

**반도체 제조 공정에서 장비와 호스트간 SECS 프로토콜 개발**

김대원, 전종만, 이병훈, 김홍석, 이호길

명지대학교 정보제어공학과(Tel: 0335-330-6472; E-mail: dwkim@wh.myongji.ac.kr)

한국생산기술연구원

**Development of the SECS Protocol between Equipments and a Host in a Semiconductor Process**

Dae-Won Kim, Jong-Man Jeon, Byong-Hoon Lee, Hong-Seok Kim, Ho-Gil Lee  
Department of Information Control Engineering, Myongji University  
KITECH

**Abstract** - 본 논문에서는 반도체 제조 공정에서 장비와 호스트간에 통신을 할 수 있는 SECS(SEMI Equipment Communications Standard) 프로토콜의 개발을 제안한다. SECS 프로토콜은 메시지 전송을 위한 헤더 부분을 정의하는 SECS-I 프로토콜과 메시지 내용을 정의하는 SECS-II 프로토콜로 나뉘어지는데, RS232 시리얼 통신을 하는 SECS-I 프로토콜 대신에 이더넷(ethernet)을 통해 TCP/IP 통신을 할 수 있는 HSMS 프로토콜을 구현하고자 한다. HSMS(High-speed SECS Message Services) 프로토콜은 SECS-I과 마찬가지로 SECS-II 메시지 내용을 전송할 수 있도록 10바이트 크기의 헤더로 정의된다. HSMS 프로토콜 통신은 TCP/IP를 기반으로 하기 때문에 SECS 메시지 전송을 위한 통신 선로를 설정하기 위해 소켓 API를 응용하고 항상 통신 대기상태를 유지하기 위해 데몬(daemon) 형태로 구성한다. 실제 메시지 내용을 정의하고 있는 SECS-II 프로토콜은 데이터 인덱스 테이블과 표준에 정의된 형식에 맞게 파일형태나 DLL(Dynamic Link Library)형태로 구성하고 프로세스 프로그램(process program)을 수행하기 위해 SECS 프로토콜 표준에서 정의하는 SML(SECS Message Language)형식으로 변환할 수 있는 스크립트 변환기(script translator)를 구현한다. 또한 HSMS 프로토콜이 전송할 SECS-II 메시지를 저장하기 위한 파라미터를 정의하고 실제 통신을 위한 테스트 베드를 위한 응용 프로그램을 제작한다.

**1. 서론**

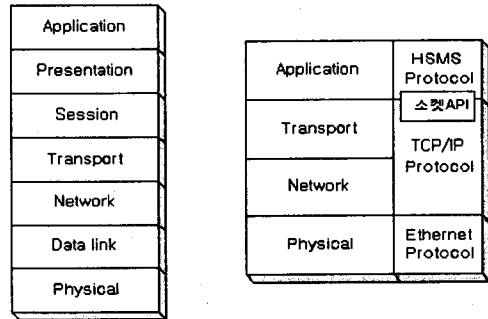
FA(Factory Automation)시장을 살펴보면, 대부분의 기기들이 서로간에 데이터 통신을 할 수 있는 시스템으로 전환되고 있으며 실제 DeviceNet이나 ControlNet등을 이용한 통신 시스템을 구축하고 있다. 이러한 경향은 반도체 장비에서도 같이 적용되고 있는 실정이며, 이들 장비간에 상호동작성(interoperability)을 보장하여 시스템 통합이 용이하도록 하고, 시스템의 유연성, 확장성, 유지보수성을 제고시키는 개방형 통신 기술의 요구가 증대되고 있다. 이에 본 논문은 반도체 제조공정에서의 기기들간의 통신을 할 수 있는 SECS 프로토콜 개발을 제안한다. 반도체 장비 통신 프로토콜 표준인 SECS는 1970년 후반 반도체 제조업체였던 휴렛 팩커드가 자체 내에서 사용할 통신 표준을 정의하고자 하는 시도에 의해 자사내 Deer Creek 실험실의 클리어에 의해 수행된 일부 작업에서 발전되었다. SECS 표준은 메시지 전송 표준인 SECS-I과 메시지 내용 표준인 SECS-II의 두 부분으로 나누어지는데, 본 논문에서는 RS232 시리얼 통신을 하는 SECS-I 프로토콜 대신에 이더넷을 통해 TCP/IP 통신을 할 수 있는 HSMS 프로토콜을 구현하고 SECS-II 메시지 전송을 HSMS를 이용해 호스트와 장비간의 SECS 프로토콜 통신을 가능하게 한다. 메시지 전송을 위한 헤더부분을 정의하고 있는 HSMS 프로토콜은

TCP/IP 프로토콜을 이용한 통신이기 때문에 소켓API를 응용해서 구현이 되고 SECS-II 프로토콜과의 상호작용을 위한 인터페이스를 DLL형태로 제공한다. 또한 실제 메시지 내용을 정의하고 있는 SECS-II 프로토콜은 SECS 프로토콜 표준<sup>[2]</sup>에 정의된 SML 형식에 맞게 표준 SECS 메시지를 파일 형태로 구성하고 SML을 해석할 수 있는 변환기를 구현한다. 본 논문의 본문에서는 통신 프로토콜의 표준 모델인 OSI 7계층과 SECS 프로토콜의 계층을 비교 분석하고 HSMS 프로토콜의 헤더 부분을 소개한다. 또한 SECS-II 프로토콜의 메시지 형식의 정의를 살펴보고 실제 구현에 관한 내용을 기술한다.

**2. HSMS 프로토콜과 SECS-II 프로토콜**

**2.1 OSI 7계층과 SECS 프로토콜 계층**

아래 그림2-1에서 보여지는 것처럼 SECS 프로토콜 계층은 OSI 7계층과 비교해 볼 때 물리적 계층, 네트워크 계층, 트랜스포트 계층 그리고 어플리케이션 계층으로 구분된다. 이더넷을 기반으로 TCP/IP 프로토콜을 이용해 HSMS 프로토콜을 통한 SECS 메시지 전송을 계층별로 보여주고 있다.



OSI 7계층 표준 통신 모델

SECS 프로토콜 계층

그림 2-1 프로토콜 계층 비교

**2.2 HSMS 프로토콜 Review**

HSMS는 SECS-I의 대안으로 개발된 프로토콜로서 반도체 제조 공정에서 컴퓨터간에 메시지 교환을 목적으로 하는 통신 인터페이스로 정의할 수 있다. 먼저 HSMS 프로토콜 표준에서 정의한, 본 논문에서 사용되는 용어를 정의한다.

- 스트림(Stream): 특정 행위에 대한 메시지 목록
- 함수(Function): 특정 행위에 대한 특정 메시지
- 엔티티(Entity): 장비와 호스트의 통신을 위한 응용 프로그램

- 세션(Session): 두 개의 엔티티간에 메시지의 전송을 위해 설립된 관계
- 세션 아이디(Session ID): 세션의 고유 번호
- 컨트롤 메시지(Control Message): HSMS의 세션 관리를 위한 제어 메시지
- 1차 메시지(Primary Message): 데이터 전송의 첫 번째 메시지
- 2차 메시지(Secondary Message): 1차 메시지의 응답 메시지

### 2.2.1 HSMS 프로토콜 헤더

HSMS 프로토콜의 메시지 형식은 아래 그림2-1과 같은 헤더는 총 10바이트로 구성된다.

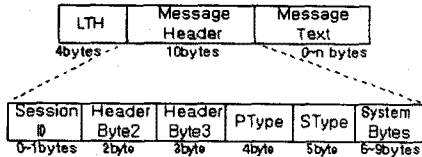


그림 2-2 HSMS 프로토콜 헤더

LTH는 4바이트를 차지하며 전송되는 메시지 길이를 나타내고 메시지 텍스트(message text)는 0~n 바이트까지의 메시지를 나타내는데 실제 전송되는 최대 메시지 길이는 0~2<sup>32</sup>까지 가능하다. 헤더 부분의 세션 아이디(Session ID)는 세션의 고유번호를 나타내고 이것은 응용 프로그램에서 부여 할 수 있다. 2번째 그리고 3번째의 헤더 바이트(header byte)는 전송되는 메시지가 컨트롤 메시지인 경우에는 상태에 따라 0과 1로 나타내어지고 SECS메시지인 경우에는 스트림과 함수를 나타낸다. PType은 SECS 메시지로 인코딩(encoding)된 경우에는 0으로 나타내어지고 SType은 컨트롤 메시지에 따라 값이 정해진다<sup>(1)</sup>.

### 2.2.2 데이터 전송을 위한 연결 설정

아래 그림2-3는 HSMS의 TCP/IP 프로토콜을 이용한 통신 상태 다이어그램을 보여주고 있다.

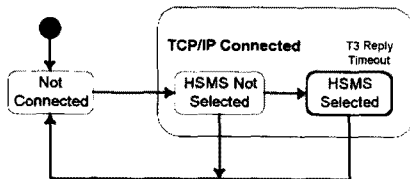


그림 2-3 연결 상태 다이어그램

SECS 프로토콜 통신을 통해 메시지를 전송하기 위해서는 먼저 통신 선로가 설립이 되어야 한다. 위 그림에서 보여 주듯이 초기 상태는 TCP/IP의 비연결(not-connected) 상태다. TCP/IP가 연결(connected) 상태가 되면 HSMS 선택(selected)과 비선택(not-selected) 상태에서 HSMS 선택 상태가 되면 메시지를 전송 할 수 있는 선로가 완성된다.

표 2-1 소켓 API 함수와 컨트롤 메시지

소켓 API 함수들	컨트롤 메시지
WSAStartup(),	
WSACleanup(), socket(),	Select.req, Select.rsp.
bind(), listen(),	Deselect.req, Deselect.rsp.
connect(), accept(),	Linktest.req.
recv(), send(),	Linktest.rsp, Reject.req.
closesocket()	Separate.req

TCP/IP 연결은 소켓 API<sup>5)</sup>를 응용해서 이루어지고 HSMS 선택은 컨트롤 메시지에 의해 이루어진다. 위 표

2-1에서는 실제 응용되는 함수와 HSMS 표준에 정의된 연결 설정을 위한 컨트롤 메시지를 보여준다.

### 2.3 SECS-II 프로토콜 메시지 형식

SECS-II 프로토콜은 반도체 제조공정에서 호스트 컴퓨터와 장비와의 정보교환을 위한 메시지 형식을 정의한다. SECS-II 메시지는 헤더를 정의하고 있는 HSMS 프로토콜에 의해 전송된다. SECS-II 메시지는 반도체 제조공정에서 필요한 표준 메시지들을 정의하고 있으며 각 장비의 특성에 맞게 새로운 메시지를 사용자가 정의 할 수 있다. 각 메시지들은 비슷한 사용목적과 연관성 있는 활동, 예를 들면 제어 프로그램 전송, 화학물질, 제조물체등의 이동 정보, 측정된 데이터, 누적된 데이터, 알람등의 범주로 나뉘는데 이를 스트림이라 한다. 스트림은 다시 동작에 대해 세부적으로 나뉘어지는데, 이 세부적인 항목을 함수라 한다. 스트림과 함수는 SECS-II 표준 메시지 정의 부분과, 새로운 메시지를 사용자가 정의하여 사용할 수 있도록 사용자 정의 부분으로 나뉜다. 아래 그림2-4는 스트림과 함수의 사용 범위를 나타낸다<sup>(2)</sup>.

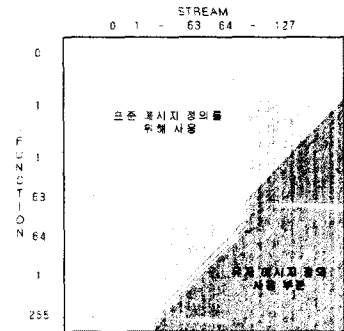


그림 2-4 STREAM 과

### FUNCTION의 할당

SECS-II 메시지 교환을 트랜잭션이라 하는데, 하나의 트랜잭션은 1차 메시지 전송과, 이에 대한 응답으로 보내는 2차 메시지 전송으로 구성된다. 1차 메시지는 항상 홀수 함수(odd function)로 표현하고, 2차 메시지는 항상 짝수 함수(even function)으로 표현한다. 호스트에서 장비, 장비에서 호스트로 보내는 데이터, 즉 메시지는 논리적 분할을 위해 리스트(list)와 아이템(item)의 구조로 이루어진다. 리스트와 아이템은 각각 헤더를 가지고 있고, 아래 그림2-5는 리스트와 아이템의 헤더를 나타낸 것이다.

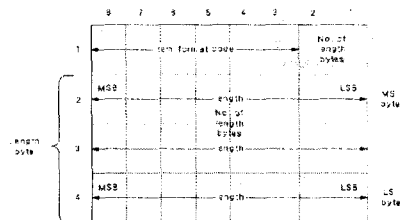


그림 2-5 리스트와 아이템 구조

아이템은 정보의 최소단위로 아이템 헤더에는 아이템의 형식, 아이템의 길이를 나타내는 길이 바이트로 구성된다. 여러 개의 아이템을 보내기 위해 아이템의 앞 부분에 리스트가 붙는데 리스트 헤더는 아이템 형식 코드(format code)에 0을 넣어 아이템 헤더와 구분하고, 길이 바이트에는 아이템의 개수를 기록한다.

지금까지는 SECS 프로토콜을 구성하고 있는 HSMS 전송 프로토콜과 SECS-II 메시지 프로토콜에 대한 표준을 기술하였고 다음은 SECS 메시지를 실제 전송할 수 있는 SECS 프로토콜 통신을 위한 드라이버 구현 방법에 대한 내용을 기술한다.

### 3. SECS 프로토콜 통신을 위한 드라이버와 응용 프로그램 구현

#### 3.1 HSMS 프로토콜 드라이버 설계 및 구현 방법

HSMS 프로토콜 아래 그림 3-1에서 보여 주듯이 세 단계를 거쳐 통신 선로를 성립한 후 SECS 메시지를 전송하는 과정으로 구성된다.

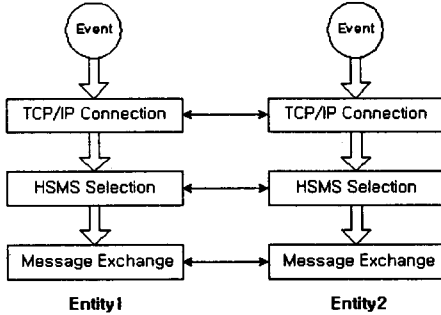


그림 3-1 HSMS 통신 프로토콜 흐름도

이런 과정에서 설정되는 통신 타임아웃<sup>(1)</sup> 파라미터 값들은 아래 표 3-1에서 보여준다.

표 3-1 타임아웃 파라미터

파라미터 이름	일반적인 값
T3(Reply Timeout)	45 sec
T5 Connect Separation Timeout	10 sec
T6 Control Transaction Timeout	5 sec
T7 NOT SELECTED Timeout	10 sec
T8 Network Intercharacter Timeout	5 sec

HSMS는 SECS 프로토콜 통신을 할 수 있도록 하는 통신 선로의 대문 역할을 맡고 있다. 그래서 항상 감시자처럼 이벤트를 기다리고 있는 데몬 형태로 대기하고 있어야 하고 실행 파일의 크기가 크지 않아야 한다. 이러한 형태는 DLL을 이용하는 방법으로 구현이 되어져야 한다. 다음은 HSMS 프로토콜 드라이버의 모듈별 구분과 상호작용을 보여주는 그림이다.

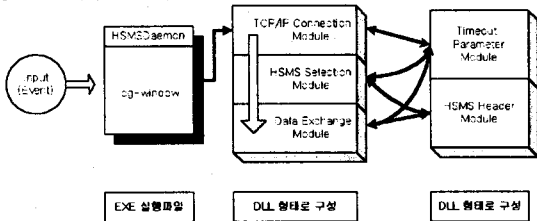


그림 3-2 HSMS 프로토콜 드라이버의 모듈별 구분 TCP/IP 연결, HSMS 선택 그리고 데이터 전송 모듈을 하나의 컴포넌트로 구성을 하고 타임아웃 파라미터 모듈은 메인 프로그램에서 관리를 하면서 독립적으로 수행이 되는 형태로 구성을 한다. 또한 HSMS 헤더를 구성하는 모듈을 구성하고 각각을 독립적인 DLL 형태로 구성한다.

#### 3.2 SECS-II 응용 프로그램 설계 및 구현 방법

SECS-II 프로토콜의 응용 프로그램은 HSMS 드라이버가 헤더를 분석하고 난 후 헤더와 메시지를 정의된 파라미터를 통해 보내면 SECS-II는 표준 메시지를 정의한 SML(SECS Message Language) 파일을 참조하여 데이터구조를 해석한다. 데이터구조를 해석한 후에 필요한 정보를 응용 프로그램이 이벤트 버스를 통하여 다음 단계로 보내준다. 만약 응답이 필요한 메시지면 이벤트 버스를 통해 장비에서 측정된 데이터를 얻어 2차 메시지를 제

작한 후 HSMS에게 보내준다. 응용 프로그램은 매퍼 메모리(mapper memory)를 사용하여 호스트에 보낼 데이터만을 가지고 있고 SECS-II가 요청시에 이벤트 버스에 올린다. SECS-II 부분에서 해야할 일은 HSMS에서 받은 스트림을 SML 변환기를 통해 해석하면 응용 프로그램이 다음 단계로 보내주고, 이에 대한 응답인 2차 메시지를 만드는 작업을 하게 된다. 이 때에 이미 표준에 의해 정의한 SML 파일을 참조하는데 SML 파일은 응용 프로그램에서 동작을 위해 접근하고 관리한다. 아래 그림 3-3은 장비와 호스트간에 SECS 프로토콜 통신이 가능하도록 구현된 응용프로그램을 보여준다.

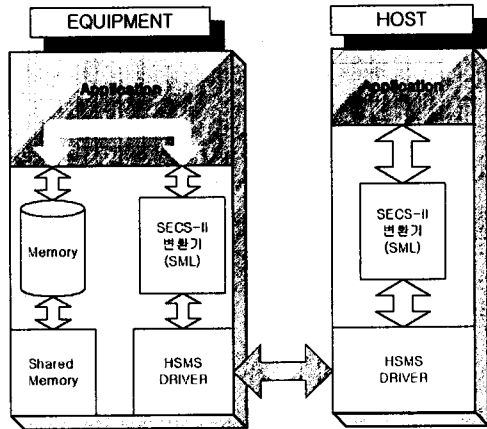


그림 3-3 장비와 호스트간의 통신 응용프로그램

#### 구조

위에서 구현된 응용 프로그램을 장비에 탑재된 컴퓨터와 호스트 컴퓨터에 직접 설치하고 SECS 프로토콜 통신을 테스트한다. 통신되는 과정은 HSMS 데몬에서 모니터링(monitoring)한다.

### 4. 결론 및 향후 발전 계획

자동화 추세에 따라, 기기들간의 네트워크 통신이 필수적이 되면서 그 적용 범위도 넓어지고 있다. 본 논문에서는 반도체 제조 공정에서의 기기들간에 통신을 통한 공정의 운영 상태 및 생산 제품 정보를 얻을 수 있도록 하는 SECS 프로토콜 개발을 제안했고 실제 SECS 통신을 할 수 있는 응용 프로그램을 구현하였다. 향후 계획으로는 반도체 제조공정의 일반적인 공정 과정을 컴포넌트 형태로 구성하여 서비스를 제공할 수 있는 표준 모델인 GEM을 구현함으로써 통합서비스를 할 수 있는 환경을 구성하고 객체 지향적인 개방형 시스템을 구성하는 것이다.

### 5. 참고 문헌

- [1] SEMI E37-0298 "HIGH-SPEED SECS MESSAGE SERVICES(HSMS) GENERIC SERVICES"
- [2] SEMI E5-0999 SEMI EQUIPMENT COMMUNICATIONS STANDARD 2 MESSAGE CONTENT (SECS-II)
- [3] SEMI E4-0699 SEMI EQUIPMENT COMMUNICATIONS STANDARD 1 MESSAGE TRANSFER(SECS-I)
- [4] SEMI E30-0299 GENERIC MODEL FOR COMMUNICATIONS AND CONTROL OF SEMI EQUIPMENT(GEM)
- [5] "Windows Sockets Network Programming" Bob Quinn & Dave Shute
- [6] "정보통신 프로토콜 공학" 한국전자통신연구원
- [7] "Visual C++ Programming Bible" 이상엽 저