

Microprocessor의 計測 System에 대한 應用

技術解説

—溫度 計測 System—

西村 謙一郎*

—차 례—

- 1. 서 론
- 2. System의 實例
- 3. Microprocessor에 관한 문제점
- 4. Assembler 및 Simulator의 실제
- 5. 결 론

1. 서 론

종래의 單體測定器를 조합한 System은 독립적인 기능을 가진 機器의 集合體이므로, 구성이 복잡하며 불필요한 점이 많았고, 또한 그들의 機器는 Hard Wiring을 主體로 하고 있기 때문에 融通性이 부족하고 擴張性 역시 매우 작다.

한편 Microprocessor를 기본으로 한 System은 従來形 System의 缺點을 보충함과 동시에 다음과 같은 點에서 특히 그 특징이 있게 된다.

- 1) System의 低價格化
- 2) System構成의 간략화, 혹은 機動性의 증대
- 3) 信賴性의 向上
- 4) 融通性, 應用性의 擴大
- 5) 기능, 성능의 向上

2. System의 實例

다음으로, 소규모의 온도, 전압계측을 주제로 한 Microprocessor Based Data Aquisition System을 실례로 소개하며, Microprocessor에 의한 특징 등을 기술하고자 한다.

주된 사양 일부를 소개하면 (構成에 관하여는 그림 1참고)

- 1) 入力點 : max 500채
- 2) 入力の 종류 : 0~±20/50/100mV/1/5V의 DC電壓 PR/CA/CC/CRC/IC의 5종류의 열전대
- 3) 精度 : ±(0.01% of F.S+0.04% of R.D.G+1 Digit) at 0~40°C
- 4) 入出力 機器 : Printer, PTP, TTY, Typewriter 등이 표준사양이나, 다음과 같은 點에서, Microprocessor採用에 의한 機能, 性能상의 특징을 높일수 있다.

- 4. Assembler 및 Simulator의 실제
- 5. 결 론

1) 自己校正(Self-Caribration)

Analog入力は 20mV~5V의 5range을 高精度, 高安定으로 측정할 수 있다. 이것은 Self-Caribration方式을 Software演算으로 실현시킨 결과로써 특징이라 할 수 있다.

$$E_x = \frac{V_x - V_{zx}}{V_{F5} - V_{z5}} \cdot E_F$$

단, V_x : 入力電壓

V_{zx} : 計測 range의 zero電壓

V_{F5} : 5 Volt range의 Full電壓

V_{z5} : 5 Volt range의 zero電壓

E_F : 기준전압

E_x : 교정된 計測電壓

Analog量으로써는 唯一한 安定, 精確한 電壓, E_F 를 준비하는 것으로 高精度, 高安定을 만족할 수가 있다.

2) Linearizer의 table化

5종류의 Linearizer Table를 0.5K Byte의 ROM에 기억시킨다.

Analog方式에 비교하여 價格, 安定性, 再現性 精度 면에서 우수하다.

3) 基準冷接點 補償의 自動化

基準冷接點 補償은 온도계측에서는 가장 까다로운 문제의 하나이지만 Scanner入力部の 冬溫化의 實現과 Scanner Case에 마다 설치된 1개의 實溫檢出, Sensor의 Software처리에 의하여, 모든 열전대의 實溫補償을 할 수 있다.

4) 自己診斷

Hardwiring System에서는 自己診斷(System의)을 하기 위하여는 매우 곤란하지만, Microprocessor System에서는 거의 價格上昇없이 自己診斷의 기능을 附加할 수가 있다.

이 System에서는 BUS의 故障, I/O의 異常, Ana-

* 日本横河電機製作所 測定器技術部課長代理

log系의 異常등을 檢知할 수 있고, 故障상태를 display 시킨다.

이상으로 특징으로서의 具體例를 두세가지 기술하였다.

한편 이 System의 구성 및 入力, 出力用 bus에 관해서는 別圖에 표시하였다.

3. Microprocessor에 관한 문제점

3.1 Hardware에 관한 문제점

Microprocessor 및 그 주변에 관한 문제점을 열거하면 다음과 같은 내용이 되지만, 이들의 문제점을 충분히 파악한 후에 採用하면 큰 문제가 되지 않는다.

- 1) 處理速度 및 演算機能이 떨어진다.
- 2) Interrupt기능이 빈약
- 3) Device에 互換性이 없다.
- 4) Address방식이 限定되어 있다.

3.2 Software에 관한 問題點

3.2.1 言語

Microprocessor의 言語로서는 Assembler가 主流이지만, Intel社의 PL/M와 같은 Compiler言語도 발표되어 있다. 그러나 반복 사용하는 Software에서는 Program의 最適化를 하지 않으면 여유성이 커지게 되어 반드시 Compiler가 유리하다고는 말할 수 없다.

1) Self-Assembler

일반적으로 Microprocessor Maker가 제공하는 Self-Assembler는 다음과 같은 점에서 機能이 떨어진다.

- (1) 사용할 수 있는 Label數가 적다.
- (2) 擬似命令이 빈약하다.
- (3) Speed가 늦다.

이 때문에 실험, 개발 단계에서의 이용은 좋지만, 實用面에서는 Cross-Assembler가 필요하게 된다.

2) Cross-Assembler

Cross-Assembler에서 문제가 되는 것은 Use:측의 Computer(혹은 Mini-Computer)의 言語 및 Operating System이다. 또한 Maker측에서는 일반적으로 사용되고 있는 Computer의 Cross-Assembler를 제공하게 되므로, 기타의 Computer를 사용하고 있는 User의 경우에는 자기 자신이 Cross-Assembler를 작성하여야 한다.

위에서 기술한 바와같이 言語에 관한 문제점으로, Software作成에 Assembler를 사용하는 경우에는 Microprocessor 또는 Computer의 言語에 互換性이 없다는 점을 들 수 있다.

3.2.2 Debug

Microprocessor의 Soft ware의 debug은 이하에서 기술한 點에서 Computer의 debug에 비교하여 어려운 점이 있다.

(1) 主 Memory에 ROM(PROM)을 사용하고 있기 때문에 수정이 쉽지 않다.

(2) Hardware으로서의 기능을 附加시키지 않으면 Single Step로서 program의 實行이 곤란하다.

(3) 內部 register 및 program counter에 내용의 reading out이 불가능하다.

(4) 임의의 Address로 부터의 實行 및 停止가 곤란하다. 이러한 문제를 해결하는 方法으로서 Emulation Simulation을 생각할 수 있다.

1) Emulation

Emulation를 하기 위하여는 그 나뉠대로의 Hardware (Computer의 Microprogram이 변경할 수 있는 점 등)가 필요하므로 완전히 실시하기는 어렵다.

2) Simulation

Simulation의 경우에는 속도는 늦지 않으나 Emulation인 때와 같은 문제는 Software로 해결된다. 더욱이 Hardware인 면에서 본다면 Microprocessor의 BUS와 Computer의 BUS를 접속할 수 있으면 직접 주변 I/O의 동작 체크도 할 수 있다. 그러나 Simulation program에 관하여는 진출한 Cross-Assembler에서 발생하는 것과 같은 문제는 존재한다.

4. Assembler 및 Simulator의 실제

自社製의 Mini-Computer를 사용하여 Assembler 및 Simulation를 한 내용을 간단히 소개한다.

1) Assembler

自社製 Mini-Computer의 Assembler의 Macro命令을 이용하여 Assembler에 내포된 의사命令 등은 전부 이용할 수 있도록 하였다.

2) Simulator

Mini-Computer의 Operating System의 관리하에서 작동하는 약 1KW의 Simulation Program을 작성하여, 前述한 것과 같이 BUS에 직접 접속하여 周邊 I/O로 동작시켜 Simulation을 실시한다.

구체적으로는

(1) Resource의 Mapping

Microprocessor → Minicomputer

ALU → Simulation program

內部 Register → Memory (RAM)

Memory → Memory (RAM)

I/O → Mini-Computer의 System typewriter 혹은 實 I/O

(2) Simulator의 기능

- 1 step당의 내부 register등의 trace
- 임의의 address에서의 start, stop
- 실제 I/O를 접속한 상태에의 체크
- 실제 I/O를 접속하지 않은 경우에는 Mini-Computer의 System typewriter에 의한 會話形式의 Simulation
- Memory writer란이 있을때의 Test Stop
- Page單位로서의 BNPF Tape의 作成

5. 결 론

이상으로 간단히 Microprocessor의 計測 System으로의 應用 일부를 소개 하였으나 Microprocessor는 금 후 계속 그 形態를 바꾸어 새로운 것이 出現 되리라고 생각된다. Microprocessor를 사용하는 사용자 측에서는 이들의 동향을 注視함과 동시에 使用目的에 알맞는 올바른 것을 선택하는 일이 중요하다고 생각된다.

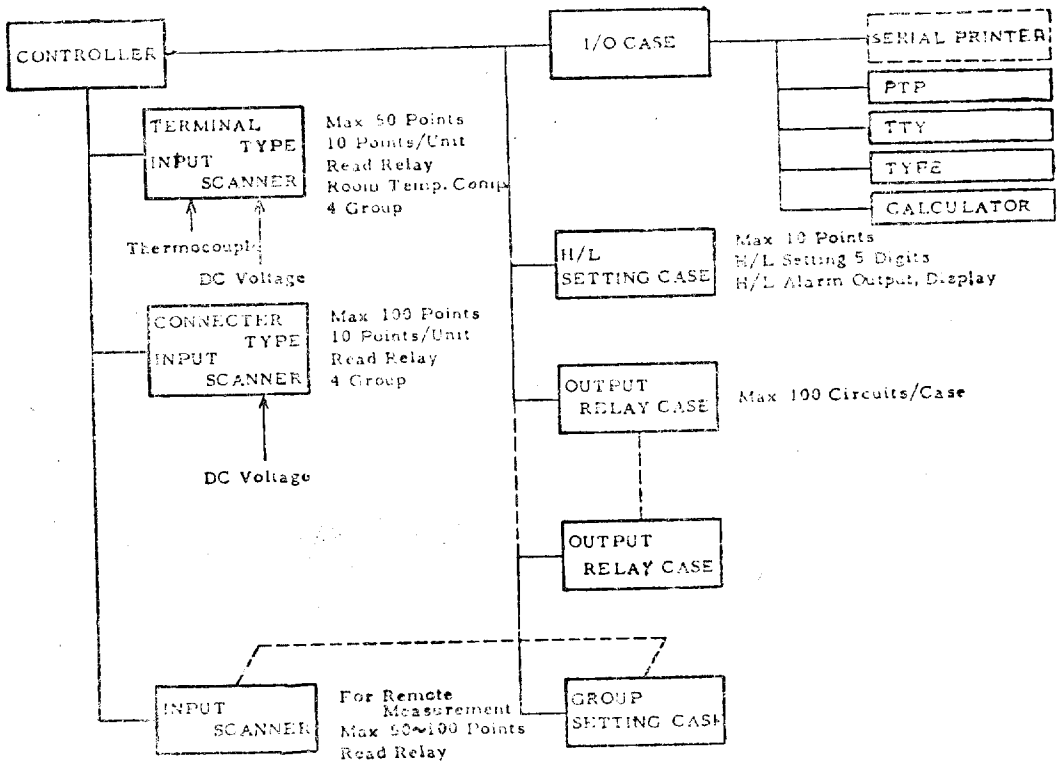


Fig. 1. System Construction

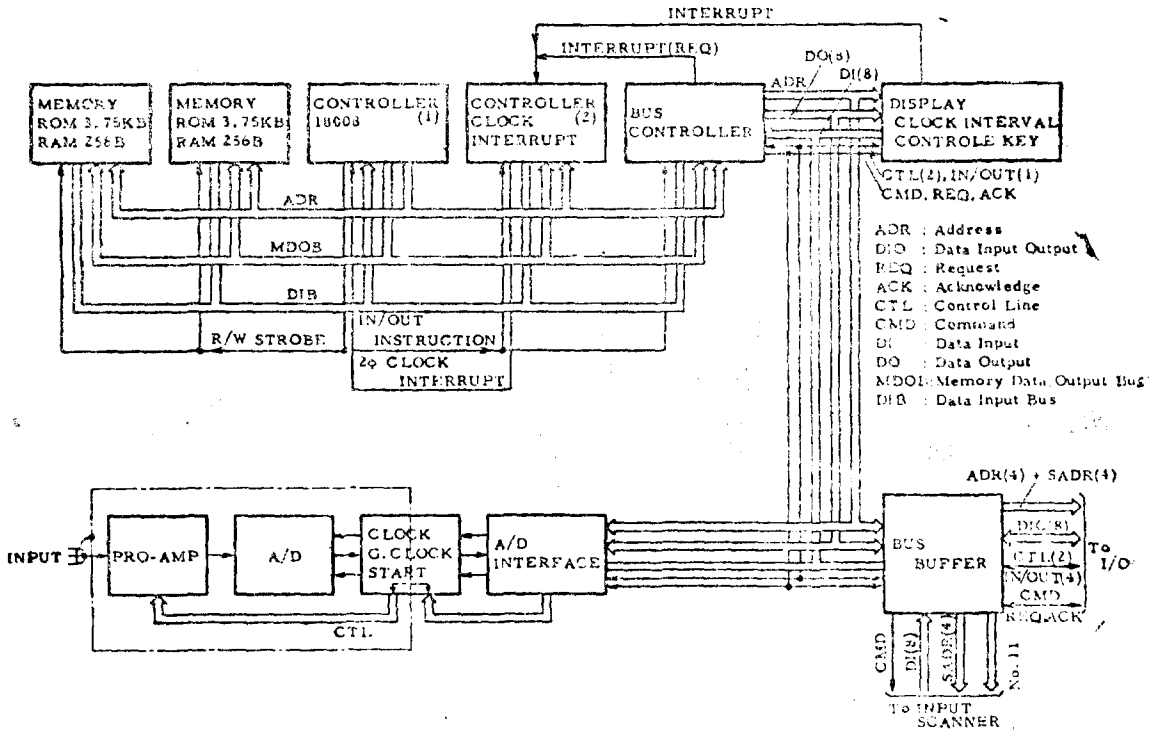


Fig. 2. Controller block diagram

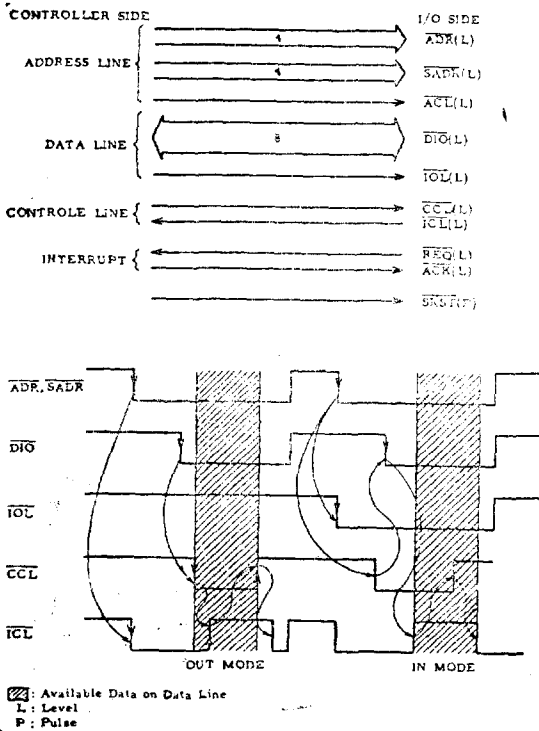


Fig. 3. Bus construction & the timing

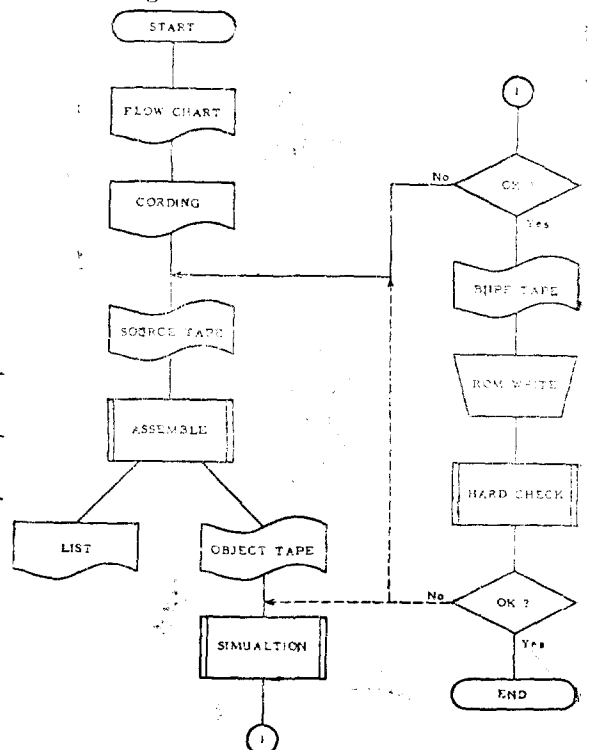


Fig. 4. Programming flow