IE Interfaces
Vol. 21, No. 3, pp. 294-301, September 2008.

PLC 코드 작성을 위한 공정 분석 및 적용 방법

구락조[1] · 여성주[1] · 이강구[1] · 홍상현[3] · 박창목[2] · 박상철[2] · 왕지남[2†]

[1]아주대학교 산업공학과 / [2]아주대학교 산업정보시스템공학과 / [3]기아 자동차

The Process Analysis and Application Methods for PLC Code Programming

Lock-Jo Koo[1] · Sung-Joo Yeo[1] · Kang-Gu Lee[1] · Sang-Hyun Hong[3] · Chang-Mok Park[2]
Sang-Chul Park[2] · Gi-Nam Wang[2]

[1]Department of Industrial Engineering Ajou University
[2]Department of Industrial and Information Systems Engineering Ajou University
[3]KIA Motors Corporation

Agile and flexible manufacturing systems make it mandatory that a control program should have features such as agility, flexibility, and reusability in order to run manufacturing unit smoothly. PLCs are the most frequently used control program in manufacturing systems. PLC programs are mostly programmed by subcontraction, which makes correction of code very difficult. As a result, it may cause delay during down time and ramp up time which leads to big loss of revenue and goodwill. To prevent delay during the times, this paper proposes systematic process analysis and application method for programmable logic controller like LLD (Ladder Logic Diagram). The proposed method uses modified human-error investing techniques for documentation and transforming technique to program LLD from the documentation. Furthermore, this paper demonstrates an example of piston mechanism to explain the proposed method.

Keyword: PLC(Programmable Logic Control), process analysis, code generation

1. 서 론

현대 사회에서 기업의 성공 요건은 급변하는 소비자의 요구를 빠른 시간에 맞춰주는 데 있다(No *et al.*, 1993; Zeou and twiss, 1998). 이는 짧아진 제품 생명 주기(Product Life-cycle)에 대응가능한 유연 생산 시스템을 구축하는 문제로 귀결되며, <그림 1>과 같이 새로운 제품 양산 시스템 구축 시, 발생되는 Down-Time과 Ramp up Time을 효과적으로 감소시킴으로써 해결 가능하다(Zeou and twiss, 1998).

Down-Time과 Ramp up-Time을 감소시키기 위해서는 유연성과 사용성이 높도록 제어 코드(Control Code)가 설계되어야 한다. 하지만 현업에서 사용하는 제어 코드는 대부분 하청 업체

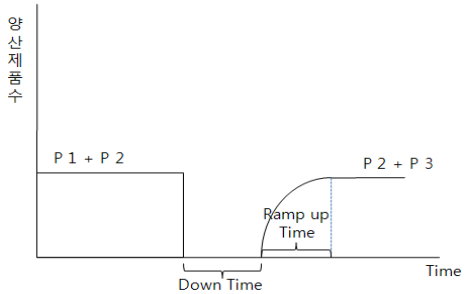


그림 1. Down-Time과 Ramp up-Time

†연락저자 : 왕지남 교수, 443-749 경기도 수원시 영통구 원천동 산 5번지 아주대학교 산업정보시스템공학부, Fax : 031-219-2983, E-mail : gnwang@ajou.ac.kr
투고일(2008년 03월 17일), 심사일(1차 : 2008년 03월 25일), 게재확정일(2008년 06월 16일).

를 통해 작성되고 있으며 코드 작성에 대한 여러 기준들의 표준화가 미비하여 실제 작성자 이외에는 추후 코드 변경에 있어 추가 분석에 대한 시간과 노력이 필수적으로 요구된다. 실제로 제어 코드 작성을 수주 받은 하청업체는 적용 라인에 대하여 일반배치도, LS(Limite sensor) 리스트, 공압 회로도, Time Table이 포함되어 있는 기계도면과 공법관련 도면, 설비 사양서 등을 기준으로 PLC 제어코드를 작성하게 된다. 작성된 PLC 제어코드는 작성자의 경험에 크게 의존되어 일반적으로 근무경력이 3년 미만인 경우 80%, 3년 이상 5년 미만일 경우 90%의 정확성을 보이며 시제작을 통한 제어 코드 검증 시 오류의 대부분은 수정되어 현장에 적용된다. 하지만 작성된 제어 코드는 새로운 제품 생산으로 인한 시스템 및 제어 코드 수정 시, PLC (Programmable Logic Controller) 언어인 LD (Ladder diagram)가 제어 장치들의 특성을 반영함으로써 일반적이지 못하고 복잡하기 때문에 Down-Time과 Ramp up-Time을 감소시키기에 많은 한계가 따른다.

본 연구에서는 코드 작성자와 분석자를 대상으로 새로운 공정에 대한 코드 설계에 있어 추후 분석 및 수정이 용이하도록 협업 및 공조 가능한 분석 방안을 제안하고자 한다. 이는 인간공학 분석 방법을 바탕으로 응용하여 정의되며, LD로의 변환을 도울 수 있는 방안도 포함한다.

2. 연구 방법

본 연구와 관련된 선행 연구는 많지 않다. 이는 특정 공정에 대한 시스템 설계 관련 문서는 대외비 문서로 관리되며 각 공정에 대한 이해는 개인적으로 이루어지기 때문이다. 그리고 현재 적용되고 있는 제어 코드는 대부분 하청 업체나 기업의 자동화 관련 부서에서 작성되어 다른 관련 부서 및 작업자의 공정 이해도를 저하시킨다. 이는 코드 작성에 있어 구체적인 문서화 작업이 이루어지지 않을 뿐만 아니라, 사후 관리에 있어 로직의 오류를 찾아낼 수 있는 기준이 미비하기 때문이다. 따라서 본 연구에서는 인간공학적 분석 방법을 사용함으로써, 전체적인 공정의 작업과 흐름을 명확히 정의하고 LD로의 변환 방법을 예시하여 현업에 도움을 주고자 하였다.

2.1 코드 설계를 위한 문서화 체계

본 연구에서 코드 설계를 위한 체계화에 사용된 방법은 사고결과 예상 계통도(Fault Tree), HTA(Hierarchical Task Analysis), Why-Because Tree이다. 이 방법들은 일반적으로 사고 조사를 위한 직무 분석에 활용되고 기본적으로 작업자의 행위나, 특정 결과에 대한 인과관계를 파악 가능하다. 하지만 본 연구에서는 PLC 간의 데이터를 통신함으로써 이루어지는 제어 장치의 흐름, 필요정보를 분석하기 위해 수정, 적용되었다.

(1) 라인 흐름 분석

코드 설계가 가능한 공정 분석은 전체 공정에 대한 정의가 선행되어 있어야 한다. 이를 위해서는 각 공정의 넘버, 공정 명, 작업 내용, 작업 로봇 대수, 작업 방식, 다음 공정으로의 이동 방식이 <표 1>과 같은 정의에 따라 파악되어야 한다.

표 1. 라인 흐름 분석을 위한 필요 항목

항 목	정 의
공정의 넘버	• 해당 생산 라인 상에 정의되어진 공정 넘버
공정 명	• 한 공정에서 이루어지는 주 작업
작업 내용	• 주 작업을 이행하기 위한 구체적인 작업 내용
작업 로봇 대수	• 작업 수행에 있어 관여되는 로봇의 수
작업 방식	• 자동방식 : 로봇을 통한 이행 • 수동방식 : 작업자를 통한 이행 • 혼합방식 : 자동방식+혼합방식
이동 방식	• 다음 공정으로 이동 될 때의 구동 방식(셔틀방식과 롤러방식)

본 연구에서는 전체 공정의 흐름을 파악하기 위해서 본 연구에서는 사고결과 예상 계통도를 응용하여 적용하였다. 사고결과 예상 계통도는 사고 및 인간 행위 분석을 통한 인적 오류 분석에서 주로 사용되는 방법으로써, 사고의 근본원인(Root Cause) 분석이 가능하다(Ladkin, 2001). 하지만 본 연구에서는 시작 공정을 사고로 간주하고 다음 공정을 이전 공정의 사고 원인이 아닌 결과로 정의하였다. 이는 전 공정의 결과를 다음 공정의 입력으로 제공하는 일련의 흐름으로서 간주하기 때문이다. 그리고 사고결과 예상 계통도의 event로써 표현되는 각 공정은 앞서 정의된 6가지의 기본사항을 정의 가능하게 하였다. <그림 2>와 <그림 3>은 사고결과 예상 계통도와 본 연구에서 제시되는 공정 흐름 분석 양식이다.

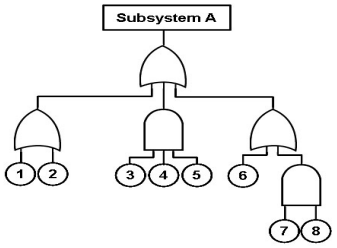


그림 2. 사고결과 예상 계통도

본 연구에서 제시한 공정 흐름 분석도는 기계도면을 통하여 공정 넘버와 공정 명, 작업 내용, 로봇 대수를 쉽게 파악할 수 있다. 이는 전체적인 프레임웍(Framework)을 파악하는 작업으로써, 전체적인 라인 구성을 제공하게 된다. 그리고 인터뷰나 공법 분석을 통하여 파악해야 하는 작업

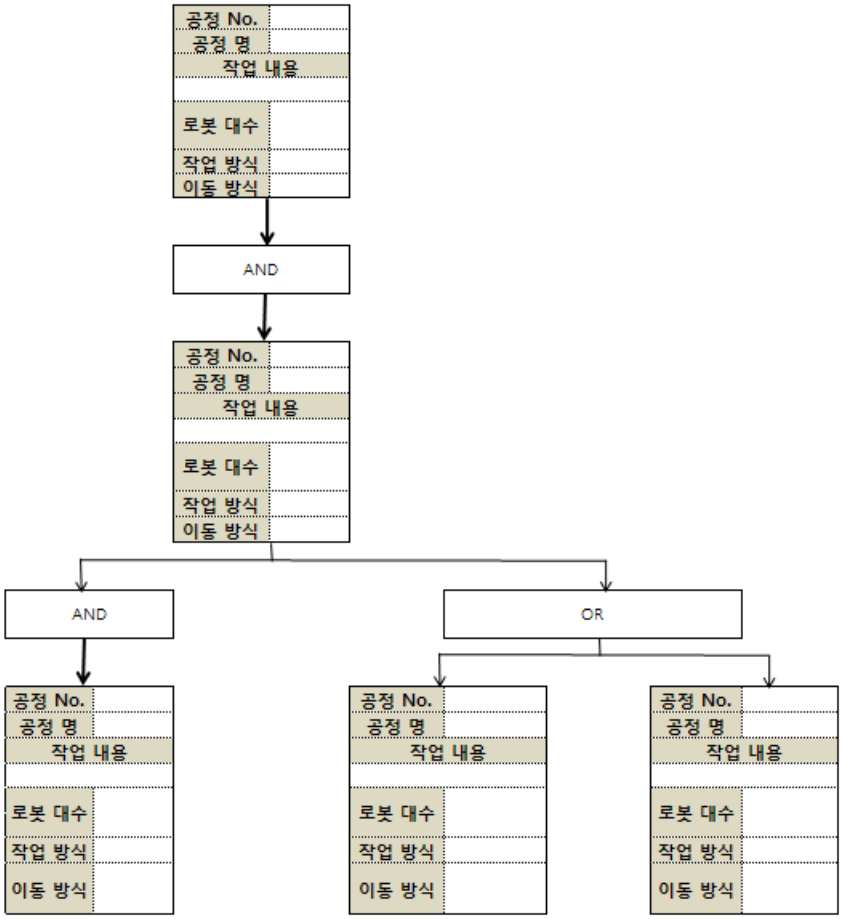


그림 3. 라인흐름 분석도

방식과 이동 방식은 그 결과에 따라 구동되는 장치가 상이하므로 세부 공정 파악 및 제어코드 작성에 필수적인 정보가 된다.

(2) 공정 흐름 분석

앞서 라인 흐름 분석을 통해 각 공정에서 이루어지는 작업과 그 형식에 대해 파악 가능하다. 하지만 제어 코드 작성을 위해서는 각 공정에 위치한 장치들의 동작 순서와 구동 입력 센서 등이 정의되어야 한다. 이런 정의 과정을 공정 흐름 분석이라 하며, 본 연구에서는 HTA와 Why-Because Tree를 적용하여 이루어졌다.

HTA는 작업자의 직무를 순서대로 분석함으로써 인적 오류를 발견 및 분석하는 방법으로, 인적 오류와 훈련(Training)의 내용, UI(User Interface)등의 분석이 가능하다(Annett *et al*., 2000; Annett and Satnton, 2000). Why-Because Tree는 사고 조사 시 근본 원인을 찾아내는 방법이다. 이는 사고결과 예상 계통도와 유사하게 진행되나, 외부요인과 사고에 대한 대응책이 그래프로 표현 되는 특징을 지닌다(Braband and Brehmke, 2002; Loughborouth University, 2006). 본 연구에서는 기본적으로 Why-Because Tree를 수정 · 적용하며, 각 그래프의 내용을 HTA 방법론에 따라 정의하였다. 정의된 내용은 동작 명, 선행 조건, 구동 센서, 입력 솔레노이드, FSA(Finite State Automata) 및 수식 표현(regular expressions) 그리고 기타 정보이다. 이 내용은 코드 작성에 있어 필수적인 내용일 뿐만 아니라 <표 2>와 같이 HTA 방법론의 목적(Goal), 운영(Operation), 계획(Plan)에 기초한다.

표 2. 정의 내용과 HTA 방법론 비교

HTA 방법론	공정흐름 분석도의 정의 내용
목적 (Goal)	• 동작 명 : 공정에 존재하는 장치명과 동작
운영 (Operation)	• 선행 조건 : 선행되어야 할 동작 • 구동 센서 : 동작 후 반응하는 센서 • 입력 SOL : 장치 동작 동력원
계획 (Plan)	• FSA 혹은 수식 표현 : 위의 정보를 표현하고 제어 코드의 인터록(Interlock)을 제거하기 위한 이산 사건 모델링 방법 • 기타 : 동작에 관여 된 기타 정보

정의된 내용은 <그림 4>와 같이 일반적인 Why-Because Tree의 표현 방식이 아닌 각 동작의 순서를 기준으로 인과 관계를 표현한다. 이는 현장에서 장치의 동작이 정해진 순서대로 이루어지지 않을 때를 에러(Error) 상황으로 간주하기 때문이다. 그리고 외부 원인(External cause)의 경우 다른 공정의 장치 동작이나 공정 흐름에 직접적인 연관성이 없는 장치의 동작으로 정의하고 그 연결은 <그림 5>와 같이 직접연결이 아닌 동작 상태 변화에 화살표로 연결하여 표현하였다. 또한 Why-Because Tree에서 표현되는 해결방안(Counter measure)은 공정의 마지막 작업으로써, 화살표가 아닌 점선으로 표현하였다. 이는 마지막 작업이 없는 대부분의 반복 생산 시스템 공정에서는 표현되지 않을 수 있다. 본 연구에서 제안된 결과는 <그림 5>와 같다.

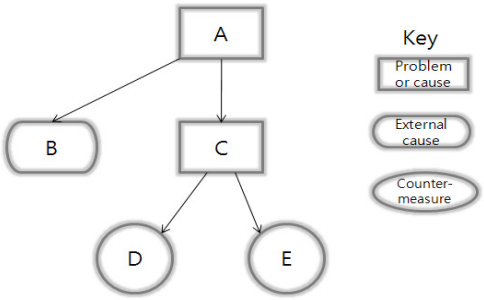


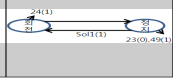
그림 4. Why-Because Tree

2.2 공정 분석도의 LD 전환

공장 자동화를 위해 가장 일반적으로 사용되는 것은 순차 제어로써 PLC을 통해 이루어진다. 이런 PLC의 제어 언어는 LD(Ladder diagram), IL(Instruction List), SFC(Sequential Function Charts), FBD (Function Block Diagram), ST(Structured Text)가 있으며 사용자에게 가장 익숙한 그래픽 언어인 LD가 널리 사용되고 있다(Bryan, 1988). 따라서 본 연구에서는 공정 흐름 분석도의 정보를 바탕으로 LD로 변환 방법을 설명한다.

동작 명	롤러회전
선행 조건	A+B
구동 센서	23: 감속단 24: 정지단 49: 확인단
입력 SOL	Sol1 : 롤러 서보
FSA(수식)	$23^{c} \times 24 \times 49$
기타	인터락은 롤러 정지

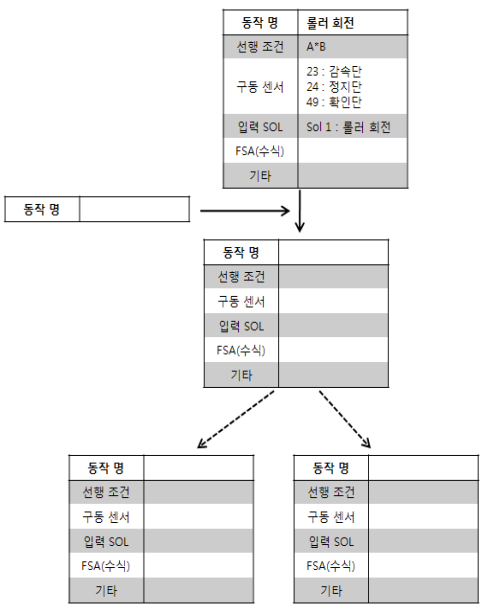
(1) 수식 정의 예

동작 명	롤러회전
선행 조건	A+B
구동 센서	23: 감속단 24: 정지단 49: 확인단
입력 SOL	Sol1 : 롤러 서보
FSA(수식)	
기타	

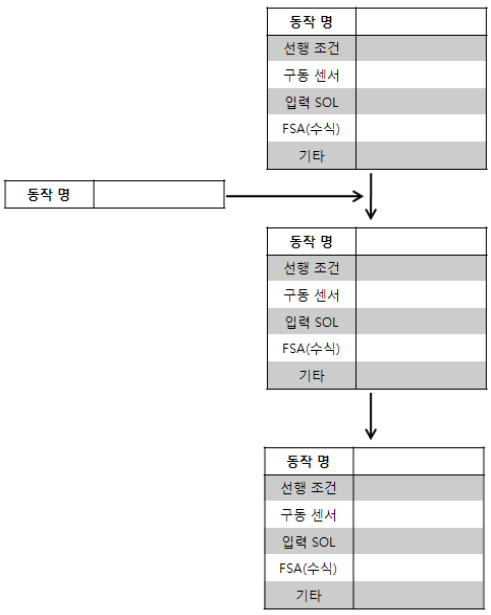
(2) FSA 정의 예

그림 6. 공정흐름 분석도 예시

본 연구에서 제시한 공정 흐름 분석도의 선행 조건과 FSA의 결과를 바탕으로 LD로 변환이 가능하다. 선행 조건과 FSA



(1) 순방향 생산 시스템



(2) 반복 생산 시스템

그림 5. 공정흐름 분석도

는 구동 센서와 입력 솔레노이드를 통해 표현 가능하며, 본 절에서는 하나의 동작을 바탕으로 이를 설명한다. <그림 6>은 이동 방식이 Roller 형태인 공정의 롤러회전 동작을 정의한 공정 흐름 분석도의 한 부분이다.

선행 조건으로는 A와 B가 있으며 롤러 회전을 위해 필요한 센서는 감속단, 정지단, 확인단이 있다. 이 정보를 바탕으로 FSA 또는 수식이 작성된다. 수식 표현에 대한 기본 정의는 다음과 같다.

- "+"는 병렬을 의미한다.
- "*"는 직렬을 의미한다.
- 각 변수는 LD의 입력을 의미하며 불리언(Boolean) 상수로써, 1(True) 또는 0(False)의 값을 지닌다.
- 입력 D에 대한 D^c은 D에 대한 여집합으로써 |(1-D)|의 값을 지니며 LD에서의 닫힌 접점(normally closed contact)을 의미한다.

FSA는 장치 별 동작 표현에 적합한 상태(State) 기반의 object 모델을 사용한다. 이는 generic 모델로써 생산 라인의 각 구성 요소들을 표현 가능하다. 이에 대한 설명은 <표 3>과 같고 아래와 같이 총 8개의 Tuple로 구성된다(Park *et al*., 2006).

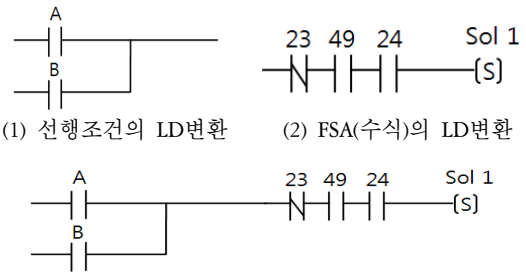
$FSA = \{I, O, S, \delta_{ext}, \delta_{int}, \lambda_{ent}, \lambda_{leave}, \lambda_{disable}\}$

I : 입력 집합,
O : 출력 집합
S : 상태 집합
δ_{ext} *: 외부 변환 함수*
δ_{int} *: 내부 변환 함수*
λ_{ent} *: 진입 출력 함수*
λ_{leave} *: 이탈 출력 함수*
$\lambda_{disable}$ *: 억제 출력 함수*

표 3. Generic 모델의 Tuple 설명

Tuple 예시	설 명
$\delta_{ext}(A, I\|\Delta t) = B$	상태가 A이고 I가 입력이 되면 Δt 후, B로 상태가 변한다.
$\delta_{int}(A, \Delta t) = C$	현재 상태가 A이면 Δt 후, C로 상태가 변한다.
$\lambda_{ent}(B) = O_1$	모델이 상태 B로 들어올 때, PLC로 O_1을 출력한다.
$\lambda_{leave}(B) = O_2$	모델이 상태 B에서 나갈 때, PLC로 O_2을 출력한다.
$\lambda_{disable}(B) = M_1$	모델이 상태 B로 들어올 때, 모델 M_1을 억제한다.

본 연구에서는 FSA의 δ_{int}과λ_{leave}를 LD 변환의 접점으로 간주하고 δ_{ext}을 LD의 출력으로 정의하였다. 정의된 방법에 따라 변환된 LD는 <그림 7>과 같다.



(1) 선행조건의 LD변환 (2) FSA(수식)의 LD변환

(3) LD 변환의 완성

그림 7. LD로 변환 과정 예시

3. 사례 적용

본 연구는 국내 한 자동차 생산 라인을 분석하기 위해 적용 진행 중이다. 하지만 이는 앞서 언급하였듯이, 기업의 대외비 자료에 해당한다. 따라서 본 연구에서는 간략한 사례를 바탕으로 적용 방안에 대해 설명하고 적용 결과를 예시한다.

3.1 사례 적용 문제 정의

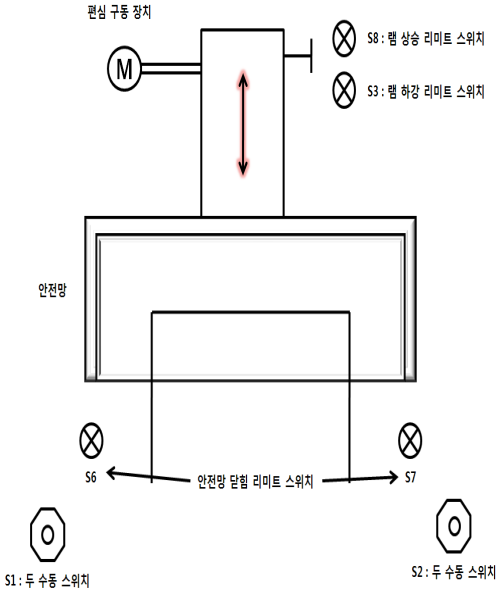


그림 8. 피스톤기계공정개요모형

본 연구에서는 프레스 기계의 동작에 관련하여 제시한 방법을 적용하며, 프레스 기계는 다음과 같이 동작이 정의된다(Kwangwoon University, 1993).

"프레스는 기계가 주기를 시작하기 전에 열려있던 안전망이 닫힐 때, 프레스가 홈 위치에 있을 때, 그리고 2개의 활성화되지 않은 수동 스위치가 동작할 때 하강한다. 두 개의 수동 스위치 중 하나가 풀려 있을 때, 안전망이 열려있을 때, 그리고 피스톤이 아래 위치에 있을 때 피스톤은 정지한다. 피스톤

이 아래에 있을 때 구동기에 의해 상승되며 안전망은 이때 열릴 수 있고, 두 수동 스위치는 풀릴 수 있다. 또한 기계의 주기는 피스톤이 홈 위치에 있을 때 완전히 끝나게 된다."

이에 대한 공정에 전체 개요는 <그림 8>과 같다.

3.2 공정 흐름 분석 및 LD 변환

본 연구에서 적용하고자 하는 프레스 기계는 특정 단위 시스템의 한 라인이 아닌, 한 공정으로 간주될 수 있다. 따라서 일반적인 기계 도면으로부터 정보를 획득하여 제안 형식에 맞게 작성되는 라인 흐름 분석도를 제외하고 공정 흐름 분석도 작성에 대해 언급한다.

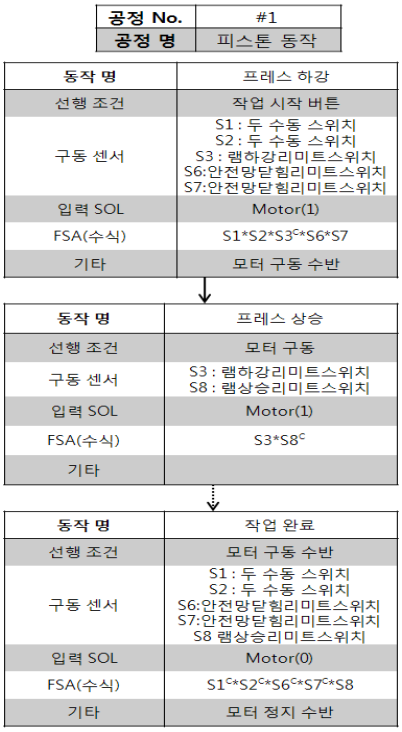


그림 9. 피스톤기계공정흐름분석도

앞서 정의된 문제로부터 동작은 프레스 하강, 상승, 완료로 구분 가능하다.

첫 번째 동작은 작업 완료 이후 새로운 공정 시작으로써, 프레스의 하강이 발생한다. 이때 필요 센서는 안전망 닫힘 스위치와 두 수동 스위치이다. 두 번째 동작은 피스톤 상승이며 편심 구동 모터에 의해 작동되며 피스톤 하강 이후, 램 하강 리미트 스위치 반응 이후 발생한다. 마지막으로 작업 완료 동작은 피스톤 상승의 결과물이며 이 공정의 최종 작업으로써 모든 공정의 신호를 해지함과 동시에 피스톤의 하강 준비를 완료한다.

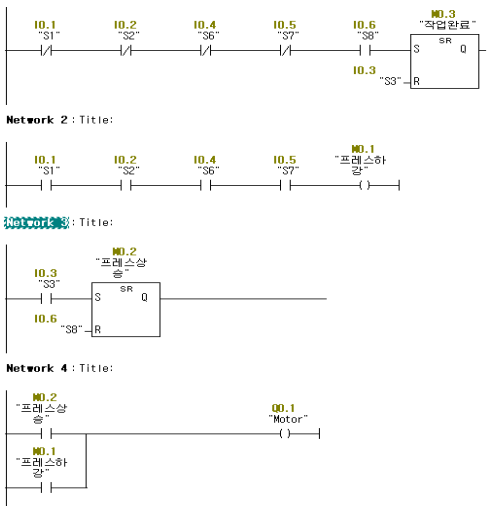
이를 본 연구에서 제시한 공정 흐름 분석도로 작성하면 <그림 9>와 같다.

4. 결 과

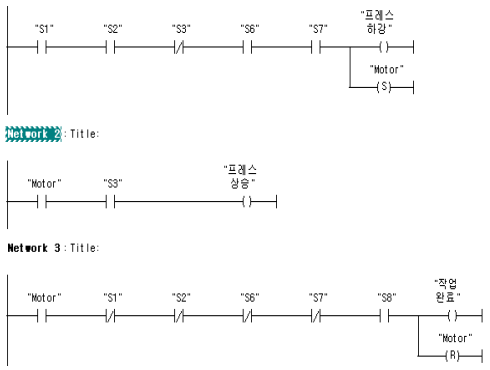
본 연구를 통해 제시되는 결과는 분석 대상 라인의 공정 순서와 그에 따른 정보이다. 이는 현재 대부분의 제어 코드가 하청업체를 통해 이루어지고, 자동화 이외의 관련 부서에서 제어 코드에 대한 충분한 이해가 이루어지지 않고 있음을 해결하고자 함이다. 본 연구에서는 앞서 제시한 사례의 해답 LD와 본 연구에서 제시한 방법을 통해 표현된 LD를 <표 4>와 같이 비교해 보았다. 피스톤 기계에 대한 공정 흐름 분석도를 바탕으로 작성된 LD와 사례의 해답 LD는 <그림 10>과 같다. 이는 독일 Siemens의 Step 7을 사용하여 작성되었다.

표 4. 피스톤 기계 공정 흐름 분석도

	모범 LD변환	제시된 LD
I(입력) 변수 수	6	6
Q(출력) 변수 수	1	1
M(메모리) 변수 수	3	1
Network 수	4	3
로직의 오류	없음	없음

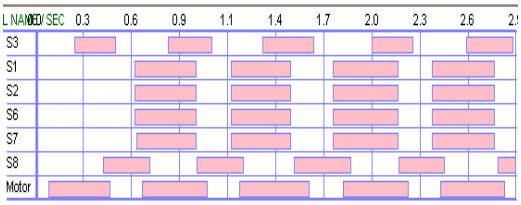


(1) 모범 LD변환 예

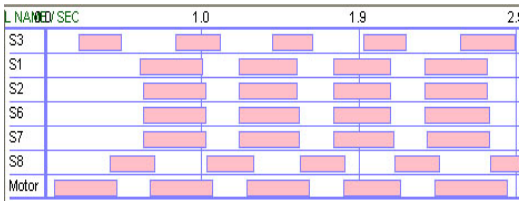


(2) 제시된 방법을 통한 LD 변환

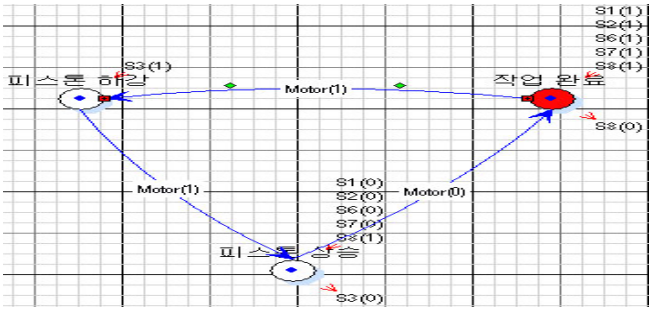
그림 10. LD 변환



(1) 모범 LD변환 예의 검증 결과



(2) 제안 방법 검증 결과



(3) PLC-Studio 검증 소프트웨어 모델링

그림 11. 결과 LD의 오류 검증과 결과 비교

제어 코드의 작성은 작성자의 경험과 지식 등에 의존한다. 따라서 모든 LD 로직에 대한 검증이 필요하며, 본 연구에서 제시한 모범 LD 또한 그 검증이 필요하다. 따라서, 본 연구에서는 아주대에서 개발한 PLC-Studio 검증 소프트웨어와 Siemens의 가상 시뮬레이터 Sim과의 연결을 통하여 제시된 LD 검증이 이루어졌다(Development of Next Generation Technology, 2007). 그 결과, <그림 11>과 같으며 "모범 LD 변환 예"와 본 연구의 결과로써 "제시 된 LD" 모두 오류가 없고 각 센서 및 작동 모터가 같은 시간에 동일한 결과를 나타냄을 확인하였다(Development of Next Generation Technology, 2007; Kirwan, 1994).

5. 결론 및 토의

본 연구는 LD로의 코드 설계에 있어 협업 및 공조 가능한 체계화 방안에 대한 연구이다. 이를 위해 인간 공학의 사고 조사 및 인적 오류(Human Error)의 분석 기법을 활용하였다. 이는 사고 조사 및 오류 조사 시 밝혀진 결과에 대하여 그 근본 원인과 해결안을 찾는 과정을 기계 장치의 동작의 순서로 정의함으로써 가능하게 되었다.

즉, 기계장치의 동작은 전 동작의 결과인 동시에 다음 동작의 원인으로 정의함으로써, 원인 분석과 같은 형식을 응용 할 수 있었다.

제어 코드 자동 생성을 위한 소프트웨어는 Rockwell Automation사의 Enterprise Controls와 Tecnomatix의 eM-PLC를 대표적이다(Music *et al*., 2005). 이와 같은 소프트웨어를 사용하기 위해서는 기본적으로 동작 및 공정(Operation) 순서 및 입력 센서 등이 정의되어 있어야 한다. 하지만 현장에서는 이와 같은 지식이 공유되지 않고 있다.

본 연구를 통해 제시된 문서화 방법은 주어진 정보를 체계적으로 정리가능하고, 제어 코드 작성 관련 직무자들이 로직 개발 초기부터 사용한다면, LD 로직 작성에 있어 도움을 받을 수 있을 것이다. 또한 추후 공정 분석 및 제어 코드 수정 시 체계화된 문서로부터 정보를 추출하여 수정 · 보완이 용이할 뿐만 아니라, 앞서 제시한 코드 생성 소프트웨어 사용 시 활용 가능하며 공정에 대한 CAD 데이터 획득 시, 즉시 적용 가능할 것이라 판단된다. 하지만 현재 제안된 방법은 비록 국내 한 자동차 생산 라인에 시도되고 있긴 하나 그 검증이 아직 이루어지지 않았고 본 연구에서 제시한 사례 검증 또한 대규모 시스템이 아닌 단일 공정으로 그 한계가 있다. 이는 현장에서 코드 작성 시 많이 사용되는 FB(Function Block)이나 DB(Data Block) 형태의 입력 및 출력 접점 표현 방안과 같이 향후에 지속적으로 연구되어야 한다. 또한 효율적인 사용을 위하여 본 연구에서 제시한 방법을 전산화함으로써 제어 장치에 대한 전사적 정보 공유 및 이해가 가능하고 이를 통해 코드 분석 및 변경에 따른 소요 시간이 감소될 것이라 사료된다.

참고문헌

Annett, J., Cunningham, D. J., and Mathias-Jones P. (2000), A method for measuring team skills, *Ergonomics*, 43(8), 1076-1094.

Annett, J. and Stanton, N. A. (2000), *Task Analysis*, Taylor and Francis.

Braband, J. and Brehmke, B. (2002), Human factors application area of Why-Because Graphs to Railway Near-Misses, In Workshop on the investigation and Reporting of Accidents (IRIA).

Bryan, L. A. (1988), *Programmable Controllers-Theory and Implementation*, Industrial Text, New York.

Development of Next Generation Technology. (2007), Digital Progress Simula-

tor and OLP System for Automobil Part Industry, Ajou University industry academic collaboration foundation.

Kirwan, B. (1994), *A Guide to Practical Human Reliability Assessment*, Taylor and Francis.

Kwangwoon University. (1993), *Simatic S5 PLC Programming Example and Solving*, 134-135, Kwangwoon University Press.

Ladkin, P. (2001), *Causal System Analysis-Formal Reasoning About Safety and Failure*, University of Bielefeld.

Loughborough University, University of Michican. (2006), *Automatic Generattion of Logic Control*, 10-15, Ford Company.

Music, M., Gradisar, D., and Matko, D. (2005), IEC 61131-3 Compliant Control Code Generation from Discrete Event Models, *Proceedings of the 13th Mediterranean Conference on Control and Automation*, 346-351.

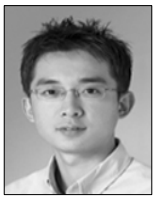
No, S.-D., Kim. K.-B., Kim. J.-W., and Lee. K.-I. (1993), A Study on the Real Time Control of Flexible Manufacturing System, *Proceedings of The Korean Society of Mechanical Engineers*, 1(1), 976-981

Park, C.-M., Bajumaya, S. M., Park, S.-C., Wang, G.-N., Kwak, J.-G., Han, K.-H., and Chang, M.-H., (2006), Development of Virtual Simulator for Visual Validation of PLC Program, International Conference on Computational Intelligence for Modelling, Control and Automation, 29 November - 1 December, Sydney, Australia.

Zeou, M. and Twiss, E. (1998), Design of industrial automated systems via relay Ladder Logic Programming and Petri Nets, Transactions on systems, man- and cybernetics-Part C : applications and reviews, **28**(1), 137-150.

구 락 조
아주대학교 산업정보시스템공학부 학사
아주대학교 산업공학과 석사
현재: 아주대학교 산업공학과 박사 과정
관심분야: 공장자동화, 이산사건시스템, 시뮬레이션, 이미지 프로세싱

여 성 주
아주대학교 산업정보시스템공학부 학사
아주대학교 산업공학과 석사
현재: 아주대학교 산업공학과 박사 과정
관심분야: 공장자동화, 이산사건시스템, 시뮬레이션, Data Mining, CIM/ERP/SCM



이 강 구
아주대학교 산업정보시스템공학부 학사
현재: 아주대학교 산업공학과 석사 과정
관심분야: 공장자동화, 이산사건시스템, 시뮬레이션, 공정 최적화

홍 상 현
건국대학교 전기공학과 학사
울산대학교 공업교육학 석사
현재: 기아자동차 설비제어기술팀 & 아주대학교 산업공학과 박사 과정
관심분야: 이산사건시스템, 시뮬레이션, 설비제어, 제어로직 자동화

박 창 목
아주대학교 산업정보시스템공학부 학사
아주대학교 산업공학과 석사
아주대학교 산업공학과 박사
현재: 아주대학교 산업공학과 연구 교수
관심분야: 공장자동화, 이산사건시스템, 시뮬레이션, 설비제어

박 상 철
KAIST 산업공학과 학사
KAIST 산업공학과 석사
KAIST 산업공학과 박사
현재: 아주대학교 산업공학과 교수
관심분야: CAD/CAM, 공정 계획, 이산사건 시스템, 시뮬레이션

왕 지 남
아주대학교 산업공학과 학사
KAIST 산업공학과 석사
미 Texas A&M 대학 산업공학과 박사
현재: 아주대학교 산업공학과 교수
관심분야: CIM/ERP/SCM, 공정 계획, 이산사건 시스템, 시뮬레이션