

SECS Emulator v1.5 使用手冊

工研院機械所 設備控制部

副研究員 廖勝泰

880731@itri.org.tw

2001/09/20

1. SECS Emulator 簡介	3
2. 快速上手	3
2.1 SECS-I 通訊功能測試	3
2.2 HSMS 通訊功能測試	7
3. 選單說明	8
3.1 檔案	8
3.2 編輯	8
3.3 檢視	9
3.4 特殊測試	10
4. SML 檔案格式	12
4.1 SECS-II 訊息格式	12
4.2 自動化測試命令格式	13

1. SECS Emulator 簡介

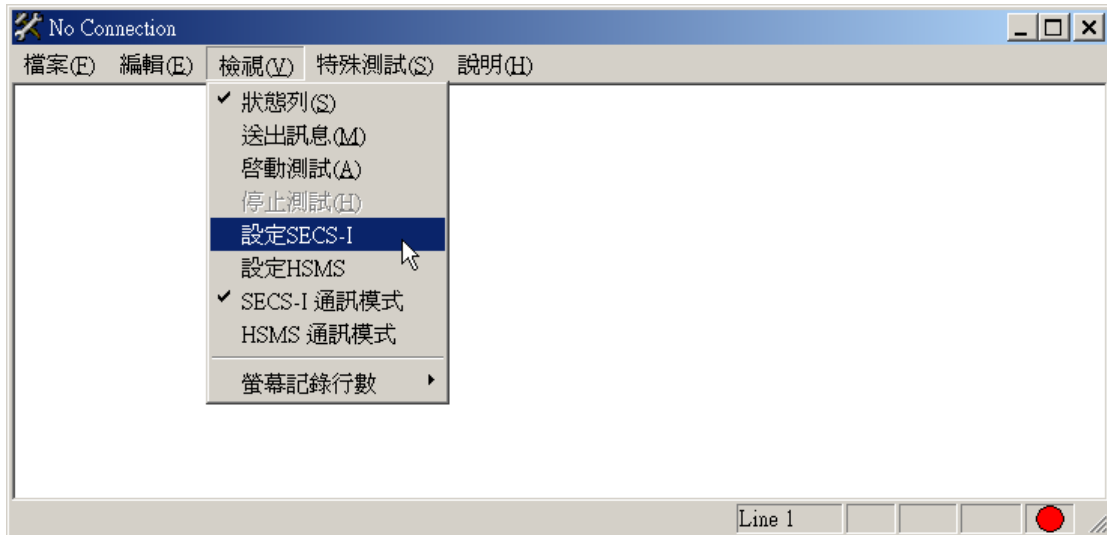
SECS Emulator 主要功能是用來測試半導體設備的通訊功能。它的特點有：

- 支援 Windows 95/98/2000/NT 作業平台
- 支援 SECS-I/SECS-II/HSMS-SS 通訊協定
- screen logging 及 file logging 能力
- 支援 SML 格式
- 強大易用的自動化測試功能
- 免安裝，即拷即用

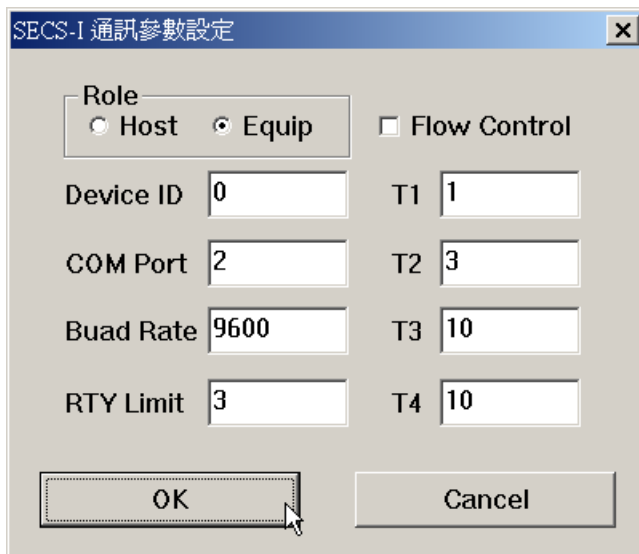
2. 快速上手

2.1 SECS-I 通訊功能測試

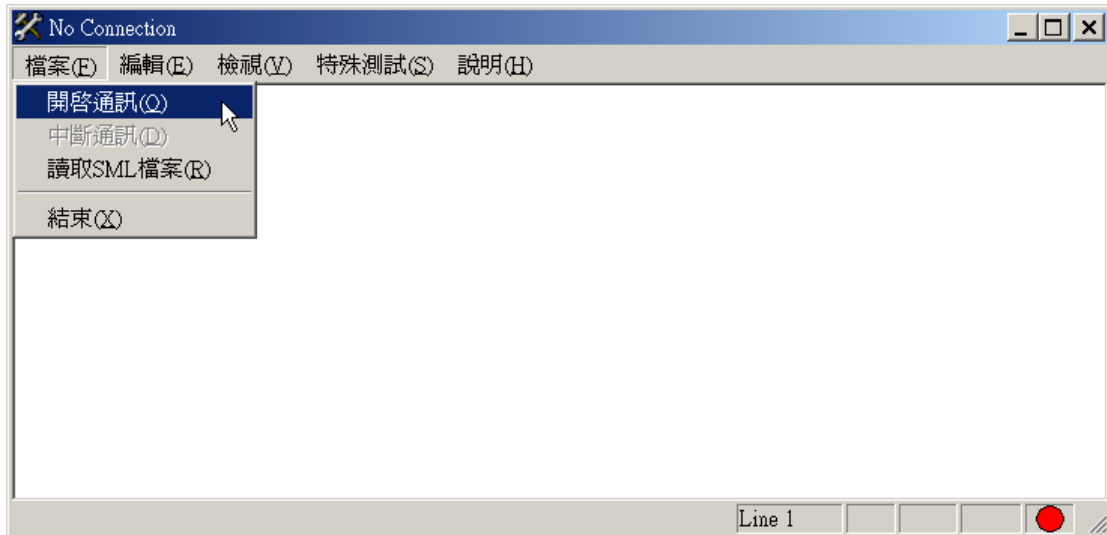
先將 RS232 連接線接好。執行 SECS Emulator 之後，選擇「檢視」：



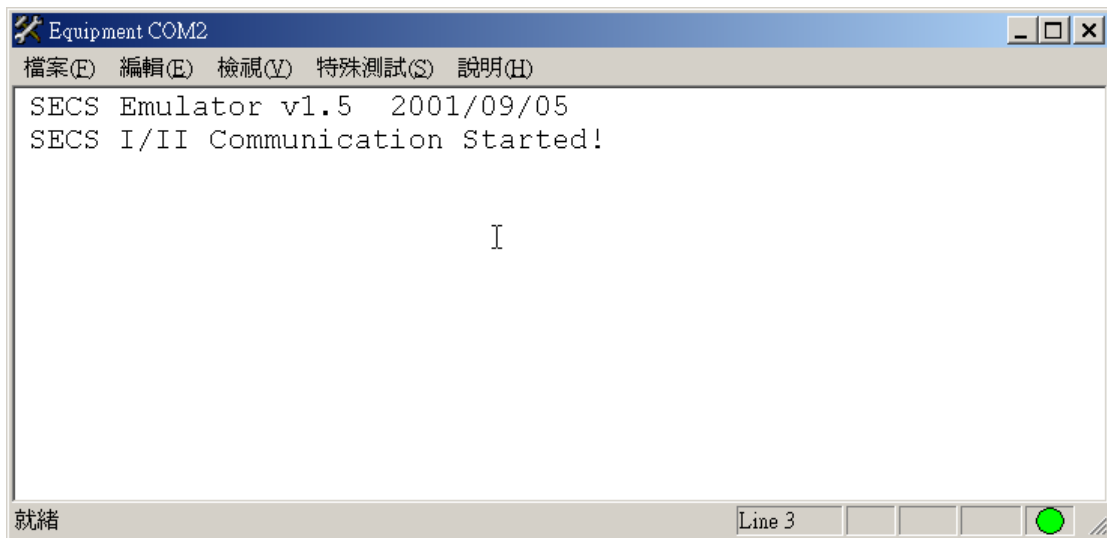
按下「設定 SECS-I」，在對話盒中輸入相關的通訊參數：



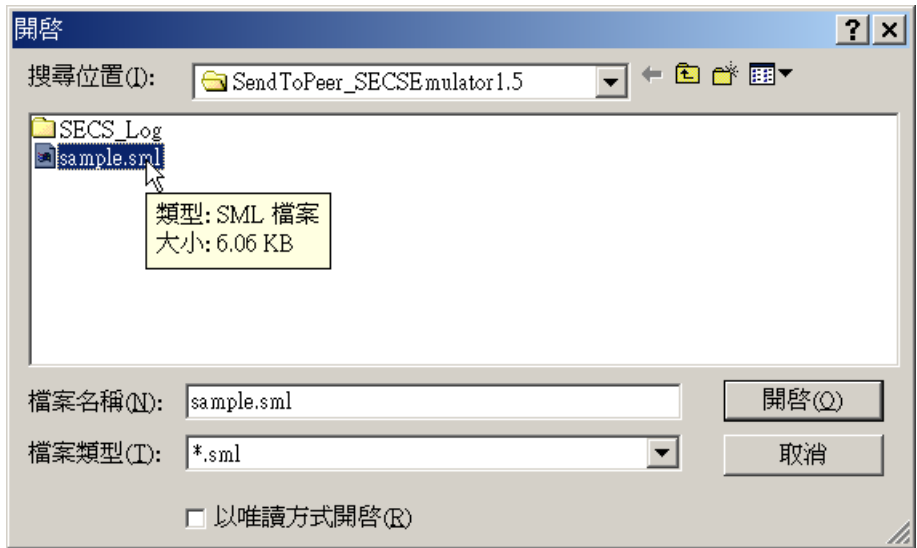
設定完畢之後，按下「OK」按鈕。接下來選擇「檔案」中的「開啓通訊」：



成功開啓通訊之後，SECS Emulator 右下角的紅燈會轉變為綠燈：



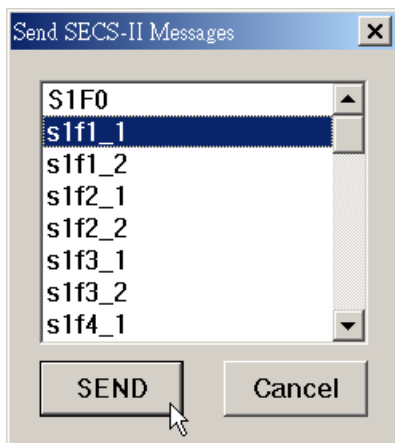
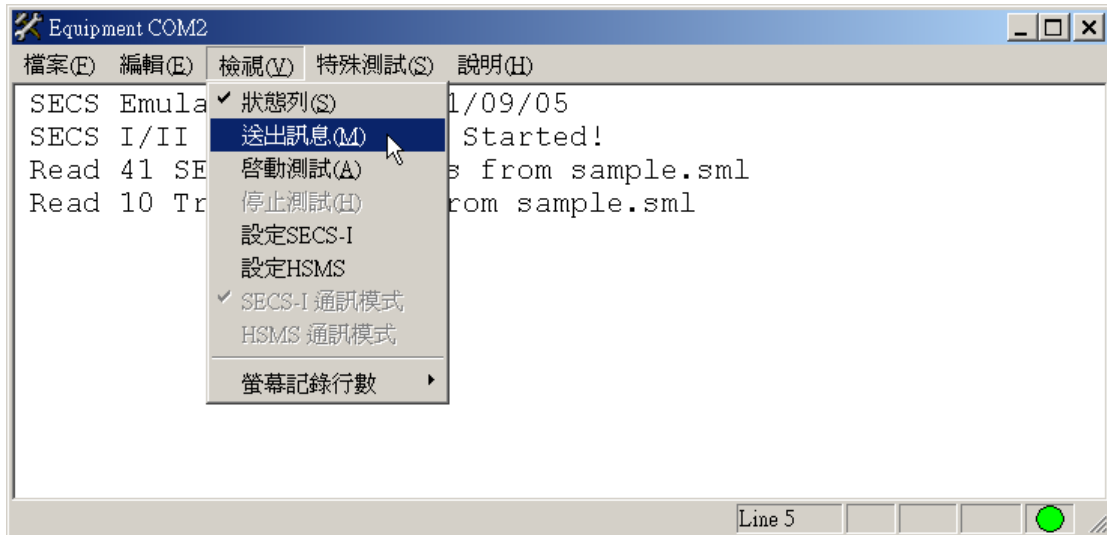
接下來選擇「檔案」中的「讀取 SML 檔案」，將內含 SECS-II 訊息定義的 SML 檔案讀取進來。使用者可以先讀取 SECS Emulator 目錄下所附的範例檔 `sample.sml`：



SECS Emulator 讀取完畢之後，會在螢幕上顯示所讀取的 SML 檔，其中所含的 SECS-II 訊息定義的數量和自動測試命令的數量：



接下來使用者就可以選擇「檢視」中的「送出訊息」，主動送出在 SML 檔中所定義的 SECS-II 訊息：



另外使用者也可以按下「檢視」中的「啓動測試」，讓 SECS Emulator 按照在 SML 檔案中的 every 命令的設定，週期性地送出訊息。

所有通訊過程中的資料交換，都會自動記錄在 SECS Emulator 執行檔所在位置的子目錄「SECS_Log」之下。

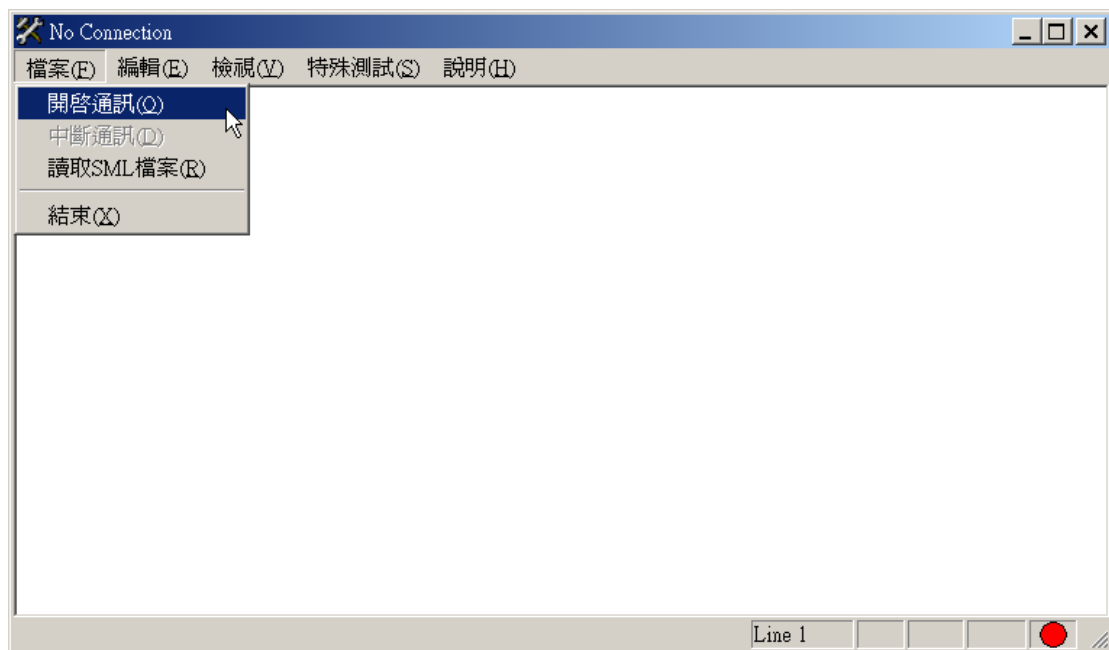
2.2 HSMS 通訊功能測試

除了一開始必須先選擇「檢視」的「設定 HSMS」來設定相關通訊

參數，並且選擇「檢視」中的「HSMS 通訊模式」之外，其餘的使用方法和功能和 2.1 節中所描述的是完全一樣的。

3.選單說明

3.1 檔案



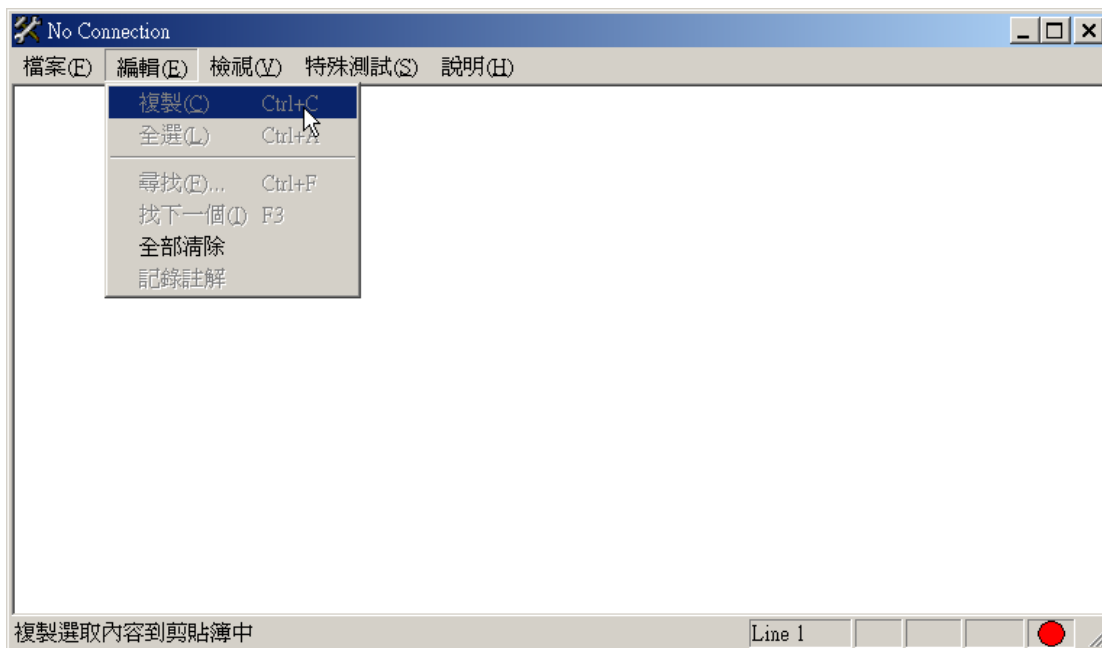
開啓通訊：此選項用來開啓通訊管道。

中斷通訊：此選項用來關閉目前的通訊管道。

讀取 SML 檔案：讀取選定的 SML 檔案。

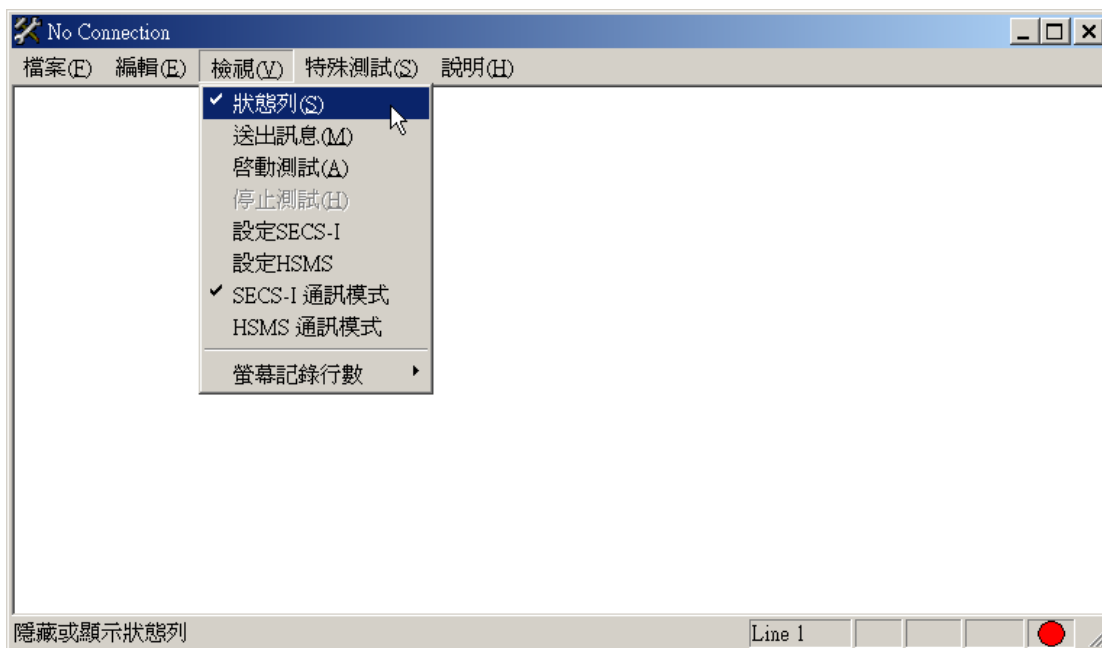
結束：結束關閉 SECS Emulator 程式。

3.2 編輯



提供標準的文字編輯功能。

3.3 檢視



狀態列：用來隱藏或顯示狀態列。

送出訊息：在建立通訊之後，可以主動送出在 SML 檔案中所定義的 SECS-II 訊息。

啓動測試：用來啓動在 SML 檔案中使用 every 命令所定義，需週期性送出的訊息。

停止測試：用來停止執行在 SML 檔案中所定義的 every 命令。

設定 SECS-I：設定 SECS-I 的相關通訊參數。

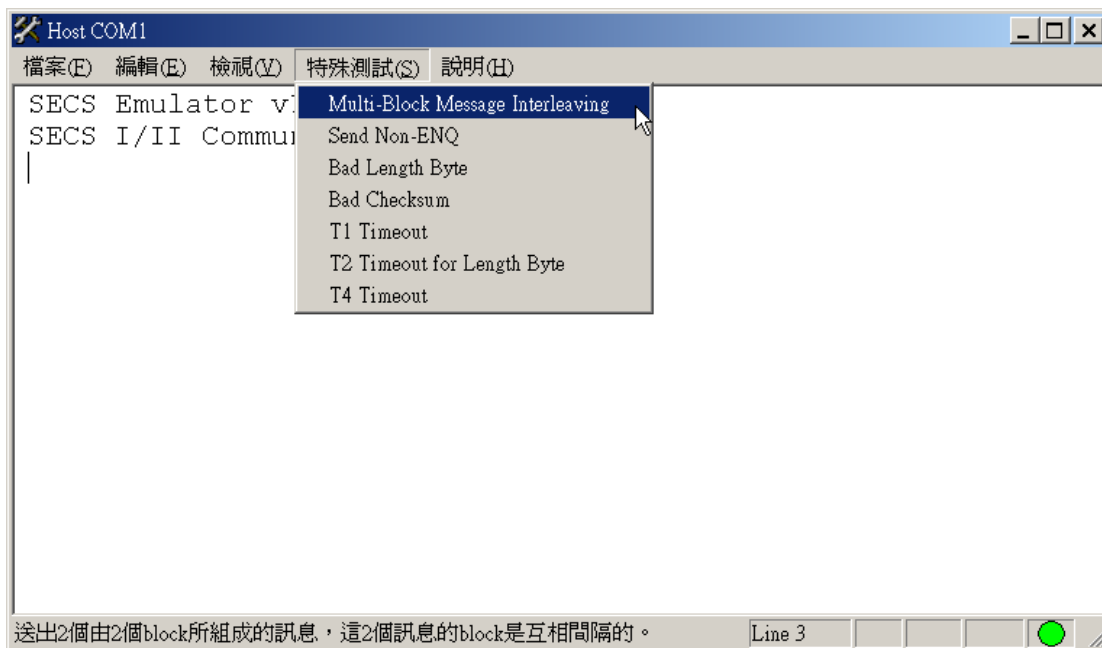
設定 HSMS：設定 HSMS 的相關通訊參數。

SECS-I 通訊模式：選擇 SECS-I 作為底層通訊方式。

HSMS 通訊模式：選擇 HSMS 作為底層通訊方式。

螢幕記錄行數：設定螢幕記錄行數的最大值，超過的話程式就會自動上捲（wrapping）。

3.4 特殊測試



必須在使用 SECS-I 作為底層通訊模式且已建立通訊之後，所有的選項才會被致能（enabled）。這個選單下的各個選項，主要是用來製造出特定的錯誤狀況，測試連線端的容錯能力。

Multi-Block Message Interleaving：送出 2 個具備 2 個 block 的訊息，測試連線端是否具備處理 Interleaving Multi-Block Message 的能力。

Send Non-ENQ：送出一個非 ENQ 字元。

Bad Length Byte：送出一個 Length Byte 不合法的訊息。

Bad Checksum：送出一個 Checksum 不正確的訊息。

T1 Timeout：測試連線端處理 T1 Timeout 的能力。

T2 Timeout for Length Byte：測試連線端處理 T2 Timeout 的能力。

T4 Timeout：測試連線端處理 T4 Timeout 的能力。

4.SML 檔案格式

在 SML 檔案中，使用者可以撰寫 SECS-II 訊息的格式，定義完畢所有的 SECS-II 訊息之後，可以透過 SECS Emulator 的自動測試命令來撰寫自動化測試的功能。

4.1 SECS-II 訊息格式

請參考 SECS Emulator 所附的範例檔 `sample.sml` 中的寫法。完整的訊息定義語法如下：

```
[name:] SxFy [W]  
items .
```

每個訊息可以由使用者指定一個名字 (*name*)，如果沒有指定的話，SECS Emulator 就以 *SxFy* 作為訊息的名字。訊息最後要以一個句點作為結束符號。至於訊息內含的 *item* 格式如下：

```
<type [length] [value]>
```

每個 *item* 的型別由 *type* 指定：

A	ASCII
J	JIS-8
B	Binary

I1	1-byte signed integer
I2	2-byte signed integer
I4	4-byte signed integer
U1	1-byte unsigned integer
U2	2-byte unsigned integer
U4	4-byte unsigned integer
F4	4-byte floating point number
F8	8-byte floating point number
Boolean	boolean
L	List

4.2 自動化測試命令格式

自動化測試的命令共有「if」和「every」兩種：前者主要是用來指定當收到某種訊息時，要回應哪一個對應的訊息；後者則是用來指定 SECS Emulator 週期性地送出使用者指定的訊息。

以下面的例子來看：

```
if(s1f1 (9)==<F4 0.5567>) s1f2_1.
```

上面的這個 if 命令句是用來指定：當收到 s1f1 且第 9 個 item 的型別為 F4、值為 0.5567 時，就回覆 s1f2_1。

```
every 5 send s1f1_1.
```

上面的這個 every 命令句是用來指定：每隔 5 秒，就主動送出 s1f1_1。當使用者在 SECS Emulator 中開啓通訊成功之後，只要再按下「檢視」中的「開始測試」，SECS Emulator 就會依照 SML 檔案中的 every

命令週期性送出訊息。