4_TYPE	4096	GBK 四區中的漢字
IN_HK_TYPE	2048	香港增補字符集(HKSCS-2001) — 有香港字時使用
SIMPLIFIED_ONLY	4096	只返回簡體字 — 繁簡合一版本使用
TRADITIONAL_ONLY	8192	只返回繁體字 — 繁簡合一版本使用
HIRA_TYPE	16384	日文平假名
KATA_TYPE	32768	日文片假名
RUSSIAN_UPPER_ALPHA_TYPE	8192	俄文大寫字母(俄文版使用)
RUSSIAN_LOWER_ALPHA_TYPE	16384	俄文小寫字母(俄文版使用)

默認值：全部開啟（除了 IN_HK_TYPE/SIMPLIFIED_ONLY/TRADITIONAL_ONLY 以外）。

4. unsigned short PPHWRSetBox(HWRData* pData, LPHWRBOX pbox)

輸入：識別物件指針以及當次書寫格的方塊座標的指針。

輸出：運行後的檢查碼。

描述：對於英文字母的大小寫混合時，以及和位置有關的符號的判定，有絕對的影響。如果沒有適當的設置，可能會造成核心的誤判。
默認值為（0，0，60，60）。

5. unsigned short PPHWRSetCandidateNum(HWRData* pData, WORD number)

輸入：識別物件指針以及預期收到的最多候選字數目。

輸出：運行後的檢查碼。

描述：識別核心將不會返回比這個數目還多的候選字；允許的候選字數目為1~15。

默認值：10。

6. unsigned short PPHWRRecognize(HWRData* pData, WORD* pTrace, WORD* pResult)

輸入：識別物件指針，筆跡矩陣指針，以及識別結果矩陣指針。

輸出：運行後的檢查碼。

描述：（1）應用程序必須在檢查碼為STATUS_OK的情形下，才去檢驗識別結果矩陣。

（2）應用程序配置的結果矩陣的大小，必須至少比預期的最多候選字個數多2個BYTE，因為識別核心不再回傳候選字的個數，而是在結果矩陣的最後放一個0。

7. int find_phrase(unsigned short code,
 unsigned short size_of_return_buffer,
 unsigned short *return_buffer,
 signed char area_flag);

輸入：code：用來搜尋詞庫的鍵值。

size_of_return_buffer：返回結果暫存區的大小 in bytes。

return_buffer：返回結果暫存區的指針。

area_flag：繁體(0)或者是簡體(1)詞庫搜尋。

輸出：在暫存區中的結果的個數（in WORD）。

其他說明：

1：即時筆跡顯示：

上述的函式，除了” PPHWRRecognize” 以外，都只需要幾毫秒的時間就可以執行完畢，也就是說，應用程式將可以取得快速的回應。但是，” PPHWRRecognize” 將會至少使用 1 秒（視 C P U 的能力而有不同）左右。如果應用程式在識別的過程中，必須要顯示第二個書寫格的筆跡，那麼程式應該要分為兩個部分（thread/task），一個是專門處理筆跡收點以及結果顯示的部分，另外一個則是進行識別的。或者是筆跡顯示以及存放座標點的部分，由觸控筆驅動程式來執行，而應用程式則專門處理識別與結果顯示。

2：重進入機制以及記憶體再利用：

藉著傳送手寫核心內部資料結構的指針，識別核心現在可以做到” 重進入” 了。也就是說，每一個要使用手寫識別核心的 Thread/Task 都必須配置專屬的手寫區塊。當不使用識別核心的時候，可以將物件指針中的 pPrivate 所標定的區塊釋放或者挪作他用；等到要再次使用核心的時候，則再次運行一次初始化的步驟。

3：現在符號／數字／英文字母都是以他們的半形編碼當作傳回值而不是他們的全形編碼。

4：建議調用流程：

- (1) PPHWRGetRamSize -- 取得手寫核心需要的記憶體數量，以便進行配置。
- (2) PPHWRInit -- 進行手寫識別核心初始化。
必須先把物件指針（可以是全域或者是動態配置）的內容先行填寫完畢。
物件中 pInternalDict 是基本資料數據庫的地址。
pExternalDict 是擴充資料數據庫的地址（學習字，人工智慧）
pPrivate 則是步驟一所配置的內存空間。
這三個物件的內容一但有所變更，那麼就必須調用本函數一次。
- (3) PPHWRSetType/PPHWRSetBox/PPHWRSetCandidateNum

這三個函數之間不存在順序的問題，也就是說，三者可以用任何次序調用，而不會互相影響。

(4) PPHWRRecognize -- 當 A P 收集到一個字的筆跡之後（可能是透過 Timer，換格書寫，或者是點選〔識別〕按鈕來標記），就可以調用本函數，將筆跡送給核心進行識別；識別結果將會被存放在參數三所表示的陣列中。

(5) 重複回到步驟三，直到不需要手寫功能為止。

5：傳送筆跡的資料結構，以及標示資料結尾的記號的說明。

由於新的核心不需要每一個筆劃進行處理，所以 A P 必須先將所收集到的點座標存放在自己的暫存區之中，直到需要識別為止。A P 收到每一個筆劃資料之後，必須以 $(-1, -1)$ 來標示一個筆劃結束（如果 PenDriver 本身已經會增加這個記號，則無須重複增加，每一個筆劃只能有一個結束標記。）；但是，如果是最後一個筆劃，則必須將結束標記中的 Y 座標更改為 0；換言之，最後一個筆劃的結束標記將會（必須）是 $(-1, 0)$ 。

範例：

假設〔一〕這個字有四點（包含結束標記）：

$(10, 10), (20, 10), (30, 10), (-1, 0)$

假設〔二〕這個字的點數為八點（包含結束標記）：

$(10, 10), (20, 10), (30, 10), (-1, -1)$

$(10, 20), (20, 20), (30, 20), (-1, 0)$

6：除了 PPHWRGetRamSize 這個函數之外，其他函數的返回值都是用來指示運行的檢查碼。返回 0 表示一切正常，不為零則表示有錯誤發生。

7：PPHWRRecognize 返回之後，如果其返回值為 0，那麼表示我們可以去檢查 pResult 陣列中的內容。陣列之中，每一個非 0 的 WORD 都是表示一個候選字，越有把握的候選字會放在越前面，0 則表示陣列瀏覽結束。

範例：

$(*pResult)[\] = \{0xA440, 0xA441, 0xA442, 0\}$ ；表示識別結果有三個候選字，內碼分別是 0xA440, 0xA441, 以及 0xA442。

- 8：IN_HK_TYPE 所返回的內碼，大部分還是與繁體中文相同的 Unicode，但是，它也會返回 HKSCS-2001 所定義的增補字符集的 Unicode；SIMPLIFIED_ONLY 則是告訴核心，把所有返回的內碼都轉為簡體字的 Unicode；TRADITIONAL_ONLY 則是把所有的返回內碼都轉為繁體字的 Unicode。如果以上三種擴展型態的選項都沒有設置，系統默認的返回內碼為：寫簡得簡，寫繁得繁，不包含香港造字。
- 9：所有版本的識別型態設定都會包含到漢字二級字庫，以後的設定則可以根據不同的需求由客戶與我們共同討論後加以定義。

聯信科技股份有限公司