

FA정보시스템에서의 PLC 정보처리에 대한 연구

이헌준* · 김영태 · 김성권

A Study on a Communication Data Item and Method in PLC communication with Computer for FA Information System

Hun-Joon LEE · Young-Tae Kim · Sung-Kwun Kim

(Abstract)

FA SI(System Intergration)분야에 있어서 PLC(Programmable Logic Controller)와 컴퓨터간의 접속은 필수 불가결한 요소기술로 자리잡고 있으며, 이러한 기술변화에 맞춰서 기존 유니트의 기술적 발전뿐 아니라 타 유니트와 접속등의 네트워크에 관련된 내용이 하드웨어, 소프트웨어적으로 발전되고 있는 추세이다.

시스템통합을 하기 위해서는 기본적으로 PLC Networking을 하드웨어, 소프트웨어적으로 수행하여야 하나, 많은 연구들이 PLC 통신 유니트의 기술적 향상 및 표준화에 대한 부분으로 되어왔다.

본 논문은 정보시스템을 구축할때 PLC에서 처리하여야 하는 데이터, 혹은 컴퓨터와 송수신 받아야 하는 자료들에 대한 내용과 이들 자료를 PLC 내부에서 처리하는 방법론에 대해 기술코자 한다. 일반적인 Interface 방법으로 점접연결(Point to Point Connection)과 컴퓨터링크유니트에 대한 내용을 파악해보고, 설비고장진단 및 이상발생에 대한 추적이 가능하도록 PLC Memory내에 PLC접점데이터를 직접접근방식(Direct Accessing Method)과 간접접근방식(Indirect Accessing Method)으로 구분하고, 간접접근방식을 요소(Element), 동작(Event)에 의한 방법론을 이용하여 PLC DATA를 처리토록 하는 내용을 설명코자 한다.

주요어: PLC Interface, 직접접근방식(Direct Accessing Method), 간접접근방식(Indirect Accessing Method), 요소(Element), 동작(Event), 동작코드(Event Code).

1. 서론

PC와 PLC Interface부분은 하드웨어, 소프트웨어적으로 많은 방법과 많은 기술개발이 있으나, 실제 PLC의 점접연결(Point to Point)수준에서의 자료,

정보처리 만이 수행되어왔고, 이러한 작업은 컴퓨터 시스템설계자가 Interface부분의 정보 처리를 전부 맡아온 것이 현실이다.

이러한 내용은 정보시스템계층화의 개념에 위배될 뿐 아니라 시스템 기능 향상시 필요 자료등을 제공치

* 삼성전자 생산기술센터 자동화연구소 선임연구원.

못하고 단순 데이터 집계용으로 시스템의 기능, 성능이 축소 될 수가 있다.

2장에서는 PLC Interface를 컴퓨터가 PLC의 입출력 접점(Input /Output Point)을 직접 접근하느냐, 아니면 PLC Memory 영역을 통해 접근하느냐에 따라서 직접 접근방식(Direct Accessing Method)과 간접접근방식(Indirect Accessing Method)으로 구분 설명하였고, 간접접근방식(Indirect Accessing Method)에서 PLC Memory에 저장하는 데이터의 형태를 요소(Element), 동작(Event)으로 구분하여 PLC Memory에 Mapping 하는 절차에 대해 기술하였다.

2. PC와 PLC 접속방법

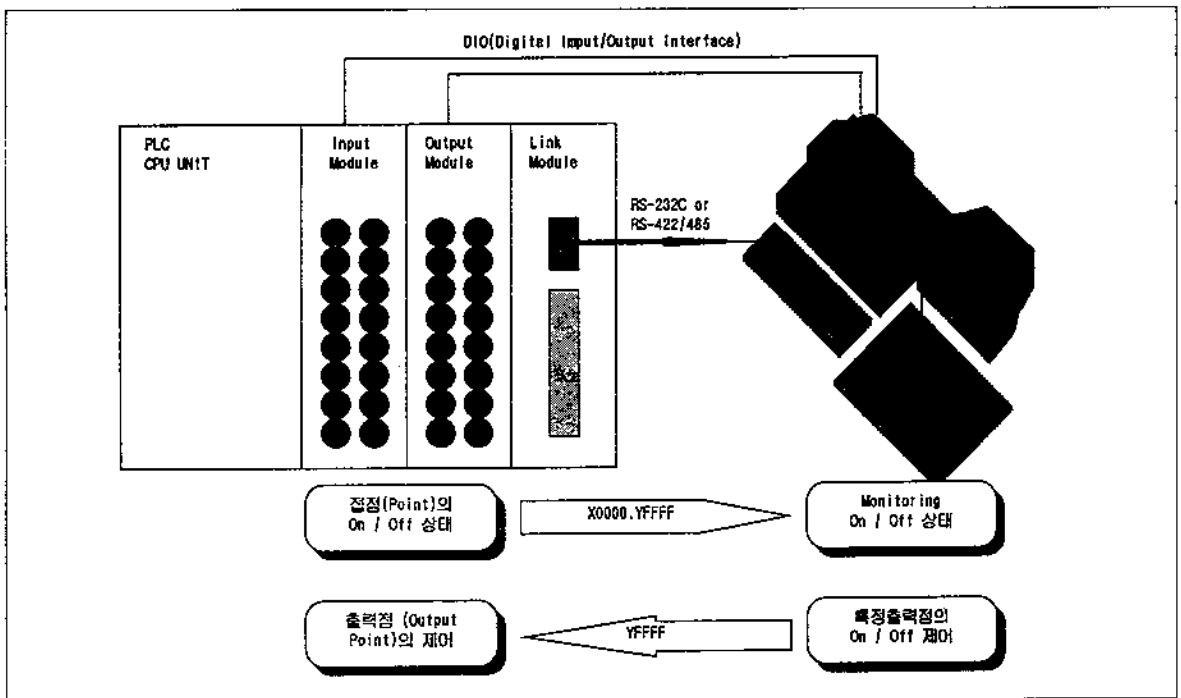
2.1. 직접접근방식(Direct Accessing Method)

직접접근방식이란 PLC의 입력점(Input Point), 출력점(Output Point)을 읽거나 쓰는 방식으로 대부분의 PLC 통신은 이러한 방식으로 컴퓨터와 주요 정보를

상호 교환하게 된다.

PLC면에서 볼때, 컴퓨터와의 인터페이스를 위해서는 하드웨어적으로는 접점을 연결하는 (DIO:Digital Input/Output Interface)구조나 PLC의 컴퓨터연결유닛(Computer Link Unit)을 이용하는 시리얼인터페이스(Serial Interface)구조를 이용할 수 있으며, 소프트웨어적으로는 모니터링용인 경우에는 PLC Program의 변경은 거의 없으며, 제어용인 경우 약간의 프로그램 수정 작업이 필요하게 된다. 실질적으로 Interface 위한 작업은 컴퓨터가 사용할 접점들의 정의(Definition)만 이루어 지게 되므로, 일반적으로 가장 널리 사용되고 있다. 아래의 <그림 1>은 DIO를 이용한 구조를 도식화 한 것이다.

이 방식은 기계의 현재 상태를 실시간(Real Time)으로 처리할 수 있는 장점이 있지만, 반대로 접점의 상태를 정보화하는 일은 컴퓨터가 일임받음으로써, 컴퓨터에서는 신호레벨의 자료, On/Off정보를 매번 처리하여야 하는 부담감이 있다.



<그림 1> 직접접근방식 개념도

2.2. 간접접근방식(Indirect Accessing Method)

간접접근방식이란 PLC의 메모리(Memory)를 읽거나 쓰는 방식으로 직접방식과 하드웨어적으로는 틀린 점은 DIO(Digital Input/Output) 인터페이스를 할 수 없다는 점이다. 간접방식은 PLC의 Memory 영역을 접근하므로 PLC에서 컴퓨터연결에 제공되는 유니트를 사용하여야 한다는 제한조건이 있다.

이런 PLC의 컴퓨터 인터페이스 유니트는 대부분 표준 RS-232C, RS-422/485 Interface 를 제공하며, 자체내 프로토콜(Protocol)를 보유하고있어, 시스템설계자는 약속된 프로토콜에 맞춰 인터페이스를 하여야 한다.

Memory 접근 방식은 접점 혹은 신호레벨이 아닌, 약속된 Memory내에 정의된 내용(Content)을 기록하는 방식을 뜻한다. 여기서 정의된 내용이란 앞서 설명한 컴퓨터가 사용할 접점들의 정의를 정보화(Information)한 것으로 이는 다음 장에서 보다 상세히 설명키로 한다.

직접접근방식과 달리 접점에 대한 정보화를 하여야 하므로 PLC 프로그램의 많은 부분을 수정하거나 추가 하여야 한다. 다음의 그림은 간접접근방식의 개념

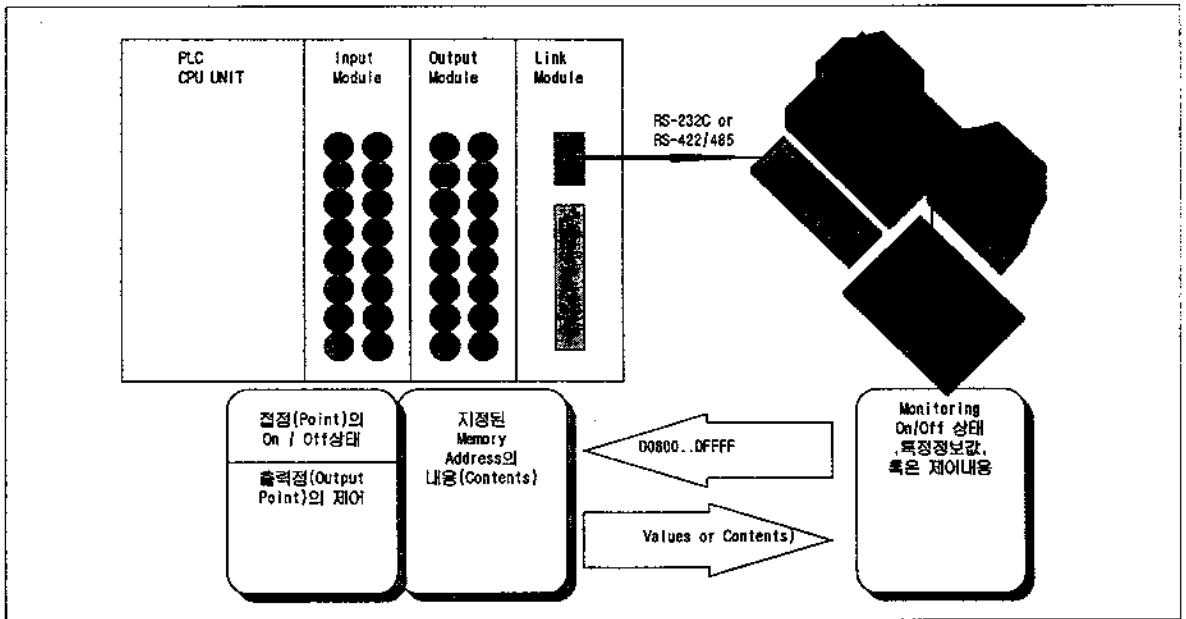
을 도식화 한 것이다.

약속된 Memory Address에 정의된 내용(Defined Content)에 대해서 설명코자 한다. 정의된 내용(Defined Content)란 결국 기계를 제어, 관리하고자 하는 데이터 혹은 정보에 따라 분류되어 사용하게된다.

본 논문에서는 이러한 정의된 내용(Defined Content)을 기계를 제어, 관리하고자 하는 요소(Element) 혹은 동작(Event)으로 분류하였고, 요소(Element)와 동작(Event)에 따라 각기 Memory에 mapping하는 방법에 대해 설명키로 한다.

2.2.1. 요소정의(Element Definition)에 의한 Memory Mapping.

요소(Element)란 크게 나누어 단순모니터링과 관리 지표계산을 위한 기초자료와 기계제어를 위한 명령으로 분류 할수있다. 관리, 제어자료들은 각각의 기계 및 현장 상황에 따라 달리 정의될 수 있으나, 공통적으로 불때 기계의 가동상태, 생산수, 정지수 및 주요 에러포인트(고장장소)등을 관리지표 기초자료로 정의 할수있고 비상정지, One Cycle Start/Stop, Operation Start/Stop 및 모델변경등의 제어명령 등을 정의 할수



〈그림 2〉 간접접근방식 개념도

있다.

한 요소에 대해 관리하고자하는 자료들이 다시 정의하여야 하는데, 이때에 상태값, 누적값의 의미를 부여하는 것이 바람직 하다. 예를들면 요소(Element)의 발생유무, 누적발생수, 누적발생시간등으로 각각의 관리자료들을 정의하여 각각에 대해 고유 Address를 지정토록한다.

요소관리자료들은 PLC 프로그램 내부에서 지정된 계산절차에 의해 해당 접점의 상태가 변화될때마다 지정된Memory Address에 계산된 결과를 자동적으로 기록하며, PC는 주기적으로 PLC Memory의 값을 읽어오거나 쓰는 형태가 된다. (<그림 3> 참고)

PC와의 통신을 하기 위해서는 요소(Element)의 PLC 정의 Address가 필요하며, 이는 다음과같은 간단한 공식에 의해 계산되어 질 수 있다. 한요소(One Element)의 길이는 다음과 같이 계산되어지며,

$$EDL = nm \times l \quad (1)$$

EDL : One Element Data Length (bytes)
 nm : the Number of defined management item
 l : defined bytes for PLC One Address.

전체 요소 데이터의 길이는

$$TEDL = n * EDL \quad (2)$$

EDL : Total Element Data Length(bytes)

n : the Number of element

EDL : One Element Data Length (bytes)

로 계산되어진다.

PC에서 원하는 요소의 내용을 읽을때 사용되는PLC Memory Address는 다음과 같이 계산된다.

$$\text{the Start Address of } i \text{ th element} = SA + (EDL \times (i-1)) \quad (3)$$

SA : the first Address of PLC Memory using PC Communication

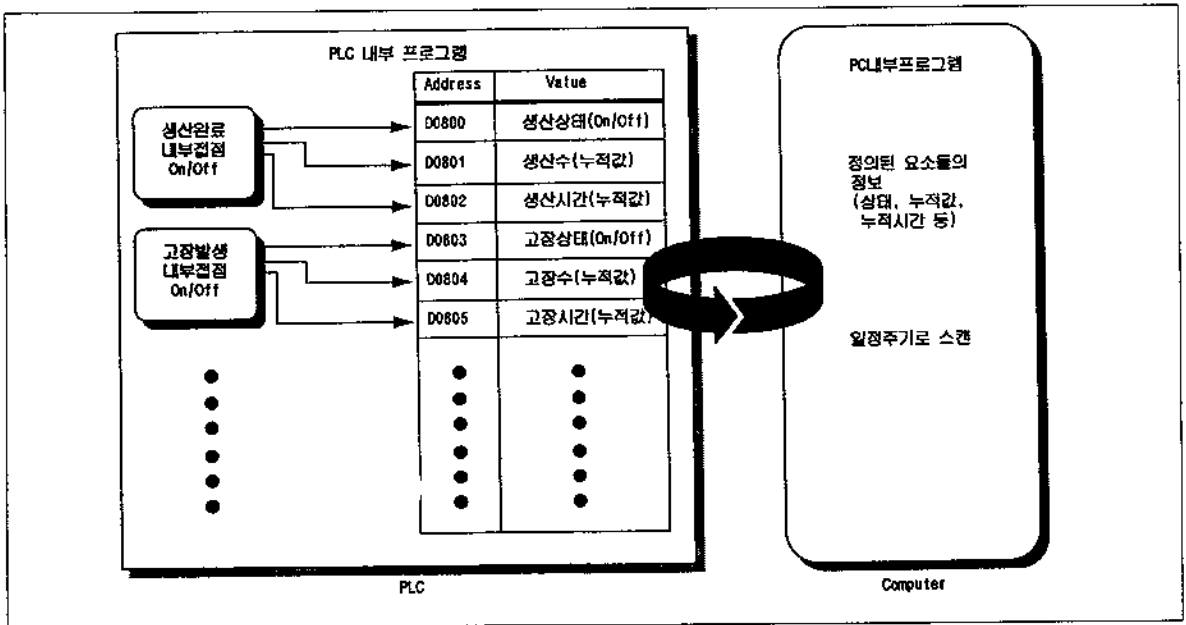
EDL : One Element Data Length (bytes)

i : the index of delaring order for PC Communication

(2),(3)항에서 구해지는 요소데이터들의 길이와 시작번지의 값을 이용하면 쉽게 PLC로 부터 원하는 자료들의 값을 구할 수 있다.

2.2.2. 동작정의(Event Definition)에 의한 Memory Mapping.

앞장에서 설명한 요소정의방법으로 지표관리 및 제



<그림 3> 요소방식의 처리 프로세스

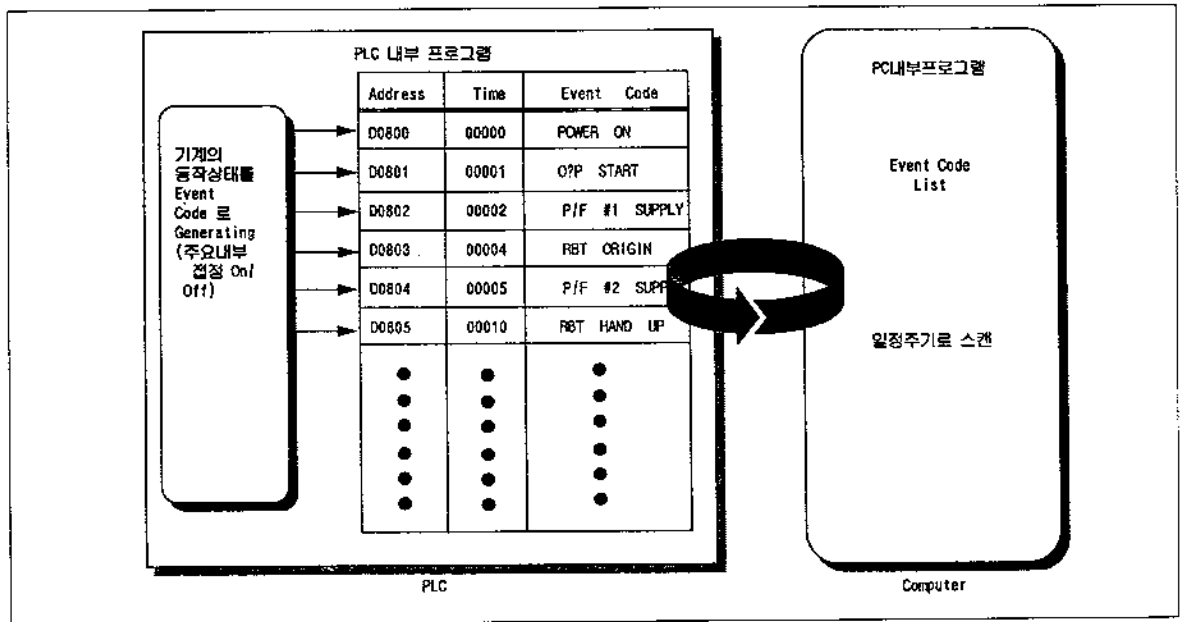
어가 가능하지만, 요소정의방법으로 기록한 PLC의 자료는 결론론적인 정보로서의 의미인 반면에 과정론적인 내용은 포함하지 않고 있다.

예를 들어, 설비의 특정부위의 고장회수, 고장시간 등의 결과가 어떤 작업의 수행 도중 발생하였는가에 대해서는 그 연관 관계를 규명 할수 없는 것이다. 결국 설비의 성능향상이나 애러감소를 위한 개선활동의 분석용으로는 시스템이 충분한 역할을 수행키 어렵다. 이러한 까닭에 고장원인의 규명 및 설비 동작상태를 감시키 위해 기계의 주요 동작을 각기 구분하여 이를 동작(Event)이라 정의하고, 이를 코드화하여 기계가 일련의 작업을 할때마다 해당되는 동작코드 (Event Code)가 PLC내부에서 발생, 이를 PLC Memory 에 순차적으로 기록하도록 하는 개념인 것이다. (<그림 4> 참고)

<표 1> 동작코드(Event Code)의 예

코드값	코드내용(Event)	코드값	코드내용(Event)
001	Operation Start	006	Normal Operation
002	Operation Stop	007	Normal Stop
003	Power On	008	Normal Waiting
004	Power Off	009	Error Stop
005	System Ready	010	Error Emergency

발생시각은 PLC가 Power On된 이후의 경과 시간을 초(second) 단위로 표시한 것으로 이를 컴퓨터에서 처리할때는 시각의 차가 발생된다. 발생시각을 컴퓨터 시간인 yymmddhhmmss형식으로 변환하기 위해서는 PLC가 Power On한 후 현재까지의 경과한 시간을 알아야 하며, 이 정보는 동작코드로 지정된 PLC Memory영역에 PLC가 지속적으로 기록하여야 한다.



<그림 4> 동작방식의 처리 프로세스

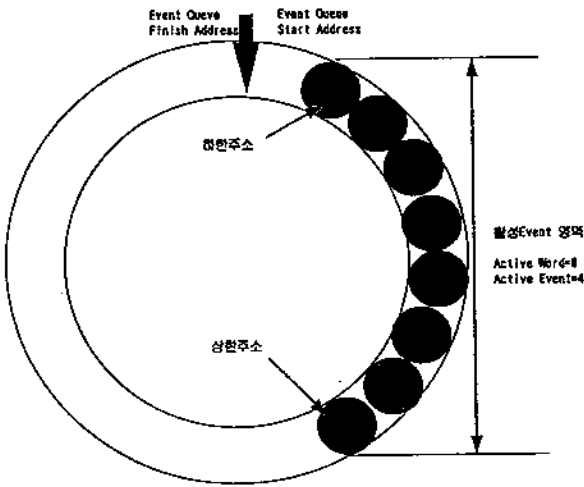
PLC Memory에는 동작코드 (Event Code)와 발생시각 (Event Time)을 한 쌍으로 기록하게 되며, 동작코드 (Event Code)는 설비설계자와 협의하여 설비특성에 맞도록 정의하여야 하며, 공통적으로 다음과 같은 내용의 코드는 포함하는 것이 바람직 하다.

이때 발생시각은 다음과 같은 절차로 구할 수 있다.

1. PC의 현재 System Date/Time을 읽는다.
2. PLC의 Power On 후의 결과시각을 읽는다.
3. 시간차 = (PLC Power On후 경과시간) - (Event 발생시각)

4. PC의 현 시각에서 3항에서 구한 시간차를 감인 시각을 계산한다.

한 동작을 기록하기 위해서는 앞서 설명한 동작코드 (Event Code)와 발생시각 (Event Time)만 기록하면 되나, 요소방법과 달리 발생하는 동작코드 (Event Code)의 양을 계산 할수 없으므로 발생하는 동작코드 (Event Code)에 대한 저장관리가 PLC Memory내에 필요하며, PLC내부에서 환형대기열(Circular Queue)로 관리토록 한다. 환형대기열의 구조는 다음과 같다.



(그림 5) 환형대기열(Cucular Queue)

환형대기열을 관리키 위해서는 다음과 같은 2개의 포인터(pointer)가 사용된다.

1. 하한주소포인터: 컴퓨터가 읽어가지 않은 가장 오래된 동작(Event)정보의 시작주소.
2. 상한주소포인터: PLC가 새로 발생할 동작(Event)을 기록할 선두주소.

하한주소와 상한주소 사이에 존재하는 동작자료 (Event Data)를 활성 동작자료라 하고, 그 영역을 활성동작자료영역(Active Event Data Region)이라 할때 발생 동작의 수 및 Memory크기는 다음과 같이 계산 되어진다.

활성 Event영역의 크기 = (상한주소) - (하한주소)

(5)

활성 Event Data 수 = (활성 Event영역의 크기 / Event Data Size)

(6)

또한,대기열에 아무런 동작자료(Event Data)가 없는 경우 및 대기열에 동작자료가 다 찬 경우의 상한, 하한주소의 관계는 다음과 같다.

상한주소 = 하한주소, if Event Data is empty. (7)

상한주소 = 하한주소 - (the size of One Event Data)

(8)

,if Circular Queue Buffer full

위의 공식 및 조건을 이용하여 PC와 PLC간의 프로그램 및 처리절차는 다음과 같다.

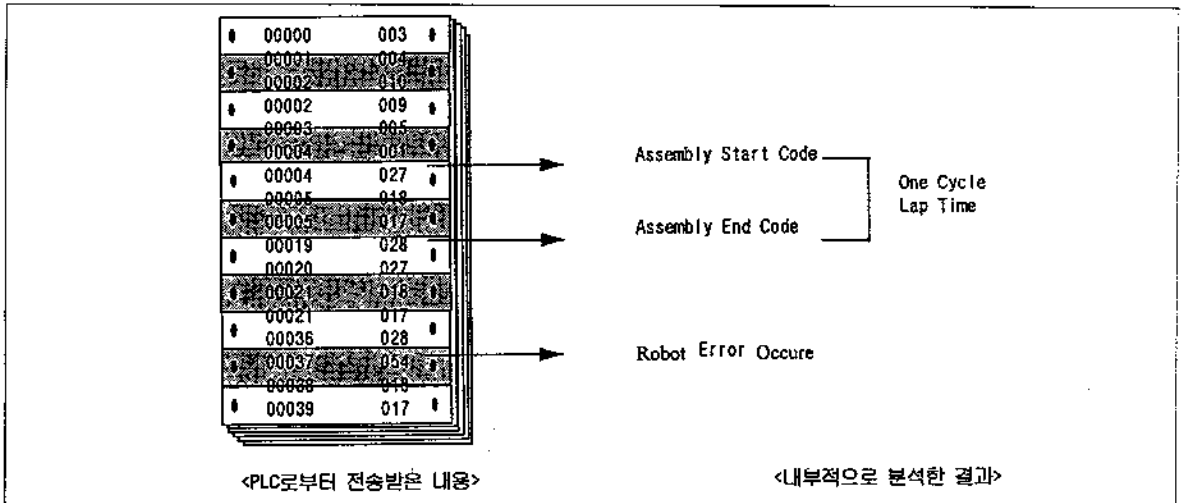
1. 상한,하한주소포인터는 지속적으로PC에서 모니터링하여야 하므로 PLC의 고정된 Memory Address 별로 저장하여야 한다.
2. PLC는 Power On시에 상한, 하한주소포인터를 Circular Queue의 시작번지로 설정한다.
3. 동작발생시 PLC는 동작자료(Event Data)를 구성하여 상한 포인터가 가리키는 Event Queue에 기록하고 상한포인터를 Event Data Size만큼 증가시킨다. 이때, 상한포인터주소에서 Event Queue의 끝번지가 한 동작자료의 크기보다 작은 경우에는 상한포인터 주소에 Event Queue시작주소를 기록토록 한다.
4. 컴퓨터는 1.항에서 설정된 상한, 하한포인터주소가 다르며, 즉Queue가 비어있지 않으면 상한, 하한 포인터주소 사이의 활성자료 영역의 데이터를 원하는 만큼 읽고, 읽지 않은 PLC의 가장 마지막 주소를 하한포인터주소로 설정한다.

이러한 방법으로 컴퓨터에서는 설비의 주요동작 과정을 얻을 수 있으며, 특정 동작의 수행 시간 (Lap Time)을 별도의 계산을 통해 얻을 수도 있다.

(그림 6)은 Queue Code방식에서 얻을수 있는 결과의 예이다.

3. 결론

본 연구는 시스템통합시 PC와 PLC간의 연결부분에 대한 기술적 구현을 PC 위주가 아닌 PLC위주로 기술 하였다. 앞서 서술한 직접접근방식, 간접접근방식 그



〈그림 6〉 출력결과 활용 예

리고 간접접근방식에서의 요소(Element), 동작(Event) Mapping에 대한 방법론은 결국 시스템설계자가 어떠한 관점으로 각 기계들을 관리, 제어하겠느냐에 따라 구분되어 사용될 수 있으나, 혼합된 형태로도 사용될 수 있다.

이 방법은 기본적으로 PLC의 점점에 점점에 대한 정의(Definition) 혹은 기계동작의 코드화(Event Code)라는 작업이 선행되어야 하며, 특히 동작코드(Event Code)의에 대한 부분은 기계의 동작 상태를 완전히 파악하여야만 가능하고 또한 많은 인력과 시간이 필요하다.

일례로, 동작코드(Event Code)의 경우 로봇셀은 126가지, 부품공급장치류는 133가지의, 그리고 검사기는 80가지로 분류, 정의하는데 약 2개월에 걸쳐 4명의 인력이 투입되었다.

이에 반해 아울러 요소(Element)의 정의는 기계설계표준화차원에서 진행하였으므로 별도의 인력 투입은 발생치 않았으며 로봇은 6가지, 부품공급장치는 40가지로 정의되어 PLC Memory 특정부분에 지정되어 사용하고 있다.

이러한 방법을 몇몇 시스템에 적용해본 결과 설비의 정지 원인 및 과정등을 정확히 제시 함으로써, MTBF 및 설비가동율을 향상시킬 수 있었다.

동작정의에 의한 방식이 실질적인 효과를 얻을 수

는 있으나, 앞서 기술한 대로 많은 시간과 인력이 소요되므로 주요 설비개발시만 이용되며, 대부분은 동작정의를 간이 형태로 수행하고 요소(Element)를 혼합한 방식을 사용한다.

동작코드(Event Code)의 주요 코드가 자동적으로 요소(Element)화되는 기술을 해결하여야 하는 것이 앞으로 지속적으로 수행하여야 하는 대상이다.

【참고문헌】

- [1] 신승헌, 작업관리, 형설출판사, 1988, pp.79-95.
- [2] John A. White, *PRODUCTION Handbook*, 1987, pp. 88-115.
- [3] Miklell P.Groover, *INDUSTRIAL ROBOTS*, 1986, pp. 88-115.
- [4] X-1 Deck 생산관리시스템설계서, 삼성전자, 1991, pp. 99-111.
- [5] MGT Final Assy 조립공정 설비관리시스템, 삼성전자, 1992, pp 70-85.
- [6] Kang G. Shin, *Real-Time Communication in a Computer-Controller Workcell*, IEEE Trans. Robotics and Automation, vol. 7, no. 1. pp 105-109, Feb. 1991.
- [7] Brian R. G. Shin, THE DESIGN OF EXPERT

SYSTEMS FOR PLANNING FLEXIBLE
MANUFACTURING, Japan-USA Symposium on
Flexible Automation, vol 1 , pp 761-768, 1991.



이현준

1961년 1월 26일생

1985년 삼성전자 입사

1991년 VCR Robot자동라인관리시스
템개발

1993년 MGT Final Assy조립라인제어
관리시스템개발, 자동삼입기군
관리시스템개발.

1994년 FANSY v1.0 개발

현재 자기진단및 관리제어시스템개발
중(G7과제)

김영태

1957년 12월 29일생

1992년 Georgia Institute of Technol-
ogy, 미국 전기공학박사

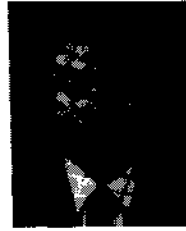
1988~92년 Fault Tolerant control
system 개발

1992년 삼성전자 입사

1993년 C/TV 이형삼입라인생산관리
시스템개발

1994년 FANSY v1.0 개발

현재 자기진단 및 관리제어시스템 책
임자(G7 과제)



김성권

1949년 8월 1일생

1988 Minnesota Univ., 미국 ROBOT
공학박사

1991년 수평다관절로봇개발(장영실상)

1992년 6축다관절로봇(장영실상), 범
용검사조정제어기개발(장영
실상)

1993년 다산기술상대상수상(한국경제
신문주관)