

# TFT LCD 자동생산시스템에서 Data 통신 및 응용

조민호\*

## Data Communications and their Applications in the Automated TFT LCD Manufacturing System

Minho Jo

### 〈Abstract〉

The SECS I and II Protocol has been widely used for the TFT LCD and semiconductor industry. This paper shows how the SECS protocol is implemented for data communications between the Host (CAM) and the TFT LCD equipments. In addition, this study introduces a way to apply the SECS protocol to the manufacturing systems control. It provides better throughput in terms of production and faster control of the automated TFT LCD manufacturing system by way of shortcut and distribution of control and data communications. The SECS protocol is successfully used for the control of the real TFT LCD manufacturing system.

### 1. 연구의 배경

TFT LCD(Thin Film Transistor Liquid Crystal Display, 박막 트랜지스터 액정화면)는 반도체와 같은 일련의 연속적인 박막공정(Deposition Process)과 식각공정(Etching Process)을 통해서 전기적 특성을 갖는 미세한 트랜지스터 회로들을 사각의 Glass 위에 Patterning 하여 가는 일종의 Fabrication(또는 Fab) 생산공정이다. 반도체 Chip Fab 공정은 원형의 웨이퍼(Wafer) 표면 위에서 트랜지스터 회로Pattern이 이루어진다는 점에서 TFT LCD Fab 공정과 다르다. 그리고 또 하나 다른 점은 TFT LCD는 상판 Glass와 하판 Glass 두 장으로 이루어져 있으며 조립(Assembly)된 두 장의 Glass 사이에 액정이 주입되어지는 공정이 포함된다는 것이다.

그러나 TFT LCD와 반도체 칩(Chip) 모두 고청정도 및 무정전기 생산환경 요구(즉 무인 생산환경 요구)와 라인내의 고기압 환경(인간에게 부적합), 공정의 복잡성, Material Handling(특히 Glass)의 어려움, 대량의 Data Handling 등이 On-line 통신을 통한 자동생산 시스템을 필요로 하게 되었다.

초기 반도체 생산은 생산자(Manufacturer)가 정의한 고유의 내부 통신으로만 가지고서 이루어졌다. 그리고 대부분의 반도체 공정장비는 장비 내장형 디지털 컴퓨터(Imbedded Digital Computers)나 장비전용 컴퓨터 시스템(Stand-alone Computer System)에 의해서 제어되었다[1]. 그러나 반도체 산업에도 CAM(Computer Aided Manufacturing) 시스템이 도입되는 시점에서 한 단계 높은 자동화를 위해서는 Host 컴퓨터(CAM)와 장비 컴퓨터 사이의 통신 또는 장비와 장비

\* 삼성전자(반도체부문)

사이간 통신을 위한 통일된 통신언어가 필요하게 되었던 것이다.

이러한 동기에 의해서 1980년 통신의 Hardware 규약(Protocol)인 SECS I(Semi Equipment Communications Standard I)이 발표되었으며 차례로 1982년 통신 메시지 규약(Protocol)인 SECS II(Semi Equipment Communications Standard II)가 발표되었다[2].

특별히 이 논문에서는 대량의 Data를 통신하고 복잡한 생산 Logic을 다루어야 하는 Host를 위해서 SECS Protocol을 이용하여 장비 서버(Eqp Server)를 통해서 Host의 역할인 생산제어 및 데이터 통신의 일부를 분담하는 통신 및 제어의 분산 방법을 제시 하고 있다. 이 연구에서는 실제로 현업에서 적용 하고 있는 TFT LCD의 Coater/Developer 장비에 대한 SECS Message Sequence Design을 통해서 SECS Protocol 응용 방법을 제시하며 장비와 장비 서버 간 통신 시뮬레이션 결과를 보여준다. 먼저 SECS I & II Protocol 을 간단히 소개 한다.

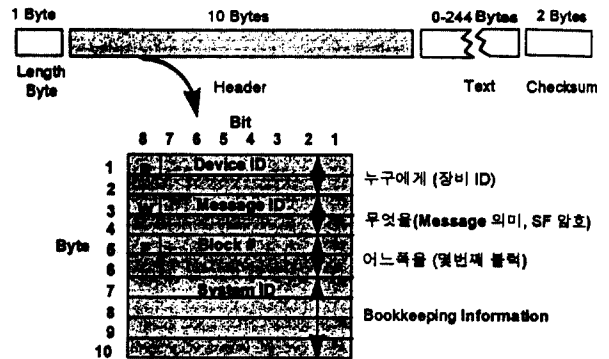
## 2. SECS Protocol

### 2.1 SECS I Protocol

SECS Protocol은 7 Layer로 규정되어 있다. 4 개 (Physical, Block, Message, Transaction)의 하위 Layer 는 SECS I에 관한 것을 규정하고 있으며 상위 3 개 (Data Item Format, Message Format, Conversation)의 Layer는 SCES II에 대해서 규정하고 있다.

SECS-I 표준은 국제표준인 EIA RS-232C[3][4]에서 정의된 일부분(Pin# 1,2,3,4,5,6,7,8,18,20,25 번만을 사용)을 이용하여 Point-to-Point 통신을 정의하고 있다. 통신은 양방향으로 이루어지고 있으며 비동기(Asynchronous) 통신이다. 따라서 8-bit Character 중심 통신이며 각 Character에는 Start Bit와 Stop Bit가 존재한다[5]. 통신방향은 한 번에 한 방향만 허용하며 방향 결정은 <그림1>에서 보여주고 있는 10-byte로 구성된 Header와 Handshake에 의해서 결정된다.

메세지는 반드시 Request(Primary 메세지, Data 요청 또는 Data 송신)가 있으면 Reply(Secondary 메세지, 응



<그림 1> SECS Logical Layers (Block과 Message Layer)

답 또는 Data 송신)가 있는 형식의 한 짝으로 구성됨을 원칙으로 하나 Reply가 필요치 않을 때는 Reply 요구를 생략할 수도 있다. Data는 254-byte를 한 Block 단위로 하며 한 번에 최대 32,767 Block 단위 까지 전송할 수 있다.

SECS 통신 Protocol 중 중요한 Logical 부분을 정의하고 있는 것은 Block Layer와 Message Layer이다. Block Layer는 사용하는 Block(또는Packet)을 정의하며 Master(Sender)와 Slave(Receiver)가 취해야할 사양도 규정하고 있다. 그리고 Error Detection(Collision, Checksum)도 정의하고 있다. <그림1>에서는 하나의 SECS 메세지 Block이 Length Byte(1 Byte), Header(10 Byte), Text(0-244 Byte), Checksum(2 Byte)으로 구성되어 있음을 보여주고 있다. 메세지 Layer는 Multi-block인 경우 중복 Block 방지(Block ID를 통해서), Block Interleaving(Block 중간끼우기)등의 방법을 규정하고 있다. <그림1>을 보면 하나의 메세지 Block은Length Byte 자신을 제외한 나머지 총 Byte의 수를 나타내는 Length Byte부터 시작한다.

통신의 기본 정보를 가지고 있는 Header, Header의 메세지 ID(Stream Function, SECS II에서 정의되어진 암호)Byte에서 정의된 것을 보다 구체적인 정보를 담고 있는 Text(때때로 Block이라 부름, Multi-block가능), 그리고 통신 Error를 확인하기 위한 Checksum 등이 차례로 이어진다. Header 부분에서 Device ID(2 Byte)는 메세지가 가야할 장비 또는 Host 이름을 가지고 있으며 8 번째 Bit(R Bit)가 '0' 이면 Host에서 장비로,

'1' 이면 장비에서 Host로 가는 메시지임을 알린다. 메시지 ID(2 Byte)는 Stream Function(SF 암호)을 담고 있으며 마지막 8번째 Bit(W Bit)가 '1'이면 Reply를 요구하며 '0'이면 요구하지 않는다. W Bit를 통해서 메시지가 Primary(Sender) 인지 Secondary(Receiver) 인지를 구분 한다. Block #(2 Byte)는 다중 Block인 경우를 대비하여 Block ID를 담고 있으며 E Bit가 '1'이면 마지막 Block을 의미하며 '0'이면 마지막 Block이 아니다. System ID는 Error 발생시 메시지 Block을 추적하기 위해서 예비해 둔 것이다. 그러나 실제로 거의 사용하지 않고 있다.

### 2.2 SECS II Protocol

SECS I이 메시지 송수신을 하기 위한 규약을 정의한 반면 SECS II는 메시지 내용에 대해서 정의하고 있다. SECS I과 II는 서로 관련이 있지만 통신에 있어서 서로 다른 영역을 다루고 있다. SECS II는 SECS I Protocol 외에 Ethernet, MAP 등의 다른 송수신 Protocol에서도 사용 되어질 수 있다. SECS II는 구체적으로 Stream Function에 값을 부여하고 Data Item Format 및 표준 메시지 구조를 정의한다.

〈표 1〉 TFT LCD Data 통신에서 사용되는 Stream Function 예

Message	설 명	통신방향
S1F1	Online Check Request	Host→장비
S1F2	Online Check Request Reply	Host→장비
S2F41	Host Commend Send	Host→장비
S2F41	Host Commend Acknowledge	Host→장비
S6F11	Event Report Send	Host→장비
S6F12	Event Report Acknowledge	Host→장비

Stream Function : 〈표1〉은 SECS II에서 정의하고 있는 대표적인 Stream Function 중 몇 개를 소개하고 있다. Sending Message의 Function값은 언제나 홀수이며 Reply Message의 Function 값은 언제나 짝수이다. S1F1은 Host 또는 장비 어느 쪽에서도 보낼 수 있는 메시지이다. S2F41은 Host 쪽에서 Command를 보낼

때 사용 하며 보내는 Command가 1 개 이상 종류이면 RCMD(Remote Command)에 번호를 부여하여 구분한다. S6F11은 장비 쪽에서 일어나는 Event(Lot Loading, Process Start, 장비고장 등)를 Host에 보고할 때 사용 하며 Event가 하나 이상일 때는 CEID(Collected Event ID)에 번호를 붙여 구분한다.

〈표 2〉 S6F11 Message 구조

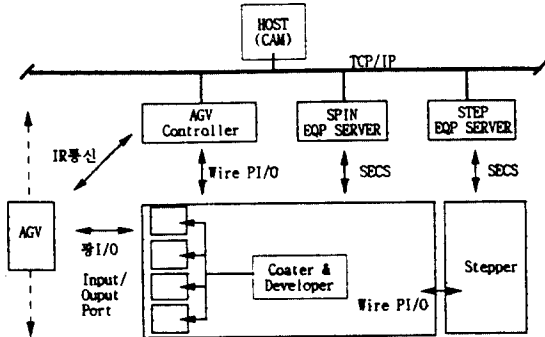
S6F11 Event Report Send	
Structure :	L, 3
1.	〈DataID〉
2.	〈CEID〉
3.	L, a
1.	〈RPTID#1〉
2.	L, b
1.	〈V1〉
:	
b.	〈Vb〉
a.	L, 2
1.	〈RPTID#a〉
2.	L, c
1.	〈V1〉
:	
c.	〈Vc〉

Message 구조 : 〈표2〉는 SECS II에서 정의한 S6F11에 대한 메시지 구조를 보여주고 있다.

'L,a'에서 L은 Line을 의미 하고 a는 하부 구조에 있는 Line의 개수(a 개)를 의미 하며 만약 그 하부구조에 구체적인 Data Parameter가 있으면 Parameter수를 말한다. CEID는 S6F11구조를 하나 이상 사용할 때 구분을 위한 번호를 담고 있다. 〈DATAID〉와 〈RPTID〉는 각각 Data와 Report를 구분하기 위한 것이지만 실무적으로 대부분 경우에 있어서 이를 사용하지 않는다. 〈V〉는 실제 전달할 Data Value를 말한다. 구체적인 Parameter수와 종류 및 Format은 User에 의해서 정해져야 한다. 나머지 Stream Function에 대한 Message 구조도 이와 같지만 Parameter의 종류는 Stream Function의 특성에 따라 달라진다.

### 3. 통신Network 구성

〈그림2〉에서는 Photolithography (Coater/Developer, Stepper)를 중심으로 한 실제 TFT LCD 생산환경에서 적용 되고 있는 통신 Network을 보여주고 있다. SECS



〈그림 2〉 TFT LCD 생산시스템에서 통신 Network 구성도

통신은 장비와 CAM 소프트웨어가 내장되어 있는 Host와 생산장비 또는 자동반송(AS/RS, Lifter)과 같은 기타 장비 사이를 이어주는 역할을 하며 그 사이에는 장비 서버가 장비로부터 올라오는 SECS 메시지를 Host 통신 Protocol인 TCP/IP 메시지 Format으로 Parsing도 하며 CAM(Host) 대신 공정시작 및 종료와 같은 일부 생산제어도 담당한다.

Coater와 Developer는 하나의 일체 장비이며 Stepper는 그들과 분리되어 있지만 Wire PI/O(Parallel Input/Output) 통신으로 Data를 서로 주고 받는다. Wire PI/O를 통해 주로 Recipe ID(생산조건 Parameter를 정의한 것)를 전달한다. 그리고 Coater/Developer를 위한 장비 서버(Spin Eqp Server)가 있으며 Stepper를 위한 장비 서버(Step Eqp Server)도 별도 존재한다.

Coater/Developer에는 Glass의 Container(=Cassette)를 Loading과 Unloading하는 Loader(또는Input Port)/Unloader(또는 Output Port)가 장착되어 있다. AGV (Automated Guided Vehicle)와 장비의 Port 사이에는 광 PI/O 통신을 통해서 Handshake Checking을 한다.

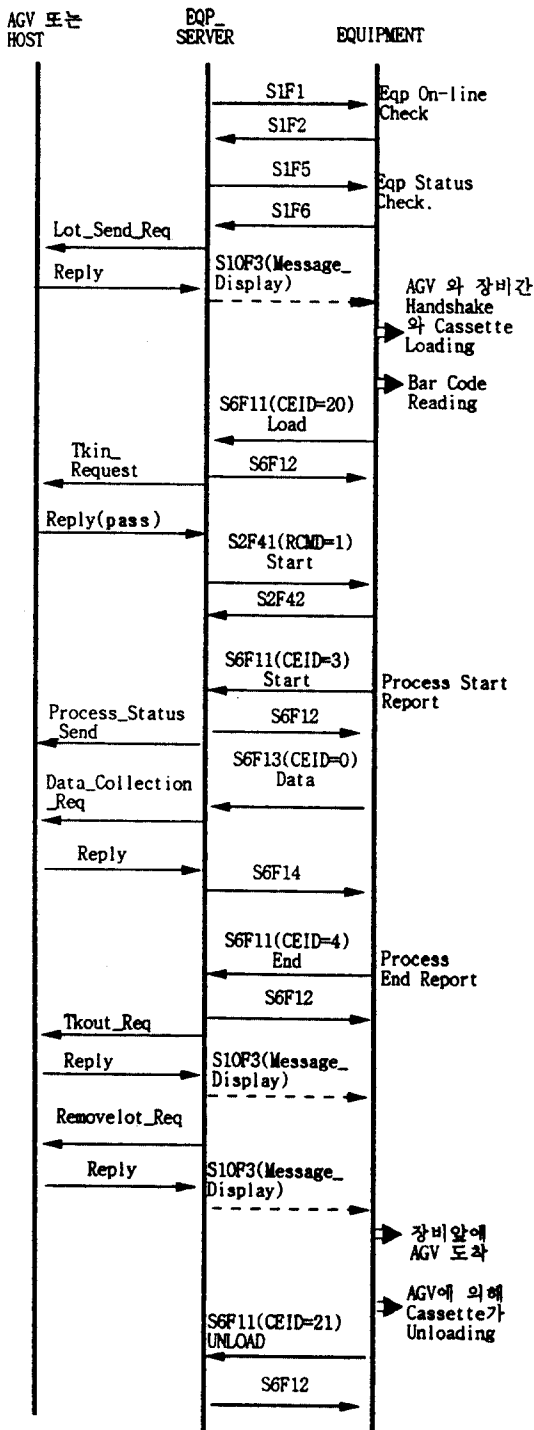
### 4. TFT LCD 공정장비제어를 위한 SECS 통신이용

여기에서는 일반적으로 Data 전송이 주 목적인 SECS 통신의 SECS Message를 이용한 생산제어를 위해서 응용 되고 있다. 이를 위해서는 장비의 생산 환경 및 조건에 따라서 생산제어를 담당할 장비 서버가 필요 하다. 생산장비와 장비 서버간 SECS 통신에 있어서 가장 중요한 것은 SECS Message Sequence이다. 실제로 한 생산시스템의 Throughput은 SECS Message Sequence와 밀접한 관계가 있다. 물론 생산시스템의 특성과 자동화 정도에 따라서 SECS Message Sequence Design이 되어야 하겠지만 Design 시 반드시 생산시스템의 Throughput을 고려해야만 한다.

참고로 TFT LCD 생산 시스템은 크게 박막공정 (박막을형성), Photolithography 공정 (회로 Patterning), Etching 공정 (Pattern에 따라 식각), 액정공정 (상판과 하판을 접합하고 액정을 주입), PCB Assembly (구동에 필요한 부품 조립) 공정으로 나누어져 있으며 Color Filter Glass (상판 Glass)와 TFT Glass (하판Glass)가 박막 공정, Photolithography 공정, Etching 공정을 7 회 순환 (7 층 박막인 경우)한 후 액정 공정에서 상판과 하판이 접합되어 PCB Assembly 공정에서 TFT LCD 구동에 필요한 PCB, Back Light, Frame 등을 부착하는 공정을 끝으로 제품 생산이 끝난다. 각 공정은 공정에 필요한 장비들이 Group으로 배치되어 있으며 Glass는 Cassette에 담겨져 AGV나 MGV (Manual-guided Vehicle)에 의해서 반송 되어진다. 그리고 각 공정간 효과적인 Line Balance를 위해서 공정간 또는 장비간 AS/RS (자동창고, Automated Storage/Retrieval System)이 설치되어 있다.

#### 4.1 SECS Message Sequence Design

〈그림3〉은 Host 또는 AGV와 Coater/Developer간 SECS 통신및 메시지를 이용한 생산제어와 Data Collection을 위한 SECS Message Sequence Design 예를 보여주고 있다. 본 예에서는 AGV를 사용시 AGV에 대한 제어 Command 사용도 관련하여 보여주고 있



〈그림 3〉 생산제어를 위한 SECS 통신 Message Sequence

다. Host(또는 AGV Controller)와 Eqp Server 사이에는 TCP/IP 통신 상에서 이루어지는 제어 및 응답 메시지가이며 장비 서버와 Coater/Developer와는 SECS 통신상에서 주고 받는 SECS 메시지이다. 다음은 Host와 장비 서버간 통신 Command 및 장비 서버와 TFT LCD 생산장비에서 주로 사용되어지는 User 정의의 SECS 통신 메시지들에 대한 설명이다.

**TrackIn(TKIN):** 하나의 Lot(Cassette)의 Process을 위해 Resource(설비) 확보를 위한Command. 이 Command가 내려질 시 설비의 확보된 특정 Port에는 다른 Lot이 TrackIn 할 수 없다.

**TrackOut(TKOUT):** 하나의 Lot이 Process를 마치고 Resource(설비)를 Release(확보해제) 하는 Command.

**Data\_Collection:** Data를 장비로부터 취합하여 Host로 보내기 위한 메시지.

**Lot\_Send:** Host가 특정한 Lot을 특정한 장소로 이동시키는 Command.

**Remove\_Lot:** 특정한 Lot을 지정된 장비의 Port로 부터 제거하는 Host의 Command.

**Eqp\_On\_line Check(S1F1):** 장비와 Host간 Network 연결상태를 점검하는 SECS 메시지.

**Eqp\_Status(S1F5):** 장비상태(Not Ready, Ready, Busy, Alarm등)를 점검하는 SECS 메시지.

**Load(S6F11, CEID=20):** Cassette(Lot)가 장비 Port에 Loading이 되었음을 알리는 SECS 메시지.

**Unload(S6F11, CEID=21):** Cassette(Lot)가 장비의 Port에 안전하게 Unloading 되었음을 알리는 SECS 메시지.

**Data(S6F13, CEID=0):** 장비로부터 발생한 Data를 Eqp Server로 올려주는 SECS 메시지.

**Rstart(S2F41, RCMD=1):** Eqp Server로부터 장비에게 Process Start를 명하는 SECS 메시지.

**Start\_Report(S6F11):** 장비가 Process를 안전하게 시작했다는 SECS 메시지.

**End(S6F11, CEID=4):** 장비가 하나의 Lot에 대한 Process 종료를 알리는 SECS Message.

**Message\_Display(S10F3):** Host나 Eqp Server로부터

Operator를 위해서 온 메시지를 장비화면 상에 보여주는 SECS 메시지.

#### 4.2. SECS Message 구조 및 Format

앞에서 언급 했듯이 SECS 메시지 구조 및 Format 은 SECS II Protocol에서 정의된 것에 따른다. 그러나 메시지의 Item 종류, 내용, Data Type, CEID(Collected Event ID), RCMD 등은 장비의 종류와 생산제어의 전략과 Logic 형태에 따라서 결정되어야 한다. <표3>에서는 Coater/Developer 장비를 위하여 이미 <그림3>에서 보여준 Sequence에 따른 S6F11(CEID=3, Process

Start) SECS 메시지 구조와 Format을 보여주고 있다.

이 연구에서 S6F11는 Load 외에도 Unload, TrackIn, TrackOut, Process End, Process Pause, Process Resume, Process Abort와 같은 유사한 Event를 위해서 사용되었다. 각 Event 별 구분은 CEID(Collected Event ID)에 의해서 이루어진다.

SECS Message Event 는 CEID에 따라서 메시지 Item 의 추가 및 삭제 뿐만 아니라 내용 및 Data Type을 달리 할 수도 있다.

<표3>에서 \$IPMD(Input Port Mode)는 장비의 Port 에서 수행될 Lot(Cassette) Load의 방법이 AGV에 의해서인지 수동(Manual)에 의해서 수행될 것인지를

<표 3> Load(S6F11, CEID=3) SECS Message 구조 및 Format

S6F11(CEID=3)	
L 3	
1. <U2 \$DATAID>	* Dataid = 0 (fixed)
2. <U2 \$CEID>	* CEID = Event ID
3. L, 2	
1. L, 2	
1. <U2 \$RPTID>	* RPTID = 0 (fixed)
2. L, 5	
1. <A[6] \$EQPID>	* 장비 ID
2. <B1 \$IPMD>	* Input Port Mode (AGV/수동 mode)
3. <B1 \$OPMD>	* Output Port Mode (AGV/수동 mode)
4. <B1 \$MCMD>	* 장비 Operation Mode
5. <B1 \$EQST>	* 장비 Status
2. L, 2	
1. <U2 \$RPTID>	* RPTID = 1 (fixed)
2. L, 13	
1. <A[16] \$MID>	* Material(=Lot) ID
2. <A[2] \$IPID>	* Input Port ID
3. <A[2] \$OPID>	* Output Port ID
4. <A[16] \$ICID>	* Input Cassette ID
5. <A[16] \$OCID>	* Output Cassette ID
6. <B1 \$STAUS>	* Tracking Status
7. <A[6] \$FCL>	* First Component's Location
8. <A[6] \$LCL>	* Last Component's Location
9. <A[16] \$PPID>	* Process Plan ID
10. <A[16] \$RECPID>	* Recipe (Process Step)
11. <A[14] \$STIME>	* Startime of Lot
12. <A[16] \$OPERID>	* Operator's ID
13. <U1 \$CCNT>	* Component's Count

Data Type : Un(Unsigned Integer n Byte), ASCII[n](ASCII n Byte), Bn(Binary n Byte).

Binary 1 Byte에 의해서 구분시켜주는 Item이다. \$OPMD(Output Port Mode) Item은 Load시 Blank(공란)가 된다. \$FCL(First Component's Location) 메세지Item은 Glass를 담고 있는 Cassette에서 첫번째 Glass를 가지고 있는 Cassette Slot의 번호를 말해 주고 있다. \$RECPID (Recipe ID 또는 Process Step ID)은 특정한 Lot에 대해서 특정한 장비(여기서는Coater/Developer)가 진행해야 할 공정의 조건(Recipe)을 담고 있는 ID를 가지고 있다. 이것은 Remote Start(S2F41, RCMD=1)나 Process Start Report 시 사용되어지는 메세지 Item이다. \$Dataid(Data ID), \$RPTID(Report ID)들은 중요한 정보가 아니기 때문에 대부분 경우 ID 값이 고정되어 있다.

#### 4.3 Message Sequence 및 Logic 설명

〈그림3〉은 장비 상태 체크부터, 특히 Coater/Developer 장비에서 하나의 Lot에 대해 TrackIn이 Fail 없이 성공적일 경우, Lot이 Port로부터 Unloading 될 때 까지 SECS 통신과 TCP/IP 통신을 이용하여 Host와 장비 서버 및 장비서버와 장비 사이의 생산제어 응용 Message Sequence Design을 보여주고 있다. 〈그림3〉에 대한 메세지 및 Logic 상세 설명은 다음과 같다.

- ① 먼저 Host와 Coater/Developer간 통신을 시작하기 전 Eqp Server가 S1F1(Are You There ?)을 이용하여 On-line 상태 유무를 확인한다. 만약 장비로부터 S1F2 Reply 메세지를 받는다면 On-line 상태인 것으로 인식한다.
- ② S1F6 Reply Message를 통해 장비가 'Ready' 상태이면 Eqp Server는 즉각 Host에게 새로운 Lot을 요청 (Lot\_Send\_Req)한다. 이때 Host는 AGV Controller에게 하나의 Lot을 Coater/Developer 장비로 수송하도록 요청한다.
- ③ Eqp Server는 Host로부터 Lot\_Send\_Req에 대한 Reply(수송될 Lot ID 정보포함)를 받는 즉시 어떤 Lot이 선택되어서 장비로 오고 있는지를 S10F3 SECS 메세지를 통해서 장비화면에 보여준다.
- ④ Lot이 AGV로부터 장비 Port에 성공적으로 Loading되는 순간 Eqp Server는 즉각 Host에게 TrackIn

선언을 요청한다. Host(CAM)는 장비에 Loading된 Lot에 대한 정보를 검증한 후 이상이 없을 경우 TrackIn 허용 Reply Message를 보낸다.

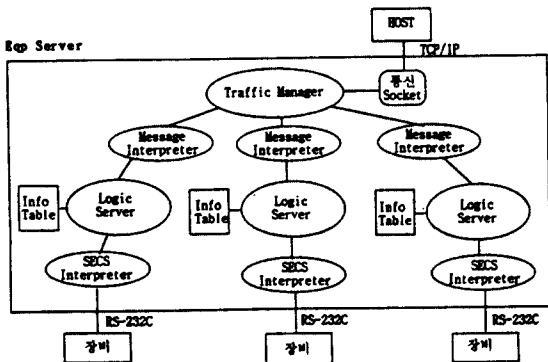
- ⑤ Host로부터 TrackIn 허용 Reply Message를 받은 Eqp Server는 장비에게 Remote Process Start Message(S2F41, CEID=0)를 보낸다.
  - ⑥ 만약장비가 순조롭게 Process을 시작할 경우 장비는 Eqp Server에게 Start Report(S6F11, CEID=3)를 보낸다. 그리고 다시 Eqp Server는 Host에게 Process\_Status\_Send Message를 통해 장비의 Process 상황을 보낸다.
  - ⑦ Process 중에 발생한 Data는 S6F13(CEID=0)을 통해서 Eqp Server로 전달 된다. Eqp Server는 다시 Data\_Collection\_Req를 통해 Data를 Host에 전달하며 Host는 Reply Message를 다시 Eqp Server에 보냄으로써 무사히 받았음을 알리며 곧바로 Eqp Server 역시 장비에 Reply(S6F12)를 보낸다.
  - ⑧ S6F12(Data Reply)가 성공인 경우 장비는 즉시 Process를 마침과 동시에 Eqp Server에 Process End 통보(S6F11, CEID=4)를 하며 Host는 특정한 Lot이 Coater/Developer의 지정된 Port에서 TrackOut 되었음을 선언 한다.
  - ⑨ 그리고Eqp Server는 Host에게 AGV Controller로 하여금 Lot을 Port에서 Unloading 하도록 요청 (Removelot\_Req) 한다.
  - ⑩ Unloading 요청을 받은 AGV는 Lot을 Port로부터 Unloading을 하며 장비는 Eqp Server에게 Unloading 되었음을 알린다(S6F11, CEID=21). 이로써 한 Lot의 공정이 완전히 끝나면서 다른 Lot이 그 Port에 Loading 할 수가 있게 된다.
- 〈그림3〉에서 Design된 생산제어 및 메세지 Sequence는 생산제어 전략과 환경에 따라서 변경 될 수가 있으며 생산 Throughput이 달라질 수가 있다. 예를 들어서 Resource(설비)를 확보하는 메세지인 TrackIn이 현재는 Cassette가 Coater/Developer 설비의 Port에 Loading이 된 직후에 이루어지지만 Loading 전(때에 따라선 Cassette가 이 전 설비로부터 이동도 하기 전에 이루어지는 Sequence도 물류시스템 Throughput 측면에서 고려해볼 만 하다. 즉 후자의 경우 비교적 먼 거

리에 위치한 Lot에 대해서 다음 Step의 설비로 출발하기 전에 미리 TrackIn함으로써 다른 Lot이 Loading하는 것을 방지할 수가 있으며 설비 앞에서 Lot의 대기로 인해 손실되는 생산시간을 줄일 수 있으며 AGV의 Jamming도 방지할 수 있으나 반면에 생산 및 물류 제어가 복잡해 질 수가 있다. 이렇듯 사용자가 정의하는 SECS 통신 메시지와 Sequence는 생산제어 전략 및 Logic과 직접적인 관계가 있으며 TFT LCD 생산 시스템 Throughput에 매우 중요하다.

### 5. Eqp Server

#### 5.1. 시스템 구성 및 기능

장비 서버 (Eqp Server, Windows NT)는 <그림4>에서 보듯이 기능적인 측면에서 4 부분 (Message Interpreter, Logic Server, SECS Interpreter, Traffic Manager)으로 나누어져 있다. 그리고 하나의 장비 서버는 1대 이상의 장비를 제어할 수가 있다. 따라서 여러대의 장비로부터 Host로 가는 Message와 그것들에 대한 Host의 Reply Message를 통제하고 배분하는 기능을 수행하는 Traffic Manager가 필요하다. Message Interpreter와 SECS Interpreter는 각각 Host와 Eqp Server간과 Eqp Server와 장비간 Message Parsing (또는 Interpreting) 기능을 수행한다. 이 Interpreter들은 외부로부터 오는 Message를 Eqp Server의 Message



<그림 4> Eqp Server 시스템 구성도

Format으로 전환하거나 그 반대로 Eqp Server로부터 나가는 모든 Message를 재전환하며 특히 SECS Interpreter는 SECS I 사양을 검증하고 Error시 Error처리 기능도 포함하고 있다. Logic Server는 SECS 통신 Message Sequence에서 정의된 장비 Process에 대한 제어 Logic 뿐만 아니라 기타 Non-sequential Event들에 대해서도 처리하는 기능을 수행한다. 이에 대해서는 다음 절에서 자세하게 설명되어진다.

한편 Info Table은 현재 Logic Server가 진행하고 있는 특정 Lot 및 장비에 대한 모든 정보를 일시 저장하고 요청 시 제공한다. Info Table에 있는 Data는 Host와 장비로부터 받은 것들이다.

#### 5.2. Logic Server

Logic Server는 모든 장비에 대해서 공통적인 생산 및 물류 제어기능을 포함하기도 하며 장비의 종류 및 특성에 따라서 차이가 있을 수도 있다.

Logic Server의 기능은 크게 두 가지로 구분할 수 있다. 그것은 SECS 통신 Message Sequence에서 정의된 Logic 처리 기능과 그의 여기에서 정의되지 않은 Non-sequential Event Logic 처리 기능이다. Non-sequential Event는 어떤 순서에 따라서 일어나는 Event가 아니라 무작위로 일어나며 일반적으로 생산제어에 필요한 부수적인 기능들이다. Non-sequential Event의 대표적인 예로서 Port Status Checking, 장비 상태 변경, Lot List 요청, Error발생 시 Error Message 발생, TrackIn 취소, 불량 Glass 발생 시 처리, Line Mode 변경(On-line 또는 Off-line), 물류반송 Mode 변경(AGV 또는 Manual-guided Vehicle) 등이 있다.

<그림 3>에서 이미 정의된 Process Start Logic 처리에 대한 Logic Server의 Algorithm을 <표4>에서 보여주고 있다. <그림 3>에서는 Glass를 담고 있는 Cassette가 장비 Input Port (두개의 InPort중 하나)에 올려지는 순간 TrackIn이 시작되고 Host로부터 TrackIn 허가를 Eqp Server가 받은 후 차례로 장비에 Eqp Server가 On-line (Remote) Start를 명령하게 되어 있다. 따라서 Process Start는 Load Event와 함께 시작 된다. 때때로 Lot이 이미 TrackIn된 상태에서 Cassette가 Port로부터



〈표 4〉 Process Start Logic Algorithm

```

Select Case PortEvent. CEID
  Case Load
    {Select PortID
      Case InPort#1{
        Update Info Table;
        Call RemoteStart( );}
      Case InPort#2{
        Update Info Table;
        Call RemoteStart ( );}
    }
  Case UnLoad
    {Select PortID
      Case OutPort#1
        {....}
      Case OutPort#2
        {....}
    }
  Remotestart( )
    {If Port.CSTID=TI.CSTID or Port.Lot ID=TI.CSTID
      Then {
        Update Info Table;
        Send Start Message to Secs Interpreter;
      }
      Else {
        Call TrackIn_Req;
        Update Info Table;
        Send Start Message to SECS Interpreter;
      }
    }

```

잠시 Unloading된 경우도 발생할 수 있다. 이때는 RemoteStart( ) Procedure에서 TrackIn 유무를 Check한 후 TrackIn이 되어 있는 경우에는 Remote Start Message를 SECS Interpreter에 보내고 그렇지 않은 경우에는 Host로 TrackIn Request Message를 보낸 후 Remote Start Message를 보낸다. 그러므로 RemoteStart( )에서는 Port로부터 올라온 Cassette ID (또는 Lot ID)와 TrackIn시 Host로부터 내려온 Cassette ID (또는 Lot ID)를 비교함으로써 특정 Lot에 대한 TrackIn 유무를 확인 한다.

## 6. SECS Data 통신 Simulation

장비와 Eqp Server간 SECS 메시지 Sequence, 생산 제어 Logic 및 Data Format Checking을 위해서 SECS

통신 Simulation Tool을 이용하여서 Simulation의 실시가 필요하다. Simulation은 실제 장비와 Eqp Server 사이의 통신 메시지를 해석 하여서 분석하며 Error 수정은 물론이며 SECS Message Sequence와 생산제어 Logic도 개선 할 수 있다. 〈표5〉는 Simulation Tool인 SECSIM (GW Associates Inc.)와 Eqp Server를 이용하여 Process End Report (S6F11, CEID=4)에 대한 SECS 통신 Simulation의 Log File을 보여주고 있다.

Process End Report는 Process Start Report(S6F11, CEID=3) 메시지 구조와 Item종류 및 Data Format이 동일 하다. 여기에서는 송신 메시지인 S6F11(CEID=4)과 그것의 Reply Message인 S6F12 만을 보여주고 있지만 실제 Simulation 에서는 모든 SECS Message Sequence들의 연속적인 통신 내용을 볼수 있다.

〈표5〉의 맨 좌측열은 통신 메시지 송수신 시간을 나타낸다. 모든 송수신 메시지는 Hexadecimal Code(16진수)로 표기되며, 이는 다시 Text Type으로 변환되어서 Hexadecimal Digital Code와 함께 나란히 보여 준다. Log File에서 보여주고 있듯이 메시지를 보내기 전 언제나 Enquiry(ENQ)와 OK(EOT)에 의한 Handshaking을 통해 통신 준비 확인을 한 다는 것을 알 수가 있다. 그리고 메시지를 받는 측(Receiver)은 수신 직후 반드시 수신 유무(ACK=정확한 수신 또는 NAK=불량한수신)를 Reply 해야만 한다. Primary 메시지에서 Hexadecimal Code가 먼저 위치하고 Text Type이 나중에 명시되며 반면에 Secondary(Reply) 메시지에서는 그 반대이다. 메시지 Sequence Check 시에는 주로 Text Type을 이용 하며 통신 Error Check나 Data Format Check시에는 Hexadecimal Code를 이용 한다.

두 번 째 수신(RECD) 메시지의 첫 줄 11 Byte(CC부터 6B까지)는 S6F11의 Length Byte와 Header에 해당 한다. Length Byte(1 Byte) 'CC'는 자신을 제외하고 총156 Byte의 Data를 받았음을 알 수 있다. Device ID (2 Byte, '80 01') Filed를 보면 S6F11이 장비에서 Host로 보낸 메시지임을 알 수 있는데, 이는 가장 좌측의 '8'을 Binary로 펼칠 때('1000') Device ID의 가장 좌측에 위치한 R Bit의 값이 '1'임을 알 수 있는데, 이는 SECS I 규정에 의해 장비에서 Host로 보낸 메시지임을 나타낸다. 그리고 Device ID의 나머지 Bit는



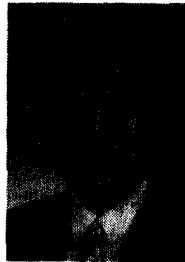
과적인 결과를 얻을 수가 있다.

앞으로 연구과제는 TCP/IP Protocol을 사용하여 SECS II 메시지를 통신 하는 것과 Dynamic SECS 메시지 Sequence 사용이다. TCP/IP는 기존의 SECS I Protocol보다 단위 시간당 보다 많은 양의 Data를 통신 할 수 있을 것으로 기대 된다. 그리고 지금과 같은 미리 고정된 SECS Message Sequence를 사용 하기 보다는 생산 환경의 변동에 따라서 보다 능동적으로 대응이 가능 할 것으로 기대가 되는 Dynamic SECS Sequence 개발이 요구 된다.

### 【참 고 문 헌】

- [1] SEMI International, Book of SEMI Standards: Equipment Automation/Software Volume , SEMI International, Mountain View, CA, 1993.  
 [2] GW Associates Inc., The SECS Standards Documents , GW Associates Inc., LA, CA, 1992.

- [3] Stallings, William, Data and Computer Communications , Macmillan Publishing Co., New York, 1988.  
 [4] Dertsekas, Dimitri and Gallager, Robert, Data Networks. Prentice-hall, Inc., New Jersey, 1987.  
 [5] Rorabaugh, Britt, Data Communications and Local Area Networking Handbook , TAB Professional and Reference Books Inc., PA, 1985.



조민호

1984년 조선대학교 산업공학과 학사  
 1993년 미국 Lehigh University 산업공학과 박사

현 재 삼성전자반도체부문 정보 및 생산자동화 선임연구원

관심분야: 정보통신(SI, AI응용, Diagnostics, ATM), 생산자동화(MES, 물류시스템, Robotics)