

# Grafchart의 Procedure step에 관한 고찰

유정봉

공주대학교 전기전자제어공학부

e-mail : jbyou@kongju.ac.kr

## A Study of Procedure Step by Grafchart

Jeong-Bong You

Dept of Electrical & Electronics & Control Engineering,

Kongju University

### 요 약

본 논문에서는 Grafchart의 기본 정의 및 설계 방법을 설명하고, 이 Grafchart와 Petrinet의 차이점을 비교하고 산업용 제어에 응용할 수 있는지를 검토한다. 본 논문에서는 LCD 장비중에 LCD의 제어정들을 처리하기 위해 LCD를 이송시키는 캐링 로봇에 Grafchart를 적용하고 그이 타당성을 확인하였다.

### 1. 서 론

Grafchart나 SFC(Sequential Function Chart)는 IEC 848과 IEC 1131-3 표준을 통해 로컬 레벨에서 순차 제어 논리를 위한 표현 형태로 산업에서 널리 사용되고 있다. 그러나 현대의 제어 형태가 복잡하고 다양한 기능을 요구함에 따라 관리자 제어 레벨에서 순차 요소에 대한 공통 표현 형태의 필요성이 있게 된다. 관리자 제어 응용은 제어 학술 부문 과 산업 제어 부문에서 관심이 높아지고 있다. 이러한 이유로 품질에 대한 인식, 환경의 변화 그리고 고객의 직접적인 제품에 의해 발생하는 기능, 호환성 그리고 안전성에 대한 요구 또한 높아지고 있다.

순차 요소는 관리자 제어에서 두개의 다른 상황을 보여준다. 첫 번째 상황은 공정제어에서 프로세스들은 대표적으로 결합된 연속적이고 순차적인 특성이 라는 사실이다. 모든 프로세스는 다른 동작모드를 갖는다. 두 번째 상황은 그 자신을 해결해야 하는 관리자 시스템은 순차 스텝으로 분리될 수 있다는 문제가 발생하는 경우에 관련되어 있다.[1][2]

Grafchart는 순차제어의 수학적 모델이다. 이 것은 Grafchart, Petri-net의 개념에 근거한다. Grafchart는 로컬 레벨의 응용뿐만 아니고, 관리자 레벨의 응용

에 목적이 있다.[3]

이 Grafchart의 기본 적인 개념을 파악하고 응용할 수 있는 분야에 적용하여 타당성을 확인한다.

### 2. Grafchart의 정의

#### 2.1 기본 정의

Grafchart는 두개의 다른 버전이 있다.[4]

하나는 Grafchart의 기본 버전으로 Grafchart에 근거한 버전이며, 두 번째는 하이 레벨 버전으로 하이 레벨 Petri-net와 통합된 개념이다.

Grafchart의 언어는 다른 언어와 마찬가지로 구문이나 의미, 활용에 있어 똑같이 정의된다. Grafchart는 기본 버전의 표현(Grafchart-BV)과 하이 레벨 버전(Grafchart-HLV)이 있다. 그러나 본 논문에서는 기본버전의 Grafchart만을 정의한다.

#### 2.2 그래픽 언어 요소

Grafchart의 그래픽 표현은 Grafchart와 아주 유사하다. 이 것은 선택분기와 병렬분기도 지원한다. Grafchart의 그래픽 요소는 스텝(Step), 트랜지션(Transition), 매크로스텝(Macro Step), 프로시저 스텝(Procesure Step), 프로세스 스텝(Process Step),

Grafchart Procedure, Grafchart Process가 있다.

(1) Grafchart Processes

전체 기능 chart는 Grafchart의 프로세스 목적으로 표현될 수 있다. 이것은 그림 1과 같다. 이 기능 chart는 사각형안에 놓여진다. 기능 chart는 하위 작업 공간에 놓여진다.

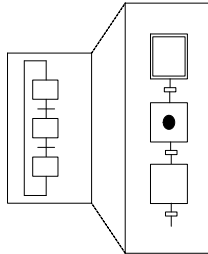


그림 1. Grafchart Process

(2) 스텝(Step)

스텝은 사각형으로 표현된다. 이것은 그림 2와 같으며, 액션이 이 스텝과 관련되고, 활성화와 비활성으로 나누어 진다.



그림 2. 스텝

(3) 초기 스텝(Initial Step)

초기 스텝은 이중 사각형으로 표현되며, 그림 3과 같다. 기능 chart가 시작될 때는 활성화되어야 한다.

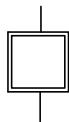


그림 3. 초기 스텝

(4) 천이 조건(Transition)

천이 조건은 다음 루틴으로 가기위한 조건을 나타내며, 그림 4와 같다.



그림 4. 트랜지션

(5) 병렬 바(Parallel Bar)

병렬 바는 병렬 가지의 시작과 끝을 나타내며, 그림 5와 같다. 병렬 가지의 시작은 'and-divergence'라하며, 병렬 가지의 끝은 'and-convergence'라 한다.

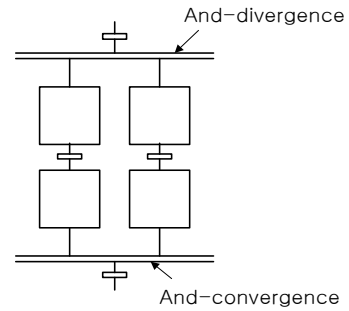


그림 5. 병렬 가지

(6) 매크로 스텝(Macro Step)

매크로 스텝은 스텝과 천이조건 그리고 매크로 스텝의 내부 구조를 표현하기 위해 사용되며, 그림 6과 같다.

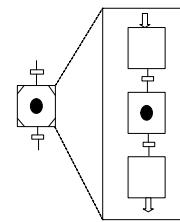


그림 6. 매크로 스텝

(7) 프로시저 스텝(Procedure Step)

Grafchart의 프로시저는 프로시저 스텝으로부터 호출된다. 프로시저 스텝은 호출되어야 하는 Grafchart의 프로시저 이름이 규정되어야 하는 프로시저 속성을 갖는다.

프로시저 스텝이 활성화될 때 즉시 Grafchart 프로시저가 호출되고 새로운 토큰이 Grafchart 프로시저의 입구 스텝으로 놓여진다. 그림 7은 프로시저 스텝을 나타낸다.

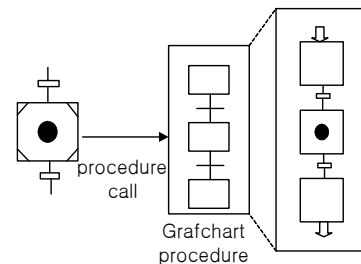


그림 7. 프로시저 스텝

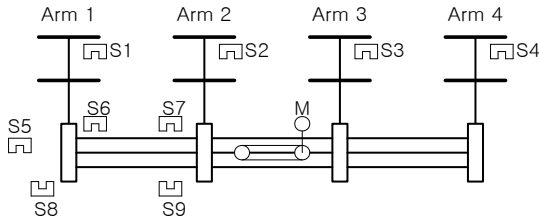
3. 알고리즘 적용 예

3.1 캐링-로봇 시스템

본 연구에 사용된 캐링-로봇의 구성도는 그림 8과 같다.

캐링-로봇은 제품을 이송시키는 것으로 주로 LCD 제조장비, 유리제조장비, 기관 제조장비 등에서

사용하게 된다. 본 논문에서는 인덱싱(indexing)이송 방식에 비해 장비의 공간적 크기를 줄일 수 있도록 셔틀에 의한 리니어 이송방식을 채택한다.



- \*Arm1~Arm4 : 기판 이송부
- \*S1~S4 : 기판 검지센서
- \*M : 구동 모터
- \*S5, S7 : Over run 센서
- \*S6 : 초기 위치 센서
- \*S8 : 상위치 확인 센서
- \*S9 : 하위치 확인 센서

그림 8. 캐링-로봇 구성도

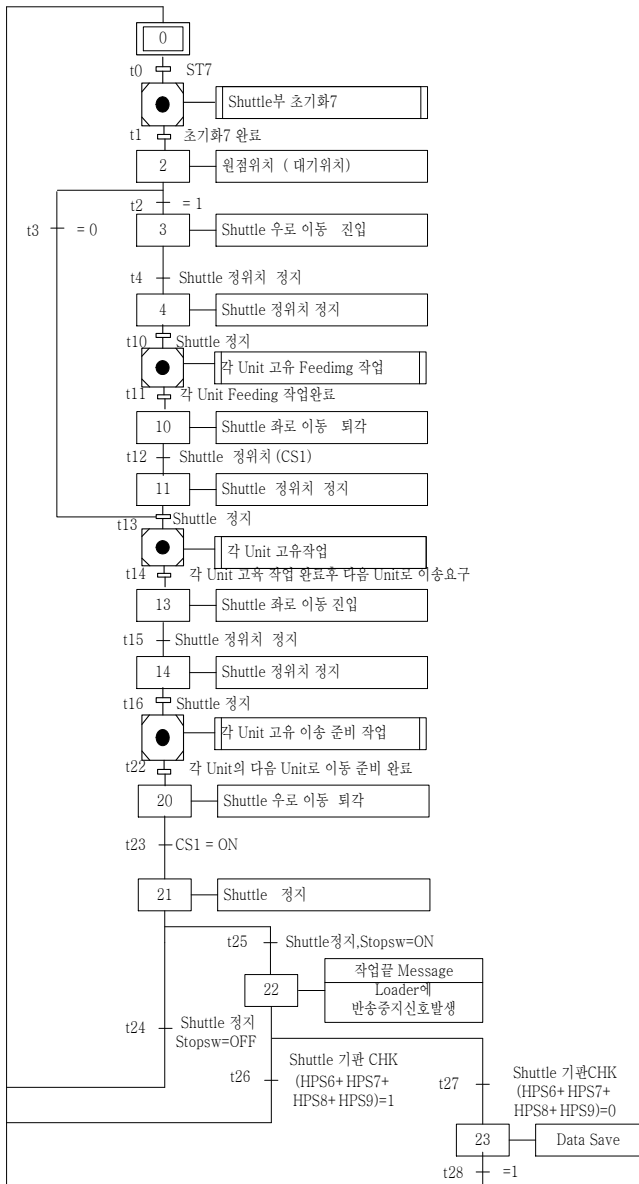


그림 9. Grafchart 알고리즘

### 3.2. 알고리즘

본 알고리즘을 설계하고 타당성을 확인하기 위해 유닛을 5개로 하여 설계하였다. 그리고 각 유닛이 동작종료 후에 각 유닛의 기판들을 동시에 오른쪽 또는 왼쪽으로 이동시키는 동작을 한다. 또한, 셔틀의 동작은 원점복귀동작, 수동조작, 자동운전 및 경보등의 기능을 가지며 셔틀이 원점위치로 복귀한 후에 제어 동작이 시작된다. 완성된 알고리즘은 그림 9와 같다.

### 4. 결 론

Grafchart의 정의를 알아본 결과 Grafcet와 아주 유사함을 확인할 수 있었다. 그러나 매크로 스텝에서 약간 상이한 점을 발견하였지만 기본적인 형태는 유사하다. 이 Grafchart를 사용하여 설계하면 산업용 제어 시스템에서 편리하게 적용할 수 있음을 확인할 수 있었고, 그 응용 분야 또한 확장될 것으로 판단된다.

### 참고문헌

- [1] Rene David, "Grafcet: A powerful Tool for Specification of Logic Controllers", IEEE Transactions on Control Systems Technology, Vol.3, No.3, pp253~268, 1979.
- [2] Bpng-Suk, Kang and Kwanghyun, Cho, "Discrete Event Model Conversion Algorithm for Systematic Analysis of Ladder Diagrams in PLCs", Journal of Control, Automation and Systems Engineering, Vol.8, No.5, p401-406, May. 2002.
- [3] R.W.Lewis "Programming Industrial Control systems Using IEC1131-3", The Institution of Electrical Engineers, 1992.
- [4] Charlotta Johnsson, "A Graphical Language for Batch Control", Department of Automatic Control Lund Institute of Technology, 1999.