

ET-7H16 快速使用手冊

版本 1.0.1, 2019 1 月

產品技術服務與使用資訊

PET-7H16M



Written by Sean

Edited by Anna Huang

文件說明

本份文件將引導使用者快速使用 PET-7H16M 模組。

1. **前置作業**：PET-7H16M 網路設定。
2. **採集及觸發模式簡介**：PET-7H16M 的數據採集及觸發模式簡單說明
3. **簡單使用**：透過 ET-7H16 utility 簡單操作，採集 PET-7H16M 資料，並於 utility 圖表顯示採集資料。
4. **程式開發**：
VC/.NET 程式:依不同的採集及觸發模式簡單說明函式調用流程及片斷程式說明。
5. **LabVIEW 程式**:簡單使用流程說明。

目錄

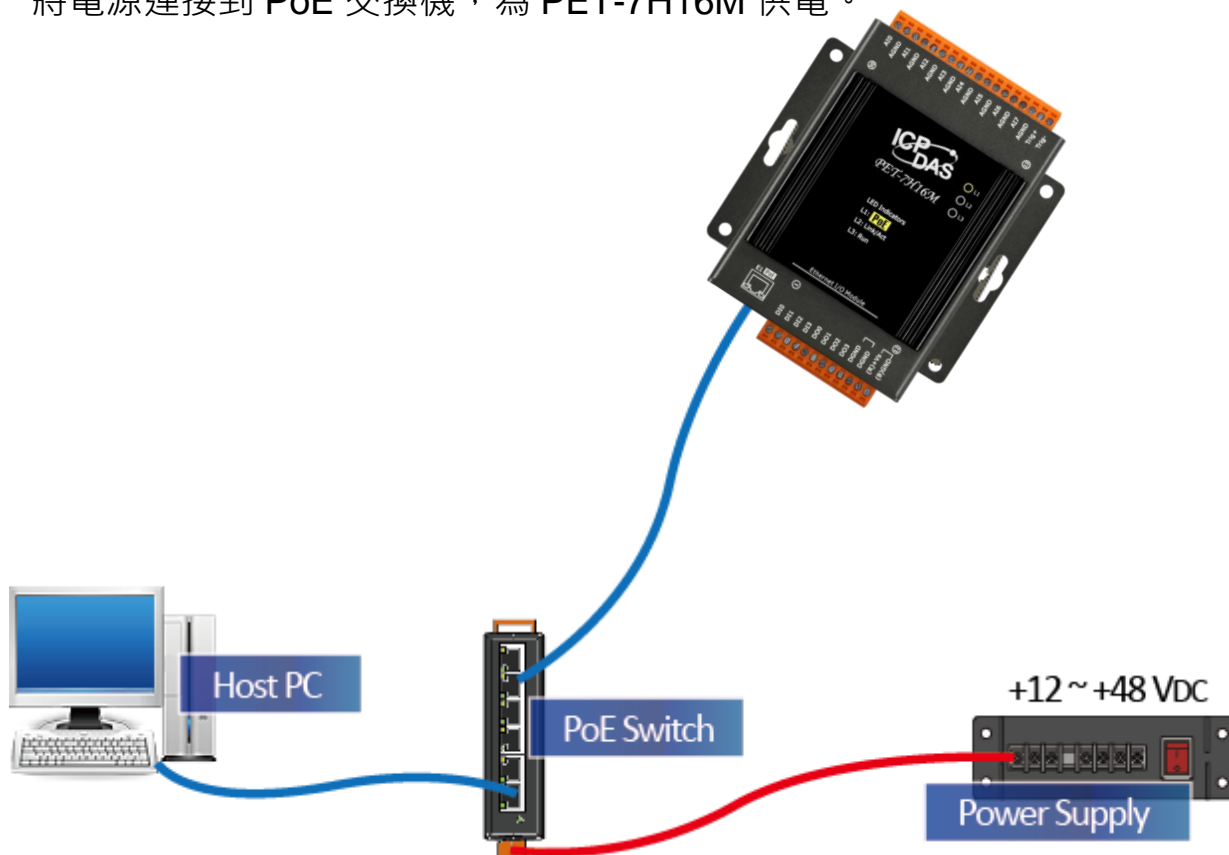
前置作業	4
高速採集及觸發模式簡介	8
簡單使用	10
程式開發	13
LabVIEW 程式	23

前置作業

PET-7H16M 模組有兩種供電方式。一種是通過 PoE 交換機供電，另一種是通過外部電源供應器供電。

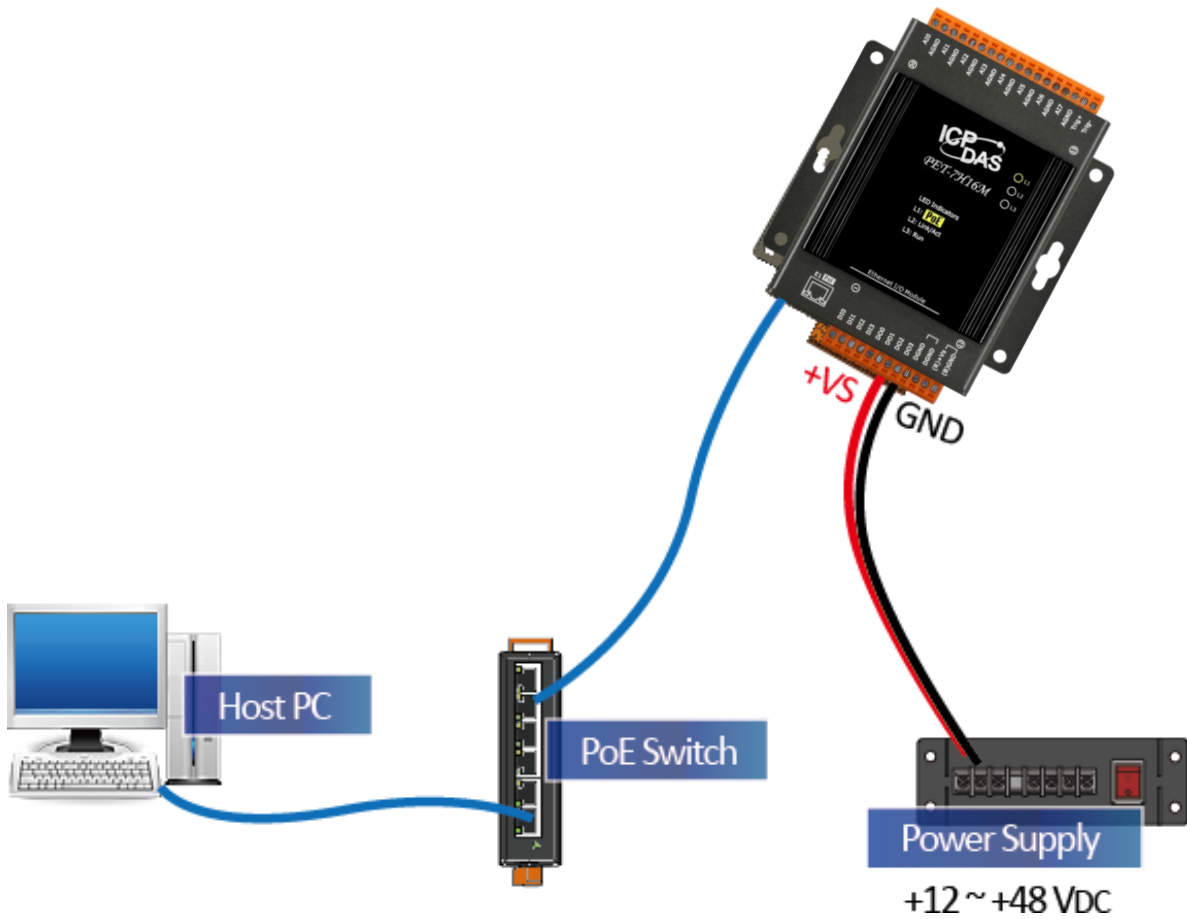
➤ POE 供電

- 將 PC 連接至 PoE 交換機。
- 將電源連接到 PoE 交換機，為 PET-7H16M 供電。



外部電源供電

- 將 PC 及 PET-7H16M 連接到乙太網交換機。
- 將電源連接到交換機和 PET-7H16M。(外部電源應在+12 VDC 至 48 VDC 範圍內)



► 網路連線設定

PET-7H16 的網路出廠預設值如下:

項目	出廠預設值
IP Address	192.168.255.1
Subnet Mask	255.255.0.0
Gateway	192.168.0.1

修改 PET-7H16M 網路組態設定的方式是透過 ET-7H16 utility。

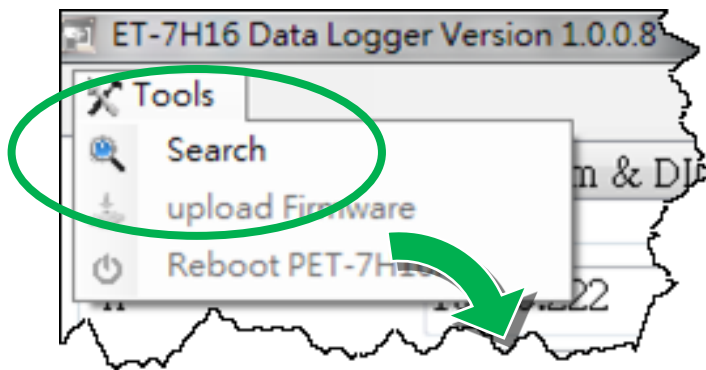
步驟:

1. 從以下路徑取得並安裝 ET-7H16 Utility

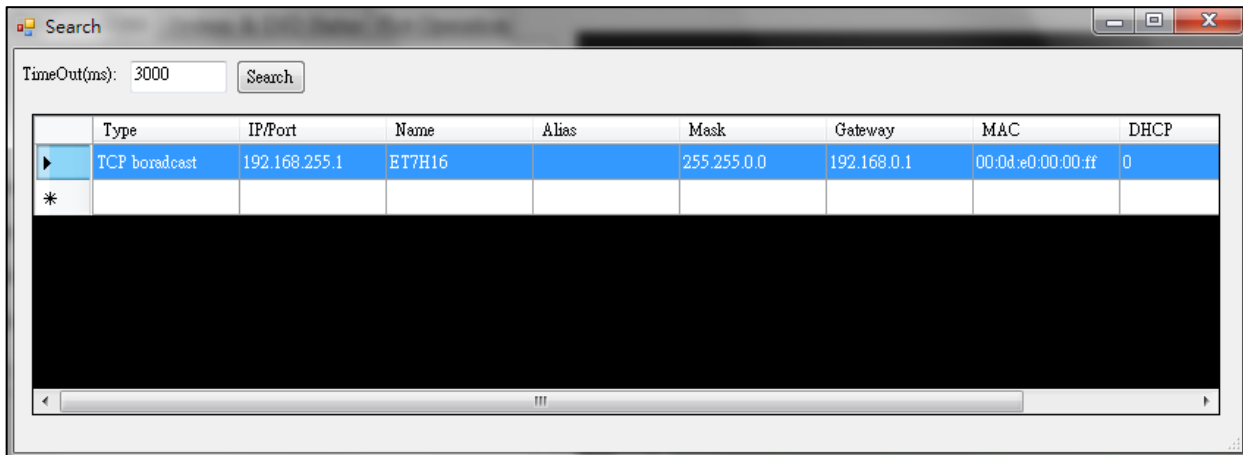
<http://ftp.icpdas.com/pub/cd/6000cd/napdos/et7h16/tools/utility>

2. 執行 ET-7H16 Utility

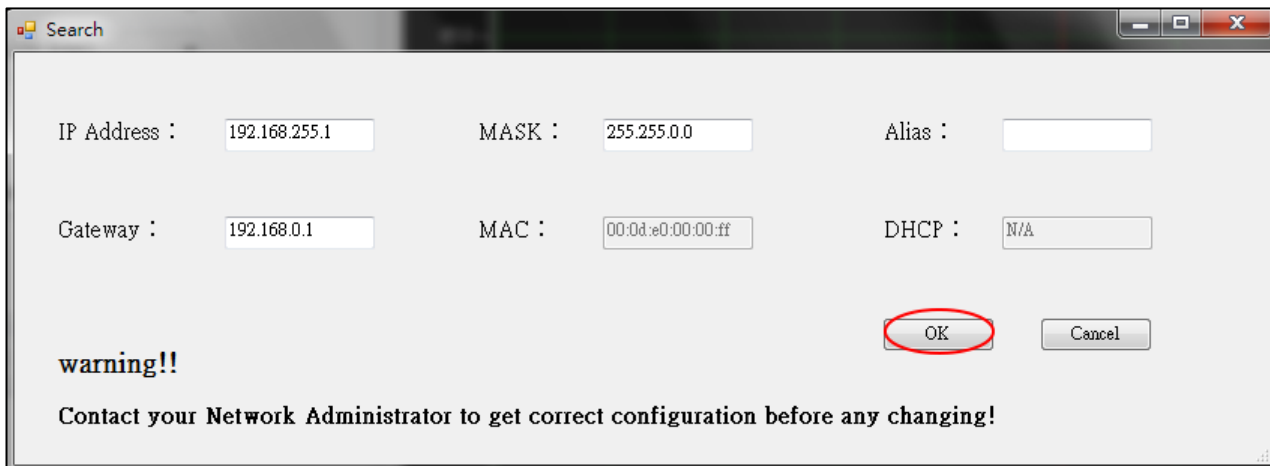
3. 於視窗上面的“Tools” 功能表，點選“Search” 功能。



4. 所有搜尋到的 PET-7H16M 模組都會顯示在這個程式的畫面上



5. 直接點選要設定的 PET-7H16M，會出現以下設定畫面



6. 於 IP address/ Mask/Gateway/Alias 欄位寫入新的設定值，然後按下“OK” 按鈕。

7. 等待 PET-7H16M 重新開機。(重開機，新的設定值即生效)

高速採集及觸發模式簡介

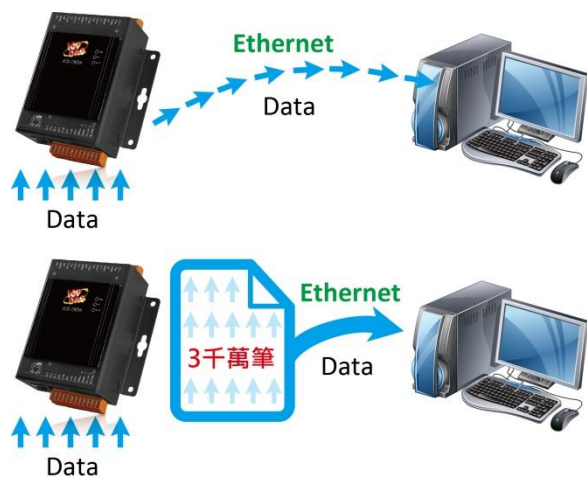
► PET-7H16M 的數據採集及觸發模式說明

二種數據採集方式及多種觸發模式提供高速採集 AI 通道的輸入值。下圖顯示採集及觸發模式其相對應有效的操作頻率範圍。

觸發 \ 採集模式	連續即時傳輸	N 筆數據採集
軟體命令觸發	1~30KHz	1 Hz ~ 200 KHz
外部時脈訊號觸發	1~30KHz	-
預觸發 (Pre-Trigger)	-	1 Hz ~ 200 KHz
後觸發 (Post-Trigger)	-	1 Hz ~ 200 KHz

資料採集及傳輸模式

1. 連續即時傳輸 (最高速 30 kHz/每通道)
2. 採集N筆數據後，再進行傳輸(最高速 200 kHz/每通道)



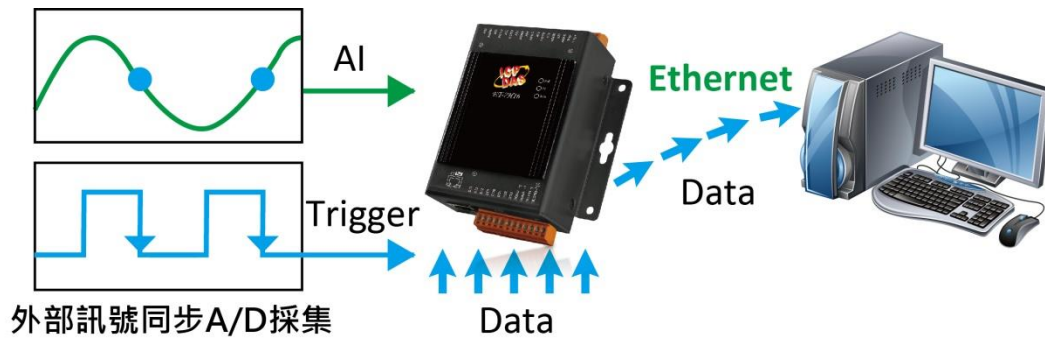
觸發模式

1. 軟體命令觸發

由 PC端下命令設定好 A/D 採集的參數，再下命令觸發後，開始進行連續或是N筆數據的A/D採集。

2. 外部時脈訊號同步A/D採集模式

A/D 採集的速度及數據筆數，皆由外部的電氣訊號控制，每一個電氣訊號的負緣觸發一次的A/D採集。



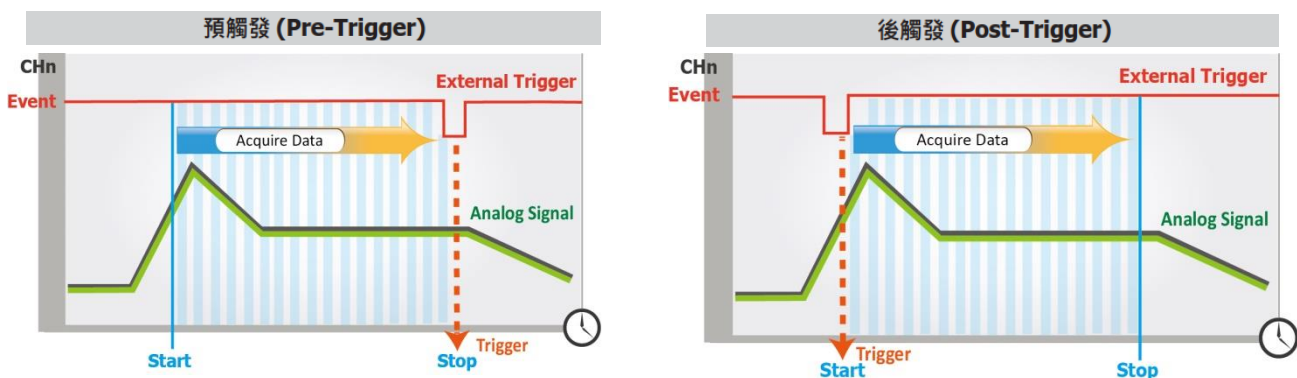
3. 外部數位訊號觸發模式

1. 預觸發，採集N筆數據

在收到觸發訊號之前，A/D 數據就一直被採集，且暫存在PET-7H16M的記憶體之中。等到接收到觸發訊號時，再將已採集的N筆數據傳輸到 PC 端。

2. 後觸發，採集N筆數據

一接收到觸發訊號後，就開始進行N筆數據的 A/D 採集。

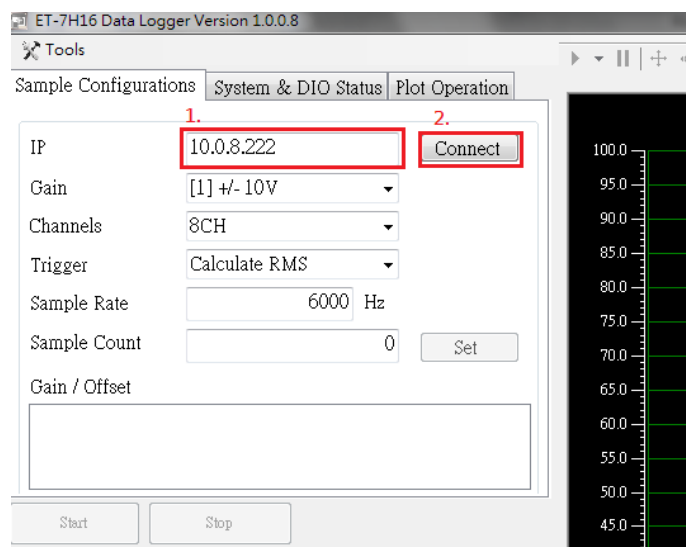


簡單使用

利用 ET-7H16 utility 可以做簡單的 AI 高速採集操作，採集的資料會自動儲存在 PC 磁碟內，資料也會顯示在畫面圖表上。此工具可以做為入門的測試及簡單的診斷使用。

步驟:

- 1 執行 ET-7H16 utility。於以下畫面輸入 ET-7H16 的 IP address，點選”Connect”連線。



2 設定 AI 高速採集的參數值

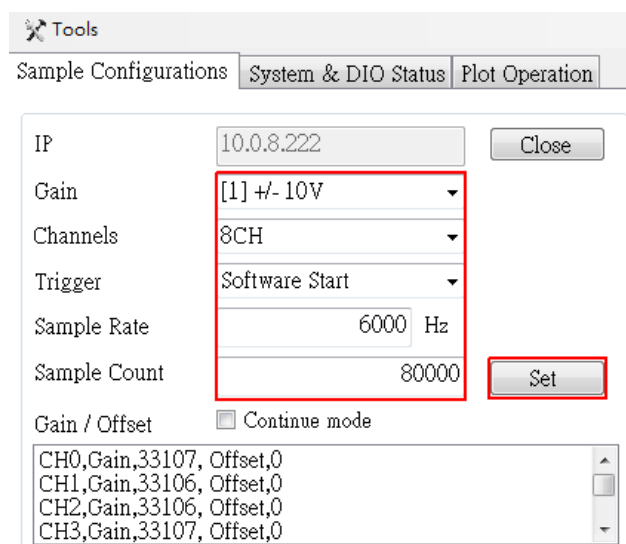
- 2.1 選擇 Gain，是 +/-5V 或 +/-10V。
- 2.2 選擇掃描的 AI 通道數。
- 2.3 選擇觸發模式(Trigger)。
 - 0: 軟體命令觸發
 - 1: 外部時脈訊號觸發
 - 2: 預觸發 (Pre-Trigger)
 - 3: 後觸發 (Post-Trigger)
- 2.4 輸入 Sample Rate。(1~200KHz)

2.5 設置採集資料的筆數

設定值為 0，為連續即時傳輸模式，會一直儲存資料。需按”Stop”鈕才會停止採集。

設定值大 0，當採集的筆數達到設定的筆數，採集會自動停止。

2.6 上面步驟完成後，按下”Set”鈕，將設定值寫至 PET-7H16M 模組。



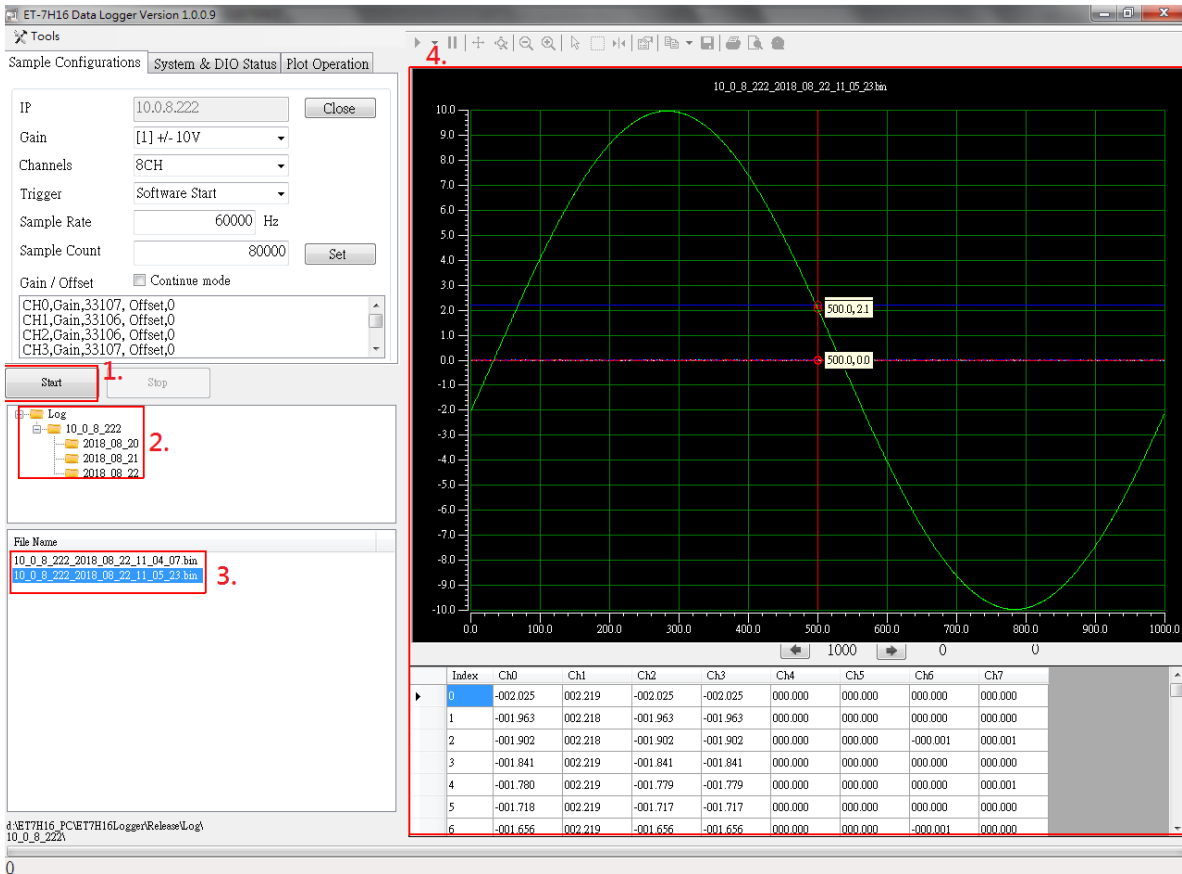
3 開始 AI 高速資料採集

3.1 按下 Start 鈕。如有設置筆數，採集筆數達到這設定值時，會自動停止。若筆數為 0，需手動點選 Stop 停止。

3.2 選擇要顯示的資料所在的資料夾，資料夾分層為 1.IP 2.日期。

3.3 選取要顯示哪一筆資料檔案。

3.4 以下畫面的上面是資料所繪成的波形圖，下面是每個 Channel 的數據，每一頁顯示 1000 筆資料。



程式開發

► 開發前置準備

取得程式開發使用的函式庫

ET-7H16 DAQ SDK 包含的檔案

檔案	敘述
HSDAQ.dll HSDAQ.lib HSDAQ.h	應用於 VC 程式開發
HSDAQNet.dll	應用於 .Net 程式開發

由網站取得最新版 SDK 函式庫的相關檔案,路徑如下

- <http://ftp.icpdas.com/pub/cd/6000cd/napdos/et7h16/tools/dll/sdk/>

準備開發工具

於 PC 上安裝 Microsoft Visual Studio 2008 或以上版本

如何於 Visual Studio 開發環境中使用 SDK 開發程式

參考以下路徑內的 PET-7H16M 標準 API 使用者手冊

http://ftp.icpdas.com/pub/cd/6000cd/napdos/et7h16/document/sdk_document/

於 1.4.1 節，詳細了解於 Visual Studio 使用 SDK 開發 C# 程式

於 1.4.2 節，詳細了解於 Visual Studio 使用 SDK 開發 VB.NET 程式

於 1.4.3 節，詳細了解於 Visual Studio 使用 SDK 開發 VC 程式

Demo 程式下載路徑

- VC :
<http://ftp.icpdas.com/pub/cd/6000cd/napdos/et7h16/tools/DLL/Demo/VC/>
- C# :
<http://ftp.icpdas.com/pub/cd/6000cd/napdos/et7h16/tools/DLL/Demo/c%23/>

- VB.NET
<http://ftp.icpdas.com/pub/cd/6000cd/napdos/et7h16/tools/DLL/Demo/VB.net>

► 範例說明

使用程式開發高速 AI 採集功能依據不同的觸發模式及資料採集/傳輸方式提供各種範例程式。

高速 AI 採集相關 API 函式說明及函式調用流程，請參考以下路徑內的 PET-7H16M 標準 API 使用者手冊，第 2.4 節高速 IO API。

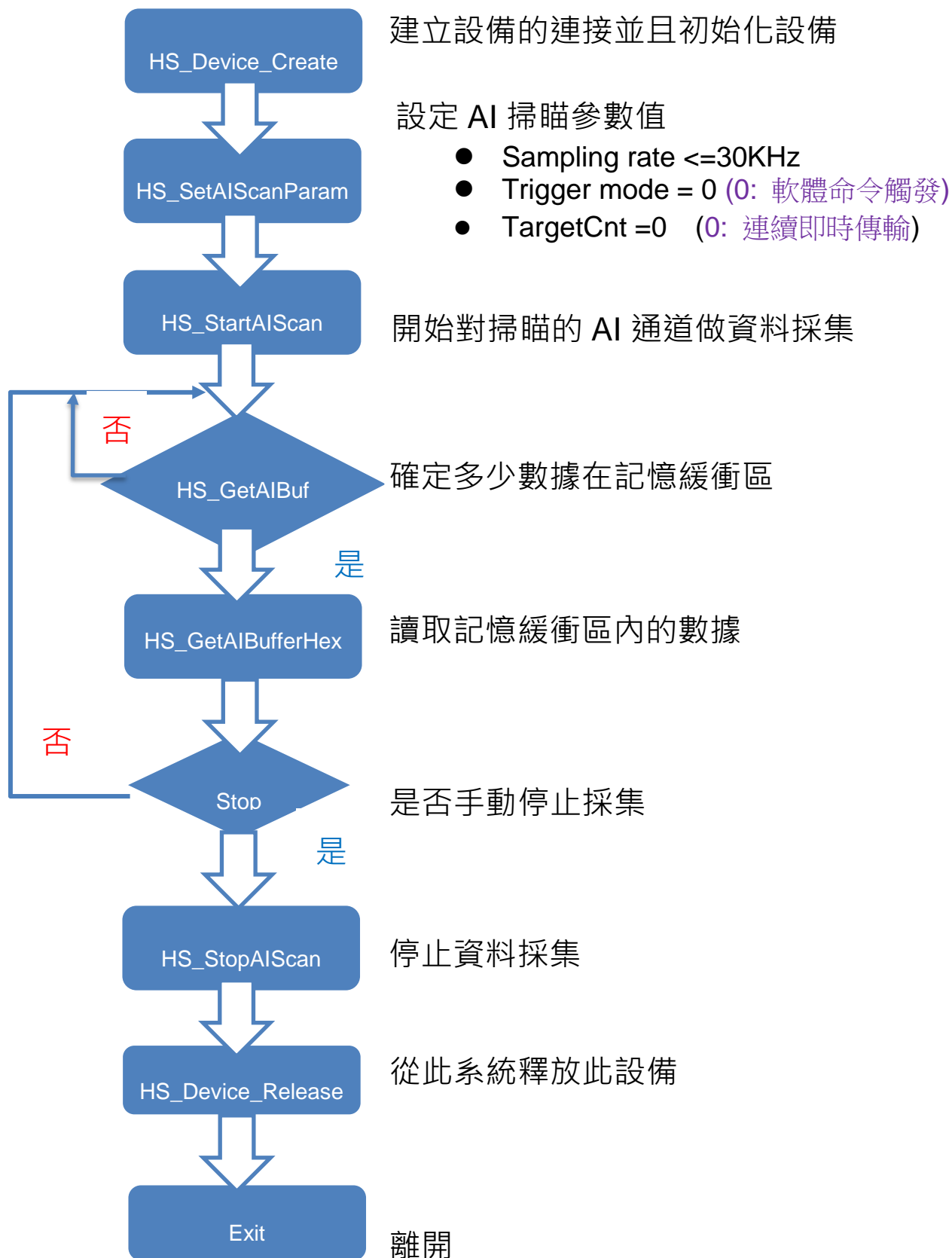
http://ftp.icpdas.com/pub/cd/6000cd/napdos/et7h16/document/sdk_document/

以下以二種模式高速採集範例及一種資料記錄範例。

1. 連續即時傳輸及軟體命令觸發模式的高速採集範例
2. N 筆數據採集及軟體命令觸發模式的高速採集範例
3. N 筆數據採集及軟體命令觸發模式的資料記錄範例

1.連續即時傳輸及軟體命令觸發模式的高速採集範例

程式 API 函數調用流程圖



[C 程式片斷程式碼]

```
HANDLE hHS;
WORD BufferStatus=0;
float fdataBuffer[10000];
unsigned long ulleng=0;
hHS = HS_Device_Create("192.168.1.1"); //建立設備的連接並且初始化設備

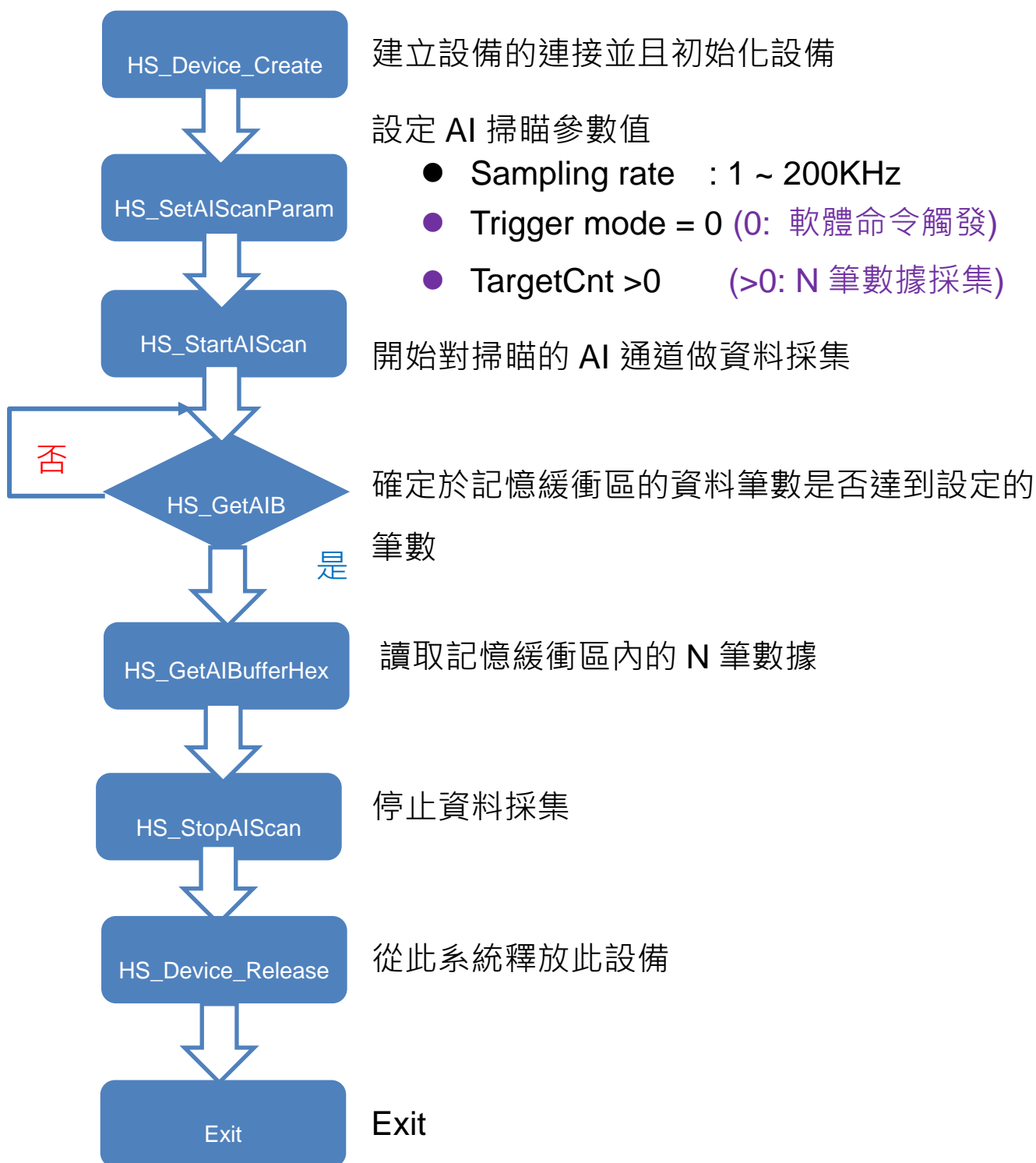
HS_SetAIScanParam(hHS, 8, 0, 0, 20000, 0, 0,0);
/*設定 AI 掃描參數值
● Sampling rate =20KHz
● Trigger mode = 0 (0: 軟體命令觸發)
● TargetCnt =0 (0: 連續即時傳輸)
*/
HS_StartAIScan (hHS); //開始對掃描的 AI 通道做資料採集

ret=HS_GetAIBufferStatus(hHS,&BufferStatus,&ulleng); //確定多少數據在記憶緩衝區

if(ret==false){
    printf("Error code 0x%x\r\n",HS_GetLastError());
}
else
{
    if(ulleng)
    {
        readsize=HS_GetAIBuffer(hHS,fdataBuffer, ulleng); //讀取記憶緩衝區內的數據
        ...
    }
}
HS_StopAIScan (hHS); //停止資料採集
HS_Device_Release (hHS); //從此系統釋放此設備
```

2.N 筆數據採集及軟體命令觸發模式的高速採集範例

API 函數調用流程圖



[C 程式片斷程式碼]

```
HANDLE hHS;
WORD BufferStatus=0;
float fdataBuffer[10000];
unsigned long ulleng=0;
hHS = HS_Device_Create("192.168.1.1"); //建立設備的連接並且初始化設備

HS_SetAIScanParam(hHS, 8, 0, 0, 100000, 2000000, 0,0);
/*設定 AI 掃描參數值
● Sampling rate =100KHz
● Trigger mode = 0 (0: 軟體命令觸發)
● TargetCnt =2000000 (2000000 筆數據採集)
*/
HS_StartAIScan (hHS); //開始對掃描的 AI 通道做資料採集

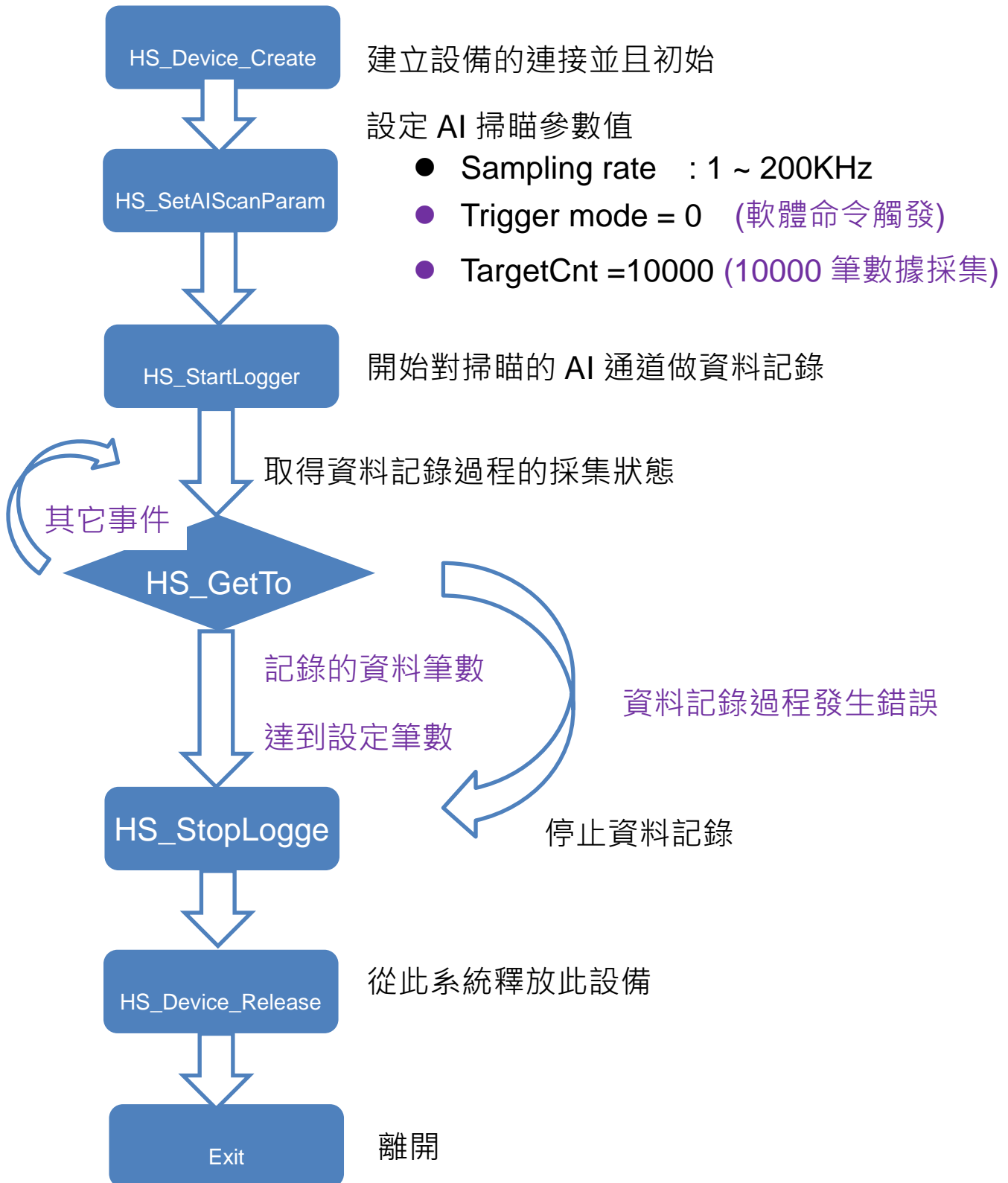
ret=HS_GetAIBufferStatus(hHS,&BufferStatus,&ulleng); //確定多少數據在記憶緩衝區

if(ret==false){
    printf("Error code 0x%x\r\n",HS_GetLastError());
}
else
{
    if(BufferStatus>2) //AI buffer overflow
    {
        /* 2:  AD_BUF_OVERFLOW
           4:  AD_SCAN_STOP
           8:  AD_DATA_SAMPLING_TIMEOUT
        */
        break;
    }
    if(ulleng==targetCnt) //記憶緩衝區的資料筆數達到設定的筆數
    {
        unsigned long size=targetCnt;
        readsize=HS_GetAIBufferHex(hHS,(WORD *)dataBuffer,size);
        //讀取記憶緩衝區內的 N 筆數據
    }
}
HS_StopAIScan (hHS); //停止資料採集
HS_Device_Release (hHS); //從此系統釋放此設備
```

3. N 筆數據採集及軟體命令觸發模式的資料記錄範例

ET-7H16 模組具有資料記錄功能。由模組上採集的資料可以傳輸至 Host PC 並保存成資料記錄檔(.bin、.txt 和 etc)

API 函數調用流程圖



[C 程式片斷程式碼]

```
HANDLE hHS;
hHS = HS_Device_Create("192.168.1.1"); //建立設備的連接並且初始化設備
HS_SetAIScanParam(hHS, 8, 0, 0, 20000, 10000, 0,0);
/*設定 AI 掃描參數值
● Sampling rate =20KHz
● Trigger mode = 0 (0: 軟體命令觸發)
● TargetCnt =10000 (10000: 10000 筆資料)
*/
HS_StartLogger(hHS,NULL,2,0); //開始對掃描的 AI 通道做資料記錄

while(;;)
{
    ret=HS_GetTotalSamplingStatus(hHS,&ulleng,&triggerStatus);
    //於資料記錄過程取得的採集狀態
    if(ret==false)
    {
        printf("Error code 0x%x\r\n",HS_GetLastError());
    }
    else
    {
        if(CHECK_BIT(triggerStatus,7)==true && targetCnt>0) //記錄的資料筆數達到設定筆數
        {
            if(ulleng>=targetCnt)
            {
                ...
                break; //跳出迴圈
            }
        }
        else
        {
            if(CHECK_BIT(triggerStatus,4) && HECK_BIT(triggerStatus,5) ) //資料記錄過程發生錯誤
            {
                break; //跳出迴圈
            }
            else
            {
                .... //持續在迴圈內
            }
        }
    }
}
HS_StopLogger(hHS); //停止資料記錄
```

```
HS_Device_Release(hHS); //從此系統釋放此設備

/*利用 Data logger API 函式讀取資料記錄檔資料*/
HANDLE hlf;
TCHAR tcgetfulfilePath[MAX_PATH]={0};
int ind=HS_GetAllLogFiles(NULL,1);

if(ind>0)
{
for(int i=0;i<ind;i++)
{
    hlf=HS_LogFile_Open_byIndex(i,tcgetfulfilePath);
    DWORD samplecount;
    char startdate[32],starttime[32];
    HS_GetLogFile_AIScanSampleInfo(hlf,&samplecount,startdate,starttime);
    float *fdatbuff=( float *)malloc(sizeof(float)*samplecount);
    HS_GetLogFile_AIData(hlf, 0, samplecount, fdatbuff);
    HS_LogFile_Close(hlf);
}
}
}
```

LabVIEW 程式

HSDAQ LabVIEW 工具組用於監控 ICP DAS 高速資料擷取模組，具有界面簡潔，容易開發等優點。工具組中包含 HSDAQ.lib 資料庫-驅動所需的子 VI，與多個範例程式。

簡單的範例操作說明與子 VI 介紹。請參考

<http://ftp.icpdas.com/pub/cd/6000cd/napdos/et7h16/document/application/labview/>

LabVIEW 範例及相關元件，可由以下路徑取得

<http://ftp.icpdas.com/pub/cd/6000cd/napdos/et7h16/tools/labview/>