

國立交通大學

機械工程學系碩士班

碩士論文

半導體與光電製程設備通訊模組之研究與開發

The Development of SECS Communication Module for  
Cluster Tools and Photoelectric Process Equipments



研究生：詹孟璋

指導教授：李安謙 博士

中華民國九十三年七月

# 半導體與光電製程設備通訊模組之研究與開發

## The Development of SECS Communication Module for Cluster Tools and Photoelectric Process Equipments

研究生：詹孟璋

Student: Meng-Chang CHAN

指導教授：李安謙 博士

Advisor: Dr. An-Chen LEE

國立交通大學

機械工程學系碩士班



Submitted to Department of Mechanical Engineering

College of Engineering

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master of Science

in

Mechanical Engineering

July 2004

Hsinchu, Taiwan, Republic of China

中華民國 九十三年 七月

# 半導體與光電製程設備通訊模組之研究與開發

學生：詹孟璋

指導教授：李安謙 博士

國立交通大學機械工程研究所 碩士班

## 摘 要



本論文主要目的在利用軟體工程上物件導向分析與設計(Object-Oriented Analysis/Design, OOA/D)的方法，設計一套滿足半導體工業標準(SEMI Standard)的獨立介面通訊模組控制器(Independent-Interface SECS Communication Module Controller, i-SECS)軟體。研究重點在於利用統一塑模語言(UML)、設計模式(Design Patterns)等開發工具以物件導向觀念設計一個整合容易、擴充方便的獨立介面通訊模組軟體控制器架構，並以Java程式語言來實作，達到控制器跨平台操作的需求。論文中說明集束型半導體製程設備(Cluster Tools)與光電產業製程設備的硬體與軟體架構，同時介紹有關通訊介面的規範以及獨立介面通訊模組所應具備的功能，最後並以實驗室內現有的集束型半導體製程設備來驗證所實作出來的獨立介面通訊模組控制器，此通訊模組可以整合於集束型製程設備中亦可單機獨立操作測試機台通訊。

# **The Development of SECS Communication Module for Cluster Tools and Photoelectric Process Equipments**

Student : Meng-Chang CHAN

Advisor : Dr. An-Chen LEE

Institute of Mechanical Engineering  
National Chiao Tung University

Abstract

The logo of National Chiao Tung University is a circular emblem with a gear-like border. Inside the circle, there are stylized letters 'E', 'S', and 'A' arranged vertically, and the year '1896' at the bottom. The logo is semi-transparent and overlaid on the abstract text.

In this thesis, we use Object-Oriented Analysis/Design (OOA/D) method to develop an Independent-Interface SECS Communication Module Controller (i-SECS) that conforms to SEMI standard. The main point is to use Unified Modeling Language (UML) and Design Patterns to develop a flexible and reusable software architecture, where the UML is adopted to analyze/design the controller, and Java programming language is used to implement i-SECS controller. In the presentation flow of the thesis, we will introduce cluster tools, photoelectric process equipments, communication interface and the capabilities of i-SECS. Finally, we link i-SECS controller to cluster tools in our Laboratory to test its capacity and performance, and the results are presented as well. The developed SECS communication module can be integrated into cluster tools, and also run stand-alone to test equipment communication.

## 誌 謝

終於到了論文完稿的時候，轉眼間兩年的研究生生涯即將結束，感恩指導教授李安謙博士在這兩年多來的教導與鼓勵，使我不論是在『做人、做事、做學問』各方面都受到老師深遠的教誨與督促，讓我更有信心可以面對未來職場上的競爭壓力。

感謝陳岳汶、童一峰、周亨隆同學這兩年來彼此的互相鼓勵與扶持，使我的研究生涯留下一段非常值得留念的回憶。感謝在職專班的曹洪泰、蔡俊德、王志峰、洪志明、黃啟業、陳文燦同學的技術指導與建議，讓我能順利完成我的論文，同時學到很多業界的經驗與知識，對我的人生視野擴展許多。

感謝張文耀學長、戴慶忠學長、李妍慧學姊的傳承與教導，洪榮煌學長的意見諮詢與鼓勵，以及建峰、坤樺與柏壽的支持與幫忙，讓我能順利完成論文的工作。還有去年畢業的志明學長跟老潘學長，碩一的時光，因為有你們的鼓勵與提攜，讓我成長許多，感謝你們，謝謝！

最後，我最感謝的是爸爸、媽媽跟兩位哥哥，從我退伍後到碩士畢業這三年多來一直在背後傾全力的支持，讓我在求學階段無後顧之憂，有你們的關懷、支持與鼓勵，一直是我最大的動力來源，今天我要將這份榮耀獻給你們，謝謝你們。

# 目 錄

|                               |     |
|-------------------------------|-----|
| 摘 要 .....                     | I   |
| ABSTRACT .....                | II  |
| 誌 謝 .....                     | III |
| 目 錄 .....                     | IV  |
| 圖 目 錄 .....                   | IX  |
| 表 目 錄 .....                   | XII |
| 第 1 章 緒論 .....                | 1   |
| 1.1 研究動機與目的 .....             | 1   |
| 1.2 文獻回顧 .....                | 2   |
| 1.3 研究方法 .....                | 4   |
| 1.4 本文組織架構 .....              | 6   |
| 第 2 章 統一塑模語言介紹 .....          | 7   |
| 2.1 UML的特色與優點 .....           | 7   |
| 2.2 UML圖形 .....               | 8   |
| 第 3 章 半導體與光電產業製程設備介紹 .....    | 15  |
| 3.1 半導體製程設備介紹 .....           | 15  |
| 3.1.1 集束型製程設備之硬體架構 .....      | 16  |
| 3.1.2 集束型製程設備之軟體架構 .....      | 19  |
| 3.2 光電產業製程設備介紹 .....          | 20  |
| 3.2.1 TFT-LCD陣列製程設備硬體架構 ..... | 20  |
| 3.2.2 TFT-LCD陣列製程設備軟體架構 ..... | 21  |

|         |   |    |
|---------|---|----|
| 3.3     | 通訊模組於半導體與光電製程設備上之應用 .....                       | 22 |
| 第 4 章   | 通訊模組之設計與實作 .....                                | 23 |
| 4.1     | HSMS (HIGH SPEED SECS MESSAGE SERVICES)介紹 ..... | 23 |
| 4.1.1   | HSMS訊息格式 .....                                  | 23 |
| 4.1.2   | HSMS的逾時機制 .....                                 | 26 |
| 4.1.3   | HSMS連線 .....                                    | 26 |
| 4.1.3.1 | 被動端模式(Passive Mode).....                        | 27 |
| 4.1.3.2 | 主動端模式(Active Mode) .....                        | 28 |
| 4.1.4   | HSMS訊息交握程序.....                                 | 29 |
| 4.2     | SECS-II訊息資料介紹.....                              | 31 |
| 4.2.1   | SECS-II訊息資料結構 .....                             | 31 |
| 4.3     | DESIGN PATTERNS介紹與應用 .....                      | 33 |
| 4.3.1   | Facade Pattern .....                            | 33 |
| 4.3.2   | Bridge Pattern.....                             | 34 |
| 4.3.3   | Composite Pattern.....                          | 34 |
| 4.3.4   | Visitor Pattern .....                           | 36 |
| 4.3.5   | Design Patterns在通訊模組上之應用 .....                  | 37 |
| 4.4     | 通訊模組核心程式系統架構 .....                              | 39 |
| 4.4.1   | 通訊模組的靜態架構 .....                                 | 39 |
| 4.4.2   | 訊息編譯循序圖 .....                                   | 40 |
| 4.4.3   | 訊息解譯循序圖 .....                                   | 41 |
| 4.4.4   | 傳送訊息循序圖 .....                                   | 43 |
| 4.4.5   | 接收訊息活動圖 .....                                   | 44 |
| 4.5     | SECS MESSAGE LANGUAGE介紹.....                    | 45 |
| 4.5.1   | SML標記方式.....                                    | 45 |

|           |                                      |    |
|-----------|--------------------------------------|----|
| 4.5.2     | SML於通訊模組上的應用 .....                   | 47 |
| 第 5 章     | 獨立介面通訊模組控制器軟體設計與實作 .....             | 51 |
| 5.1       | i-SECS控制器介紹 .....                    | 51 |
| 5.1.1     | i-SECS控制器功能分析 .....                  | 52 |
| 5.1.2     | i-SECS功能模組介紹 .....                   | 53 |
| 5.2       | i-SECS控制器架構設計 .....                  | 54 |
| 5.2.1     | 架構(Framework)介紹 .....                | 54 |
| 5.2.2     | i-SECS控制器的架構設計 .....                 | 54 |
| 5.2.3     | i-SECS控制器程式實作 .....                  | 55 |
| 5.3       | i-SECS使用者介面介紹 .....                  | 58 |
| 5.3.1     | GUI概述 .....                          | 58 |
| 5.3.2     | 控制畫面視窗介紹 .....                       | 60 |
| 5.3.2.1   | Main View (SML Logging) GUI說明 .....  | 60 |
| 5.3.2.2   | State Diagram GUI說明 .....            | 61 |
| 5.3.2.3   | i-SECS Monitor GUI說明 .....           | 62 |
| 5.3.2.4   | Data Collect GUI說明 .....             | 63 |
| 5.3.2.4.1 | Event Data Collection GUI說明 .....    | 63 |
| 5.3.2.4.2 | Variable Data Collection GUI說明 ..... | 64 |
| 5.3.2.4.3 | Trace Data Collection GUI說明 .....    | 65 |
| 5.3.2.5   | Recipe Editor GUI說明 .....            | 66 |
| 5.3.2.6   | Alarm Record GUI說明 .....             | 67 |
| 5.3.2.7   | Event Record GUI說明 .....             | 67 |
| 5.3.2.8   | Terminal Services GUI說明 .....        | 68 |
| 第 6 章     | 系統測試 .....                           | 69 |
| 6.1       | GEM SCENARIO介紹 .....                 | 69 |

|           |   |    |
|-----------|---|----|
| 6.1.1     | GEM簡介 .....                               | 69 |
| 6.1.2     | 設備性能和上層主機通訊介面劇本 .....                     | 70 |
| 6.1.2.1   | 建立連線 (Establish Communications).....      | 71 |
| 6.1.2.2   | 資料收集 (Data Collection).....               | 71 |
| 6.1.2.2.1 | Event Data Collection (事件報告收集) .....      | 71 |
| 6.1.2.2.2 | Variable Data Collection (變數資料收集) .....   | 73 |
| 6.1.2.2.3 | Trace Data Collection (追蹤資料收集) .....      | 75 |
| 6.1.2.2.4 | ON-LINE Identification (ON-LINE狀態確認)..... | 75 |
| 6.1.2.3   | 警報管理 (Alarm Management).....              | 76 |
| 6.1.2.4   | 製程配方管理 (Process Recipe Management).....   | 76 |
| 6.1.2.5   | 材料搬運 (Material Movement) .....            | 77 |
| 6.1.2.6   | 設備終端機服務(Equipment Terminal Services)..... | 78 |
| 6.1.2.7   | 訊息格式錯誤回報 (Error Message) .....            | 78 |
| 6.1.2.8   | 時間調整 (Clock).....                         | 80 |
| 6.1.2.9   | 控制狀態圖的遷移.....                             | 81 |
| 6.2       | I-SECS與CTC之整合測試 .....                     | 83 |
| 6.2.1     | 連線測試硬體介紹.....                             | 83 |
| 6.2.2     | 連線測試軟體工具介紹.....                           | 83 |
| 6.2.3     | i-SECS與CTC實機測試結果 .....                    | 84 |
| 6.2.4     | 測試結論 .....                                | 90 |
| 第 7 章     | I-SECS在光電製程設備上的應用規劃.....                  | 91 |
| 7.1       | 通訊模組在光電製程設備上的應用 .....                     | 91 |
| 7.2       | I-SECS在光電製程設備上的應用規劃.....                  | 94 |
| 第 8 章     | 結論與未來展望 .....                             | 95 |
| 8.1       | 結論 .....                                  | 95 |

|                      |    |
|----------------------|----|
| 8.2 未來展望.....        | 96 |
| 參考文獻.....            | 97 |
| 附錄一 SECS-II訊息明細..... | 99 |



## 圖目錄

|       |                             |    |
|-------|-----------------------------|----|
| 圖 2.1 | 使用案例圖 .....                 | 8  |
| 圖 2.2 | 類別圖 .....                   | 9  |
| 圖 2.3 | 物件圖 .....                   | 10 |
| 圖 2.4 | 循序圖 .....                   | 11 |
| 圖 2.5 | 狀態圖 .....                   | 12 |
| 圖 2.6 | 活動圖 .....                   | 13 |
| 圖 2.7 | 元件圖 .....                   | 13 |
| 圖 2.8 | 部署圖 .....                   | 14 |
| 圖 2.9 | 合作圖 .....                   | 14 |
| 圖 3.1 | 集束型製程設備 .....               | 15 |
| 圖 3.2 | 300mm集束型製程設備硬體示意圖 .....     | 16 |
| 圖 3.3 | 傳輸腔體晶圓承載系統 .....            | 17 |
| 圖 3.4 | 傳輸腔體真空機械手臂系統 .....          | 17 |
| 圖 3.5 | 製程模組系統(RTP) .....           | 18 |
| 圖 3.6 | 300mm集束型製程設備控制器軟體架構圖 .....  | 19 |
| 圖 3.7 | TFT-LCD陣列製程設備 .....         | 21 |
| 圖 3.8 | 光電產業製程設備控制器軟體架構圖 .....      | 21 |
| 圖 4.1 | HSMS訊息格式 .....              | 24 |
| 圖 4.2 | HSMS狀態圖 .....               | 26 |
| 圖 4.3 | HSMS循序圖 .....               | 30 |
| 圖 4.4 | HSMS循序圖 (續) .....           | 30 |
| 圖 4.5 | SECS-II S4F27 訊息架構 .....    | 32 |
| 圖 4.6 | Data Item與List的Header ..... | 32 |

|        |                                  |    |
|--------|----------------------------------|----|
| 圖 4.7  | Bridge Pattern類別圖 .....          | 34 |
| 圖 4.8  | Composite Pattern 類別圖 .....      | 35 |
| 圖 4.9  | Visitor Pattern 類別圖 .....        | 36 |
| 圖 4.10 | 應用Design Patterns的通訊模組類別圖 .....  | 38 |
| 圖 4.11 | SECS通訊模組類別圖 .....                | 39 |
| 圖 4.12 | S4F19 訊息編譯的循序圖 .....             | 41 |
| 圖 4.13 | 訊息解譯的循序圖 .....                   | 42 |
| 圖 4.14 | 傳送訊息循序圖 .....                    | 43 |
| 圖 4.15 | 接收訊息活動圖 .....                    | 44 |
| 圖 4.16 | 通訊模組SML語法類別圖 .....               | 47 |
| 圖 4.17 | 接收訊息後顯示SML語法的循序圖 .....           | 49 |
| 圖 4.18 | 讀取SML檔案傳送訊息時的循序圖 .....           | 50 |
| 圖 5.1  | i-SECS硬體架構圖 .....                | 51 |
| 圖 5.2  | i-SECS Use Case Diagram .....    | 52 |
| 圖 5.3  | i-SECS接收HOST端傳遞訊息循序圖 .....       | 56 |
| 圖 5.4  | 應用Bridge Pattern的時間調整類別圖 .....   | 57 |
| 圖 5.5  | i-SECS GUI Layout .....          | 58 |
| 圖 5.6  | Communication Status Panel ..... | 59 |
| 圖 5.7  | Screen Select Panel .....        | 59 |
| 圖 5.8  | i-SECS GUI主畫面視窗 .....            | 60 |
| 圖 5.9  | HSMS連線狀態圖 .....                  | 61 |
| 圖 5.10 | GEM連線狀態圖 .....                   | 61 |
| 圖 5.11 | i-SECS Monitor .....             | 62 |
| 圖 5.12 | Event Data Collection .....      | 63 |
| 圖 5.13 | Variable Data Collection .....   | 64 |
| 圖 5.14 | Trace Data Collection .....      | 65 |

|        |                        |    |
|--------|------------------------|----|
| 圖 5.15 | Recipe Editor .....    | 66 |
| 圖 5.16 | Alarm Recoed .....     | 67 |
| 圖 5.17 | Event Record .....     | 68 |
| 圖 5.18 | Terminal Services..... | 68 |
| 圖 6.1  | ON-LINE架構圖 .....       | 69 |
| 圖 6.2  | 設備連線事件報告測試.....        | 85 |
| 圖 6.3  | 製程配方管理資料庫連結.....       | 86 |
| 圖 6.4  | 製程配方管理建立與刪除資料表.....    | 86 |
| 圖 6.5  | 事件資料收集測試 .....         | 87 |
| 圖 6.6  | 變數資料收集測試 .....         | 88 |
| 圖 6.7  | 追蹤資料收集測試 .....         | 89 |



## 表目錄

|        |   |    |
|--------|---|----|
| 表 4.1  | HSMS 訊息格式說明 .....   | 24 |
| 表 4.2  | HSMS Message Header .....   | 24 |
| 表 4.3  | HSMS 訊息格式比較 .....   | 25 |
| 表 4.4  | SECS-II 訊息的資料型別 .....   | 31 |
| 表 4.5  | 通訊模組參與物件說明表 .....   | 40 |
| 表 4.6  | SML 標示語法說明 .....  | 45 |
| 表 4.7  | SML Data Item Format .....  | 46 |
| 表 4.8  | 通訊模組 SML 語法參與物件說明表 .....  | 48 |
| 表 5.1  | 時間調整類別說明 .....  | 57 |
| 表 6.1  | Host Attempts to Establish Communication .....                    | 71 |
| 表 6.2  | Event Report from Equipment .....                                 | 72 |
| 表 6.3  | Host Requests Event Report .....                                  | 72 |
| 表 6.4  | Event Reporting Setup .....                                       | 72 |
| 表 6.5  | Host Requests Individual Report .....                             | 73 |
| 表 6.6  | Host Requests Current Status Variable Value .....                 | 74 |
| 表 6.7  | Host Requests Status Variable Namelist .....                      | 74 |
| 表 6.8  | Host Requests Equipment Current Constant Values .....             | 74 |
| 表 6.9  | Host Requests Equipment Constant Namelist .....                   | 74 |
| 表 6.10 | Trace Data Collection .....                                       | 75 |
| 表 6.11 | Host Initiated .....  | 75 |
| 表 6.12 | Send Alarm Report .....   | 76 |
| 表 6.13 | Process Program Creating, Editing, or Deleting on Equipment. .... | 76 |
| 表 6.14 | Deleting a Process Program by Host .....                          | 76 |

|        |   |    |
|--------|---|----|
| 表 6.15 | Host Requests Equipment Process Program Directory ..... | 77 |
| 表 6.16 | Equipment-initiated Process Program Uploading.....      | 77 |
| 表 6.17 | Host-initiated Process Program Downloading.....         | 77 |
| 表 6.18 | Material Movement .....                                 | 77 |
| 表 6.19 | Host Sends Information to Equipment .....               | 78 |
| 表 6.20 | Operator Sends TEXT Information to Host .....           | 78 |
| 表 6.21 | Message Fault due to Unrecognized Device ID .....       | 79 |
| 表 6.22 | Message Fault due to Unrecognized Stream Type .....     | 79 |
| 表 6.23 | Message Fault due to Unrecognized Function Type .....   | 79 |
| 表 6.24 | Message Fault due to Unrecognized Data Format.....      | 79 |
| 表 6.25 | Message Fault due to Transaction Timer Timeout .....    | 79 |
| 表 6.26 | Equipment Requests Time Value from HOST .....           | 80 |
| 表 6.27 | Host Requests Equipment to Set Time .....               | 80 |
| 表 6.28 | Host Requests Time Value from Equipment.....            | 80 |
| 表 6.29 | Host Accepts ON-LINE Request from Equipment.....        | 81 |
| 表 6.30 | Host Refuses ON-LINE Request.....                       | 81 |
| 表 6.31 | Operator Attempts to Enter OFF-LINE, REMOTE and LOCAL . | 81 |
| 表 6.32 | Host Requests to Enter OFFLINE .....                    | 82 |
| 表 6.33 | Host Attempts Enter ONLINE.....                         | 82 |
| 表 6.34 | Host Confirms Equipment Presence .....                  | 82 |
| 表 6.35 | 連線測試硬體規格說明表.....  | 83 |
| 表 6.36 | i-SECS與Emulator的比較.....                                 | 90 |
| 表 7.1  | S5F1 訊息在半導體與光電製程設備上的比較.....                             | 92 |
| 表 7.2  | S6F11 訊息在半導體與光電製程設備上的比較.....                            | 92 |
| 表 7.3  | 光電製程設備Damper機台訊息摘要說明 .....                              | 93 |

# 第1章 緒論

## 1.1 研究動機與目的

近年來政府大力推動兩兆雙星計畫，期能提升台灣高科技產業之競爭力，而國內的半導體廠也積極的投入 12 吋(300mm)晶圓廠的規劃與興建，光電廠則紛紛投入第六代、第七代的大尺寸面板生產的行列。面對此兩種產業在大面積基材的處理與搬運，傳統的人工搬運方式已無法應付，必須仰賴機械手臂(Robot)、無人搬運車(Automatic Guided Vehicles, AGV)運送才能達成，因此半導體和光電產業的設備製造商皆以多製程腔體的集束型製程設備(Cluster Tools)來將具有相關性的製程腔體整合到一個機台上，透過設備自動化降低晶圓和玻璃基板在不同製程運輸過程中可能遭受到的人為污染，以提高整體的設備有效利用率(Overall Equipment Effectiveness, OEE)。

集束型製程設備的優點包括：高真空度、潔淨度、可靠度、精密度、設備佔地面積小、產出率高、並提供製程選擇彈性化等優點。此外由於半導體產業對於生產環境的潔淨度要求極高，因此生產過程中任何搬運動作所造成的污染都會降低產品的良率。在光電廠中，雖然對生產環境的潔淨度不像半導體設備嚴苛，但是設備在製程中產生的靜電問題同樣也會降低產品的良率。一個功能完善的集束型製程設備控制器(Cluster Tools Controller, CTC)必須有能力產生並監管在集束型製程設備當中所發生的各種處理程序(包括製程、傳輸等處理程序)，一旦操作員或設備上層主機指派新的生產任務時，集束型製程設備控制器即依據生產目的產生傳輸工作及製程工作，並透過通訊模組的訊息交握將傳輸工作與製程工作交與相關模組，再由相關模組對晶圓或玻璃基板進行處理，以完成生產的目的。

在集束型製程設備當中，各個模組都有一個控制器用來管理模組內部資源與監督動作行程，模組之間並透過嚴密規範的通訊機制來相互溝通，CTC負責模組之間相互協調、整合及監控的工作，並提供介面給設備上層主機及操作員使用。集束型製程設備可

根據廠商需求，在傳輸腔體周圍掛上所需的製程腔體，因此有很大的彈性可以調整所需要的製程。然而設備商各自發展自有品牌的製程腔體，對製造廠而言雖然選擇增加，但是在整合上卻是一大難題。因此經由國際半導體設備及材料協會(Semiconductor Equipment and Materials International, SEMI)推動，及全球半導體製造廠和設備廠共同參與討論表決訂定集束型製程設備的硬體介面標準及軟體通訊協定標準，以提供不同模組之間傳輸與溝通介面的標準化，以利於整合到集束型製程設備內。

本文的目的是利用軟體工程上物件導向分析與設計(Object-Oriented Analysis/Design, OOA/D)的方法，以統一塑模語言(Unified Modeling Language, UML)發展出一套階層式的軟體控制器架構，並實際地以物件導向語言Java來設計出滿足SEMI 標準的獨立介面通訊模組控制器軟體。此通訊模組可以整合於集束型製程設備中亦可單機獨立操作測試機台通訊。



## 1.2 文獻回顧

在文獻[1]中Ernest J. Wood先生提到利用C++程式以物件導向觀念來架構通訊模組程式，稱之為OOSECS(Object-Oriented SECS System)，其特點為利用物件導向的特性，將每個資料型別的物件串成一個訊息的結構物件，來達到通訊模組訊息溝通的目的。OOSECS以SECS-I(RS232)[2]的通訊介面來傳遞訊息，並使用C++程式語言來開發，每個訊息都是由物件所組成，佔用了很大的動態隨機存取記憶體，若是沒有自動的記憶體釋放機制，很容易在訊息傳遞過程中因為訊息物件太大而造成記憶體耗盡的問題。為了能有效的管理記憶體，因此我們以Java程式語言來實作通訊模組，Java提供良好的記憶體回收(Garbage Collection)機制，讓程式在開發時不需要去煩惱記憶體管理的問題，可以將開發重心放在訊息物件內容的正確性上，以減少訊息發生錯誤的機會。

在文獻[3][4]中，通訊模組的設計構想是來自於集束型半導體製程設備(Cluster Tools)上的應用，在開發集束型半導體製程設備控制器(CTC)時，為了要統一各模組控

制器的訊息傳遞格式，因此遵照 SEMI Standard 的通訊協定介面 SECS-II(E5)[5] 與 HSMS(E37)[6] 標準設計了此通訊模組，其應用在集束型半導體製程設備的發展上已相當的完備，目前較欠缺的是通訊模組的使用者介面(Graphic User Interface, GUI)以及與上層主機間的通訊介面(GEM)[7]。為了充分利用通訊模組訊息傳遞的功能，使其能夠廣泛的應用在半導體與光電領域的製程設備上，因此本文延續集束型半導體製程設備通訊模組的架構，建構一套具有獨立操作介面的通訊模組軟體控制器，利用樹狀結構的方式來表示訊息物件的內容，以更友善的使用者介面進行連線的建立與訊息物件的傳遞，並顯示所有傳送與接收的訊息，使得整個通訊模組在應用上更為廣泛便捷，可以快速的瞭解並使用通訊模組的功能。

文獻[8]提出了一種標記法(Notation)的觀念，用在表示不同 Patterns 所組合而成的 UML 架構上，此種標記法可以維持 Patterns 之間相關的資訊，不會因為不同 Patterns 的結合而失去 Patterns 的資訊，因此可以擴展 UML 的功能，讓設計者更快速的理解程式所要表達的意涵。

在本文中，我們提出以 UML 與 Design Patterns 的觀點來設計通訊模組，其目的是使通訊模組達到可再利用(Reusable)與可再擴充(Extensible)的特性，此模組主要是應用 UML 來做程式的分析與設計，同時在程式開發時加入 Design Patterns 的概念，以 Composite Pattern 與 Visitor Pattern 來實現通訊模組資料結構與應用(編碼、解碼)分開的目的，讓程式在未來擴充上有很大的彈性，可以直接透過新增訊息物件與編碼解碼的方式來擴充通訊模組的功能。

## 1.3 研究方法

本文主要設計符合 SEMI 標準的通訊模組軟體。通訊模組介面採用 SEMI 標準中的 SECS-II、HSMS 的規範。本節說明在通訊模組軟體發展過程中所會應用到的軟體工具與研究方法，分述如下：

### ■ 物件導向(Object-Oriented) [9]

物件導向程式設計具有以下優點：

1. 物件的模型可對應到真實世界的實際物體，利於溝通瞭解；
2. 物件具有封裝性，當程式異動時，物件介面維持不變，只對內部做些微更動即可，使程式擴展容易；
3. 物件具有可再利用性，功能類似的物件其程式碼可再利用，以減少程式撰寫時間與測試時間，容易維護並提高可靠度。



### ■ 統一塑模語言(UML) [10]

這是一種用來針對物件導向軟體作分析與設計的模式語言，它提供豐富的圖形表示法來表示物件的靜態架構、動態行為、生命週期、狀態變化等，從各種圖形表示法當中設計者可以瞭解到可能發生的問題及是否滿足設計需求。在物件導向程式語言已成為軟體撰寫主流之際，這種物件導向分析模型語言也愈來愈多，UML 在 1997 年 11 月已正式通過物件管理組織(OMG)的認證，成為物件導向模型語言的工業標準，因此以 UML 作為本文物件導向分析與設計的工具。

### ■ Java 2 SDK1.4 [11]

這是昇陽公司(Sun Microsystems)所發展的一套程式語言，它提供比 C++ 更完整的物件導向機制，使得用 Java 寫成的程式更精簡、更易於瞭解與維護。Java 語言本身亦

提供了許多發展套件，程式開發者可以直接運用這些套件來發展應用程式，大大縮短程式發展時間。此外 Java 語言最大的特點在於其具有跨平台的特性，一但程式碼寫成，在任何作業平台都可以執行。Java 現今已成為物件導向語言的主流，因此本文採用 Java 作為發展通訊模組的程式語言工具。

#### ■ Rational Rose Enterprise [12]

這是由 Rational 公司所發展用來架構 UML 程式的開發軟體，Rose 提供了程式開發者架構 UML 的工作環境，可以利用正向工程(Forward Engineer)產生程式碼，當程式碼變更時，也可以利用逆向工程(Reverse Engineer)將變更後的程式架構轉為 UML 語言，減少程式開發的時程。

#### ■ JBuilder® 9 Personal [13]

這是由 Borland 公司所開發的跨平台 Java 語言開發環境，JBuilder9 Personal 是 Borland 公司針對欲學習 Java 程式開發之學生或個人，提供一個最佳的 Java 整合開發環境，它提供了一般開發程式所需的架構功能與使用者介面(GUI)，讓程式開發者可以更快速的建構與除錯所開發的應用程式，提升程式開發的效率。

#### ■ Design Patterns [14]

Design Patterns 是一種敘述在不同情形時，物件導向程式設計的思維，尤其著重在該如何利用物件導向的性質以引出「再利用性」和「彈性」的觀念。在設計程式時只要利用已經存在的設計模式，就能活用原來設計者的寶貴經驗。許多物件導向系統裡面都有一再出現的類別互動模式，這些模式不僅解決了特定問題，同時也讓物件導向系統更富彈性、且更具可再利用的性質。

## 1.4 本文組織架構

本文共分八章，第一章緒論；第二章為統一化模式語言(UML)之介紹，介紹 UML 的特性與其所定義的各種圖形；第三章介紹集束型半導體製程設備與光電產業製程設備的硬體架構與控制器的軟體架構，並說明獨立介面通訊模組在上述的製程設備中所扮演的角色；第四章為通訊模組之設計與實作，介紹通訊介面的標準規範以及通訊模組所應具備的功能；第五章介紹獨立介面通訊模組控制器軟體之設計與實作，說明控制器的功能及使用者介面的操作；第六章以實驗室的集束型製程設備來做系統測試，驗證設備控制器與上層主機間的通訊介面；第七章說明通訊模組在光電製程設備上的應用規劃，提出一個程式修改步驟以擴充獨立介面通訊模組的功能，期能應用到光電產業的製程設備上；第八章則說明結論與未來的展望。



## 第2章 統一塑模語言介紹

統一塑模語言(Unified Modeling Language, UML)[10]是一種物件導向模型語言，提供規格化、視覺化以及文件化的表示方法，使用這些表示法可以描述一個真實或抽象的系統，包括：使用案例圖、類別圖、物件圖、循序圖、狀態圖、活動圖、元件圖、部署圖和合作圖等。目標主要在於發展一個簡單、可擴展、並可立即使用的視覺化模型語言。自UML推出以來，即廣受領導廠商與正式組織的支援，並且經物件管理組織(Object Management Group, OMG)採納為標準語言，已經廣被物件導向發展產業所接受。UML運用廣泛，涵蓋多種工具，而且是業界標準的模型語言，適用於各種型態的系統、專業領域、以及實務方法，因此本文使用UML工具以物件導向方式來開發通訊模組的架構。

### 2.1 UML 的特色與優點



1. UML 是一種塑模語言，它不只是一種用來描繪圖形的表示法，而是一種為了達成溝通的目標，並可描繪某個主題的知識(語意)，以及展現關於此主題之知識(語法)的語言。
2. UML 是整合資訊系統與工業界最佳工程實作(原理、技術、方法和工具)的結果，適用於各種不同型態的系統、領域、方法或程序。
3. UML 是以物件導向的方法將系統作完整的描述、設計、建構與註解，並且提供策略及行動知識的保存、傳遞及平衡，藉由提高品質、降低成本以提高價值，並控制可能增加的改變與複雜度所帶來的風險。
4. UML 是一個簡單、可擴展、並可立即使用的視覺化模型語言，擁有擴展所需的延展性及專門化機制，並與發展的過程和實作的程式語言無關，因此具有廣泛的適用性。

## 2.2 UML 圖形

UML 定義了九種型態的圖形：使用案例圖、類別圖、物件圖、循序圖、狀態圖、活動圖、元件圖、部署圖和合作圖。所有圖形的基本原則，就是將概念描繪成符號，並將概念之間的相互關係描繪為連接符號的路徑(直線)。其內容分述如下：

1. 使用案例圖(Use Case Diagram)：沒有系統是隔離單獨存在的，因此系統通常需與人或其他系統交互作用來達到預期的目標，使用案例圖描述了一個系統的行為與功能，以及此系統的使用者(如圖 2.1)及其之間的關係。這些圖形包含以下的元素：
  - a. 行為者(Actor)：表示此系統的使用者，包括人類的使用者及其它的系統。
  - b. 使用案例(Use Case)：表示一個系統提供給使用者的功能或服務。

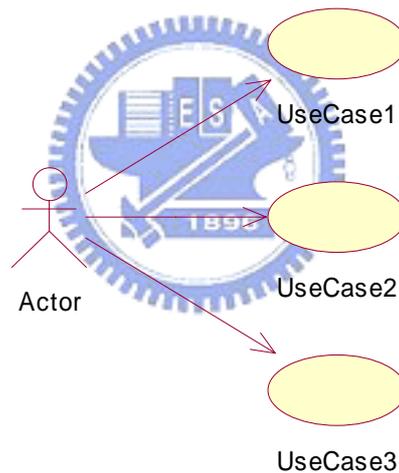


圖 2.1 使用案例圖

2. 類別圖(Class Diagram)：類別圖是一般物件導向模型最常見的圖形，描述一個系統的靜態結構，而不討論涉及動態行為的活動，其可以將一組類別、介面、合作及其之間的關係顯示出來。類別圖主要是由屬性、操作方法和結合關係所組成；屬性是類別的結構化特徵，也就是關於類別的資訊，操作方法是類別的行為化特徵，也就是類別所提供的服務，而結合關係則說明了類別與類別間彼此的互動。這些圖形包含以下元素：

- a. **類別(Class)**：表示擁有共同性質或特徵的物體，這些特徵包含屬性、操作方法。類別中所定義的可視性(Visibility)，用來表示此屬性或操作方法是否可在類別之外被使用。公開(Public)的可視性，表示在類別之外可以使用；受保護的(Protected)可視性，表示在類別之外無法存取，但是若與此類別有一般化的關係則能存取；私有(Private)的可視性，表示類別之外無法存取。
- b. **相依關係(Dependency)**：指類別與類別之間的使用關係，它代表被使用類別(Event)的改變會影響到另一個類別(Window)的操作。在 UML 中以虛線表示，方向會從某一類別指到被依靠(Depended on)的類別。以圖 2.2 為例，一個視窗(Window)類別應該具備的屬性有起始位置和大小，而其方法包含：打開、關上、移動和顯示。而且視窗(Window)和事件(Event)之間有相依(Dependency)的關係。

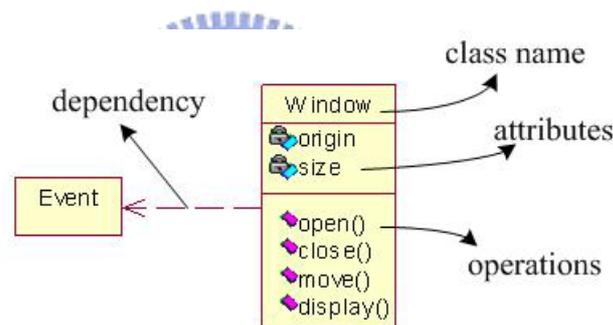


圖 2.2 類別圖

- c. **結合關係(Association)**：表示與其它兩個或多個類別之間的相互關係，這些相互關係擁有共同的性質或特徵，即一個類別可以知道另一個類別的公開屬性與操作方法的關係，在 UML 中以實線表示。
- **聚合關係(Aggregation)**：說明一個類別由其它的類別所組成，但彼此之間並不相關聯，當上層類別被刪除後，其所聚集的類別依然存在。由於聚合關係是結合關係的一種，在 UML 中以實線連接，菱形指向整體(Whole)。菱形為空心即表示依照參考(By-reference)的聚合關係；若為實心即表示依照值(By-value)的聚合關係。



類別必須提供這些訊息和操作方法。循序圖也可以用來實現使用案例，因為其指明了類別角色之間藉由訊息的互換來達成某種功能，藉以實現使用案例。

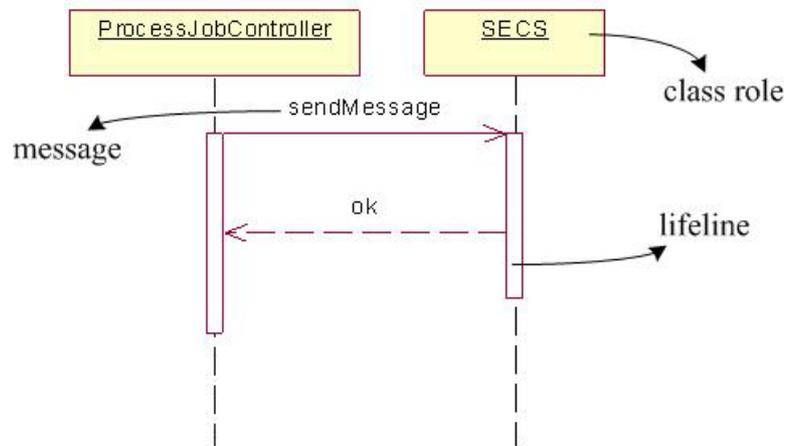


圖 2.4 循序圖

以圖 2.4 說明控制器送出 SECS-II 訊息給外界的循序圖。ProcessJobController 物件呼叫 SECS 物件的 sendMessage() 方法來送出 SECS-II 訊息給外界，並回傳 ok 表示送出訊息成功。

5. 狀態圖(State-Chart Diagram)：狀態圖描述了一個類別的狀態和回應，它描述了一個類別對於外來刺激的反應，以強調物件行為的事件順序。它包含了以下元素：
  - a. **狀態(state)**：表示一個物件在生命週期中之狀況，在此期間此物件滿足某些條件、執行某些活動或是等待某些事件出現。狀態圖中包含了開始狀態(Start State)，以小實心圓表示，也包含了終止狀態(End State)，以實心的同心圓表示。當到達終止狀態時，該物件在圖中所描繪的狀態都將被摧毀。
  - b. **轉換(transition)**：表示一個物件在不同狀態之間的交互關係。

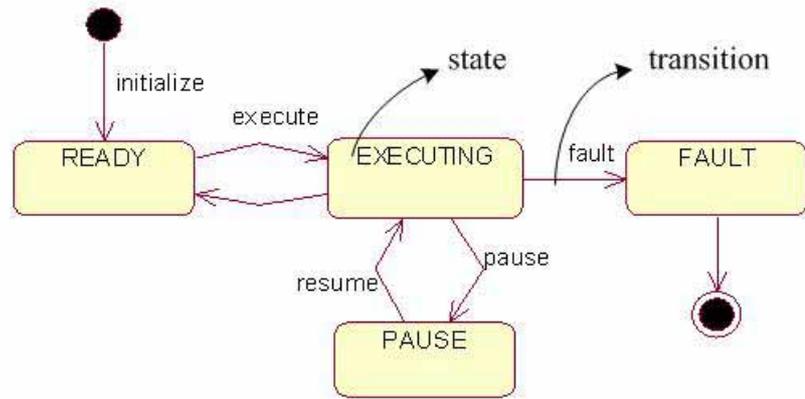


圖 2.5 狀態圖

圖 2.5 說明一般控制系統狀態的改變。初始化後進入 READY 的狀態；當收到 execute 的命令後進入 EXECUTING 的狀態；執行的過程中發生錯誤則跳到 FAULT 的狀態；若收到 pause 命令則進入 PAUSE 的狀態；等收到 resume 命令再回到 EXECUTING 的狀態；如果正常執行完畢則回到 READY 的狀態。



6. 活動圖(Activity Diagram)：活動圖敘述一個類別的活動，強調活動與活動之間的流程，這種圖形很類似狀態圖，也都使用相似的慣例。活動圖是描述一個類別回應內部處理的行為，狀態圖則是描述回應外部事件的行為。游泳水道(Swinlane)用來表示在整個活動中，一個或多個負責動作的物件；亦即將活動狀態分成好幾組，再將這些群組分配給必須執行此活動的物件。動作狀態(如圖 2.6 之 Activity1)是用來表示物體無法被中斷的動作，或是一個演算法執行過程中的步驟。狀態中間也可以是判斷式(如圖 2.6 之菱形圖形)，作為下一動作狀態的條件。如同狀態圖一般，活動圖也同樣使用開始狀態和終止狀態。

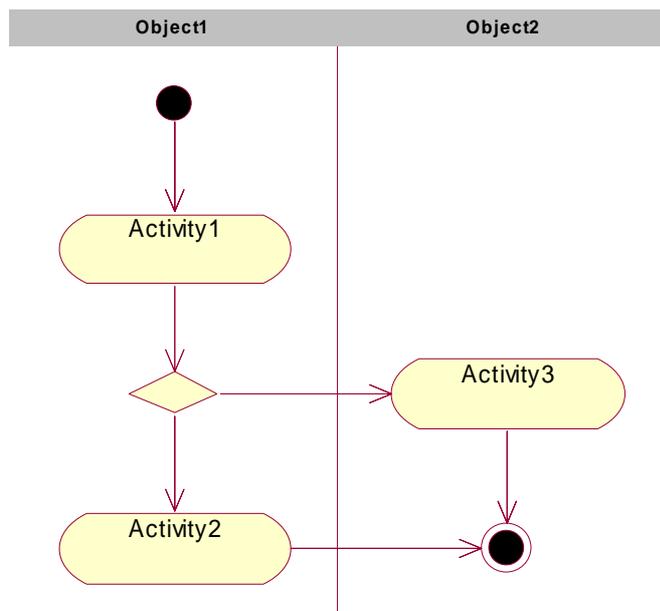


圖 2.6 活動圖

7. 元件圖(Component Diagram)：元件圖敘述軟體實作元件的組織架構及其相依關係，這些圖形包含元件，元件表示可拆散的實體單位，包括原始程式碼、目的碼、以及可執行程式碼或函式庫。圖 2.7 所示為一個執行製程的可執行程式的示意圖。

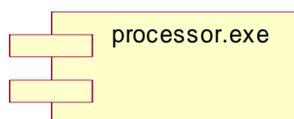


圖 2.7 元件圖

8. 部署圖(Deployment Diagram)：部署圖敘述處理資源元素的組態，以及軟體實作元件的對應方式。這些圖形包含了元件和節點(node)，節點表示一種處理或計算的資源。圖 2.8 說明個人電腦透過網際網路向伺服器請求一些資訊的示意圖。Client 透過 TCP/IP 連上 Application Server，Application Server 從資料庫裡搜尋資料，再將結果回傳給 Client。

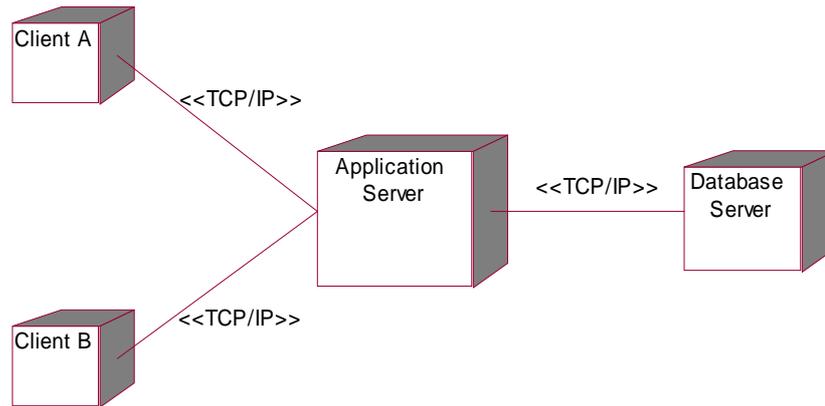


圖 2.8 部署圖

9. 合作圖(Collaboration Diagram)：合作圖也是互動圖的一種，其定義如何透過系統中的元件將行為具體化，包含類別和結合關係之間的相互作用，這些相互作用被模組化成類別之間的訊息交握，而訊息的交握是藉由類別之間的結合關係來進行。圖 2.9 說明新增一個顧客的合作圖。使用者新增一個顧客時，MainWindow 物件會建立 Customer 物件，並顯示在視窗上，當使用者輸入資料後，將這些資料更新在 Customer 物件中。

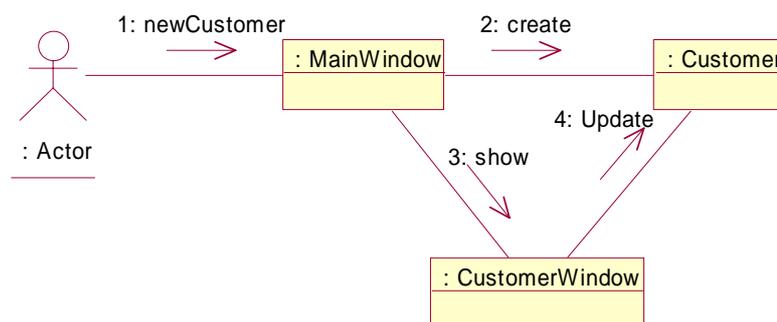


圖 2.9 合作圖

### 第3章 半導體與光電產業製程設備介紹

現今半導體製程設備依據型態來分大致可區分為三類：第一類是專用機型，如一般的離子佈植機、微影機等，在這方面目前仍沒有共通的硬體標準；第二類為自訂式集束型型態，乃是設備廠以自行訂定的標準所組成的集束型架構，此種架構主要是由應用材料公司(Applied Material Co.)所採用，由於應用材料公司設備之市場佔有率高，其衛星工廠有意願與應用材料公司配合生產晶圓製程模組，乃採用此架構；第三類則為開放式集束型型態，此為透過 SEMI 及全球半導體製造廠及設備廠共同參與討論表決所訂定的集束型製程設備硬體介面標準及軟體通訊協定標準，使提供不同模組的專業製造廠之介面標準化，並整合於一集束型製程設備內，本文所討論的集束型製程設備即屬此類型。

#### 3.1 半導體製程設備介紹

300mm 集束型半導體製程設備在十二吋晶圓廠的興建趨勢中已是未來的主流，如圖 3.1 所示，其相關的研究發展對未來的應用是相當重要的一環。

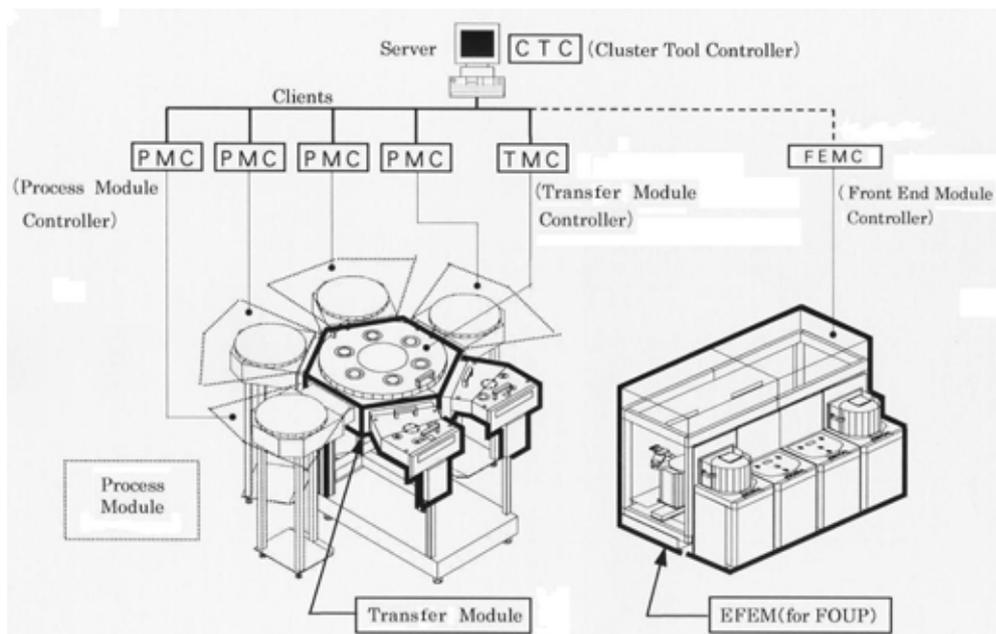


圖 3.1 集束型製程設備

實驗室內現有的集束型製程設備主要可分為兩個部份：由傳輸模組與製程模組所組成的**集束型製程設備模組(Cluster Tool Module)**，LAM Research(科林研發)所開發的Alliance 4.1；以及**前端處理模組(Front-End Module, FEM)**，我們以 PST 公司所開發的 Lot Sortors 來模擬 FEM 的功能，接下來將討論集束型製程設備的硬體與軟體架構。

### 3.1.1 集束型製程設備之硬體架構

集束型製程設備硬體包含集束型製程設備模組以及前端處理模組，如圖 3.2 所示。而集束型製程設備模組又分為傳輸模組(Transfer Module)與製程模組(Process Module)，以下為各個硬體模組的介紹：

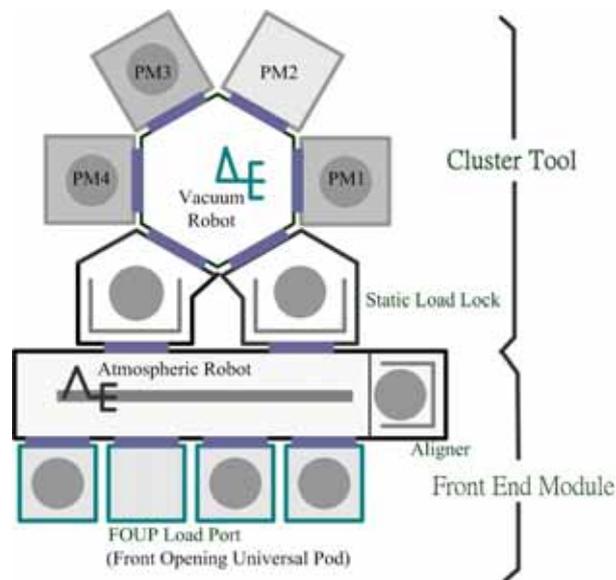


圖 3.2 300mm 集束型製程設備硬體示意圖

- **前端模組(Front End Module, FEM)**：前端模組內包含卡匣模組(FOUF)、大氣輸送模組(Atmospheric Robot)、晶圓定位處理模組(Aligner)。其主要的功能為接收無人搬運車上的晶圓盒，在接收到晶圓盒後將晶圓盒開啟，以大氣機械手臂將晶圓運送到前置定位器中做晶圓的方向定位，在晶圓完成定位後，將晶圓送至傳輸模組的靜態承載室中，等待晶圓在集束型製程設備中完成各項製程加工後，再將晶圓由靜態承載室中取出，然後放置到晶圓盒中，並由無人搬運車帶走。

- **傳輸模組(Transfer Module, TM)**:可分為兩個部份，一為靜態承載室(Static Load Lock)，主要功能為暫時存放晶圓，作為前端模組與集束型製程設備模組間的緩衝區，如圖 3.3 所示。承載室內為固定式之儲存槽(Slots)，接受大氣機械手臂(Atmospheric Robot)與真空機械手臂(Vacuum Robot)抓取或置放晶圓。另一部份為由真空機械手臂與抽真空系統所組成的真空輸送腔體，如圖 3.4 所示，其主要功能是負責晶圓於靜態承載室與製程腔體間的傳輸。為了使晶圓不受污染，製程中傳輸腔體內必須保持在真空的狀態。



圖 3.3 傳輸腔體晶圓承載系統

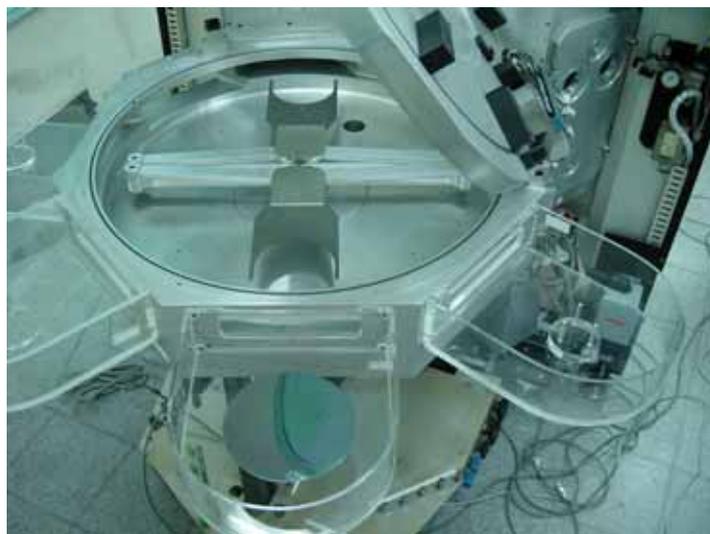


圖 3.4 傳輸腔體真空機械手臂系統

- **製程模組(Process Module, PM)**：掛載於真空輸送腔體上，如圖 3.5 所示，以隔離閘門和傳輸腔體之真空環境隔離，主要功能為執行晶圓製程處理，例如：化學氣相沈積(Chemical Vapor Deposition, CVD)、快速熱處理(Rapid Thermal Processing, RTP)、化學機械研磨(Chemical Mechanical Polishing, CMP)...等製程處理。由於製程模組和真空輸送模組間是以標準化介面做連結，因此對集束型製程設備而言，可針對不同的製程需求選用不同的製程模組，使得製程處理功能有很大的彈性空間。而集束型製程模組主要在接收前端模組送達的晶圓，並將這些晶圓做各項製程處理，完成處理後放到輸出靜態承載室中等待前端模組將晶圓從晶圓盒中取出。



圖 3.5 製程模組系統(RTP)

### 3.1.2 集束型製程設備之軟體架構

3.1.1 小節所介紹的各個硬體模組皆有相對應的控制器在操作，而這些控制器又由集束型製程設備控制器(Cluster Tools Controller, CTC)所管理與控制。由圖 3.6 可看出整個集束型製程設備的軟體架構，由最上層HOST端的Intra-bay Controller透過上層主機通訊介面(GEM)[7]與CTC做溝通，CTC透過HSMS、SECS-II通訊介面指揮傳輸模組控制器(Transfer Module Controller, TMC)、製程模組控制器(Process Module Controller, PMC)與前端模組控制器(Front-End Module Controller, FEMC)完成晶圓的搬運與製程工作。為了能夠彈性的更換或附加製程控制器，因此統一了通訊的介面使得整合上更加方便，並利用TCP/IP(Ethernet)方式來傳遞訊息，增加傳輸的效率。同時將各模組控制器安裝在不同的電腦上，避免系統因為個別模組控制器的錯誤而當機，可以減輕CTC運算和執行上的負擔，進而增加整體的效能。

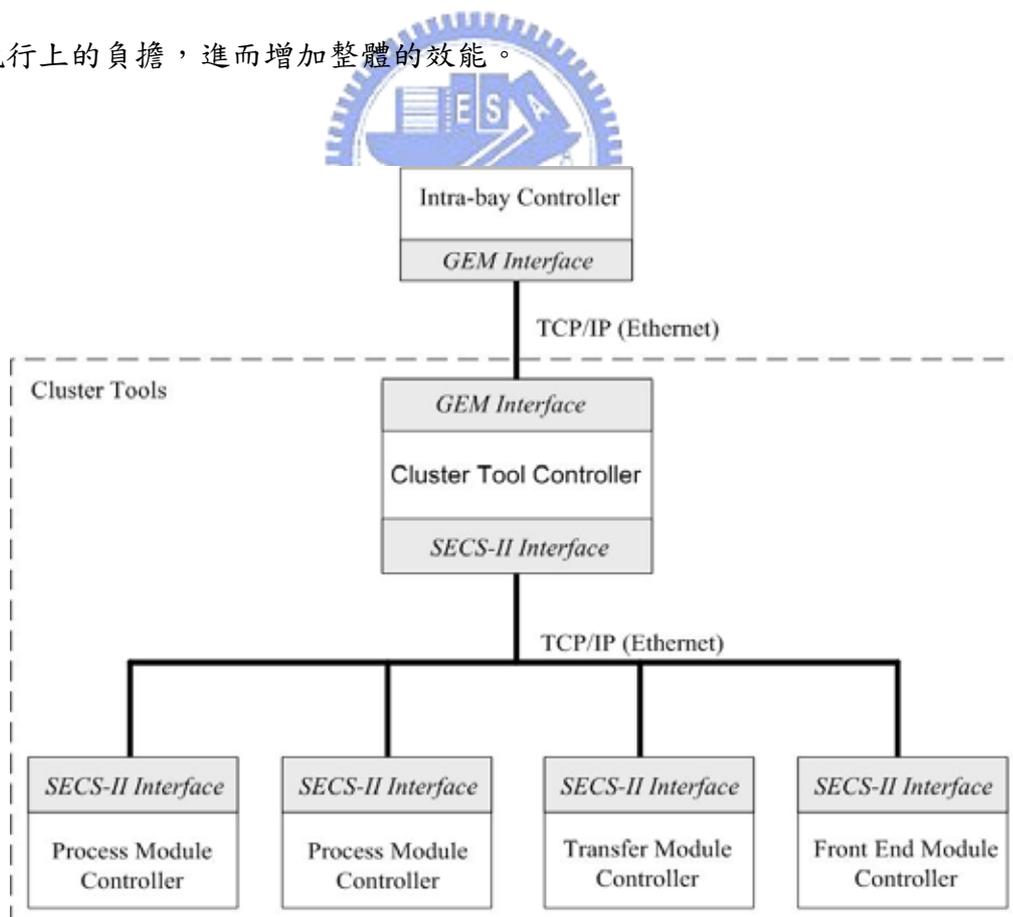


圖 3.6 300mm 集束型製程設備控制器軟體架構圖

## 3.2 光電產業製程設備介紹

光電產業是光、機、電相關技術的整合應用，近年來我國電腦、光電產業蓬勃發展，已成為世界高科技產品之重要生產基地，然而相關的生產設備、關鍵零組件、模組及次系統的技術發展仍未臻成熟。TFT-LCD 生產設備包含製造 LCD 面板與彩色濾光片的設備，可分為基板製程關連設備、面板與模組構裝關連設備及檢查與修整設備三大類。由於半導體與光電製程設備必須在無塵室內操作，所以設備所使用的元件必須能適用於無塵室內的高潔淨度環境，本節說明 TFT-LCD 陣列製程薄膜電晶體成膜設備[15]之硬體與軟體架構，目的是希望未來能將通訊模組相關的研究發展應用在此類的製程設備上。

### 3.2.1 TFT-LCD 陣列製程設備硬體架構

成膜製程中的濺鍍成膜法是應用於成長閘電極、源電極、汲電極、掃描線、儲存電容電極以及信號線等功能性金屬薄膜。濺鍍裝置的型態有所謂的第一代將基板固定於大型圓盤，而縱向傳送的分批式處理方法(In-Line Batch)，此一方法已成為主流的成膜技術，在長方形陰極的表面上，以一定速度濺鍍而形成所須的薄膜於基板上。至於第二代裝置則與矽半導體積體電路製造技術相近似的單片集束型處理方法(Cluster)，而此一裝置可使製程設備趨向小型化。

圖 3.7 所示為應用於 TFT-LCD 製程中 Sputtering 的陣列製程設備，其基本架構跟應用於半導體的集束型製程設備相似。兩者比較最大的差別為：在半導體製程中製程環境的潔淨度要求非常高；但在 TFT-LCD 的製程中，必須避免操作時所產生的靜電問題，減少因靜電造成的品質不良，如此才可提升整體製程的良率。

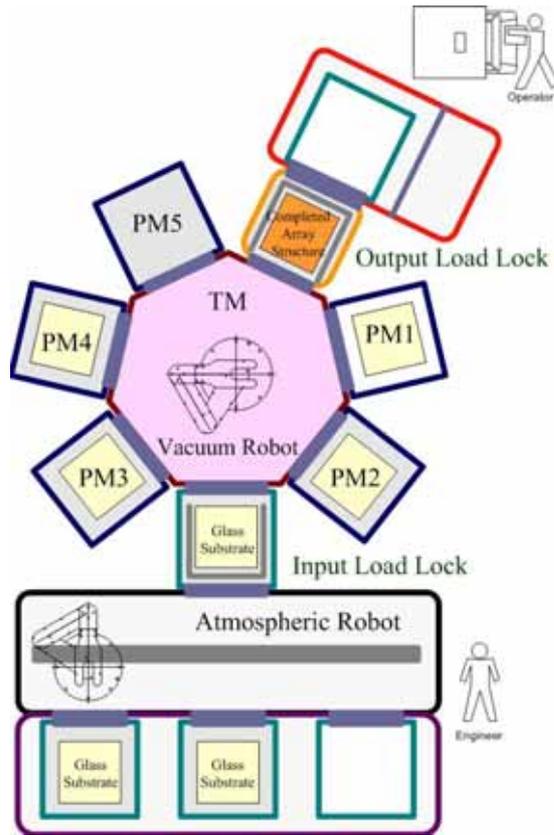


圖 3.7 TFT-LCD 陣列製程設備

### 3.2.2 TFT-LCD 陣列製程設備軟體架構

如圖 3.8 所示是本文所規劃應用在 TFT-LCD 製程設備的控制器軟體架構，由最上層的 HOST 端透過 HSMS 與 SECS-II 的介面與 SECS 控制器溝通，而 SECS 控制器透過 Socket 介面，指揮設備端的控制器，完成玻璃基板的搬運動作。為了能夠彈性的更換設備控制器，因此以統一的通訊介面加以整合，並利用 TCP/IP 或乙太網路(Ethernet)來傳遞訊息，可以增加傳輸的效率，進而增進整體的效能。

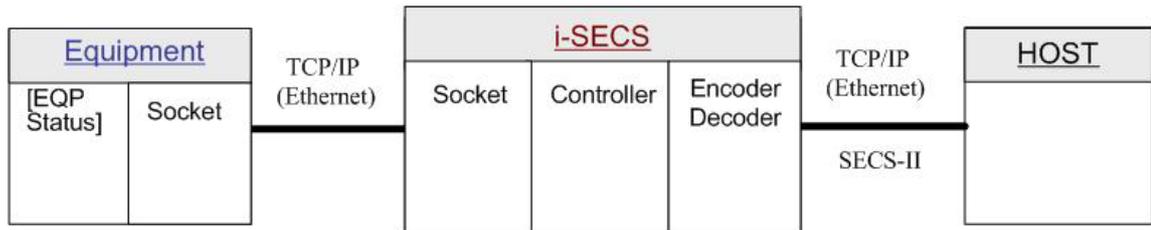


圖 3.8 光電產業製程設備控制器軟體架構圖

### 3.3 通訊模組於半導體與光電製程設備上之應用

通訊模組的開發是依照 SEMI Standard 所訂定的 HSMS 與 SECS-II 標準來實作。一般來說，因為半導體廠與設備製造大廠皆已走向 SEMI Standard 的規範，因此通訊模組在半導體製程設備上的應用相當廣泛與普及。而我們在通訊模組內所建立的訊息物件，在半導體的領域上都是共通的，不會因為製程機台不同而變更 SECS-II 訊息物件的架構，因此我們可以將通訊模組廣泛運用在半導體製程設備上，達到設備自動化的目的。

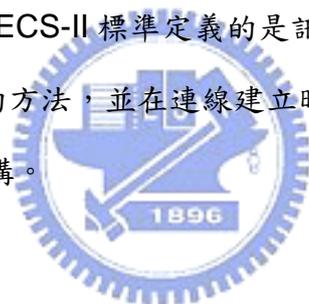
在光電產業的製程設備上，由於目前沒有統一的標準介面，因此形成了各家光電大廠或設備製造商自訂規格的情況。目前通訊模組在光電產業的應用上會有相當大的差異，主要的規格都是由市佔率大的廠商自訂，如：IBM、友達、奇美...等。「自訂通訊訊息規範」是目前各光電大廠在取得標準與量產時程上折衷的替代方案。為了要在競爭快速的光電領域取得優勢，因此很多廠商都藉由自訂通訊訊息規範來取得量產上的優勢，並且保有技術上的 Know-How。對不同的機台，可能使用不同的通訊訊息物件，因此我們發展的通訊模組必須能夠隨機台的不同而彈性擴增訊息物件，並且以讓設備可以透過通訊模組來傳遞符合 SEMI Standard 的訊息為目標。

目前，通訊模組在半導體產業的應用上已相當廣泛；未來，光電產業在經歷快速的擴張發展後勢將走向一個共通標準，在共通的使用標準與介面建立後，通訊模組可以擴展到光電領域的製程設備上，未來通訊模組在光電產業製程設備的領域上仍有相當大的發展空間。後面幾章我們就通訊模組的內容來做介紹，並規劃了使用者介面，希望能夠在兩個不同的領域中歸納出共通的特點，達到通訊的目的。

## 第4章 通訊模組之設計與實作

SECS(SEMI Equipment Communication Standard)通訊協定是由SEMI所制訂，其目的為使製程設備與上層主機電腦控制器之間能遵照一定的通訊介面溝通，當製程設備更換時，因為有共同的通訊介面，設備間得以繼續以SECS通訊介面互相溝通，不需重新設計架構，使系統擴充性增加，減少系統開發的時程。

通訊模組[4]的作用在於交換控制器與控制器之間的訊息，達到各控制模組間訊息溝通的目的，本論文所研究的通訊模組是利用 HSMS 透過乙太網路(Ethernet)以 TCP/IP(Socket)的形式來進行訊息的交握(Handshake)，而訊息的內容則依據 SEMI Standard 編寫成 SECS-II 所規範的格式，SECS 通訊模組亦包含了建立連線與接收傳送訊息的功能。簡單來說，SECS-II 標準定義的是訊息資料的內容；而 HSMS 標準則定義了利用 TCP/IP 建立連線的方法，並在連線建立時所進行的訊息交握。本章將說明通訊模組的訊息規範與設計架構。



### 4.1 HSMS (High Speed SECS Message Services)介紹

SECS-I [2]定義了底層的通訊介面，透過 RS-232 的方式來傳遞資料；而 HSMS [6]則定義了利用 TCP/IP(Ethernet)傳遞訊息的通訊介面，它規範了如何建立連線與連線狀態的變化，也定義了訊息表頭(Message Header)，使遵照 HSMS 標準的主從式電腦兩端可藉由訊息表頭中的資訊了解訊息的種類。

#### 4.1.1 HSMS 訊息格式

本節主要說明 HSMS 所定義的訊息格式，HSMS 的訊息可以分為兩種：控制訊息(Control Message)和資料訊息(Data Message)。控制訊息用來控制系統的通訊狀態，

資料訊息則是用來進行真實資料的交換。如圖 4.1 所示為 HSMS 訊息的格式，格式內說明了一個完整的訊息需包含四個位元組的訊息長度(Message Length)，與十個位元組的訊息表頭(Message Header)，表頭內說明此訊息為控制訊息或資料訊息。若為控制訊息，則訊息長度為 14 個位元組且無訊息內容；若為資料訊息，長度必大於或等於 14 個位元組。表 4.1 所示為 HSMS 訊息格式說明。表 4.2 則說明了 HSMS 所定義訊息表頭的內容。

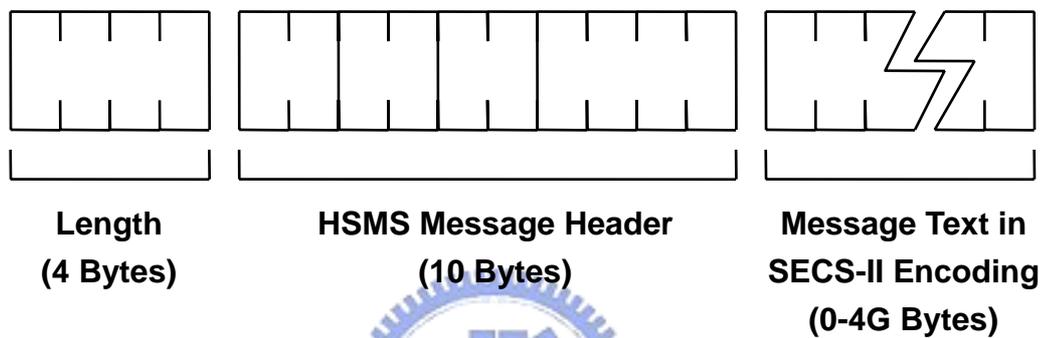


圖 4.1 HSMS 訊息格式

表 4.1 HSMS 訊息格式說明

| 位元組數 | 說明                                     |
|------|--|
| 4    | 以 4 個位元組表示訊息的長度，0x0000000D 表示此訊息長度為 13 |
| 10   | 表示訊息表頭的位元組數                            |
| 0~n  | 訊息內容，須遵照 SECS-II 所規範的訊息架構              |

表 4.2 HSMS Message Header

| 位元組編號 | 說明                                     |
|-------|--|
| 0~1   | 設備編號，為兩個位元組的正整數                        |
| 2     | 若為控制訊息，則為 0；若為資料訊息，則為 W-bit 與 StreamID |
| 3     | 若為控制訊息，則為 0 或狀態值；若為資料訊息，則為 FunctionID  |
| 4     | PType，遵照 SECS-II，所以值為 0                |
| 5     | SType，依各個不同的訊息而有不同的值，請參考表 4.3          |
| 6~9   | 系統位元組                                  |

資料訊息其訊息表頭的 0~1 位元組為設備編號，依據 SEMI(E37.1) HSMS-SS 標準，所有訊息表頭的設備編號皆為 0xFFFF 值。第 2 位元組則用來表示此資料訊息是否須回傳(即此資料訊息為主要訊息)及資料訊息的 StreamID。而第 3 位元組則表示此資料訊息的 FunctionID，藉由這兩個位元組，即可判斷此資料訊息為何。第 4 位元組代表 PType，其值為零代表此資料訊息是遵照 SECS-II 的訊息架構，而第 5 位元組 SType 為零，代表此訊息為資料訊息；若 SType 為 1-9，則為對應的控制訊息。第 6-9 位元組為系統位元組，用來分辨訊息，每個訊息都有獨特的系統位元組，如此才能分辨訊息與訊息間的差異。在資料訊息中，回傳訊息的系統位元組必須與主要訊息的系統位元組相同；在控制訊息中，回應訊息的系統位元組也必須與要求訊息的系統位元組相同。要分辨控制訊息為何種訊息，一般都以 SType 位元組來分辨，而 PType 位元組皆為零。但控制訊息中的回傳訊息，其第三位元組代表著此控制訊息的狀態值，Reject 訊息則是表示理由碼。Linktest 控制訊息其設備編號為 0xFFFF 值。

表 4.3 HSMS 訊息格式比較

| Message Type | Message Header                 |                |                 |              |              |   | Message Text |
|--------------|--------------------------------|----------------|-----------------|--------------|--------------|---|--------------|
|              | Byte 0-1 SessionID             | Byte 2         | Byte 3          | Byte 4 PType | Byte 5 SType | Byte 6-9 System Bytes                       |              |
| DataMessage  | 自訂                             | W-bit、StreamID | FunctionID      | 0            | 0            | Primary : Unique<br>Reply : Same as primary | 遵照 SECS-II   |
| Select.req   | 自訂                             | 0              | 0               | 0            | 1            | Unique                                      | none         |
| Select.rsp   | Same as req.                   | 0              | Select Status   | 0            | 2            | Same as req.                                | none         |
| Deselect.req | 自訂                             | 0              | 0               | 0            | 3            | Unique                                      | none         |
| Deselect.rsp | Same as req.                   | 0              | Deselect Status | 0            | 4            | Same as req.                                | none         |
| Linktest.req | 0xFFFF                         | 0              | 0               | 0            | 5            | Unique                                      | none         |
| Linktest.rsp | 0xFFFF                         | 0              | 0               | 0            | 6            | Same as req.                                | none         |
| Reject.req   | Same as message being rejected | 註 <sup>1</sup> | Reason Code     | 0            | 7            | Same as message being rejected              | none         |
| Separate.req | 自訂                             | 0              | 0               | 0            | 9            | Unique                                      | none         |

<sup>1</sup>若因PType錯誤而被回絕，則值為被回絕訊息的PType值；若因其他原因被回絕，則值為被回絕訊息的SType值。

#### 4.1.2 HSMS 的逾時機制

HSMS 所定義的逾時(Timeout)機制一共有五種，分別為 T3、T5、T6、T7、T8，下面分別就此五種不同的 Timeout 機制來介紹：

1. T3(Reply Timeout)：T3 主要是針對資料訊息傳遞時，傳送端送出主要訊息後等待接收端回傳訊息的時間，一般建議值為 45 秒。
2. T5(Separation Timeout)：當 TCP/IP 連線失效後，主動端實體(Active Entity)至少須等待 T5 的時間才能再次進行連線，一般建議值為 10 秒。
3. T6(Control Timeout)：T6 與 T3 類似，T3 是應用在資料訊息傳遞上；而 T6 則是應用在控制訊息的傳遞上。T6 定義為當傳送端送出\*.req 訊息後，等待接收端回傳\*.rsp 所需的時間，一般建議值為 5 秒。
4. T7(Not Selected Timeout)：當連線狀態進入“Connected State”後，等待進入“Selected State”所需的時間，一般建議值為 10 秒。
5. T8(Inter-Character Timeout)：當接收端接收到訊息時，每一個訊息內字元間所能允許的最大間隔時間，一般建議值為 5 秒。

#### 4.1.3 HSMS 連線

如圖 4.2 所示，我們以 UML 的狀態圖來說明 HSMS 建立連線、傳送訊息、中斷連線之間的狀態變化。

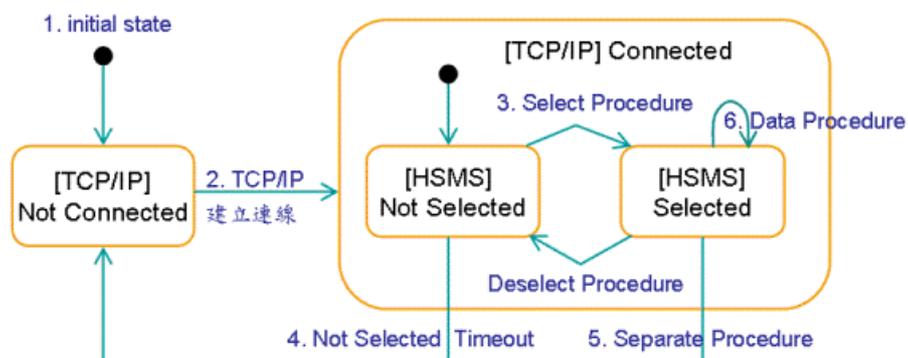


圖 4.2 HSMS 狀態圖

在建立連線時，我們將連線兩端分為被動端模式(Passive Mode)，與主動端模式(Active Mode)，並分別加以討論。

#### 4.1.3.1 被動端模式(Passive Mode)

1. 被動端先建立一個 Server Socket，並等待主動端 Client Socket 的連線，此時狀態為[TCP/IP] NOT CONNECTED。
2. 當被動端收到主動端 Client Socket 的連線後，accept 對方的連線，並將狀態由[TCP/IP] NOT CONNECTED 進入到[TCP/IP] CONNECTED 狀態及[HSMS] NOT SELECTED 子狀態，並開始 T7(Not Selected Timeout)的逾時器計時。
3. 當收到主動端的 Select.req 後，回傳 Select.rsp，完成 Select 程序。並取消 T7 的逾時器計時，子狀態進入[HSMS] SELECTED 狀態。
4. 當 T7 逾時器時間超過 T7 所限制的時間且未收到主動端的 Select.req 時，則中斷連線，狀態由 [HSMS] NOT SELECTED 子狀態進入 [TCP/IP] NOT CONNECTED 狀態。
5. 當狀態在 [HSMS] SELECTED 子狀態下時，進行 Separate 程序，收到 Separate.req 後立即中斷連線，此時狀態進入[TCP/IP] NOT CONNECTED 狀態。
6. 當狀態在[HSMS] SELECTED 子狀態下時，才可以進行資料訊息的傳遞程序，其程序就如 SECS-II 所規範的主要訊息與回傳訊息的訊息交握，如：當收到 S6F11 的訊息時則回傳 S6F12 的訊息。

#### 4.1.3.2 主動端模式(Active Mode)

1. 主動端進行初始化並建立 Client Socket 的連線，此時狀態為[TCP/IP] NOT CONNECTED。
2. 連線至被動端，連線成功後，傳送 Select.req 給被動端，等待被動端回傳 Select.rsp，並開始 T6(Control Timeout)的逾時器計時，此時狀態由 [TCP/IP] NOT CONNECTED 進入到 [TCP/IP] CONNECTED 狀態及 [HSMS] NOT SELECTED 子狀態。
3. 當收到被動端的 Select.rsp，且 Select.rsp 狀態值為零，完成 Select 程序。並取消 T6 逾時器計時，子狀態進入[HSMS] SELECTED 狀態。
4. 當 T6 逾時器時間超過 T6 所限制的時間且未收到被動端的 Select.rsp 時，則中斷連線，此時狀態由[HSMS] NOT SELECTED 子狀態進入[TCP/IP] NOT CONNECTED 狀態，並開始 T5(Separation Timeout)的逾時器計時。
5. 當狀態在[HSMS] SELECTED 子狀態下時，進行 Separate 程序，收到 Separate 要求後立即中斷連線，此時狀態進入[TCP/IP] NOT CONNECTED 狀態。
6. 當狀態在[HSMS] SELECTED 子狀態下時，才可以進行資料訊息的傳輸程序，其程序就如 SECS-II 所規範的主要訊息與回傳訊息的訊息交握，如：當收到 S6F11 的訊息時則回傳 S6F12 的訊息。

#### 4.1.4 HSMS 訊息交握程序

圖 4.3 及圖 4.4 說明 HSMS 所定義的主動端與被動端之間各個訊息交握程序的循序圖。

1. 系統初始化，被動端建立一個 Server Socket 物件並等待主動端建立 Client Socket 的物件的連線。
2. 當連線建立完成後，由主動端向被動端送出 Select.req，等待被動端收到後，回傳 Select.rsp，完成整個 Select 程序。由於 Select 程序是控制訊息，所以在要求訊息與回傳訊息之間有 T6 的逾時限制。
3. 當 Select 程序完成後，狀態即進入[HSMS] SELECTED 的狀態，依據 HSMS 的規範必須在此狀態下才可進行資料的傳輸，而此時資料傳遞程序時會有 T3 的逾時限制。
4. 當狀態在[HSMS] SELECTED 的狀態下時，可以透過 Deselect 程序將狀態更改為[HSMS] NOT SELECTED 的狀態。在此狀態下，雖然 TCP/IP 連線依舊存在，但已不可進行資料的傳輸。由於 Deselect 程序是控制訊息，所以在要求訊息與回傳訊息之間有 T6 的逾時限制。
5. 當 TCP/IP 連線存在時，也就是[HSMS] SELECTED 與[HSMS] NOT SELECTED 的狀態下，隨時都可以進行 Linktest 程序，測試系統連線的狀況。由於 Linktest 程序是控制訊息，所以在要求訊息與回傳訊息之間有 T6 的逾時限制。
6. 當狀態在[HSMS] SELECTED 的狀態下時，可以透過 Separate 程序將 TCP/IP 連線直接中斷，此時狀態變為[TCP/IP] NOT CONNECTED。因為接收到 Separate.req 要求是不需要回傳訊息的，因此直接中斷連線。
7. 當收到不適當的訊息時，接收端回傳 Reject.req 訊息，並由訊息中的理由碼 (Reason Code)，告訴傳送端是哪裡發生錯誤。

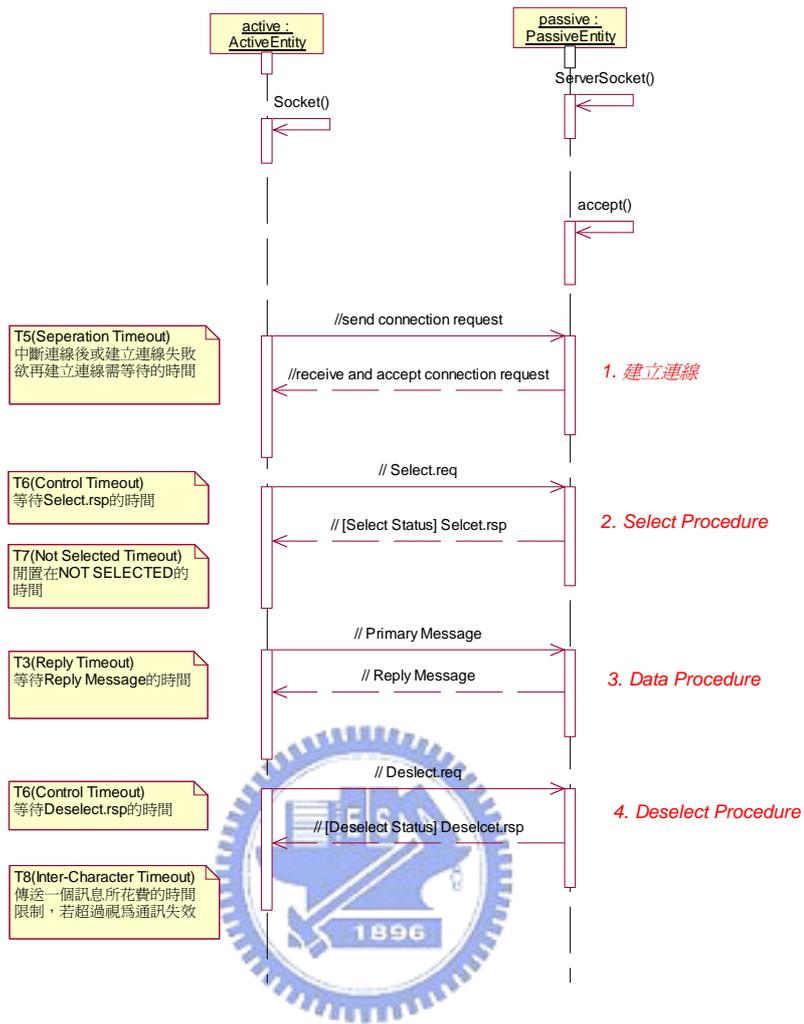


圖 4.3 HSMS 循序圖

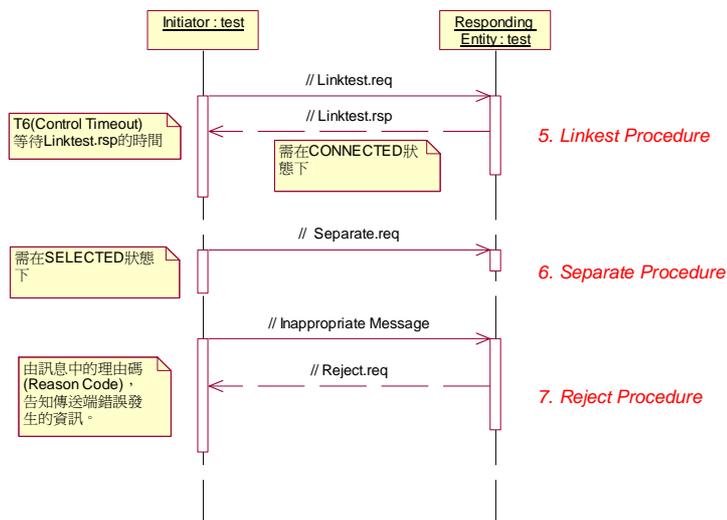


圖 4.4 HSMS 循序圖 (續)

## 4.2 SECS-II 訊息資料介紹

SECS-I 與 HSMS 分別是用 RS-232 與 TCP/IP 來傳遞訊息的通訊介面; SECS-II [5] 則定義了訊息資料的編碼格式。不論連線的方式是利用 SECS-I 或 HSMS，其後所掛的訊息資料皆須符合 SECS-II 的資料編碼格式。簡單來說，SECS-I 與 HSMS 可以想像成是我們常用的電話機，而 SECS-II 想像成在電話兩端互相溝通時所講的語言。

### 4.2.1 SECS-II 訊息資料結構

當傳送或接收到一個訊息時，首先我們必須先清楚整個訊息的架構，再由訊息架構中得到每個資料項目(Data Item)的資料型別為何。在 SECS-II 中每個 Data Item 的資料型別可以查閱 Data Item Dictionary，得到資料型別後找出相對應的 Format Code，如表 4.4 所示，如此才能完整的表示一個 SECS-II 所定義的訊息。

表 4.4 SECS-II 訊息的資料型別

| Format Code<br>(Bit 876543) | Octal | Meaning                  |
|-----------------------------|-------|--------------------------|
| 000000                      | 00    | LIST(length in elements) |
| 001000                      | 10    | Binary                   |
| 001001                      | 11    | Boolean                  |
| 010000                      | 20    | ASCII                    |
| 010001                      | 21    | JIS-8                    |
| 010010                      | 22    | 2-byte character         |
| 011000                      | 30    | 8-byte integer(signed)   |
| 011001                      | 31    | 1-byte integer(signed)   |
| 011010                      | 32    | 2-byte integer(signed)   |
| 011100                      | 34    | 4-byte integer(signed)   |
| 100000                      | 40    | 8-byte floating point    |
| 100100                      | 44    | 4-byte floating point    |
| 101000                      | 50    | 8-byte integer(unsigned) |
| 101001                      | 51    | 1-byte integer(unsigned) |
| 101010                      | 52    | 2-byte integer(unsigned) |
| 101100                      | 54    | 4-byte integer(unsigned) |

以 S4F27 訊息為例，由圖 4.5 可得知 S4F27 的訊息架構，再由 SECS-II 的 Data Item Dictionary 中可以得知 S4F27 訊息中的每個 Data Item 的資料型別及其相對應的 Format Code。得知這些資訊後，依照圖 4.6 的規範，可將 Data Item 的 Format Code 放在第一個位元組的 3~8 的位元上，而 1~2 的位元則是表示的接在第一個位元組後面有幾個位元組是用來表示 Data Item 值的長度，而這兩個位元所表示的值稱為 Item Length。當 Item Length 為 1，表示第 2 位元組的值為此 Data Item 值的位元組長度；若第 2 位元組的值為 4，則表示第 3 位元組到第 6 位元組的值為此 Data Item 的值。依此類推，所以一個 Data Item 最大可以表示到  $2^{24}$  的數值。

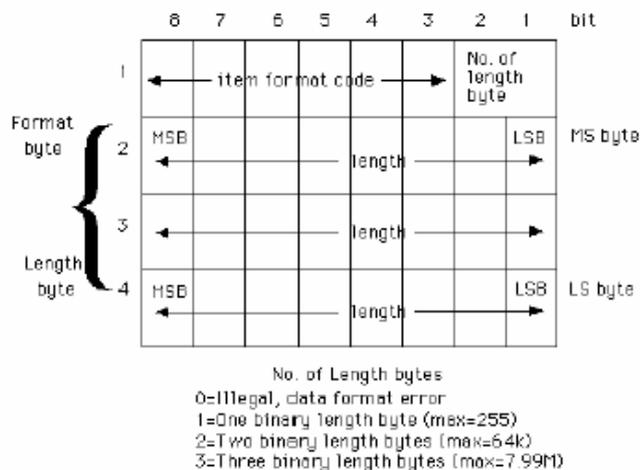
S4,F27 Handoff Ready (HR) S,P<—>S

Description: Each transfer partner informs the other when they are ready to perform a specified atomic transfer. The TRLINK values from the two partners must match. The values contained in the atomic transfer specification pertain to the sender of the message (except where specified).

Structure: L,2

- 1.<EQNAME>
- 2.L,11 [Specification for atomic xfr]
  - 1.<TRLINK> [Atomic transfer identifier]
  - 2.<TRPORT> [Port to be used for transfer]
  - 3.<TROBJNAME> [Transfer object identifier]
  - 4.<TROBJTYPE> [Object type-what form the material is in]
  - 5.<TRROLE> [Role in transfer-primary/secondary]
  - 7.<TRPTNR> [Identifier of transfer partner]
  - 8.<TRPTPORT> [Partner's Port to be Used]
  - 9.<TRDIR> [Transfer direction-send or receive]
  - 10.<TRTYPE> [Active or Passive]
  - 11.<TRLOCATION> [Location to send/receive mt1]

圖 4.5 SECS-II S4F27 訊息架構



Length refers to the number of bytes except in a list format where it refers to the number of elements (items or lists) in the list.

圖 4.6 Data Item 與 List 的 Header

## 4.3 Design Patterns 介紹與應用

學習 Design Patterns[14][16]後，可以讓我們以另一個不同的觀點來重新評估已編寫的程式，讓軟體設計與開發時能具有可再利用(Reusable)與擴充容易(Extensible)的功能。要設計出可再利用的物件導向軟體時，必須先歸納出適當的物件，分解成大小合宜的類別，定義類別介面和繼承關係，並建立彼此之間的關連。許多物件導向系統裡面都有一再出現的類別互動模式，這些模式不僅解決了特定問題，同時也讓物件導向系統更富彈性、更具可再利用的性質。

利用 Java 語言撰寫程式時，常常會用到 Java 提供的類別函式庫，但是 Design Patterns 並不是類別函式庫，Design Patterns 比類別函式庫更具有一般性概念的味道，類別函式庫是用來當作零組件的程式；而 Design Patterns 則是構思要如何組合零組件、如何結合不同的零組件以發揮最大的功能。藉由過去成功的經驗引領出新的設計，不必再苦思解決方案，就能馬上把 Design Patterns 套用上去。下面一小節開始將介紹通訊模組內所使用到的 Design Patterns。



### 4.3.1 Façade Pattern

Façade Pattern 要求一個子系統的外部與其內部的通訊必須通過一個統一的門面 (Façade)物件進行。Façade Pattern 的門面類別將使用端與子系統的內部複雜性分隔開，使得使用端只需要與門面物件打交道。

使用 Façade Pattern 的時機：

1. **為複雜子系統提供一個簡單介面**—子系統往往因為不斷演化而變得越來越複雜，使用 Façade Pattern 可以使得子系統更具可複用性。
2. **子系統的獨立性**—引入 Façade Pattern 將一個子系統與它的使用端以及其它子系統分離，可以提高子系統的獨立性和可攜性。
3. **層級化結構**—在建構一個層級化的系統時，可以使用 Façade 定義系統每一層的

入口。如果層與層之間是相互依賴的，則可以限定它們僅通過 Facade 進行溝通，從而簡化層與層之間的依賴關係。

### 4.3.2 Bridge Pattern

Bridge Pattern 的目的是要將「功能的類別階層」和「實作的類別階層」分成兩個獨立的類別階層。如果此 2 類別階層區隔清楚，就可以分別擴充兩邊的類別階層。如果想新增功能的話，就在功能的類別階層新增類別。不需要修改實作的類別階層，所有新增增加的功能都可以利用「所有實作」來使用。

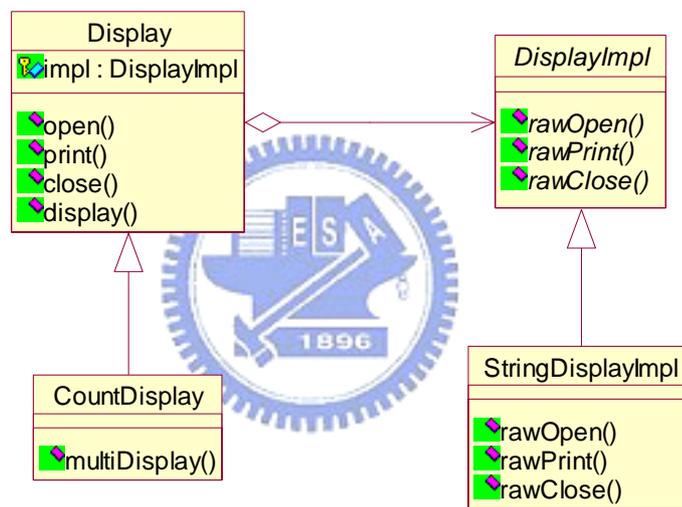


圖 4.7 Bridge Pattern 類別圖

### 4.3.3 Composite Pattern

對容器和內容物一視同仁，建立遞迴結構的 Design Pattern 就是 Composite Pattern。容器裡面可以是內容物，也可以是容器；而這個小一號的容器裡面還可以再放更小一號的容器...，如此可以建立出以大套小的遞迴結構。

在 SECS-II 中所定義的 List Item 就好比是一個容器；而每個 List Item 後面所接的 Data Item(如：Integer Item、Boolean Item...等)代表的則是內容物。List Item 內可以放

Data Item，也可以放 List Item。以圖 4.5 中的 S4F27 這個訊息架構來說，第一個 List Item 中放了兩個內容物，其一是名為 EQNAME 的 Data Item，其二又是一個 List Item，而此 List Item 內則放了 11 個 Data Item，分別是 TRLINK、TRPORT... 等。因此在 SECS-II 的訊息物件架構上採用了 Composite Pattern，可以使得在對訊息物件做處理時變得簡單許多。

圖 4.8 是參考[16]中的範例程式，利用電腦檔案系統中，資料夾與檔案之間的關係來說明 Composite Pattern 中各物件的關係。父類別 Entry 是一個抽象類別，由其子類別容器(Directory 類別)、內容(File 類別)來實作。由圖 4.8 的類別圖中可看出 Directory 類別與 Entry 類別有 aggregation 的關係，其表示 Directory 類別是由 Entry 物件的子類別—Directory 類別、File 類別所組成，而 Directory 類別又是 Entry 類別的子類別；也就是說，Directory 物件中可能又包含了 Directory 物件，容器中可能有小一號的容器的意思，而構成遞迴結構。

利用同樣的架構應用到 SECS-II 的訊息架構中，再配合下一節將介紹的 Visitor Pattern，可以使得整個通訊模組在訊息編譯與解譯上更加清晰。詳細應用將在 4.3.5 節中說明。

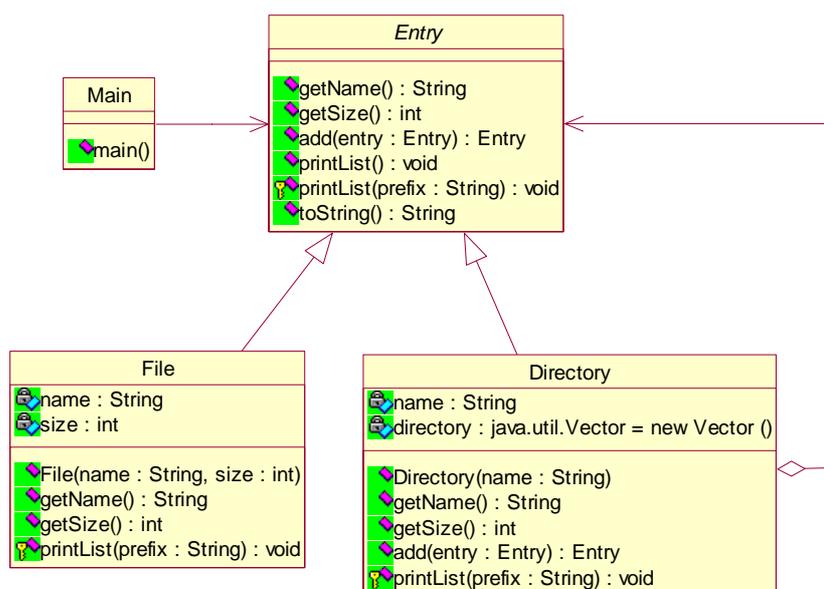


圖 4.8 Composite Pattern 類別圖

#### 4.3.4 Visitor Pattern

Visitor Pattern 是把「資料結構」和「處理」兩者分開的 Design Pattern。SECS-II 中所規範的訊息架構就是一種「資料結構」，而對訊息所做的編譯與解譯的動作就相當於是對這個資料結構所做的「處理」。若是能將「資料結構」和「處理」分開，當日後需要增加或是修改「處理」時，「資料結構」依舊保持不變，如此可以保證資料結構的完整性。

圖 4.9 中的 *NewClass\_Element* 類別與 *NewClass\_Visitor* 類別皆為抽象類別，透過子類別實作 *visitor()* 與 *accept()* 方法，達到「應用」能在「資料結構」中交互穿梭。

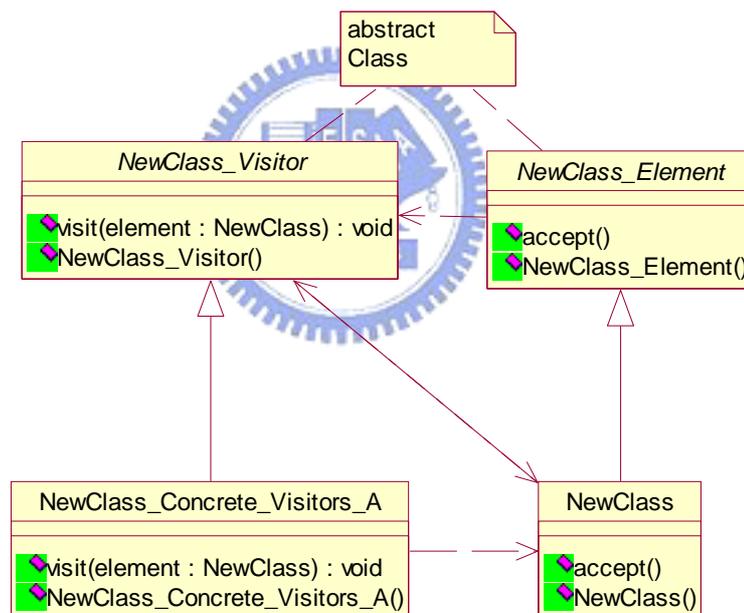


圖 4.9 Visitor Pattern 類別圖

### 4.3.5 Design Patterns 在通訊模組上之應用

Façade Pattern 在通訊模組內的應用可以參考 4.4.1 節的介紹，其目的是要提供控制器使用通訊模組的窗口，控制器需透過 Façade 類別才可存取通訊模組內的訊息物件。Bridge Pattern 在通訊模組的應用可以參考 5.2.3 節及圖 5.4 應用 Bridge Pattern 的時間調整類別圖，其目的是要把控制器程式的功能層與實作層分開，達到功能與實作擴充分開的目的。

本節主要說明 Composite Pattern 與 Visitor Pattern 在通訊模組上的應用。由 4.2.1 節中可知一個訊息是由 List 與 Data Item 所組成，而從表 4.4 中可得知每個訊息對應 List 與 Data Item 的 Format Code。通訊模組的訊息架構部份，就是利用訊息中的樹狀結構—List 與 Data Item 所組成而建立類似 Composite Pattern 的架構，如圖 4.10 右半部所示。Item 類別為抽象類別，List Item 物件則是一個由 Item 物件所構成的，所以從類別圖中可看出，List Item 類別與 Item 類別有 aggregation 的關係。依照表 4.4 所規範的資料型態分別建立相對應的資料型態類別，如：String Item、Integer Item、Boolean Item...等類別，而這些資料型態類別中都有一個表示 Format Code 的屬性，之後的應用也都是利用 Format Code 來分辨 Data Item 的資料型態。所以通訊模組中訊息物件的架構就是利用 Composite Pattern 來建立。

有了依照 Composite Pattern 架構所建立的訊息物件後，對於訊息的編譯 (Encoder)、解譯 (Decoder) 等「應用」，則可以使用 Visitor Pattern 來實作。如圖 4.10 左半部所示，有幾個應用就建立幾個 Visitor 物件，如：EncoderVisitor 類別是實作如何將訊息物件編譯為文字字串、Encoder2ByteVisitor 類別則是實作如何將訊息物件編譯為位元組、DecoderVisitor 類別則是實作如何將位元組解譯為訊息物件。

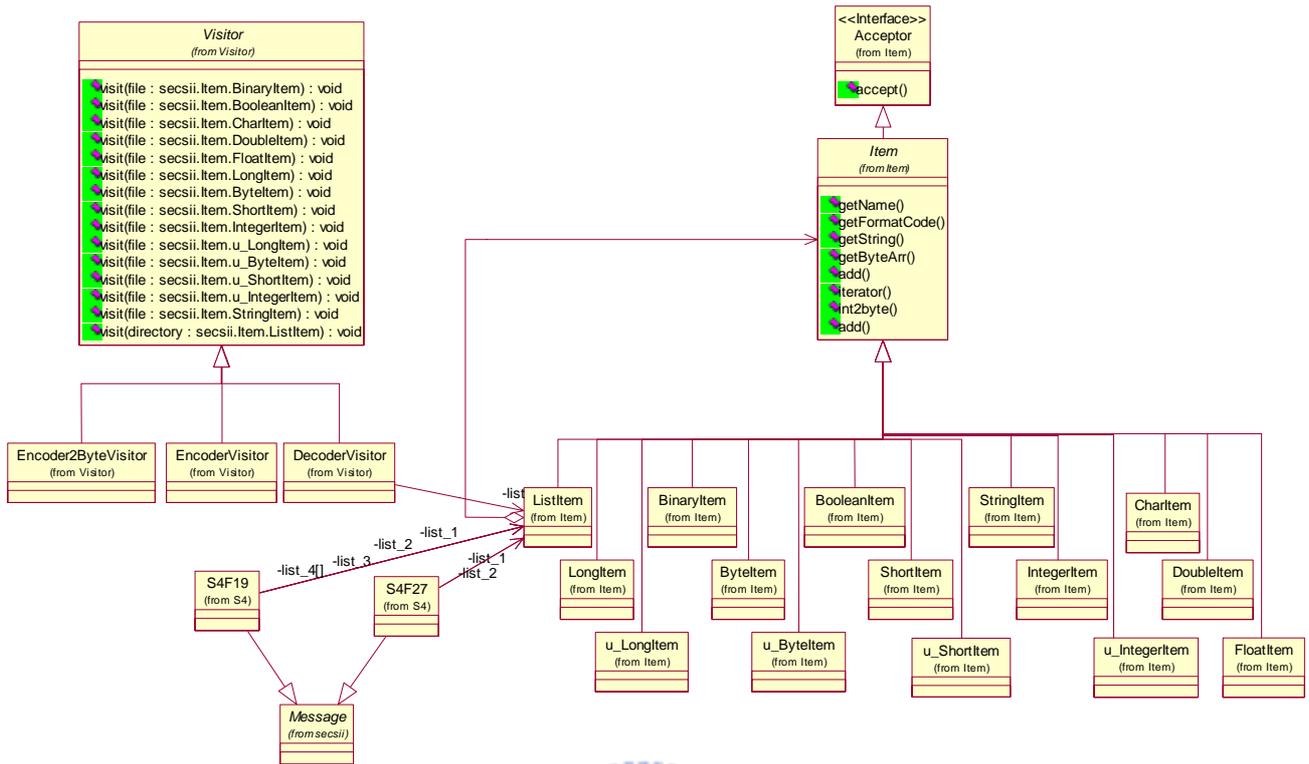


圖 4.10 應用 Design Patterns 的通訊模組類別圖



利用 Visitor Pattern 將資料結構與處理分開，一個 Visitor 物件可以對整個通訊模組的訊息物件做編譯與解譯，而不需要每個訊息物件都需去編寫一再重複的程式碼。對於編譯、解譯動作的修改，也只需更改 Visitor 物件即可，而不需要更動全部的訊息物件。因為使用了 Composite Pattern 的資料架構，可以建構出複雜的訊息架構，只要資料架構完成，Visitor 物件就可穿梭在資料架構中作動。

## 4.4 通訊模組核心程式系統架構

通訊模組核心程式提供了對訊息物件的讀取、設定、編譯與解譯的動作，同時也提供控制器與控制器之間 TCP/IP 連線建立以及訊息的傳送與接收功能。為了遵照 SECS-II 標準，每個訊息必須加上 HSMS 所定義的 Header 並串接 SECS-II 所定義的訊息格式後以位元組的方式傳送。當接受到訊息時，通訊模組將所收到的位元組解譯成相對應的訊息物件，並將訊息物件放入 MessageQueue 物件中，提供控制器讀取通訊模組所接收到的訊息物件。

### 4.4.1 通訊模組的靜態架構

圖 4.11 所示為通訊模組各個物件間的類別關係。表 4.5 則是通訊模組參與物件說明表，說明類別圖內所有參與的物件功能。

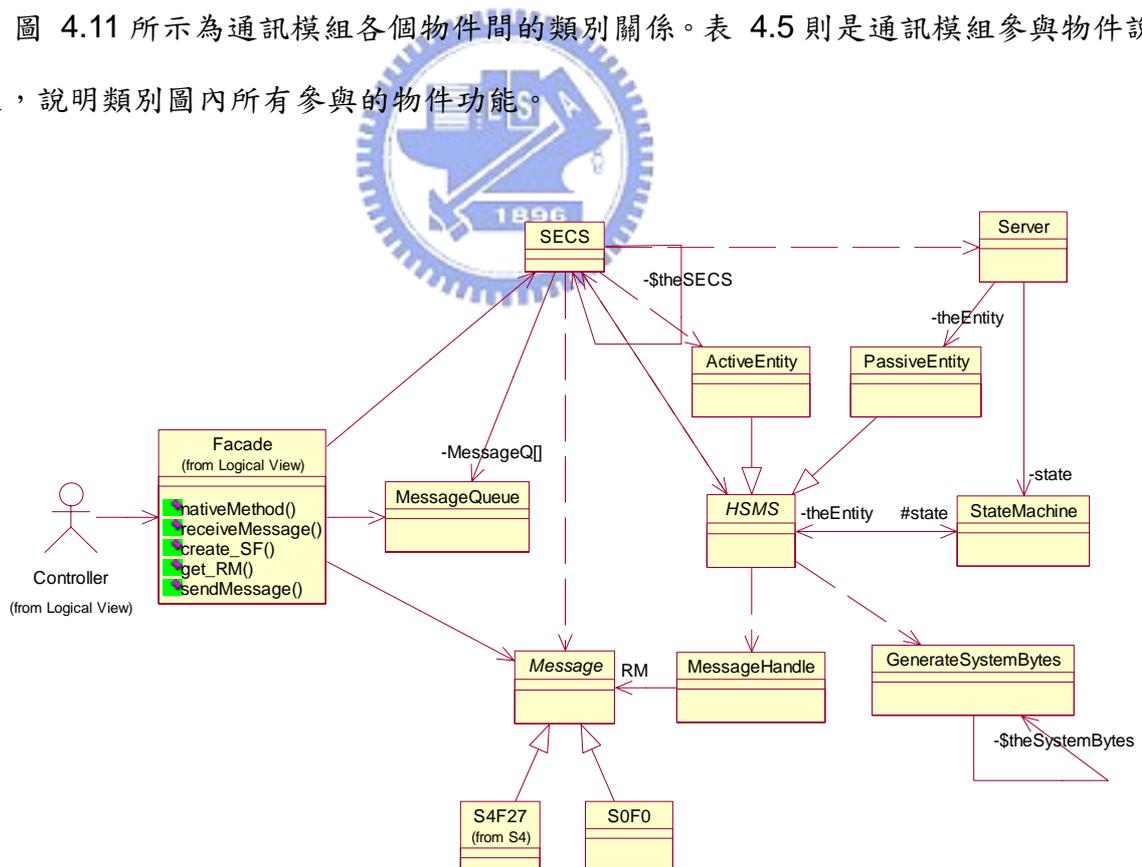


圖 4.11 SECS 通訊模組類別圖

表 4.5 通訊模組參與物件說明表

| 參與物件                | 說明  |
|---------------------|---|
| Façade              | 提供控制器使用通訊模組的窗口，控制器需透過 Façade 物件才可存取通訊模組內的訊息物件。  |
| SECS                | 提供了控制器初始化、建立連線、測試連線、傳送訊息、讀取訊息等方法。   |
| HSMS                | 為一抽象類別，主要實作 HSMS 所定義的訊息交握流程的動作及判斷。其子類別需實作 SendData()抽象方法。                                 |
| ActiveEntity        | 實作 HSMS 的子類別，為建立連線的主動方。   |
| PassiveEntity       | 實作 HSMS 的子類別，為建立連線的被動方。   |
| Server              | 實作 Server Thread，當控制器為 Passive Mode 時建立 Socket 後即交給 PassiveEntity 物件等待 Active 端的連線。       |
| StateMachine        | 遵照 HSMS 定義，紀錄每個連線 Socket 的狀態及實作逾時(Timeout)機制。   |
| MessageHandle       | 將所接收的訊息 Header 解譯並產生對應的訊息物件。  |
| MessageQueue        | 用來貯存已解譯完成的訊息物件，並提供方法給使用者讀取所貯存的訊息物件。   |
| Message             | 抽象類別，為所有訊息物件的父類別，提供所有訊息物件共通的屬性與方法。  |
| S4F27               | 為 SECS-II 所規範的訊息，以一個物件代表一個訊息，內含對其 Data Item 設定與讀取的方法。                                     |
| GenerateSystemBytes | HSMS 定義每個訊息的 Header 須有 SystemBytes 部份，且得是唯一，不可重複，因此，此類別為 Singleton 物件，專司 SystemBytes 的產生。 |

#### 4.4.2 訊息編譯循序圖

圖 4.12 說明了訊息物件 S4F19 的編譯流程與相關物件之間的互動。當使用者建立好一個訊息物件後，呼叫 SECS 物件的 SendMessage()方法將其傳送出去時，其訊息編譯步驟為：

1. SECS 物件呼叫訊息物件 S4F19 的 Encoder()。
2. 將每個 Data Item 依照 SECS-II 所規範的訊息架構，利用 ListItem 物件的 add() 方法將各個資料型態物件加入 ListItem 物件中。
3. 最上層的 ListItem 物件建立完成後，利用 EncoderVisitor 物件將 S4F19 訊息物件編譯為文字字串。
4. 執行最上層的 ListItem 物件的 accept()，而此方法接著會去呼叫 EncoderVisitor

物件的 Visitor()，在 Visitor() 中，會去將 ListItem 物件中的各個資料型態物件取出，並分別去執行其 accept()，各個資料型態物件的 accept() 又會去呼叫對應的 EncoderVisitor 物件的 Visitor()，利用這樣的結構，將所有的「處理」在 EncoderVisitor 物件的 Visitor() 中去實現，而達到 Visitor Pattern 的目的。

5. 最後再執行 EncoderVisitor 物件的 getString()，將所編譯的文字字串取出列印或傳送。

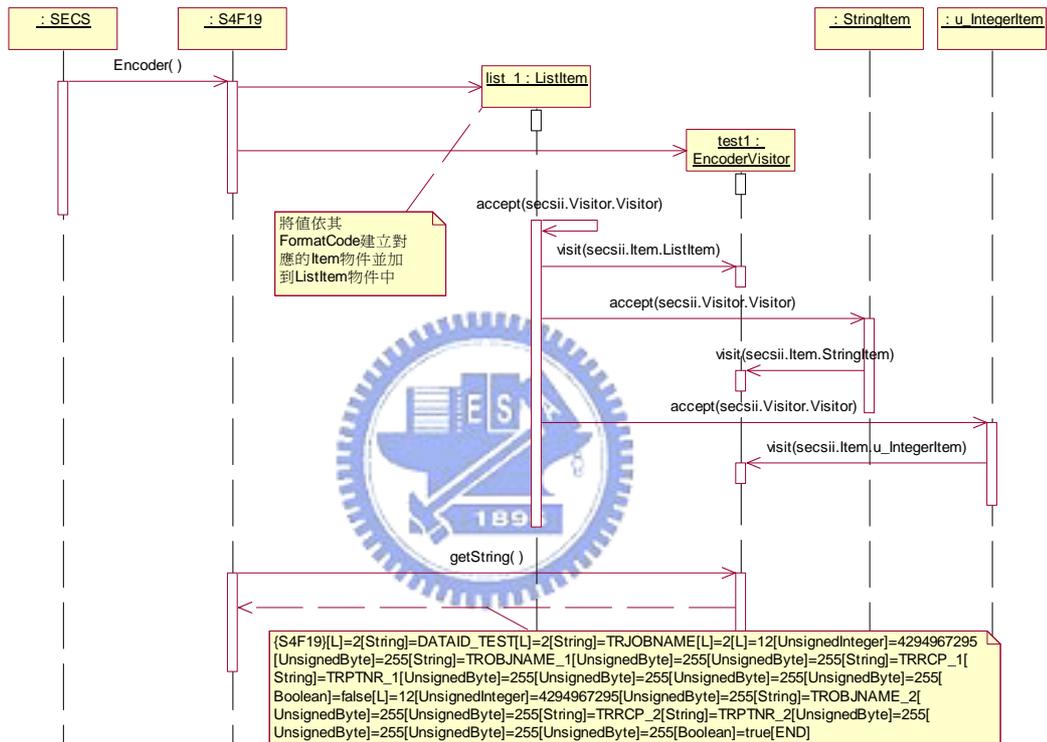


圖 4.12 S4F19 訊息編譯的循序圖

#### 4.4.3 訊息解譯循序圖

圖 4.13 說明了如何將所接收到的位元組訊息解譯為訊息物件，並將此訊息物件放入 MessageQueue 物件中的流程，其訊息解譯步驟為：

1. 將所接收到的位元組訊息傳入 MessageHandle 物件中的 Decoder() 執行，分析訊息 Header，找出對應的訊息類別，並建立此訊息物件。
2. 去掉位元組訊息的訊息 Header，將剩下的位元組訊息傳入所建立訊息物件的

Decoder() 中，執行解譯的動作。

3. 此訊息物件的 Decoder() 會建立一個 DecoderVisitor 物件，並將位元組訊息傳入此物件的 start() 中。
4. start() 依照 SECS-II 所規範的 Data Item 格式去解譯，並利用 Format Code 產生對應的資料型態物件。
5. 所產生的資料型態物件，利用 ListItem 物件的 add() 將各個資料型態物件加入 ListItem 物件中，並透過執行資料型態物件的 accept() 及 DecoderVisitor 物件的 visitor() 方法完成訊息物件的解譯工作。
6. 最後再將解譯出來得到的訊息物件放入 MessageQueue 中，等待控制器建立執行緒來將訊息物件取出。

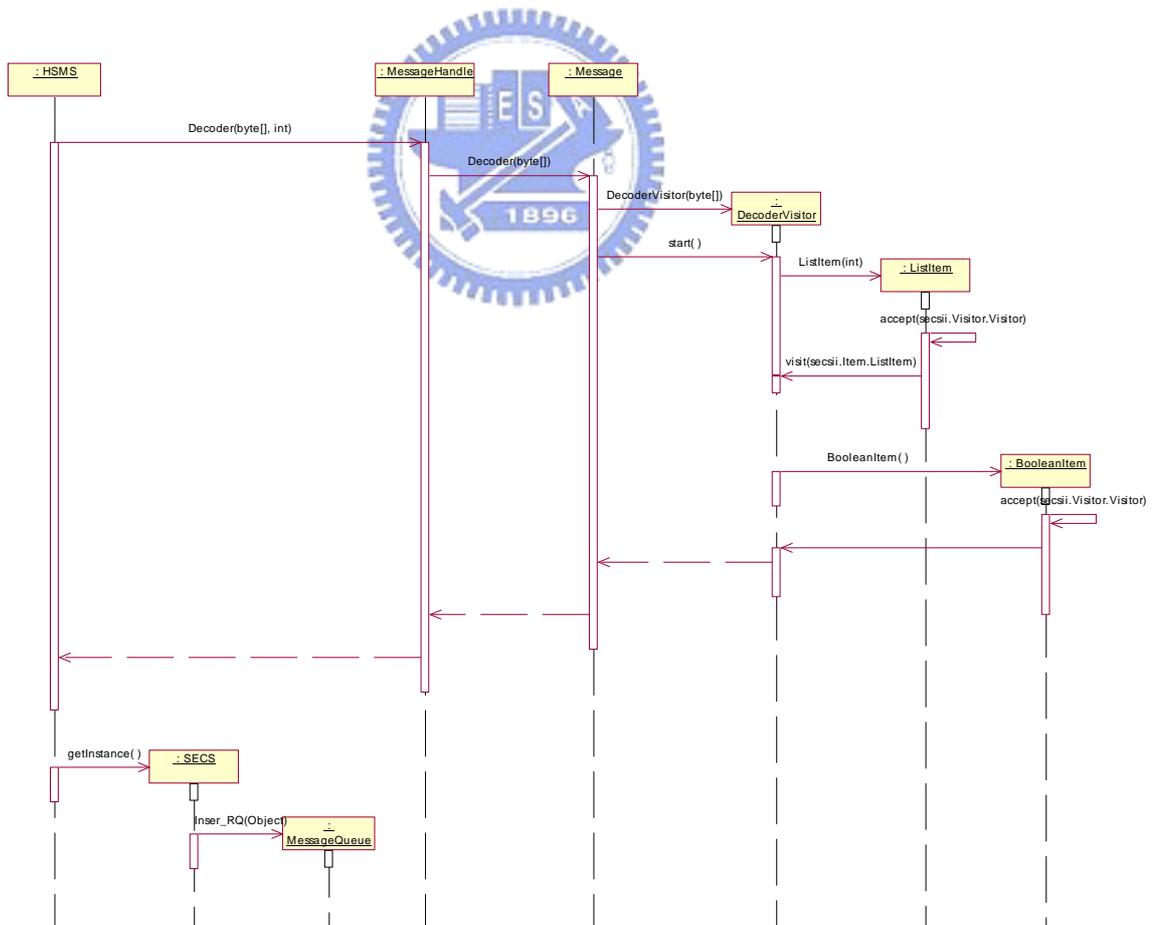


圖 4.13 訊息解譯的循序圖

#### 4.4.4 傳送訊息循序圖

圖 4.14 說明了使用者(Actor)如何建立一個訊息物件，並將其利用 SECS 物件傳送出去，傳送訊息的步驟為：

1. Actor 先建立一個 S4F19 訊息物件，並執行 `set_DATAID()`、`set_TRJOBNAME()` 等方法，設定訊息的 Data Item 值。
2. 接著 Actor 執行 SECS 物件的 `SendMessage()` 方法並傳入 S4F19 訊息物件。
3. SECS 物件會執行所傳入訊息物件的 `Encoder2Byte()`，將訊息物件編譯為位元組訊息。
4. SECS 物件執行 HSMS 物件的 `SendData()`，將位元組透過 Socket 傳送出去。

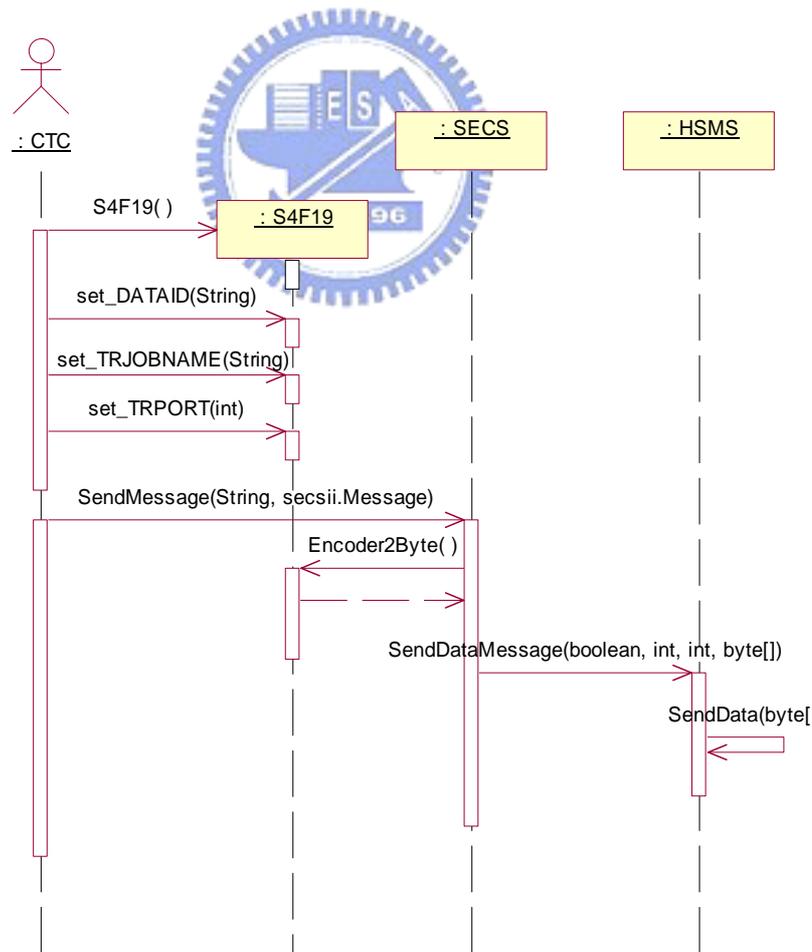


圖 4.14 傳送訊息循序圖

#### 4.4.5 接收訊息活動圖

圖 4.15 說明訊息接收的活動圖，所有經 Socket 傳送而接收的訊息，都會被解譯為訊息物件並存放在 MessageQueue 物件中，使用者(Actor)對於所接收到的訊息，只需要從 MessageQueue 物件中將訊息物件取出，其訊息接收的步驟為：

1. 建立一執行緒反覆檢查 MessageQueue 物件中是否有訊息物件。
2. 若無訊息物件，則繼續檢查；若有訊息物件，則將第一個訊息物件取出，並將其從 MessageQueue 物件中刪除。
3. 判斷所取出訊息物件的 Stream ID 及 Function ID，決定此訊息物件為何，使用者(Actor)才能對此訊息物件做處理。
4. 訊息處理完後，再次檢查 MessageQueue 物件中是否有訊息物件。

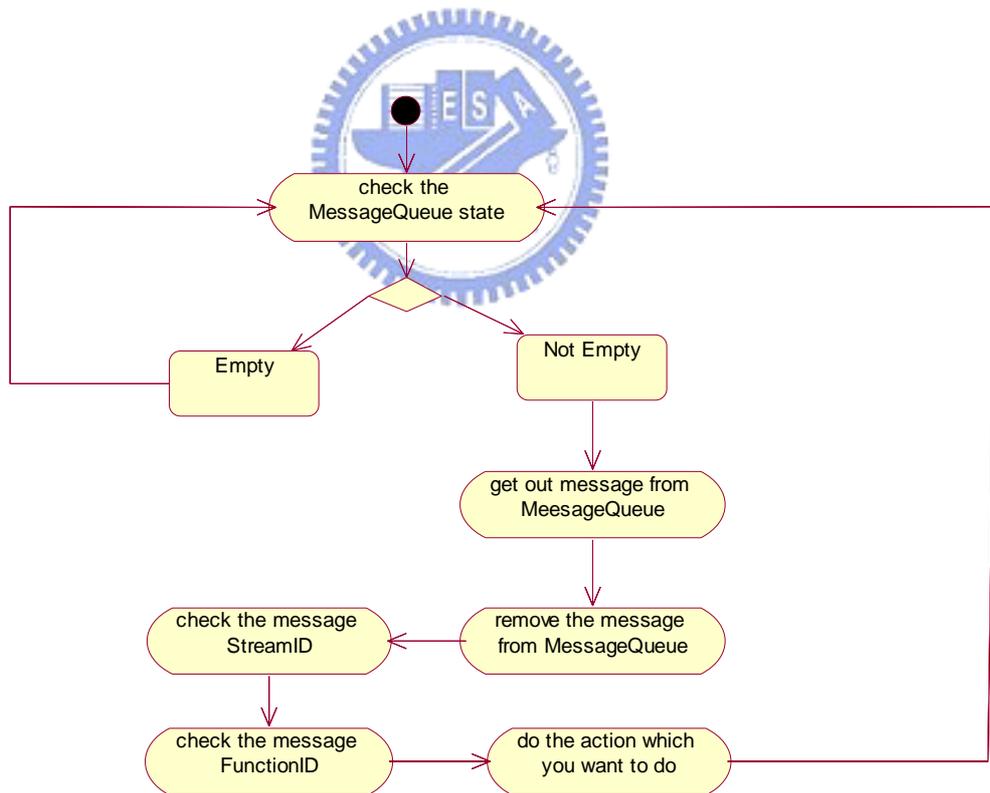


圖 4.15 接收訊息活動圖

## 4.5 SECS Message Language 介紹

SECS Message Language(SML)[17]是由GW Associates公司所發展用來描述SECS訊息的一種表示方法，SML的標記法類似於SEMI Standard SECS-II的標示規範，SML的優點是可以讓操作人員容易理解所傳遞的訊息內容，增加訊息物件的可讀性。本節將討論SML的標記方式與其在通訊模組內的應用。

### 4.5.1 SML 標記方式

在 SML 中，每個訊息內含的資料項目(Data Item)都可以表示成下列的形式：

$SxFy [W] < \text{型別} [\text{長度}] \text{數值} >$ .

表 4.6 所示介紹 SML 訊息的格式，並清楚解釋每一個符號在 SML 語法內所代表的意義。

表 4.6 SML 標示語法說明

| 符號   | 描述  |
|------|---|
| SxFy | 訊息的 Stream 與 Function Codes。  |
| W    | 判斷訊息是否須回傳，若 W 為 1 代表須回傳，0 表不須回傳。  |
| <>   | 每個資料項目都是由一對角括號所包圍，其內所表示的是 SECS-II 所規範資料項目的型別、長度與數值。                     |
| 型別   | SECS-II 所規範的資料項目格式，如表 4.4 所示，對照 SECS-II 在 SML 內的表示方式如表 4.7 所示。          |
| 長度   | 說明每個組成資料項目數值元件的數量。如：當型別是字串時，長度代表的是字串內字元的數量；若型別是整數時，length 代表的是陣列內數值的數量。 |
| 數值   | 依據型別，對應該型別所表示的數值內容。   |
| .    | 結束字元，每個訊息內只能存在一個結束字元。   |

當熟悉 SML 的表示法之後，我們實際以 SECS-II 訊息 S5F1 為例，從 S5F1 訊息的 SML 語法中可以得出訊息內掛了一個 List，該 List 內又掛了三個 Data Item，第一個 Data Item(參照表 4.7)的資料型別為 Binary，長度為 1，其值為 0x03；第二個 Data Item 資料型別為 2-byte unsigned integer，長度為 1，其值為 18；第三個 Data Item 資料型別為 ASCII，長度為 10，其值為"Fire Alarm"。

Example : S5F1

16 : 27 : 45.723 : Send S5F1(10000001)

S5F1 W

<L[3]

<B[1] 0x03>

<U2[1] 18>

<A[10] "Fire Alarm">

>

.



表 4.7 SML Data Item Format

| SML Data Item Format | Format Meaning           |
|----------------------|--------------------------|
| L                    | LIST(length in elements) |
| A                    | ASCII                    |
| B                    | Binary                   |
| Boolean              | True/False               |
| J                    | JIS-8                    |
| I1                   | 1-byte Integer(Signed)   |
| I2                   | 2-byte Integer(Signed)   |
| I4                   | 4-byte Integer(Signed)   |
| I8                   | 8-byte Integer(Signed)   |
| U1                   | 1-byte Integer(Unsigned) |
| U2                   | 2-byte Integer(Unsigned) |
| U4                   | 4-byte Integer(Unsigned) |
| U8                   | 8-byte Integer(Unsigned) |
| F4                   | 4-byte Floating Point    |
| F8                   | 8-byte Floating Point    |

## 4.5.2 SML 於通訊模組上的應用

本文所研究的通訊模組中，除了獨立的使用者介面之外，還必須實作 SML 使其具有支援 SML 語法的功能。內含 SML 功能的通訊模組須具備有 Screen Logging 和 Data Logging 的功能。所謂的 Screen Logging 指的是當通訊模組在傳送與接收訊息時，可以在使用者介面上將所收到的訊息以 SML 的表示方法呈現，目的是讓操作員可以立即的從螢幕上得到傳遞的訊息資訊，並做立即的處置；另一種功能為 Data Logging，指的是通訊模組可以把編成\*.sml 副檔名的 SML 檔讀入通訊模組內，操作員可以從使用者介面上選擇要送出的訊息，也可以將訊息傳遞過程中所紀錄下來的內容存成\*.log 檔，留作將來查閱的紀錄。

下圖 4.16 所示的是通訊模組中用來顯示 SML 訊息的類別圖，表 4.8 則說明了通訊模組中 SML 語法編譯與解譯的參與物件表。

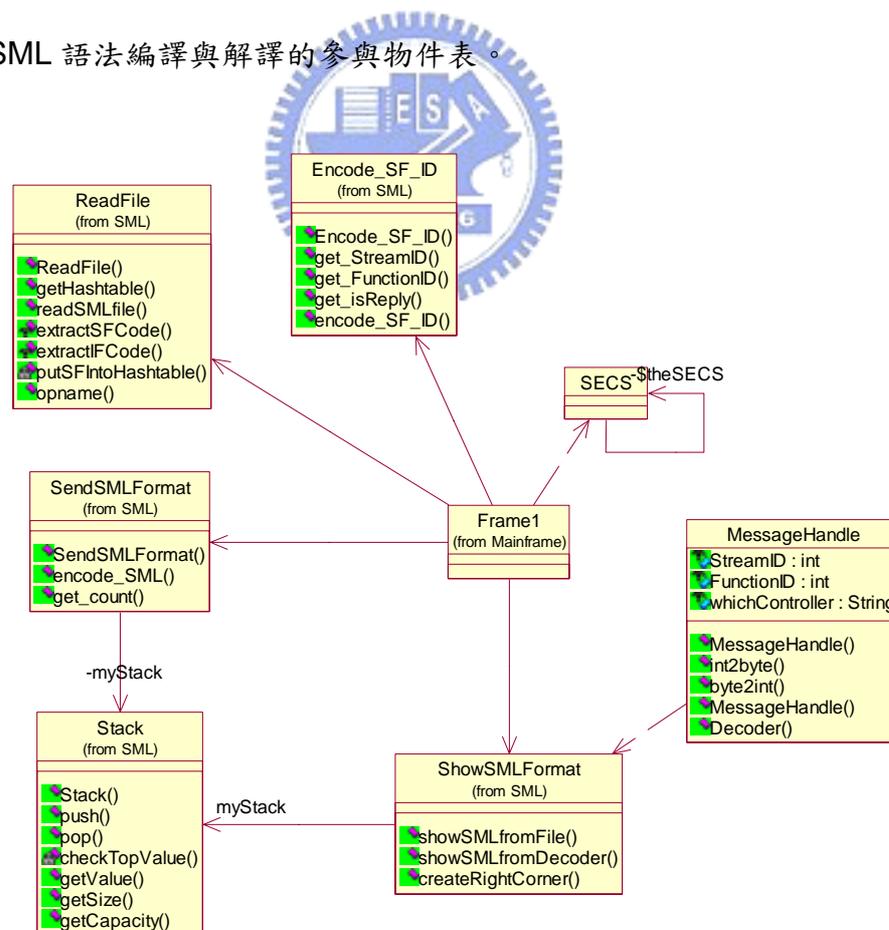


圖 4.16 通訊模組 SML 語法類別圖

表 4.8 通訊模組 SML 語法參與物件說明表

| 參與物件          | 說明   |
|---------------|--|
| Frame1        | 繪出使用者介面，可以在 Frame1 上顯示出訊息物件的 SML 語法                            |
| ReadFile      | 將*.sml 檔案讀取到的資料儲存到 Hashtable 中，讓 Frame1 主程式類別可以手動選擇要送出的訊息。     |
| SendSMLFormat | 將送出的訊息串成訊息字串，然後呼叫 Encoder2ByteVisitor 物件，將字串編譯成訊息位元組陣列，等待送出訊息。 |
| ShowSMLFormat | 用來解譯傳遞或接收到的訊息物件字串，並將訊息的 SML 語法顯示在使用者介面上。                       |
| Encoder_SF_ID | 用來解譯 GUI 上送出的訊息是哪種 Stream Function，是否須回傳。                      |
| Stack         | 用堆疊的方式來紀錄 SML 編碼時的層數。  |

下圖 4.17 所示為通訊模組接收訊息物件後顯示 SML 語法的循序圖，其訊息解譯步驟為：

1. 當通訊模組在接收到訊息物件後會先經由 HSMS 物件解譯，取出該訊息的 HSMS Header，從 Header 中可以判斷出該物件的 Stream ID 和 Function ID；
2. HSMS 物件呼叫 SECS 物件的 getInstance() 方法，等待將建立好的訊息物件放入 MessageQueue 中；
3. HSMS 呼叫 MessageHandle 物件的 Decoder() 方法，由 MessageHandle 物件建立該訊息的訊息物件；
4. 在 MessageHandle 物件中建立 ShowSMLFormat 的物件並將呼叫訊息物件 Encoder() 方法所得到的訊息字串傳入該物件的 showSMLfromDecoder() 中；
5. 經由堆疊的方法判斷訊息資料項目的層數，並一層一層的將訊息內容串成一個文字字串；
6. 將所得到的字串內容傳遞到 Frame1 物件的使用者介面上，顯示出所收到訊息的 SML 語法。

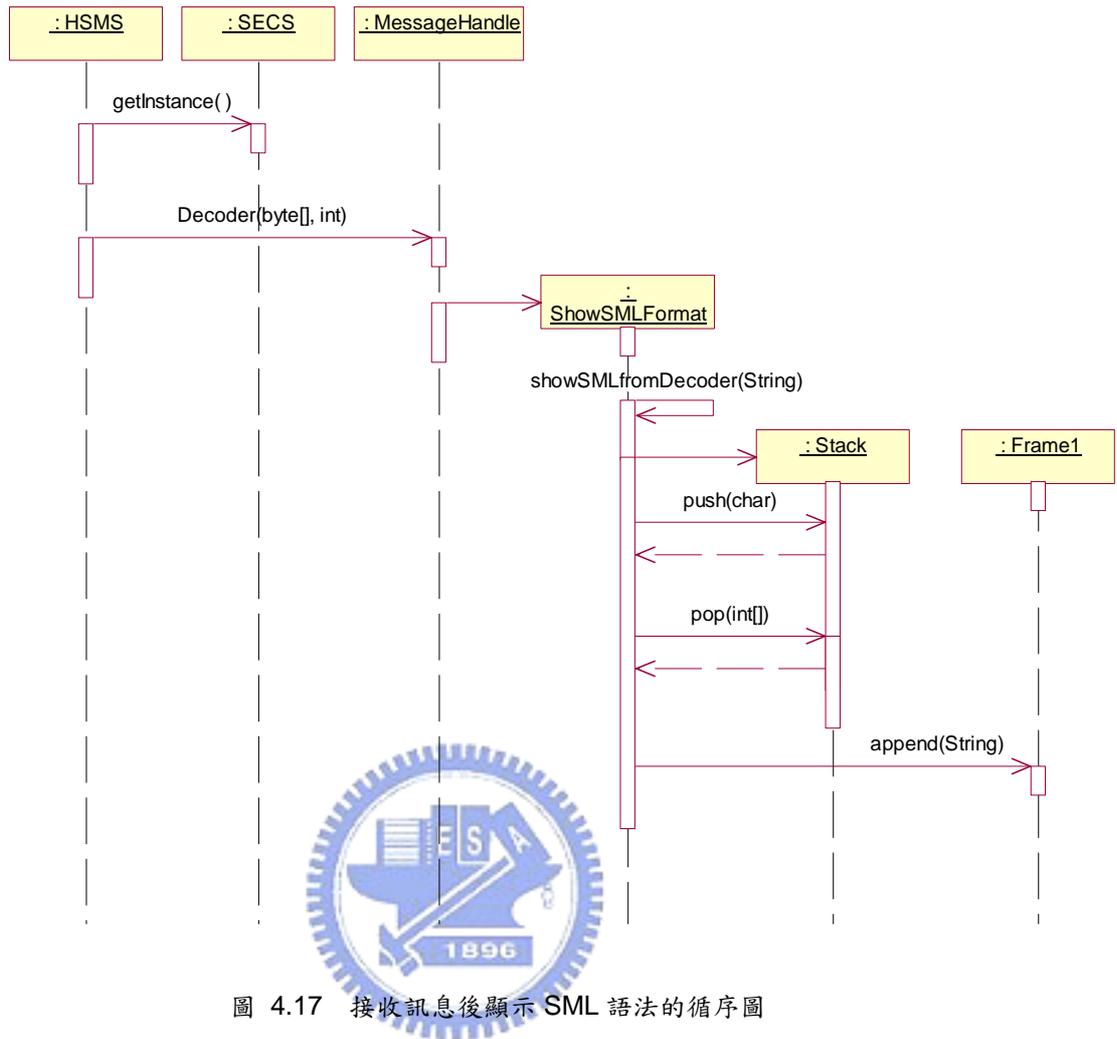


圖 4.17 接收訊息後顯示 SML 語法的循序圖

下圖 4.18 所示的是通訊模組在讀取 SML 檔案後傳送訊息時的循序圖，其訊息傳遞順序為：

1. 由 Frame1 物件建立一 ReadFile 物件，並呼叫該物件的 readSMLfile()方法，解譯從檔案中所得到的訊息內容；
2. ReadFile 物件建立一 Hashtable 物件並呼叫 pushSFIntoHashtable()方法，將 SML 檔案所得到的訊息內容存到 Hashtable 中；
3. 當按下 Frame1 上的 Send 按鈕後，建立一個 SendSMLFormat 物件並呼叫該物件的 encode\_SML()方法；
4. 經由堆疊的方式判斷訊息資料項目的層數，一層一層的將訊息內容串成一個文

- 字字串，並透過呼叫 Encoder2ByteVisitor 物件，將字串編譯成訊息位元組陣列；
5. 建立 Encode\_SF\_ID 物件，將 Frame1 上所點選到的訊息傳遞給該物件，用來解譯 GUI 上送出訊息 Stream ID 與 Function ID，以及是否須回傳的 W bit；
  6. 呼叫 SECS 物件的 SendMessage() 方法，將所得到的位元組陣列及 Encode\_SF\_ID 物件傳入方法內，並將訊息送出。

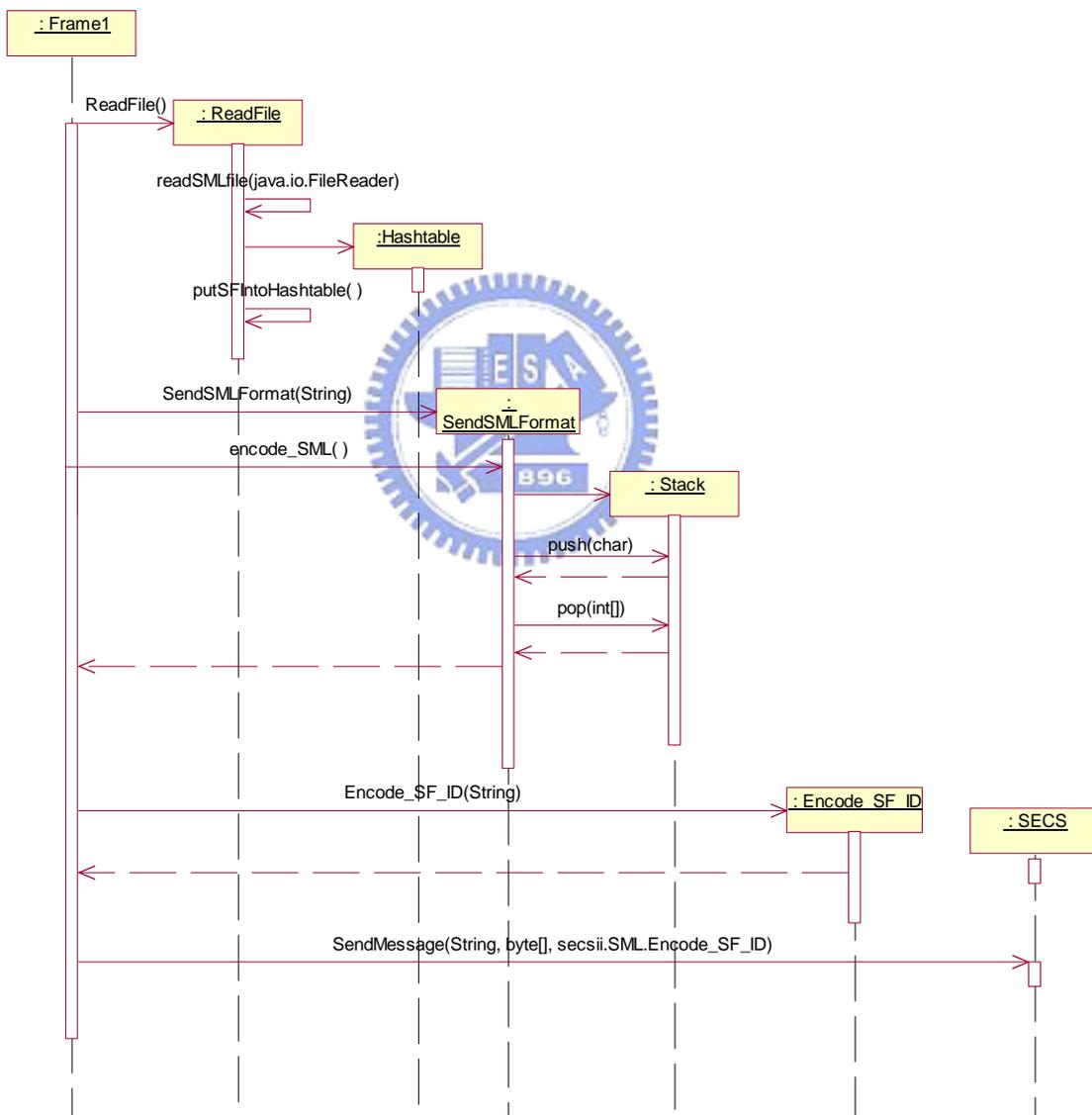


圖 4.18 讀取 SML 檔案傳送訊息時的循序圖

## 第5章 獨立介面通訊模組控制器軟體設計與實作

獨立介面通訊模組控制器(Independent-Interface SECS Communication Module Controller, i-SECS)的設計目的是希望將通訊模組的功能整合到一獨立的使用者界面上，並藉由訊息的傳遞達到設備控制與製程監控的目的。為了能夠快速擴充及替換設計元件，因此在程式設計上採用Framework的觀念，將程式模組化，並應用Design Patterns來達到程式「可再利用」的目的。在功能擴充時，可以直接新增類別到控制器，避免新增的功能影響到原來的程式設定；在程式修改時，因為採取模組化方式將程式區分成不同的功能模組，因此可以直接修改模組或是將模組替換，減少程式重新開發的時程。本章將說明i-SECS的設計架構並介紹i-SECS控制器程式的使用者介面。

### 5.1 i-SECS 控制器介紹

圖 5.1 所示為 i-SECS 硬體架構圖，i-SECS 整合了通訊模組與資料庫，並將其建構在實驗室現有的集束型製程設備上，達到擷取 CTC 與 HOST 間訊息內容的功能。

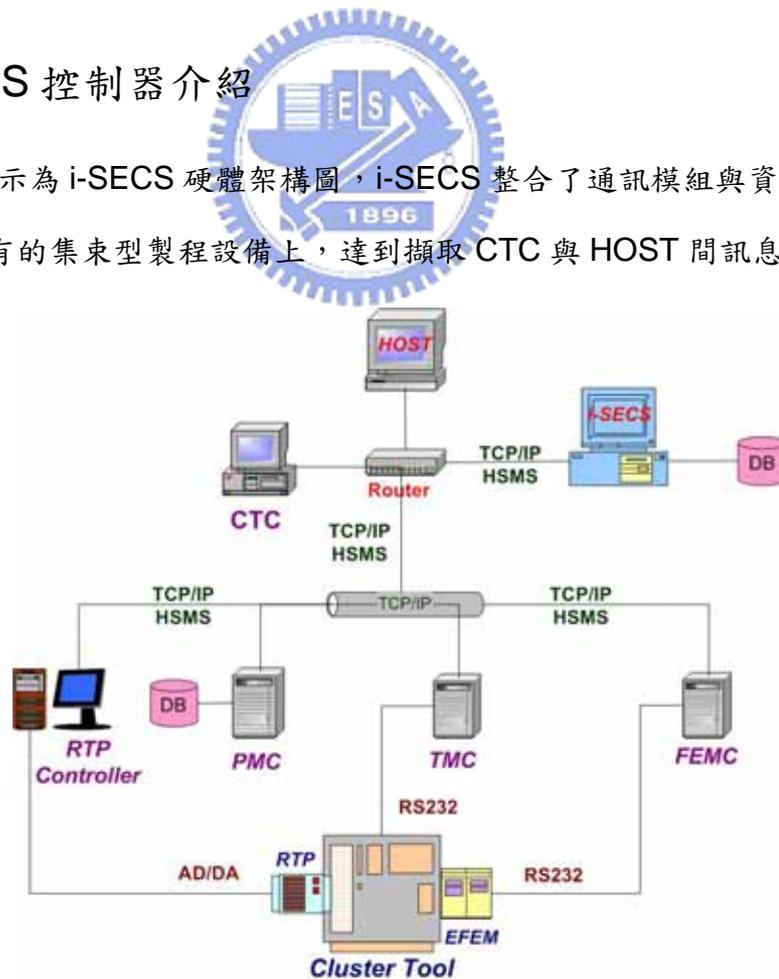


圖 5.1 i-SECS 硬體架構圖

### 5.1.1 i-SECS 控制器功能分析

i-SECS控制器在設計時必須符合SEMI Standard (HSMS、SECS-II)的通訊模組訊息傳遞規範，同時整合SEMI (E30) GEM Scenario [7][18]，目的是要擷取集束型製程設備控制器(CTC)與上層主機(HOST)間的訊息資料，並將接收到的資料存入資料庫中，以供後續發展先進製程控制(Advanced Process Control, APC)時使用；其次，加上獨立的使用者介面(Graphic User Interface, GUI)後，可以讓使用者從GUI得到訊息傳遞的資訊，並且以SML的格式顯示，可以立即得知訊息傳遞內容並且可以將訊息儲存以供後續追蹤查閱之用。下面我們以UML的使用案例圖來說明i-SECS 控制器所規劃的功能。

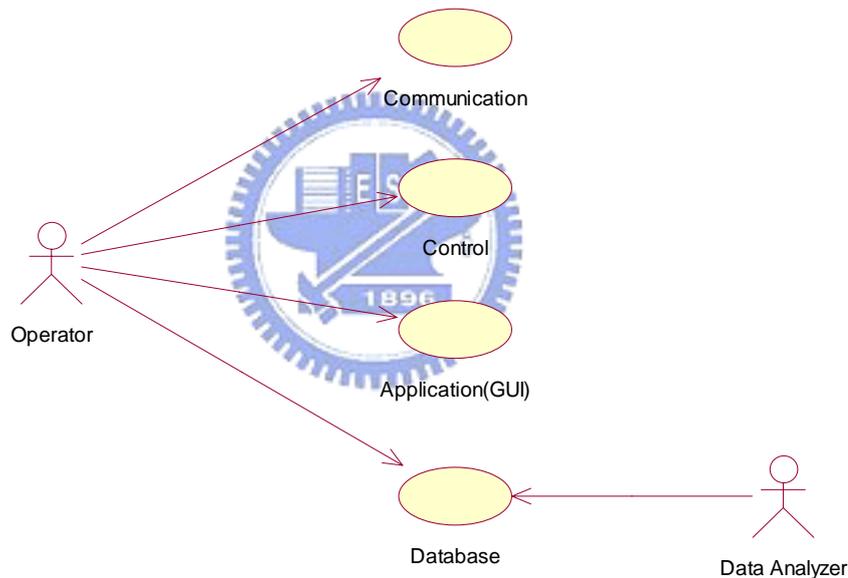


圖 5.2 i-SECS Use Case Diagram

圖 5.2 所示為 i-SECS 的使用案例圖，圖中我們將 i-SECS 分為四個功能模組，分別為通訊模組(Communication Module)、控制模組(Control Module)、使用者介面模組(GUI Module)以及資料庫處理模組(Database Module)。5.1.2 節我們將個別介紹各模組所負責的功能。

## 5.1.2 i-SECS 功能模組介紹

1. **通訊模組**：此為通訊模組的核心程式，負責 SECS-II 訊息物件的編譯、解譯以及遵照 HSMS 規範來傳送接收訊息的功能；同時，增加了 SML 的功能，目的是為了讓使用者能立即從 GUI 上得知所有訊息傳遞的內容並紀錄所有資料，達到我們所規劃的 Screen Logging 與 Data Logging 的功能。
2. **控制模組**：此模組是以實驗室內的集束型製程設備並依上層主機通訊介面[18]所訂定的功能來做程式的設計與規劃。功能包含了接收到訊息時的分派 (Dispatching) 以及控制器對訊息物件處理的控制邏輯等。在程式中，我們實現上層主機通訊介面中所規劃要實現的 GEM Scenario。將來當程式需要擴充或更改時，因為是依照 GEM Scenario 規劃，因此僅需對訊息內容定義不同的地方稍作修改即可正常運作。
3. **GUI 模組**：此模組提供 GUI 給使用者來操作 i-SECS，包含：SML Data Logging、Monitoring、State Diagram、Data Collection、Recipe Editor、Event Record、Alarm Record 以及 Terminal Services 等服務。
4. **資料庫處理模組**：資料庫的開發是以 MySQL 為程式的資料處理系統，i-SECS 透過資料庫處理模組，可以傳遞 SQL 語法到 MySQL 資料庫中，並將特定訊息的內容儲存到資料庫中，如：追蹤報告(Trace Report)時要求設備控制器上傳的週期性資料，或是上傳下載製程配方時的配方管理資料都需要儲存到資料庫中。

## 5.2 i-SECS 控制器架構設計

### 5.2.1 架構(Framework)介紹

架構[19][20]是一種基於特定目的而設計成可再利用的模組群。為了能以最少的工時和人力開發出能滿足所有要求的應用軟體，一般應用軟體的開發工程師會使用架構作局部自訂修改(Customize)並與其他零件搭配應用或擴充架構。利用物件導向語言所設計出來的架構稱為物件導向架構(Object-Oriented Framework)。架構具有類似抽象類別的性質，在設計物件導向架構時多半會利用抽象類別的性質，先在抽象類別定義幾個實作的方法，至於欲更有彈性修改實作的方法，則在子類別中定義。

Framework 的設計目標：

- 能更有彈性的自行修改或擴充
- 減少使用架構時須寫的程式
- 簡化介面讓使用者容易瞭解
- 製作文件讓使用者容易瞭解



### 5.2.2 i-SECS 控制器的架構設計

在設計控制器前，我們先以 UML 來規劃，包含了使用案例圖與各模組的類別圖、循序圖等，為了要達到架構設計的功能，我們採用兩種架構設計的方法：

1. 如 5.1.2 節所述，將控制器的功能區分為四個不同的功能模組。將模組分開主要的目的是希望能將模組獨立運作，減少將來程式擴充或修改時所花費的時間。
2. 利用 Design Patterns(節 4.3 所示)來協助程式達到 Framework 的功能，如：使用 Composite Pattern 與 Visitor Pattern 可以將訊息的處理(編碼解碼)與資料結構(SECS-II)分開，當我們要擴充訊息時只要增加訊息物件即可，並不會影響到編碼解碼的處理、使用 Bridge Pattern 可以將控制模組程式的功能層與實作層分開，達到 Framework 的目的。

### 5.2.3 i-SECS 控制器程式實作

i-SECS 控制器是由通訊模組核心程式(節 4.4)所開發出來的獨立使用者介面控制器。通訊核心程式在歷經紀森炫學長與李妍慧學姊[4]的研究開發後，程式通訊功能已趨於完備。為了配合 i-SECS 控制器的應用，因此在通訊模組核心程式內做了部份的修改與功能的擴充：

- **修改功能部份**— 在 HSMS 類別中增加 Hashtable 來存放 HSMS Header 的 SystemBytes 值，目的是要讓控制器可以同時送出多個 Primary Message，並等待接收對應訊息的 Secondary Message。
- **新增功能部份**— 增加通訊模組的 SML 功能(如節 4.5)，讓通訊模組可以將訊息以 SML 的格式顯示在 GUI 上，增加訊息的可讀性。另外，為配合曹洪泰同學的 GEM Scenario 功能[18]，因此在通訊模組內增加了 74 個訊息物件，擴充後通訊模組內可用的訊息物件為 146 個，詳細訊息內容格式請參照附錄一。

在 i-SECS 控制器程式的實作上，主要是在應用層軟體控制器程式的撰寫，目的是希望延伸通訊模組的功能，讓通訊模組可以獨立操作，達到設備控制與製程監控的目的。本節以圖 5.3 i-SECS 接收 HOST 端傳遞訊息循序圖及圖 5.4 應用 Bridge Pattern 的時間調整類別圖來說明 i-SECS 的程式架構。

- 循序圖說明

- [1] 建立一執行緒，反覆不斷的檢查 HOST 端的 MessageQueue 中是否有訊息物件存在，如果有，則將訊息物件取出。
- [2] 將訊息物件交由 MessageDispatcher 物件作訊息的分派。
- [3] 檢查訊息格式，解析此訊息內容是否正確，得到訊息中所包含的資訊，並判

斷訊息應該做的處理，最後再交由相關的功能物件執行。

[4] 將訊息轉傳到 CTC 端。

[5] 刪除 MessageQueue 中的訊息物件。

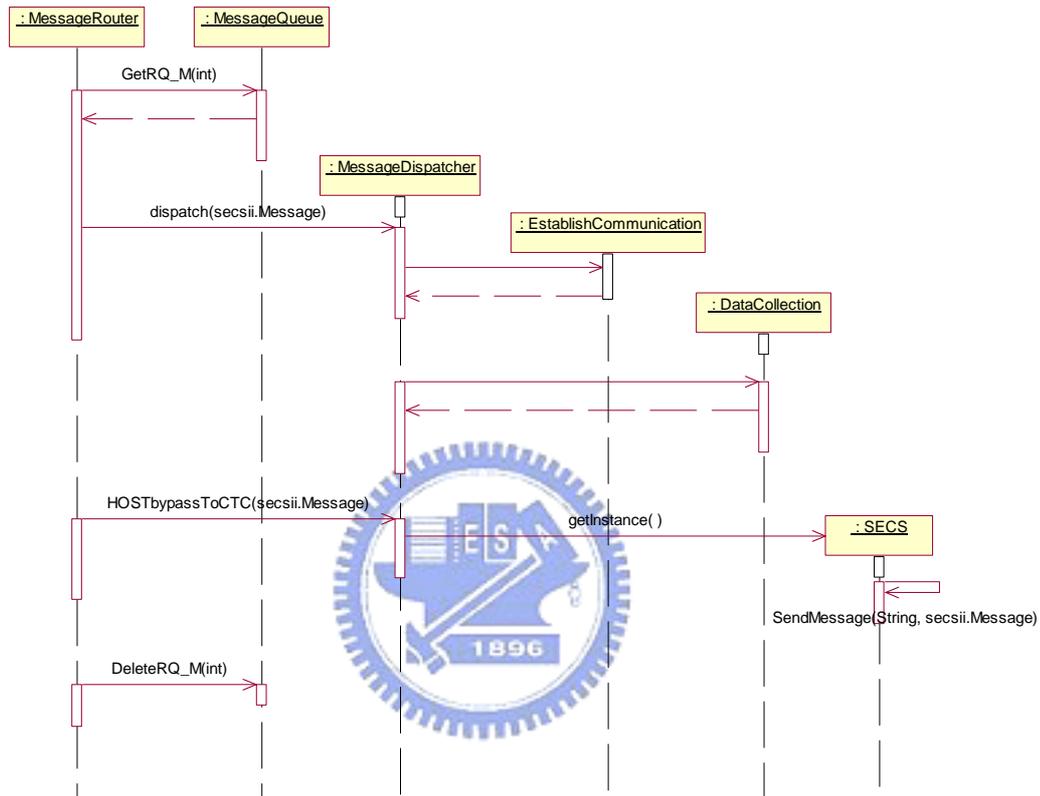


圖 5.3 i-SECS 接收 HOST 端傳遞訊息循序圖

圖 5.4 所示的是 i-SECS 應用 Bridge Pattern 的時間調整類別圖。TimeAdjust 是一個符合 GEM Scenario 時間調整的功能類別，其主要的目的是當接收到時間調整訊息時，可以透過 Bridge Pattern 呼叫實作層 Clock\_Impl 的 setclock() 方法，來調整 i-SECS 控制器的時間，讓 i-SECS 的時間能跟上層主機及設備控制器的時間一致。我們以表 5.1 來說明時間調整類別中每個參與物件的工作。

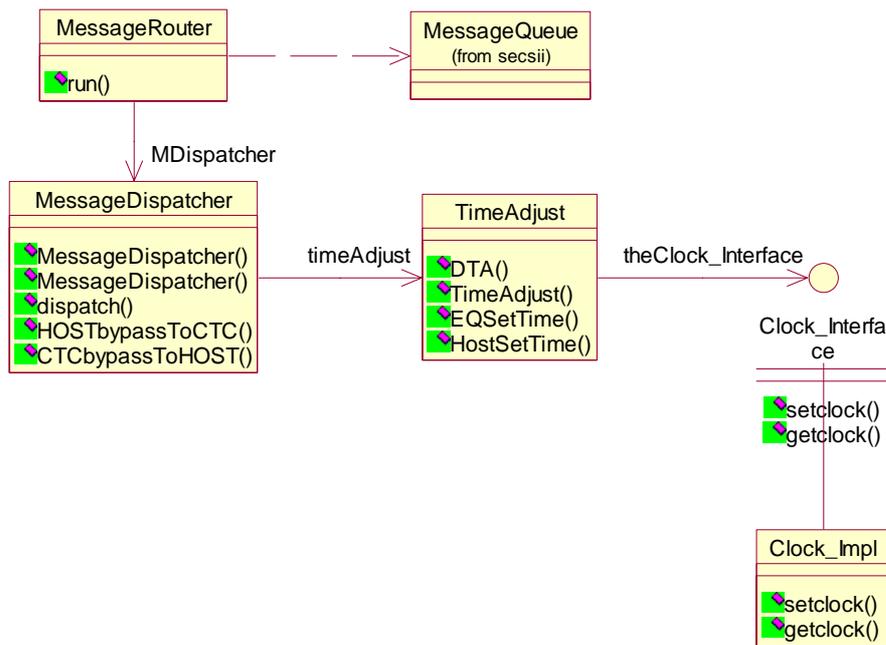


圖 5.4 應用 Bridge Pattern 的時間調整類別圖

表 5.1 時間調整類別說明

| 物件名稱              | 說明   |
|-------------------|--|
| MessageQueue      | 屬於 SECS 通訊模組的一個類別，負責暫時存放所接收到的訊息物件。   |
| MessageRouter     | 由控制器所建立的獨立執行緒，負責檢查通訊模組內的訊息佇列(MessageQueue)是否接收到訊息，並將訊息傳遞給 MessageDispatcher 物件作訊息的分派。  |
| MessageDispatcher | 負責分派所接收到的訊息物件，判斷訊息應該做的處理，交由相關的功能物件執行。  |
| TimeAdjust        | 為時間調整的功能類別，有兩種可能的時間調整方式：<br>1. 當上層主機主動要求設定時間，<br>2. 上層主機要求設備回報時間，<br>當收到此兩種時間調整的訊息時，呼叫 Clock_Impl 類別，執行 setclock() 的方法，更改控制器的時間。 |
| Clock_Interface   | 為一介面，介面中定義了時間調整的方法，並交由其子類別 Clock_Impl 來實作。   |
| Clock_Impl        | 實作時間調整的工作，提供設定時間與取得控制器時間的方法  |

## 5.3 i-SECS 使用者介面介紹

i-SECS 控制器提供使用者介面(GUI)供操作人員使用，GUI 上主要的功能包含了 SML 訊息、機台連線狀態、資料收集、製程配方管理、事件回報、警報管理及設備終端服務等，可以執行並監看製程工作的狀態。以下將對 GUI 部份做詳細的說明。

### 5.3.1 GUI 概述

首先介紹 GUI 的畫面安排，如圖 5.5 所示，每一個 UI 可以分做三個區塊，分別是：Communication Status Panel、Control Screens Panel 以及 Screen Select Panel。

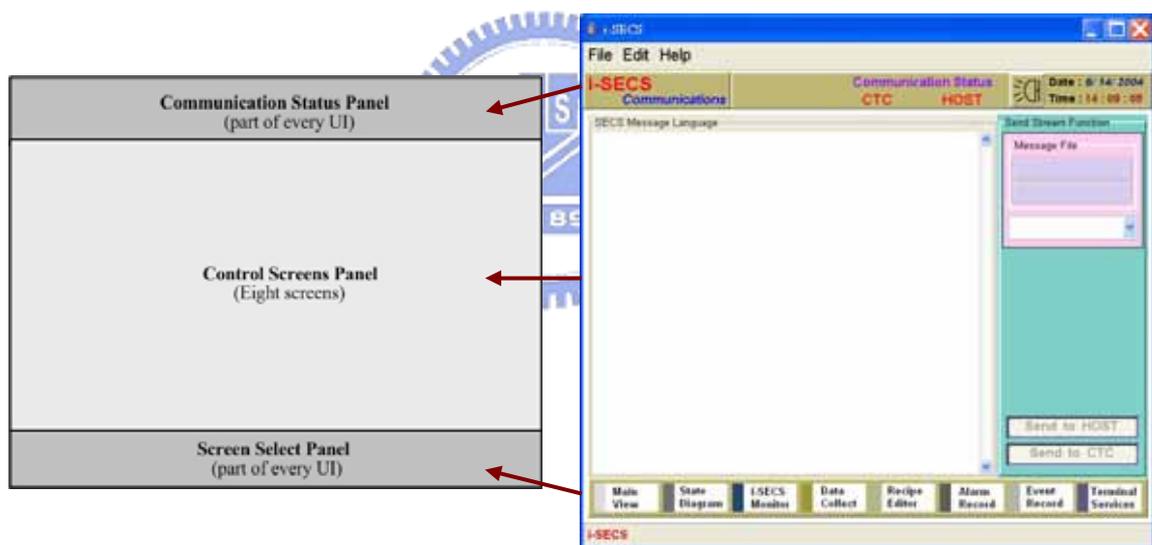


圖 5.5 i-SECS GUI Layout

#### ■ Communication Status Panel

如圖 5.6 所示，此 Panel 可以顯示 i-SECS 控制器目前的連線狀態，包含了 i-SECS 與設備控制器(CTC)以及和上層主機(HOST)間的連線狀態;另外也可以顯示是否發生警報狀況以及顯示控制器目前的时间。

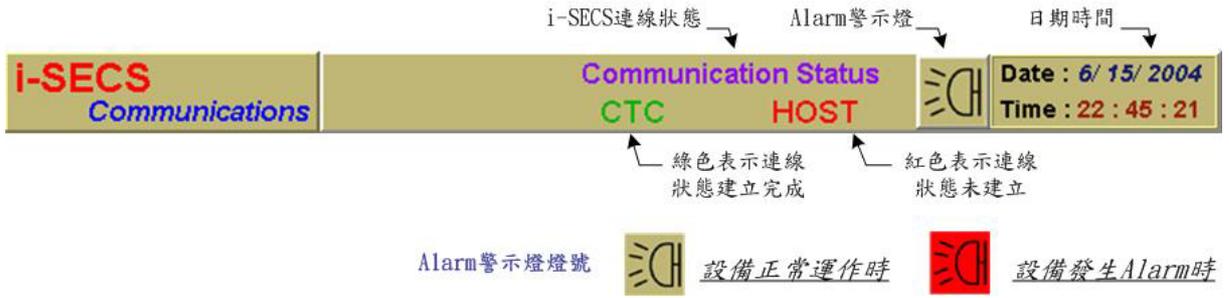


圖 5.6 Communication Status Panel

## ■ Control Screens Panel

Control Screens Panel 內有八個「控制畫面」來對 i-SECS 做設定、操作與監控。藉由選擇圖 5.7「Screen Select Panel」中的按鈕可顯示出相對應的「控制畫面」，分別為：

- Main View (SML Logging)
- State Diagram
- i-SECS Monitor
- Data Collect
- Recipe Editor
- Alarm Record
- Event Record
- Terminal Services



## ■ Screen Select Panel

此為視窗切換面板，選擇每個功能鈕可以顯示出相對應的「控制畫面」，下一小節開始將對此八個畫面做詳細的操作畫面介紹。

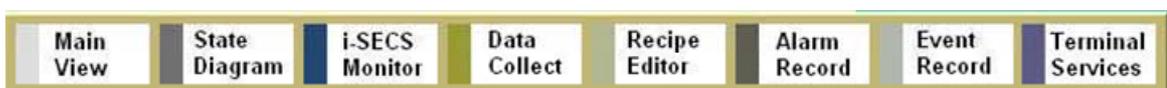


圖 5.7 Screen Select Panel

## 5.3.2 控制畫面視窗介紹

### 5.3.2.1 Main View (SML Logging) GUI 說明

主畫面的設計，是希望讓 i-SECS 具有單機測試通訊訊息的功能，功能如：『工研院的 SECS Emulator 測試軟體[21]』；同時能顯示 HOST、i-SECS 與 CTC 之間所有訊息傳遞的 SML 語法。如圖 5.8 所示可將畫面可以分做左、右兩大方塊：

- ◆ 左大方塊(SECS Message Language)用來顯示通訊建立狀態、訊息傳遞的 SML 語法，包含訊息傳送接收時間、訊息內容等；
- ◆ 右大方塊(Send Stream Function)則是用來連結已編好的\*.sml 檔案，並列出可傳遞的訊息，讓 i-SECS 控制器可以手動傳遞測試訊息，達到單機測試設備通訊之目的；下方的兩個按鈕(  、  )可以讓操作者選擇測試訊息所要傳遞的對象，並將訊息送出。

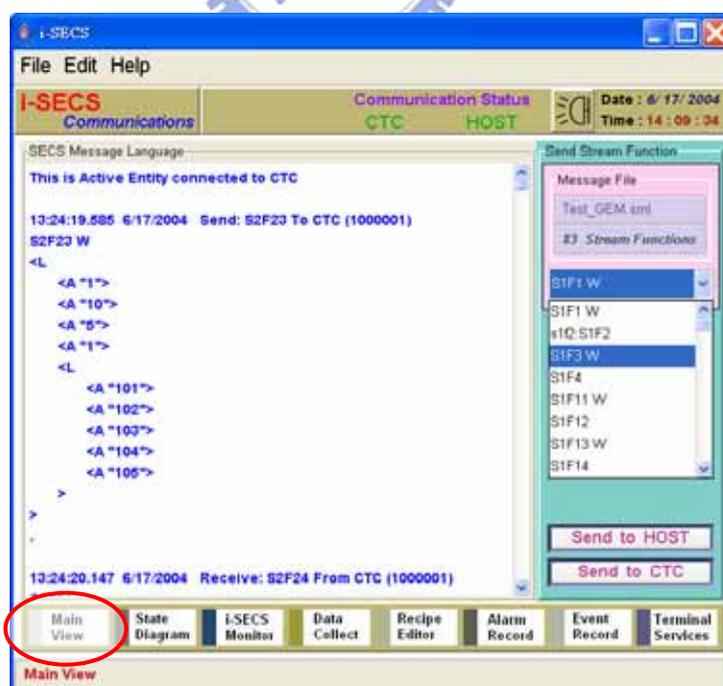


圖 5.8 i-SECS GUI 主畫面視窗

### 5.3.2.2 State Diagram GUI 說明

狀態圖的設計是希望讓 i-SECS 的操作者可以監控設備端與上層主機間的連線狀態，i-SECS 的狀態圖部份分為兩種：

1. **HSMS 連線狀態圖**：這裡的 HSMS 連線狀態圖是用來檢視 i-SECS 與 CTC 間的通訊連線是否完成，圖 5.9 HSMS 連線狀態圖的說明請參考 4.1.3 節的介紹。i-SECS 在與 CTC 連線時所扮演的角色預設為 Active Mode，當 i-SECS 與 CTC 連線建立完成後會進入[TCP/IP] CONNECTED 狀態，同時 i-SECS 會送出 Select.req 的控制訊息給 CTC，等待 CTC 回傳 Select.rsp 訊息；當 i-SECS 收到 Select.rsp 控制訊息後，連線狀態會進入[HSMS]SELECTED 狀態，在此狀態下 i-SECS 才可與 CTC 端作資料訊息的傳遞。
2. **GEM 連線狀態圖**：如圖 5.10 所示，i-SECS 在此所扮演的角色為監控設備控制器跟上層主機間通訊與控制狀態的變化。因為 i-SECS 無法操作 GEM 的遠端監控功能，所以這裡所看到的狀態變化都是依照 CTC 與 HOST 間傳遞的訊息內容來判斷。由狀態圖的改變可以得知目前兩端控制器間的操作狀態。

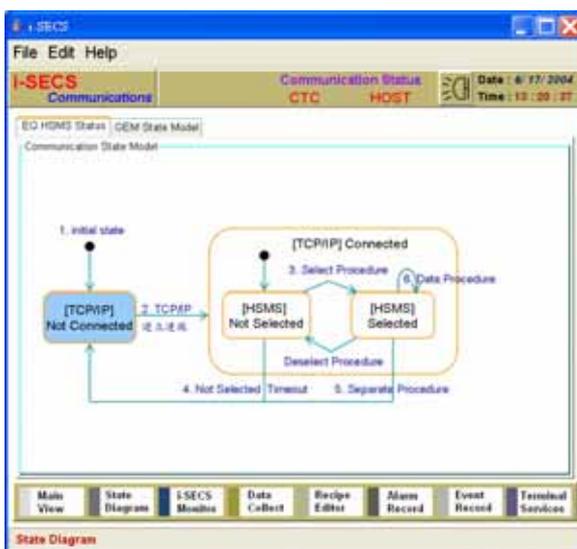


圖 5.9 HSMS 連線狀態圖

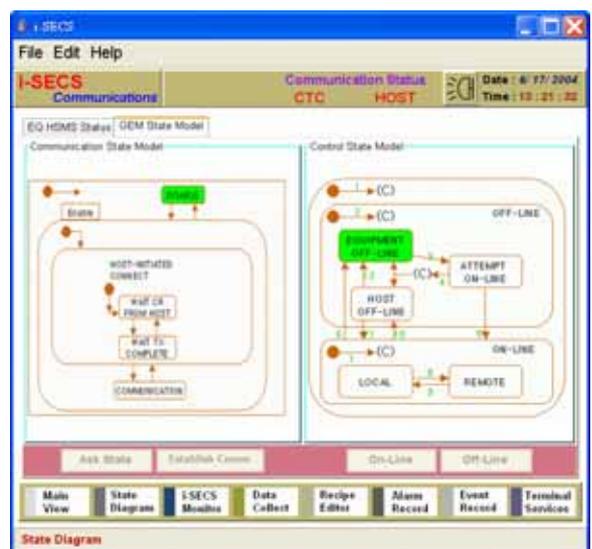


圖 5.10 GEM 連線狀態圖

### 5.3.2.3 i-SECS Monitor GUI 說明

i-SECS 控制器 Monitor 使用者畫面設計的功能是希望能監控晶圓的製程與傳輸的工作，當傳輸模組控制器(TMC)與製程模組控制器(PMC)執行傳輸或製程工作時，都會回報一個事件報告給 CTC，CTC 在收到這些事件報告後，必須依上層主機的要求傳遞事件報告到 HOST。i-SECS 控制器可由 CTC 與 HOST 間傳遞的訊息中判斷目前集束型製程設備的工作是否正常；同時在製程中，可以將製程的資訊顯示到使用者介面上，讓操作人員立即察覺目前的製程參數是否正確。圖 5.11 所示的是以實驗室內現有的集束型半導體製程設備為監控對象，當傳輸工作與製程工作進行時，藉由 CTC 上傳 HOST 端的事件回報訊息可以在 i-SECS 控制器的 GUI 上顯示出目前的製程或傳輸工作，並且可以監控設備端操作是否正常。

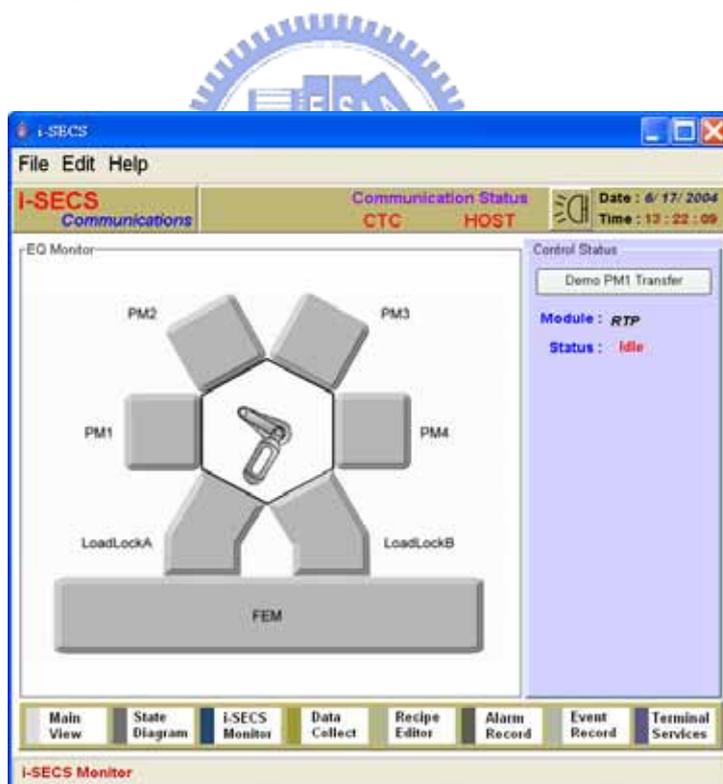


圖 5.11 i-SECS Monitor

### 5.3.2.4 Data Collect GUI 說明

本節所介紹的操作畫面是以第六章系統測試中 GEM Scenario 的資料收集劇本來設計，訊息劇本請參考 6.1.2.2 資料收集 (Data Collection)一節有詳細的說明。在此，資料收集分為三種不同的收集畫面，下面開始分別說明三個不同的資料收集畫面：

#### 5.3.2.4.1 Event Data Collection GUI 說明

圖 5.12 所示的是事件資料收集(Event Data Collection)的畫面。事件資料收集的對象是針對事件回報的訊息，包含每個 CEID 所連結到的 RPTID，以及每個 RPTID 所應連結到的 VID 有哪些。圖 5.12 我們將其分為三部份來說明：

- ◆ 圖的左方(CEID List)部份列出的是設備控制器提供的所有 CEID 清單；
- ◆ 右上方(Request Event Report)部份則是提供 i-SECS 以表 6.3「請求事件回報劇本」跟 CTC 要求上傳每個 CEID 所掛載的內容。
- ◆ 右下方(Event Report Setup)部份可以讓 i-SECS 設定每個 CEID 所連結的 RPTID，以及每個 RPTID 所連結的 VID 有哪些，並將連結的結果下傳給 CTC，等待 CTC 設定資料的連結後回傳確認訊息。



圖 5.12 Event Data Collection

### 5.3.2.4.2 Variable Data Collection GUI 說明

圖 5.13 所示的是變數資料收集(Variable Data Collection)的畫面，這裡變數資料收集的部份包含了連續不間斷的狀態變數(Status Variables, SV)，例如：熱板溫度、濕度、風速、排氣及時間等；以及設備的固定常數(Equipment Constants, EC)，例如：軟體版本、設備編號等資料的連結。詳細的內容說明請參考 6.1.2.2.2 Variable Data Collection一節。圖 5.13 所示的畫面可分為兩部份來說明：

- ◆ 上方的部份(Status Variable Requests)是 i-SECS 用來要求設備控制器上傳目前每個 SVID 所對應連結到的 SV 值，同時也可以要求上傳對應 SVID 的狀態變數名稱清單(SV Namelist)並顯示在右方的欄位上。
- ◆ 下方的部份(Equipment Constant Requests)則是用來要求設備控制器上傳每個 ECID 所對應到的 EC 值，同時也可以要求上傳對應 ECID 的設備常數名稱清單 (EC Namelist)並顯示在右方欄位上。



圖 5.13 Variable Data Collection

### 5.3.2.4.3 Trace Data Collection GUI 說明

圖 5.14 所示的是追蹤資料收集(Trace Data Collection)的設計畫面，追蹤資料收集是上層主機週期性監控設備端控制器 SVID 值變化的劇本，有關追蹤資料收集的詳細資料請參照 6.1.2.2.3 一節。圖 5.14 所示的畫面可分為兩部份來說明：

- ◆ 上方的部份(Trace Data Collection)為追蹤資料的設定，用來設定 i-SECS 與 MySQL 資料庫的連結以及每筆追蹤資料的 Trace Report ID(TRID)、取樣週期、取樣次數等參數。當設備控制器開始週期性上傳狀態變數資料時，i-SECS 會將每筆收集到的資料儲存到 MySQL 資料庫中，同時再由右邊的資料庫表格將收集到的資料顯示到 GUI 上，可以讓 i-SECS 操作員了解到目前所收集到的資料內容；
- ◆ 下方的部份是希望將收集到的資料繪製成圖表，這裡所設計的畫面，是依照 RTP Controller 回報製程溫度而設計的製程溫度圖，橫軸為時間(t,sec)，縱軸為溫度(T, °C)，可以週期性的將設備端上傳的溫度紀錄到製程溫度時間圖上。

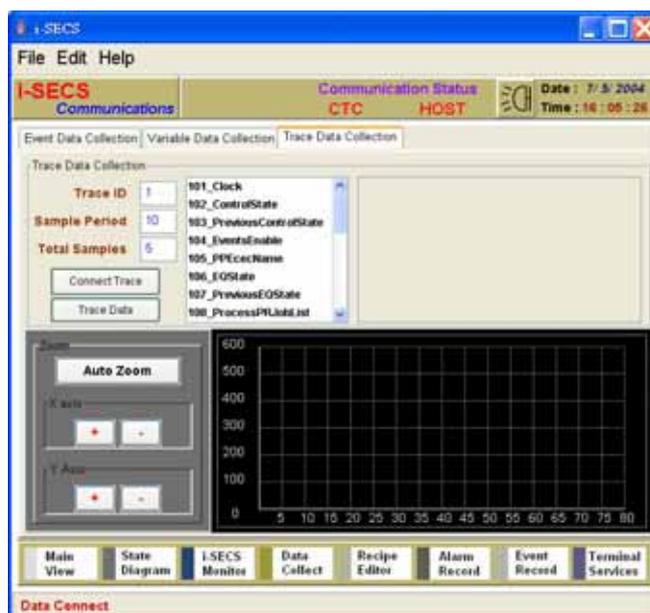


圖 5.14 Trace Data Collection

### 5.3.2.5 Recipe Editor GUI 說明

圖 5.15 所示的是針對製程配方管理所設計的畫面，有關製程配方管理劇本的詳細說明請參照 6.1.2.4 一節。圖 5.15 所示的畫面可分為兩部份來說明：

- ◆ 左上方的部份(EQ Recipe List Management)目的是要讓 i-SECS 可以顯示出設備控制器目前的製程配方清單，i-SECS 控制器經由下傳訊息後可以要求刪除設備控制器製程配方清單中的項目，當按下  按鈕後可以檢視該筆製程配方的內容，並顯示在右上方(EQ Process Program Uploading)的表格中，當 i-SECS 操作員按下 Save 按鈕後，可以將此配方的內容儲存到資料庫中；
- ◆ 下方的部份(i-SECS Recipe Management)則是讓 i-SECS 連結到資料庫中，可以建立並刪除製程配方資料表，當按下  按鈕後可以檢視該筆製程配方的內容，並且可以將製程配方資料下傳到設備端控制器，完成製程配方下載的功能。

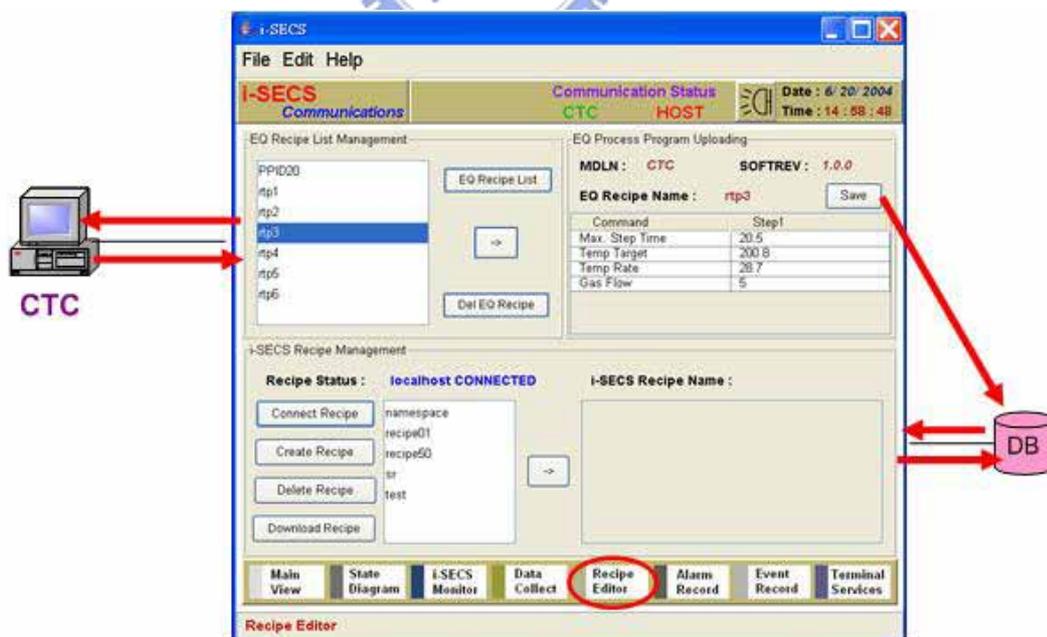


圖 5.15 Recipe Editor

### 5.3.2.6 Alarm Record GUI 說明

圖 5.16所示為當設備端發生警報(Alarm)時，CTC會主動發出警報訊息通知上層主機，i-SECS在接收到CTC上傳的警報訊息後，會改變Alarm警示燈的燈號為紅色，同時紀錄所有發生Alarm的訊息內容，當CTC排除Alarm後會上傳警報解除訊息，i-SECS在接收到警報解除訊息後將燈號變為正常燈號，回復到設備正常運作狀況。

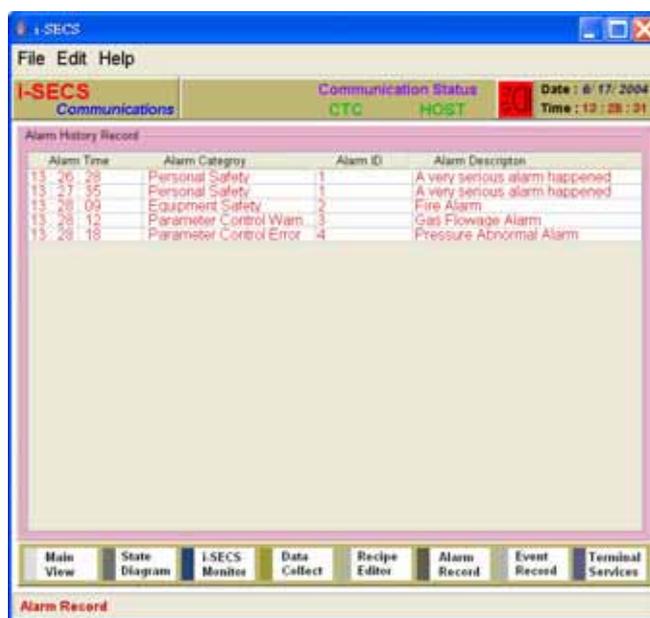


圖 5.16 Alarm Record

### 5.3.2.7 Event Record GUI 說明

當設備的每一個事件發生時，設備控制器都會送出 S6F11 的事件回報訊息來通知上層主機，上層主機可經由設備上傳的資料來監控目前的製程或傳輸工作。圖 5.17 所示的是 i-SECS 控制器藉由接收到的事件回報訊息紀錄所有設備控制器上傳上層主機的事件報告(Event Report)，所紀錄的內容包含：事件發生時間、事件 DATAID、事件 CEID 以及事件的內容描述等。

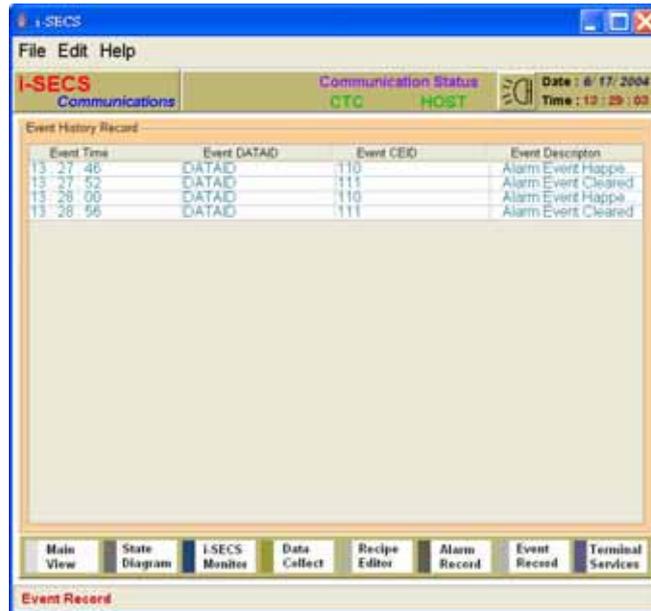


圖 5.17 Event Record

### 5.3.2.8 Terminal Services GUI 說明

圖 5.18 所示的是 i-SECS 實作 Terminal Services 功能的畫面，在圖中 Terminal Message History 部份可以紀錄所有 CTC 與 HOST 間傳遞的設備終端訊息，而 Message Description 部份則允許 i-SECS 控制器主動傳遞設備終端訊息給設備端控制器。

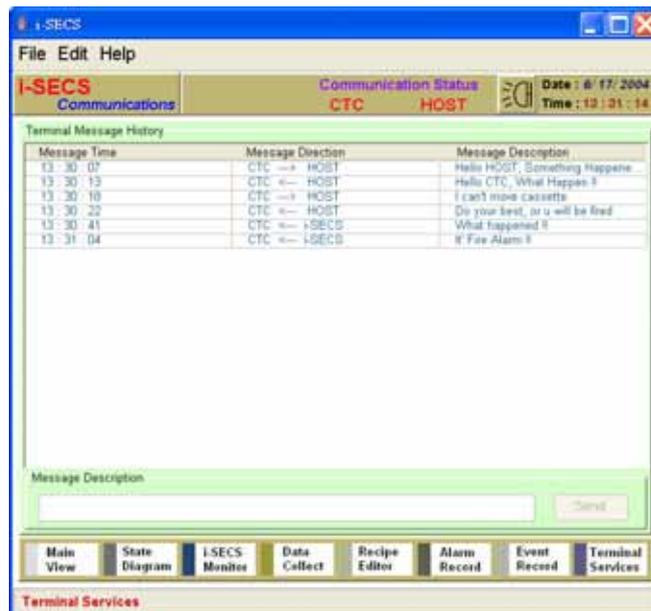


圖 5.18 Terminal Services

## 第6章 系統測試

系統測試是以實驗室現有的集束型半導體製程設備控制器(CTC)來驗證 i-SECS 的功能。CTC 是 91 年中科院期末報告[3]中所完成的控制器程式；在 92 年的中科院報告中[22]，由曹洪泰同學在 CTC 上規劃出與上層主機溝通的通訊介面[18]，讓 CTC 能夠透過 GEM 與 HOST 溝通來做實際機台的連線測試。本章首先介紹上層主機的通訊介面規劃，接下來再以實機連線測試來說明 i-SECS 與 CTC 跟 HOST 間連線測試的結果。

### 6.1 GEM Scenario 介紹

#### 6.1.1 GEM 簡介

現代化的製程設備都必須有能力回報製程中的工作資訊到上層主機(HOST)，並允許上層主機可以參與設備的操作以整合所有的製程設備，因此 SEMI E30 GEM [7]定義了半導體製程設備控制器與上層主機之間的通訊規則，目的是希望將所有半導體製程設備有效的整合到整廠自動化的系統中。圖 6.1 說明 GEM 的訊息傳遞是依 SECS-II 的格式以 HSMS(TCP/IP)方式來傳送。GEM 規範了訊息傳遞的時機與劇本(Scenario)，因此要實現 GEM 的功能之前一定要先建構完整的底層通訊架構。

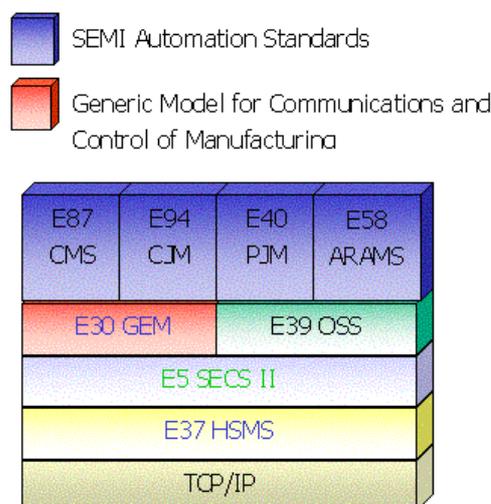


圖 6.1 ON-LINE 架構圖

## 6.1.2 設備性能和上層主機通訊介面劇本

本節將介紹的是設備性能和上層主機通訊介面劇本(Equipment Capabilities and GEM Scenarios)。上層主機通訊介面軟體在設計時必須依據 SEMI E30 GEM 的規範來訂定劇本的內容。因此，本節說明在 i-SECS 控制器內所實作的 GEM Scenario，其種類如下：

- 建立連線的劇本 (Establish Communications)
- 資料收集的劇本 (Data Collection)
- 警報管理的劇本 (Alarm Management)
- 製程配方管理的劇本 (Process Recipe Management)
- 材料搬運的劇本 (Material Movement)
- 設備終端服務的劇本 (Equipment Terminal Services)
- 錯誤訊息的劇本 (Error Message)
- 時間調整的劇本 (Clock)
- 控制狀態遷移的劇本 (Control State Transition)

為驗證 i-SECS 控制器的操作功能，因此我們以上述所列的劇本來做控制器的程式規劃，並將劇本的功能實作到 i-SECS 控制器中，下節開始將說明各劇本的訊息傳遞順序與劇本功能的內容，劇本中所提到的訊息格式內容請參照附錄一所示；6.2 節開始我們以實機驗證的方式來驗證 i-SECS 控制器的功能，最後並以 GUI 的畫面來顯示實機驗證的結果。

### 6.1.2.1 建立連線 (Establish Communications)

表 6.1 所示說明設備控制器(CTC)與上層主機(HOST)之間建立連線的劇本，HOST 主動下傳 S1F13 訊息給 CTC，箭頭(→)表示訊息傳遞的方向，當 CTC 收到 S1F13 後必須回傳 S1F14 的訊息給 HOST 完成訊息交握。但是如果訊息的交握無法順利完成，例如：「設備不想連線」，則必須在 S1F14 回傳訊息中做設定；如果是「S1F13 的訊息格式錯誤」，則 CTC 無法對應訊息，因此必須以 Stream 9(Error Message)來回覆 HOST。

表 6.1 Host Attempts to Establish Communication

| Comment    | Host  |   | Equipment | Comment   |
|------------|-------|---|-----------|-----------|
| 上層主機要求建立連線 | S1F13 | → |           |           |
|            |       | ← | S1F14     | 建立連線後回覆訊息 |

### 6.1.2.2 資料收集 (Data Collection)

資料收集劇本可以提供上層主機監控設備狀況、製程狀態以及製程參數資料收集等功能，在發展先進製程控制(APC)前，HOST 必須先藉由資料收集的劇本與 CTC 溝通並擷取設備內的變數資料。接下來將說明 i-SECS 控制器內使用到的設備資料收集劇本。

#### 6.1.2.2.1 Event Data Collection (事件報告收集)

事件報告收集的劇本提供在製程開始前先完成事件報告連結的功能，等待製程開始後且事件發生時，設備控制器會主動上傳事件報告。表 6.2 所示說明當設備事件發生時，會主動送出事件報告 S6F11 訊息來通知上層主機。表 6.3 則說明上層主機主動要求設備上傳特定一筆 CEID(Collected Event ID)所連結的資料數據。表 6.4 說明上層主機設定事件報告所連結的資料項目然後下傳到設備控制器，事件資料收集的對象是針對

事件回報的訊息，包含每個 CEID 所連結到的 RPTID(Report ID)，以及每個 RPTID 所應連結到的 VID(Variable ID)有哪些，並可設定設備端控制器的 CEID 為 Enable 或 Disable。經由這些上傳的資料上層主機可以利用事件報告裡的 VID 變化來做為遠端控制參考依據，例如：Wafer 的 Mapping 動作需要等待 FOUP 的定位的事件報告送出後才能進行下一動作。

表 6.2 Event Report from Equipment

| Comment  | Host  |   | Equipment | Comment               |
|----------|-------|---|-----------|-----------------------|
|          |       | ← | S6F11     | 當事件發生時,CTC上傳事件報告給Host |
| 上層主機回覆訊息 | S6F12 | → |           |                       |

表 6.3 Host Requests Event Report

| Comment               | Host  |   | Equipment | Comment                           |
|-----------------------|-------|---|-----------|-----------------------------------|
| Host要求上傳CEID所連結的資料數據. | S6F15 | → |           |                                   |
|                       |       | ← | S6F16     | CTC上傳CEID所連結的資料.<br>註:DV上傳的內容為空白. |

表 6.4 Event Reporting Setup

| Comment               | Host  |   | Equipment | Comment |
|-----------------------|-------|---|-----------|---------|
| 設定RPTID與VID之間的連結      | S2F33 | → |           |         |
|                       |       | ← | S2F34     | 設定成功    |
| 設定CEID與RPTID之間的連結     | S2F35 | → |           |         |
|                       |       | ← | S2F36     | 設定成功    |
| 設定CEID的Enable/Disable | S2F37 | → |           |         |
|                       |       | ← | S2F38     | 設定成功    |

### 6.1.2.2.2 Variable Data Collection (變數資料收集)

由上一節得知上層主機可以藉由設備上傳的事件報告來收集設備的狀態。而上層主機也可以使用變數資料收集的劇本主動向設備控制器要求上傳目前的變數資料，這個劇本用於 SVID 的監控。變數代碼(VID)的分類有 3 種，分別是 SVID、ECID、DVID。

- SVID (Status Variables)

連續不間斷的狀態變數，例如：熱板溫度、濕度、風速、排氣、時間等。

- ECID (Equipment Constants)

設備的固定常數，例如：軟體版本、設備編號等。

- DVID (Data Variable)

因事件而產生的資料變數(Data Variable)，例如：當 wafer 進入製程模組時算是一個事件的發生，所以設備控制器會上傳事件報告給上層主機，而此事件報告會連結的資料包括製程開始的時間、製程結束的時間等資訊後上傳到上層主機。



表 6.5 說明上層主機可以要求設備上傳指定的 RPTID 所連結的資料數據，當上層主機下傳 S6F19 訊息時，設備必須將 RPTID 所連結的資料數據用 S6F20 的訊息來回覆給上層主機。

表 6.5 Host Requests Individual Report

| Comment                  | Host  |   | Equipment | Comment                               |
|--------------------------|-------|---|-----------|---------------------------------------|
| Host 要求上傳 RPTID 所連結的資料數據 | S6F19 | → |           |                                       |
|                          |       | ← | S6F20     | 上傳 RPTID 所連結的 SV 值.<br>註:DV 上傳的內容為空白. |

表 6.6 說明上層主機詢問設備狀態變數(SV)的資料數據,表 6.7 則說明上層主機可以要求設備控制器上傳 CTC 內的狀態變數名稱清單(SV Namelist)。

表 6.6 Host Requests Current Status Variable Value

| Comment          | Host |   | Equipment | Comment        |
|------------------|------|---|-----------|----------------|
| 上層主機要求狀態變數(SV)資料 | S1F3 | → |           |                |
|                  |      | ← | S1F4      | 設備上傳狀態變數(SV)資料 |

表 6.7 Host Requests Status Variable Namelist

| Comment                     | Host  |   | Equipment | Comment      |
|-----------------------------|-------|---|-----------|--------------|
| 上層主機要求狀態變數名稱清單(SV Namelist) | S1F11 | → |           |              |
|                             |       | ← | S1F12     | 設備上傳狀態變數名稱清單 |

表 6.8 說明上層主機可以要求設備上傳特定一筆設備常數(Equipment Constant)的資料內容,也可以依照表 6.9 的劇本請求設備上傳設備常數名稱清單(EC Namelist)。

表 6.8 Host Requests Equipment Current Constant Values

| Comment      | Host  |   | Equipment | Comment    |
|--------------|-------|---|-----------|------------|
| 上層主機要求設備常數資料 | S2F13 | → |           |            |
|              |       | ← | S2F14     | 設備上傳設備常數資料 |

表 6.9 Host Requests Equipment Constant Namelist

| Comment        | Host  |   | Equipment | Comment      |
|----------------|-------|---|-----------|--------------|
| 上層主機要求設備常數名稱清單 | S2F29 | → |           |              |
|                |       | ← | S2F30     | 設備上傳設備常數名稱清單 |

### 6.1.2.2.3 Trace Data Collection (追蹤資料收集)

表 6.10 說明的是追蹤資料收集的劇本，根據此劇本 HOST 可以監控 CTC 狀態變數值的變化、設定資料收集的 SVID 項目與取樣週期後下傳 S2F23 訊息給 CTC，當 CTC 收到 S2F23 後依取樣週期與取樣次數，週期性的以 S6F1 上傳資料數據給 HOST，而 HOST 可將收集到的資料存到資料庫並轉換成各種圖表，作為製程改善的參考依據。

表 6.10 Trace Data Collection

| Comment                    | Host  |   | Equipment | Comment                        |
|----------------------------|-------|---|-----------|--------------------------------|
| 上層主機要求設備控制器上傳 Trace Report | S2F23 | → |           |                                |
|                            |       | ← | S2F24     | 收到命令、建立 Trace Report，並回覆給 HOST |
|                            |       | ← | S6F1      | 週期性上傳對應SVID的SV值                |
| 回覆資料Trace Report已收到        | S6F2  | → |           |                                |
|                            |       | ← | S6F11     | Trace Report結束送出事件報告           |
| 回覆訊息已收到                    | S6F12 | → |           |                                |

### 6.1.2.2.4 ON-LINE Identification (ON-LINE 狀態確認)

表 6.11 說明 CTC 與 HOST 可以送出 S1F1 訊息來詢問的對方連線是否為 ON-LINE 狀態，當回覆 S1F2 訊息時需要連結變數 MDLN 與 SOFTREV 的資料，MDLN 表示的是設備型號，SOFTREV 則是代表軟體版本。

表 6.11 Host Initiated

| Comment    | Host |   | Equipment | Comment                 |
|------------|------|---|-----------|-------------------------|
| 測試連線狀態是否正常 | S1F1 | → |           |                         |
|            |      | ← | S1F2      | 設備回覆上層主機 (MDLN、SOFTREV) |

### 6.1.2.3 警報管理 (Alarm Management)

本節說明警報管理的方式與訊息傳遞的劇本，表 6.12 說明當設備端發生警報時，設備控制器利用 S5F1 訊息來上傳 Alarm 報告，因為 Alarm 的發生也是一種事件的發生，因此設備端控制器必須再上傳 S6F11 事件報告給上層主機。

表 6.12 Send Alarm Report

| Comment    | Host  |   | Equipment | Comment               |
|------------|-------|---|-----------|-----------------------|
|            |       | ← | S5F1      | 當Alarm發生時，設備送出Alarm報告 |
| 上層主機回覆確認訊息 | S5F2  | → |           |                       |
|            |       | ← | S6F11     | 送出Alarm發生的事件報告        |
| 上層主機回覆確認訊息 | S6F12 | → |           |                       |



### 6.1.2.4 製程配方管理 (Process Recipe Management)

本節說明製程配方管理的訊息劇本，表 6.13、表 6.14、表 6.15、表 6.16、表 6.17 分別說明上層主機與設備進行建立、編輯、刪除製程配方的劇本。

表 6.13 Process Program Creating, Editing, or Deleting on Equipment

| Comment    | Host  |   | Equipment | Comment                    |
|------------|-------|---|-----------|----------------------------|
|            |       | ← | S6F11     | 當對製程配方進行建立、編輯和刪除時，會回報事件報告。 |
| 上層主機回覆確認訊息 | S6F12 | → |           |                            |

表 6.14 Deleting a Process Program by Host

| Comment      | Host  |   | Equipment | Comment   |
|--------------|-------|---|-----------|-----------|
| 上層主機要求刪除製程配方 | S7F17 | → |           |           |
|              |       | ← | S7F18     | 確認製程配方已刪除 |

表 6.15 Host Requests Equipment Process Program Directory

| Comment        | Host  |   | Equipment | Comment       |
|----------------|-------|---|-----------|---------------|
| 上層主機要求上傳製程配方清單 | S7F19 | → |           |               |
|                |       | ← | S7F20     | 設備控制器上傳製程配方清單 |

表 6.16 Equipment-initiated Process Program Uploading

| Comment      | Host  |   | Equipment | Comment     |
|--------------|-------|---|-----------|-------------|
| 上層主機要求上傳製程配方 | S7F25 | → |           |             |
|              |       | ← | S7F26     | 設備控制器上傳製程配方 |

表 6.17 Host-initiated Process Program Downloading

| Comment        | Host  |   | Equipment | Comment     |
|----------------|-------|---|-----------|-------------|
| 詢問設備製程配方是否可以下傳 | S7F1  | → |           |             |
|                |       | ← | S7F2      | 同意下傳        |
| 開始下傳製程配方       | S7F23 | → |           |             |
|                |       | ← | S7F24     | 確認製程配方已收到   |
|                |       | ← | S7F27     | 回報製程配方的檢查結果 |
| 回覆已收到檢查結果      | S7F28 | → |           |             |

### 6.1.2.5 材料搬運 (Material Movement)

表 6.18 說明當 Wafer 在搬運時 CTC 會產生事件報告上傳到 HOST，例如：Wafer 進入製程模組時，CTC 會上傳事件報告給 HOST；當晶圓製程後要離開製程模組時，CTC 也會上傳事件報告給 HOST，告知 HOST 目前的設備狀況。

表 6.18 Material Movement

| Comment     | Host  |   | Equipment | Comment                            |
|-------------|-------|---|-----------|------------------------------------|
|             |       | ← | S6F11     | 當製程模組收到或送出 Wafer 時<br>控制器會主動送出事件報告 |
| 上層主機回覆訊息已收到 | S6F12 | → |           |                                    |

### 6.1.2.6 設備終端機服務(Equipment Terminal Services)

本節說明設備終端機服務的功能，在表 6.19、表 6.20 中說明 CTC 與 HOST 間可以透過此劇本的功能相互傳遞文字訊息，其功能類似一般的線上即時通訊軟體。HOST 與 CTC 間可以收到對方輸入的文字訊息，當 CTC 收到 HOST 的文字訊息後，會主動發出一個事件報告給 HOST。而 i-SECS 控制器經由擷取訊息物件可以得到文字訊息的內容，並且實作主動傳遞文字訊息給設備控制器的功能。

表 6.19 Host Sends Information to Equipment

| Comment          | Host  |   | Equipment | Comment        |
|------------------|-------|---|-----------|----------------|
| 上層主機送出文字訊息給設備控制器 | S10F3 | → |           |                |
|                  |       | ← | S10F4     | 設備控制器收到文字訊息    |
|                  |       | ← | S6F11     | 訊息文字內容已顯示在CTC上 |
| 上層主機回覆訊息已收到      | S6F12 | → |           |                |

表 6.20 Operator Sends TEXT Information to Host

| Comment     | Host  |   | Equipment | Comment     |
|-------------|-------|---|-----------|-------------|
|             |       | ← | S10F1     | 設備控制器送出文字訊息 |
| 上層主機回覆訊息已收到 | S10F2 | → |           |             |

### 6.1.2.7 訊息格式錯誤回報 (Error Message)

當 HOST 與 CTC 之間的訊息傳遞失敗或發生逾時(Timeout)時，會發出「訊息格式錯誤報告」Stream 9 的訊息，讓對方控制器知道訊息傳遞失敗，並作相關的處置。表 6.21、表 6.22、表 6.23、表 6.24、表 6.25 說明當上層主機下傳的訊息有錯誤時，設

備控制器回應錯誤訊息的劇本。一般來說，訊息格式檢查的項目有：Device ID、Stream Type、Function Type、Data Format 以及 T3 Timeout 等。

表 6.21 Message Fault due to Unrecognized Device ID

| Comment  | Host |   | Equipment | Comment       |
|----------|------|---|-----------|---------------|
| 上層主機下傳訊息 | SxFy | → |           |               |
|          |      | ← | S9F1      | Device ID未被認可 |

表 6.22 Message Fault due to Unrecognized Stream Type

| Comment  | Host |   | Equipment | Comment          |
|----------|------|---|-----------|------------------|
| 上層主機下傳訊息 | SxFy | → |           |                  |
|          |      | ← | S9F3      | Stream Type 未被認可 |

表 6.23 Message Fault due to Unrecognized Function Type

| Comment  | Host |   | Equipment | Comment            |
|----------|------|---|-----------|--------------------|
| 上層主機下傳訊息 | SxFy | → |           |                    |
|          |      | ← | S9F5      | Function Type 未被認可 |

表 6.24 Message Fault due to Unrecognized Data Format

| Comment  | Host |   | Equipment | Comment          |
|----------|------|---|-----------|------------------|
| 上層主機下傳訊息 | SxFy | → |           |                  |
|          |      | ← | S9F7      | Data Format 未被認可 |

表 6.25 Message Fault due to Transaction Timer Timeout

| Comment  | Host |   | Equipment | Comment    |
|----------|------|---|-----------|------------|
| 上層主機下傳訊息 | SxFy | → |           |            |
|          |      | ← | S9F9      | T3 timeout |

### 6.1.2.8 時間調整 (Clock)

本節說明上層主機與設備控制器的時間調整方式，表 6.26、表 6.27、表 6.28 說明 CTC 與 HOST 間可以經由時間調整的劇本，由 CTC 主動要求 HOST 下傳正確的時或由 HOST 下傳訊息要求 CTC 立即更正時間，避免因為時間不一致而導致擷取到的訊息資料產生誤差。i-SECS 控制器在設計上並不主動要求調整時間，僅被動的等待接收到時間調整訊息時才更改 i-SECS 控制器的時間。

表 6.26 Equipment Requests Time Value from HOST

| Comment | Host  |   | Equipment | Comment           |
|---------|-------|---|-----------|-------------------|
|         |       | ← | S2F17     | 設備控制器要求上層主機下傳正確時間 |
| 下傳正確時間  | S2F18 | → |           |                   |

表 6.27 Host Requests Equipment to Set Time

| Comment  | Host  |   | Equipment | Comment                     |
|----------|-------|---|-----------|-----------------------------|
| 上層主機下傳時間 | S2F31 | → |           | 格式如下:<br>14:49:58.55(時分秒毫秒) |
|          |       | ← | S2F32     | 回覆收到，並重新設定時間                |

表 6.28 Host Requests Time Value from Equipment

| Comment    | Host  |   | Equipment | Comment   |
|------------|-------|---|-----------|-----------|
| 上層主機要求設備時間 | S2F17 | → |           |           |
|            |       | ← | S2F18     | 上傳設備控制器時間 |

### 6.1.2.9 控制狀態圖的遷移

本節說明設備控制器與上層主機間控制狀態遷移的劇本。上層主機可以藉由特定的訊息劇本下達命令來要求設備控制器改變控制狀態，而設備控制器可以經由手動操作方式通知上層主機控制狀態已改變。表 6.29 說明 CTC 手動切換狀態從 OFF-LINE 變成 ON-LINE。表 6.30 說明 HOST 拒絕設備端 OFF-LINE 變成 ON-LINE 的要求。表 6.31、表 6.32、表 6.33、表 6.34 則分別說明上層主機改變連線控制狀態的訊息溝通方式。i-SECS 控制器依據 CTC 上傳 HOST 的事件報告來判斷目前的控制狀態。為避免控制狀態與上層主機衝突，因此 i-SECS 控制器設計只能接收狀態的改變，不能主動去要求更改 CTC 設備狀態。

表 6.29 Host Accepts ON-LINE Request from Equipment

| Comment     | Host  |   | Equipment | Comment                          |
|-------------|-------|---|-----------|----------------------------------|
|             |       | ← | S1F1      | 手動操作使設備控制器從 OFF-LINE 切換到 ON-LINE |
| 上層主機回覆訊息已收到 | S1F2  | → |           |                                  |
|             |       | ← | S6F11     | 控制狀態改變送出事件報告                     |
| 上層主機回覆訊息已收到 | S6F12 | → |           |                                  |

表 6.30 Host Refuses ON-LINE Request

| Comment           | Host |   | Equipment | Comment                          |
|-------------------|------|---|-----------|----------------------------------|
|                   |      | ← | S1F1      | 手動操作使設備控制器從 OFF-LINE 切換到 ON-LINE |
| 上層主機拒絕 ON-LINE 要求 | S1F0 | → |           |                                  |

表 6.31 Operator Attempts to Enter OFF-LINE, REMOTE and LOCAL

| Comment     | Host  |   | Equipment | Comment      |
|-------------|-------|---|-----------|--------------|
|             |       | ← | S6F11     | 控制狀態改變送出事件報告 |
| 上層主機回覆訊息已收到 | S6F12 | → |           |              |

表 6.32 Host Requests to Enter OFFLINE

| Comment                | Host  |   | Equipment | Comment                |
|------------------------|-------|---|-----------|------------------------|
| 上層主機要求控制狀態切換為 OFF-LINE | S1F15 | → |           |                        |
|                        |       | ← | S1F16     | 切換成 OFF-LINE 狀態後回覆確認訊息 |
|                        |       | ← | S6F11     | 控制狀態改變後送出事件報告          |
| 上層主機回覆訊息已收到            | S6F12 | → |           |                        |

表 6.33 Host Attempts Enter ONLINE.

| Comment               | Host  |   | Equipment | Comment               |
|-----------------------|-------|---|-----------|-----------------------|
| 上層主機要求控制狀態切換為 ON-LINE | S1F17 | → |           |                       |
|                       |       | ← | S1F18     | 切換成 ON-LINE 狀態後回覆確認訊息 |
|                       |       | ← | S6F11     | 控制狀態改變後送出事件報告         |
| 上層主機回覆訊息已收到           | S6F12 | → |           |                       |

表 6.34 Host Confirms Equipment Presence

| Comment    | Host |   | Equipment | Comment                           |
|------------|------|---|-----------|-----------------------------------|
| 測試連線狀態是否正常 | S1F1 | → |           |                                   |
|            |      | ← | S1F2      | 設備回覆上層主機(包括 ECID 項目 MDLN、SOFTREV) |

## 6.2 i-SECS 與 CTC 之整合測試

### 6.2.1 連線測試硬體介紹

表 6.35 列出實機驗證時系統的硬體規格說明，在進行連線測試時 CTC 的執行環境是在實驗室內的工業電腦上執行；i-SECS 控制器和 HOST 端的 Emulator 則在同一部個人電腦上執行，i-SECS 與 Emulator 間利用本地端位址(Localhost) 127.0.0.1 給定本機 Port 來進行連線，工業電腦與個人電腦間則透過區域網路來連線，實際的連線圖請參考圖 5.1 i-SECS 硬體架構圖。

表 6.35 連線測試硬體規格說明表

|               |      | CTC                        | i-SECS                     | HOST(Emulator)             |
|---------------|------|----------------------------|----------------------------|----------------------------|
| 系統執行時<br>硬體需求 | CPU  | PIII-1.3GHz                | P4-2.4GHz                  | P4-2.4GHz                  |
|               | 網路卡  | 100Mbps                    | 100Mbps                    | 100Mbps                    |
|               | 記憶體  | 512M DRAM                  | 512M RAM                   | 512M RAM                   |
|               | 作業系統 | Windows XP<br>Professional | Windows XP<br>Professional | Windows XP<br>Professional |
|               | 軟體版本 | CTC(GEM) v1.0              | i-SECS v1.0                | SECS<br>Emulator v1.5      |
| 軟體開發環境        |      | JBuilder 9<br>Personal     | JBuilder 9<br>Personal     | C 語言                       |

### 6.2.2 連線測試軟體工具介紹

在做 i-SECS 控制器實機連線驗證前，我們先介紹兩種連線測試的軟體工具：

1. **SECS Emulator**：i-SECS 控制器在開發時，我們以工研院機械所所發展的通訊測試軟體 SECS Emulator v1.5 [21] 來做連線測試，如圖 6.2 右上角的 HOST 端模擬器即是 Emulator，此軟體主要的功能為測試設備間的通訊功能，其特點為：

- 支援 SECS-I/SECS-II/HSMS-SS 通訊協定
- Screen Logging 及 File Logging 能力
- 支援 SML 格式

2. **CTC(GEM)程式**：i-SECS控制器開發完成後，我們以曹洪泰同學所實作的 CTC(GEM)程式來做連線測試。CTC程式在加上GEM的功能後，除了可以執行晶圓的排程、事件報告、配方管理等工作外，也可以將製程中的參數、晶圓傳輸等資訊以特定的訊息劇本上傳到上層主機，並接受上層主機的操作命令完成相關的工作。以CTC(GEM)程式來測試i-SECS控制器可以完整的擷取設備與上層主機間所傳遞的訊息、監控設備內部晶圓的傳輸工作、收集設備上傳上層主機的製程資訊以及事件報告等功能。

### 6.2.3 i-SECS 與 CTC 實機測試結果

本節以實驗室內現有的集束型製程設備來驗證i-SECS控制器的上層主機通訊介面劇本，目前主要驗證的功能有：

1. **事件報告機制**：集束型製程設備在進行晶圓輸送排程時，製程模組(PMC)和傳輸模組(TMC)會上傳事件報告給集束型製程設備控制器(CTC)，告知目前製程的操作進度；CTC 在接收到事件報告後，以上層主機通訊介面(GEM)所規範的訊息劇本上傳到上層主機，讓上層主機也可以了解目前機台的運作狀況。如圖 6.2 所示的是當 CTC 在接收到 TMC 或 PMC 上傳的事件報告後會回報給上層主機，i-SECS 在接收到 CTC 上傳的 S6F11 訊息後會以 SML 語法記錄訊息內容並在操作介面上改變 Monitor 畫面與記錄事件報告，最後再將訊息轉傳到上層主機，等待上層主機回覆確認訊息。

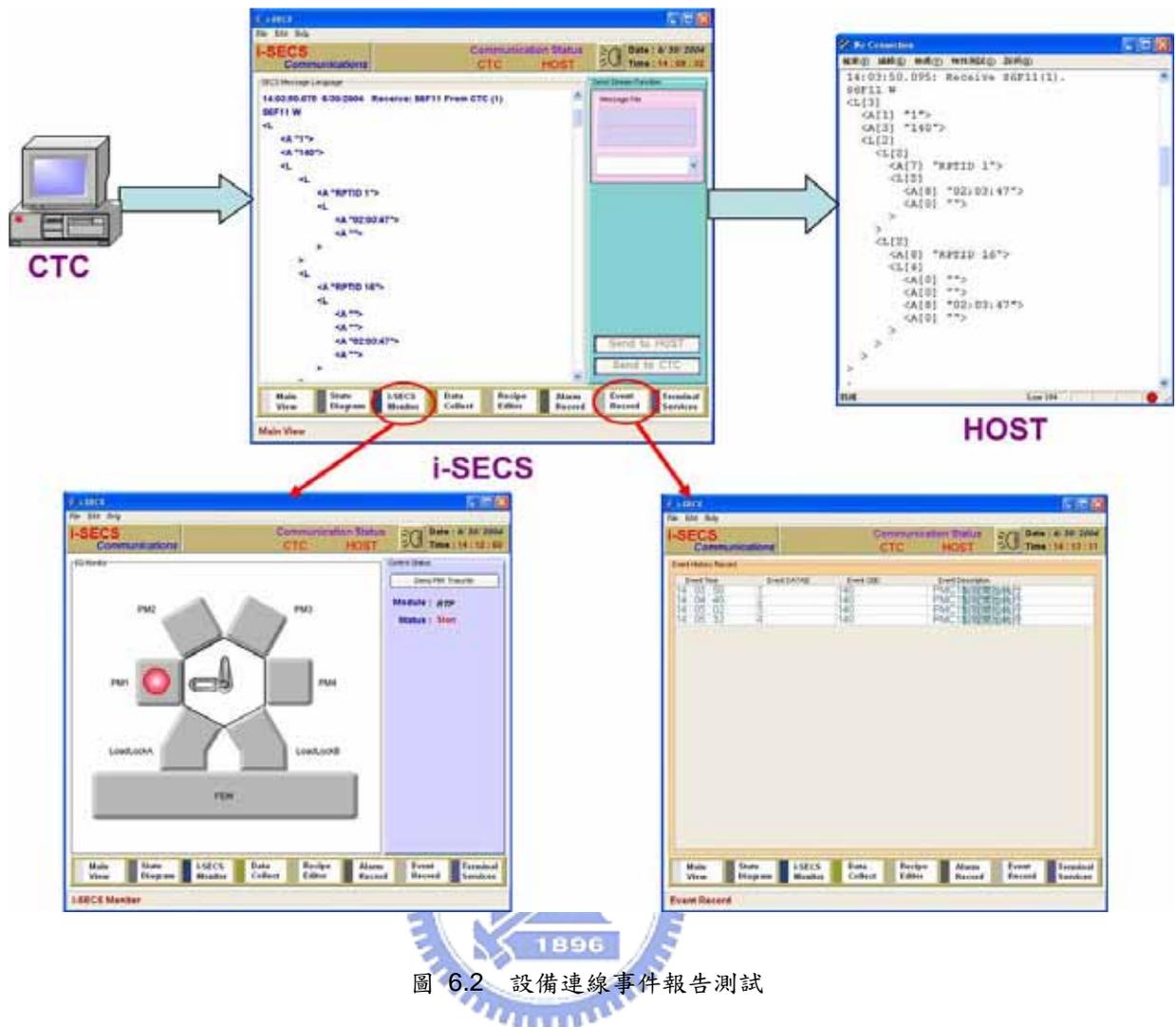


圖 6.2 設備連線事件報告測試

2. **製程配方管理機制**：製程配方管理機制主要的目的是驗證 CTC 與上層主機間可否完成上傳、下載、建立與刪除製程配方的功能，讓上層主機可以透過製程配方管理的劇本，決定 CTC 的製程參數。如圖 6.3 所示的是製程配方管理的操作畫面，當按下 Connect Recipe 的按鈕後會開啟一個資料庫連結的畫面，此畫面的目的是希望以手動的方式連結到製程配方的資料庫中；在圖 6.4 中所示的是 i-SECS 允許操作者建立或刪除特定的製程配方資料表，當按下 Create Recipe 的按鈕後會出現一個建立製程配方的畫面，可以在畫面上新增一筆製程配方；當按下 Delete Recipe 的按鈕後，會出現一個刪除製程配方的確認畫面，確認是否要刪除該筆製程配方；當按下  的按鈕後可以在畫面上顯示資料庫中

該筆製程配方的內容，當按下 Download Recipe 按鈕後，i-SECS 會將所點選到的製程配方資料內容下傳到設備控制器，完成製程配方下傳的功能。

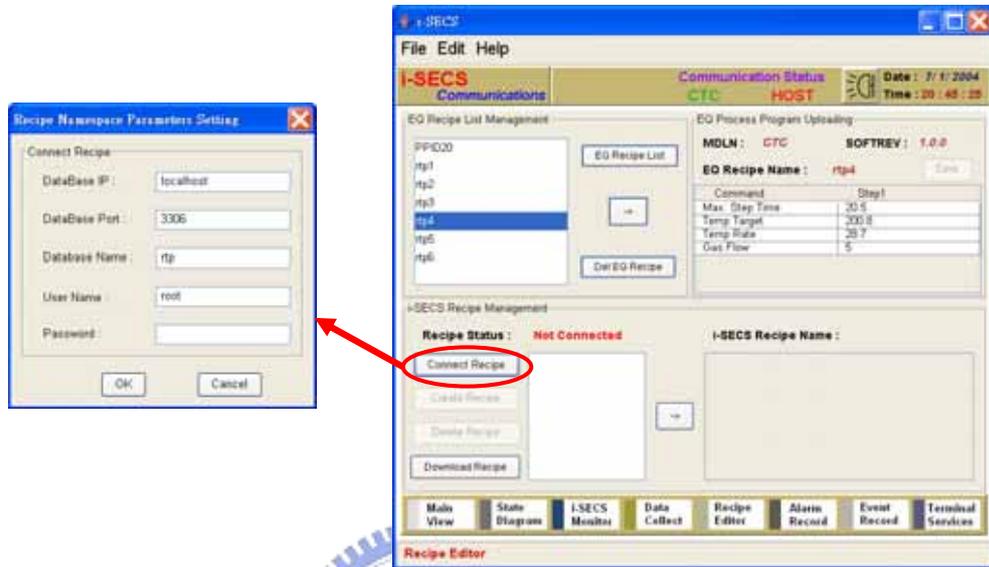


圖 6.3 製程配方管理資料庫連結

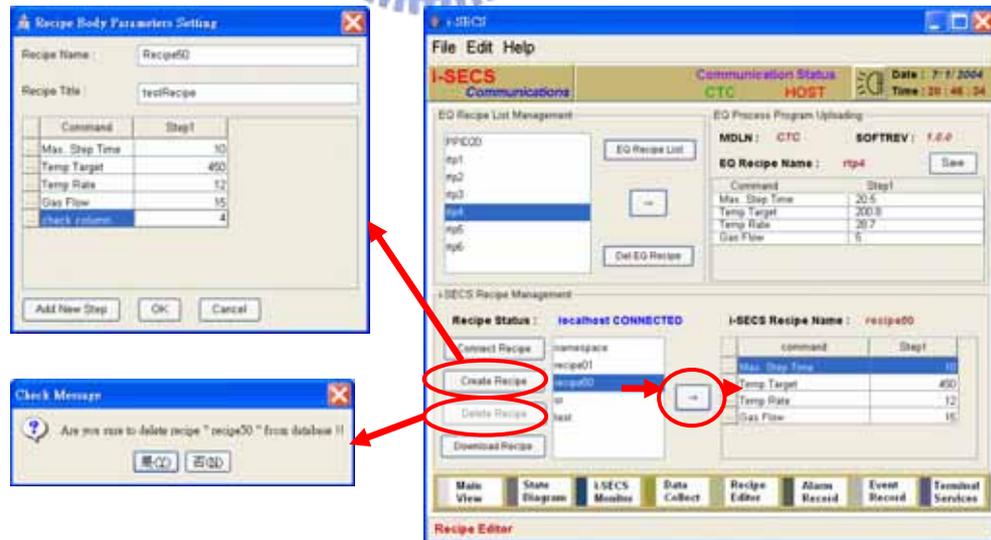


圖 6.4 製程配方管理建立與刪除資料表

3. **資料收集機制**：本節主要在驗證 6.1.2.2 節的三個資料收集劇本，上層主機透過資料收集劇本可以得知設備控制器的變數資料。這裡所測試的資料收集機制分為三種：事件資料收集(Event Data Collection)、變數資料收集(Variable Data Collection)以及追蹤資料收集(Trace Data Collection)等劇本。

**a. 事件資料收集**

圖 6.5 所示的是 i-SECS 控制器驗證上層主機透過事件資料收集劇本得到 CTC 中事件報告的資料連結，右上方的(Request Event Report)部份代表了 Event Report CEID 101 這筆資料內包含了兩筆 Report ID，分別為 RPTID 1 及 RPTID 2，其中的 RPTID 2 內所掛載的變數資料分別為 30 與 10。右下方的(Event Reporting Setup)部份則是 i-SECS 手動設定 CTC 的 CEID、RPTID 與 VID 之間的連結關係。當選定好 CEID 所應連結的 RPTID 或 RPTID 所應連結的 VID 後，按下  按鈕後便將之間的連結關係下傳到 CTC，並等待 CTC 回覆確認訊息，完成事件報告的設定。



圖 6.5 事件資料收集測試

## b. 變數資料收集

圖 6.6 所示的是i-SECS操作畫面上列出CTC的SVID(Status Variable ID)、ECID(Equipment Constant ID)所分別連結的變數資料以及名稱清單列表(Namelist)。上方的(Status Variable Requests)部份是由i-SECS主動要求CTC上傳SVID 101、102、103、104 所連結的狀態變數值，CTC回傳此四筆SVID的狀態變數值分別為 09:12:53、30、10 與DATAID104。在右邊部份則是顯示對應此四筆SVID的狀態變數名稱清單(SV Namelist)，其目的是列出SVID對應的狀態變數名稱(SVNAME)與其表示的單位(UNITS)；同理，下方的(Equipment Constant Requests)部份則是i-SECS要求CTC上傳的設備常數值以及設備常數名稱清單，在右邊的設備常數名稱清單部份列出對應ECID的設備常數名稱(ECNAME)、設備常數最小值(ECMIN)、設備常數最大值(ECMAX)、設備常數預設值(ECDEF)以及設備常數對應的單位(UNITS)等設備常數資料。

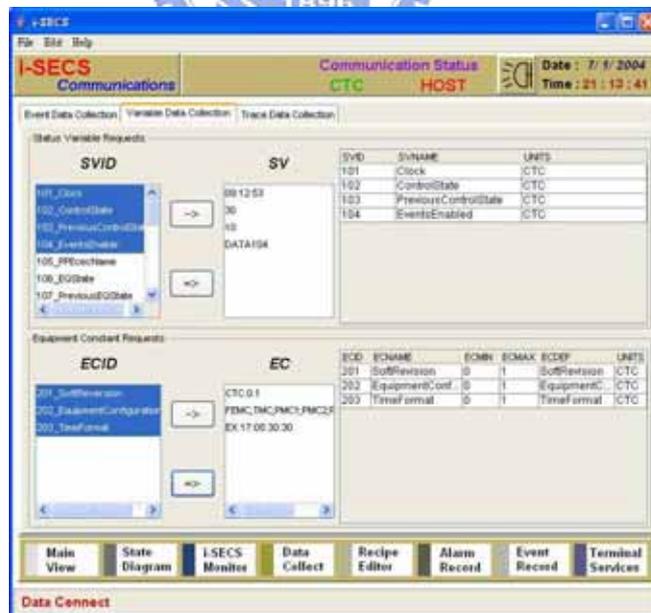


圖 6.6 變數資料收集測試

### C. 追蹤資料收集

如圖 6.7 所示的是追蹤資料收集劇本測試的結果，上方的(Trace Data Collection)部份可以由 i-SECS 先設定好追蹤報告的 TRID(Trace Report ID)、取樣週期(Sample Period)、取樣次數(Total Samples)以及 SVID 的連結後，透過追蹤資料收集的劇本要求 CTC 週期性上傳所要追蹤的變數資料。如圖中所示，i-SECS 要求 CTC 週期性的上傳 SVID 101、102、103、104 等四筆資料的狀態變數值，當 CTC 收到追蹤報告開始的訊息(S2F23)後，會依據追蹤的取樣週期及取樣次數上傳追蹤資料，在右方的表格即是 CTC 週期性上傳的狀態變數資料，由 SVID 101 所連結到的狀態資料值(Clock)可以看出 CTC 一共上傳了五筆資料，每筆資料間隔 10 秒，i-SECS 在接收到資料收集的訊息(S6F1)後，會將狀態變數資料儲存在資料庫中，以做為後續發展先進製程控制(APC)時所需的參數資料。圖 6.7 下方的溫度(T)-時間(t)曲線圖則是將所收集到的製程溫度資料呈現在使用者介面上，可以即時的了解目前製程的狀況。

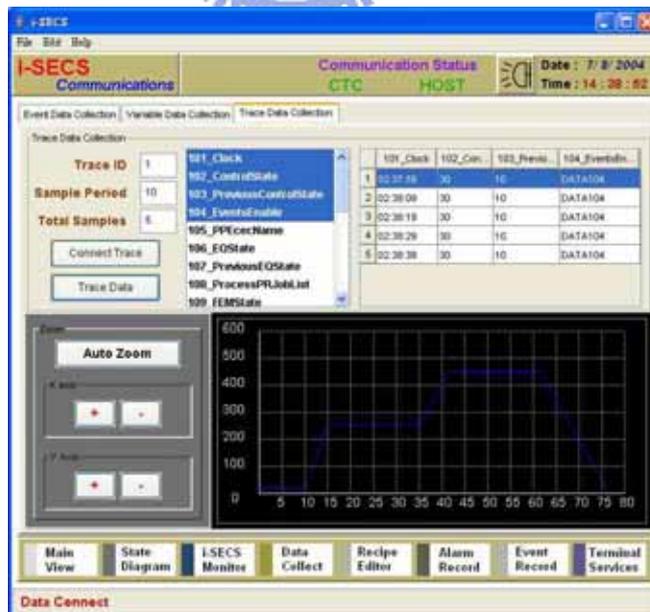


圖 6.7 追蹤資料收集測試

## 6.2.4 測試結論

在經過系統測試實機驗證後，我們實際來比較一下 i-SECS 控制器與工研院 Emulator 軟體在使用上的差異。在應用上，Emulator 軟體的操作非常方便，Emulator 在進行連線測試時，可以藉由讀取 SML 的檔案，送出訊息來進行連線測試的功能。i-SECS 定義為軟體控制器，同時也提供通訊測試的功能，單獨執行的 i-SECS 控制器同樣可以藉由讀取 SML 的檔案來進行連線測試，i-SECS 連線測試的優點是可以同時測試兩個以上的設備控制器。當 CTC 與 i-SECS 及 HOST 端連線完成後，i-SECS 可以選擇送出測試訊息到 CTC 端或 HOST 端，因此在使用上較 Emulator 多樣化。

因為 i-SECS 是利用 Java 程式以物件導向的方式來實作，因此在程式執行時，訊息接收與送出的編碼解碼時間較 Emulator 來的長。就功能來分析，i-SECS 已經具備 Emulator(除 SECS-I 外)的連線測試功能，下表 6.36 列出 i-SECS 與 Emulator 的功能比較，由表中可以清楚的比較出 i-SECS 的差異及其應用。

表 6.36 i-SECS 與 Emulator 的比較

|           | i-SECS  | Emulator  |
|-----------|---|---|
| <b>特色</b> | <ul style="list-style-type: none"> <li>• 具有機台連線測試的功能</li> <li>• 可建立多方連線，同時連線多個不同的控制器</li> <li>• HSMS/SECS-II 通訊介面</li> <li>• Screen Logging &amp; Data Logging</li> <li>• 支援 SML 語法格式</li> <li>• 可對機台做連線控制</li> <li>• 可連結資料庫，儲存接收到的訊息資料內容</li> <li>• 可以連線 CTC 與上層主機控制器，具備訊息轉傳的功能</li> <li>• 利用 Java 程式語言來實作，以物件導向的方式來建立與接收訊息</li> </ul> | <ul style="list-style-type: none"> <li>• 具有機台連線測試的功能</li> <li>• 支援 SECS-I/SECS-II/HSMS 通訊協定</li> <li>• Screen Logging &amp; File Logging</li> <li>• 支援 SML 格式</li> <li>• 具備訊息自動回傳的功能</li> <li>• 根據訊息的資料結構來解譯，不需要事先建立訊息物件</li> </ul> |

## 第7章 i-SECS 在光電製程設備上的應用規劃

在半導體領域的製程設備中，由於 SEMI 規範了共通的標準介面，因此所有符合 SEMI Standard 的製程設備均使用相同的訊息內容以及通訊介面來溝通；在光電領域的製程設備中，由於目前尚未訂定共通的標準，因此各製造廠或設備廠都會自行定義通訊介面來溝通。在前面幾章我們分別介紹了通訊模組架構、i-SECS 控制器的設計以及實機連線系統測試等，目前 i-SECS 控制器的應用主要是在實驗室內的集束型半導體製程設備上，為了要達到 i-SECS 控制器也能在光電製程設備上使用的目的，因此我們以實際已經建立好的 i-SECS 程式來做範例，規劃並說明如何將目前的 i-SECS 控制器程式修改成在光電製程設備上可用的控制器模組，以下我們分別介紹目前已建立的光電製程設備通訊模組，並將規劃出應用在光電製程設備的 i-SECS 控制器程式。



### 7.1 通訊模組在光電製程設備上的應用

光電製程設備通訊模組在開發時，我們以取放玻璃基板的 Damper 通訊規格訂定訊息格式，建立應用在此類型製程設備上的光電通訊模組程式。該通訊模組一共建立 31 個訊息物件，其訊息物件名稱雖與半導體製程設備的通訊介面一致，但實際的訊息內容卻大不相同。

表 7.1 舉例說明在不同領域上 S5F1 訊息的內容比較，在半導體與光電製程設備上，S5F1 的訊息皆表示當設備端發生警報時，設備控制器上傳警報管理的訊息；同理，表 7.2 所示的 S6F11 訊息皆表示設備控制器上傳上層主機事件報告時的訊息，在不同領域的應用上，該訊息的意涵雖相同，但是訊息內容架構卻大為不同。通訊模組在不同領域應用上訊息內容的差異很大，因此我們在建立訊息物件時，必須將兩個不同領域的通訊模組分開建立，各自發展對應訊息物件的通訊模組程式。

表 7.1 S5F1 訊息在半導體與光電製程設備上的比較

| 半導體製程設備表示法  | 光電製程設備表示法   |
|---|---|
| <b>S5F1</b><br>L,3<br>1. <ALCD><br>2. <ALID><br>3. <ALTX> | <b>S5F1</b><br>L,10<br>1. <Data ID><br>2. <Line ID><br>3. <Tool ID><br>4. <Unit ID><br>5. <Date Time><br>6. <Category><br>7. <Event Flag><br>8. <AW code><br>9. <Unit code><br>10. <AW Description> |

表 7.2 S6F11 訊息在半導體與光電製程設備上的比較

| 半導體製程設備表示法  | 光電製程設備表示法  |
|---|--|
| <b>S6F11</b><br>L,3<br>1. <DATAID><br>2. <CEID><br>3. L,a<br>1. L,2<br>1. <RPTID <sub>1</sub> ><br>2. L,b<br>1. <V <sub>1</sub> ><br>.<br>.<br>b. <V <sub>b</sub> ><br>.<br>.<br>a. L,2<br>1. <RPTID <sub>a</sub> ><br>2. L,c<br>1. <V <sub>1</sub> ><br>.<br>.<br>b. <V <sub>c</sub> > | <b>S6F11 (CEID=1)</b><br>L,4<br>1. <CEID><br>2. <Report ID><br>3. <Data ID><br>4. L,9<br>1. <Line ID ><br>2. <Tool ID ><br>3. <Unit ID ><br>4. <Date Time><br>5.<Tool STAT><br>6. <Insp Mode><br>7. <Tool Mode><br>8. <Current Recipe ID><br>9. <Current CST Size> |

表 7.3 所述的是應用在光電製程設備上 Damper 機台的訊息摘要說明表，表中說明各訊息內容的敘述，依據這些訊息我們建立了一套可應用在光電製程設備 Damper 機台上的通訊模組軟體。

表 7.3 光電製程設備 Damper 機台訊息摘要說明

| Primary | Reply | Description   |
|---------|-------|---|
| S1F1    | S1F2  | Are You There / Reply                               |
| S1F67   | S1F68 | Process Start Report for Inline                     |
| S1F69   | S1F70 | Process End Report for Inline                       |
| S1F71   | S1F72 | Process Data Request / DownLoad for Batch           |
| S1F77   | S1F78 | Process Start / End Report for Batch                |
| S1F79   | S1F80 | Work Removal Report(Batch Only)                     |
| S1F81   | S1F82 | Cassette Removal Report(Batch Only)                 |
| S2F17   | S2F18 | Data and Time Request / Reply                       |
| S5F1    | S5F2  | Alarm / Warning Report                              |
| S6F3    | S6F4  | Process Data Report                                 |
| S6F9    | S6F10 | Dummy Work Request                                  |
| S6F11   | S6F12 | Equipment Status Report(CEID=1)                     |
| S6F11   | S6F12 | Unit Status Report(CEID=2)                          |
| S6F11   | S6F12 | Port Status Report(CEID=3)                          |
| S7F71   | S7F72 | Process Data Request / DownLoad for Inline          |
| S7F73   | S7F74 | Asm Product ID Code Request / DownLoad for Asm Line |
| S7F75   | S7F76 | Unloader Process Data Request / DownLoad for Inline |
| S9F9    | --    | Transaction Timeout                                 |

## 7.2 i-SECS 在光電製程設備上的應用規劃

7.1 節說明應用在半導體與光電製程設備上通訊模組訊息的差異，且已經建立好應用在光電製程設備上的通訊模組核心程式，本節將說明如何將 i-SECS 控制器修改成符合光電領域所需的控制器架構。

在 6.2.4 一節中提到，i-SECS 本身具備有通訊連線測試的功能，當程式的控制邏輯尚未建立前，i-SECS 仍舊可以透過讀入 SML 檔來進行機台的連線測試，另一個功能是可以將 i-SECS 架構在設備控制器(Block Controller)與上層主機控制器(HOST)間，如圖 3.8 光電產業製程設備控制器軟體架構圖所示，來進行機台的連線測試，i-SECS 因為支援 SML 語法，因此可以在使用者介面上顯示出所有訊息傳遞的內容，作為連線測試的參考依據，此部份為 i-SECS 在半導體與光電製程設備應用上共通的功能。

在 5.1 節中，我們提到將 i-SECS 分為四個不同的功能模組，分別為通訊模組、控制模組、GUI 模組以及資料庫處理模組，當 i-SECS 要擴充到光電的製程設備上時，我們規劃了幾個修改 i-SECS 控制器的步驟：

1. 首先，取得該製程設備的所有訊息格式，將訊息物件編譯到通訊模組核心程式內，編譯完訊息物件後，通訊核心程式已經可以達到收送訊息的功能；
2. 修改 i-SECS 使用者介面，要在光電產業的製程設備上應用，需要針對光電製程設備的特性，來建立所要呈現的操作畫面，同時規劃介面間的功能；
3. 更改控制模組的程式，因為在不同領域的應用下，控制模組的控制判斷邏輯會有差異，因此依據訊息格式規範以及訊息傳遞的劇本來更改 i-SECS 控制器的控制模組，達到符合前兩項建立訊息格式與修改使用者介面的功能；
4. 在資料庫處理模組部份，依據設備控制器的需求，建立資料庫連結，並將 i-SECS 控制器程式連結到資料庫系統中。
5. 實機連線測試，檢查上述四項步驟是否正確。

## 第8章 結論與未來展望

### 8.1 結論

本文提出利用 UML 模型語言，以物件導向的方式來分析與設計獨立介面通訊模組控制器軟體，並利用具有跨平台特性之 Java 語言來完成整個控制器程式的實作，使得 i-SECS 控制器具有：

- 遵循 SEMI E5 SECS-II 標準的規範，可以讓控制器間傳遞符合 SEMI Standard 共通的 SECS-II 格式的訊息；
- 遵循 SEMI E37 HSMS 標準的規範，可以讓控制器以 TCP/IP(Ethernet)的方式來傳遞 SECS-II 格式的訊息；
- 遵循 SEMI E30 GEM 標準的規範，可以讓 i-SECS 與設備控制器及上層主機間根據上層主機通訊介面劇本來傳遞訊息；允許上層主機可以參與設備的操作，以有效的整合製程設備到整廠自動化的系統中。
- 實作 SML 的編碼格式，可在使用者介面上呈現出訊息的 SML 語法，使控制器間所傳遞的訊息內容更易於理解與追蹤。
- 建立資料庫連結，可以透過上層主機通訊介面的劇本將製程中的資訊週期性的儲存到資料庫中，達到資料追蹤與分析的目的。

整個控制器軟體的設計是將各個功能模組化，以達到整合容易、擴充方便的軟體架構。而由於機台的限制，目前是以集束型製程設備控制器(CTC)跟上層主機通訊介面(GEM)來驗證 i-SECS 控制器的通訊功能，並整合 i-SECS 控制器到集束型半導體製程設備上，與 CTC、TMC、FEMC、PMC 等控制器模組做全系統自動化模式的測試工作。

## 8.2 未來展望

本文的最終目的是希望能將i-SECS控制器廣泛普及的應用在半導體及光電製程設備上，經由乙太網路(Ethernet)連線的方式擷取通訊訊息，達到設備控制與製程監控的目的。目前，我們已完成的測試是以實驗室內的集束型半導體製程設備來驗證i-SECS的功能；未來，希望能夠將i-SECS控制器整合到光電領域的製程設備上，透過兩種領域中共通的訊息傳遞特性，經由擷取到的製程資訊作分析，進而發展先進製程控制(Advanced Process Control, APC)，期能提高整體的設備有效利用率(Overall Equipment Effectiveness, OEE)。



## 參考文獻

- [1] Ernest J. Wood, An Object-Oriented SECS Programming Environment, IEEE TRANSACTIONS ON SEMICONDUCTOR MANUFACTURING , Vol. 6. NO. 2, MAY 1993), pp.119-127.
- [2] SEMI E4-0301, “SEMI Equipment Communications Standard 2 Message Transfer (SECS-I)”, 2002.
- [3] 李安謙教授，“集束型製程設備物件導向控制軟體製作測試驗證”，中山科學研究院 期末報告，2002。
- [4] 李妍慧，“集束型製程設備傳輸模組控制器及通訊模組設計與測試”，國立交通大學 機械工程研究所碩士論文，民 92 年。
- [5] SEMI E5-0301, “SEMI Equipment Communications Standard 2 Message Content (SECS-II)”, 2002.
- [6] SEMI E37, “High-Speed SECS Message Services (HSMS) Generic Services”, 2002.
- [7] SEMI E30, “Generic Model for Communications and Control of Manufacturing Equipment (GEM) ”, 2002.
- [8] Jing Dong, UML Extension for Design Patterns Compositions, Journal of Object Technology Vol.1, No.5, November-December 2002.
- [9] David A. Taylor, “Object Oriented Information System”, Wiley, 1992
- [10] Grady Booch, James Rumbaugh, Ivar Jacobson, “The Unified Modeling Language User Guide”, ADDISON-WESLEY, 1999.

- [11] Mary Campione, Kathy Walrath, Alison Huml, "The Java™ Tutorial Third Edition", ADDISON-WESLEY, 2001.
- [12] Wendy Boggs, Michael Boogs, "*Mastering UML with Rational Rose 2002*", Sybex, January 2002.
- [13] Borland, "JBuilder® 9 實用技術手冊", 碁峰, 2003.
- [14] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, "Design Patterns- Elements of Reusable Object-Oriented Software", ADDISON-WESLEY, 2001.
- [15] 顧鴻壽, "光電液晶平面顯示器-技術基礎及應用", 新文京, 2004。
- [16] 結城 浩著, 李于青譯, "Design Pattern 於 Java 語言上的實例應用", 博碩文化, 2002。
- [17] "GW Associates Product Manual", GW Associates.
- [18] 曹洪泰, "集束型製程設備與上層主機通訊介面之軟體設計與應用", 國立交通大學工學院精密與自動化專班碩士論文, 民 93 年。
- [19] 戶松 豐和著, 李于青譯, "Java2 物件導向技術專題—Design Patterns、Framework、Multithread、Concurrent", 博碩文化, 2002。
- [20] Darren Govoni, "Java Application Frameworks", WILEY, 1999
- [21] 工研院機械所, SECS Emulator v1.5, <http://secs.itri.org.tw/>
- [22] 李安謙教授, "具物件導向功能之集束型設備控制器整合測試驗證研究", 中山科學研究院期末報告, 2003。

## 附錄一 SECS-II 訊息明細

| Stream 1 : 設備狀態 |             |                        |            |
|-----------------|-------------|------------------------|------------|
|                 | Item        | FormatCode             | Type       |
| S1F1            | Header Only |                        |            |
| S1F2            | MDLN        | 20                     | String(20) |
|                 | SOFTREV     | 20                     | String(20) |
| S1F3            | SVID        | 20,30,50               | String(20) |
| S1F4            | SV          | 0,10,11,20,21,30,40,50 | String(20) |
| S1F11           | SVID        | 20,30,50               | String(20) |
| S1F12           | SVID        | 20,30,50               | String(20) |
|                 | SVNAME      | 20                     | String(20) |
|                 | UNITS       | 20                     | String(20) |
| S1F13           | MDLN        | 20                     | String(20) |
|                 | SOFTREV     | 20                     | String(20) |
| S1F14           | COMMACK     | 10                     | binary(10) |
| S1F15           | Header Only |                        |            |
| S1F16           | OFLACK      | 10                     | binary(10) |
| S1F17           | Header Only |                        |            |
| S1F18           | ONFLACK     | 10                     | binary(10) |

| Stream 2 : 設備控制與診斷 |             |                   |            |
|--------------------|-------------|-------------------|------------|
|                    | Item        | FormatCode        | Type       |
| S2F13              | ECID        | 20,30,50          | String(20) |
| S2F14              | ECV         | 10,11,20,30,40,50 | String(20) |
| S2F15              | ECID        | 20,30,50          | String(20) |
|                    | ECV         | 10,11,20,30,40,50 | String(20) |
| S2F16              | EAC         | 10                | binary(10) |
| S2F17              | Header Only |                   |            |
| S2F18              | TIME        | 20                | String(20) |
| S2F23              | TRID        | 20,30,50          | String(20) |
|                    | DSPER       | 20                | String(20) |
|                    | TOTSMP      | 20,30,50          | String(20) |

|       |             |                      |             |
|-------|-------------|----------------------|-------------|
|       | REPGSZ      | 20,30,50             | String(20)  |
|       | SVID        | 20,30,50             | String(20)  |
| S2F24 | TIAACK      | 10                   | binary(10)  |
| S2F29 | ECID        | 20,30,50             | String(20)  |
| S2F30 | ECID        | 20,30,50             | String(20)  |
|       | ECNAME      | 20                   | String(20)  |
|       | ECMIN       | 10,11,20,21,30,40,50 | String(20)  |
|       | ECMAX       | 10,11,20,21,30,40,50 | String(20)  |
|       | ECDEF       | 10,11,20,21,30,40,50 | String(20)  |
|       | UNITS       | 20                   | String(20)  |
| S2F31 | TIME        | 20                   | String(20)  |
| S2F32 | TIACK       | 10                   | binary(10)  |
| S2F33 | DATAID      | 20,30,50             | String(20)  |
|       | RPTID       | 20,30,50             | String(20)  |
|       | VID         | 20,30,50             | String(20)  |
| S2F34 | DRACK       | 10                   | binary(10)  |
| S2F35 | DATAID      | 20,30,50             | String(20)  |
|       | CEID        | 20,30,50             | String(20)  |
|       | RPTID       | 20,30,50             | String(20)  |
| S2F36 | LRACK       | 10                   | binary(10)  |
| S2F37 | CEED        | 11                   | boolean(11) |
|       | CEID        | 20,30,50             | String(20)  |
| S2F38 | ERACK       | 10                   | binary(10)  |
| S2F39 | DATAID      | 20,30,50             | String(20)  |
|       | DATALLENGTH | 30,50                | int(34)     |
| S2F40 | GRANT       | 10                   | binary(10)  |
| S2F41 | RCMD        | 20,31,51             | String(20)  |
|       | CPNAME      | 20,30,50             | String(20)  |
|       | CPVAL       | 10,11,20,21,30,50    | String(20)  |
| S2F42 | HACK        | 10                   | binary(10)  |
|       | CPNAME      | 20,30,50             | String(20)  |
|       | CPACK       | 10                   | binary(10)  |
| S2F65 | CEED        | 11                   | boolean(11) |
|       | TRID        | 20,30,50             | String(20)  |
| S2F66 | ERACK       | 10                   | binary(10)  |

## Stream 4 : 材料控制

|       | Item        | FormatCode        | Type           |
|-------|-------------|-------------------|----------------|
| S4F19 | DATAID      | 20,30,50          | String(20)     |
|       | TRJOBNAME   | 20                | String(20)     |
|       | TRLINK      | 50                | unsign int(54) |
|       | TRPORT      | 50                | unsignbyte(51) |
|       | TROBJNAME   | 20                | String(20)     |
|       | TROBJTYPE   | 50                | unsignbyte(51) |
|       | TRROLE      | 51                | unsignbyte(51) |
|       | TRRCP       | 20                | String(20)     |
|       | TRPTNR      | 20                | String(20)     |
|       | TRPTPORT    | 50                | unsignbyte(51) |
|       | TRDIR       | 51                | unsignbyte(51) |
|       | TRTYPE      | 51                | unsignbyte(51) |
|       | TRLOCATION  | 50                | unsignbyte(51) |
|       | TRAUTOSTART | 11                | boolean(11)    |
| S4F20 | TRJOBID     | 10                | binary(10)     |
|       | TRATOMICID  | 50                | unsign int(54) |
|       | TRACK       | 11                | boolean(11)    |
|       | ERRCODE     | 50                | unsignbyte(51) |
|       | ERRTEXT     | 20                | String(20)     |
| S4F21 | TRJOBID     | 10                | binary(10)     |
|       | TRCMDNAME   | 20                | String(20)     |
|       | CPNAME      | 20,30,50          | String(20)     |
|       | CPVAL       | 10,11,20,21,30,50 | String(20)     |
| S4F22 | TRACK       | 11                | boolean(11)    |
|       | ERRCODE     | 50                | unsignbyte(51) |
|       | ERRTEXT     | 20                | String(20)     |
| S4F23 | TRJOBID     | 10                | binary(10)     |
|       | TRJOBNAME   | 20                | String(20)     |
|       | TRJOBMS     | 51                | unsignbyte(51) |
|       | TRACK       | 11                | boolean(11)    |
|       | ERRCODE     | 50                | unsignbyte(51) |
|       | ERRTEXT     | 20                | String(20)     |
| S4F24 | Header Only |                   |                |
| S4F27 | TRLINK      | 50                | unsign int(54) |

|       |            |                   |                |
|-------|------------|-------------------|----------------|
|       | TRPORT     | 50                | unsignbyte(51) |
|       | TROBJNAME  | 20                | String(20)     |
|       | TROBJTYPE  | 50                | unsignbyte(51) |
|       | TRROLE     | 51                | unsignbyte(51) |
|       | TRRCP      | 20                | String(20)     |
|       | TRPTNR     | 20                | String(20)     |
|       | TRPTPORT   | 50                | unsignbyte(51) |
|       | TRDIR      | 51                | unsignbyte(51) |
|       | TRTYPE     | 51                | unsignbyte(51) |
|       | TRLOCATION | 50                | unsignbyte(51) |
| S4F29 | TRLINK     | 50                | unsign int(54) |
|       | MCINDEX    | 50                | unsign int(54) |
|       | HOCMDNAME  | 20                | String(20)     |
|       | CPNAME     | 20,30,50          | String(20)     |
|       | CPVAL      | 10,11,20,21,30,50 | String(20)     |
| S4F31 | TRLINK     | 50                | unsign int(54) |
|       | MCINDEX    | 50                | unsign int(54) |
|       | HOACK      | 11                | boolean(11)    |
|       | ERRCODE    | 50                | unsignbyte(51) |
|       | ERRTEXT    | 20                | String(20)     |
| S4F33 | TRLINK     | 50                | unsign int(54) |
|       | HOACK      | 11                | boolean(11)    |
|       | ERRCODE    | 50                | unsignbyte(51) |
|       | ERRTEXT    | 20                | String(20)     |

| Stream 5 : 例外處理 |       |            |            |
|-----------------|-------|------------|------------|
|                 | Item  | FormatCode | Type       |
| S5F1            | ALCD  | 10         | binary(10) |
|                 | ALID  | 30,50      | int (34)   |
|                 | ALTX  | 20         | String(20) |
| S5F2            | ACKC5 | 10         | binary(10) |
| S5F3            | ALED  | 10         | binary(10) |
|                 | ALID  | 30,50      | int (34)   |
| S5F4            | ACKC5 | 10         | binary(10) |

|       |             |       |                  |
|-------|-------------|-------|------------------|
| S5F5  | ALID        | 30,50 | int (34)         |
| S5F6  | ALCD        | 10    | binary(10)       |
|       | ALID        | 30,50 | int (34)         |
|       | ALTX        | 20    | String(20)       |
| S5F7  | Header Only |       |                  |
| S5F8  | ALCD        | 10    | binary(10)       |
|       | ALID        | 30,50 | int (34)         |
|       | ALTX        | 20    | String(20)       |
| S5F9  | TIMESTAMP   | 20    | String(20)       |
|       | EXID        | 20    | String(20)       |
|       | EXTYPE      | 20    | String(20)       |
|       | EXMESSAGE   | 20    | String(20)       |
|       | EXRECVRA    | 20    | String(20)       |
| S5F10 | Header Only |       |                  |
| S5F11 | TIMESTAMP   | 20    | String(20)       |
|       | EXID        | 20    | String(20)       |
|       | EXTYPE      | 20    | String(20)       |
|       | EXMESSAGE   | 20    | String(20)       |
| S5F12 | Header Only |       |                  |
| S5F13 | EXID        | 20    | String(20)       |
|       | EXRECVRA    | 20    | String(20)       |
| S5F14 | EXID        | 20    | String(20)       |
|       | ACKA        | 11    | boolean(11)      |
|       | ERRCODE     | 50    | unsignedbyte(51) |
|       | ERRTEXT     | 20    | String(20)       |
| S5F15 | TIMESTAMP   | 20    | String(20)       |
|       | EXID        | 20    | String(20)       |
|       | ACKA        | 11    | boolean(11)      |
|       | ERRCODE     | 50    | unsignedbyte(51) |
|       | ERRTEXT     | 20    | String(20)       |
| S5F16 | Header Only |       |                  |
| S5F17 | EXID        | 20    | String(20)       |
| S5F18 | EXID        | 20    | String(20)       |
|       | ACKA        | 11    | boolean(11)      |
|       | ERRCODE     | 50    | unsignedbyte(51) |
|       | ERRTEXT     | 20    | String(20)       |

## Stream 6 : 資料收集

|       | Item      | FormatCode             | Type           |
|-------|-----------|------------------------|----------------|
| S6F1  | TRID      | 20,30,50               | String(20)     |
|       | SMPLN     | 30,50                  | int(34)        |
|       | STIME     | 20                     | String(20)     |
|       | SV        | 0,10,11,20,21,30,40,50 | String(20)     |
| S6F2  | ACKC6     | 10                     | binary(10)     |
| S6F5  | DATAID    | 20,30,50               | String(20)     |
|       | DATALNGTH | 30,50                  | int(34)        |
| S6F6  | GRANT6    | 10                     | binary(10)     |
| S6F11 | DATAID    | 20,30,50               | String(20)     |
|       | CEID      | 20,30,50               | String(20)     |
|       | RPTID     | 20,30,50               | String(20)     |
|       | V         | 0,10,11,20,21,30,40,50 | String(20)     |
| S6F12 | ACKC6     | 10                     | binary(10)     |
| S6F15 | CEID      | 20,30,50               | String(20)     |
| S6F16 | DATAID    | 20,30,50               | String(20)     |
|       | CEID      | 20,30,50               | String(20)     |
|       | RPTID     | 20,30,50               | String(20)     |
|       | V         | 0,10,11,20,21,30,40,50 | String(20)     |
| S6F19 | RPTID     | 20,30,50               | String(20)     |
| S6F20 | V         | 0,10,11,20,21,30,40,50 | String(20)     |
| S6F23 | RSDC      | 51                     | unsignbyte(51) |
| S6F24 | RSDA      | 10                     | binary(10)     |
| S6F27 | DATAID    | 20,30,50               | String(20)     |
|       | TRID      | 20,30,50               | String(20)     |
|       | RPTID     | 20,30,50               | String(20)     |
|       | V         | 0,10,11,20,21,30,40,50 | String(20)     |
| S6F28 | TRID      | 20,30,50               | String(20)     |
| S6F29 | TRID      | 20,30,50               | String(20)     |
| S6F30 | TRID      | 20,30,50               | String(20)     |
|       | RPTID     | 20,30,50               | String(20)     |
|       | V         | 0,10,11,20,21,30,40,50 | String(20)     |
|       | ERRCODE   | 50                     | unsignbyte(51) |
| S6F65 | DATAID    | 20,30,50               | String(20)     |
| S6F66 | DATAID    | 20,30,50               | String(20)     |

|       |           |                        |            |
|-------|-----------|------------------------|------------|
|       | EVNTRCID  | 20                     | String(20) |
|       | CEID      | 20,30,50               | String(20) |
| S6F67 | DATAID    | 20,30,50               | String(20) |
| S6F68 | DATAID    | 20,30,50               | String(20) |
|       | DATASRCID | 20                     | String(20) |
|       | V         | 0,10,11,20,21,30,40,50 | String(20) |

| Stream 7 : 材料控制 |             |                |            |
|-----------------|-------------|----------------|------------|
|                 | Item        | FormatCode     | Type       |
| S7F1            | PPID        | 10,20          | String(20) |
|                 | LENGTH      | 30,50          | int(34)    |
| S7F2            | PPGNT       | 10             | binary(10) |
| S7F3            | PPID        | 10,20          | String(20) |
|                 | PPBODY      | 10,20,30,50    | String(20) |
| S7F4            | ACKC7       | 10             | binary(10) |
| S7F5            | PPID        | 10,20          | String(20) |
| S7F6            | PPID        | 10,20          | String(20) |
|                 | PPBODY      | 10,20,30,50    | String(20) |
| S7F17           | PPID        | 10,20          | String(20) |
| S7F18           | ACKC7       | 10             | binary(10) |
| S7F19           | Header Only |                |            |
| S7F20           | PPID        | 10,20          | String(20) |
| S7F23           | PPID        | 10,20          | String(20) |
|                 | MDLN        | 20             | String(20) |
|                 | SOFTREV     | 20             | String(20) |
|                 | CCODE       | 32,52          | short(32)  |
|                 | PPARM       | 11,20,30,40,50 | String(20) |
| S7F24           | ACKC7       | 10             | binary(10) |
| S7F25           | PPID        | 10,20          | String(20) |
| S7F26           | PPID        | 10,20          | String(20) |
|                 | MDLN        | 20             | String(20) |
|                 | SOFTREV     | 20             | String(20) |
|                 | CCODE       | 32,52          | short(32)  |
|                 | PPARM       | 11,20,30,40,50 | String(20) |
| S7F27           | PPID        | 10,20          | String(20) |

|       |             |       |                  |
|-------|-------------|-------|------------------|
|       | ACKC7A      | 31,51 | unsignedbyte(51) |
|       | SEQNUM      | 30,50 | unsignedbyte(51) |
|       | ERRW7       | 20    | String(20)       |
| S7F28 | Header Only |       |                  |
| S7F29 | LENGTH      | 30,50 | signedint(34)    |
| S7F30 | PPGNT       | 10    | binary(10)       |

| Stream 9 : 系統錯誤 |       |             |            |
|-----------------|-------|-------------|------------|
|                 | Item  | FormatCode  | Type       |
| S9F1            | MHEAD | 10          | binary(10) |
| S9F3            | MHEAD | 10          | binary(10) |
| S9F5            | MHEAD | 10          | binary(10) |
| S9F7            | MHEAD | 10          | binary(10) |
| S9F9            | SHEAD | 10          | binary(10) |
| S9F11           | MHEAD | 10          | binary(10) |
| S9F13           | MEXP  | 20          | String(20) |
|                 | EDID  | 10,20,30,50 | String(20) |



| Stream 10 : 終端機服務 |        |                |            |
|-------------------|--------|----------------|------------|
|                   | Item   | FormatCode     | Type       |
| S10F1             | TID    | 10             | binary(10) |
|                   | TEXT   | 10,20,22,30,50 | String(20) |
| S10F2             | ACKC10 | 10             | binary(10) |
| S10F3             | TID    | 10             | binary(10) |
|                   | TEXT   | 10,20,22,30,50 | String(20) |
| S10F4             | ACKC10 | 10             | binary(10) |

## Stream 16 : 製程管理

|        | Item           | FormatCode             | Type           |
|--------|----------------|------------------------|----------------|
| S16F3  | DATAID         | 20,30,50               | String(20)     |
|        | MF             | 10,20                  | binary(10)     |
|        | MID            | 10,20                  | String(20)     |
|        | PRRECIPEMETHOD | 51                     | unsignbyte(51) |
|        | RCPSPEC        | 20                     | String(20)     |
|        | RCPPARNM       | 20                     | String(20)     |
|        | RCPPARVAL      | 10,11,20,30,40,50      | String(20)     |
|        | PRPROCESSSTART | 11                     | boolean(11)    |
| S16F4  | PRJOBID        | 20                     | String(20)     |
|        | ACKA           | 11                     | boolean(11)    |
|        | ERRCODE        | 50                     | unsignbyte(51) |
|        | ERRTEXT        | 20                     | String(20)     |
| S16F5  | DATAID         | 20,30,50               | String(20)     |
|        | PRJOBID        | 20                     | String(20)     |
|        | PRCMDNAME      | 20                     | String(20)     |
|        | CPNAME         | 20,30,50               | String(20)     |
|        | CPVAL          | 10,11,20,21,30,50      | String(20)     |
| S16F6  | PRJOBID        | 20                     | String(20)     |
|        | ACKA           | 11                     | boolean(11)    |
|        | ERRCODE        | 50                     | unsignbyte(51) |
|        | ERRTEXT        | 20                     | String(20)     |
| S16F7  | TIMESTAMP      | 20                     | String(20)     |
|        | PRJOBID        | 20                     | String(20)     |
|        | PRJOBMILESTONE | 50                     | unsignbyte(51) |
|        | ACKA           | 11                     | boolean(11)    |
|        | ERRCODE        | 50                     | unsignbyte(51) |
|        | ERRTEXT        | 20                     | String(20)     |
| S16F8  | Header Only    |                        |                |
| S16F9  | PREVENTID      | 50                     | unsignbyte(51) |
|        | TIMESTAMP      | 20                     | String(20)     |
|        | PRJOBID        | 20                     | String(20)     |
|        | VID            | 20,30,50               | String(20)     |
|        | V              | 0,10,11,20,21,30,40,50 | String(20)     |
| S16F10 | Header Only    |                        |                |
| S16F25 | PRJOBID        | 20                     | String(20)     |
|        | PRPROCESSSTART | 11                     | boolean(11)    |

## Stream 17 : 設備控制與診斷

|       | Item        | FormatCode | Type           |
|-------|-------------|------------|----------------|
| S17F0 | Header Only |            |                |
| S17F1 | DATAID      | 20,30,50   | String(20)     |
|       | RPTID       | 20,30,50   | String(20)     |
|       | DATASRC     | 20         | String(20)     |
|       | VID         | 20,30,50   | String(20)     |
| S17F2 | RPTID       | 20,30,50   | String(20)     |
|       | ERRCODE     | 50         | unsignbyte(51) |
| S17F3 | RPTID       | 20,30,50   | String(20)     |
| S17F4 | ACKA        | 11         | boolean(11)    |
|       | RPTID       | 20,30,50   | String(20)     |
|       | ERRCODE     | 50         | unsignbyte(51) |
|       | ERRTEXT     | 20         | String(20)     |
| S17F5 | DATAID      | 20,30,50   | String(20)     |
|       | TRID        | 20,30,50   | String(20)     |
|       | CEED        | 11         | boolean(11)    |
|       | RPTID       | 20,30,50   | String(20)     |
|       | TRSPER      | 40         | float (44)     |
|       | TOTSMP      | 20,30,50   | int (34)       |
|       | REPGSZ      | 20,30,50   | int (34)       |
|       | EVNTSRC     | 20         | String(20)     |
|       | CEID        | 20,30,50   | String(20)     |
|       | TRAUTOD     | 11         | boolean(11)    |
|       | RPTOC       | 11         | boolean(11)    |
| S17F6 | TRID        | 20,30,50   | String(20)     |
|       | ERRCODE     | 50         | unsignbyte(51) |
| S17F7 | TRID        | 20,30,50   | String(20)     |
| S17F8 | ACKA        | 11         | boolean(11)    |
|       | TRID        | 20,30,50   | String(20)     |
|       | ERRCODE     | 50         | unsignbyte(51) |
|       | ERRTEXT     | 20         | String(20)     |
| S17F9 | DATAID      | 20,30,50   | String(20)     |
|       | EVNTSRC     | 20         | String(20)     |
|       | CEID        | 20,30,50   | String(20)     |
|       | RPTID       | 20,30,50   | String(20)     |

|        |         |          |                  |
|--------|---------|----------|------------------|
| S17F10 | EVNTSRC | 20       | String(20)       |
|        | CEID    | 20,30,50 | String(20)       |
|        | ERRCODE | 50       | unsignedbyte(51) |
| S17F11 | EVNTSRC | 20       | String(20)       |
|        | CEID    | 20,30,50 | String(20)       |
|        | RPTID   | 20,30,50 | String(20)       |
| S17F12 | EVNTSRC | 20       | String(20)       |
|        | CEID    | 20,30,50 | String(20)       |
|        | RPTID   | 20,30,50 | String(20)       |
|        | ERRCODE | 50       | unsignedbyte(51) |
| S17F13 | TRID    | 20,30,50 | String(20)       |
| S17F14 | ACKA    | 11       | boolean(11)      |
|        | TRID    | 20,30,50 | String(20)       |
|        | ERRCODE | 50       | unsignedbyte(51) |
|        | ERRTEXT | 20       | String(20)       |

