

**AN001: 從 IWF 檔讀出樣本點**

本 Application Note 說明 IWF 檔案部份格式與欄位定義，使用者可據此讀出檔案記錄的樣本點進行自訂分析。

**IWF File Format**

下表列出從 LA-08 資料檔計算出原始取樣點所需的資訊。

offset (hex)	size (dec)	type	name	description
0	16	CHAR[16]	signature, starting	"Ideofy LA-08 000", note that there are two space characters (ascii 0x20).
10	4	UINT32	iwf version	iwf file format index, currently 0x00010000
14	4	UINT32	sw_version	version number of the LA08.EXE which created the file. 0x01090201, means v1.9.2.1
144	4	UINT32	sample_rate	sample rate in KHz
14C	4	UINT32	n_channel	number of channels, valid values are 8, 4, and 2.
154	4	UINT32	n_sample	number of sample points
158	4	UINT32	trigger_pos	trigger position (%), valid value 0 to 100
160	8	UINT8[8]	trigger_set[8]	trigger settings, one byte per channel, possible values are 0 to 5, indicating <b>none</b> , <b>high</b> , <b>low</b> , <b>rising edge</b> , <b>falling edge</b> , and <b>either edge</b> , respectively. trigger_set[0] is for channel 1.
168	256	CHAR[8][32]	ch_text[8]	text for channel 1 to 8
4C8	4	UINT8[4]	signature, ending	55, AA, 55, AA (hex)
4CC			sample_data	sample data starts here

\* UINT8 代表寬度是 8 bits 的 unsigned int, UINT32 寬度是 32 bits, CHAR 寬度是 8-bit.

\* 佔多個 byte 的資料格式，如 UINT32，各個 byte 是以 little endian 方式存放，也就是 LSB (least significant byte) 最先存放。舉例來說 0x01020304 這個 4-byte 的 UINT32 值，是以 0x04, 0x03, 0x02, 0x01 之順序存放在檔案中。

\* 資料格式後加上[n]代表由 n 個資料組成的陣列。多維陣列以 row-major 順序[2]存放。

## Run-length Encoding

爲了減少空間浪費 iwf 檔採用 Run-Length Encoding (RLE)[1]來壓縮資料. 原始資料會以 byte 爲單位將相同且連續的值分成一個個 run, 每個 run 再以兩個 bytes 分別代表其值與長度記錄. 舉例來說, 55,55,55,55,55,aa,aa,aa,aa,aa,aa,77,77,77,77,77,77,... 可用 RLE 壓縮成 55,3,aa,6,77,6,..

本節最後會附上一段 iwf 檔解碼用的 sample code. 在執行這段程式之前, inf, acq\_data, acq\_size 必須先初始化. inf 是輸入檔, 且其內部指標必須指到 RLE 資料開始處(offset 0x44C). acq\_size 是 sample memory 所佔的 byte 數, 可用 n\_sample 與 n\_channel 求出, 公式是  $acq\_size = n\_sample / (8/n\_channel)$ . acq\_data 是指到 sample memory 的指標, 此塊 memory 大小至少須爲 acq\_size 且必須先配置完成.

```
// FILE * inf;    // input file
// int acq_size  // number of data bytes
// unsigned char * acq_data; // pointer to sample memory

int i, b, c, e, bc;

i = 0; bc = 0;
while(!feof(inf)) {
    b = fgetc(inf);
    c = fgetc(inf);
    i = 0; e = bc+c+1;
    if(b==EOF||c==EOF) break;
    if(feof(inf)) break;
    acq_data[bc++] = b;

    while(bc<e) {
        acq_data[bc++] = b;
        if(bc>=acq_size) break;
    }
}
```

## Mapping a sample point to a sample memory location

在 8 通道時，每組取樣點有 8 bits，通道 8(Data 8)是 msb，通道 1(Data1)是 lsb。在 sample memory 中每個 byte 即為一取樣點。

在 4 通道時，每組取樣點有 4 bits，通道 4(Data 4)是 msb，通道 1(Data1)是 lsb。在 sample memory 中每個 byte 存放兩個取樣點。bit7~4 是前面的取樣點，bit3-0 是後面。

在 2 通道時，每組取樣點有 2 bits，通道 2(Data 2)是 msb，通道 1(Data1)是 lsb。在 sample memory 中每個 byte 存放四個取樣點。bit7,6 是第取樣點，bit5,4 第二個，bit3,2 第三，bit1,0 是最後一個取樣點。

以 2 通道為例，第 12345 個取樣點，存放在 sample memory 中第  $12345/4=3086$  個 byte 的 bit5 與 bit4 (因  $12345\%4=1$ )。Bit 5 是 Data 1 的值，bit 4 是 Data 0 的值。

## Reference

[1] Wikipedia, *Run-length encoding*, [http://en.wikipedia.org/wiki/Run-length\\_encoding](http://en.wikipedia.org/wiki/Run-length_encoding)

[2] Wikipedia, *Row-major order*, [http://en.wikipedia.org/wiki/Row-major\\_order](http://en.wikipedia.org/wiki/Row-major_order)