

SFC로 記述된 계층제어 구조에서 메모리 효율 향상

유정봉^{1*}

Improvement of Memory Efficiency in Hierarchical Control Structure described by SFC

Jeong-Bong You^{1*}

요 약 PLC는 산업용 제어시스템에서 널리 사용되고 있으며, 중요한 역할을 하고 있다. PLC에서 사용되는 언어중 SFC는 이산 제어 프로그램에서 순차제어 논리의 기술에 적합한 그래픽 언어이며, SFC로 공정제어 시스템을 구현할 때는 분산제어 구조와 계층제어 구조로 구현하게 된다.

계층제어 구조에서는 서브시스템 간에 동기를 맞추기 위해서 서브시스템은 별도로 구성하고 동기신호만을 주고받게 된다. 그러면 메모리 효율 및 시스템 성능이 좋지 않게 된다.

본 논문에서는 계층제어 구조에서 메모리 효율을 향상시키는 방법을 제안하였고, 실례를 통해 그의 타당성을 확인하였다.

Abstract Programmable Logic Controller(PLC) is the most widely utilized and plays an important role in industrial control system. Sequential Function Chart(SFC) is a graphic language which is suitable for describing a sequential control logic in discrete control system. We can design a distribute control construction and a hierarchical control construction in process control system described by SFC.

In hierarchical control structure, we construct each subsystems to synchronize a synchronous signal between subsystems, and the command system gives and takes a synchronous signal with subsystems. Therefore, the system has a low memory efficiency and a low system performance.

In this paper, we propose the method that improved the efficiency of memory in hierarchical control construction, and confirm its feasibility through an actual example.

Key Words : PLC, SFC, Discrete control system, Hierarchical control construction, Memory efficiency

1. 서론

복잡하고 다양한 기능을 수행하는 현대의 공정제어에서 사용되는 PLC(Programmable Logic Controller)는 LD(Ladder Diagram) 언어를 가장 많이 사용한다. LD 언어는 릴레이 제어반의 여러 요소를 소프트웨어로 대체시켜 점접 및 코일로 표현할 수 있게 한 언어로써, 조합논리의 표현에 기반을 둔 언어이므로 조건과 인터록 논리의 기술에는 우수한 장점이 있다. 그러나 제어의 흐름을 나타내는 순차제어 논리의 기술에는 많은 어려움이 있다.

SFC(Sequential Function Chart)는 이산 제어 프로그램

에서 순차 제어 논리의 기술에 적합한 강력한 그래픽 언어이므로 제어의 흐름을 이해하기 쉬우며, 유지보수가 용이하고, 프로그램의 기술성이 뛰어나고, 기계장치의 진단성이 우수하다는 장점이 있다. SFC로 기술하여 공정제어 시스템을 구현할 때는 분산제어 구조와 계층제어 구조의 형태를 사용하게 된다[1-2].

분산제어구조는 각 서브시스템을 별도의 제어기로 구성하고, 이들 서브시스템 상호간 동기를 위한 공통시퀀스를 두게 된다. 그리고 각 서브시스템들은 독립성을 유지하면서 서브시스템 간에 동기를 이루게 된다. 이와 같은 구조는 공정시스템의 동작면에서 제약이 따르게 되어 응답시간 또는 유지보수가 용이해야 되는 경우에 적합하다.

계층제어구조는 각 서브시스템을 별도의 제어기로 구성하여 각 서브시스템을 공통 유니트와 동기를 주고받도

¹공주대학교 전기전자공학부

*교신저자: 유정봉(jbyou@kongju.ac.kr)

록 결합하여 동작을 행하게 된다. 이러한 구조는 서브시스템간에 직접적인 정보교환은 없고, 서브시스템간 동기를 위한 제 3의 동기시퀀스를 구성하여 각 서브시스템의 어느 한 스텝으로 제 3의 동기 시퀀스를 통해서만 정보를 교환하게 된다. 이와 같은 구조는 시스템 확장이 용이한 곳에서 많이 사용하게 된다[3-5]. 그러나 이러한 경우는 각각의 스텝에 대한 동기를 취하기 위한 동기스텝들이 각 서브시스템의 각 스텝에 할당되기 때문에 메모리의 효율이 떨어지고, 시스템 성능이 떨어지게 된다.

본 논문에서는 계층제어구조로 제어기를 설계할 때 각각의 서브시스템을 설계하고 결합하여 메모리 효율을 높일 수 있는 방법을 제안한 후 In-Line Spin Coater에 적용하여 메모리 효율이 향상됨을 확인한다.

2. 계층제어구조

그림 1과 같이 제 3의 동기 신호를 통해 서브시스템 1과 command part의 정보를 교환하고, 다시 서브시스템 2와 command part의 정보를 교환하는데, 이 때 각 서브시스템과 command part 간의 동기 실현이 어려운 점이다. 또한 각 서브시스템간에는 직접적인 정보의 교환은 없다[3].

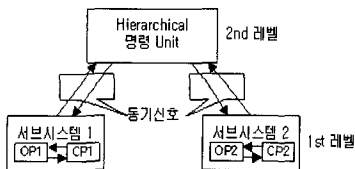


그림 1. 계층제어구조에서의 동기

계층제어구조 제어기를 설계할 때는 병렬시퀀스에 의해 서브시스템의 독립성을 보장하고, 1레벨의 서브시스템들이 각각 2레벨에 동기를 실현함으로써 서브시스템들이 통합된 하나의 제어 구조를 갖도록 해야 한다. 따라서 계층 제어 구조의 필요조건으로 다음 2가지를 만족해야 한다.

- (1) 각 유닛은 독립성을 유지해야한다.
- (2) 각 유닛은 동기화가 가능해야한다.

위의 2가지 조건을 만족시키면서 1레벨의 서브시스템을 결합한 후 2레벨의 Host 시스템에 동기시켜 결합하고 하나의 전체 시스템을 설계하게 된다.

그림 1에서 두 서브시스템간에는 직접적인 정보교환이 없고, 서브시스템간의 동기를 위해 제3의 동기시퀀스

를 구성하여 각 서브시스템들을 기능연계 상 필요한 어느 한 스텝으로 제3의 동기 시퀀스를 통해서만 정보를 교환하게 된다.

본 논문에서 제안하고자 하는 계층제어구조에서 각 서브시스템의 결합을 위해서는 다음의 과정을 거치게 된다.

- (1) 개별 서브시스템을 설계한다(1레벨).
- (2) 각 서브시스템의 동기화를 위한 명령 유니트를 공통으로 설정한다.
- (3) 명령 유니트에서 각 서브시스템으로 분산되는 트랜지션의 동기를 설정한다.
- (4) 각 서브시스템이 합쳐지는 재동기 트랜지션의 동기를 설정한다.
- (5) 각 서브시스템을 결합한다.

3. 적용 예

3.1 시스템 개요

최근 초고속 정보화의 시대에 발맞추어 디스플레이 산업은 급속한 성장을 보이고 있고, 이러한 추세는 앞으로 상당기간 지속될 것으로 예측된다. 이중 PDP 및 LCD의 기술은 세계의 선두를 유지하고 있다. 그러나 디스플레이에 대한 장치산업은 30%의 국산화율에 머무르고 있는 실정이다. 본 연구에 사용된 시스템은 TFT-LCD 생산공정중 일부 장비로써 그림 2와 같이 구성되고 있다[6].

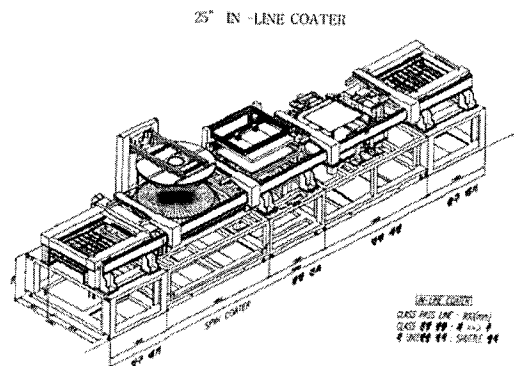


그림 2. In-Line Spin Coater 구성도

그림 2의 In-Line Spin Coater는 입구대기부, 스펀코터부, 감압건조부, 단면세정부, 출구대기부 그리고 셔틀부의 6개 유닛으로 구성된다. In-Line Spin Coater는 유리기판에 포토레지스트 도포액을 얇게 도포하는 장비이다. 입구대기부는 포토레지스트를 도포하기 위해 이송하는 유닛이며, 스펀코터부는 포토레지스트가 도포된 기판을 회

전시켜 포토레지스트를 기판 전체에 균일하게 분산 도포 시켜주는 유닛이다. 그리고 감압건조부는 스핀코터부의 진공 척과 단면세정부의 진공 척의 온도차이로 인한 유리기판 이면에 흔적이 발생하는 것을 방지하기 위한 유닛이다. 또한 레지스트가 도포된 것을 세정하는 단면세정부와 기판을 다른 공정으로 이동시켜주는 출구대기부가 있다.[6]

3.2 알고리즘 설계

전체 알고리즘을 설계하기 위하여 그림 2의 In-Line Spin Coater에서 입구대기부, Spin Coater부, 감압건조부, 단면세정부, 출구대기부 및 Shuttle부의 6개 서브시스템으로 구성하고, 기능 및 구조의 특성상 서브시스템 기능의 독립성을 보장하고, Shuttle부에 동기를 맞춰 병렬 기능을 수행하도록 하였다. 이에 대한 구조는 그림 3과 같다.

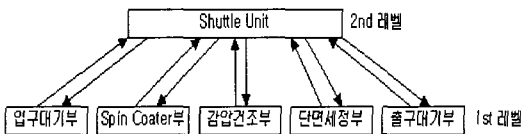


그림 3. In-Line Spin Coater의 계층제어구조

그림 3에서 병렬 시퀀스에 의해 1레벨을 구성하는 5개의 서브시스템이 2레벨을 구성하는 Shuttle부와 각각 기능상 독립성을 유지하며, 동기를 맞추었다.

본 알고리즘은 서브시스템 결합을 설명하기 위해 간단히 입구대기부와 출구대기부에 대한 알고리즘만을 설명한다.

그림 3의 In-Line Spin Coater 계층제어 구조에 맞는 입구대기부의 서브시스템은 그림 4와 같고, 출구대기부의 서브시스템은 그림 5와 같다.

그림 4에서 S101, S104, S107 스텝은 1st 레벨의 각 스텝에 대한 동기 스텝으로 사용되며, S103, S106, S109 스텝은 1st 레벨의 각 스텝에 대한 재동기 스텝으로 사용되기 위한 'wait' 스텝이다.

또한, t100, t103, t106, t109 천이조건은 2nd 레벨과의 동기 및 재동기를 위한 천이조건으로 사용된다.

그림 5에서 S201, S204, S207 스텝은 1st 레벨의 각 스텝에 대한 동기 스텝으로 사용되며, S203, S206, S209 스텝은 1st 레벨의 각 스텝에 대한 재동기 스텝으로 사용되기 위한 'wait' 스텝이다.

또한, t200, t203, t206, t209 천이조건은 2nd 레벨과의 동기 및 재동기를 위한 천이조건으로 사용된다.

그림 6은 2nd 레벨인 Shuttle 부의 서브시스템을 나타낸다.

그림 6에서 S1, S2, S3 스텝은 전체 시스템의 공통 스텝을 나타내고, S4, S7, S9 스텝은 Shuttle 부 서브시스템의 고유 스텝을 나타낸다. t4, t5, t6, t7, t8, t9 천이조건은 그림 4의 입구대기부 서브시스템과 그림 5의 출구대기부 서브시스템에 각각 동기되는 천이조건이다.

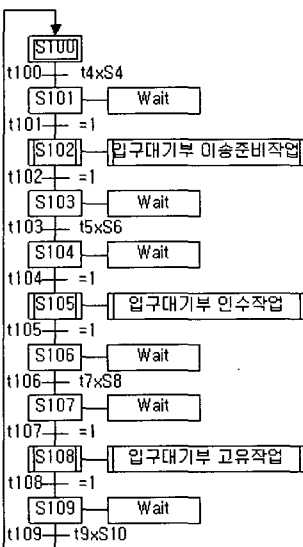


그림 4. 입구대기부 서브시스템

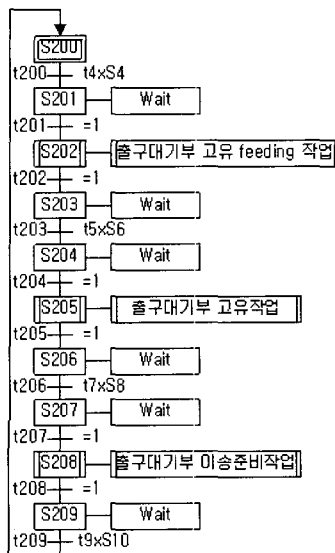


그림 5. 출구대기부 서브시스템

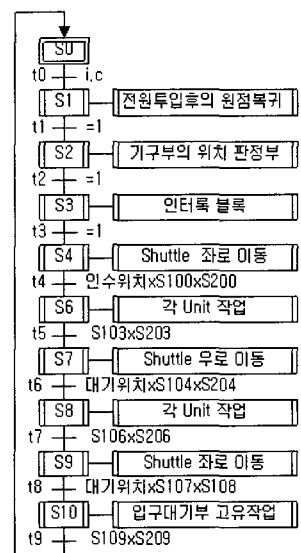


그림 6. Shuttle 부 서브시스템

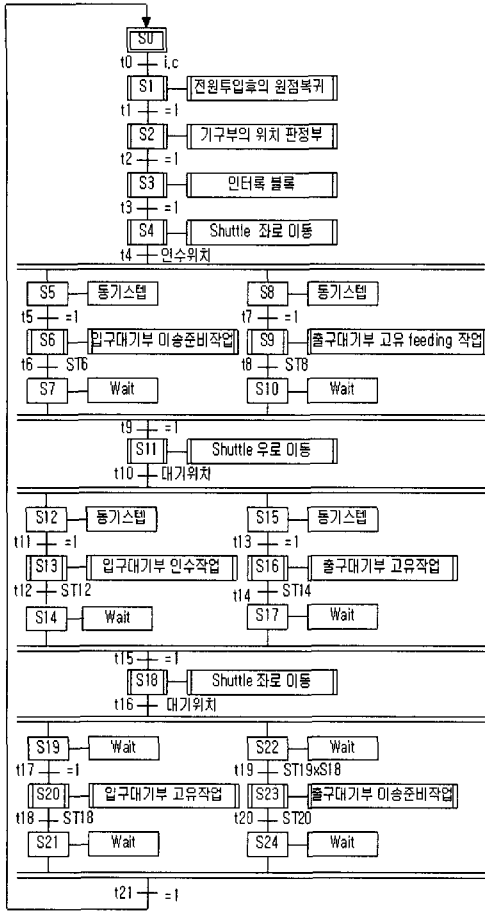


그림 7. 완성된 시스템

4. 실험결과 및 검토

본 연구를 위해서 PLC는 LG 산전의 GLOFA GM4기종의 CPU를 사용하였으며, 편집 프로그램은 GMWIN Ver 4.0을 사용하였다[7].

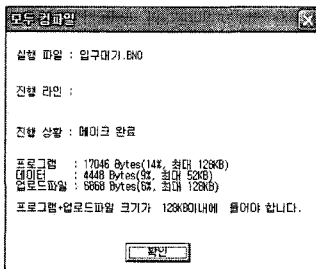


그림 8. 입구대기부 메모리 크기

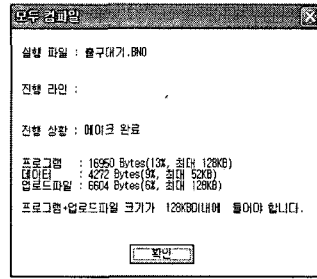


그림 9. 출구대기부 메모리 크기

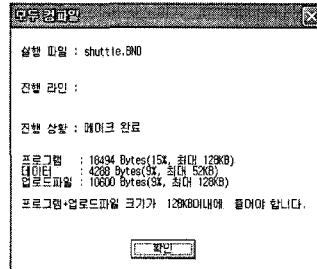


그림 10. Shuttle 부 메모리 크기

그림 8은 입구대기부 서브시스템 메모리 크기를, 그림 9는 출구대기부 서브시스템 메모리 크기를, 그림 10은 Shuttle 부 서브시스템 메모리 크기를 나타낸다. 따라서 3개의 서브시스템을 작성하여 통신으로 서브시스템 간 동작시 메모리 크기는 $17046\text{Byte} + 16950\text{Byte} + 18494\text{Byte} = 52490\text{Byte}$ 가 소요된다.

그림 11은 본 연구에서 제안한 방법을 사용했을 때의 메모리 크기를 나타낸다.

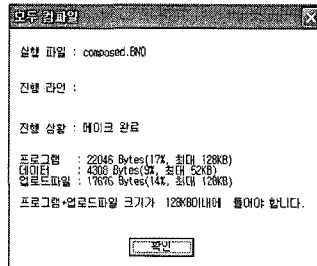


그림 11. 완성된 시스템 메모리 크기

기존의 계층제어 구조로 설계했을 때는 공통 시퀀스와 각 서브시스템을 결합할 때 각각의 스텝에 대한 동기를 취하기 위한 동기시스템들이 각 서브시스템의 각 스텝에 할당되기 때문에 그림 8, 그림 9, 그림 10의 메모리가 전부 필요하게 된다. 따라서 기존의 방법에서는 50490 Byte의 메모리가 필요하지만, 본 논문에서 제안한 결합방법을

사용하게 되면 그림 11에서처럼 22046 Byte의 메모리가 필요하게 되어 메모리가 42% 줄어들었음을 확인할 수 있었다.

5. 결론

산업 제어 시스템에서 제어기로써 PLC를 가장 많이 사용하고 있으며, 대부분 LD언어를 사용하여 프로그래밍한다. 최근에는 간단한 제어에서는 SFC 언어를 사용하는 빈도수가 높아지고 있으며, 이에 따라 SFC 언어의 연구도 활발히 이루어지고 있다.

SFC 언어는 현재 간단한 형태의 제어기 설계에서는 가능하지만, PLC 종류에 따라서 SFC의 설계 가능한 규모가 다르기 때문에 PLC 종류를 잘 선택해야 한다. 따라서 본 연구에서 제시된 방법은 좀 더 효율적인 설계가 가능하게 될 것이다.

SFC로 기술된 계층제어구조의 설계에서는 본 연구를 통한 방법을 사용하게 되면 시스템 설계도 간편하고, 메모리 효율도 향상 시키는 것이 가능하게 되어 제어기 설계기술이 한 단계 발전할 수 있을 것으로 기대된다.

참고문헌

[1] Bong-Suk Kang and Kwang-Hjun Cho, "Discrete Event Model Conversion Algorithm for Systematic Analysis of Ladder Diagrams in PLCs" Journal of Control, Automation and systems Engineering, Vol 8. No5, p401-406, May, 2002.

[2] R.W.Lewis, "Programming Industrial Control Systems Using IEC1131-3", The Institution of Electrical Engineers, 1992.

[3] Giuseppe Casalino, Giorgio Cannata, Giorgio Panin, Adrea Caffaz "On a Two level Hierarchical Structure for the Dynamic Control of Multifingered Manipulation", Proceedings of the 2001 IEEE, International Conference on Robotics & Automation Seoul Korea, 2001.

[4] Young Woo Kim, Akio Inaba, Tatsuya Suzuki, shigeru Okuma, "FMS Scheduling Based on Timed Petri Net Model and RTA Algorithm", Proceedings of the 2001 IEEE, International Conference on Robotics & Automation Seoul Korea, 2001.

[5] M. Zhou and E Twiss, "Design of Industrial automated systems via relay ladder logic programming and Petrinets", IEEE Trans on Systems, Man and Cybernetics-part C ; Applications and Reviews, Vol 28, No 1, pp 137-150, 1998.

[6] Jeong-Bong You, "Improvement Implementation of Interlock Using Management Step Described by SFC", 조명·전기 설비학회 논문지, Vol. 19, No. 3, May 2005.

[7] "LG Programmable Logic Controller Glofa-GM", LG Industrial Systems, 2004.

유 정 봉(Jeong-Bong You)

[정회원]



- 1988년 2월 : 단국대학교 전자공학과 (공학사)
- 1990년 8월 : 단국대학교 전자공학과 (공학석사)
- 1998년 8월 : 단국대학교 전자공학과(공학박사)
- 1990년 7월 ~ 1993년 3월 : (주) 신도리코
- 1999년 8월 ~ 현재 : 공주대학교 전기전자공학부 부교수

<관심분야>

PLC제어, 마이크로프로세서 제어, BLDC 모터제어, 공장자동화 알고리즘 설계 등