

반도체 공정의 SCRUBBER 감시 시스템 개발

°김준환, 김상우, 김병진, 문학룡, 전희중
송실대학교 전기공학과

The Development of Monitoring System in the Scrubber of Semiconductor Manufacture Processing

°Joon-Hwan Kim, Sang-Woo Kim, Beung-Jin Kim, Hak-Yong Moon, Hee-Jong Jeon
Dept. of Electrical Eng., Soongsil University

Abstract - In this paper, we discuss the development of monitoring system with data process equipment which transfers data from Remote Terminal Unit(RTU) to monitoring computer. The RTUs sense temperature, pressure and PLC(Programmable Logic Controller) nodes conditions of scrubber in semiconductor manufacture processing. The data process equipment is connected every RTU and a monitoring computer through serial communication. This equipment receives informations from RTU, process data, and transfers them to monitoring computer. To avoid congestion in data communication, task scheduling algorithm used RT O/S(Real-Time Operating System) is embedded in ROM which is a part of data process equipment.

1. 서 론

현재 국내의 반도체 제품 생산량은 기술적인 측면에 있어서 세계적으로도 상위 그룹에 속하는 높은 경쟁력을 보유하고 있다. 그러나 이러한 반도체를 생산하는 장비에 관한 부문은 전량 외국 제품에 의지하고 있으며 심지어는 이러한 시스템의 관리 시스템마저도 외국 기술에 의지하고 있는 실정이다. 또한 최근의 생산 자동화 시스템에서는 컴퓨터를 이용하여 단위 공정의 자동화를 이루고, 이들을 통합하여 전체 공정을 일관되게 관리함으로써 생산성 향상, 생산비용의 절감을 비롯하여 생산 공정의 유지 및 관리에 유연성을 제공하는 등의 효과를 거두고 있다. 이러한 자동화 시스템을 구축하기 위해서는 각각의 공정에서 생성되는 정보들을 수집, 분석, 가공 및 저장하고 또한 각 공정에서 필요한 정보를 적시에 제공하기 위한 정보의 통합화를 이뤄내야 한다.

본 논문에서는 반도체 제조공정 중 scrubber 공정의 온도, 압력 및 PLC 접점 상태의 데이터들을 수집하고, 이 수집한 데이터를 통하여 공정의 동작상태와 이상 여부를 판별하는 모니터링 시스템을 개발하였다. 특히 여러 개의 RTU와 데이터 처리 장치간의 통신 중에는 같은 전송매체를 사용하기 때문에 예기치 못한 데이터 충돌이나 데이터 폭주 등과 같은 현상이 발생할 수 있다. 이러한 현상들을 방지하기 위해 전송매체 공유방법과 스

케줄링 전략이 필요하다.

데이터 수집 및 처리장치에서는 RTU가 전송하는 여러 가지 센서들의 신호를 일정시간 내에 처리할 필요가 있다. 인터럽트(interrupt)를 이용하는 방식은 타이머가 주기적으로 인터럽트를 발생시켜 알고리즘을 처리한다. 이 방법은 프로그래밍이 용이하고 시스템이 안정적으로 운영되지만, 서로 다른 주기를 가진 알고리즘을 동시에 수행할 때에는 각 알고리즘의 주기를 유지하기 어렵고 많은 이벤트(event)가 동시에 발생할 때는 각 알고리즘의 종료시간을 예측할 수 없는 시스템 폭주가 일어날 수 있다. 따라서, 본 논문에서는 시스템이 필요로 하는 기능들을 여러 가지 태스크(task)로 나누어 멀티-태스킹(multi-tasking)을 할 수 있도록 하였다. 이 방법은 구현이 다소 어렵지만 여러 가지 이벤트를 효과적으로 처리할 수 있는 장점이 있다.

그러므로 본 논문에서는 RT O/S를 이용하여 RTU와 데이터 처리 장치간의 데이터 통신 작업을 여러 태스크로 나누어 스케줄링 하는 기법을 제시하고, 또한 모니터링 컴퓨터로는 수집된 정보를 토대로 데이터 베이스를 구축하였다.

2. 본 론

2.1 시스템의 구성

그림 1과 같이 전체 시스템은 RTU, 데이터 처리 장치와 모니터링 컴퓨터로 구성된다. RTU는 공정의 여러 물리량과 디지털 접점 상태를 수집하고, 데이터 처리 장치는 스케줄링 알고리즘을 사용하여 RTU가 수집한 데이터들을 적절하게 모니터링 컴퓨터로 전송한다. 모니터링 컴퓨터는 수신한 데이터를 이용해 데이터 베이스를 구축하고, 또한 현재의 상태를 사용자에게 알려주고 사용자로부터 명령을 받기 위해 MMI(Man Machine Interface) 기능을 수행한다.

시스템의 입력은 scrubber 공정의 온도를 0~800℃ 범위로 검출할 수 있는 온도 센서, 압력 센서 등의 아날로그 접점과 반도체 제조장비의 동작을 알 수 있도록 하는 디지털 접점을 이용하여 데이터를 수집한다. RTU는 수집된 데이터를 RS-485 통신을 통하여 데이터 수집 및 처리 장치에 전달한다. 그리고 데이터 처리 장치는 모니터링 컴퓨터와 RS-232 통신을 통하여 수집된 정보

를 전송하고 이상 감지 시에는 알람(alarm) 신호를 발생시켜서 제조공정을 적절하게 운영되게 한다.

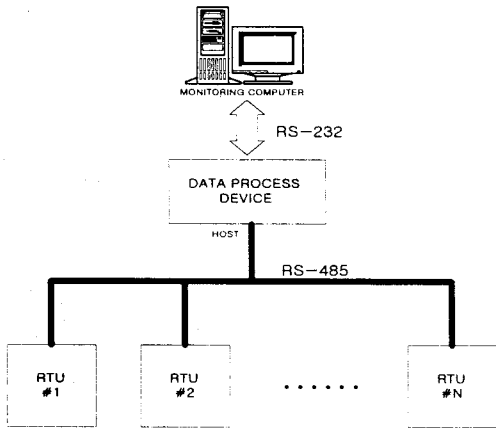


그림 1. 전체 시스템 블록도

RTU는 그림 2와 같이 각 scrubber에서 검출되는 온도 신호들과 압력 신호들을 센싱해서 디지털 신호로 바꾸어 주는 A/D 변환부와 PLC의 디지털 점점상태를 입력을 받기 위한 디지털 입력부, 초기상태 설정 및 디스플레이를 위한 키보드 및 7-세그먼트(7-segment)로 구성된 인터페이스부, 그리고 데이터 처리 장치와의 통신을 위한 통신부로 구성되어 있다.

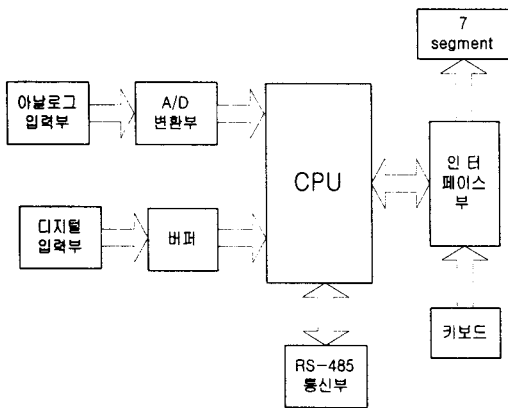


그림 2. RTU의 블록도

2.2 시스템의 통신

RTU는 데이터 처리 장치, 데이터 처리 장치는 모니터링 컴퓨터와 통신하여 센싱한 데이터를 전송한다. 이를 위한 통신 방식으로 첫째, Polling/Selecting 방식은 매체의 사용권을 시스템 내에 하나만 존재하는 Master가 관리를 하고, Slave는 Master의 허가가 나왔을 때만 매체의 사용이 허가된다. 이러한 방식은 하나의 Master가 여러 개의 Slave들을 순차적으로 호출하여 데이터 통신을 하므로 Slave의 수가 많으면 통신 지연이 심화되고, 최악일 경우에는 데이터들이 Slave의 버퍼를 초과 할 수도 있다. 또 다른 방식인 토큰패싱(Token Passing)은 토큰이라는 제어 패킷을 항상 장

치 안에서 순환시켜 통신을 한다. 송신 요구를 발생한 장치는 토큰이 돌아올 때까지 송신을 기다리고, 토큰이 돌아오면 1패킷만 송신권을 얻어 1패킷 송신을 행하고 다음 장치에 패킷을 돌린다. 마지막으로 CSMA/CD (Carrier Sense Multiple Access with Collision Detection) 방식은 매체에 접속되어진 각 장치가 송신 요구를 하였을 때 매체가 사용중인지를 검출하여 아무도 사용하지 않으면 송신을 개시하는 다중 액세스(access) 방식이다. 만일 거의 동시에 송신할 장치가 있을 경우에는 패킷의 충돌이 일어나고 이를 검출한다. 그리고 각 장치는 랜덤한 시간을 기다린 후에 통신매체를 사용하거나 검출하는 동작을 반복하여 재전송한다.

CSMA/CD는 송신권을 확률적으로 확보하기 때문에 통신매체의 트래픽이 적은 경우에도 전송 효율은 뛰어나지만, 트래픽이 증가하면 효율이 저하되는 특성을 가진다. 토큰패싱은 제어 패킷인 토큰을 순환시켜 송신권을 주므로 트래픽이 적어도 어느 정도 오버헤드(overhead)가 걸리지만 트래픽이 증가해도 효율이 급격하게 저하되는 일은 없다.

2.3 시스템 스케줄링

데이터 처리 장치는 데이터를 효과적으로 전송하기 위해 RT O/S를 이용한 태스크 스케줄링을 한다. RT O/S는 선점 가능한(preemptible) 멀티쓰레드(multi-thread)를 지원하므로써, 현재 자원을 사용하는 태스크보다 우선 순위가 높은 태스크가 생성되면 CPU를 현재의 태스크로부터 가로채 우선 순위가 높은 태스크를 먼저 수행하도록 한다. 또한 태스크의 우선 순위에 의한 자원의 할당이나 태스크간의 선행 관계를 위해 동기화(synchronization) 메커니즘을 지원하고, 메모리와 I/O를 관리한다. 그리고 IPC(Inter Process Communication)를 지원하여 태스크들 서로간의 정보 교환을 가능케 하며 시간적인 제약 조건들을 지원한다. 이런 RT O/S의 기능 중 태스크간의 동기화(synchronization), 우선 순위(priority) 및 자원의 할당을 위해 세마포어(semaphore), 메일박스(mailbox) 그리고 메시지 큐(message queue)등을 사용한다. 먼저 세마포어는 공유 자원의 액세스를 제어하고, 이벤트가 일어나는 것을 알리며 태스크간의 동기화를 위해 사용된다. 세마포어는 어떤 코드를 계속 실행하기 위해 취득하는 열쇠로서 이미 세마포어가 사용되고 있다면, 어떤 요청을 요구하는 태스크는 현재 사용중인 태스크가 세마포어를 풀어줄 때까지 실행대기 상태가 된다. 메일박스는 보통 포인터 변수로서 메시지를 커널이 제공하는 서비스를 이용하여 메시지를 태스크로 보낸다. 태스크나 ISR(Interrupt Service Routine)은 커널이 제공하는 서비스를 개입시켜 메시지(실제로는 포인터)를 메일박스로 넣는다. 마찬가지로, 한 태스크 혹은 복수의 태스크가 커널이 제공하는 서비스를 통해 메시지를 받을 수 있다. 메시지 큐는 메일박스의 배열로서, 메일박스와 메시지 큐와의 관계는 C언어의 포인터와 포인터 배열의 관계와 비슷하다.

태스크는 RTU로부터 전송되어진 데이터를 받는 패킷 수신 태스크와 모니터링 컴퓨터로 데이터를 보내는 패킷 송신 태스크로 나누어진다. 그리고 태스크간의 우선 순위와 데이터간의 우선 순위를 위해, 또 자원의 할당과

동기화를 위해서 세마포어와 메일박스, 메시지 큐를 사용하였다.

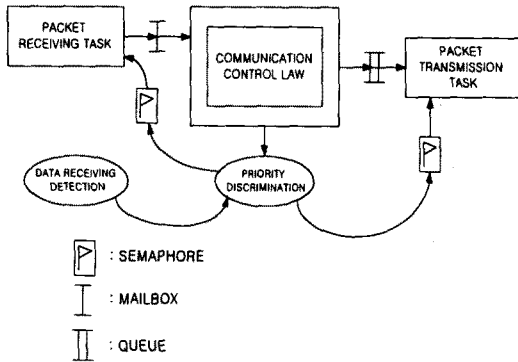


그림 3. 태스크 맵(TASK MAP)

그림 3은 태스크 스케줄링을 간단히 도식화한 태스크 맵(Task Map)이며, 각 태스크 수행 시에는 RT O/S의 커널을 호출하여 서비스를 받는다.

먼저 RTU에 의해 센싱되고 전송된 데이터는 데이터 처리장치로 전송되고, 그와 동시에 데이터 처리 장치는 RS-485 드라이버의 버퍼를 통해 수신을 감지하게 되고 데이터를 받을 수 있도록 패킷 수신 태스크를 동작시킨다. 이 때 데이터 처리 장치의 자원을 해당 태스크만 사용할 수 있도록 세마포어를 이용하고, 데이터를 읽은 후 그 데이터는 메일박스로 보내진다. 그리고 이 데이터를 모니터링 컴퓨터로 보내기 위해 패킷 송신 태스크를 생성시킨 후, 이 송신 태스크는 메시지 큐를 통해 데이터를 전송한다. 이 때 데이터의 우선 순위를 위하여 메일박스 대신 메시지 큐를 사용하게 되고 모니터링 컴퓨터로 데이터를 전송하기 위한 패킷 송신 태스크의 요청은 FIFO(First In First Out)에 의해 큐닝 된다. 따라서, 큐닝된 순서로 데이터를 모니터링 컴퓨터로 보내야 하지만, 만약 수신한 데이터가 송신할 데이터보다 우선 순위가 높을 시에는 송신할 데이터가 큐닝되어 있는 메시지를 우선 순위에 따라 바꾼다. 즉, 패킷 송신 태스크 큐의 메시지를 태스크의 우선 순위에 따라 바꿔 넣음으로서 시간순서가 아닌 그 시점에서의 우선 순위도에 따라 패킷 송신을 실행하는 것이다.

2.4 데이터 처리 장치의 구성

그림4와 같이 데이터 처리 장치는 외부의 RAM과 EPROM, RS-232, 485 드라이버, 그리고 CPU인 MC68302로 구성된다. 외부의 RAM과 EPROM은 시스템 버스에 연결되고, 모니터링 컴퓨터와의 통신을 위해서 RS-232 드라이버를 사용하였다. 하부 RTU와의 통신을 위해서는 MC68302의 SCC(Serial Communication Controller)와 RS-485 드라이버를 연결하였다.

데이터 처리 장치의 ROM에는 Memory, I/O, CPU 등의 자원을 효율적으로 관리하기 위하여 RT O/S가 내장되며, 또한 태스크들을 스케줄링하기 위한 스케줄링 알고리즘이 내장된다.

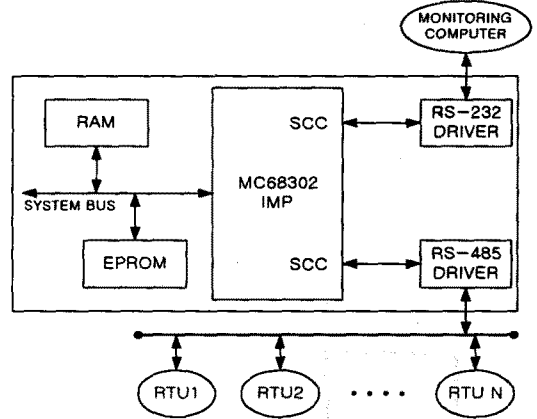


그림 4. 데이터 처리 장치의 블록도

3. 결 론

본 논문에서는 RTU로부터 전송되는 데이터를 모니터링 컴퓨터로 보내어 데이터 베이스를 구축하고, 이들간의 통신을 위해 RTU와 모니터링 컴퓨터 사이에 매개장치인 데이터 처리 장치를 개발하였다. RTU에서는 센서로 데이터를 수집하고, 데이터 처리 장치에서는 수집한 데이터들을 모니터링 컴퓨터로 적절히 전송하기 위해 RT O/S를 이용한 스케줄링 기법을 사용하였다. 또한 CPU, Memory, I/O등의 자원을 공유하고 데이터의 우선 순위와 H/W 자원의 할당을 위하여 세마포어, 메일박스, 메시지 큐를 사용하였다. 이렇게 구현된 전체 시스템은 반도체 공정에 대한 모니터링 시스템의 구축으로서 제품의 생산성, 장비의 유지관리, 장비의 보수 및 사고 예방 측면에서 중요한 역할을 담당할 수 있을 것으로 예상되며 반도체 제조 공정에 대한 모니터링 기술의 국산화와 컨트롤 기능의 부역을 통한 분산 제어 시스템으로의 확대화가 기대된다.

(참 고 문 헌)

- [1] K.M. Kavi and S.-M. Yang, "Real-Time Systems Design Methodologies: An Introduction and a Survey", The Journal of Systems and Software, pp85-99, April 1992
- [2] K. Arvind, K. Ramamritham, and J. Stankovic, "A Local Area Network Architecture for Communication in Distributed Real-Time Systems", Real-Time Systems, Vol. 3, No. 2, pp115-147, 1991
- [3] A. Van Tilborg and G.M. Koob (Eds.), "Foundations of Real-Time Computing. Scheduling and Resource Management", Kluwer Academic Publishers, Boston, 1991
- [4] Motorola, MC68302 Integrated Multi-protocol Processor User's Manual
- [5] Jean J. Labrosse, "μC/OS The Real-Time Kernel", R & D Publications, 1992