

현장실무자를 위한 프로그래머블 콘트롤러(12)



글쓰는 순서

1. 프로그래머블 콘트롤러 소개 (1)
 - 정의
 - 역사적 배경
 - 동작 원리
2. 프로그래머블 콘트롤러 소개 (2)
 - 타 기종제어에 대한 PLC
 - 대표적 PLC 응용산업
 - PLC 제품의 응용범위
3. 프로그래머블 콘트롤러 소개 (3)
 - 래더다이아그램과 PLC
 - PLC 사용의 이점
4. 디스크리트 입·출력 시스템 (1)
 - 소개
 - 입·출력 탭과 테이블 매핑
 - 원격 입·출력 시스템
5. 디스크리트 입·출력 시스템 (2)
 - 디스크리트 입력
 - 디스크리트 출력
6. 아나로그 입·출력 시스템(1)
 - 아나로그 입력
 - 아나로그 입력 데이터 표시
 - 아나로그 입력 데이터 취급
 - 아나로그 입력 결선
7. 아나로그 입·출력 시스템 (2)
 - 아나로그 출력 데이터 표시
 - 아나로그 출력 데이터 취급
 - 아나로그 출력 결선
8. 특수 기능 입·출력 시스템 (1)
 - 소개
 - 특수 디스크리트 인터페이스
 - 온도 인터페이스
9. 특수 기능 입·출력 시스템 (2)
 - 위치 인터페이스
10. 통신 인터페이스 시스템
 - 아스키 인터페이스
 - 베이직 모듈
 - 네트워크 인터페이스
 - 주변기기 인터페이스
11. PLC 시스템 다큐멘테이션
 - 소개
 - 다큐멘테이션의 단계
 - PLC 다큐멘테이션 시스템
12. PLC 시스템 수행 및 프로그래밍(1)
 - 제어 정의
 - 제어 원칙
 - 수행 지침
 - 수행 절차
13. PLC 시스템 수행 및 프로그래밍 (2)
 - 디스크리트 입·출력 제어 프로그래밍
 - 아나로그 입·출력 제어 프로그래밍
14. PLC 시스템 수행 및 프로그래밍 (3)
 - 간단한 프로그래밍 예제
15. PLC 시스템 수행 및 프로그래밍 (4)
 - 설치, 시운전 및 보수 지침 (1)
 - PLC 시스템 배치
 - 시스템 전환 및 안전 회로
 - 노이즈, 열 및 전압 고려사항
16. 설치, 시운전 및 보수 지침 (2)
 - 입·출력 설치, 배선 및 주의사항
 - PLC 시스템 및 점검 절차
 - PLC 시스템 보수
 - PLC 시스템 고장진단
17. 설치, 시운전 및 보수 지침 (3)
 - PLC 시스템 선정 지침 (1)
 - 소개
 - PLC 크기 및 응용범위
18. PLC 시스템 선정 지침 (2)
 - 프로세스 제어시스템 정의
 - 기타 고려사항들
 - 요약

PLC 시스템

수행 및 프로그래밍(1)

글/동양화학공업(주) 자동화사업부

제어 프로그램의 수행에 있어서는 구성과 순서에 관한 한, 많은 단계를 포함한다. 우리들 대부분은 반드시 고려해야만 하는 모든 요인을 배려하지 않고 프로그래밍 RUNG, 복합적인 제어의 시퀀스 등으로 보통 뛰어들게 마련이다. 본 연재를 숙독함으로써 PLC를 프로그래밍하기 위한 좋은 출발점이 어디에 있는가를 여러분은 깨닫게 될 것이라고 믿는다.

본 연재에서 제시된 구성과 제어의 접근 방법을 여러분의 특수한 응용을 위한 하나의 참고사항으로써 사용을 해야만 한다. 어느 누구도 여러분에게 특정 응용문제를 프로그래밍하는 방법을 실질적으로 가르쳐 줄 수는 없고 오히려 프로그래머블 콘트롤러에서 이용 가능한 기법과 기능을 사용하여 그 문제 해결을 위해 시도할 수 있는 방법을 가르쳐 줄 수가 있다.

7-1 제어 정의

사용자는 제어임무를 정의하고 수행함에 있어 무엇이 필요한 지를 확인함으로써 그 문제 해결 프로세스를 착수해야 한다. 이러한 정의 정보는 수행되어야 하는 프로그래머블 콘트롤러 운전 결정을 위한 기본을 제공해 준다. 수행되어야 하는 제어 정의는 기계 또는 프로세스의 운전엔 친숙한 사람들에 의해서 행해져야 한다. 이것이 프로세스를 잘못 이해함으로써 발생하는 에러를 가능한 한 최소화 하는 데에 도움이 될 것이다.

제어임무 정의는 통상적으로 많은 레벨에서 발생한다. 관계된 각 부분의 개인들은 어떤 입력이 요구되고 어떤 입력이 제공되어야 하는 지를 결정하기 위해서는 서로 협의해야 하며 모두가 그 프로젝트에 있어서 무엇을 수행해야 할 지를 이해할 수 있어야 한다. 예를 들면, 어떤 자재를 창고에서 찾아내서 자동포장 지역에 보내야 하는 제조 플랜트의 자동화를 포함하는 프로젝트에 있어서, 창고와 포장 지역의 담당자는 시스템을 정의하는 동안 엔지니어링 그룹과 협의를 해야만 한다. 또한 데이터 리포트를 필요로 하는 경우에는 관리가 포함되어야 한다.

만일 하나의 임무를 현재 수동으로 또는 릴레이 로직을 통해서 수행한다면, 이러한 절차의 단계는 만일 가능하다면, 어떤 것을 개량해야 할 것인가를 결정하기 위해서 재검토되어야 한다. 비록 릴레이 로직이 PLC에서 직접 실행할 수 있을 지라도, 현재 응용의 필요성을 충족시키고 프로그래머블 콘트롤러가 제공하는 기능의 이점을 택하는 절차를, 가능하다면 재설계하도록 조언할 수 있다.

그러한 임무를 설계할 때에 고려해야 할 요인들은 결과적인 프로그램의 성패에 모든 것이 긴밀히 관련되어 있다. 이러한 관계는 기계 또는 프로세스의 올바른 제어를 제공하는 기능으로 밝혀질 것이다.

7-2 제어 원칙

제어임무가 정의된 후에 그 해결을 위한 계획이

시작될 수 있다. 이 절차는 출력제어를 만들어내기 위해서 한 프로그램내에서 이루어져야 할 단계 처리 순서의 결정이 일반적으로 포함된다. 이러한 부분의 프로그램 개발은 알고리즘의 개발로서 알려지고 있다.

알고리즘이란 용어는 어떤 독자들에게는 새롭고 낯선 용어일 것이다. 우리들 각자는 우리의 일상생활에서 어떤 임무를 수행하기 위해서 알고리즘을 따른다. 한사람이 집에서 학교로 가거나 또는 일하러 가기 위해 따르는 절차가 하나의 알고리즘이다. 그 사람이 집을 떠나서, 자동차를 타고, 엔진에 시동을 걸거나 하는 등등. 한정된 수의 최종 단계에서, 그 목적에 도달된다.

PLC를 사용한 제어임무를 위한 원칙적 수행은 알고리즘의 개발을 밀접하게 따른다. 사용자는 주어진 세트의 명령어로부터 제어를 수행하고 한정된 수의 그러한 명령어내에서 해결책 또는 해답을 만들어 내야만 한다. 대부분의 경우에 있어서 문제 해결을 위한 알고리즘의 개발은 가능하다. 만일 그렇게 하는 것이 어렵게 된다면 좀더 나아가는 정의가 필요할 수도 있다. 이러한 경우에, 문제 정의 단계로 되돌아갈 필요가 있게 된다.

프로그램 원칙을 정의하는 데 있어서 기본적인 원칙은 다음과 같다. 우선 생각하고 다음에 프로그램 한다는 것이다. 문제 해결에 대한적절한 접근방법을 고려한다. 제어기능의 프로그램을 시도하기 전에 그 접근방법(해결책 알고리즘)을 다듬을 수 있는 시간을 갖도록 한다. 이 원칙의 적용은 프로그램 시간을 단축시키고, 디버깅 시간을 감소시키며, 스타트 업을 가속시키고, 설계할 때에는 설계에 주의를 집중시키고, 프로그램할 때에는 프로그램에만 주의를 집중시키게 해준다.

원칙의 형성 단계중에는, 사용자는 새로운 응용 또는 기존 프로세스 또는 기계의 현대화에 직면하게 될 것이다. 어떤 응용이 행해져야 하는가에 관계없이 사용자는 일어나는 사건의 순서를 검토해야만 하고, 단계의 추가 또는 삭제를 통하여 제어를 최적화 해야만 한다.

입력 및 출력의 고려사항이 언급되어야 하며, PLC가 제어하게 될 어떤 현장장치에 관한 지식이 요구된다.

7-3 수행지침

프로그래머블 콘트롤러는 수행 지시받은 것만을 행하고, 그렇게 하도록 지시받은 데로만 할 수 있는 강력한 기계이다. 이 기계는 제어 프로그램에서 지시된 모든 내용, 프로그래머에 의해서 만들어진 명령어 또는 해결책 알고리즘을 받는다.

PLC 제어 프로그램의 성공은 사용자가 어떻게 구성했는가에 크게 의존한다. 문제 접근에는 많은 방법이 있지만 만일 응용을 체계적 방법으로 접근한다면, 실수를 할 확률은 줄어든다.

제어 프로그램의 수행을 위해서 사용되는 기법은 프로그램에 참여하고 있는 사람에 달려있다. 그러나 어떤 지침을 따라야 한다는 것이 항상 권장되고 있다. <표 7-1>에는 프로그래머블 콘트롤러 시스템 수행에 매우 유용한 것으로 입증된 2가지의 접근지침을 보여준다. 이들 접근중의 한 가지는 새로운 시

<표 7-1> 접근지침

새로운 응용	현 대 화
-시스템의 바람직한 기능적인 서술의 이해	-실제적인 프로세스 또는 기계기능을 이해
-가능한 제어방법을 검토하고 프로세스 운전을 최적화한다.	--가능하면 기계운전 로직을 검토하고 최적화한다.
-프로세스 운전을 플로우 차트화	--실제 I/O내부 어드레스를 할당
-로직 다이어그램 또는 릴레이 로직 심볼 사용에 의한 플로우 차트를 수행	-릴레이 래더 다이어그램을 PLC코딩으로 번역
-실제 I/O어드레스 및 내부 어드레스의 할당	
-로직 수행을 PLC 코딩으로 번역	

시스템을 위한 지침이고, 다른 한 가지는, 예를 들면 전자기계 제어 또는 개별적인 아날로그 루프 콘트롤러 등 PLC 기능을 갖추지 않은 기존 시스템을 위한 제어 시스템의 현대화 또는 개량화를 위한 것이다.

전에 언급한 바와 같이, 프로세스 또는 기계 운전의 이해가 제어 프로그램 해결을 위한 체계적 접근으로의 첫번째 단계이다. 새로운 응용을 위해서 원칙의 계획은 문제 정의를 따를 것이다. 현대화를 위한 실제 제어방법의 수정뿐만 아니라 새로운 응용에 대한 원칙의 검토는 계획 단계중에 도입되는 가능한 에러를 탐지하거나 또는 최소화하는 데에 도움을 줄 것이다.

새로운 현대화한 프로젝트간의 접근시 차이는 프로그래밍 단계중에 명백하게 된다. 현대화 프로젝트에 있어서, 사용자는 기계 또는 프로세스의 운전과 그리고 무엇을 제어할 필요가 있는가를 철저히 이해한다. 이 벤트의 순서는 통상적으로 <그림 7-1>에 보여준 것처럼 보통 PLC 래더 다이어그램으로 해석될 수 있는 기존 릴레이 래더 다이어그램에 의해서 정의된다. 그러나 가능한 에러를 피하고 구성을 유지하기 위해서는 하나의 절차를 따라야 한다.

새로운 응용은 통상적으로 제어 시스템을 설계하

고 설치할 개인에게 주어지는 사양을 가지고 시작한다. 이들 사양서들은 제어의 가능한 형태를 설명해주는 문서의 기술로 번역된다. 문서상의 설명서는 혼동을 피하도록 간단한 용어로 표현되어야 한다.

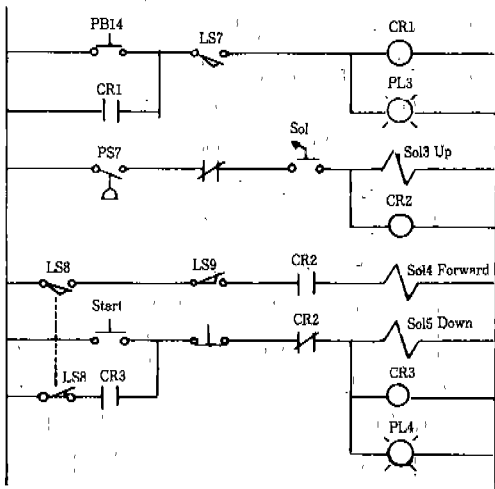
7-4 수행절차

어떤 PLC 프로그램을 시작하기 전에 반드시 수행해야만 하는 몇가지 중요한 세부사항에 관해서 본절에서 다루게 된다. 전에 언급한 바와 같이, 구성은 제어 해결을 프로그래밍하고 수행하게 될 때에 하나의 중요한 단어이다. 프로젝트가 크면 클수록 더욱 많은 구성이 필요하며 특히 많은 사람들이 관련될 때 그렇다.

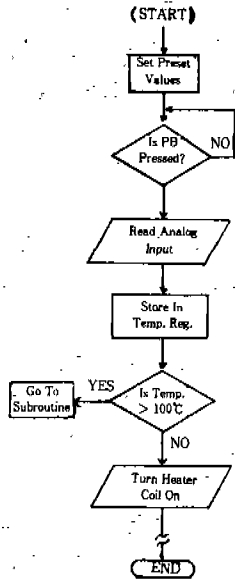
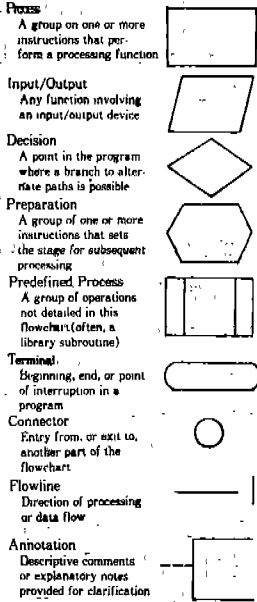
제어 해결에 대한 열쇠는 또한 그 해결을 수행할 수 있는 능력에 따른다. 사용자는 PLC 시스템에 의해서 무엇이 제어되어야 하는 지를 알아야 하고, 그 일(하드웨어와 소프트웨어)을 위한 올바른 장비를 선택하여야 하며, 그 PLC를 이해해야 한다. 일단이 모든 사전 예비 세부사항이 망라되면 우리는 제어 프로그램 해결을 스케치하기 시작할 수 있다. 이때 수행되는 모든 일이 헛되지 않으면 오히려 그것은 시스템 또는 프로젝트 다큐멘테이션의 중요부분을 형성한다. 우리를 대부분은 일단 작업한 시스템으로 되돌아가서 다큐먼트 작업을 하기를 싫어한다. 그러므로 이러한 구성을 처음부터 지키는 것이 프로젝트 말기에 가서는 그 보상을 받게 될 것이다.

플로우 차트

플로우 차트란 문서로써 기술되어진 후에 프로그램 계획에 있어서 가끔 사용되는 기법이다. 플로우 차트는 문제 정보를 기록, 분석 및 통신하는 수단으로써 역할을 하는 도식적 표현이다. 사소한 세부사항 및 서로의 관계뿐만 아니라 광범위한 개념도 쉽게 나타난다. 일반적인 기술로부터 발췌해내기 어려운 순서와 관계들은 플로우 차트로 디스플레이하면 분명해진다. 플로우 차트 기호 자체도 특정한 의미



<그림 7-1> 대표적인 전자 기계식 릴레이 회로 다이어그램



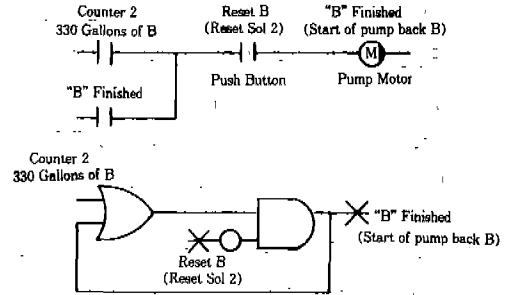
<그림 7-2> 일반적인 플로우 차트 기호 <그림 7-3> 간단한 플로우 차트 예

를 가지고 있어, 이것은 해결 알고리즘을 해석하는데 있어서 도움이 된다. <그림 7-2>는 가장 일반적으로 사용되고 있는 플로우 차트 기호와 그 의미에 관한 설명이다.

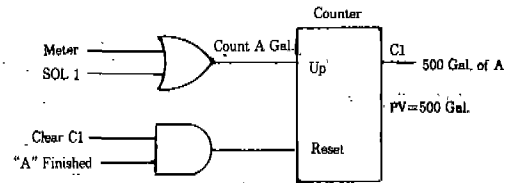
플로우 차트는 통상적인 순서 방법으로 운전의 프로세스를 기술해 놓은 것이다. 간단한 플로우 차트가 <그림 7-3>에 도시되어 있다. 이 차트의 각 단계는 입력/출력, 결정, 또는 데이터 처리이전 간에 한 운전을 수행한다.

즉, 플로우 차트 그 자체는 복잡하고 길 필요가 없다. 대신에 이것은 수행해야 할 기능을 지적해 내야 한다(예 : 아나로그 입력 카운트로부터 엔지니어링 단위를 계산해낸다). 수개의 보다 작은 플로우 차트를 주 플로우 차트에서 지정된 기능을 수행하기 위해서 사용할 수도 있다.

일단 플로우 차트가 완성되면, 로직 시퀀스는 2가지 방법중에서 한 가지로 얻을 수 있다. 첫째, 로직



<그림 7-4> (a) PLC 점접 기호 (b) 로직 게이트 표시



<그림 7-5> 로직 게이트 및 PLC 기호의 조합

게이트를 실제이전 내부이전 간에 입력 조건을 지정하기 위해서 사용할 수 있다. 둘째, PLC 점접 기호를 직접 사용하여 출력 RUNG을 표시하는 데 필요한 로직을 수행할 수 있다. <그림 7-4>는 이러한 두 가지 방법을 도시한 것이다. 사용자는 어느 것이 간에 가장 편리한 것이라고 느끼는 방법을 사용해야 한다. 아마도 필요하다면 이 두 가지 방법을 조합하여 사용할 수도 있다<그림 7-5 참조> 그러나 로직 게이트 다이어그램은 만일 콘트롤러가 BOOLEAN 명령들을 사용할 경우에는 보다 더 적합하게 된다는 것을 알게 될 것이다.

<그림 7-4>의 (b)에서 X로 표시된 입출력을 사용하여 시스템의 실제 I/O를 표시한다. 만일 표시가 없다면 I/O 포인트는 내부적인 것으로 해석할 수 있다. 실제 입력 신호를 위한 명칭은 실제 장치 (LSI, PB10, AUTO 등)일 수 있거나 또는 각 현장 요소와 관련이 있는 숫자 또는 기호적 문자일 수도 있다. 시퀀스에 관한 간단한 설명이 나중에 프로그래밍하는 동안 도움이 될 수 있으며 본 단계중에 강력히 권장된다.

PLC 시스템의 구성

다른 많은 컴퓨터 시스템 응용에서처럼, PLC 응용시에도 여러가지가 동시에 고려될 필요가 있다. 그들의 한 가지가 PLC 시스템의 구성이다. PLC 구성은 어떤 I/O 모듈이 어떤 형태의 입출력 신호와 함께 사용되며 그 모듈을 현장 또는 원격 랙 인클로우저의 어디에 둘 것인가를 정의해준다. 모듈의 위치는 우리가 제어 프로그램내에서 사용하고자 하는 입출력에 대한 어드레스를 정의해준다는 것을 기억하자.

구성하는 동안, 여러분은 차후 확장 가능성, 어떤 특수 형태의 I/O 모듈 및 랙 내에서의 인터페이스 배치를 염두에 두어야만 한다(모든 A/C I/O를 함께 그리고 모든 DC 및 하위 아나로그를 함께 등). 이러한 모든 세부사항들은 보다 좋은 시스템 설계를 수행하는 방향으로 도움을 줄 것이다. 전 연재에서 시스템 구성 다큐멘테이션의 필요사항 및 그래프적으로 그것을 표시하는 방법에 대해서 논의하였다.

실제 및 내부 I/O 할당

입력과 출력의 할당은 프로그래밍 편성 및 수행 단계중에 일어나는 가장 중요한 절차중의 하나이다. I/O 할당은 순서적인 양상으로 다큐먼트되고 지금까지 수행된 것을 편성할 것이다. 이것은 PLC 입력이 어떤 입력장치에 접속되어 있고 PLC 출력이 어떤 출력장치를 구동시키는가를 표시할 것이다. 타이머, 카운터, 그리고 MCR 등을 포함한 내부의 할당이 또한 이때 일어난다. 이들 할당은 래더 다이어그램 프로그램에서 사용되는 실제 접점 및 코일을 표시해준다. 전자 기계식 릴레이 다이어그램이 이용가능한 응용(예 : 기계 또는 프로세스의 현대화 등)에서는, 실제 I/O의 확인은 그 장치에 원표시를 하고 그 다음에 I/O 어드레스를 할당함으로써 간단히 행하여질 수 있다.

내부와 마찬가지로, 실제 입출력의 할당은 <그림 7-6>에서처럼 도표화될 수 있다. I/O 어드레스 할당과 관련된 숫자 번호는 사용되는 PLC 모델에

Module Type	I/O Address			Description
	Rack	Group	Terminal	
Input	0	0	0	LS1—Position
	0	0	1	LS2—Detect
	0	0	2	Sel Switch—Select 1
	0	0	3	PB1—Start
Output	0	0	4	PL1
	0	0	5	PL2
	0	0	6	Motor M1
	0	0	7	Sol 1
Output	0	1	0	Sol 2
	0	1	1	PL3

(a) 실제 입력 및 출력용 샘플 I/O 어드레스 할당 다큐먼트

Device	Internal	Description
CR7	1010	CR7 Replacement
TDR10	T200	Timer on Delay 12 sec
CR10	1011	CR10 Replacement
CR14	1012	CR14 Replacement
-	1013	Set-up Interlock
.	.	.
.	.	.
.	.	.

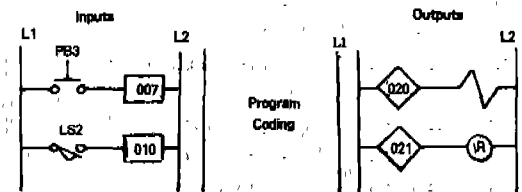
(b) 내부 출력용 샘플 어드레스 할당 다큐먼트

<그림 7-6>

달려있다. 이들 어드레스는 8진수, 10진수 및 16진수로 표시될 수 있다.

이들 할당표의 설명부분은 내부 출력 사용뿐만 아니라 입력 또는 출력현장장치(실제 I/O)를 기술하기 위해 사용된다. I/O 할당은 로직 시퀀스를 설명하기 위해 사용되었던 로직 게이트 다이어그램 또는 래더 기호로부터 또는 전자 기계식 다이어그램의 원 표시로부터 발췌될 수 있다.

할당표는 <그림 7-7>에서 보여주고 있는 입/출력 접속과 밀접하게 관련되어 있다. 입출력 표시를 위한 공업표준이 사용자들간에 다양하지만, 입력 및 출력은 대표적으로 구형과 다이아몬드로 표시된다. 이러한 I/O 접속 다이어그램은 다큐멘테이션 패키지의 한 부분을 이룬다.



<그림 7-7> 대표적인 I/O 접속 다이어그램

I/O 할당시에는 관련된 입력과 출력의 접속은 의도적으로 집합시킬 것을 권장한다. 이러한 집합은 I/O(I/O 레지스터를 통하여)의 같은 집단을 동시에 감시 및 조작할 수 있다. 예를 들면, 만일 16개의 모터가 순차적으로 기동되어야 하는 경우, 그들의 기동 시퀀스는 16개의 매핑된 I/O 점수와 관련된 I/O 레지스터를 모니터함으로써 관찰될 수 있다. 또한 I/O 시스템의 모듈성으로 인하여 모든 입력 또는 출력을 동시에 할당할 것을 권장한다. 이것을 실천하는 것이 출력 모듈에 입력 어드레스를 할당하거나 또는 그 반대의 경우를 방지해 주는 것이다.

레지스터 어드레스 할당

제어 프로그램내에서 사용되는 레지스터에 대한 어드레스 할당은 실행되어야만 하는 또 다른 중요한 편성상의 세부사항이다. 이것은 어떤 수단에 의한 지리한 작업이라기 보다는 오히려 프로그램내에서 독특한 사용을 위한 레지스터를 할당하기 위한 규율상의 접근이다.

레지스터 할당의 가장 쉬운 방법은 PLC내에서 이용가능한 모든 레지스터의 일람표를 유지하는 것이다. 그리고 이들이 사용될 때 그들에 관한 설명과 기

Register	Contents	Description
2000	Analog input	Temperature input Temp 3 (inside)
2001	Analog input	Temperature input Temp 3 (outside)
2002	spare	-
2003	spare	-
2004	TWS input	Set point(SP1) input from TWS panel 1
2005	TWS input	Set point volume (V1) from TWS panel 2
2006	constant 2350	Timer constant of 23.5 sec(.01 sec TB)
2007	Accumulated (C)	Accumulated value for counter R2010
2010	spare	-
2011	spare	-
2012	Constant 1000	Beginning of look-up table #1
2013	Constant 1010	Look-up value #2
2014	Constant 1023	Look-up value #3
2015	Constant 1089	Look-up value #4
2016	Constant 1100	Look-up value #5

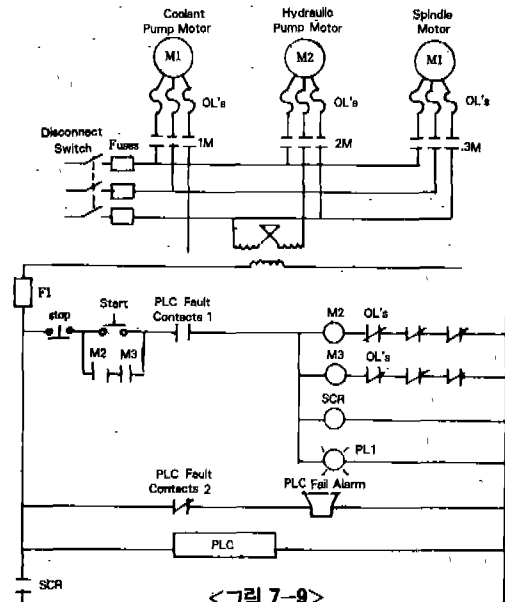
<그림 7-8> 샘플 레지스터 할당

능을 포함하여 레지스터의 내용을 표로 작성해 두는 것이다. <그림 7-8>은 어드레스 200(8진수)에서 2016(8진수)까지 PLC 시스템의 첫번째 15레지스터와 샘플 일람표를 설명해준다.

하드 와이어로 남겨야 할 부분

입력과 출력을 할당하는 동안, 어떤 요소를 콘트롤러에 배선해서는 안되는가에 대한 결정이 필요하다. 이러한 요소들은 자기 제어 로직의 부분으로 맡겨 두어야 하고, 통상적으로 콤프레서, 유압펌프 등과 같은 기동후에 자주 스위치를 끄지 않는 요소가 이에 포함될 것이다. 비상 정지(예 : 로프스위치, 푸시버튼 등)와 같은 요소 및 마스터 스타트 푸시버튼들은 또한 주로 안전 목적을 위해서 하드 와이어로 남겨 두어야만 한다. 만일 어떤 이유 때문에 콘트롤러에 고장이 발생해서 비상 정지 상황에 처한다면, 그 시스템은 PLC 개입없이도 중단될 수 있어야 한다.

<그림 7-9>는 하드 와이어로 남겨두어야 할 부분의 일 예를 대표적으로 설명한다. 이 다이어그램에서 PLC 고장점점(또는 위치독 타이머 점점)이



<그림 7-9>

직렬로 다른 비상 조건과 배선되어 있는 것에 유념한다. 이들 접점들은 콘트롤러가 정확하게 작동되고 있는 동안에는 닫혀 있으나 고장이 발생할 때 또는 시스템 설계자가 사용할 때에는 개방된다.

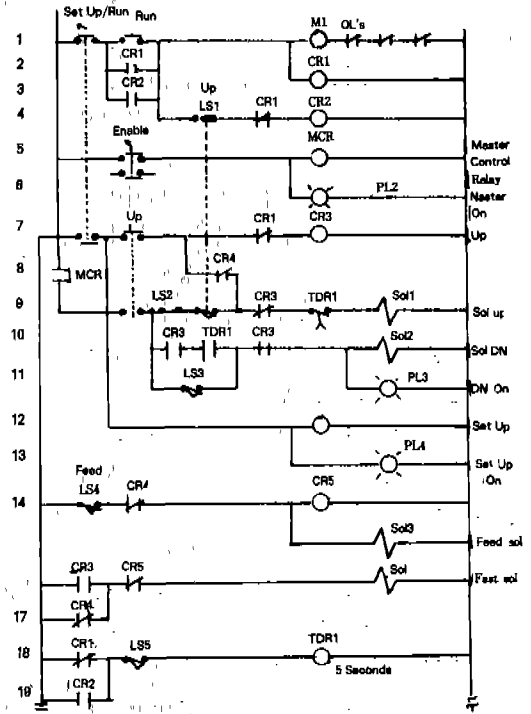
<그림 7-9>에서 보여주고 있는 다이어그램에서 만일 어떤 비상상황, 즉 PLC 오기능 등이 발생한다면 그 회로(예 : I/O 모듈)의 잔여부분에 이르는 전원(LL)은 차단될 것이다. SCR을 OFF로 바꾸면 전원을 흐르게 하는 SCR 접점을 개방하는 것이 된다. 따라서, 전원의 흐름이 차단된다. 더구나 일련의 정상 닫힘 PLC 고장접점이 하드 와이어된 부분에서 또한 사용되어 사람에게 PLC 오기능으로 인한 시스템 고장이라는 것을 경고해준다. 이러한 형태의 경고 신호를 수행하는 것이 CPU가 자리잡고 있는 주 PLC 랙에 대해서 뿐만 아니라 각각의 원격위치에 있는 I/O 랙에 대해서도 좋은 실천사항이다. 이것은 보조 시스템 고장까지도 신속하게 알려서 사람에게 위험을 주지 않고서도 신속히 문제가 해결될 수 있도록 해준다.

입력장치 프로그래밍의 특수한 경우

프로그래머블 콘트롤러 설치시의 프로그래밍 단계 중에 주의를 요하는 어떤 회로와 입력접속이 있다. 한 장치의 프로그래밍은 제어 프로그램시에 우리는 그 장치가 어떻게 동작하기를 바라는가에 밀접히 관계되고 있다는 것을 기억하여야 한다.

정상 열림으로 배선된 입력장치는 정상 열림 또는 정상 닫힘 장치로써 동작하도록 프로그램될 수 있다. 같은 규칙이 정상 닫힘 입력에게도 적용된다. 일반적으로, 만일 한 장치가 정상 닫힘으로 배선되어 있고 그것이 정상 닫힘 입력으로써 동작할 필요가 있다면, NC 입력의 참조 어드레스가 정상 열림으로 프로그램된다.

프로그래머가 알고있지 않으면 안되는 또 다른 회로는 MCR이다. 전자 기계식 회로 다이어그램에 있어서 MCR 코일은 그 관련 RUNG에 전원을 ON 또는 OFF함으로써 그 회로내의 여러개의 RUNG을



<그림 7-10> MCR로 쉐스되어있는 전자 기계식 릴레이 회로도

제어한다. 그 회로가 모두 관련되어 있을 때를 제외하고는 MCR은 일정한 끝이 없다. 하드 와이어된 래더 회로가 PLC 기호로 번역되었을 때, END MCR 명령어는 MCR이 제어하고자 하는 최종 RUNG 다음에 두어야 한다. <그림 7-10>은 이러한 형태의 MCR 조건을 갖는 부분적인 래더 RUNG을 설명한다. END MCR 명령어는 PL3 출력을 포함하는 RUNG 다음의 PLC 프로그램에 두어야 한다.

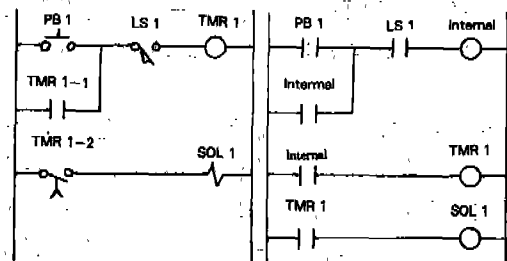
본 다이어그램에 그렇게 분명하지 않은 또 다른 조건은 정상 닫힘 CR4 접점, TDR1(10번째 라인)의 순서 접점, 그리고 물론 NC 또는 NO의 어떤 접점이 PLC에 접속되어 있는가에 따른 LS1의 프로그래밍에 의한 양방향성의 전원흐름의 가능성이다. 이것을 해결하기 위하여 우리는, 기계 기능에 따라서

CR4가 그것이 접속된 2개의 출력 RUNG에 영향을 미치는 것에 관계없이, 즉 CR3 제어 릴레이 출력 및 솔레노이드 SOL1 출력을 알아야만 한다.

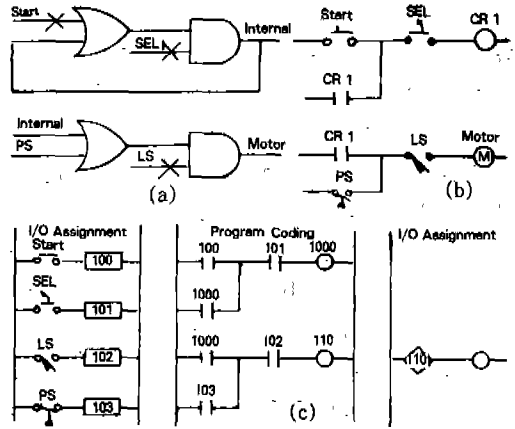
PLC에 있어서, 순시 타이머 접점은 일반적으로 이용가능치 않다. 순시 타이머 접점(타이머가 여자 될 때 바로 닫히거나 열리는 접점)을 수행하기 위해서 프로그래머는 타이머를 트랩하기 위해 내부 출력을 사용하고 타이머의 로직을 구동하기 위해서 뿐만 아니라 순시 접점으로써 내부 접점을 사용해야 한다. <그림 7-11>은 이러한 타이머 트랩 기법(어드레스 할당없이)을 설명한다.

<그림 7-10>에 보여주고 있는 회로의 또 다른 중요한 특징은 CR4(NC접점) 및 이전 장치의 전방향 통로로서 MCR로 웬스된 존재의 RUNG을 구동하면서 직렬로(라인7과 8번) 접속된 UP과 SET UP/RUN이다. 솔레노이드 SOL1은 MCR이 OFF 일 때라도 로직에 따라서 ON으로 바꿀 수 있다. 따라서 SOL1 출력 RUNG은 MCR 존 외부에서 (PLC 프로그램에서) 프로그램되어야 하고 또한 MCR 회로(LS1, LS2 등)에 포함된 로직부에 의해서 ON으로 바꿀 수 있어야 한다.

로직 RUNG이 너무 혼동스럽게 보일 때에 가장 행하기에 좋은 것은 상호간에 그것을 격리시켜 놓은 것이다. 그 다음에 회로의 좌측을 향하여 출력에서 출발하여 우측에서부터 모든 가능한 로직 통로를 재구성시키고 따라서 RUNG의 시초에서 끝낸다. 만일 RUNG의 어떤 부분이 전에 논의하였던 것처럼, 또 다른 RUNG에 직접 접속하거나 또는 상호작용을 갖



<그림 7-11> 내부출력(우측)을 사용한 TMR1(좌측)의 트랩핑



<그림 7-12> (a)로직 (b)릴레이 다이어그램 (c)코딩

는다면 문제를 피하기 위한 가장 좋은 방법은 2개의 RUNG이 교차하는 지점에서 내부 출력을 만드는 것이다. 그 다음에 나머지 로직을 구동하기 위해서 내부 출력을 사용한다. <그림 7-10>에 보여준 회로에 대해서, 이러한 교차점은 정상 닫힌 CR4 및 CR3(NC)와 라인 9에 있는 LS1(NO)간에 있을 것이다.

프로그램 코딩/번역

프로그램 코딩은 로직 또는 릴레이 다이어그램을 PLC 래더 프로그램 형태로 쓰는 과정이다. 이러한 래더 프로그램은 기계 또는 프로세스에 대한 제어를 수행하는 실제적인 로직이며 응용 메모리에 저장된다. 프로그램 코딩의 용이성은 이전 단계(할당 등)를 얼마나 정연하게 행하여 왔는가에 직접적으로 관계된다. PLC 래더 프로그램내에 있는 각 요소는 I/O 할당 다큐멘테이션에 따라서 이에 할당된 어드레스를 갖고 있다. <그림 7-12>는 로직 또는 릴레이 다이어그램으로 부터 만들어지는 샘플 프로그램을 보여준다(내부 코일 1000이 제어 릴레이를 대치한다).

코딩(Coding)이란, 이것이 새로운 응용에서 또는 현대화에서 부터 나왔던 간에 로직에 대한 PLC 표시법이다. <다음호에 계속...>