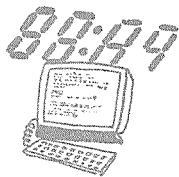




吳 吉 祿

韓國電子通信研究所  
컴퓨터연구부장 / 工博

# 공정제어용 프로그램식 제어기에 관한 고찰 (I)



註: 본고는 전자통신연구소 전자기기 개발부에서 84년  
도 파기처 특정 연구과제로 수행한 결과에서 나온  
것으로 전자기기 개발부의 동의를 받아 2회에 걸  
쳐 전제합니다.

일반적으로 사용되고 있는 제어를 대별하면 feedback control과 sequence control의 두가지가 있는데 sequence 제어는 제어 대상의 움직임을 미리 정해놓고, 이에 따라 일반적으로 명령을 내려나가는 순서 제어와 조건 제어로 분류되어 산업계의 현장의 요구를 실현시켜 왔다. 이러한 요구에 의해 등장한 sequence는 sequence 제어를 위해 사용된 relay회로에 공정 변화에 따른 유용성을 향상시켰으며 초기에는 rotary cam method에 의한 것과 pin board method에 의한 것으로 구현되었으며 이는 설계상의 문제, 시운전 조정상의 문제, 설치 보수면의 문제, 기능상의 문제 등이 대두되어 sequence 제어를 software로 처리하여 고도의 기능을 추가하는 필요성을 고려하게 되었다. 즉 relay회로의 logic을 조립하고 있던 “배선”을 program으로 대체하여 memory가 필요하게 되었다.

또한  $\mu P$ ,  $\mu C$ 의 발달과 더불어 수행 속도가 빨라지고, 수행기능이 다양화되어 이른바 stored program method에 의한 sequencer로 발전 Programmable Controller (프로그램식 제어기, 이하 PC로 표기)로서 명명하게 되었다.

이 PC는 1978년 NEMA standard 1-28에 의해 정식으로 “PC”라는 이름을 채택하게 되었으며, 다음과 같이 정의되었다.

“디지털 또는 아날로그 입·출력 장치를 통하여 논리 연산, 순서 제어, timer/counter, 산술 연산 등을 특정한 명령어로 memory에 program하고 기계나 process를 제어하는 digital 동작의 전자장치 또한 이것과 같은 목적으로 사용되는 digital 계산기도 PC의 범위내에 간주한다. 단, drum형의 기계적인 sequencer는 제외한다.”

이러한 PC는 General Motor사의 자동차 조

립 line의 relay반 대신에 적용을 위해 여러 규격을 제시하여 Gould사에서 Micro-84PC를 개발한 것이 효시였으며, 차후 analog 신호 처리, PID 제어 기능뿐만 아니라 다수의 PC와 main computer와의 interface를 통한 큰 규모의 flant controller까지 확장되었으며, 반도체의 급속한 발전으로 소형화, 고기능화, 다기능화, 저가격화 방향으로 계속 발전해 가고 있다.

## I. PC의 구성과 동작

PC는 구체적으로 6개의 module로 나눌 수 있으며 어떤 module은 option으로 제공되기도 한다.

- 1) User의 program이 용이하도록 programming장치(Field programming device와 CRT programing device로 나뉨)
- 2) 센서 및 제어 대상과의 interface를 위한 입·출력 module.
- 3) CPU, Memory 및 Timer/Counter 등의 중심 module.
- 4) 상용 전원을 공급받아 PC내부에서 사용하는 전원으로 변환해 주는 전원부 및 이상시의 정전 복구 기능을 위한 power module.
- 5) PC주변의 주변 장치module 및 network module.
- 6) PC에 내장되어 있는 monitor(load/run program) 및 Control software module.

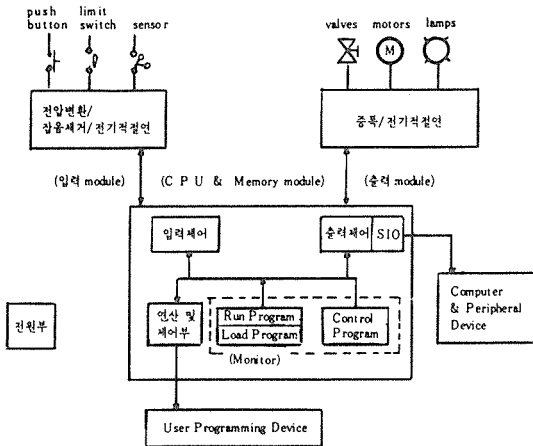


그림 1. PC의 Block Diagram

이 module들을 포함한 구체적인 PC의 모형은 위의 그림 1과 같으며 이 module들의 실제 구현은 일반적으로 board type과 block type으로 나누어진다.

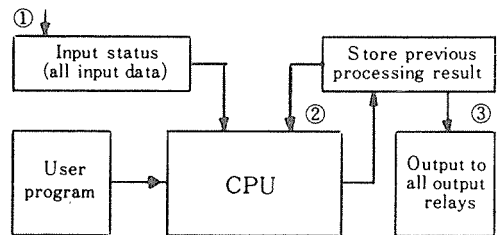
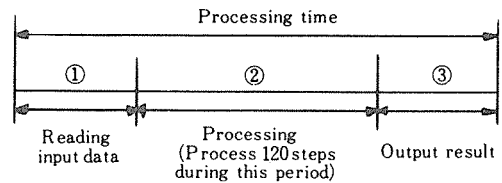
## 1. PC의 Hardware

### 가. The Central Processing Unit

이 부분은 processor와 system memory로 나누어지며  $\mu P$ 의 발달로 communication 기능, multi-function 기능 등의 높은 level의 수행이 가능한 system으로 되었으며 timers/counters 등의 기능을 software적으로 처리하고 있다. 이들은 programming device 혹은 주변 장치와의 communications, field들간의 monitoring, system diagnostics, process control, multi-processing 등의 기능을 수행한다. PC에서의 CPU 기능은 다음과 같이 3부분으로 수행된다.

#### - CPU의 scan

공정상상의 한 cycle(loop) time으로서 입력부로부터 처리 data를 읽고, 프로그램을 수행하고 그리고 출력부에 내주는 일련의 과정으로 cycle method step 방식이다. 이의 수행을 그림



#### I/O Update

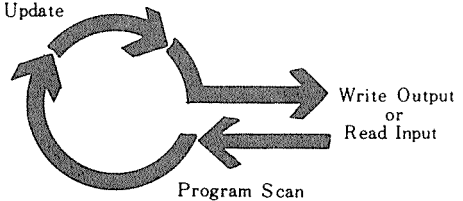


그림 2. CPU Process 과정과 Scan time

2와 같이 나타낼 수 있다.

이 scan time은 다음의 식으로 표현할 수 있다.

$$\text{scan time} = (\text{입력기입 시간} \times \text{입력점수}) + (\text{연산 속도} \times \text{프로그램 step수}) + (\text{출력 전송 시간} \times \text{출력점수}) + \text{고장 진단 시간}$$

이 식에 의해 scan time을 계산할 수 있으나 실제로 예기치 못한 지연을 20%정도로 고려하여 통상 PC의 동작 속도는 거의 내장된  $\mu\text{P}$ 의 동작 속도에 의해서 결정되므로 10~100msec 정도이다. 최근에 scan time의 고속화가 많이 연구되고 있다.

#### - Subsystem communication

CPU와 I/O subsystem과의 연결에서는 거리와 전송 속도가 고려되며 실질적으로는 communication media의 문제점도 고려된다. CPU에서의 필요 사항은 error checking technique이며 일반적인 format형태로서 data, bits, start/stop bits, error detection codes로서 이루어진다. Error checking의 방법은 다음의 종류로 이루어진다.

- Parity check (Vertical Redundance Check : VRC)

- every parity, odd parity

- Check sum (Block Check Character : BCC)

- CRC (Cycle Redundancy Check)

- LRC (Longitudinal Redundancy Check)

- CPU diagnostics & reset

이상적인 diagnostics로서는 memory, processor, battery power supply OK를 자체 진단하여, software적으로는 power 초기화, 주변 chip 초기화, parameter 초기화로 이루어진다. 또한 시스템 이상시 보호, 감시 회로에 의해 이상 발생시 CPU에 그 상태를 알려거나 강제로 CPU를 reset시킨다. 그 예로는 battery 또는 condensor 이상시 CPU에 경보, CPU 및 전원 전압이상시 CPU강제 reset 등으로 오동작에 의한 사고를 사전에 예방할 수 있다.

#### 나. Memory Unit

PC에서의 일반적인 memory system은 (1) Executive (2) Scratch Pad (3) Application memory (4) Data Table로 분류되며 executive program은 시스템 자체를 수행하는 system monitor와 주변 장치와의 communication program이 들어있다. Scratch pad는 CPU의 monitor 수행이나 control program수행중 생기는 일시적인 data를 보관하며 application memory는 user에 의해서 program instruction이 저장되는 영역과 control program이 들어 있는 영역이다. Data table은 control program과 관련된 data를 저장하며 즉 timer/counter 값, control program의 상수, 변수값 등이 저장된다. 이를 memory map으로 나누면 system memory와 application memory로 나눌 수 있는데 그림 3과 같이 표현된다.

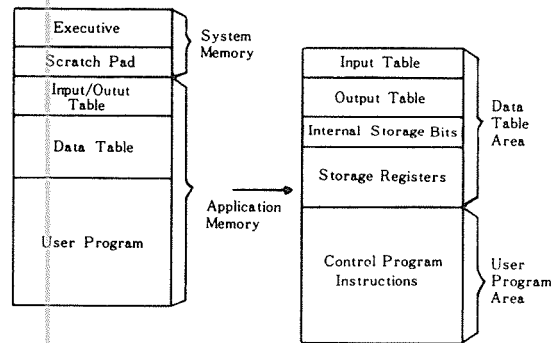


그림 3. 일반적인 PC의 Memory Map

#### 다. The Input/Output Unit

입·출력부는 주로 현장에서의 공정 test, sensor 등의 입력들과 valve, motor 등의 actuators로서의 출력을 의미한다. 실질적으로 이러한 입·출력의 연결로서 PC를 “현장용 computer”라고도 부르고 있으며, 이들 입·출력은 크게 4분류된다.

- (1) Analog Input/Output
- (2) Digital Input/Output
- (3) Special Input/Output
- (4) Remote Input/Output

이의 입·출력부의 각각의 module에서는 특히 입·출력 점수를 구하는 것이 중요하며 실질

적으로 analog 점수와 digital 점수를 최대 16점, 128점들로서 구성되고 있다. 또한 이 입·출력 부에서는 각각의 점수, 입출력 type 및 rating resolution 등이 고려되고 있으며, 입·출력부의 signal conditioning 회로 및 입·출력 상태를 점

점하는 표시장치 및 입·출력 점수의 증설을 고려하여 여유점수를 두고 있다. 일반적인 PC용으로서의 analog 입·출력부와 digital 입·출력부를 의미하며 이의 device들과 표준 rating은 표 1, 2와 같다.

표 1. Analog Input/Output Devices & Standard Rating

Devices		Standard Rating	
Inputs	Outputs	Input	Output
Temperature transducer	Analog valves & actuators	4-20mA 0-+1volts DC	4-20mA 10-50mA
Pressure transducer	Chart Recorders	0-+5volts DC	0-+5volts DC
Load cell transducer	Electric Motor Devices	0-+10volts DC	0-+10volts DC
Humidity transducer	Analog Meters	1-+5volts DC	±2.5volts DC
Flow transducer		±5volts DC	±5volts DC
Potentiometers		±10volts DC	±10volts DC

표 2. Discrete Input/Output Devices & Standard Rating

Devices		Standard Rating	
Inputs	Outputs	Input	Output
Selector switches	Alarms	24volts AC/DC	12-48volts AC
Push buttons	Control Relays	48volts AC/DC	120volts AC
Photoelectric Eyes	Fans	120volts AC/DC	230volts AC
Limit switches	Lights	230volts AC/DC	12-48volts DC
Circuit Breakers	Horns	TTL levels	120volts DC
Proximity switches	Valves	Non-Voltage	Contact (Relay)
Motor starter Contacts	Motor starter	Isolated Input	Isolated output
Relay Contact	Solenoid		TTL level

이외 PC의 주변장치로서 programming device(program loader, programmer 등) power fail 등의 power부, network을 위한 port들로서 구성되어 있다. 이들은 interfacing의 경우에 외부적으로 standard에 맞추어 제공하는 단자로서 EIA RS232C, RS422, 4~20mA current loop가 사용되도록 되어 있다.

## 2. PC의 software

PC에서의 system software라고 할 때 일반적인 운영체제 또는 monitor program에서 특수 용도의 운영 체제를 의미하며 즉 PC에서 적합한 제어 분야의 software를 의미하며 micro-

51, iRMX, VRTX 등이 있으며 이러한 운영 체제를 ROM에 내장시켜  $\mu P$ 와 함께 제공하고 있다. 이로써 자체적으로 개발할 필요없이 응용 program의 개발에 전념하게 할 수 있도록 해줌으로서 ROM에 내장된 운영 체제를 사용함으로써 얻을 수 있는 software의 신뢰성 제고 및 system 실행 속도를 단축시킬 수 있는 장점을 제공받기도 한다. 또한 갖가지의 control program도 option으로 제공하고 있으며 PC용 프로그램으로 APS (Automatic Programing System for Sequence Control) 등도 고려되어 편리한 user-friendly system으로 개발되고 있다.

가. PC의 일반적인 flow chart

Software의 전체적인 구조는 PC를 초기화한

후 monitor program에서 user program을 user-program 영역에 입력하는 일과 user가 원하는 내용을 display하도록 하는 일 및 새로운 user-program 입력을 용이하게 하는 일을 해주며 user가 원할 때 입력부의 신호를 받아들임으로써 execution program (control program)을 수행하도록 해준다.

또한 부수적으로 갑작스런 정전에 대해 system을 정전전의 상태로 복구시키는데 필요한 정전 복구 프로그램이 있다. 이의 일반적인 프로그램 내용을 3부분으로 나누고 있다.

- Initialization Program

사용되는 여러 주변 chip의 초기화, power initialization, 또한 sensor 등의 비선형 보상을 위한 선형화 program 초기화, control program에 필요한 parameter 초기화 등의 프로그램을 수행한다. 또한 정전 복구 프로그램 등이 제공된다.

- Monitor Program

Programming device로부터 들어온 key의 분류 및 해당 operation 상태를 수행하며, system의 여러 상태 등을 점검, 보관한다. 이는 loading program과 run program으로 나뉜다.

- Execution Program (Control Program)

User program을 1 block씩 fetch하며 해당하는 algorithm에 따라 반복하게 되며, 일반적으로 PID algorithm에 의한 program이 option으로 제공되고 있다. 이는 필요한 상수 P값, I값, D값을 user의 규격에 따라 제공한다.

나. PC의 program 언어

PC에서 주로 사용되는 프로그램 언어는 다음과 같이 4 가지 형태로 나눌 수 있다.

- 계전기 래더 도형 (Relay ladder diagram) } 기본 언어
- Boolean mnemonics } 기본 언어
- 기능 블록 (Function block) 언어 } 고급 언어
- 컴퓨터식 (English statement) 언어 } 고급 언어

실질적으로는 위 4 가지 언어를 복합적으로 사용하고 있으며, 기본 언어는 계전기식 순차 제어에 필요한 순차 제어, 계시 및 계수 (timing and counting) 제어와 논리회로 구성 등을 pro-

gram하는 데 주로 쓰이며, 고급언어는 analog 제어, data 처리 및 보고 등 기본언어로는 수행하기 힘든 복잡한 기능을 program하는 데 사용되고 있다.

또한 PC에 사용되는 instruction set는 다음 6 group으로 나뉜다.

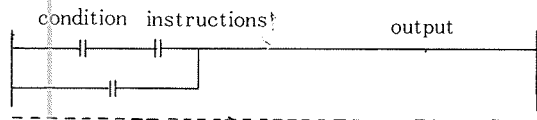
- Relay logic
- Arithmetic
- Timing and counting
- Data manipulation
- Data transfer
- Flow of control

각각의 program 언어 특징은 다음과 같다.

(1) Relay ladder diagram

계전기의 coil이나 접점의 상태를 알기 쉬운 기호로 표기하는 프로그램 언어이며 비교적 초기 PC는 이 언어를 주로 사용하였으며 그후  $\mu P$ 의 도입으로 data 처리, 연산 및 프로그램 수행을 제어하는 언어구사가 가능해졌다. 이의 명령어로서 relay ladder logic, timing & counting, 연산, data 처리, program 제어 등의 기능을 가진다.

일반적인 format은 그림 4 와 같다.



A Continuous path is required for logic continuity

그림 4. Relay ladder diagram의 format

(2) Boolean Mnemonics 언어

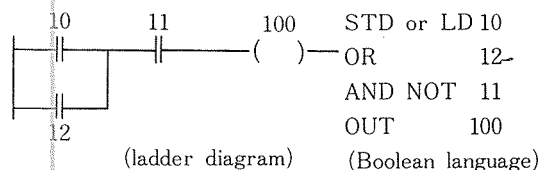


그림 5. PC의 ladder diagram과 Boolean 언어의 비교

기본적인 boolean연산자인 AND, OR, NOT 이 그 주요구성 요소이며 그림 5와 같이 몇개의 mnemonics언어가 ladder 도형을 나타내기 위해 정의되어 있으며 기본지시 명령어는 ladder도형의 경우와 유사하다.

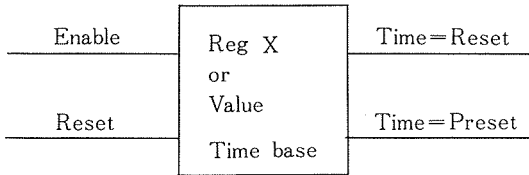


그림 6. Function Block언어의 예

### (3) Function Block 언어

기능 블록 언어는 고급 언어의 일종으로서 그림 6과 같이 특정 기능을 블록으로 표시하고 입출력 제어 조건을 표기하는 방식이다. 이는 기본적인 기능 외에 data변환, parameter 설정, block data이동, ASCII변환 등의 기능이 추가되며, 특수한 경우 PID block drum timer 의 등가인 순차 제어 block기능도 가진다.

### (4) 컴퓨터식(English Statement) 언어

이는 일반 컴퓨터가 사용하는 BASIC, FORTRAN과 같은 언어를 말하며, PC에 적용된 경우 조작자가 익히기에 더 간편하게 되어있다. 이 언어를 사용하면 PC의 입·출력이 수치 data를 직접 register에 옮길 수 있고, 주변기기와 정보 교환이 용이할 뿐 아니라 주로 대형(입·출력 점수 1000이상)의 PC언어로 이용되고 있다.

이상의 4가지 언어를 살펴보면 각각의 장단점이 있으나 이보다는 PC본체의 선택을 고려하여, 또한 제어 대상 공정에 대한 충분한 지식과 PC의 비용을 검토한 후 언어를 결정할 필요가 있다. 이에 따른 고려 사항으로 사용의 편리성, 언어의 기본 특성, 해결해야 할 문제점의 분석, program 수행 속도 등을 검토하여야 한다.

## 3. PC의 Network 기능

공정 제어 시스템의 구현으로 분산 제어 등

범위가 넓어짐으로써 PC에도 network개념으로 data highway기능이 부여되고 있다. PC에서의 data-highway의 일반적인 응용으로서 data acquisition과 분산 제어를 가지며, 다음의 기능을 가져야 한다.

- PC들간, 그리고 다른 host들간의 Communication
- 다른 PC로부터 programmer나 host computer로 upload
- Programmer나 host computer로부터의 download
- PC의 I/O 및 register값들의 read/write
- PC의 status의 monitoring 그리고 PC 동작의 제어

위의 기능을 가진 data highway를 설계할 때 throughput, implementation cost, reliability 등을 고려하여 star, common bus, ring type으로 구성하게 된다. 또한 access 방법으로서 polling, collision detection, token passing 방법을 사용하며 전송 매체로서 twisted-pair, coaxial-cable, optical fibers 등을 이용하고 있다.

표 3은 대표적인 PC업체로부터 data-highway 기능을 비교하여 나타낸 것이다.

이러한 기능을 가진 network의 설계시 고려하여야 할 점은 다음과 같다.

#### (1) device의 maximum 수

설계시 network상에 얼마나 많은 node 점을 요구하게 되며 각각의 node에 대한 device의 type을 조사한다. 이러한 device는 PC, vendor-supplied programmer, host computer, intelligent terminal이며 이들의 수를 조사한다. 또한 device 각 type에 따라서 장래 확장을 위한 여유점의 수를 가져야 한다.

#### (2) Maximum length

Data highway의 거리는 두 분류로 main cable의 최대 거리와 device와 main cable 간의 각 drop의 최대 거리를 설정하여야 한다. 일반적으로 PC에서는 drop거리는 30~190feet이며 또한 중요한 점은 최대 거리에 따른 cable의 적당한 type이 요구되며 cable상의 설계시의 ground점 설정 등 감소에 유의하여야 한다.

표 3. 대표적인 PC의 Data High Way 특성 비교표

Manufacturer	Highway Name	Max. Nodes	Max. Length (ft.) Max. Baud Rate	Baud Rate (max)	Access Method	Comments
Allen-Bradley	Data Highway	64	10,000	56K	Token	
GTE Sylvania	Control Net	254	5,000	1 M	Token	X.25 Compatible Gateway
General Electric	GEnet	999	15,000	5 M	Collision Detection	IEEE 802.3 Compatible
Gould-Modicon	Modbus	247	15,000	19.2K	Master-Slave	Uses 2 Twisted Pairs, Shielded
Gould-Modicon	Modway	250	15,000	1.544M	Token	
Industrial Solid State Controls	Copnet	254	32,000	115.2K	Master-Slave	Includes interfaces to Allen-Bradley, Gould-Modicon and Texas instruments PC's. HDLC Protocol.
Measurex	Data-Freeway	63	10,000	1 M	Collision Detection	
Reliance Electric	R-Net	255	12,000	800K	Token	ASCII/X3.28/HDLC Gateway Uses HDLC Framing
Square D	SY/Net	200	2,000	500K	Timed Token	Worst case access=500 msec with 50 PC's.
Texas Instruments	TIWAYI	254	10,000	115.2K	Master-Slave	HDLC Protocol
Texas Instruments	TIWAYII	2*	32,000	5 M	Token	Broadband IEEE 802.4 compatible
Westinghouse	WDPF	254	18,000	2 M	Token	100 msec Fixed Access Time. 10,000 Points/Sec Throughput
Westinghouse	Westnet	50	10,000	1 M	Master-Slave	Gateway Interface to Westnet

(3) Response Time

Response time이란 한 device(한 node)의 출력에서 다른 device(다른 node)의 입력까지의 시간을 의미하며 이는 다음 식으로 표현된다.

$$RT = IT + (2 \times ST1) + PT1 + AT + TT + PT2 + (2 \times ST2) + OT$$

여기서 IT : input delay time

ST1 : scan time for sending node

ST2 : scan time for receiving node

PT1 : processing time for sending node

PT2 : processing time for receiving node

AT : access time

TT : transmission time

OS : output delay time을 의미한다.

4. PC의 동작

가. Programming the controller

어떤 공정 제어 과정을 program할 때에 제어해야 할 대상의 특성을 이해하여야 한다. 즉 제어 목적, 운전 방법, 동작순서, 각종 전기적인 조건을 알아야 한다. 또한 제어 장치에 대한 sequence를 작성키 위한 공정 상호간의 관계, 제어loop, 제어parameter를 결정하여야 한다.

일반적인 공정의 단계를 나타내보면

(1) 초기 설계

- 공정 제어 대상의 control loop의 분류 및 과정정리
- 사용되고 있는 입출력 접수 파악
- 공정의 제어 parameter조사

(2) 제어 공정 단계

- 각 공정의 state diagram 및 control level

형성

- PC가 어떤 동작의 순서로 움직여야 할지 제어 대상의 순서를 결정하는 것으로 command level, algorithm control level, dynamic control level 등으로 나눈다.

- 각 공정의 operating mode 분류

- 각 공정상의 loop를 결정지으며, 새롭게 결정할 수 있는 loop도 결정하고 필요한 입·출력을 loop상에 기록한다.

- 각 operating mode의 state transition diagram 작성

- Operating mode에 따라서 most common state, normal state, purposeless state, system-ignored state 등의 transition diagram을 분류한다.

- Mode processing table 작성

- 공정의 입·출력점을 table로 작성하여 scheduling을 정한다.

나. PC의 control algorithm에 의한 예

일반적으로 PC의 module에 PID 제어를 option으로 제공하는데 이는 proportional-integral-derivative interface 혹은 three mode closed-loop feedback control을 뜻한다. 이는 제어 대

상의 전달함수, set point (SP), process variable (PV) 및 error를 구하여 P값, I값, D값이라 불리는 3 가지 parameter를 적당히 정함으로써 설정 온도를 최적 제어하는 기법으로 일반적인 PID 공정을 그림 7에 나타내었다.

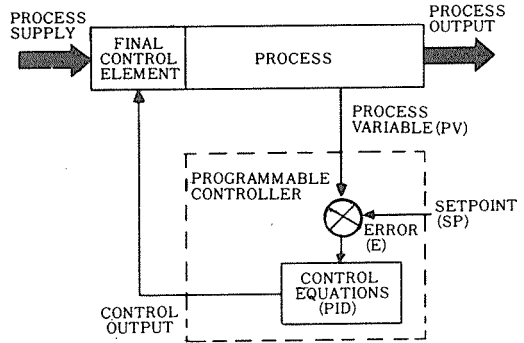


그림 7. PID algorithm의 diagram

이때 출력  $V_{out}$ 은 다음 식처럼 표시된다.

$$V_{out} = K_p E + K_i E t + K_d \frac{dE}{dt}$$

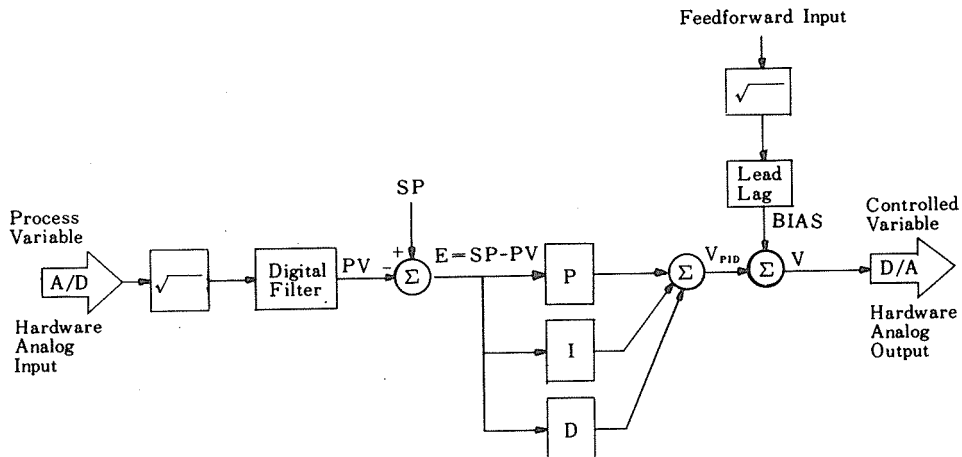
여기서  $K_p$  : (proportional gain)

$K_i = K_p / T_i$  (integral gain,  $T_i$  : reset time)

$K_d = K_p T_d$  (derivative gain,  $T_d$  : rate time)

$E = PV - SP$  (error)를 의미한다.

실질적으로 PID 제어에 따른 공정 block diagram과 이에 따른 프로그램 예를 그림 8에 나타내었다. (본 program은 Gould의 Modicon884를 이용한 것임)





0001-OFF	Time Base	TB
0002-OFF	Sign of Error DN-SP>PV	SOFER
0003-ON	Old Sign of Error	OSOFER
0004-OFF	Error Getting Bigger	ERRBIG
0005-OFF	Add Differential Term	ADDT

4001-0100	DECIMAL	Time Base	TBR
4002-0084	DECIMAL	Actual Time	ATR
4003-0051	DECIMAL	Set Point	SP
4004-0050	DECIMAL	Process Variable(Feed Back)	PV
4005-0001	DECIMAL	Err	ERR
4005-0001	DECIMAL	Oerr Old Err	OERR
4007-5000	DECIMAL	Gain for Integral	GFI
4008-0000	DECIMAL	Work Space	WS1
4009-0000	DECIMAL	Work Space	WS2
4010-3368	DECIMAL	Integral MS	IMS
4011-3100	DECIMAL	Integral LS	ILS
4012-1000	DECIMAL	Gain Portional	
4013-3368	DECIMAL	Output	OUTPUT
4014-5000	DECIMAL	Range for Portional	RFP
4015-0000	DECIMAL	Differential Change	DIFF
4016-0000	DECIMAL	Gain Differential	
4017-0000	DECIMAL	Range Differential	
4018-0000	DECIMAL		

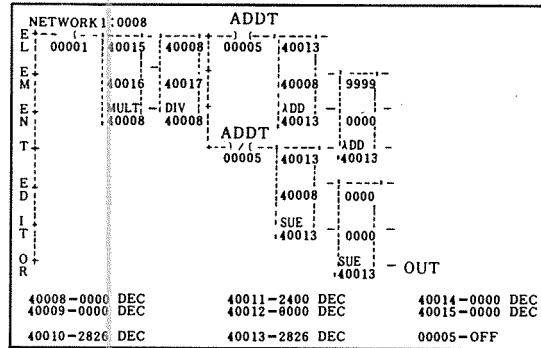


그림 8. Modicon884에 적용위한 PID 기능 모델

다. PC의 제어용 언어와 공정 적용 분류

PC의 많은 공정의 실제 적용에 있어 프로그램을 사용할 다양한 제어용 언어가 요구된다. 특히 많은 공정에서 실시간 처리와 multitasking이 요구되는데 실제 공정을 분류하면 다음과 같다.

- loop control
- bath control
- sequence control
- data acquisition and monitoring
- human interfacing
- supervisory control and plant management

표 4. 제어용 언어와 PC기능 조사표

	small PC	large PC	hybrid PC	μP-based process control- lers	Computer -based process control systems
relay ladder	○	○	○	△	×
boolean	×	△	×	×	×
flow charting	×	△	×	×	×
softwiring	×	△	○	○	○
mask editing	×	×	×	×	△
high-level language	×	×	×	△	○

(○: 적용 △: 적용가능 ×: 적용불가능)

연속 제어면에서 어떤 주어진 시간에 대해 특정한 값에 따라 공정 조건을 유지해감에 요구되며 이에 대해 제어 변수에 따라 PID loop 제어, cascade 제어, 비율 제어, feedforward 제어, 비선형 제어, 적응 제어 등의 복잡한 algorithm이 도입된다. 이에 따라 PC용 언어를 사용언어/제어 방식의 분류에 따라 symbolic language, Specific language, high-level language로 나눌 수 있다. Symbolic language는 relay ladder logic으로서 대부분의 PC에 도입되어 빠른 logic 제어와 특별한 기계 제어에 가장 적합하며, specific language는 boolean logic과 flow chart language로서 VDU-based programming 장치를 활용하여 ladder format의 CRT상에서 flow 상황과 debugging에 유리

표 5. 제어용언어와 공정적용조사표

	Seque- nce	loop	batch	arith- metic	data acqui- sition	super- visory control	human inter- facing
relay ladder	○	×	△	△	×	×	×
boolean	△	×	×	×	×	×	×
flow charting	△	×	×	×	×	×	×
softwiring	×	○	△	△	○	×	×
mask editing	×	×	×	×	×	×	○
high-level language	○	△	○	○	○	○	○

(○: 적용 △: 적용가능 ×: 적용불가능)

하다. 이는 I/O data base 구조, 간단한 algorithm의 구현, loop control의 구현에 유용하다. High-level language는 computer-based control system으로서 BASIC, FORTRAN, C 등이 사용되며 data acquisition, manipulation of pro-

cess I/O, 복잡한 supervisory control이 가능하다. 이에 따라서 표4,5에 PC의 제어용 언어와 공정 적용 분류를 나타내었다.

(다음 號에 계속)

### P. 59에서 계속

을 들 수가 있다. 현재의 Mountor, Bonding Machine 등의 기계가 선을 보이면서 半導体の 어셈블리工場이 勞働人力 立場에서 市場立地로 변해갔다. 이때문에 日本메이커도 73년까지는 東南亞에 나왔으나 그후에는 市場이 있는 곳, 또는 關稅障壁을 해쳐나가기 위해 美国이나 歐洲 各國으로 진출해 갔다. 요컨대 立地 條件이 달라진 것이다. 当社도 73년에 進出했을 때는 100% 日本으로 가지고 가는 형태를 취했다. 그러나 지금은 일본에 가지고 가는 것은 생산의 15% 정도이며 이보다도 東南亞의 市場에 對應하는 生産拠點으로서의 말레이시아 공장이라는 형태로 使命이 바뀌었다.

日本과 美国의 半導体공장 형태는 같은 것인가 그렇지 않으면 相違點이 있는 것인가. 이 점에 대해 NEC나 当社도 메모리를 만들고 있지 않으며 東南亞의 民生機器用 主体였으므로 半導体 不況 속에서도 그다지 영향이 없었다. 그래서 이 不況期에 美国系는 自然退職을 포함하여 增雇원을 30% 줄였고 日本系는 8% 정도 줄었다」고 한다. 그 半導体不況도 美国의 컴퓨터産業 回復에 따라 금년부터 회복, 지금은 말레이시아 공장에서의 생산도 好況期의 85%까지 회복되었다고 한다.

그렇게 경과해온 말레이시아의 電子工業. 政

府도 일본으로부터의 공장유치를 열심히 추진하고 있으나 금년 이후에는 어떻게 변할 것인지. 현재의 엔高에 따라 日本企業의 生産 전환이 어떻게 될 것인가인데 이미 東南亞를 비롯하여 海外로의 生産 전환으로 움직이기 시작했고 움직이려는 기업도 적지 않다.

먼저 말레이시아측이 유치기업에 부여하는 Incentive, 技術 Resource, 電力, 물 등 Infrastructure를 어느 정도 제공할 수 있는 가이다. 이에 行政지도가 없혀 製造의 認許可, 資本比率, 푸미푸트라 등이 다른 국가에 비교하여 전체의 균형까지 따져 外資가 말레이시아에 進出할 것인지, 다른 국가로 나갈 것인지 선택하게 될 것이다. 일본의 生産轉換은 지금의 엔高 環境下에서는 진전되며 그것도 東南亞에서 찾을 경향이 크다.

말레이시아가 진지하게 기업유치, 기술이전을 일본 또는 외국으로부터 얻는다면 역시 外國系에는 낮은 푸미푸트라 政策을 성급하게 내세워서 안된다. 이번 IMP에서 出資比率 등이 약간 수정되어 外資를 우대하는 형태로는 만들었으나 기술이전 외에 문제점도 많다고 한다. 너무 초조하게 생각지 말고 시간을 들여 푸미푸트라가 진실로 강해지도록 하는 것이 좋은 방법일 것이다. (다음 號에 계속)