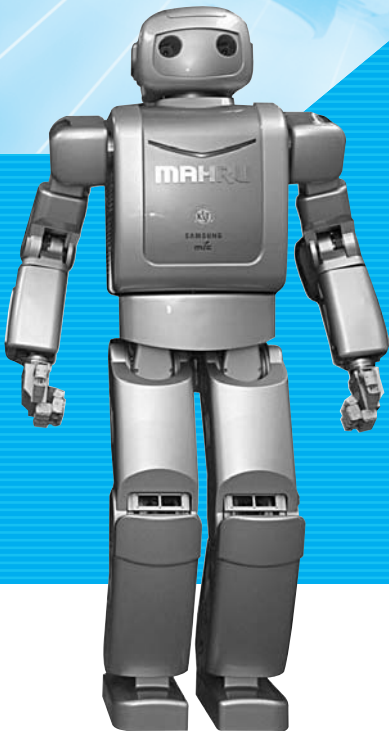


로봇 설계

고등학교

서울특별시교육감 인정

2007-000 | 2007.02 .



로봇 설계

이영욱·최순재·배희황·김송경·임덕수 저

서울특별시교육청

머 . 리 . 말 .

01010101010101010101010101010101

21세기 지식기반사회의 도래와 함께 지식의 효율적 창출과 활용은 국가와 기업의 경쟁력을 결정하는 핵심 요인이 되고 있다.

이에 따라 미래 정보화 산업의 핵심으로 떠오르고 있는 IT, BT, NT 산업과 함께 RT(Robot Technology)산업에 대한 큰 관심과 지원이 활발하게 이루어지고 있다. 특히 로봇산업은 기술의 정도에 따라 고부가가치가 매우 크며 기술집약적인 산업으로 21세기 지식 기반 사회에 대표적인 복합 기술이라 할 수 있다.

로봇은 전기·전자·기계·자동화·전산 등 복합기술이 총망 된 첨단기술이며 특히 로봇은 로봇의 기구·구조와 로봇의 동역학 및 마이크로프로세서와 모터의 제어, 센서 사용 등 다양한 기술이 접목된 첨단 기술 산업 이다.

로봇기술이 기초 과학 분야에서 응용 분야까지 그 범위가 다양한 복합기술이기 때문에 산 학 및 정부가 체계적이고 전략적으로 육성해 나가야 할 산업이라고 할 수 있다. 이에 따라 우리나라에서도 21세기 유망산업의 하나로 로봇산업을 선정하여 이를 중점 육성하고 있다.

이러한 로봇시대에 맞추어 로봇 제작의 기술력이 필요해 짐에 따라서 고등학교 과정에서 필요한 기초적인 로봇설계의 개념 및 방법에 대하여 단계적으로 기술 하였다.

이 교과서는 로봇설계의 개요, 기계·기구요소, 로봇기구 시스템, 로봇설계의 실제, 로봇 설계 실습모듈로 구성되어 있다.

- 1 로봇의 기초가 되는 지능형로봇, 산업용 로봇의 종류를 알고 로봇의 기구학 및 동역학에 대한 개념을 이해하여 로봇설계에 필요한 능력을 기르도록 한다.
- 2 기계 기구요소에서는 로봇의 구조를 이해하기 위한 기계요소 및 액추에이터를 알아보고 이를 활용할 수 도록한다.
- 3 로봇의 설계를 이해하기 위한 로봇 기구요소인 기계부품, 동력의 전달방법, 축과 축이음, 기구의 응용에 대하여 알아보고 이를 응용할 수 있도록 한다.
- 4 로봇설계의 실제에서는 설계계획부터 기계·기구 시스템 설계, 제어 센서부 설계, 프로세서 설계 운용 설계, 제작도면 설계 등 실제 로봇을 설계할 수 있도록 한다.
- 5 로봇설계 실습 모듈에서는 지금까지 배운 내용을 기초로 하여 스텝핑 장치, 엠보싱 장치, 라인트레이서를 제작할 수 있도록 한다.

끝으로 로봇설계에 관심을 가지고 공부하고자 하는 사람은 로봇을 움직이고 제어할 수 있는 분야에 좀더 많은 관심을 가지고 공부를 하였으면 하고 우리나라를 미래에 세계에서 초일류 로봇산업 국가로 발전시켜 나가야 할 것이다.



contents

I. 로봇 설계의 개요

01 로봇의 개념

1. 로봇의 개념 / 10
2. 지능형 로봇 / 11
3. 산업용 로봇 / 18
4. 국내 로봇 산업 현황 / 25

02 로봇의 기구학

1. 로봇 기구학의 개념 / 30
2. 순기구학 / 31
3. 역기구학 / 31
4. 동기기구학 / 32

03 로봇 동역학의 기초

1. 로봇 동역학의 개념 / 34
2. 동역학(Dynamics) / 35

II. 기계·기구 요소

01 기계 요소

1. 개요 / 40
2. 완전기구 / 41
3. 이동기구 / 42

02 액추에이터

1. 개요 / 46
2. 분류와 특징 / 47
3. 액추에이터 제어기 설계 / 48

III. 로봇 기구 시스템

01 로봇의 기구 요소

1. 부품의 체결 / 80
2. 동력의 전달 / 88
3. 베어링 / 101
4. 축과 축이음 / 110
5. 기구 응용 / 115

IV. 로봇 설계의 실제

01 설계 계획

1. 로봇의 기능 및 특성 / 134
2. 설계 특성 / 136

02 기계·기구 시스템 설계

1. 기계·기구 시스템의 기능 및 구동 특성 / 140
2. 기계·기구 시스템 설계 / 146

03 센서

1. 개요 / 150
2. 센서의 분류 / 151
3. 접촉형 스위치 / 152
4. 비접촉형 스위치 / 156



04 제어 센서부 설계

1. 센서부 설계 / 161
2. 모터부 설계 / 164
3. 인터페이스 설계 / 166

05 프로세서 설계

1. 마이크로프로세서의 기능 및 특성 / 168
2. CPU 동작과 로봇 제어 / 171

06 운용 시스템 설계

1. C언어와 컴파일러 / 177
2. 순서도 / 182
3. 프로그래밍 설계 / 184

07 제작도면 작성

1. 기계·기구부 / 197
2. 센서·제어부 / 199
3. 전자회로도 / 199
4. 주기판(Main Board) 구성도 / 202

V. 로봇 설계 실습 모듈

01 전기 공압제어 방식에 의한 스탬핑 장치 제어 설계

1. 전기 공압제어의 제어회로 설계 / 208

02 엠보싱 작업 제어 설계

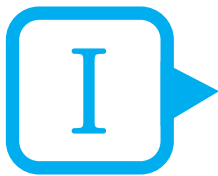
1. PLC 제어의 제어회로 설계 / 220

03 마이크로 로봇 CPU 89C5

1. 89C51 / 225
2. 마이크로 로봇의 기본회로 / 231

04 산업용 다축 로봇 제어 설계

1. 산업용 로봇의 형태별 분석 및 사용 예 / 276



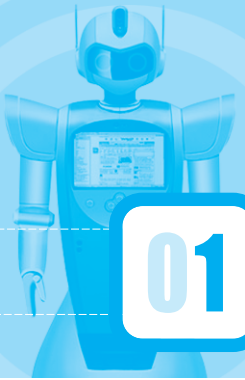
로봇 설계의 개요

로봇을 배울 때 가장 기본적으로 지능형 로봇과 산업용 로봇을 예로 든다. 이 단원에서는 지능형 로봇과 산업용 로봇을 분류하고 로봇을 움직이는데 필요한 기구학과 동역학에 대하여 알아보기로 한다.



학습목표

1. 로봇의 개념에 대하여 설명할 수 있다.
2. 로봇 기구학의 개념에 대하여 설명할 수 있다.
3. 로봇 동역학에 대하여 설명할 수 있다.



Robot Robot

01

로봇의 개념

1 로봇의 개념

로봇이란 슬라브(체코)어로 노동자 즉, 'Worker'란 뜻이다. 로봇이란 말은 1922년 체코인 극작가 카렐 차페크(Karel Capek)이라는 사람이 발표한 희극 <로섬의 유니버설 로봇(Rossum's Universal Robots)>에서 '로봇'이란 말을 사용하였다.

로봇은 프로그램된 다양한 움직임에 따라 여러 가지 일들을 수행하는 조작기라고 표현되며, 결국 '로봇은 인간이 할 일을 대신해 주는, 인간에 도움을 주는 친구'로 말할 수 있다. 최근에는 미래의 로봇이 인간을 지배한다는 내용의 공상과학영화도 많이 나오고 있지만, 로봇의 궁극적인 목적은 인간의 행복을 위해 존재하는 것이다.

공상과학소설 작가 아이작 아시모프는 소설에서 로봇 특성의 기본적인 3가지 법칙을 설정하였다.

첫째, 로봇은 인간을 다치게 하거나 위협에 빠뜨려서는 안 된다.

둘째, 로봇은 첫째 원칙에 위배되지 않는 한 인간의 명령에 복종해야 한다.

셋째, 로봇은 둘째 원칙에 위배되지 않는 한 자신의 존재를 보호해야 한다.

첫째와 둘째 법칙은 인간을 보호하기 위한 법칙인 반면 셋째 법칙은 로봇을 보호하기 위한 법칙이다.

최근에는 로봇의 개념이 확대되어 거의 모든 자동화된 기계적 전기적 시스템



에 적용되고 있을 뿐만 아니라, 점차적으로 인간과 직접적으로 교감하는 방향으로 바뀌고 있다.

하나의 로봇을 만들려면 로봇의 용도와 목적에 맞게 설계하고 해석하는 것이 필요하다. 기구학 및 동역학은 역학적인 로봇의 구조를 해결한다. 로봇의 몸체는 어떤 재료를 사용하여 만들 것인가를 고려한다. 일반적으로 최근에 활발히 연구되고 있는 의공학 분야의 소형로봇은 인간의 몸 속에서 작동하므로 가볍고 인체에 해를 끼치지 않는 재질로 만든다. 로봇이 구동하기 위해서는 제어기를 구성해야 한다. 또한 로봇을 제어하기 위한 제어 알고리즘이 필요하다.

로봇에 인간과 비슷한 지적 능력을 부여하기 위해서는 센서를 설치해야 한다. 다양한 센서융합을 통해 인간과 비슷하게 느낄 수 있게 된다. 이러한 센서 정보를 통해 얻은 데이터로부터 로봇이 판단하고 행동하는 것은 다양한 알고리즘의 프로그램을 통해 할 수 있다. 이러한 프로그램에 인공지능 알고리즘을 적용하여 인간이 가지는 판단 능력이나 학습 능력의 지능을 부여한다.

알고리즘이란 컴퓨터가 어떤 연산을 할 수 있도록 하는 프로그램의 한 집합이다.

2 지능형 로봇

1 지능형 로봇의 정의

지능형 로봇이란 인간과의 상호작용을 통하여 인간의 명령 및 감정을 이해하고 반응하며, 최첨단 기술을 바탕으로 인간에게 다양한 서비스를 제공하는 로봇이다.

지능형 로봇 기술은 주변 환경의 학습, 추론 및 이해와 인간과 로봇 간 자연스러운 상호 작용을 통해 독립적으로 인간을 보조하는 로봇의 핵심 기술이다. 따라서 로봇에는 사용 환경과 작업 능력에 따라 레벨은 다르지만 어느 정도의 지능을 반드시 가지고 있다고 볼 수 있다. 현재의 지능형 로봇은 지금 개발되고

있거나 사용 중인 전체 로봇들 중에 지능의 레벨이 상당히 높은 첨단 로봇들을 포괄적으로 의미한다고 볼 수 있다. 즉, 지능형 로봇은 기존의 제조업에 사용되어 온 반복적인 작업을 수행하는 로봇과는 달리 현재의 발달된 기술로 연구개발이 가능하며, 현재와 가까운 미래의 시장에서 요구하는 기능과 성능을 가지는 로봇을 의미한다. 기존의 제조업용 로봇은 그 로봇이 작업을 수행하는 데 불편하지 않도록 잘 만들어진 공장 환경에서 주로 반복적인 작업을 하는 것이 대부분이다. 그런데 현재 시장에서 요구하는 로봇의 중요한 기능은 이런 공장 환경을 떠나 사람이 거주하는 공간인 집, 사무실, 병원 등과 사람이 접근하기 어려운 현장인 관로, 고압선로, 화재지역, Micronano world 등으로의 진출이다. 바로 이것이 기존의 로봇과 다른 한 차원 높은 지능을 요구하는 이유이다.

2 지능형 로봇의 분류

지능형 로봇은 IFR(International Federation Robotics:국제로봇연맹)에서 다음과 같이 분류하고 있다

구분	대분류	중분류	종류
지능형 로봇	서비스로봇	개인용 로봇	애완용 로봇, 청소로봇, 경비로봇, 교사로봇 등
	전문가용 로봇	공공서비스용 로봇	의료로봇, 안내로봇 등
		극한 작업용 로봇	재난구조로봇, 원전로봇 등
	산업용(제조용) 로봇		용접로봇, 핸들링 로봇, 도장로봇 등

〈표 I-1〉 지능형 로봇의 분류

구분	서비스로봇		제조업용	네트워크 기반
	개인서비스용	전문서비스용		
종류	· 가사용(청소, 심부름, 경비, 조리 등)	· 빌딩 서비스용(경비, 안내, 배달, 청소)	· 자동차 제조용	· 정보컨텐츠서비스용 · 공공도우미(공공업무 지원용)
	· 생활지원용(간병, 장애 자보조, 재활훈련 등)	· 공공작업용(재난극복 · 인명구조)	· 전자제품 제조용	
			· 디스플레이 제조용	
			· 반도체 제조용	

IFR에서 로봇은 고정 또는 움직이는 것으로서 산업자동화 분야에 사용되며 자동 제어되고 재 프로그램이 가능한 다목적인 3축 또는 그 이상의 축을 가진 자동 조정 장치를 말한다.



· 여가지원용(오락,애완, 게임, 헬스케어 등)	· 의료 · 복지 · 사회인프라용(고소,활선,관로) · 국방용 · 농업/임업/축산업/수산업 · 광업/건설 · 우주/해양/원자력	· 바이오 · 신약제조용 · 물류자동화용	· 임베디드 로봇(건물, 교통수단 등 생활환경에 내장) · 소프트웨어 로봇
----------------------------	---	---------------------------	--

〈표 I-2〉 지능형 로봇의 응용분야별 종류

1) 개인용 로봇

개인용 로봇은 개인 생활의 본질인 운동성을 보조하며, 지능을 갖는 인간 공존형 대인 지원 로봇이다. 홈 오토메이션(home automation)을 지원하는 가정용, 고령자와 신체장애자 등에 대한 생활지원용, 오락용(entertainment) 등으로 분류할 수 있다.



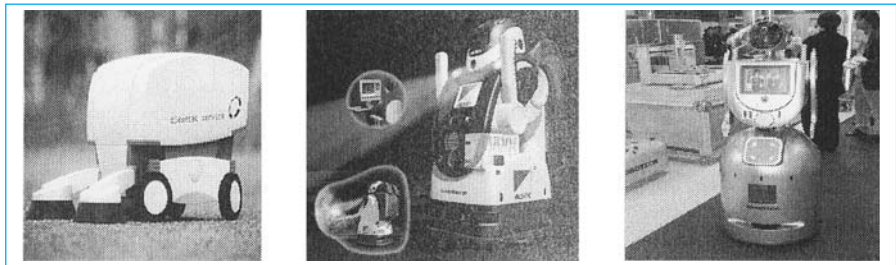
〈그림 I-1〉 개인용 로봇

2) 공공용 로봇

원격조정 및 센서(전자감응) 장치가 개량되면서 금융기관의 현금 수송이나 주택 경비에 로봇을 활용하는 기술이 잇따라 개발되고 있다. 경비용 로봇은 불황 속에서 범죄율 증가로 중저가형 원격경비에 대한 수요가 증가되고, 또한 공공

개인용 로봇은 가사 로봇, 생활 도우미 로봇, 애완용 로봇, 서빙 로봇, 교육용 로봇, 퍼포먼스 로봇, 경기용 로봇 등이 있다.

기관이나 박물관에서 방문객을 대상으로 안내를 담당하는 로봇이 생겨나고 있다. 자유로운 이동성과 지능을 가진 로봇을 만들기 위해 많은 기술이 필요하지만, 현재 개발된 안내 로봇들도 방문객들에게 큰 호응을 얻고 있으며 지속적인 성장이 기대되는 분야이다.



〈그림 I-2〉 여러 가지 공공용 로봇

3) 현장 로봇

① 의료 및 복지 로봇

국민에게 선진화된 의료 복지를 제공하기 위하여 기존의 로봇 기술을 활용하여 일상생활에서 국민의 건강을 실시간 관리하고, 건강에 문제가 있을 때 보다 정확한 진단을 내릴 수 있도록 도와주거나 환자를 보살피는 범주의 로봇 시스템을 말한다. 의료복지 로봇 시스템의 개발은 인간의 삶의 질을 향상시킴으로써 21세기를 더 살기 좋은 사회로 만들어 나갈 것이다.



da Vinci 사용 장면

외과의사 조작부

수술용 작업부

〈그림 I-3〉 의료용 로봇



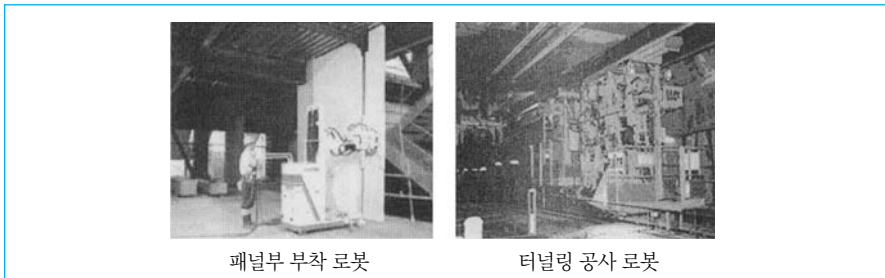
② 농업 · 건설용 로봇

생물 산업, 건설 시공에서의 로봇기술 연구개발의 목표는 작업 생산성, 작업 쾌적성, 안전성, 환경 보존성 향상을 목표로 한 인간, 컴퓨터, 기계의 복합시스템(로봇 시스템), 비구조적 Field 작업과 가변적 작업 공정의 생력화와 환경변화에 대한 적응, 유해 위험 작업 대처 시스템 및 기술 개발이다.



토마토 수확작업 로봇(일본) 자율운행 트랙터(한국)

〈그림 I-4〉 여러 가지 농업용 로봇



패널부 부착 로봇

터널링 공사 로봇

〈그림 I-5〉 여러 가지 건설용 로봇

③ 군사용 로봇(Military Robot)

미국 대통령 산하 의회 정보통신 분과 위원회에서는 지능형 로봇 기술을 국가 안보에 중요한 영향을 끼치는 분야로 분류하여 집중 육성을 하고 있다.

미군은 향후 전쟁은 전투 로봇으로 치루어질 것을 예상하여 이에 대한 연구를 대대적으로 지원하고 있다. 따라서, 우리도 특화된 형태의 군사용 로봇 분야 육성 및 21세기형 디지털 최강 국군을 위하여, 여러 첨단 기술을 종합하여 개발



할 수 있는 군사 로봇의 연구가 절실히 요구된다.



정찰로봇

〈그림 I-6〉 군사용 로봇

4 극한 환경용 로봇

극한 환경용 로봇은 방사능이 노출된 곳, 위험한 폭발물 등의 취급, 고온, 고압, 화재 등과 같은 극한 환경에서 인간을 대신하여 작업을 하는 로봇이다. 〈그림 I-7〉(a)는 원자력 발전소의 방사능 환경에서 작업을 수행할 수 있는 로봇이다.



〈그림 I-7〉 극한 환경용 로봇

5 재활 로봇

노인과 장애인의 재활치료와 일상생활을 도와주는 로봇 시스템으로서, 단순히 신체를 지지해주는 역할을 벗어나 중증환자나 노인의 신체적 움직임을 향상시키고 빠른 시일 내에 회복하도록 중요기능을 수행하며, 장애인의 장애를 극복하여 독립적인 생활이 가능하도록 도와주는 로봇이다.



〈그림 I-8〉 환자 도우미 로봇

6 탐사 로봇

탐사 로봇은 우주, 혹은 깊은 대양과 같이 위험한 곳으로 보내져 그곳의 정보를 모아 전송하는 작업을 수행하도록 하는 로봇이다. 고도의 지능을 가진 로봇은 주로 외계 탐사용으로 개발된다.



NASA 화성 탐사 로봇

〈그림 I-9〉 탐사 로봇



3 산업용 로봇

1 산업용 로봇의 정의

산업용 로봇은 1960년대 초부터 산업 현장에서 사용되고 있는 로봇으로, 대부분 인간의 팔에 해당되는 머니플레이터(manipulator) 형태이다. 즉, 산업용 로봇은, 인간의 모습과는 달리 일반적으로 하나의 팔을 가지고 있으며, 작업에 알맞도록 고안된 도구를 손에 부착하고, 제어 장치에 내장된 프로그램의 순서에 따라 작업을 수행한다.

〈그림 I-10〉은 대표적인 산업용 로봇을 나타낸 것이다.



조립용 로봇

〈그림 I-10〉 각종 산업용 로봇

산업용 로봇은 로봇 관련 단체 또는 제작사에 따라 다르게 정의되고 있다. 그 중 미국 로봇협회(Robotic Industries Association, RIA)에 의한 정의가 일반적으로 널리 인용되고 있다.

“로봇은 여러 가지 작업을 수행하기 위하여 자재, 부품, 공구 또는 특별한 장치들을 프로그램된 대로 운반하도록 설계된, 재프로그램이 가능하고 다기능을 가진 머니플레이터이다.”

산업용 로봇은 인간의 노동력을 대신함으로써 인건비를 감소시키며, 작업의 정밀도와 생산성을 향상시켜 준다. 또한, 산업용 로봇은 생산 공정의 유연성을



증대시키며, 인간을 대신하여 반복적이고 위험한 작업을 수행함으로써 인간의 작업 환경을 개선하여 준다.

2 산업용 로봇의 분류

산업용 로봇은 기준에 따라 여러 가지로 분류될 수 있다. 산업 현장에서 많이 사용되고 있는 산업용 로봇은 일반적 분류, 제어 방식, 또는 구조적 동작 특성에 따라 분류한다.

1) 일반적 분류

〈표 I-3〉은 산업용 로봇의 일반적 분류로서, 로봇이 보유하고 있는 기능에 따라 분류한 것이다.

구분	정의
의수동 로봇(manual robot)	사람의 조작에 의하여 움직이는 로봇
시퀀스 로봇(sequence robot)	미리 설정된 순서와 조건 및 위치에 따라 동작의 각 단계를 차례로 수행하는 로봇
플레이백 로봇 (playback robot)	사람이 로봇을 직접 조작하여 작업 순서, 작업 위치 및 작업 조건 등의 정보를 교시하고 기억시켜 두었다가 이를 재생함으로써 교시된 작업을 되풀이할 수 있는 로봇
수치 제어 로봇 (numerically controlled robot)	작업 순서, 위치 및 조건 등의 정보를 수치 또는 언어 등으로 표현하여 지시하면, 그 정보에 따라 작업을 수행하는 로봇
지능 로봇 (intelligent robot)	감각 및 인식 기능에 의하여 작업 수행에서 요구되는 행동을 스스로가 결정할 수 있는 기능을 가진 로봇
감각 제어 로봇 (sensory controlled robot)	감지기에 의한 감각 정보를 사용하여 스스로 동작을 제어하는 기능을 가진 로봇
적응 제어 로봇(adaptive controlled robot)	제어 특성 및 조건을 작업 환경의 변화에 따라 스스로 변화시킬 수 있는 기능을 가진 로봇
학습 제어 로봇 (learning controlled robot)	작업 경험을 반영하여 다음 작업에서 요구되는 제어 특성 및 조건을 습득해 가는 기능을 가진 로봇

〈표 I-3〉 산업용 로봇의 일반적 분류

2) 제어 방식에 따른 분류

〈표 I-4〉는 산업용 로봇의 제어 방식에 따른 분류로서, 산업용 로봇의 동작을 제어하는 방식에 따른 것이다.

구분	정의
비서보 제어 로봇 (non-servo controlled robot)	로봇의 동작을 공압 실린더와 같은 비서보 장치를 사용하여 수행하는 로봇으로, 정교한 위치 또는 방향 등의 제어를 수행하지 않는 로봇
서보 제어 로봇 (servo controlled robot)	정교한 위치 또는 방향을 제어하기 위하여 서보모터와 같은 서보 제어 장치를 사용하여 로봇의 동작을 제어하는 로봇
위치 결정 제어(PTP) 로봇(point-to-point controlled robot)	공간상에 정해진 위치로부터 다음 위치로 이동하는 동안 로봇의 이동 경로에 관계없이 지정된 작업점만을 찾아가도록 제어되는 로봇
경로 결정 제어(CP) 로봇 (continuous path controlled robot)	공간상에 정해진 위치로부터 다음 위치로 이동하는 동안, 로봇의 이동 경로가 경로 보간에 의하여 생성되고, 이를 추종함으로써 로봇의 이동 경로가 제어되는 로봇

〈표 I-4〉 산업용 로봇의 제어 방식에 따른 분류

3) 구조적 동작 특성에 따른 분류

산업용 로봇을 로봇의 구조적 동작 특성에 따라 분류한 것은 로봇의 설계 또는 응용에서 가장 많이 사용하는 분류 방법이다.

① 직교좌표 로봇

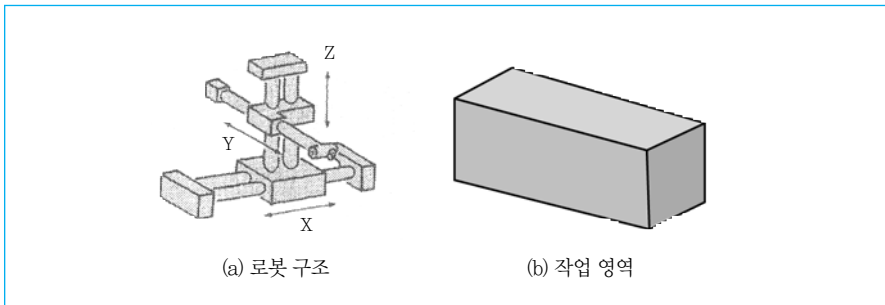
일명 XY 로봇이라고도 불리는 직교좌표 로봇(Cartesian coordinate robot)은 〈그림 I-11〉에서와 같이 직선 운동을 수행하는 축으로만 구성되어 있는 로봇이다.

일반적으로, X, Y, Z의 3축으로 구성되며, 경우에 따라서는 1개 또는 2개의 축만을 사용하기도 한다. 또, 잡은 물체의 방향을 바꿀 수 있도록 하기 위하여, 로봇의 선단에 회전축을 붙여 4축으로 구성한 것도 있다.

직교좌표 로봇은 기계적으로 튼튼하고 안정적이며 위치 정밀도가 우수하다.



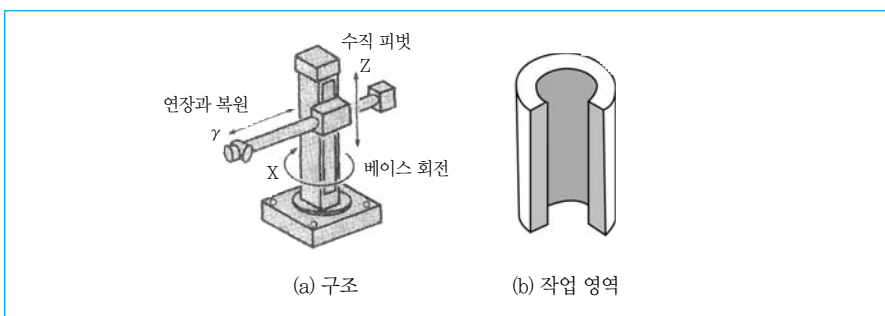
또한, 인간에게 가장 익숙한 직교좌표계를 사용함으로써 이용자가 쉽게 사용할 수 있다는 장점을 가지고 있다. 그러나 동작 영역에 비하여 설치 면적이 큰 것이 단점이다.



〈그림 I-11〉 직교좌표 로봇

② 원통좌표 로봇

원통좌표 로봇(cylindrical coordinate robot)은, 〈그림 I-12〉에서와 같이 원통의 길이와 반지름 방향으로 움직이는 2개의 직선축과 원주 방향으로 움직이는 하나의 회전축으로 구성된 로봇이다. 원통좌표 로봇은 작업 영역이 넓고, 설치 면적도 작으며, 위치 정밀도도 우수하다. 그러나 로봇의 엔드 이펙터(end effector)에 과중한 하중이 걸리는 작업이나 일감의 중량이 크면, 위치 정밀도가 떨어지는 단점을 가지고 있다.



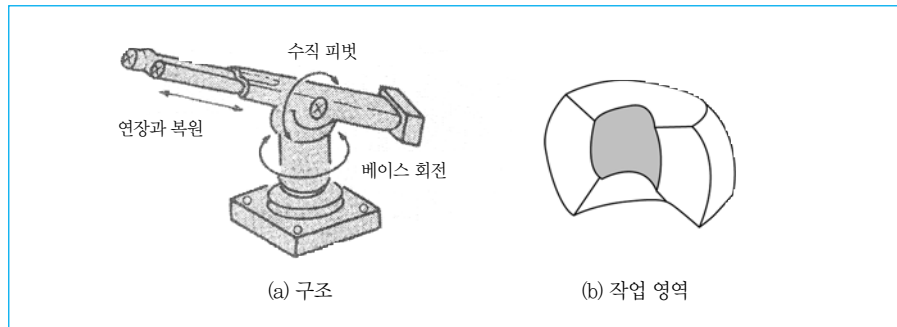
〈그림 I-12〉 원통좌표 로봇



3 극좌표 로봇

극좌표 로봇(polar coordinate robot)은 1개의 직선축과 2개의 회전축을, <그림 I-13>에서와 같이 조합하여 구성한 로봇이다.

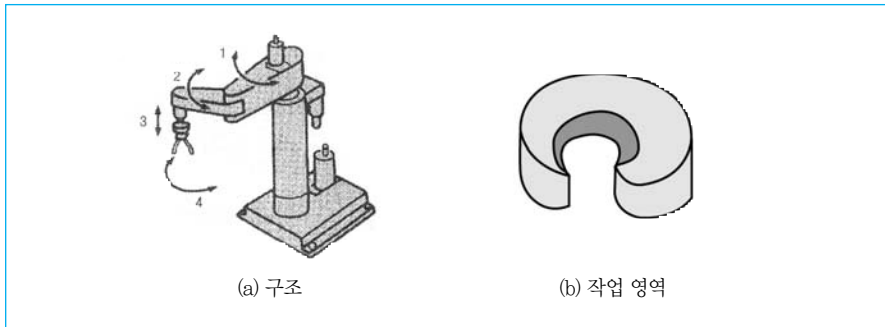
극좌표 로봇은 수직면에 대하여 상하 운동 특성이 우수하여 작업 영역이 넓고, 경사진 위치에서 작업을 수행할 수 있으므로, 용접 작업이나 도장 작업에 적합하다.



<그림 I-13> 극좌표 로봇

4 수평 다관절 로봇

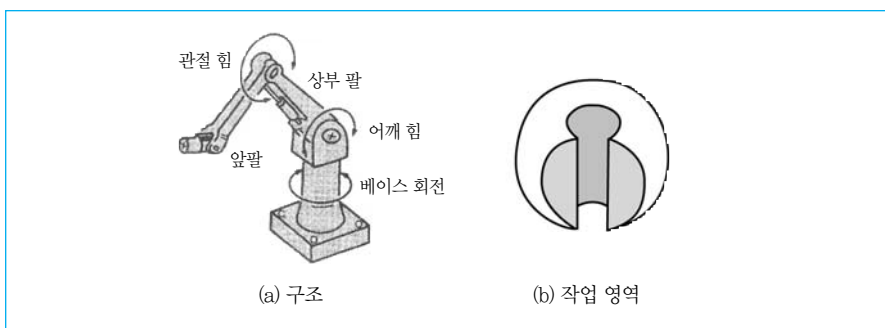
스카라 로봇(selective compliance assembly robot arm, SCARA)이라고도 불리는 수평 다관절 로봇(horizontal articulated robot)은, <그림 I-14>에서와 같이 3개의 회전축과 하나의 직선축으로 구성된 다관절 구조의 로봇이다. 1, 2축은 수평 회전 운동을 하고, 3축은 수직 상하 운동을 하며, 4축은 다시 회전 운동을 수행한다. 스카라 로봇은 수평면상의 운동 특성을 가지고 있으므로, 조립 작업 또는 팰리타이징(palletizing) 작업에 적합하다.



〈그림 I-14〉 수평 다관절 로봇

5 수직 다관절 로봇

용접 및 도장 작업 등에 가장 많이 사용되고 있는 수직 다관절 로봇(vertical articulated robot)은, 〈그림 I-15〉에서와 같이 관절의 구동이 모두 회전축에 의하여 수행되는 구조를 가지는 로봇이다. 일반적으로 6개의 회전 관절을 가지고 있으며, 작업 공간 내에 위치하는 임의의 작업 지점에 임의의 방향으로 접근이 가능하다. 회전 관절만으로 구성되어 있어 제어와 사용이 어렵지만, 속도가 빠르고 공간 활용 능력이 우수하며, 설치 면적과 본체의 체적이 작다는 장점을 가지고 있다.

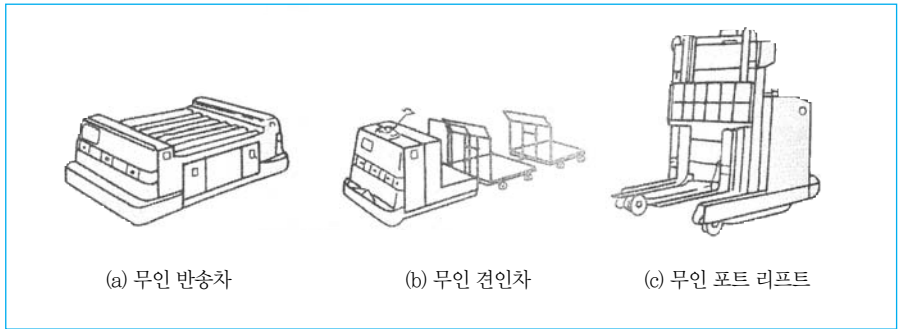


〈그림 I-15〉 수직 다관절 로봇

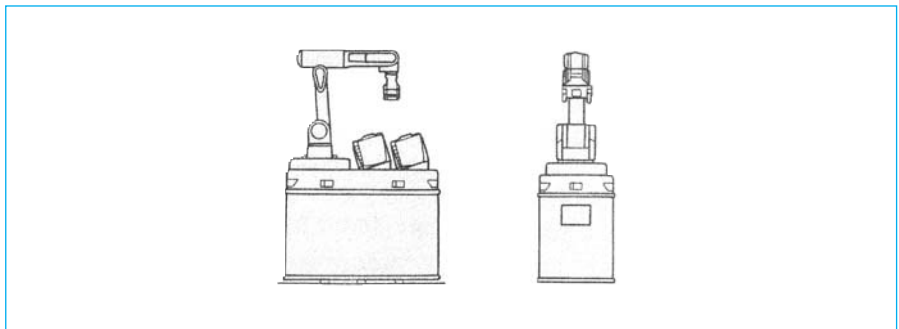


6 이동 로봇

이동 로봇은 머니플레이터와 다르게 바퀴 또는 그 밖의 구동 장치에 의하여 이동할 수 있는 기능을 가진 로봇을 총칭하는 용어이다. 산업 현장에서 사용되고 있는 대표적인 이동 로봇은, <그림 I-16>에서와 같이 물품을 적재하고, 지정된 경로를 따라서 이동하는 자율 주행 차량이고, <그림 I-17>은 반도체 생산에서 사용되는 머니플레이터를 탑재한 이동 로봇이다.



<그림 I-16> 자율 주행 차량의 종류



<그림 I-17> 머니플레이터 탑재형 이동 로봇



4 국내 로봇 산업 현황

(1) 지능형 로봇 산업의 중요성

지능형 로봇은 소득 2만 달러 시대를 선도할 미래 STAR 산업이다. 타 분야에 대한 기술적 파급효과가 큰 첨단 기술의 복합체이며, 삶의 질 향상에 따른 사회의 안정화 수단이 될 수 있다.

1) 산업적 측면

세계 각국은 지능형 로봇을 전략산업으로 육성하고 있으며, 자국의 기술력을 과시하는 데 사용하고 있다. 지능형 로봇 산업은 현재 형성 단계로 시장 및 기술 선점의 가능성이 매우 높은 산업이다. 일본에서는 아시모, 2003년에는 큐리오를 개발하였고, 우리나라에서는 2004년에 휴보 로봇, 마루 로봇 등 휴머노이드 로봇을 개발하였다.

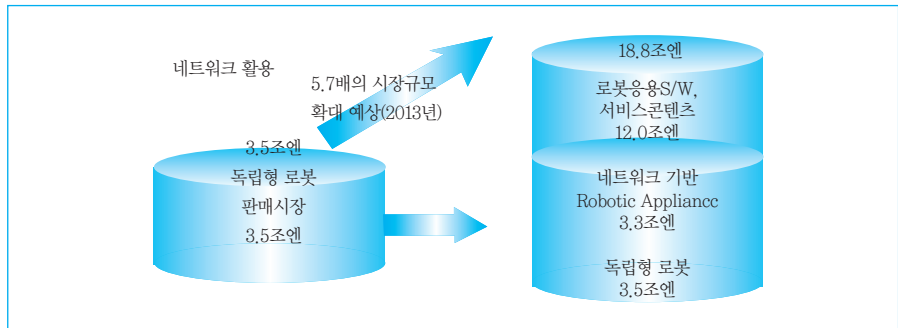
국내 로봇 산업은 2020년경에는 1가구 1로봇 시대가 도래할 것으로 예상하고 있으며, 국내시장 규모로 100조 원을 달성할 것으로 예측하고 있어, 기술 확산을 통해 연관된 전후방 산업 분야를 이끌 첨단 기술의 혁신을 촉진하는 고부가가치 산업이 될 것이다.

구분		예상 시장규모	
		2010년	2020년
세계 시장규모(억불)		1,500	5,000
한국 시장규모 (억원)	제조업용 로봇	40,000	400,000
	서비스 로봇	60,000	600,000
	계	100,000	1,000,000

〈표 I-5〉 세계 및 국내 로봇시장 전망

지능형 로봇에 대한 급속한 수요의 증가는 거대한 신규시장 창출과 차세대의 중요한 산업으로 부상하며 지식사회의 출현, 고령화 사회의 도래, 삶의 질 향상의 욕구에 따른 새로운 제조 방식 및 서비스의 수요 증가를 가져올 것이다. 또한, 최근 IT산업의 급속한 발전으로 인한 네트워크와 연계된 서비스 로봇이 지능형 로봇 산업의 새로운 한 축으로 등장하며 시장창출을 가속화할 전망이다.

일본의 경우 로봇에 네트워크를 활용함으로써 기존의 독립된 로봇시장 대비 시장규모가 5.7배 확대될 것으로 예상하고 있으며, 2013년에는 19.8조 엔으로 확대될 전망이다.



〈그림 I-18〉 일본 총무성 -네트워크 로봇의 실현을 향해서(2003)

우리나라의 경우 네트워크 로봇 시장규모는 응용 서비스 시장을 포함하여 CAGR 600% 성장하여 2010년에는 18,000억 원 규모의 시장을 형성할 것으로 전망하고 있다.

구분	2005	2006	2007	2008	2009	2010	CAGR
시장규모(억원)	0.963	90	675	1,695	3,195	18,000	600%

〈표 I-6〉 네트워크 로봇(URC) 국내 시장 전망

2) 기술적 측면

지능형 로봇 기술은 첨단 신기술 분야의 복합체라 할수 있다. 메카트로닉스,

CAGR (Compound Annual Growth Rate)은 “연평균 복합 성장률”을 말한다. 즉, 수년간의 성장률을 평균으로 환산한 것인데 매년의 성장률을 단순 평균한 것이 아니라, ‘첫 회부터 매년 일정한 성장률을 지속한다고 환산할 때의 성장률’을 의미한다.



인공지능, 컴퓨터, 첨단 IT기술 등이 융합되어 산업 간 시너지 효과를 창출한다. 따라서 지능형 로봇 시장은 컴퓨터나 자동차 시장보다도 더욱 활성화 되어 고부가가치를 창출할 것으로 기대된다.

로봇의 고유 기술인 액추에이터, 인간 환경 인식 등은 관련 분야의 기술 발전에 큰 영향을 미침으로써 그 분야의 산업이 한층 발전할 것이다.

한편 유비쿼터스 네트워크의 실현으로, 지능형 로봇은 확장된 정보 공간에서 서비스를 제공하는 수단으로 발전할 것이다. 또한 정보 공간과 실세계 연동의 인터페이스의 실현으로 로봇 사회의 진입이 가속화할 것이다.

3) 사회적 측면

우리나라는 이미 고령화 사회로 진입하여 65세 이상이 7.5%를 넘었고, 2020년경에는 노인 부양비율이 약 20%에 달하게 되어 노인복지용 서비스 로봇의 수요가 급증할 것으로 예측된다. 출산율 감소 및 고령화 사회 진입에 따른 노동력을 대체할 수 있는 미래 산업으로 기대된다.

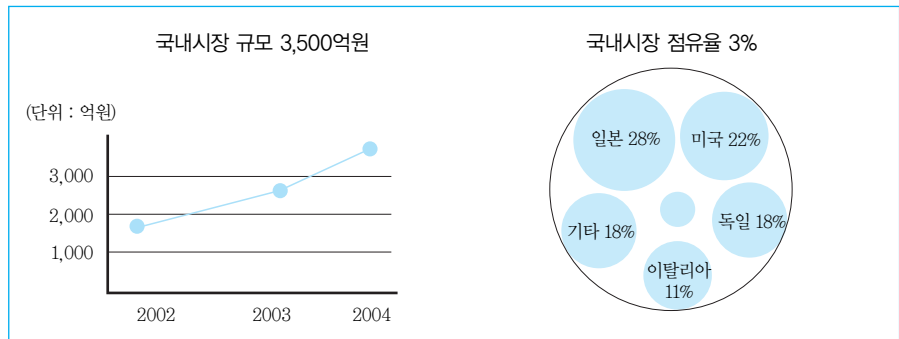
따라서 고령화 사회의 도래 및 삶의 질 향상의 욕구에 따라 지능형 로봇의 급속한 수요증가로 신규 시장과 일자리 창출로 지속적인 경제 성장이 가능하다.

뿐만 아니라 IT기술의 발전에 따른 원격로봇 오피스 등의 출현으로 재택근무, 방법 및 경비 시장의 확대 등 사회 전반에 걸친 직업군의 변화를 이룰 것이다.

(2) 우리나라의 로봇 산업의 현황

우리나라의 로봇은 2000년부터 매년 10%씩의 고도성장을 거듭하는 차세대 고부가가치 산업이다. 또한 로봇 산업의 내수 시장의 규모는 3,500억 원으로, 2004년 기준으로 세계 6위 수준이며 2020년쯤에는 시장규모가 1조 4천억 달러로 BT(생명공학)산업을 추월할 것으로 예상하고 있다.

중소벤처기업을 포함한 대부분의 로봇 생산업체의 매출 가운데 약 90%가 산업용 로봇이지만, 2015년 이후에는 개인용 로봇(Personal Robot)이 로봇시장의 주류가 될 것으로 전망한다. 차후 애완 로봇, 청소 로봇 등을 중심으로 시장을 확대될 것으로 보며, 2010년경 본격적인 서비스 로봇이 보급되면서 시장규모가 1500억 달러로 성장할 것으로 내다보고 있다. 2020년경에는 개인용 로봇이 각 가정에 필수적인 존재가 되리라고 전망된다.



〈그림 I-19〉 시장 규모 및 점유율

IFR 보고서에 의하면 한국은 로봇의 시장 규모가 6위이며 사용대수는 5위고 로봇 밀도는 3위로서 세계적인 수준이다.

순위	시장규모[M\$]	산업용로봇 설치대수[대]	로봇 사용대수[대]	로봇밀도[대/10,000명]
1	일본 : 1,134	일본 : 31,588	일본 : 348,734	일본 : 322
2	미국 : 889	독일 : 13,381	독일 : 112,693	독일 : 148
3	독일 : 754	미국 : 12,693	미국 : 112,390	한국 : 138
4	이탈리아 : 444	이탈리아 : 5,198	이탈리아 : 50,043	이탈리아 : 116
5	프랑스 : 176	한국 : 4,660	한국 : 47,845	스웨덴 : 99
6	한국 : 123	프랑스 : 3,117	프랑스 : 26,137	핀란드 : 78

〈표 I-7〉 국내 로봇산업 현황

연도별 수출입 현황은 2003년 기준으로 로봇의 수출은 609억원이고, 수입은

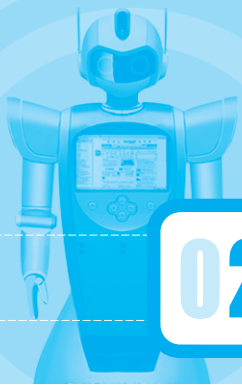


1,102억원 규모로 매년 수입의 비중이 증가하고 있다.

구분	'99년	'00년	'01년	'02년	'03년
수출	113	137	78	288	609
수입	359	681	580	554	1,102
국내생산	825	1,114	706	794	969
국내시장규모	1,071	1,658	1,208	1,060	1,462
수입비중(%)	33.5	41.1	48.0	52.3	62.8

〈표 I-8〉 연도별 수출입 현황

세계적으로 제조업용 로봇의 시장 성장률은 둔화되고 있으며, 향후 서비스 로봇 분야가 활성화 될 것으로 예상하고 있다. 정부는 지능형 로봇 산업을 제3세대 이동통신, 디지털TV 등과 함께 차세대 성장 동력산업으로 선정하였다.



1 로봇 기구학의 개념

로봇 기구학에서는 운동 그 자체만을 취급하며 운동을 일으키는 힘은 고려하지 않는다. 로봇이 움직이면 로봇의 팔 끝이 어디에 있는가를 알아야 한다. 로봇 팔의 끝을 나타내려면, 원점에서 얼마나 떨어져 있는가를 나타내는 위치와 얼마만큼 회전되어 있는가를 나타내는 자세로 정의 할 수 있다. 이를 기구학(kinematics)이라고 한다.

즉, 로봇 설계 시 고려되는 링크의 위치, 속도 및 가속도의 관계를 다루는 것을 말한다.

일반적으로 로봇 팔 끝의 위치는 로봇 각 관절이 이루는 각과 각 링크의 길이에 의해 결정된다. 로봇 기구학은 공간에서 로봇의 움직임을 시간 함수로 나타낸다.

특히 로봇 조인트 공간 변수들과 직교 공간 변수들 사이의 관계를 나타내며, 로봇의 모든 링크 움직임을 결정하는 독립변수의 수로 나타내는 것을 자유도라 한다.

로봇 기구학에는 구하고자 하는 변수가 무엇인가에 따라 순기구학과 역기구학이 있다.



2 순기구학

순기구학은 로봇의 각 조인트 공간에서 각이 주어졌을 경우 직교공간에서 로봇 위치를 나타내는 것이다.

(1) 단순한 로봇의 순기구학

순기구학의 목적은 직교 공간에서 로봇 위치와 오리엔테이션(Orientation)을 각 조인트 개수와 링크의 길이로 나타내는 것이다.

순기구학은 조인트와 링크의 길이 등의 함수로 되어 있다. 일반적으로 로봇의 순기구학은 D-H 변수들을 구함으로써 쉽게 얻을 수 있다. D-H 변수란 두 링크를 서로 연결하면 각 링크에 회전축이 생기게 되는데, 이 때 축 사이의 비틀림 각과 축을 중심으로 회전한 각 및 축 사이의 최단 거리를 말한다. 각 조인트에서의 D-H 변수들은 인접한 조인트 사이의 관계를 나타내는 변환행렬을 형성하므로, 베이스에서 팔 끝까지의 변환행렬을 연결하게 되면 로봇 팔 끝의 위치와 오리엔테이션을 베이스 좌표를 기본으로 나타낼 수 있다.

3 역기구학

역기구학은 순기구학과 반대로 직교좌표 변수로부터 조인트 변수를 구하는 것을 말한다. 순기구학에서는 각 조인트의 각이 주어지면 로봇 팔 끝의 위치가 정해졌다. 하지만 로봇 팔 끝의 위치가 주어지고 각 조인트의 각을 구하는 데는 여러 개의 각이 존재하게 된다.

직교좌표 로봇의 예를 살펴보자. 직교좌표에서 같은 한 점을 나타내는 로봇



의 자세는 모두 4가지 경우의 각이 생기게 된다. 이처럼 역기구학의 해는 유일해가 아닌 여러 가지 해가 존재한다.

이러한 역기구학을 통해 각 조인트의 각을 구하는 방법에는 두 가지가 있다. 하나는 기하학을 이용하는 방법인데, 이는 링크의 수가 몇 되지 않는 간단한 로봇일 경우에 편리하다. 다른 한 방법은 분석적인 방법으로 순기구학의 변환행렬의 특성을 사용하여 구하는 방법이다.

4 동기구학

순기구학이나 역기구학은 로봇의 팔 끝의 위치 및 오리엔테이션을 나타내는 직교좌표 변수들과 각 조인트의 각을 나타내는 조인트 변수들과의 정적인 관계를 나타낸다. 로봇이 직교좌표에서의 한 점 P1에서 다른 한 점 P2로 움직이는 경우를 생각해 보자.

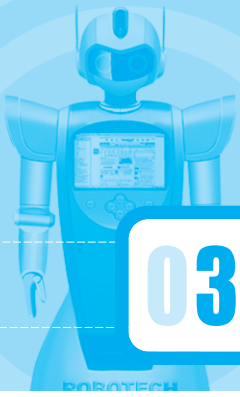
로봇이 P1에서 P2로 움직일 경우, 거리는 좌표를 이용해서 계산할 수 있지만, 목표점에 도달하는 데 걸리는 시간은 결국 로봇이 움직이는 속도에 의해 좌우된다.

따라서, 직교 공간에서 움직임의 속도를 잘 조절해야 정해진 시간에 로봇이 안전하게 주어진 일을 수행할 수 있게 된다. 직교 공간에서 로봇 팔 끝의 움직임을 정확하게 제어하려면, 기준 위치 정보뿐만 아니라 로봇 팔 끝의 속도 정보가 필요하다. 로봇 움직임의 가감속을 제어하지 못하면, 로봇의 움직임이 매우 거칠어져 로봇과 물체에 무리를 줄 뿐만 아니라 모터에도 무리를 줄 수 있다.

한 예로 로봇이 물건을 옮기는 경우를 생각해 보자. 로봇이 물건을 집어들 때는 힘이 많이 듦으로 가속을 하고, 물건을 움직여 옮길 때는 일정한 속도로



움직이고, 물건을 내려놓을 때는 안정하게 감속을 해야 한다. 이처럼 로봇의 팔 끝을 원하는 속도로 움직이게 하려면 속도를 제어하여야 한다. 그렇다면 어떻게 로봇 팔 끝의 속도를 제어할 수 있을까? 이때 필요한 것이 '동기구학'이다.



03 로봇 동역학 기초

1 로봇 동역학의 개념

로봇을 설계하거나 이미 설계된 로봇의 움직임을 조사하려면, 실제 로봇을 대상으로 다양한 변수 값을 직접 측정하여야 한다. 하지만 로봇의 크기나 가격 등의 문제로 실제 로봇에의 적용이 어려울 경우가 있다. 이러한 경우에 로봇 동역학(Dynamics)은 실제 로봇이 움직이는 행동을 미리 알아보기 위해서 로봇을 모델링하는 것이다. 먼저 기구학을 통하여 로봇 조인트의 움직임을 직교좌표 공간에서 나타낼 수 있다. 하지만 이러한 로봇의 움직임은 실제 로봇의 움직임을 고려한 것이 아니라, 기구학의 수식적인 관계를 나타낸 것이다. 실제 로봇의 움직임을 나타내기 위해서는 각 조인트에 적용되는 힘이나 토크 값을 구해야 한다. 각 조인트의 토크 값은 다양한 힘으로 구성되어 있는데, 이 값을 구하면 토크 값에 의한 로봇의 움직임을 동적으로 나타낼 수 있게 된다.

따라서 로봇의 동역학을 구하는 목적은

- (1) 새로운 로봇을 설계할 경우에 그 로봇의 움직임을 미리 알아보기 위함이다.
- (2) 로봇이 없을 경우에 가상적으로 움직임을 연구하기 위함이다.
- (3) 새로운 로봇 제어 이론을 실제 로봇에 적용하기에 앞서 가상적으로 로봇이 움직이는 것을 시뮬레이션하기 위함이다.
- (4) 가상현실과 같은 컴퓨터 그래픽에서 로봇의 현실감이 요구될 경우에 동역학을 사용한 움직임을 구성하기 위함이다.



이처럼 가상적으로 로봇의 움직임을 미리 점검하므로, 실제 응용에서 접하게 되는 문제점들을 보완할 수 있다. 하지만 아무리 로봇의 동역학 식을 정확하게 구한다 할지라도 실제 로봇과는 일치하지 않으므로 시뮬레이션의 한계가 있기 마련이다. 로봇의 모델을 근거로 제어할 때에 실제 로봇과 모델과의 불일치에서 생기는 동역학의 오차의 보정은 많이 연구되고 있는 분야의 하나이다.

로봇의 동역학식을 유도하는 데는 두 가지 방법이 있다.

하나는 뉴턴-오일러(Newton-Euler) 방식이고 다른 하나는 오일러-랑그라지(Euler-Lagrangian) 방식이다.

1) 뉴턴-오일러(Newton-Euler) 방식

이 방식은 동적 시스템을 뉴턴의 두 번째 법칙을 직접적으로 적용하여 각 링크의 좌표에서 링크의 힘과 모멘트로 나타내는 것이다. 따라서 이 방식은 힘의 균형을 기반으로 한 접근 방식으로 동역학 방정식의 유도에 효율적이며, 시뮬레이션과 컴퓨터 계산을 할 때 편리하다.

2) 오일러-랑그라지(Euler-Lagrangian) 방식

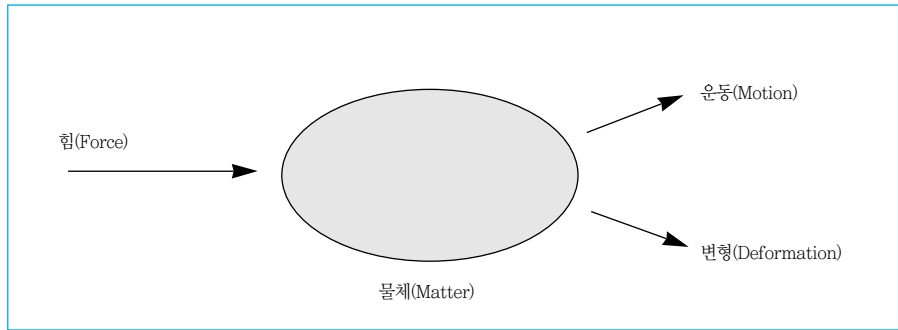
시스템의 동적 특성을 일과 에너지의 개념으로 나타낸 것으로 뉴턴 오일러 방식보다 간단하다. 에너지를 기반으로 한 동역학적 모델접근 방법으로, 비교적 간단한 운동에서 로봇의 운동에 작용하는 여러 가지 변수에 의한 효과를 이해하는 데 유용하며 일반좌표에 근거한다.

2 동역학

동역학이란 역학(Mechanics)의 한 분야로서 물체가 힘을 받아 운동을 일으

킬 때, 그 현상을 분석하는 학문이다.

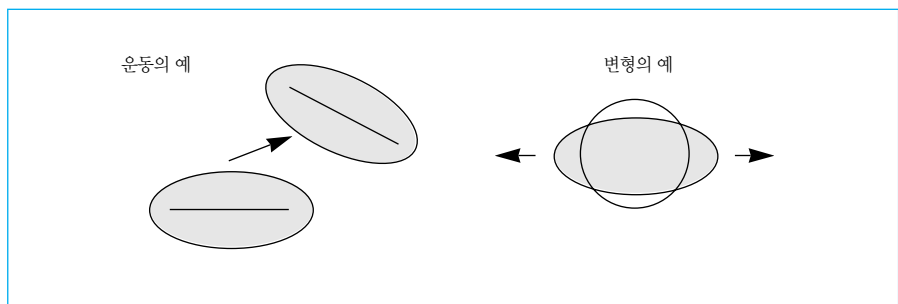
그런데 역학이란 힘, 물체, 운동과 변형 이렇게 3요소 간의 관계를 다루는 물리학의 가장 근본적인 분야를 말한다.



<그림 I-20> 역학의 3요소

힘에는 중력이나 전자기력과 같이 물체의 영역 전체에 작용하는 몸체력(Body Force)과 마찰력이나 압력과 같이 물체의 경계에 작용하는 접촉력(Traction Force)이 있다.

운동은 시간에 따른 물체의 위치 및 자세의 변화를 의미하며, 변형이란 물체 상의 두 점 간의 상대적 거리의 변화를 의미한다. 일반적으로는 운동과 변형이 섞여서 일어나며, 이 경우 둘을 구분하기가 곤란하여 변형과 운동을 하나로 취급하여 해석하기도 한다.



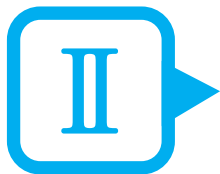
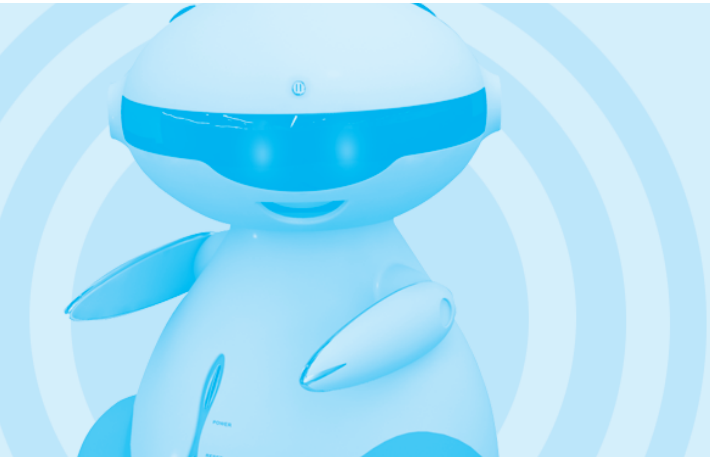
<그림 I-21> 운동의 변형



단원 학습 정리



- 1 로봇은 프로그램된 다양한 움직임에 따라 여러 가지 일들을 수행하는 조작기라고 표현된다.
- 2 지능형 로봇이란 인간과의 상호작용을 통하여 인간의 명령 및 감정을 이해하고 반응하며, 최첨단 기술을 바탕으로 인간에게 다양한 서비스를 제공하는 로봇이다.
- 3 지능형 로봇은 IFR 분류에 의해서 서비스 로봇, 전문가용 로봇, 산업용(제조용) 로봇으로 분류된다.
- 4 산업용 로봇은, 인간의 모습과는 다르게 일반적으로 하나의 팔을 가지고 있으며, 작업에 알맞도록 고안된 도구를 손에 부착하고, 제어 장치에 내장된 프로그램의 순서에 따라 작업을 수행한다.
- 5 산업용 로봇의 분류 방법은 일반적 분류, 제어 방식, 또는 구조적 동작 특성에 따라 분류하는 것이다.
- 6 로봇 기구학에서는 로봇이 움직이면 로봇의 팔 끝이 어디에 있는가를 아는 것이 중요하다. 로봇의 팔 끝을 나타내려면 원점에서 얼마나 떨어져 있는가를 나타내는 위치와 얼마만큼 회전되어 있는가를 나타내는 자세로 정의할 수 있다.
- 7 로봇의 모든 링크의 움직임을 결정하는 독립변수의 수로 나타내는 것을 '자유도' (DOF)라고 한다.
- 8 로봇의 팔 끝을 원하는 속도로 움직이게 하려면 속도를 제어하여야 하는데, 이 때 필요한 것이 '동기구학' 이다.



기계·기구 요소

로봇은 일반적으로 작업을 수행하기 위한 완적기구(Grasping device)와 그 완적기구를 작업공간의 위치까지 이동시키기 위한 이동기구(Moving device)로 이루어진다.

이 단원에서는 완적기구의 구성요소와 이동기구의 종류에 대해서 알아보기로 한다.



학습목표

1. 로봇의 구성 요소인 완적기구와 이동기구의 역할을 이해한다.
2. 구동기기인 액추에이터의 종류를 알고 공압 실린더의 동작을 이해한다.
3. 전자 밸브의 기호와 동작 원리를 이해한다.
4. 각종 제어용 전기기기를 이용하여 공압 실린더의 회로를 이해할 수 있다.

01

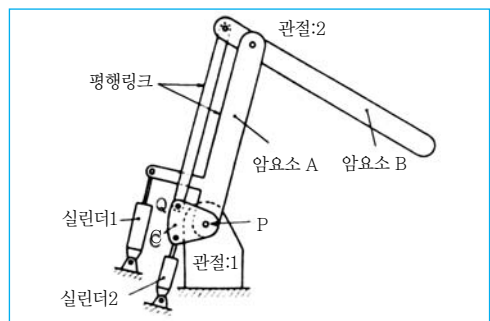
기계 요소

1 개요

로봇의 구조는 일반적으로 작업을 실행하기 위한 완적기구와 그 완적기구 (grasping device)를 작업공간의 위치까지 이동시키기 위한 이동기구 (moveing device)로 이루어진다.

완적기구는 물체를 움켜쥐는 등 작업대상과 직접 상호작용하는 핸드(hand)와 그 핸드를 작업공간 내에서 보다 세밀하게 이동시키기 위한 암(arm)으로 구성된다. 로봇의 몸체를 이동시키는 이동기구는 암의 기능을 확대하는 역할을 한다고 간주할 수 있으나, 작업범위가 한정되는 로봇의 경우 이동기구가 포함되지 않는 고정형으로 하는 경우가 많다. 또한 이동기구가 있는 로봇을 이동로봇이라고 부르기도 한다.

로봇은 각종 유연한 운동기능이 요구되기 때문에 핸드나 암은 관절(joint)로 연결하고, 그 관절에 구동기구를 넣어 그것을 제어함으로써 필요한 운동을 하게 한다. 관절을 움직이는 구동기구로는 전기모터, 유압모터, 실린더 등이 사용된다.



〈그림 II-1〉 핸드, 암, 관절의 구성 예



2 완전기구

1 핸드(hand)

핸드는 앞에서 설명한 바와 같이 암의 가장 앞쪽에 부착되어 작업대상에 직접 작용을 미치는 장치이다. 이 장치는 사람의 손에 해당하나, 사람의 손만큼 다양한 작업능력을 가지는 핸드의 설계는 현재의 기술로서는 어려우며, 용도에 따라 기능이 한정된 핸드를 제작하여 사용한다.

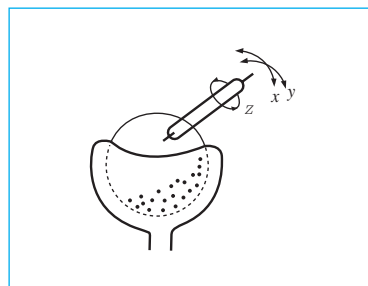


〈그림 II-2〉 용접로봇

용접로봇에서의 용접건, 평평한 금속판이나 깨지기 쉬운 유리판을 다루는 전용로봇에서의 흡착장치 등이 그 예이다.

2 암(arm)

로봇에서 암(arm)은 몇 가지 형태의 관절들을 적절하게 연결함으로써 구성된다. 또 암의 가장 앞쪽에는 핸드가 붙어 있으며, 암의 역할은 그 핸드를 작업공간 내의 적절한 위치로 이동시키는 것이다.



〈그림 II-3〉 구슬 이음매

로봇의 가동성을 나타내는 파라미터로서 자유도(dgree of freedom)가 이용되는데, 이는 핸드의 위치와 자세를 바꾸기 위해 조작되는 변수의 수를 의미한다. 즉, 고정된 위치에서 자유롭게 움직이는 구슬 이음매의 경우 3개의 변수를 조작할 수 있으므로 자유도 3을 가진다. 산업현장에서 이용되는 산업용 로봇의

위치, 자세를 자유롭게 바꿀 수 있는 로봇은 6자유도 로봇이다.

경우에는 통상 3-6의 자유도를 가진다.

3 관절(joint)

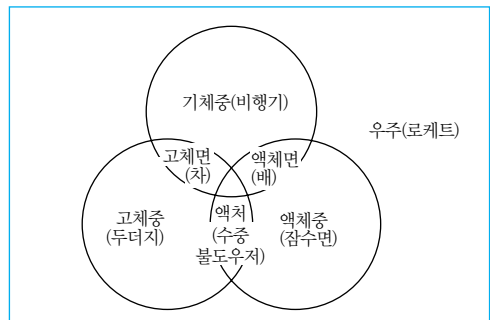
관절은 암과 암, 혹은 여러 개의 암으로 구성된 링크기구와 암을 연결하는 데 사용되며, 전기모터, 유압모터, 실린더 등의 액추에이터(actuator)에 의해 움직인다. 액추에이터는 운동 형태에 따라 직선운동을 하는 선형과 회전운동을 하는 회전형으로 구분되며, 이들의 적절한 조합에 의해 관절의 다양한 운동 형태가 나온다.

또한, 관절의 형태로는 변환 기구를 이용하지 않고 액추에이터로 관절을 직접 구동하는 방식의 직접 구동형(direct driven type)과 변환 기구를 이용하는 간접 구동형(indirect driven type)이 이용되고 있으며 <그림 II-1>의 경우 실린더로 직접 관절이 구동되는 직접 구동형에 해당된다.

3 이동기구

이동기구는 앞에서 말한 바와 같이 완적기구의 이동에 쓰이는 기구이다. 더 나아가 완적기구가 아닌 검사기 등 특수한 목적을 가진 장치를 이동시키기도 하는데, 이처럼 이동기구를 갖는 로봇을 이동로봇이라고 부른다.

이동로봇은 그 이동환경에 따라 제작과정이 크게 달라진다.



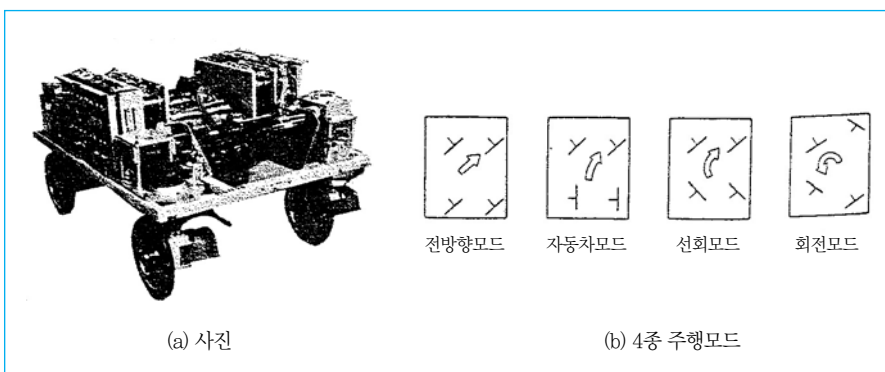
<그림 II-4> 이동환경의 분류

이동로봇의 이동환경을 살펴보면 <그림 II-4>와 같이 분류된다. 거의 모든 이동로봇은 고체와 기체와의 결계면을 이동하는 것으로, 그 밖의 다른 이동환경은 특수 환경이라 해도 좋다. 따라서 여기에서는 고체 표면을 이동하는 이동로봇을 중심으로 그 기구를 설명한다.

1 차륜형(wheeled type)

바퀴를 갖는 이동로봇이 차륜형 이동로봇이다. 차륜형은 평탄면을 이동하는데 있어서 고효율성을 갖기 때문에 광범위하게 이용되고 있다. 따라서 단순히 이동의 문제가 아닌, 얼마나 자유로운 움직임을 실현하는가에 초점이 맞춰져 있으며, 자동차보다 높은 자유도가 요망된다.

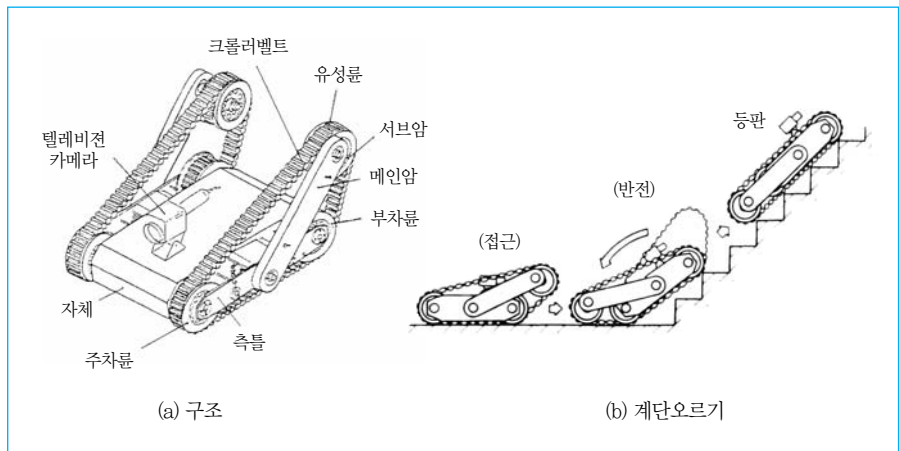
이동로봇 또한 사용목적에 맞게 설계되어야 하지만, 자유로운 이동이 요구되는 경우 전방향 이동(omni-directional mobility)이 가능해야 한다. 또한, 요철(凹凸)일 경우 이동은 차륜방식으로 대응하기 어려워 공기압이 낮은 타이어를 이용하거나, 다리 기계와의 조합을 기도하는 복합화에 의해 문제를 해결하기도 한다.



<그림 II-5> 전방향 이동차의 예

2 크롤러형(crawler type)

크롤러형 이동로봇은 마치 뱀이 땅을 기어가듯 지면을 기어서 이동하는 로봇을 말하며, 토목건축용 기계, 농림업용 기계, 전차 등과 같이 자연의 임야나 흩의 요철(凹凸)면을 주행하는 데 이용된다. 접지면이 크고 요철(凹凸)면을 따를 수 있는 구조로 되어 있기 때문에 차륜형보다 높은 요철(凹凸)면 주행능력을 가지고 있다



〈그림 II-6〉 형상가변 크롤러

3 다리식(legged type)

다리식 이동로봇은 다리를 갖는 로봇으로 평탄면, 요철(凹凸)면에 관계없이 자유로운 이동이 가능하다는 장점이 있다. 이와 같이 지면에 대한 높은 적응성으로 인해 연구가 활발하게 진행되고 있으며, 설계 시 다음의 사항을 고려해야 한다.

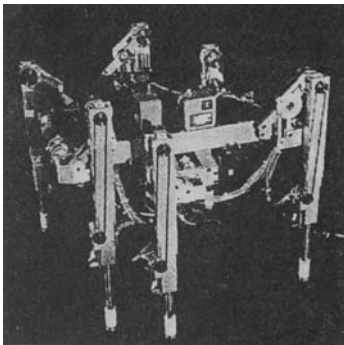
- ① 안정성 : 다리식 이동로봇은 차륜형이나 크롤러형에 비해서 지지 다각형의 변화가 크며, 보행 중에 불안정해져서 넘어지는 경우가 있을 수 있다.

따라서 안정성을 고려해야 한다.

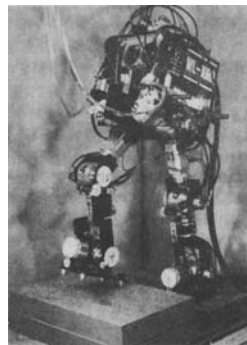
- ② 효율성 : 이동로봇은 이동 중에 외부로부터 에너지 공급을 받지 않는 자립성을 필요로 하는 경우가 많으며, 이동거리에 대한 에너지 소비가 적은 높은 효율성이 요구된다. 다리식 로봇은 다른 형에 비해 에너지 효율이 좋지 않으므로 설계 시 주의해야 된다.
- ③ 지면 적응성 : 다리식 로봇이 위력을 발휘하는 것은 차륜형 로봇이 이동하기 어려운 요철(凹凸)면, 계단 등을 잘 이동하는 데 있다. 그러나 이러한 이동도 효율성이 보장되는 범위 내에서 가능해야 한다.

그 외에 고려해야만 하는 것으로서 속도, 운반능력이 있다.

외모가 인간처럼 생긴 이족보행형 로봇을 휴머노이드(humanoid)라 말한다.



(a) 6족 보행 로봇



(b) 이족 보행 로봇

<그림 II-7> 다리식 로봇

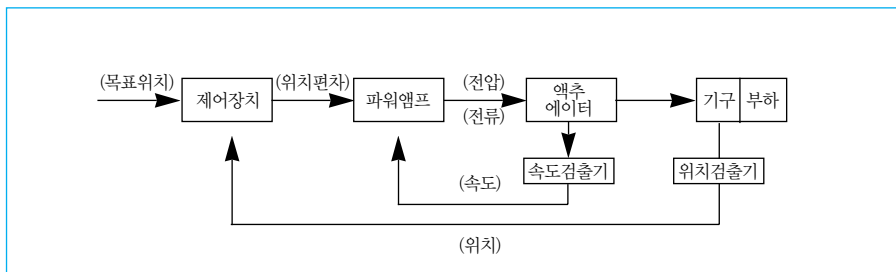
02

액추에이터

1 개요

액추에이터(actuator)는 각종 에너지를 직선운동, 회전운동 등의 기계적 에너지로 변환하여 로봇의 관절을 구동시키는 역할을 한다. 이는 인간의 팔과 다리, 또는 손과 발과 같은 기능을 수행하는 요소이다.

그러나 동작을 위해서는 제어장치가 필요하다. 즉, 구동기기는 제어장치에 의해 유기적인 동작이 가능하며, 필요한 신호의 발생과 처리과정을 통하여 구동기기에 신호를 전달함으로써 액추에이터에 요구되는 동작을 실현한다.



〈그림 II-8〉 액추에이터 제어계의 블록선도

위의 〈그림 II-8〉은 액추에이터 제어계의 전형적인 블록선도(block diagram)를 나타내었다. 제어장치로부터 부하 운동에 대한 목표위치 신호와 현재위치 신호와의 차, 즉 위치 편차가 파워 앰프에 입력되면, 파워 앰프는 부하를 필요로 하는 위치까지 운동시키기 위하여 액추에이터의 구동에 필요한 전류를 발생



시킨다. 액추에이터의 구동 결과로 부하가 운동한 양을 위치 검출기에서 항상 검출해 두며, 그 결과 제어장치는 부하가 목표 위치에 도달할 때까지 위치편차 신호를 계속해 제어장치에 되먹임(feedback)시킨다.

더욱이 제어성능을 향상시킬 목적으로 액추에이터의 속도를 검출하고, 이것을 파워앰프에 되먹임시켜 파워 앰프의 출력전류를 제어함으로써 액추에이터의 힘을 제어하기도 한다. 이것을 속도 되먹임이라고 한다. 또한, 제어하는 대상은 위치만이 아닌 각도, 속도일 경우도 있다.

2 분류와 특징

액추에이터는 여러 형태의 에너지를 기계적 에너지로 변환시키는 장치로, 변환되는 에너지원에 따라 전기식, 유압식, 공압식으로 구분되며, 아래 <표 II-1>과 같이 분류된다.

분류	에너지원	종류
전기 액추에이터 (electric actuator)	전자기에너지 압전효과	직류모터, 교류모터,스테핑모터 전자석, 솔레노이드, 압전세라믹 등
유압 액추에이터 (hydraulic actuator)	유압	유압실린더, 피스톤 모터, 톱니바퀴 모터, 벤 모터 등
공압 액추에이터 (pneumatic actuator)	압축공기	공압실린더, 벤 모터, 에어터빈, 벨로즈, 다이어프램, 고무 인공근육 등

그 밖에 형상기억합금, 고분자 재료의 팽윤을 이용한 액추에이터 등이 있다.

<표 II-1> 액추에이터의 분류

로봇의 의미를 아우르는 공장자동화 시스템은 사용되는 에너지원에 따라 여러 가지 특징을 갖게 된다. 먼저 공압 시스템은 저가로 장치를 구성할 수 있는

특징과 함께, 현재 사용하고 있는 장치나 기계를 쉽게 자동화할 수 있어 경제적 효과가 크나 큰 힘을 얻을 수 없는 단점이 있다.

항목 \ 시스템	전기식	유압식	공압식
힘	중간	크다.	작다.
응답속도	매우 빠르다.	빠르다.	늦다.
제 어 성	위치제어, 힘제어 가능	세밀한 위치제어가 가능하나, 힘제어가 어렵다.	세밀한 위치제어가 어렵다.
안 전 성	과부하에 약하나 그외의 안정성은 높다	과부하에 강하나 상당한 발열로 화재위험이 있다.	발열이 없고 과부하에 가장 강하다.
사용편리성	편리하다.	작동에 유의, 관로의 압력, 필터 관리에 주의	공급공기의 수분 제거, 윤활성 부가에 주의
수 명	길다.	기름에 윤활성이 있기 때문에 길다.	공기에 윤활성이 없어 유압 전기에 비해 짧다.
비 용	보통	높다.	낮다.
소 음	적다.	다소 크다.	크다.

〈표 II-2〉 각종 동력 시스템 비교

이에 반해 유압 시스템은 각 기계장치마다 동력원을 단독으로 사용하므로 비용이 많이 드나, 작은 장치로 큰 힘을 얻을 수 있는 장점이 있다.

3 액추에이터 제어기 설계

액추에이터를 구동하여 원하는 동작을 실현하기 위해서는 이들을 제어하는 제어장치가 필요하다. 제어장치를 다루는 방식에는 크게 두 가지가 있다. 제어 신호가 전해지는 경로의 개폐 상황에 따라 폐루프 제어(closed-loop control)와 개루프 제어(open-loop control)의 두 가지로 분류된다. 개루프 제어는 시



퀀스 제어(sequential control)와 같이 1개의 동작이 끝나면 그 결과에 따라서 다음 동작이 개시되는 식으로 순차동작을 일으켜 목적을 달성하는 방식의 것이며, 이에 대해 폐루프 제어는 제어 결과를 입력측에 되돌려(feedback) 결과를 보다 세밀하게 제어하도록 한 것이다. 예를 들면, 전기세탁기, 전기밥솥, 엘리베이터 등은 시퀀스 제어에 속하고, 자동온도 조절장치 등은 되먹임 제어에 속한다.

산업용 로봇은 제어 목적에 맞게 두 가지 제어 방식을 다 이용하나, 여기에서 다루는 전기공압 제어나 PLC 제어는 공압 실린더를 정해진 순서에 따라 동작하게 하므로 시퀀스 제어에 속한다.

또한, 제어 방식과는 별개로 액추에이터를 구동시키는 매체에 따라 제어 방법을 구분해 보면, 공압 액추에이터의 경우 각종 밸브만을 사용한 순수공압 제어, 릴레이 접점의 개폐를 이용한 전기공압 제어(릴레이 제어), 0 또는 1의 2진 신호를 처리하는 장치인 PLC(Programmable Logic Controller)를 이용한 PLC 제어로 구분할 수 있다. 또한, 마이크로프로세서(microprocessor)에 의해서도 제어가 가능하나, 여기는 전기공압 제어, PLC 제어에 대해 알아보기로 한다.

또한 구동기기인 액추에이터는 공압 액추에이터 중에서 공압 실린더로 한정하며, 전기 액추에이터인 모터의 제어는 뒤의 4장에서 다룬다.

'0' 또는 '1'의 2진 신호를 디지털 신호라 한다.

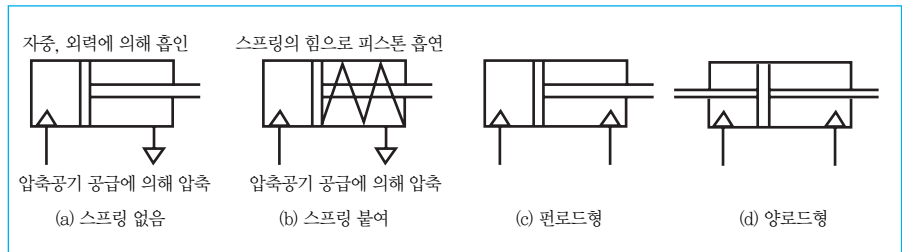
1 공압 액추에이터

공압 액추에이터는 압축공기의 압력 에너지를 기계적 에너지로 변환하며 직선운동, 회전운동, 요동운동 등을 하도록 고안된 구동기기이다. 공압 액추에이터는 직선운동을 하는 공압 실린더와 회전운동을 하는 공압모터, 제한된 회전각도의 요동운동을 하는 요동 액추에이터로 구분된다.

가장 일반적으로 사용되는 공압 실린더는 직선운동형으로 압축공기의 공급 포트 수에 따라 단동 실린더와 복동 실린더로 구분된다. 단동 실린더는 한쪽 방향의 운동은 압축 공기에 의해 일어나고, 다른 한쪽의 운동은 내장된 스프링이

나 외력에 의하여 일어난다.

반면에 복동 실린더는 양쪽 방향에서 압축공기가 공급되어 실린더가 압출 또는 흡인된다. 아래의 <그림 II-9>에서 삼각형은 압축공기가 가해지거나 배기되는 방향을 나타낸 것으로 생략해 표시하기도 한다.



<그림 II-9>

2 전기공압 제어

릴레이 접점의 ON, OFF를 이용한 제어로 릴레이 제어라고도 한다.

(1) 제어용 전기기기

제어회로에 사용되는 각종 제어용 전기기기는 크게 조작용 스위치, 검출 스위치, 릴레이 류 등으로 분류되며, 전기 회로를 이해하려면 이들 기기들의 동작 및 쓰임새를 이해해야 한다.

분류	종류
조작용 스위치	푸시버튼 스위치, 셀렉터 스위치, 로터리 스위치 등
검출 스위치(센서)	리미트 스위치, 광전 스위치, 근접 스위치 등
릴레이	제어용 릴레이, 타이머 등
작동기기	모터, 전자밸브, 솔레노이드 등
표시등	표시등, 부저 등
기 타	카운터, 저항기, 정류기, 변압기 등

<표 II-3> 제어회로에 사용되는 각종 제어용 전기기기의 분류

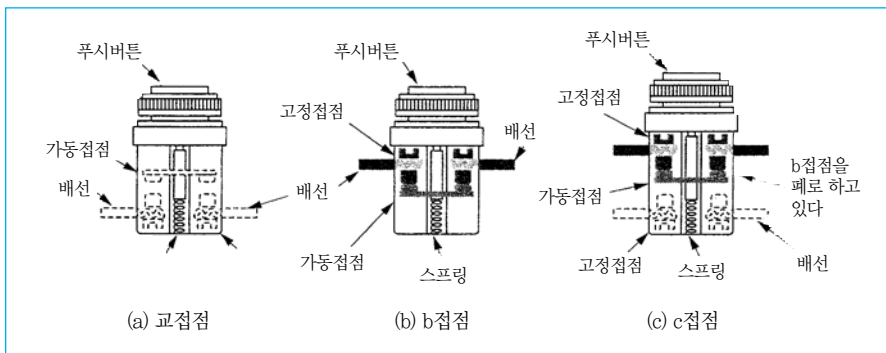


1) 조작용 스위치

조작용 스위치는 전기 회로의 개폐 또는 접속을 변경시키는 기기를 뜻한다.

① 푸시버튼 스위치(push button switch)

푸시버튼 스위치는 신호 입력 요소로 기계나 장치가 출발하도록 하는 데 사용된다. 이는 계속 누르고 있어야 되는 위치에 머무르며, 그렇지 않은 경우 원래의 위치로 돌아간다.

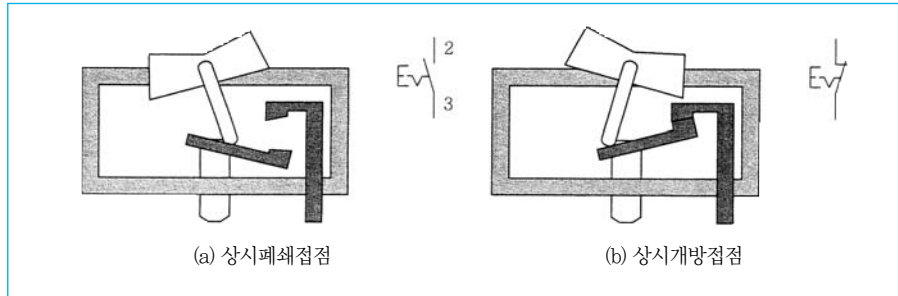


〈그림 Ⅱ-10〉 푸시버튼 스위치의 구조

그리고 a접점과 b접점의 기능을 동시에 갖는 접점을 c접점이라 하며, 위의 그림 (c)에 나타내었다. c접점은 조작 시 b접점은 열리고 a접점은 닫히며, 버튼을 놓으면 스프링에 의해 원래대로 복귀되어 b접점은 닫히고 a접점은 열린다.

② 유지형 푸시버튼 스위치(latching push button switch)

유지형 푸시버튼 스위치는 조작된 상태가 반대로 조작되기 전까지 유지되는 스위치이다. 이러한 스위치의 종류로는 셀렉터 스위치, 텀블러 스위치, 토글 스위치 등이 있다.



〈그림 II-11〉 텀블러 스위치의 구조

2) 검출용 스위치

검출용 스위치란 검출 대상의 유무나 위치 또는 온도, 압력 등의 변화를 감지해서 자동적으로 접점을 개폐하는 센서(sensor)로, 인간의 눈이나 귀와 같은 감각에 대응된다.

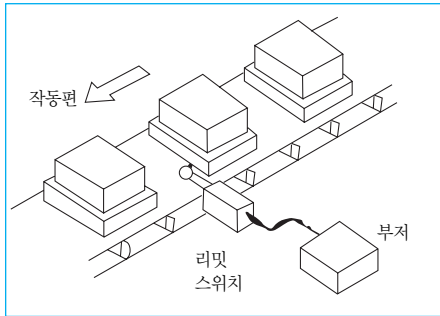
또 검출 대상과의 접촉 유무에 따라 크게 접촉형과 비접촉형으로 나눌 수 있다. 접촉형에는 리밋 스위치, 압력 스위치 등이 있으며, 비접촉형에는 광전 스위치, 근접 스위치, 초음파 스위치 등이 있다.

검출용 스위치는 기본적으로 센서이므로 자세한 설명은 뒤의 “센서”에서 설명하기로 하고 여기서는 리밋 스위치만을 다룬다.

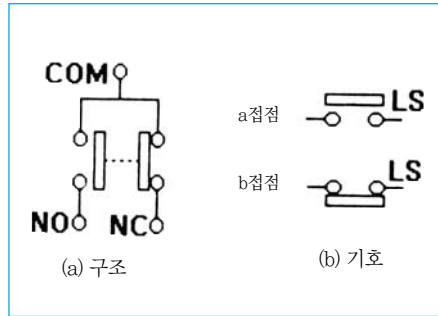
① 리밋 스위치(limit switch)

리밋 스위치는 작동편이 있는 스위치로서 검출하려고 하는 물체에 의하여 작동편이 눌러 접점이 개폐된다. 〈그림 II-12〉와 같이 물체의 이동통로에 리밋 스위치를 설치해 두면, 물체가 이동하면서 리밋 스위치의 작동편을 눌러 접점이 개폐되어 물체의 유무나 위치를 검출하게 된다.

〈그림 II-13〉에서 리밋 스위치의 a접점은 공통 단자인 COM(Common)단자와 NO 단자를 사용하고, b접점은 공통 단자인 COM단자와 NC단자를 사용한다.



〈그림 II-12〉 리미트 스위치의 사용 예



〈그림 II-13〉 리미트 스위치의 구조와 기호

3) 릴레이(relay)

① 제어용 릴레이

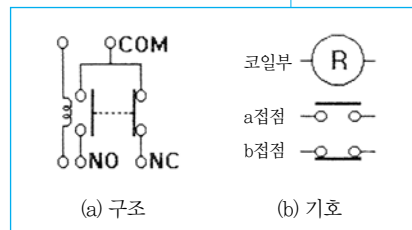
보통 제어용 릴레이를 그냥 릴레이 또는 전자 계전기라 부르며, 〈그림 II-14〉와 같이 코일부와 접점부를 구분하여 표시한다. 릴레이는 초기상태에 a접점은 OFF, b접점은 ON 상태를 유지한다. 이때 코일에 전류가 흘러 코일이 여자되면, 코일의 흡인력에 의해 릴레이 스위치의 접점이 이동하여 a접점은 ON 되고, b접점은 OFF 된다. 다시 코일에 흐르는 전류를 차단하여 코일이 소자되면 초기상태로 돌아간다.

릴레이는 여러 개가 사용되는 경우 어느 릴레이의 접점인가를 명백하게 하기 위해서 코일부의 문자기호 R에 첨자를 부가하여 $R_1, R_2 \dots$ 로 표시하고, 이 코일로 동작되는 접점도 동일문자 기호를 기입한다.

또한, 일반적으로 릴레이는 복수 개의 접점을 갖는다. 따라서, 2개 이상의 접점을 사용하고 있는 경우에도 코일부와 동일한 문자를 기입한다.

릴레이 접점 이용 시 옆의 그림에서와 같이 릴레이의 a접점은 COM단자와 NO단자를, b접점은 COM단자와 NC단자를 이용한다.

여자란 릴레이 코일에 전류를 흘려서 자화 성질을 갖게 하는 것을 말한다. → 소자



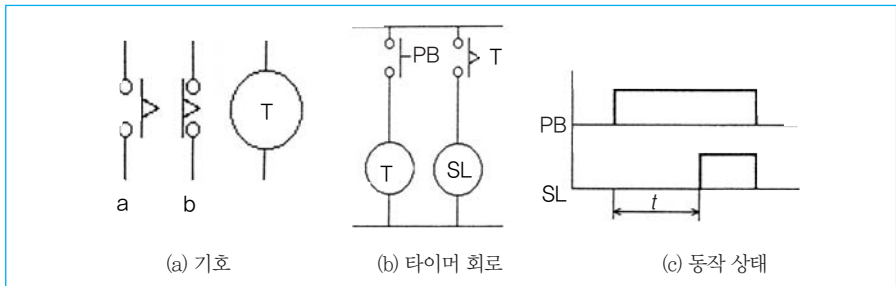
〈그림 II-14〉 릴레이의 구조와 기호

② 타이머(timer)

타이머는 미리 정해진 시간의 경과 후에 회로를 전기적으로 개폐하는 접점을 가진 계전기로, 한시 계전기라 부르기도 한다. 여기에는 코일이 여자되고 일정 시간이 경과된 후 접점이 반전되는 한시 동작 타이머(on-delay timer)와, 코일이 소자된 후 일정시간이 경과된 후 접점이 복구되는 한시 복구 타이머(off-delay timer)가 있다.

a. 한시 동작 타이머(on-delay timer)

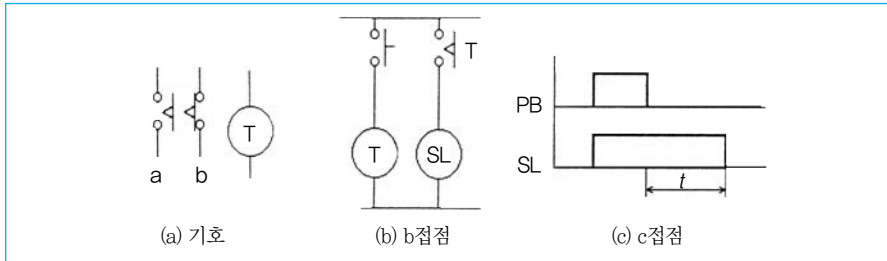
한시 동작 타이머는 용어의 의미 그대로 릴레이의 코일이 여자되고 일정시간이 경과된 후 접점이 반전된다. 반대로 코일이 소자되면 접점은 바로 복구된다. 즉, 한시 동작하고 순시 복구되는 타이머이다. 타이머의 기호 또한 릴레이와 마찬가지로 코일부와 접점부로 구분하여 표시하고, 접점부의 경우 a는 a접점을 나타내고 b는 b접점을 나타낸다.



〈그림 II-15〉 한시 복구 타이머 회로(a접점)

한시 동작 타이머 회로의 한시 접점 기호는 가로로 표시하는 경우에는 뽀족한 부분을 선의 위의 방향으로 표시하고, 세로로 표시하는 경우에는 선의 오른쪽 방향에 표시한다.

b. 한시 복구 타이머(off-delay timer)



〈그림 II-16〉 한시 복귀 타이머 회로(a점점)

한시 복귀 타이머는 릴레이의 코일이 여자되면 접점은 바로 반전된다. 반대로 코일이 소자되면 접점은 일정시간이 경과된 후 복귀된다. 즉, 순시 동작하고 한시 복귀되는 타이머이다. 한시 복귀 타이머의 접점 또한 한시 동작 타이머와 마찬가지로 a는 a점점을 나타내고 b는 b점점을 나타낸다.

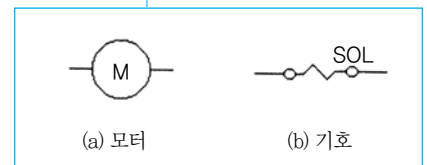
한시 복귀 타이머 회로의 한시 접점 기호는 한시 동작 타이머와는 반대이다. 따라서 가로로 표시하는 경우에는, 뾰족한 부분을 선의 아래 방향으로 표시하고, 세로로 표시하는 경우에는 선의 왼쪽 방향에 표시한다

4) 작동 기기

스위치나 릴레이의 접점의 개폐에 의해 작동되는 것으로 모터, 전자 밸브, 솔레노이드 등이 있다.

① 모터(motor)

모터는 전동기라고 부르기도 하며 전기 에너지를 기계적인 회전에너지로 변환하는 기기로, 사용하는 에너지원에 따라 크게 직류 모터와 교류 모터로 구분된다. 모터의 기호는 〈그림 II-17〉의 (a)와 같다.



〈그림 II-17〉 모터와 솔레노이드 기호

② 솔레노이드(solenoid)

솔레노이드는 코일에 전류를 흘려 전자석을 만들고, 그 흡인력으로 가동편을

움직이게 하여 인장이나 압출 등의 직선 운동을 시키는 것이다.

솔레노이드는 단독으로 사용되기보다 전자 밸브 등과 같이 사용된다. 솔레노이드의 기호는 <그림 II-17>의 (b)와 같다.

③ 전자 밸브(electric valve)

전자 밸브는 공기 흐름의 경로를 제어하는 방향제어 밸브에, 솔레노이드를 부가하여 그 제어를 전기적으로 가능하게 한 밸브로, 솔레노이드 밸브(solenoid valve)라 부르기도 한다.

표시법	기호설명	기호
2/2-way 밸브	정상상태에서 닫힌 상태에 있음(N.C형)	
	정상상태에서 공기통로가 연결됨(N.O형)(P→A)	
3/2-way 밸브	정상상태에서 P는 외부와 차단된 상태, A는 R로 배기됨(N.C형)	
	정상상태에서 P와 A가 연결됨(N.O형)	
4/2-way 밸브	2개의 작업라인이 있어 복동실린더의 제어에 사용됨	
5/2-way 밸브	2개의 작업라인이 각각의 배기공을 갖고 있음. 복동실린더의 제어에 사용됨	
3/3-way 밸브	중립위치에서 모든 라인이 닫혀 있음	
4/3-way 밸브	중립위치에서 P가 배기됨(유압용임)	
	Floating 중립위치를 갖고 있음	
5/3-way 밸브	중립위치에서 모든 라인이 차단되어 있음	
	실린더를 임의의 위치에서 정지시킬수 있음	
5/4-way 밸브	중립위치에서 모든 라인이 차단되어 있음	
	실린더를 임의의 위치에서 정지시킬 수 있음	

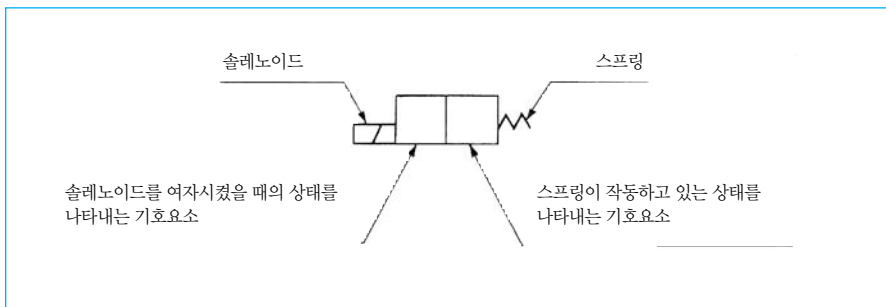
<표 II-4> 여러 가지 방향제어 밸브

방향제어 밸브는 밸브가 기계나 시스템에 설치되지 않고 혼자 존재할 경우의 상태를 정상상태라 하며, 밸브를 시스템 내에 설치하고 작업을 시작하려 할 때의 상태를 초기 상태라 한다. 밸브의 기호는 밸브에 어떠한 외력도 작용하지 않



았을 때인 정상 상태를 기준으로 한다. 정상 상태에는 공기의 흐름이 차단된 상태로 있는 닫힘 위치(NC, Normally Closed)와 공기가 흐를 수 있는 열림 위치(NO, Normally Open)의 두 가지가 있다.

아래의 표에서 4각형은 밸브의 제어위치를 나타내며, 4각형이 두 개인 경우에는 두 개의 제어위치를 갖는다. 화살표는 공기의 흐름 방향을, T자 모양의 포트(port)는 공기의 흐름이 차단됨을 의미한다. 또 3/2-way 밸브로 예를 들면, 앞의 숫자 3은 포트 수를 나타내며, 뒤의 숫자 2는 밸브의 제어위치의 수를 나타낸다.



〈그림 II-18〉 전자 밸브의 동작 방법

방향 제어 밸브는 수동, 공기압, 전기적인 방법 등으로 제어되는데 이 가운데 전자 밸브는 전기적인 방법에 의해 제어된다. 〈그림 II-18〉은 편 솔레노이드 밸브로, 큰 사각형 좌측에 붙어 있는 돌출 부분이 솔레노이드를 나타낸다. 이것이 여자되면 해당 기호요소로 제어위치가 전환됨을 의미한다.

수동에는 누름버튼, 레버, 페달 등이 이용되고, 공기압에는 압력을 가하거나 제거하여 밸브를 제어한다.

구분	ISO-1219	ISO-5599/II
작업라인	A, B, C	2, 4, 6
압축공기 공급 수	P	1
배기구	R, S, T	3, 5, 7
제어라인	X, Y, Z	10, 12, 14

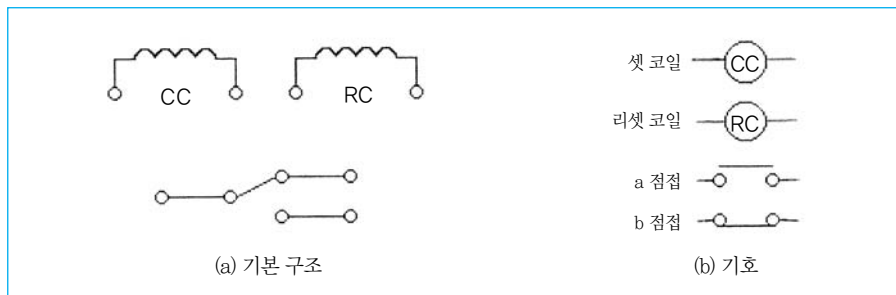
〈표 II-5〉 밸브 포트의 기호

또한, 밸브가 확실히 설치될 수 있도록 각 포트에는 고유한 문자로 표시된다. 포트의 표시방법은 대체로 모든 제조 메이커가 ISO(국제표준화기구)의 규정을 따르고 있다. ISO 규정에는 문자로 표시되는 ISO-1219 규정과 숫자로 표시되는 ISO-5599/II 규정의 두 가지가 있으며, 유압과 공압에서 모두 공통적으로 사용된다.

5) 카운터(counter)

카운터는 단순히 계수를 위한 기기로서 컨베이어 벨트를 통해 전달되는 생산품의 수량, 회전기기의 회전 수 등을 계수하는 데 이용된다.

카운터는 규칙적이거나 불규칙적으로 발생하는 신호를 검출하는 스위치와 릴레이의 조합된 기능을 갖고 있다고 할 수 있다. 따라서 릴레이와 마찬가지로 코일부와 접점부를 구분하여 표시한다.



〈그림 II-19〉 카운터의 구조와 기호

동작 원리는 카운터 코일에서 신호가 검출될 때마다 숫자 톱니바퀴를 구동하고, 설정치에 도달하면 내장된 a접점이나 b접점의 스위치가 조작된다. 리셋 코일은 현재의 계수값을 초기화한다.



(2) 기본 회로

전기공압에 사용되는 기본 회로는 논리 회로와 자기 유지 회로, 시간 지연 회로, 인터록 회로이며, 회로도 작성 시 위의 “제어용 전기기기”에서 설명한 기호와 더불어 아래의 ISO방식을 주로 이용하고, 경우에 따라서 Ladder 방식의 기호를 사용하기도 한다.

제어기기	ISO 방식		Ladder 방식	
	A 접점	B 접점	A 접점	B 접점
누름 버튼 스위치 (push button switch)				
리밋 스위치				
릴레이				
솔레노이드 (solenoid)				

〈표 II-6〉 ISO 방식과 Ladder 방식의 기호

1) 논리 회로

논리 회로란 스위치와 릴레이를 이용하여 여러 가지 논리적인 판단이 가능하도록 꾸민 회로이다. 이때 스위치는 입력 요소가 되며 릴레이는 출력 요소가 된다. 즉, 스위치의 조작에 의해 조건이 맞는 경우 릴레이의 코일이 여자되어 릴레이에 속한 접점들이 개폐된다.

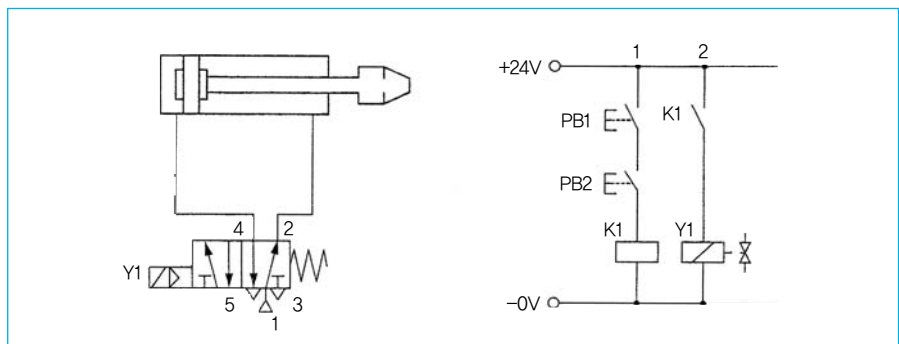
기본적으로 AND회로의 구현은 a접점 스위치를 서로 직렬로 연결하여 구현하고, OR회로의 구현은 a접점의 스위치를 서로 병렬로 연결하여 구현한다. 또 NOT 회로의 구현은 b접점 스위치를 이용한다. 이는 스위칭 이론(switching theory)의 불대수(Boole algebra)와 관계된다.

따라서, 불대수에 의해 NAND, NOR 등 보다 다양한 논리 회로의 구성이 가능하게 된다.

① AND 회로

복동 실린더와 두 개의 푸시버튼 스위치를 이용해 AND회로를 구성하면 <그림 II-20>과 같다. 이때 두 개의 푸시버튼 스위치 PB1과 PB2를 눌러야만 실린더가 전진을 하고, 두 개 중 어느 하나라도 ON되지 않으면 실린더는 후진 상태를 유지한다.

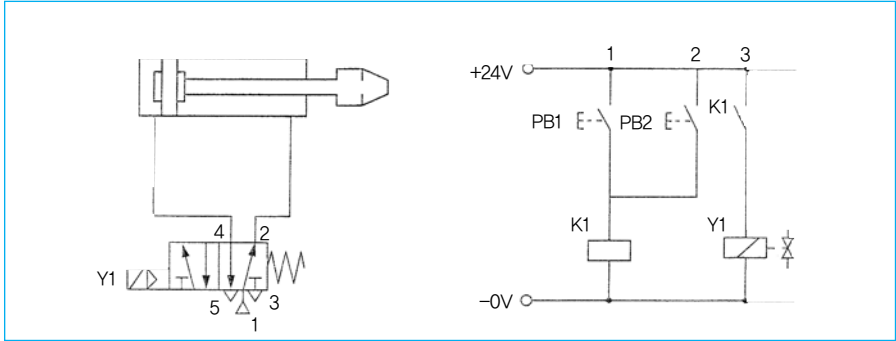
NAND는 AND+NOT,
NOR는 OR + NOT를 의미한다.



<그림 II-20> AND 회로

② OR회로

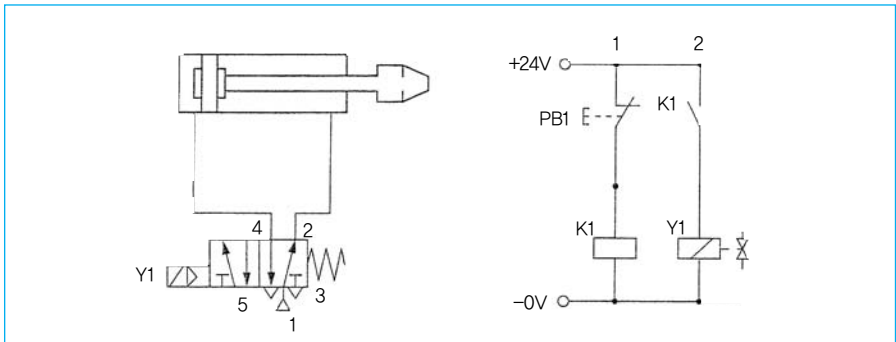
복동 실린더와 두 개의 푸시버튼 스위치를 이용해 OR회로를 구성하면 <그림 II-21>과 같다. 이때 PB1 혹은 PB2 중에서 어느 한 개 이상의 푸시버튼 스위치가 눌렸을 때 실린더가 전진을 하고, 두 개 모두 OFF 되면 실린더는 후진 상태를 유지한다.



〈그림 II-21〉 OR 회로

③ NOT 회로

〈그림 II-22〉는 스위치 PB1이 b접점이므로, 푸시버튼 스위치를 조작하지 않았을 때는 실린더가 전진하고, 푸시버튼 스위치를 누르면 실린더는 후진한다.



〈그림 II-22〉 NOT 회로

2) 자기유지 회로

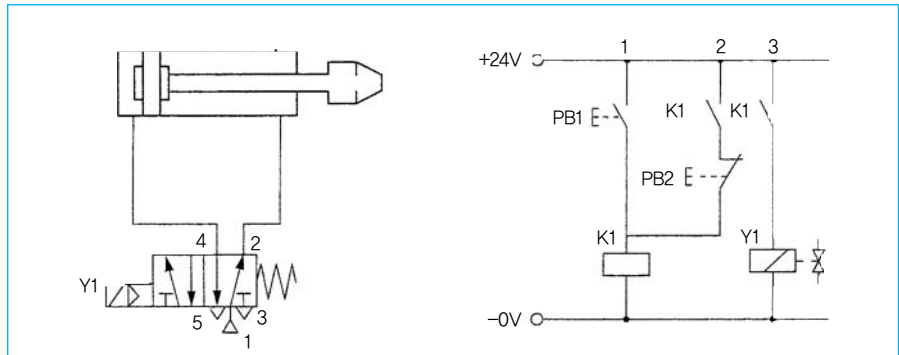
자기유지 회로는 전기신호의 기억이 필요한 전기공압 제어의 제어장치에 필요하며 ON 우선 회로와 OFF 우선 회로가 있다.

자기유지란 릴레이 코일을 여자시킨 입력이 제거된 후에도 계속 릴레이 코일이 여자 상태를 유지함을 의미한다.

① ON 우선 회로

〈그림 II-23〉에서 릴레이 코일 K1은 PB1을 누르면 실린더가 전진하고, PB1을 놓아도 PB2가 ON 상태이므로 실린더가 전진 상태를 유지한다. 또 PB2를 누르면 실린더가 후진을 하고, PB2를 놓아도 실린더는 후진 상태를 유지한다.

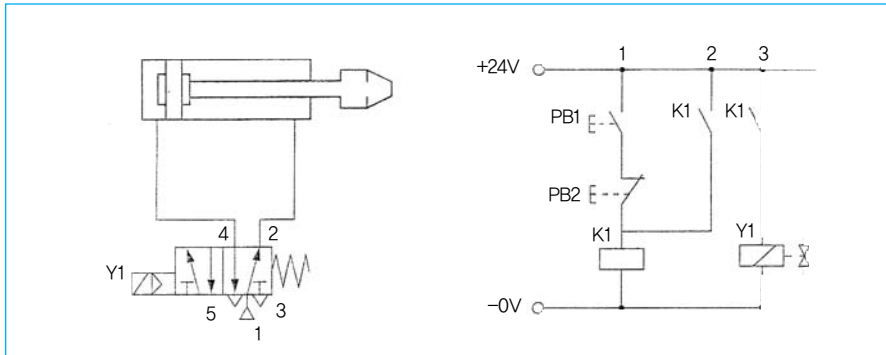
이 회로에서 PB1은 실린더 전진용 스위치이고 PB2는 실린더 후진용 스위치인데, PB1과 PB2가 동시에 눌렀을 때 릴레이 K1이 여자되므로, PB1이 출력 상태를 좌우하여 ON 우선회로가 된다.



〈그림 II-23 ON 우선회로〉

② OFF 우선 회로

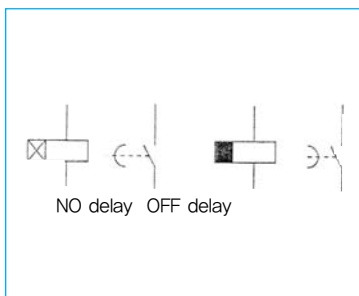
〈그림 II-24〉에서 릴레이 코일 K1은 PB1을 누르면 실린더가 전진하고, PB1을 놓아도 PB2가 ON 상태이므로 실린더가 전진 상태를 유지한다. 또 PB2를 누르면 실린더가 후진을 하고, PB2를 놓아도 실린더는 후진 상태를 유지한다. 즉, ON 우선 회로와 똑같은 동작 상태를 갖는다. 그러나 PB1과 PB2가 동시에 눌렀을 때 릴레이 K1이 소자되므로, 실린더 후진용 스위치 PB2가 출력 상태를 좌우하여 OFF 우선회로가 된다.



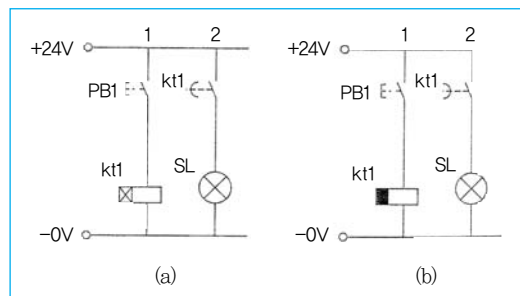
〈그림 II-24 OFF 우선회로〉

3) 시간지연 회로

시간지연 회로란 타이머에서 설명한 한시 동작 타이머(on-delay timer)와 한시 복귀 타이머(off-delay timer)를 말한다. 전기제어회로의 경우는 푸시버튼 스위치를 누르는 등의 입력 신호를 주고, 일정 시간이 경과된 후에 접점이 ON 또는 OFF하는 회로로, 타이머를 사용하여 회로를 구성하므로 타이머 회로라고도 한다. 보다 자세한 설명은 타이머(timer)를 참조한다.



〈그림 II-25〉 타이머 기호



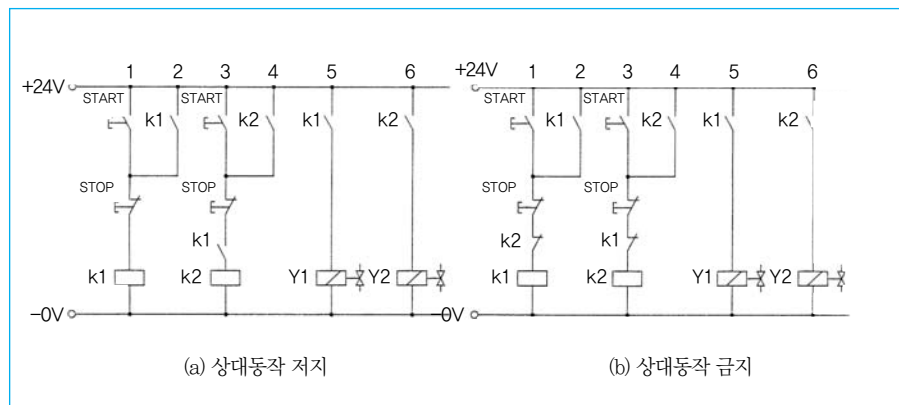
〈그림 II-26〉 시간지연 회로

4) 인터록(inter-lock) 회로

인터록 회로는 복수로 작동이 이루어질 때 어떤 조건이 구비될 때까지 작동을 저지시키거나, 한 작동이 이루어지면 다른 작동은 금지되는 회로를 말한다.

다. 인터록 회로는 시스템을 안전하고 확실하게 운전시키기 위한 판단회로로 시스템에 반드시 구현되어야 한다. 인터록 회로의 두 기능 중 전자에 해당되는 회로가 <그림 II-27>의 (a)이며, 후자에 해당되는 회로가 <그림 II-27>의 (b)이다.

<그림 II-27> (a)의 경우 K2는 K1이 작동되어야만 작동된다. 즉, K1이 작동된 후에 K2의 작동이 가능하다. <그림 II-27> (b)의 경우에는 K1이 작동되면 K2가 작동되지 않고, K2가 작동되면 K1이 작동되지 않는다. 즉, 두 개의 솔레노이드 중에서 어느 하나만을 작동시킨다.



<그림 II-27> 인터록 회로

3 PLC 제어

(1) 개요

자동화 장치를 구성하고 있는 기본요소는 기계구조를 위시해서 센서, 프로세서, 액추에이터로 크게 나눌 수 있다. 이 중 특히 중요한 의미를 갖는 것은 생산수단이 기계화를 거쳐 자동화로 옮겨가는 시점으로 볼 때 프로세서라 할 수 있다. PLC(Programmable Logic Controller)도 이러한 역할을 수행하는 릴레이를 대신하여 1970년대에 처음으로 사용되기 시작한 이래, 전자공학의 발전과 더불어 계속 진보를 거듭해 왔다. 따라서 현재의 시스템은 PLC라는 이름 자체가 갖는 기능보다 훨씬 우수한 기능을 갖고 있어 구조상 컴퓨터와 구분하기 힘



든 실정이다.

PLC는 일반적으로 다음과 같이 정의된다. 논리연산, 순서조작, 시한, 계수 및 산술연산 등의 제어 동작을 실행시키기 위해 제어 순서를 일련의 명령어 형식으로 기억하는 메모리를 가지고, 이 메모리의 내용에 따라 제어 대상의 제어를 디지털 혹은 아날로그 입·출력을 통해서 행하는 디지털 조작형의 공업용 전자장치이다.

1) 릴레이 제어와의 비교

릴레이와 PLC는 자동화 시스템에서 전체적인 제어를 행하는, 마치 인체의 두뇌와 같은 역할을 하는 프로세서라는 점에서 그 기능은 같다. 그러나 이들은 그 구성요소가 서로 다른 특성을 가지고 있으므로 여러 가지 면에서 상이하다.

아래 <표 II-8>에 간단히 두 방식의 차이점을 비교해 보았다.

구분	릴레이 제어	PCL 제어
제어 방식	소프트웨어	하드웨어
제어 기능	한정적이며 대형	고기능이며 소형
제어 요소	유접점 릴레이 등	무접점 릴레이 등
신뢰성	기계적 마모 등으로 낮다.	기계적 접촉이 없으므로 높다.
제어 내용의 변경	배선을 바꿔야 하므로 어렵다.	프로그램 변경만으로 가능하다.
컴퓨터와의 호환성	없음	있음

<표 II-8> 릴레이 제어와 PLC 제어의 비교

2) 컴퓨터와의 비교

PLC는 컴퓨터와 같은 원리로 동작한다. 따라서 하드웨어적 기본구조는 차이점이 없으나 거기에 연결되는 입·출력 장치가 다르다. 컴퓨터는 키보드, 모니터 등의 입·출력 장치가 연결되나, PLC는 각종 센서, 액추에이터의 입·출력 장치가 연결된다.

PLC의 기능상 발달 경향은 고속화, 다른 컴퓨터와의 연계, 소형 저렴화 등이다.

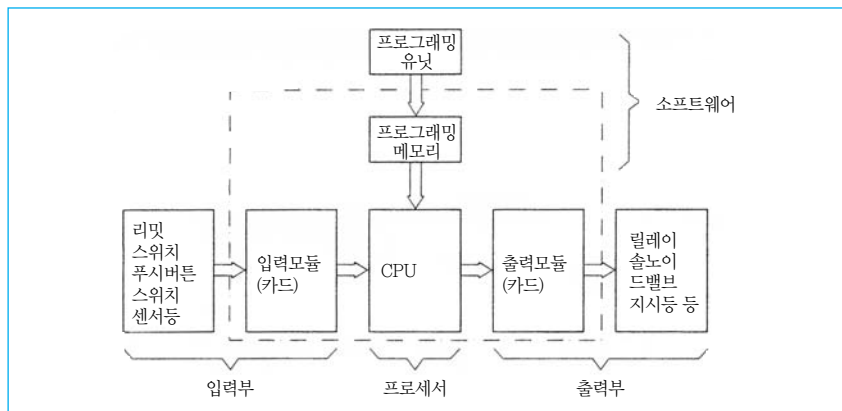
그리고 PLC와 컴퓨터의 다른 점은 사용 언어에 있다. 컴퓨터에서는 언어와 유사한 형태의 프로그램 언어가 사용됨으로써 프로그래머에게 편리함을 제공하며, PLC는 릴레이 레더 기호를 이용함으로써 사용자들이 쉽게 익힐 수 있게 된다. 따라서 PLC는 작업 현장에서 사용되며, 독특한 언어를 갖는 특수용 컴퓨터라 할 수 있다. <표 II-9>에 컴퓨터와 PLC를 비교해 보았다.

항목	구분	PLC	컴퓨터
입력장치		각종 센서	키보드, 스캐너 등
출력장치		각종 액추에이터	모니터, 프린터 등
프로그램 취급자		현장 작업자	전문가
사용언어		시퀀서 제어용 언어	전문언어
사용전압		강전, 약전 병용	약전
목적		제어용	데이터 처리용
설치장소		공장	사무실

<표 II-9> 컴퓨터와 PLC의 비교

(2) PLC의 기본 구조와 동작 원리

1) PLC의 기본 구조



<그림 II-28> PLC의 기본 구조



PLC의 기본 구조를 살펴보면 크게 4가지 요소로 구분된다. 첫째 센서에 의해 생성되는 신호를 입력받기 위한 입력부, 둘째 입력된 신호를 처리하기 위한 프로세서, 셋째 제어 명령을 담고 있는 메모리부, 넷째 처리된 신호를 외부에 연결된 솔레노이드나 릴레이에 출력하는 출력부로 나누어진다.

〈그림 II-28〉은 PLC의 기본적인 구조를 나타낸 것이다. 각 부분별로 좀더 자세히 알아보자.

① 프로세서

PLC 입력부는 인간의 오감에 비유될 수 있고, 프로세서는 인간의 두뇌에, 출력부는 인간의 손, 발에 비유될 수 있다. 따라서 PLC의 프로세서는 제어 장치의 모든 제어를 행하는 곳으로, PLC의 기능이라 함은 대부분 이 프로세서의 기능을 의미한다.

CPU의 주요기능은 메모리의 프로그램에 따른 입·출력부의 데이터 교환이나 연산, 비교, 판정, 일시 기억 등이며 주로 Z-80, 8085, M6800 등의 8bit CPU가 사용되고, 고속 처리를 위하여 16bit CPU가 사용되기도 한다.

② 메모리부

PLC에는 PLC 자체의 동작을 제어하기 위한 시스템 프로그램과 제어 대상을 제어할 목적으로 사용자가 작성한 프로그램이 저장되는데, 이들 프로그램을 저장하는 메모리가 각각 시스템 메모리와 사용자 메모리이다. 이들 메모리는 다시 각각의 프로그램을 담는 프로그램 메모리와 해당 프로그램의 실행에 필요한 데이터를 담는 데이터 메모리로 구분된다.

좀더 자세히 설명하면 시스템 메모리는 입·출력 스캔(scan), 명령 실행, 통신 등의 시스템 프로그램을 저장하는 시스템 프로그램 메모리와, 시스템 프로그램이 동작하기 위해 필요한 실시간 데이터, 통신 버퍼 내용 등을 저장하기 위한

EEPROM역시 비휘발성 메모리이다.

시스템 데이터 메모리로 구분된다. 이들 메모리에는 각각 용도에 맞게 비휘발성인 EPROM (Erasable Programmable ROM)과 휘발성인 RAM (Random Access Memory)이 사용된다.

사용자 메모리도 마찬가지로 두 가지로 구분되는데, 하나는 사용자 프로그램 메모리로 사용자가 작성한 프로그램이 저장되며 RAM, EPROM, EEPROM(electrically EPROM)이 사용되고, 또 하나는 사용자 데이터 메모리로 각 입·출력 상태, 타이머나 카운터의 설정값 및 현재값, 출력상태 등이 저장되며 RAM이 사용된다.

③ 입력부

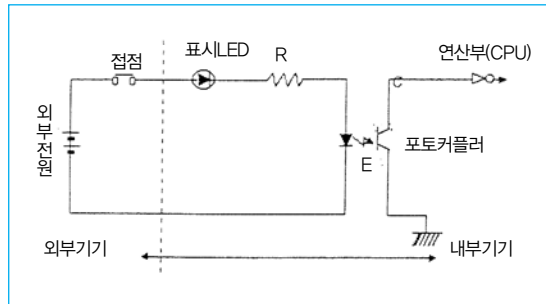
	<p>외부 입력기기 : 각종 신호를 CPU에 제공</p>
	<p>I/O모듈단자 : 외부기기와 PLC제어 시스템 사이의 연결을 담당</p>
	<p>입력신호 변환 : 외부기기의 신호를 CPU에 맞는 낮은 전위로 변환</p>
	<p>모듈상태 표시회로 : 입력모듈의 기능 상태를 가시적으로 표시하는 회로</p>
	<p>전기적 절연회로 : 외부신호와 CPU 간의 전기적 절연</p>
	<p>인터페이스(interface)/멀티플렉서(multiplexer) : 입력기기의 상태를 CPU에 전달해주는 장치</p>

〈그림 II-29〉 입력 모듈 블록도

입력부는 다양한 입력기기(각종 센서)의 신호를 PLC 내부에서 취급하는 신호로 변환하여 CPU에 전송하는 기능을 갖는다. 일반적으로 카드나 모듈(module) 형태로 구성되는데, 좀 더 자세한 내용은 〈그림 II-29〉과 같다.



위의 그림에서 절연회로는 외부기기의 전압과 CPU의 낮은 신호 전압값 사이를 절연하기 위한 회로로 트랜스포머(transformer)나 포토커플러(photocoupler) 등이 이용된다. <그림 II-30>에 개념적인 입력부 회로를 나타내었다.



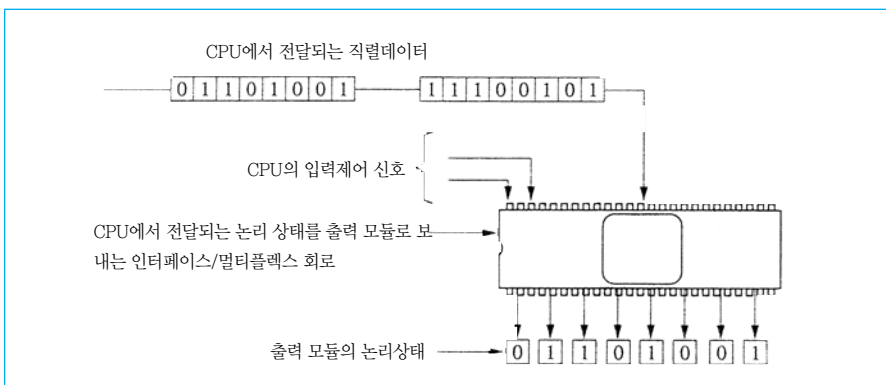
<그림 II-30> 입력부 회로

④ 출력부

PLC의 출력부 역시 모듈이나 카드 형태로 구성되며 입력부와 반대의 흐름을 갖는 동작 방식을 취한다. 먼저 <그림 II-31>의 인터페이스/멀티플렉스 회로는 CPU에서 나오는 신호를 받아 해석하여 적절한 출력점을 할당하는 역할을 한다.

멀티플렉서(multiplexer)

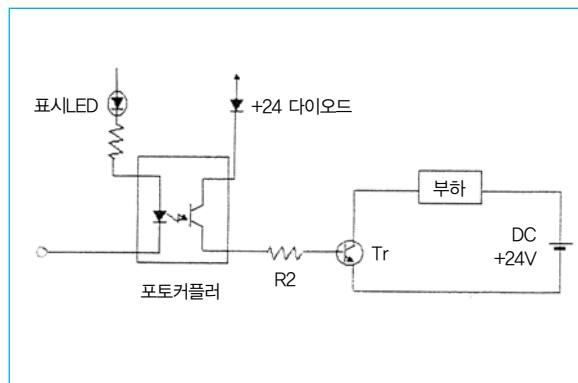
여러 개의 입력 중에서 한 개를 선택해 출력측에 전송하는 회로이다.



<그림 II-31> 인터페이스/멀티플렉서

다음의 래치(latch)회로는 인터페이스/멀티플렉서로부터 신호를 받아들여 다음 단이 준비될 때까지 신호를 저장하며, 일반적으로 직렬래치회로를 갖는다. 절연회로는 입력모듈의 회로와는 설계와 기능면에서 반대로 되어 있으며, 절연 방법은 입력부에서와 마찬가지로 트랜스포머, 포토커플러 등이 이용된다. 또한

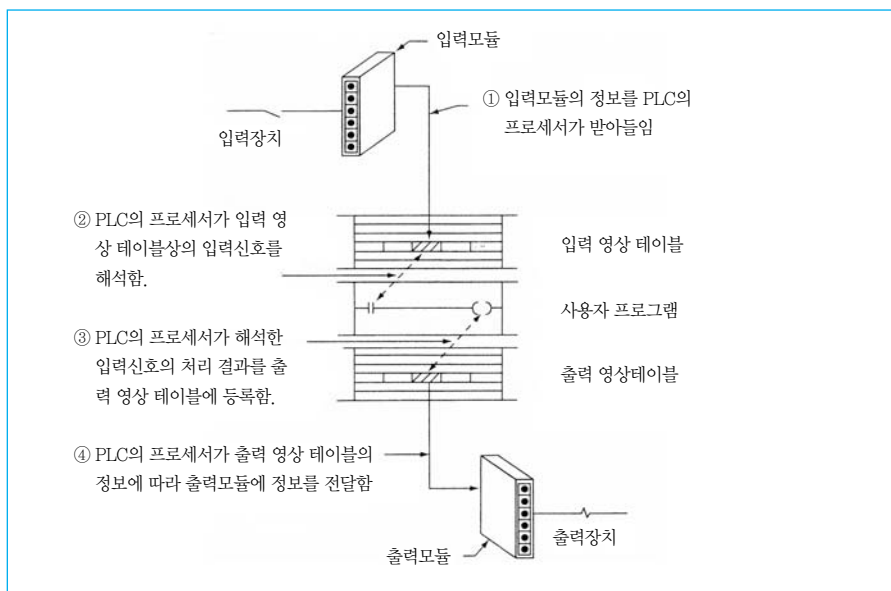
출력신호 변환회로는 절연회로에서 나오는 신호를 이용하여 외부기기를 동작시킬 수 있는 전류값으로 변환시키는 역할을 한다.



〈그림 II-32〉 출력부 회로

2) PLC 동작 원리

PLC 동작 원리를 설명하기 위해 사용자가 작성한 프로그램이 사용자 프로그램 메모리에 저장되어 있다고 가정하자.



〈그림 II-33〉 PLC의 동작 원리



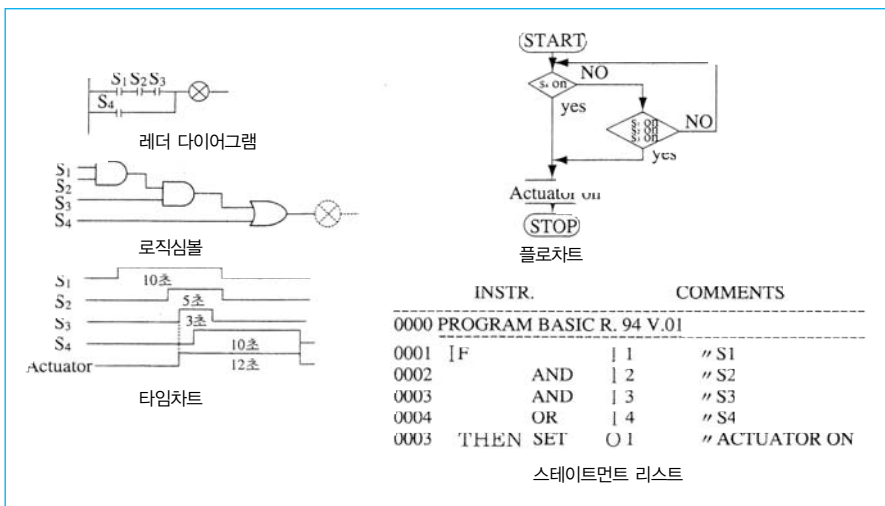
먼저 PLC가 리밋 스위치의 접점 등 입력기기의 상태를 입력받아 사용자 데이터 메모리인 입력 영상 테이블에 저장한다.

다음에 사용자가 작성한 프로그램을 입력 영상 테이블을 참조하여 차례대로 한 스텝씩 직렬 처리하여 프로그램을 실행하고, 그 결과를 다시 사용자 데이터 메모리에 있는 출력 영상 테이블에 저장하여 출력모듈로 정보를 전달하는 방식으로 동작된다.

PLC는 이렇게 입력 영상 테이블의 갱신, 프로그램의 실행, 출력 영상 테이블의 갱신을 한 주기로 동작되며, 이러한 실행 절차를 스캔 기능이라 한다. 이러한 스캔 기능은 PLC의 CPU가 실행 모드에 있는 동안 무한 루프를 돌며 계속된다.

따라서 한 번의 스캔 과정에서 입력신호 상태가 두 번 변경되면 PLC는 그 변화를 판독하지 못하며, 다음 번 스캔 과정에서 출력에 반영되게 되어 약간씩 시간 지연이 발생된다.

(3) 프로그래밍(programming)



〈그림 II-34〉 PLC에 사용되는 여러 가지 프로그래밍 언어의 예

제어 시스템이 올바른 작업을 수행하기 위해서는 사용자의 뜻대로 PLC가 움직여야 하는데, 이렇게 PLC가 제어되기 위해서는 PLC가 알아들을 수 있는 언어를 사용하여야 한다. 이처럼 기계가 알아들을 수 있는 명령어들을 순서나 상태에 맞추어 제어될 수 있도록 나열하는 것을 프로그래밍이라 한다.

한편 PLC의 특수한 작업 환경으로 인해 일반 컴퓨터에서 사용되는 프로그래밍 언어와는 달리, PLC에 사용되는 프로그래밍 언어는 현장 사람들이 손쉽게 다룰 수 있어야 하며, PLC의 하드웨어 구조에 대한 지식을 요구해서는 안 된다는 특징을 필요로 한다.

현재 이러한 조건을 만족시키는 프로그래밍 언어로는 래더 다이어그램(ladder diagram), 플로 차트(flow chart), 대수 방정식(boolean equation), 타임 차트(time chart), 기능 다이어그램(functional diagram) 방식 및 제작사마다 고유의 언어로 개발한 스테이트먼트 리스트(statement list) 방식이 있는데, 이 중에서 가장 많이 이용되고 있는 것이 래더 다이어그램 방식이다.

그 이유는 이 방식이 종래의 릴레이나 타이머에 의한 시퀀스 회로의 적용이 가능하여, 종래의 회로 설계 기술자나 릴레이 기호에 오랫동안 친숙한 현장 기술자에게 익숙하기 때문이다. 여기서는 가장 많이 사용하는 래더 다이어그램을 소개한다.

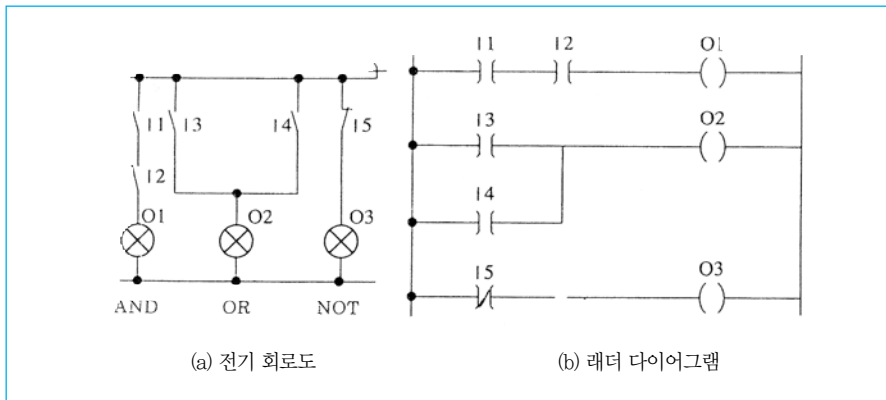
1) 래더 다이어그램(ladder diagram)

래더 다이어그램은 릴레이회로도를 표현한 것으로 전세계적으로 이용되고 있다. 래더 다이어그램은 실제의 사다리처럼 두 개의 수직선을 전원선으로 하여 구성하는데, 왼쪽 선은 전압 소스(voltage source)와 연결되고 오른쪽 선은 접지된다.

또한 두 수직선 사이에 놓여진 논리선(rung)은 제어회로의 여러 가지(branch)를 나타낸다. <그림 II-35>의 (b)는 래더 다이어그램을 나타낸 것으로,



첫 번째 논리선은 I1과 I2가 모두 논리 1의 값을 가질 때 출력 O1이 논리 1의 값을 갖는 것을 의미한다. 또한 마지막 논리선은 논리부정의 의미로 I5가 논리 0의 값을 가질 때 출력 O3가 논리 1의 값을 갖는 것을 의미한다. 일반적으로 래더 다이어그램에서는 논리선의 오른쪽 끝 부분에 항상 액추에이터를 넣는다.



〈그림 II-35〉 전기 회로도와 래더 다이어그램

단원 학습 정리



- F1** 로봇의 구조는 일반적으로 작업을 실행하기 위한 완전(腕的) 기구와 그 완전 기구를 작업 공간의 위치까지 이동시키기 위한 이동기구로 이루어진다.
- F2** 관절은 암과 암, 혹은 여러 개의 암으로 구성된 링크기구와 암을 연결 하는데 사용되며 구동기기인 액추에이터(actuator)에 의해 움직인다.
- F3** 이동기구를 갖는 로봇을 이동로봇이라 부르며, 여기에는 차륜형, 크롤러형, 다리식이 있다.
- F4** 액추에이터는 각종 에너지를 직선운동, 회전운동 등의 기계적 에너지로 변환하여 로봇의 관절을 구동시키는 역할을 하며, 제어회로의 제어에 의해 유기적인 동작을 실현한다.
- F5** 액추에이터는 변환되는 에너지원에 따라 유압 액추에이터, 공압 액추에이터, 전기 액추에이터로 나뉜다.
- F6** 공압 액추에이터의 제어 방법에는 순수공압 제어, 전기공압 제어(릴레이 제어), PLC 제어 등이 있다.



- 7** 전기공업 제어에 사용되는 기본 회로는 논리 회로와 자기 유지회로, 시간 지연회로, 인터록 회로이다.
- 8** PLC는 여러 가지 제어동작을 실행시키기 위해 제어 명령을 메모리에 저장하고, 이 메모리의 내용에 따라 제어 대상의 제어를 디지털 혹은 아날로그 입·출력을 통해서 행하는 디지털 조작형의 공업용 전자장치이다.
- 9** 릴레이와 PLC는 자동화 시스템에서 전체적인 제어를 행하는 마치 인간의 두뇌와 같은 역할을 하는 프로세서라는 점에서 그 기능은 같다.
- 10** PLC는 컴퓨터와 하드웨어적 기본구조는 차이점이 없으나 거기에 연결되는 입·출력 장치가 다르다.
- 11** PLC에 사용되는 프로그래밍 언어는 특수한 작업 환경으로 인해 일반 컴퓨터에서 사용되는 프로그래밍 언어와 다르다.

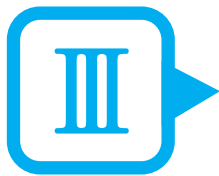


단원 종합 문제

- ▶ **1** 로봇의 완적기구에는 (), (), ()이 있다.
- ▶ **2** 로봇의 가동성을 나타내는 파라미터로서 ()가 이용되는데 이는 핸드의 위치와 자세를 바꾸기 위해 조작되는 변수의 수를 의미한다.
- ▶ **3** 다음 중 다리식 이동로봇 설계시 고려사항이 아닌 것을 고르시오?
 ① 안정성 ② 효율성 ③ 지면적응성 ④ 다리의수 ⑤ 운반능력
- ▶ **4** ()는 부하를 필요로 하는 위치까지 운동시키기 위하여 액투에이터의 구동에 필요한 전류를 발생시킨다.
- ▶ **5** 자동제어계의 제어방식을 크게 미리 정해진 순서에 따라 각 단계를 진행해 나가는 ()와 시스템의 출력을 입력단에 되돌려서 동작시키는 ()로 구분된다.
- ▶ **6** 다음 스위치 중 검출 스위치에 속하는 것을 모두 고르시오?
 ① 푸시버튼스위치 ② 로터리스위치 ③ 리밋스위치
 ④ 광전스위치 ⑤ 릴레이

- ▶ **7** 전기공압 제어에 사용되는 논리회로에서 입력 스위치 A, B의 NAND 회로를 구성해 보시오.(단 릴레이 출력은 Y로 표시)

- ▶ **8** 한시 동작 타이머와 한시 복귀 타이머의 b점점 회로를 그리고 각각에 대해 설명해 보시오.



로봇 기구 시스템

기구란 몇 개의 물체를 조합하여 기계를 구성하고, 그 중 어떤 물체에 힘을 가하여 운동을 일으키면 다른 물체도 그에 따라 일정한 운동을 하는 운동계를 말한다.

이 단원에서는 로봇에 쓰이는 기구를 익히고 여러 가지 기구와 작동 원리에 대하여 알아보기로 하자.



학습목표

1. 로봇 부품의 체결 방법을 설명할 수 있다.
2. 로봇의 운동 전달 특성을 설명할 수 있다.
3. 로봇 기구의 원활한 관절을 설계하는 방법을 설명할 수 있다.
4. 로봇 기구의 각 부분의 역할을 설명할 수 있다.
5. 로봇 기구를 이해하고 응용하는 방법을 탐색할 수 있다.

01

로봇의 기구 요소

엔코더(encoder) : 부호기라고도 한다. 입력신호에서 부호화한 출력 신호를 생성하는 전자 회로

로봇의 구성요소는 시스템의 전체 골격을 유지하는 요소로서 구조물, 운동 발생 장치, 상태의 측정 장치로 구분할 수 있다. 구조물에는 부품을 연결하는 체결용(나사, 볼트, 너트), 동력 전달용(기어, 벨트, 체인, 캠, 불나사) 등이 있으며, 운동 발생 장치는 회전운동(AC모터, DC모터, 스테핑 모터)과 직선운동(공압 실린더, 유압 실린더, 리니어 모터)을 발생시키는 액추에이터(actuator)를 의미한다. 센서는 상태량(변위, 압력, 회전량)을 측정하는 요소로서 엔코더, 리니어스케일, 레이저센서, 압력센서, 광량센서, 스위치 등이 있다.

1 부품의 체결

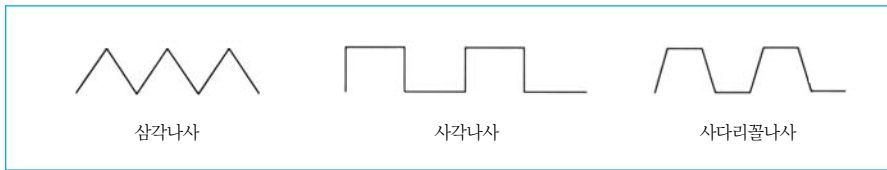
부품과 부품을 고정하기 위해서는 여러 가지 방법이 있으며 재료, 용도, 강도, 진동, 열팽창에 대한 영향 등을 충분히 고려하여 목적 달성에 가장 적합한 체결방법을 사용하여야 한다. 또한 부품의 고정 시에는 분해를 필요로 하는가에 따라서도 체결방법이 다르다. 사용 중에 고장이 발생하거나 수리를 필요로 하는 곳에는 부품의 분해가 필요하므로 볼트와 너트를 사용하여 체결하고, 분해할 필요가 없는 부품의 체결에는 리베팅 혹은 용접을 하여 체결한다.

(1) 나사

나사는 나선상의 홈을 이용하여 두 가지의 부품을 체결하는 것과 동력을 전

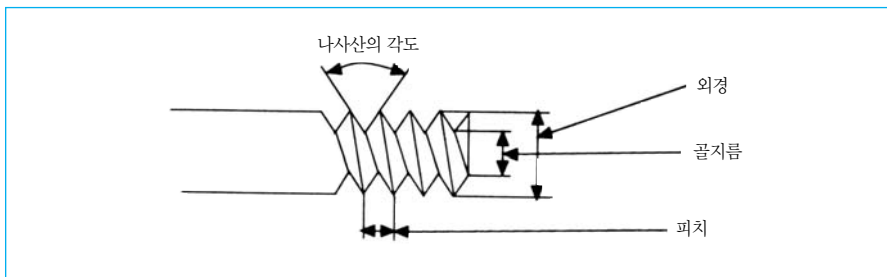


달하는 것으로 구분할 수 있다. <그림Ⅲ-1>과 같이 나사산의 모양에 따라 체결용과 동력 전달용으로 구분하여 사용한다. 체결용 나사는 산의 모양이 삼각형이고, 동력 전달용 나사의 경우는 산의 모양이 사각형 혹은 사다리꼴의 형태를 가진다.



<그림Ⅲ-1> 양 솔레노이드를 이용한 제어회로도

삼각나사는 ISO(국제표준화기구) 규격에 의해 호환성을 유지하며, 인치계열과 미터계열로서 분류하고 주로 부품의 체결용으로 사용된다.



<그림Ⅲ-2> 나사의 명칭

<그림Ⅲ-2>는 나사의 명칭을 나타내고 있다. 나사의 리드 l 은 나사가 1회전할 경우에 이동하는 거리로서 나타낸다.

$$l = np$$

$$\dots (\text{Ⅲ-1})$$

솔레노이드(Solenoid)

등근 대롱모양으로 감은 코일이다. 전류를 흘리면 전자석이 된다.

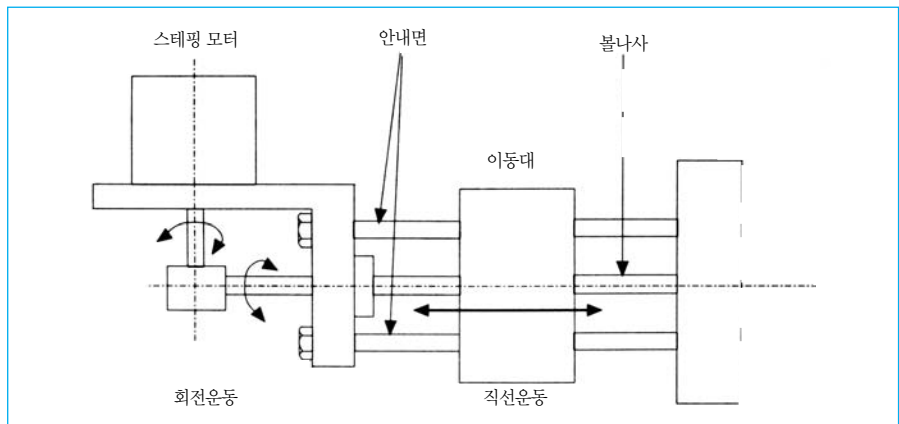
여기서, n 은 나사의 줄 수를 나타내며, p 는 나사의 피치이다.

[예제Ⅲ-1] 나사의 피치가 $1.25[mm]$ 이고 2줄 나사를 가진 삼각나사의 리드는 얼마인가?

(풀이) 리드를 구하는 공식은 $l = np$ 이므로 $p = 1.25$, $n = 2$ 를 대입하면 된다

$$l = np = 2 \times 1.25 = 2.5[mm]$$

<그림Ⅲ-3>은 볼나사를 사용하여 스텝핑 모터의 회전운동을 직선운동으로 변환하는 운동 기구(mechanism)를 나타내고 있다. 스텝핑 모터에 의해 발생된 회전운동은 기어의 조합에 의해 구성된 방향 전환 기구를 통하여 회전운동의 방향이 전환된다. 방향이 변환된 회전운동은 볼나사를 통하여 직선운동으로 변환된다. 직선운동은 이동대의 양쪽에 구축되어 있는 안내 면에 의해 운동방향이 제약되고, 이동대의 이동거리는 스텝핑 모터의 회전에 따라 회전하는 볼나사의 리드에 제약을 받고, 이동대의 위치 정밀도는 스텝핑 모터의 분해능, 볼나사의 리드, 직선운동의 안내면 마찰력의 크기에 제약을 받는다.



<그림Ⅲ-3> 스텝핑 모터에 의한 직선운동 기구



스테핑 모터의 분해능을 R_s [°/펄스]라고 하고, 볼나사의 리드는, 볼나사의 피치는 p [mm], 이동대의 이동거리를 L [mm]이라고 한다. 스텝핑 모터의 목표치를 펄스량으로 입력할 경우에 이동대의 이동거리와 이동대의 최소 이동거리 P_a [mm]를 구한다. 먼저, 스텝핑 모터가 1회전할 경우에 발생하는 총 펄스량 N [펄스]은 다음과 같다.

$$N = \frac{360}{R_s} \quad \dots \text{(III-2)}$$

스테핑 모터의 입력을 R_N [펄스]이라고 하면, 이동거리는 다음의 식에 구한다.

$$L = \frac{l}{N} R_N = \frac{np}{N} R_N \quad \dots \text{(III-3)}$$

이동대의 최소 이동거리는 다음과 같다.

$$P_a = \frac{l}{360} R_s \quad \dots \text{(III-4)}$$

단, 안내면의 마찰력을 무시한다.

[예제Ⅲ-2] 그림3-3의 직선운동 기구에 있어서 스텝핑 모터의 분해능은 1.8[°/펄스], 피치가 5[mm], 나사산의 줄 수가 1인 볼나사를 사용하며, 안내면의 마찰력은 무시한다. 테이블의 최소 이동거리(위치 정밀도)를 구하고, 스텝핑 모터에 400[펄스]를 입력할 경우에 이동대가 이동하는 거리를 구하라.

(풀이) ① 스텝핑 모터의 1펄스 회전하였을 경우에 이동대가 움직이는 최소 단위의 이동거리는 다음과 같다.

펄스(pulse)

신호의 진폭이 정상상태의 값에서 단시간에 다른 값으로 전이하여, 유한 시간만 계속한 후 다시 본래의 값으로 되돌아 가는 파형

$$P_a = \frac{l}{360} R_s = \frac{1 \times 5}{360} \times 1.8 = 0.025[\text{mm}]$$

② 스테핑 모터가 1회전할 경우에 발생하는 총 펄스량은

$$N = \frac{360}{R_s} = \frac{360}{1.8} = 200[\text{펄스}]$$

이므로, 이동거리는

$$L = \frac{l}{N} R_N = \frac{np}{N} R_N = \frac{1 \times 5}{200} \times 400 = 10[\text{mm}]$$

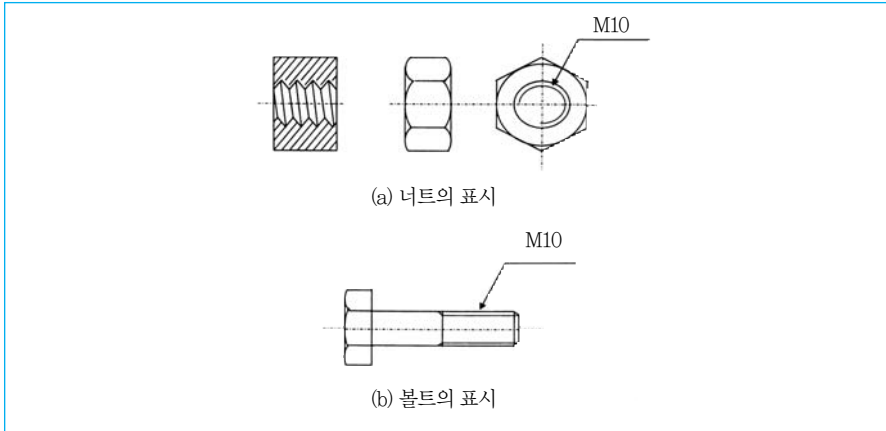
가 된다.

(2) 나사에 의한 부품 체결

나사 체결부는 볼트와 너트로 구성된 나사의 체결부품과 피체결부품으로 구성되며, 이것이 일체가 되는 것을 체결기능을 만족한다고 한다. 볼트와 너트는 두 개의 부품을 체결하기 위하여 사용되는 요소로서 기계의 구조물 형성에 많이 사용된다.

〈그림Ⅲ-4〉는 볼트와 너트를 나타내는 기호이다. 크기를 표시할 경우에는 〈그림Ⅲ-4〉에서 나타낸 것처럼, 미터나사의 경우에는 M10(지름:10[mm])으로 표시하고, 화살표는 외경을 나타내는 선에 60[°]의 각도로 입사하는 형태로 나타낸다.

나사 체결부의 강도적인 신뢰성을 확보하기 위하여 필요한 3대 신뢰성은 다음과 같다.

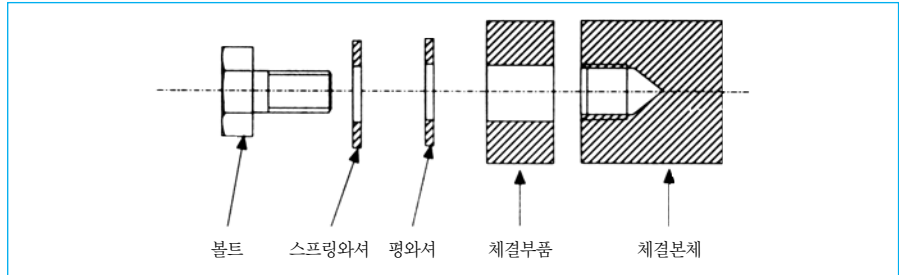


〈그림Ⅲ-4〉 볼트와 너트

- 체결부품의 신뢰성(볼트와 너트)
- 강도설계의 신뢰성(나사 체결부)
- 체결력 관리의 신뢰성

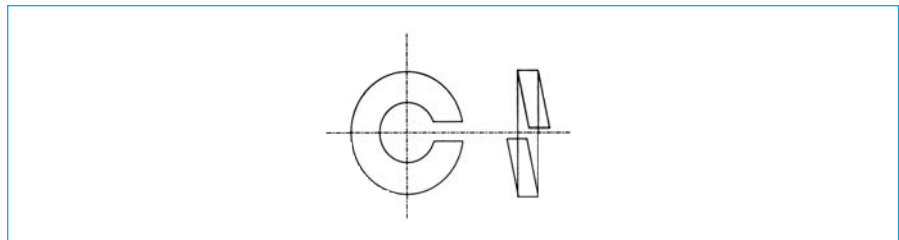
볼트와 너트가 결함이 없는 양호한 부품의 선정이 중요하며, 체결부의 설계 시에는 사용 중에 작용하는 외력을 충분히 고려하여 재료와 두께를 설계하여야 한다. 체결력의 관리는 조립 현장에서 고려해야 하는 중요한 관리요소로서 특수한 경우에는 체결 방법, 체결을 위한 토크를 사전에 지시할 필요가 있다.

〈그림Ⅲ-5〉는 볼트를 사용하여 체결하는 체결부품을 체결본체에 고정시키는 사례를 나타낸다. 체결본체에는 드릴을 사용하여 충분한 크기의 구멍을 가공하고, 나사를 만들 수 있는 탭(tab)이라는 나사 형성공구를 사용하여 나사산을 만든다. 체결 부품에는 볼트의 호칭 크기보다 약간 크게, 즉 볼트가 충분히 통과할 수 있도록 드릴을 사용하여 구멍을 가공한다. 와서는 체결부품과 볼트의 사이에 삽입되어 부품의 체결을 확실하게 도와주는 역할을 하며, 스프링와서와 평와서로 분류할 수 있다.



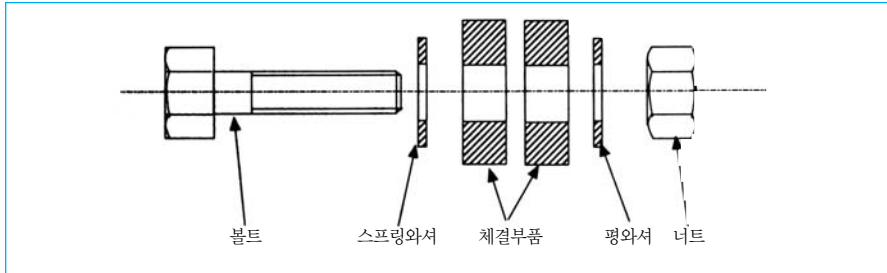
〈그림Ⅲ-5〉 볼트에 의한 부품의 체결 방법

스프링와셔는 고정을 확실하게 하는 동시에 스프링의 역할을 하도록 〈그림Ⅲ-6〉과 같은 형상을 가지고 있으며, 나사가 풀리는 현상을 방지하는 역할을 한다. 평와셔는 부품의 체결부위가 평면이 아닐 경우에 체결력을 일정하게 하여 풀리는 현상이 일어나지 않도록 하기 위하여 사용한다.



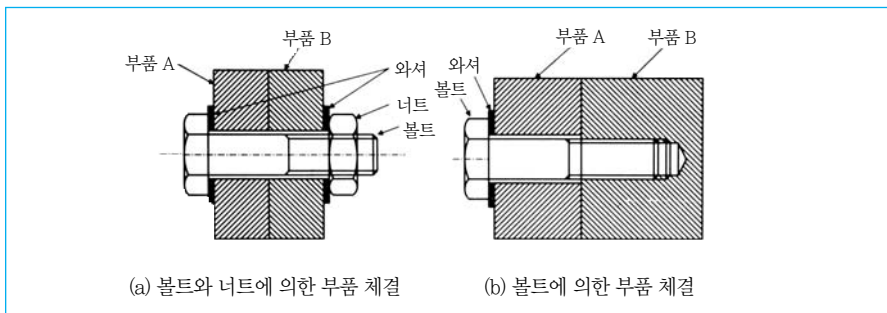
〈그림Ⅲ-6〉 스프링와셔

〈그림Ⅲ-7〉은 볼트와 너트를 사용하여 두 개의 부품을 체결하는 방법을 나타낸다. 먼저 드릴을 사용하여 체결할 부품에 볼트의 직경보다도 약간 크게 구멍을 가공한다. 스프링와셔를 볼트에 삽입한 상태에서 체결부품에 볼트를 끼워 넣는다. 볼트를 체결부품에 끼운 상태에서 평와셔를 볼트의 나사산이 있는 쪽에 끼우고 너트를 사용하여 고정을 시킨다.



〈그림 Ⅲ-7〉 볼트와 너트에 의한 부품의 체결방법

〈그림 Ⅲ-8〉은 두 개의 부품을 볼트를 사용하여 체결한 경우와 볼트와 너트를 사용하여 체결한 경우를 나타낸다.



〈그림 Ⅲ-8〉 두 개의 부품의 체결방법

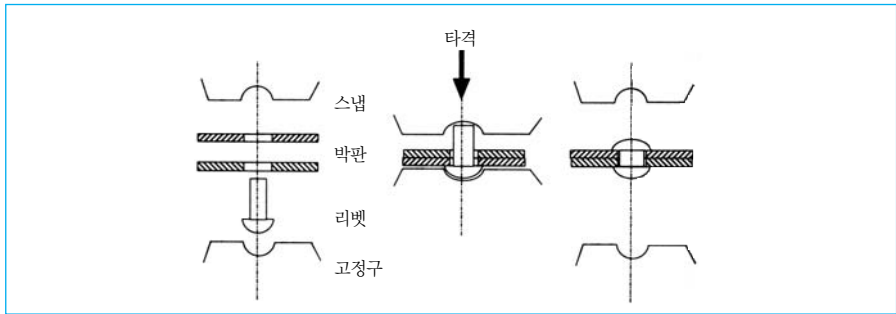
(3) 리벳에 의한 부품 체결

분해할 필요가 있는 부품을 체결할 경우에는 볼트를 이용하여 체결하거나 볼트와 너트를 사용하여 체결한다. 분해할 필요가 없는 경우에는 리벳을 사용하거나 용접 등과 같은 방법으로 두 부품을 결합한다. 두꺼운 철판과 같은 경우에는 금속을 녹여서 붙이는 전기용접을 사용하여 결합하고, 얇은 판의 경우에는 가스용접 혹은 리벳으로 고정을 한다.

〈그림 Ⅲ-9〉는 리벳을 사용하여 부품을 고정하는 방법을 나타낸다. 접합하기 위한 두 개의 얇은 판에 구멍을 내고 리벳을 구멍에 끼운다. 고정구에 리벳의

머리를 올려놓고 고정된 상태에서 리벳의 머리형상을 가진 스냅을 리벳의 위에 고정시킨다.

스냅을 망치와 같은 것으로 타격함으로써 머리를 만들어 고정시킨다. 이러한 리벳에 의한 부품의 접합 방법은 분해할 필요가 없으며 강도가 요구되는 곳에 사용되는데 주로 보일러, 선박, 비행기, 다리 등을 제작할 때 외관 및 철골의 고정에 많이 사용한다.



〈그림 Ⅲ-9〉 리벳에 의한 부품의 체결방법

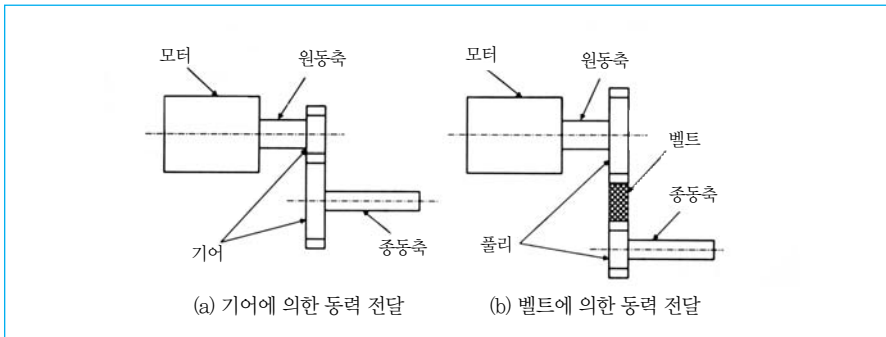
2 동력의 전달

동력은 일반적으로는 회전운동의 전달을 나타내는데, 회전운동을 전달하는 가장 기본적인 기계요소는 축이다. 모터 혹은 엔진 등에 의해 발생된 회전운동은 축에 의해 다른 동작부분으로 전달되며, 동력의 전달방법으로는 〈그림 Ⅲ-10〉과 같이 기어 및 벨트 등을 사용한다.

동력을 전달하는 두 개의 축간 거리가 짧은 경우에는 기어를 사용하여 동력을 전달하고, 축간 거리가 긴 경우에는 벨트를 사용한다. 자동화 기계에서는 주로 타이밍벨트를 많이 사용한다. 회전운동을 직선운동으로 변화시키는 경우에



는 볼나사를 사용하여 운동을 변화시킨다. 모터에 의한 회전운동은 원동축을 통하여 원동축에 고정되어 있는 기어 혹은 풀리에 전달된다. 기어의 잇수 혹은 풀리의 직경비에 따라 회전비가 결정되어 원동축의 회전운동을 종동축에 전달한다.



〈그림 Ⅲ-10〉 동력의 전달 방법

(1) 기어에 의한 동력 전달

동력을 확실하게 전달하는 방법으로서 기어를 많이 사용한다. 시계와 같은 작은 기계에서 선박, 항공기 등과 같은 대형 기계에 이르기까지 사용하고 있으므로, 기어는 응용범위가 넓고 종류도 다양하다. 동력을 전달하는 기어의 특징은 다음과 같다.

- 모터의 회전운동을 확실하게 전달한다.
- 기어 잇수의 비에 따라 회전수 조정(가감속)이 용이하다.
- 기어축의 각도를 가지고 회전운동 전달(베벨기어)을 할 수 있다.
- 높은 감속비(웜기어)를 얻을 수 있다.

〈그림 Ⅲ-11〉은 베벨기어와 웜기어를 나타낸다. 베벨기어는 회전운동의 각도를 90[°]로 바꾸는 곳에 사용하며, 웜기어는 감속비가 커서 주로 감속기에 사

주축(Main shaft)

원동기에서 직접 동력을 받는 축을 말하며 원동축이라고도 한다.

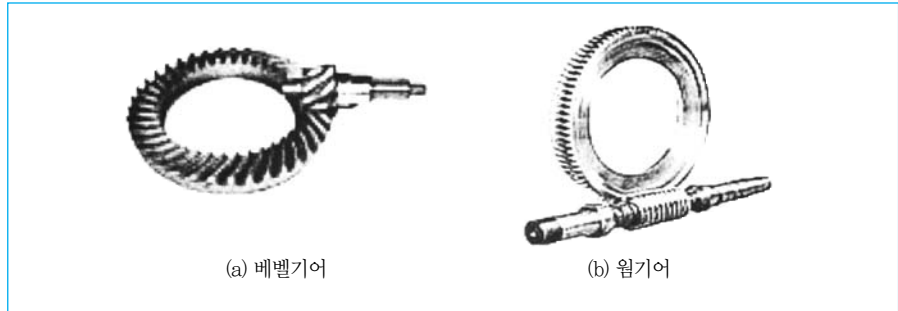
선축(Line shaft)

주축에서 동력을 받아서 각 공장에 분배하는 역할을 하는 축(종동축)

중간축(Counter shaft)

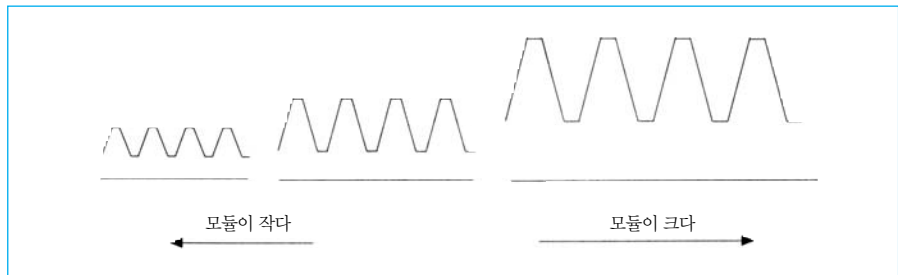
선축에서 동력을 받아서 각각의 기계에 필요한 속도와 방향을 조정해서 동력을 전달시키는 축

용한다.



〈그림Ⅲ-11〉 베벨기어와 위엄기어

〈그림Ⅲ-12〉는 기어의 크기를 표시하는 모듈(module)과 이의 크기를 나타낸다. 표준 기어는 이끝 높이와 모듈의 크기가 같으며, 기어의 이가 작으면 전달하는 동력의 힘은 약하고, 기어의 이가 크면 전달하는 힘이 강하다.



〈그림Ⅲ-12〉 모듈과 이의 크기

1) 이의 크기

이의 크기는 원주 피치, 모듈, 지름 피치의 세 가지의 종류를 기준으로 하고 있다. 지름 피치는 인치를 사용하는 나라에서 많이 사용되어 여기서는 생략한다.



① 원주 피치(circular pitch ; p)

피치원 둘레를 잇수로 나눈 값, 즉

$$p = \frac{\pi D_p}{Z} = \pi m \quad \dots \text{(III-5)}$$

여기서, Z : 잇수

D_p : 피치원 지름 (mm)

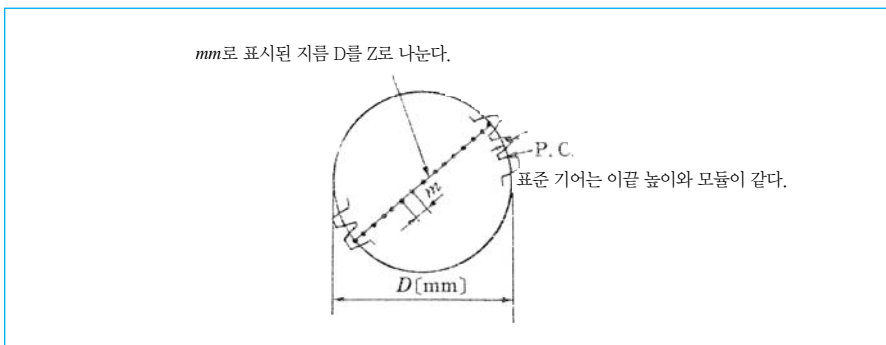
m : 모듈(module)

② 모듈(module)

피치원 지름(D_p)을 잇수(Z)로 나눈 값이며, 오늘날 미터(meter) 방식을 사용하고 있는 나라에서 이의 크기로 대부분 사용되고 있다.

피치원의 지름을 이용하여 모듈을 구하는 식은 다음과 같다.

$$m = \frac{D_p}{Z}, D_p = mZ, m = \frac{p}{\pi} \quad \dots \text{(III-6)}$$



<그림 III-13> 원주 피치와 모듈

기어의 외경을 사용하여 모듈 m 을 구하는 방법은 다음과 같다.

- 기어의 외경을 버니어캘리퍼스와 같은 측정기를 사용하여 계측한다.
- 기어의 잇수를 확인한다.
- 계산식을 사용한다.

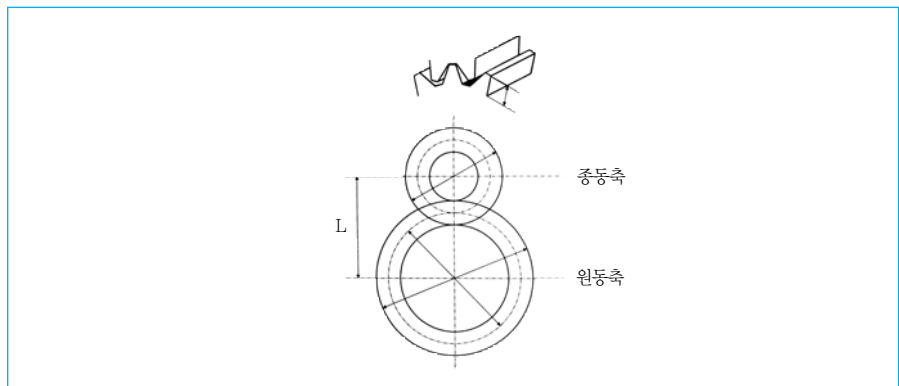
〈그림Ⅲ-14〉에서 표준 기어는 이끝 높이와 모듈의 크기 같기 때문에 피치원 지름 D_p 에 잇수를 2개 더하면 외경 D_1, D_2 를 구할 수 있다.

$$\text{따라서, } m = \frac{D_1}{Z_1+2} = \frac{D_2}{Z_2+2} \quad \dots (\text{Ⅲ-7})$$

여기서 Z 는 기어의 잇수를 나타내고, 첨자 1은 원동축을, 첨자 2는 종동축을 나타낸다.

2) 기어의 축간거리를 구하는 공식

기어의 축간거리를 구하기 위하여서는 모듈을 먼저 구하여야 한다. 〈그림Ⅲ-14〉은 원동축과 종동축 기어가 맞물려 있는 상태를 나타낸다. 기어의 축간거리를 $L[mm]$, 기어의 외경을 $D_1, D_2[mm]$, 피치원의 지름을 $D_p[mm]$ 라고 한다.



〈그림Ⅲ-14〉 기어의 축간거리



원동축과 종동축의 축간거리를 구하는 식은 다음과 같다.

$$L = \frac{m(Z_1+Z_2)}{2} \text{ [mm]} \quad \dots \text{ (III-8)}$$

[예제 III-3] 원동축의 기어 잇수는 30, 기어의 외경이 128[mm], 종동축의 기어 잇수가 20인 경우에, 이의 크기와 종동축의 외경, 축간거리를 구하여라.

(풀이) 기어의 모듈을 먼저 구한다.

$$m = \frac{D_1}{Z_1+2} = \frac{128}{32} = 4$$

기어가 맞물리기 위하여서는 모듈이 같아야 하므로 종동축의 외경은,

$$D_2 = m(Z_2+2) = 4 \times 22 = 88$$

이다. 축간거리는 다음과 같다.

$$L = \frac{m(Z_1+Z_2)}{2} = \frac{4(30+20)}{2} = 100 \text{ [mm]}$$

(2) 기어의 속도비(감속비, 기어비)를 구하는 공식

기어를 사용하여 동력을 전달하는 방법에 있어서 대부분이 감속 혹은 가속의 기능을 수행한다.

원동축과 종동축의 속도비 i 는 다음의 식으로 나타낸다.

$$i = \frac{N_2}{N_1} = \frac{D_1}{D_2} = \frac{Z_1}{Z_2} \quad \dots \text{(III-9)}$$

여기서 N 은 회전속도, D 는 기어의 외경, Z 는 기어의 잇수를 나타낸다.

[예제 III-4] 원동축의 고정되어 있는 기어의 모듈이 3이고, 기어 잇수가 30인 기어가 있다. 속도비가 0.5인 경우에 원동축의 회전속도 $N_1 = 280[\text{rpm}]$ 으로 설정하였다.

- (1) 종동축에 고정되어 있는 기어의 회전수는 얼마인가?
- (2) 종동축에 고정되어 있는 기어의 잇수는 얼마인가?
- (3) 원동축과 종동축의 기어의 외경은 얼마인가?
- (4) 원동축과 종동축의 사이의 축간거리는 얼마인가?

(풀이) (1) 속도비를 구하는 공식을 이용하면 종동축의 회전수는 다음과 같다.

$$N_2 = i \times N_1 = 0.5 \times 280[\text{rpm}] = 140[\text{rpm}]$$

(2) 종동축의 기어의 잇수는 다음과 같다.

$$Z_2 = \frac{Z_1}{i} = \frac{30}{0.5} = 60$$

(3) 기어의 외경을 구하기 위하여서는 모듈과 기어의 잇수를 사용한다.

$$D_1 = m(Z_1 + 2) = 3 \times 32 = 96[\text{mm}]$$



$$D_2 = m(Z_2 + 2) = 3 \times 62 = 186[\text{mm}]$$

(4) 기어의 축간거리는,

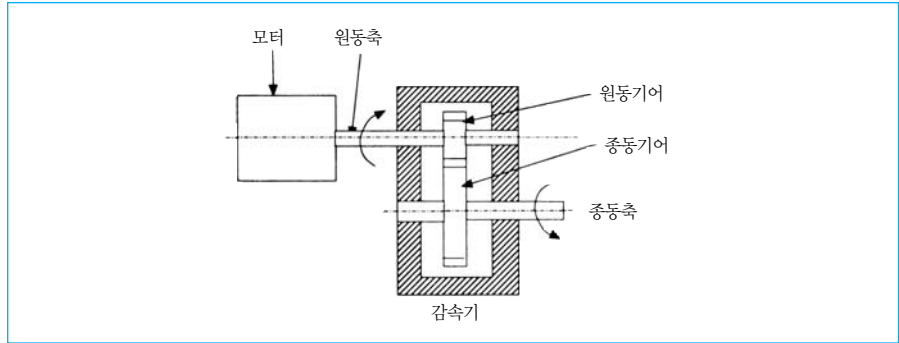
$$L = \frac{m(Z_1 + Z_2)}{2} = \frac{3(30 + 60)}{2} = 135[\text{mm}]$$

이다.

(3) 감속기를 사용하여 동력을 전달하는 방법

여러 종류를 조합함으로써 원동축과 종동축의 회전속도를 변화시키고 아울러 전달하고자 하는 동력의 크기를 변화시키는 장치를 감속기라 한다. 필요에 따라서 특수하게 설계 제작하는 경우도 있으나, 최근에는 모터에 감속기가 달려있는 감속기 부착형 모터(g geared moter)가 개발되어 시판되고 있다. 또한 감속기 자체만으로도 상품화되어 있으므로 목적에 맞는 감속기를 선정하는 것이 중요하다.

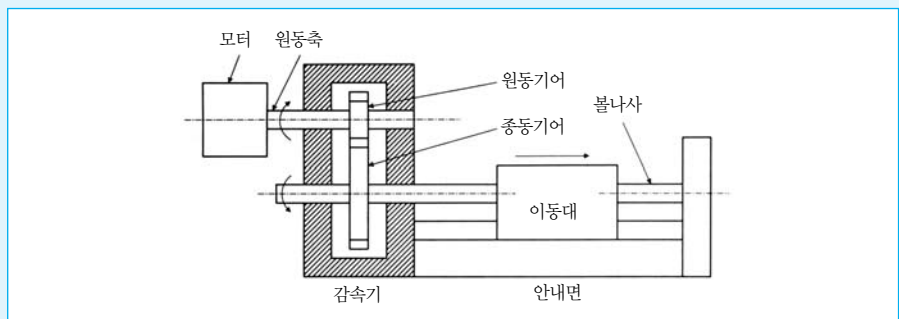
〈그림 Ⅲ-15〉는 간단한 감속기의 예를 나타낸다. 동력원으로 사용되는 모터는 고속회전이며 낮은 토크(torque)를 가지고 있다. 모터의 회전운동은 원동축을 통하여 원동기어에 전달되고, 원동기어는 고속회전의 낮은 토크를 종동기어에 전달한다. 원동기어보다 기어의 잇수가 많고 외경이 큰 종동기어에 의해, 고속회전은 저속회전의 높은 토크를 가진 동력으로 변환된다. 따라서 원동기어와 종동기어의 기어비(감속비)를 크게 하면, 회전속도는 저하되나 강한 토크를 얻을 수 있다. 기어가 들어 있는 감속기의 기어박스는 소음과 진동을 억제하기 위하여 윤활유를 사용하고 있으므로, 윤활유의 누출이 없도록 기어박스를 설계한다. 종동축에는 기계장치 등에 연결되어 소정의 작업을 수행한다.



〈그림 Ⅲ-15〉 감속기의 개략적인 구조

[예제 Ⅲ-5] 〈그림 Ⅲ-16〉과 같이 회전운동을 직선운동으로 변환하는 이송 기구가 있다. 동력원으로서 분해능 $1.8[^\circ/\text{펄스}]$ 를 가진 스텝핑 모터를 사용하고, 원동기어의 모듈은 $m=3$, 잇수는 $Z_1=30$ 이고, 속도비(감속비)가 $i=0.05$ 이다. 볼나사는 2줄 나사로서 피치가 $p=5[\text{mm}]$ 를 선정하였다. 안내면의 마찰력은 무시해도 좋은 정도로 작다고 가정한다.

1. 스텝핑 모터에 1펄스를 입력하였을 경우에 이동대가 움직이는 최소 이동거리 P_a 를 구하여라.
2. 스텝핑 모터에 500펄스를 입력하였을 경우에 이동대의 이동거리 L 을 구하라.



〈그림 Ⅲ-16〉 모터에 의한 직선 이송 기구



(풀이) 1. 이동대의 최소 이동거리 P_a 는 스테핑 모터의 분해능 R_s , 속도비 i , 볼나사의 리드 l 을 사용하여 구한다.

$$P_a = \frac{i l R_s}{360} = \frac{0.05 \times 2 \times 5 \times 1.8}{360} = 0.0025 [mm] = 2.5 [\mu m]$$

따라서 스테핑 모터에 1펄스를 입력하면 $2.5[\mu m]$ 이동하고, 안내면의 마찰력과 감속기의 백래시(backlash)의 영향을 무시하는 이론적인 정밀도를 나타낸다.

2. 스테핑 모터에 500펄스를 입력하였을 경우에 이동대의 이동거리 L 은 스테핑 모터의 입력펄스와 속도비 i , 볼나사의 리드 l , 스테핑 모터가 1회전할 경우의 총 펄스량 N 을 사용하여 구한다.

먼저 스테핑 모터가 1회전할 경우의 총 펄스량은

$$N = \frac{360}{R_s} = \frac{360}{1.8} = 200 [\text{펄스}]$$

이다. 따라서 이동거리는 다음과 같다.

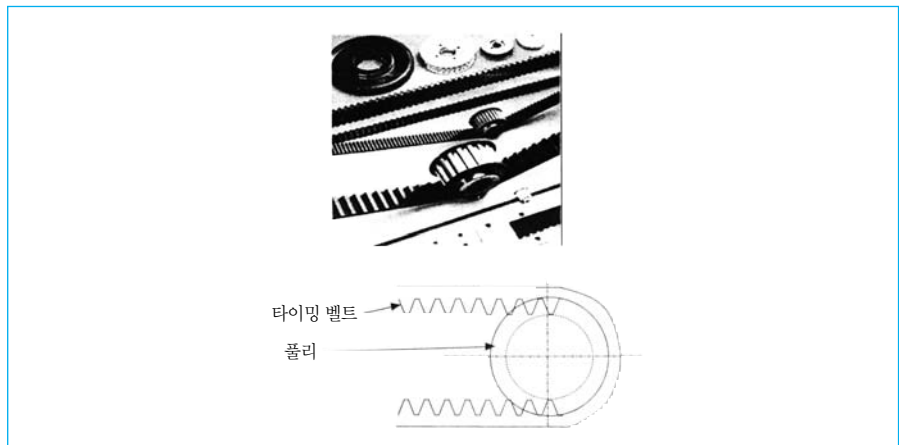
$$L = \frac{i l R_n}{N} = \frac{0.05 \times 2 \times 5 \times 500}{200} = 1.25 [\mu m]$$

(4) 벨트에 의한 동력 전달

원동축과 종동축이 인접할 경우에는 기어를 사용하여 동력을 전달하는 것이 효율적이나, 축간거리가 상당한 거리를 유지하고 있을 경우에는 벨트를 사용하거나 체인을 사용하여 동력을 전달한다. 금속체인을 사용하는 동력 전달 방법

은 스프로킷을 사용함으로써 기어와 같이 확실하게 동력을 전달하는 기능을 가진다. 하지만 큰 힘을 전달하는 이점은 가지고 있으나, 소음이 많고 고속 회전이 어려운 단점이 있다. 벨트에 의한 동력 전달방법은 체인에 비하여 소음이 적으며 고속 회전이 가능하나, 풀리와 벨트가 마찰 구조를 가지므로 큰 힘의 전달에 불리하다.

평벨트를 사용하는 경우에는 상당한 축간거리가 있을 경우에 유리하므로 주로 정미소 등에서 사용한다. 비교적 축간거리가 짧은 경우에는 V홈 벨트, 타이밍(timing)벨트를 사용한다. V홈 벨트는 주로 농업기계에 적용하고 있으며, 타이밍 벨트는 산업용 기계에 많이 적용되고 있다.



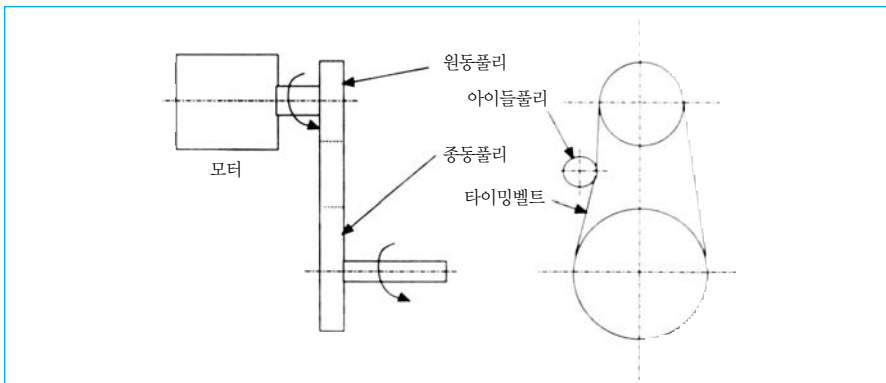
〈그림Ⅲ-17〉 타이밍 벨트와 풀리의 관계

〈그림Ⅲ-17〉은 타이밍 벨트를 사용하여 동력을 전달하는 형태를 나타내며, 타이밍 벨트가 풀리에 결합되어 있는 상태를 나타낸다. 풀리에는 평기어와 같은 기어홈이 만들어져 있고, 타이밍 벨트에도 풀리와 같은 이의 크기(모듈이 같음)를 고무와 같은 재료를 사용하여 만든다. 타이밍 벨트의 홈과 풀리의 홈에 의하여 미끄럼이 없이 확실하게 동력을 전달할 수 있다.

〈그림Ⅲ-18〉은 타이밍 벨트를 사용하여 동력을 작업축에 전달하는 간단한 구

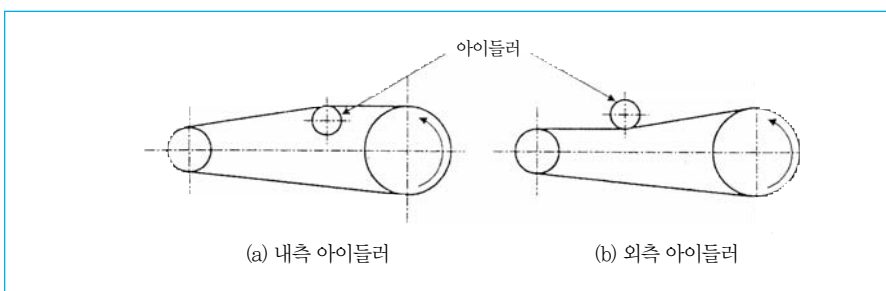


조를 나타낸다. 모터에 의해 발생한 회전력은 원동폴리를 통하여 종동폴리에 전달되는 구조를 취하여, 타이밍 벨트를 폴리에 쉽게 걸도록 원동축 혹은 종동축 중의 어느 한쪽에 축간거리의 조정이 가능한 구조를 가지도록 설계하여야 한다. 벨트의 장력 조절을 위하여서는 아이들(idle) 폴리를 별도로 사용하는데, 원동축과 종동축의 중심선과 평행을 유지하도록 설치하여야 벨트가 벗겨지는 현상을 방지할 수 있다.



〈그림Ⅲ-18〉 타이밍 벨트에 의한 동력의 전달 메커니즘

〈그림Ⅲ-19〉은 타이밍 벨트의 장력을 조정하기 위하여 사용하는 아이들러의 위치를 나타낸다. 아이들러를 내측에 설치하는 방법과 외측에 설치하는 방법이 있으며, 위치가 다르다는 것에 주의하여 설계하여야 한다.



〈그림Ⅲ-19〉 아이들러의 위치

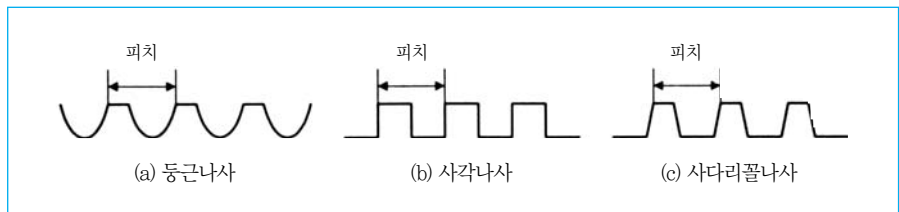
(5) 볼나사에 의한 동력전달

회전운동을 직선운동으로 변화시키는 기계요소는 볼나사이다. <그림 III-20>은 볼나사를 나타내며 너트가 조합된 상태를 나타낸다. 볼나사의 양 끝부분에는 베어링을 사용하여 지지하고, 너트에는 이동하는 테이블을 고정한다. 볼나사의 회전에 의해 너트는 직선이동을 하게 된다.



<그림 III-20> 볼나사

<그림 III-21>은 동력 전달용으로 사용되는 나사의 형태를 나타내며, 볼나사, 사각나사, 사다리꼴나사가 있다.

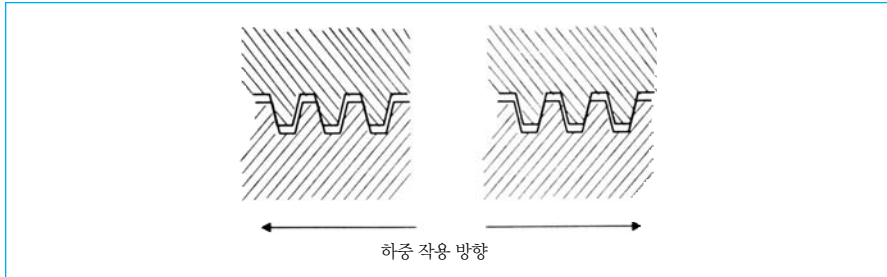


<그림 III-21> 동력 전달용 나사의 종류

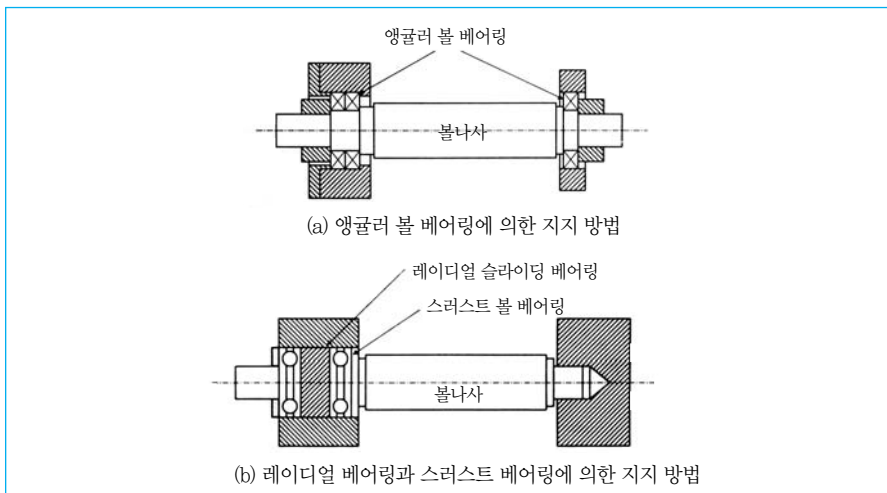
백래시(Backlash)
부품간의 헐거움으로 인하여 역회전시 헛도는 것

<그림 III-22>는 동력의 전달방향에 따라서 발생하는 백래시(backlash)가 일어나는 형태를 나타낸다. 화살표는 하중의 작용방법을 나타낸다. 볼나사를 사용하여 이동 테이블을 설계하면 백래시에 기인한 위치오차가 발생하므로, 기계의 설계 시에는 백래시를 고려하여 위치 정밀도에 맞는 부품의 선정이 중요하다.

<그림 III-23>은 볼나사를 지지하는 방법을 나타낸다. 볼나사의 양 끝부분에는 베어링을 사용하여 조립이 가능하도록 설계하여야 한다.



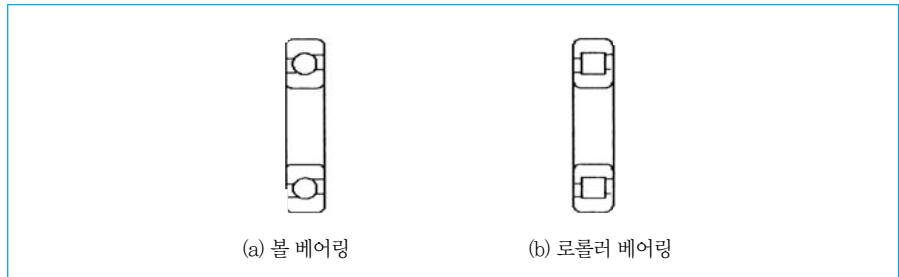
〈그림 Ⅲ-22〉 백래시



〈그림 Ⅲ-23〉 볼나사의 지지 방법

3 베어링

회전축을 지지하는 기계적인 요소를 베어링(bearing)이라고 하고, 축이 베어링에 의해 지지되는 부분을 저널(journal)이라고 한다. 베어링과 저널은 상대운동을 하므로 경계부분에서 마찰이 발생하며, 동력의 손실과 열 발생의 원인이 된다. 마찰을 감소시키고 발열을 억제하기 위하여 윤활제를 사용한다.



〈그림 Ⅲ-24〉 볼 베어링과 롤러 베어링

〈그림 Ⅲ-24〉는 볼 베어링과 롤러 베어링을 나타낸다. 베어링은 축과 베어링이 면 접촉으로 미끄럼 운동을 하는 미끄럼 베어링(sliding bearing)과 축과 베어링이 점 접촉으로 구름운동을 하는 구름 베어링(rolling bearing)으로 구분된다. 또한, 대량생산되는 정밀부품으로서 각종 산업기계, 운반용 기계, 가정용 기계 및 농업기계 등에 많이 사용되는 중요한 부품이다. 베어링의 종류 또한 수십 가지가 되며, 각 베어링마다 특징을 가지고 있고 적용대상도 다르다. 사용 조건에 따라 최적의 베어링을 선정하는 것이 로봇 설계에서 중요하다. 특히 베어링의 수명은 제품의 신뢰성에 중대한 영향을 미치는 요소이므로 신중하게 선정하여야 한다.

(1) 저널의 종류

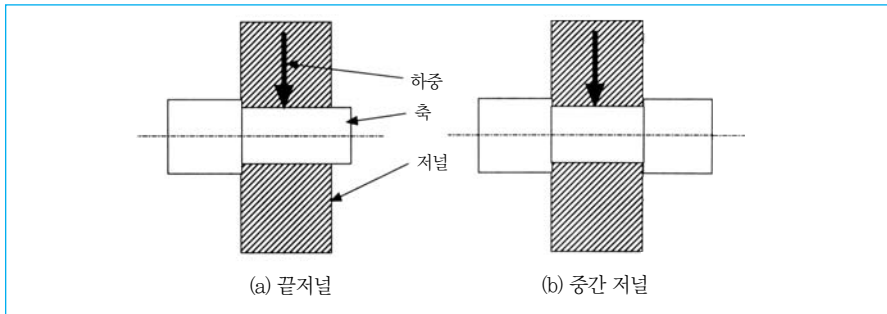
저널은 원통형(cylindrical journal), 원추형(conical journal) 및 구형(spherical journal)으로 구성되어 있다. 베어링과 저널 사이에는 윤활제를 사용하여 마찰을 작게 하여 발열을 억제한다. 저널은 하중의 방향에 따라 레이디얼 저널(radial journal)과 스러스트 저널(thrust journal)으로 분류한다.

1) 레이디얼 저널

베어링의 하중방향이 회전축에 직각인 것을 레이디얼 저널이라고 한다. 〈그



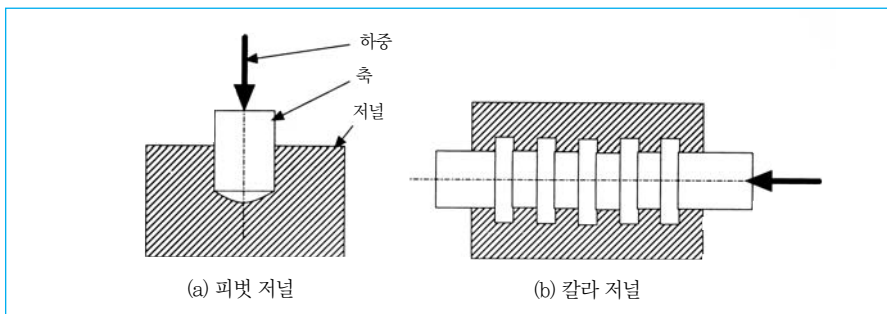
림Ⅲ-24)은 레이디얼 저널의 종류를 나타낸다. 끝저널(end journal)은 축의 한 쪽이 저널에 삽입되어 있는 외팔보 형태를 가진다. 중간 저널(neck journal)은 저널의 양쪽으로 축이 나와 있는 형태를 말한다.



〈그림Ⅲ-25〉 레이디얼 저널의 종류

2) 스러스트 저널

베어링의 하중 방향이 회전축의 중심선과 일치하는 것을 스러스트 저널이라 한다. 〈그림Ⅲ-26〉은 저널의 종류를 나타낸다.



〈그림Ⅲ-26〉 스러스트 저널의 종류

(2) 레이디얼 저널의 설계

1) 저널 설계상의 유의점

강도, 강성, 베어링, 압력, 열전도 등에 대하여 충분히 고려하고, 다음과 같은

것에 유의하여 설계하여야 한다.

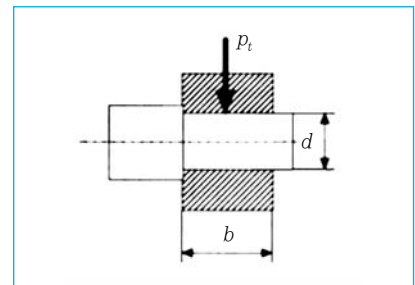
- ① 하중에 대한 충분한 강도를 가질 것.
- ② 과도한 변형률이 생기지 않도록 할 것.
- ③ 베어링 압력이 제한 내에 있을 것.
- ④ 마찰, 마멸이 적을 것.
- ⑤ 윤활유를 잘 유지하고 있을 것.
- ⑥ 마찰열의 발생이 적고 열의 발산이 좋을 것.

2) 베어링 압력

축과 저널이 접촉하는 면적에 작용하는 평균 압력을 베어링 압력이라고 한다. 베어링 압력이 너무 크면 저널과 베어링 사이의 유막이 파괴되어 윤활을 유지하지 못한다. <그림 Ⅲ-27>은 레이디얼 저널의 설계에 필요한 파라미터를 나타낸다. 저널의 지름(베어링의 안지름)을 d , 저널의 폭을 l , 베어링 하중을 W 라고 하면, 베어링 압력 P 는 다음의 관계식으로 구한다.

$$P = \frac{W}{dl} \quad \dots \text{(Ⅲ-10)}$$

베어링 압력의 허용치는 축의 회전속도, 축 및 베어링 메탈의 종류, 윤활제의 종류, 급유방법에 따라 다르므로 제조업체의 사양을 참조하여 적당한 허용치를 선택하여야 한다.



<그림 Ⅲ-27> 끝저널



2) 끝저널의 계산

저널을 외팔보이며 베어링 압력이 균일하게 분포되어 있다고 가정한다.

분포 하중은 저널의 중앙에 집중 하중이 작용하는 것으로 합력을 구할 수 있으므로, 굽힘 모멘트 M 은 다음과 같다.

$$M = \frac{Wl}{2} = Z\sigma_b = \frac{\pi d^3}{32} \sigma_b \quad \cdots \text{(III-11)}$$

여기서, Z 는 단면계수를, σ_b 는 허용 굽힘 응력을 나타낸다. 이를 다시 저널의 폭에 대하여 정리하면,

$$Wl = \frac{\pi d^3 \sigma_b}{16} \quad \cdots \text{(III-12)}$$

$$d^3 = \frac{16Wl}{\pi \cdot \sigma_b} \quad \cdots \text{(III-13)}$$

이 된다.

[예제 III-6] 회전속도가 400[rpm]인 축을 지지하는 저널 베어링에 하중 $W=1600[\text{kgf}]$ 이 작용하고 있다. 허용 베어링의 압력을 $P=0.1[\text{kg/mm}^2]$, 저널의 폭과 지름의 비가 $l/d=2$ 라고 할 경우에 저널 베어링의 폭과 지름을 구하라.

(풀이) 베어링의 압력을 구하는 공식 $P = \frac{W}{ld}$ 와 폭과 지름의 비 $\frac{l}{d}=2$ 를 이용하여 풀다.

$l=2d$ 를 베어링의 압력을 구하는 공식에 대입하면,

$$P = \frac{W}{ld} = \frac{W}{2d^2}$$

이 된다. 위의 식에서 저널의 지름을 구하면

$$d = \sqrt{\frac{W}{2P}} = \sqrt{\frac{1600}{2 \times 0.1}} = 89.5[\text{mm}]$$

가 되고, 저널의 폭은 다음과 같다.

$$l = 2d = 2 \times 89.5 = 179[\text{mm}]$$

[예제 Ⅲ-7] 회전속도가 130[rpm]인 축을 지지하는 끝저널 베어링에 하중 $W=6000[\text{kgf}]$ 이 작용하고 있다. 단, 허용 베어링의 압력을 $P=0.6[\text{kgf}/\text{mm}^2]$, 허용 굽힘응력 $\sigma_b=6[\text{kg}/\text{mm}^2]$ 이다. 끝저널 베어링의 폭과 지름을 구하라.

(풀이) 베어링의 압력을 구하는 공식 $P = \frac{W}{ld}$ 을 이용하여 폭과 지름의 관계를 구한다.

$$ld = \frac{W}{P} = \frac{6000}{0.6} = 10000[\text{mm}^2]$$

따라서 지름은 $d = \frac{10000}{l}$ 가 되므로, 이 값과 예제에서 주어진 값들을 다음 식에 대입하여 저널의 폭을 구한다.

$$l = \frac{\pi d^3 \sigma_b}{16W} = \frac{\pi \times \left(\frac{10000}{l}\right)^3 \times 6}{16 \times 6000} = \frac{1.9 \times 10^8}{l^3}$$

$$l^4 = 1.9 \times 10^8$$

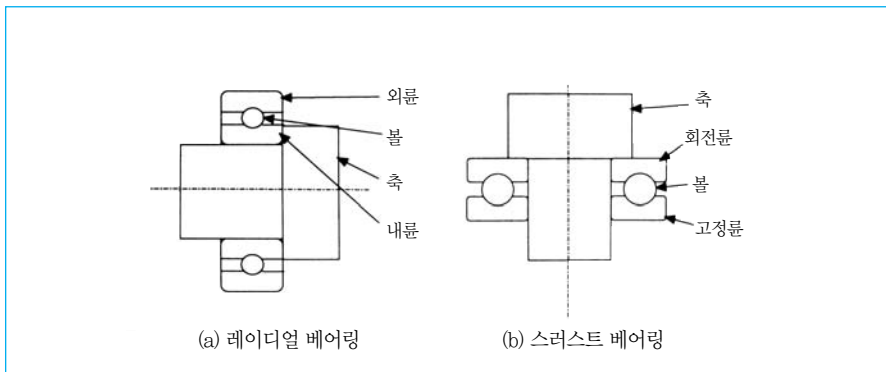


따라서, $l=117.4[mm]$ 가 되고,

$$d = \frac{10000}{117.4} = 85.2[mm] \text{ 가 된다.}$$

3) 구름 베어링

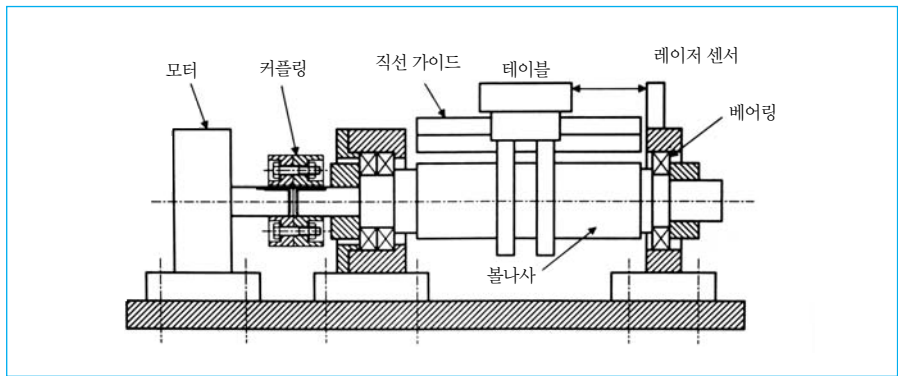
구름 베어링은 <그림 III-28>과 같이 외륜과 내륜의 사이에 볼이 있는 레이디얼 베어링과, 회전륜과 고정륜 사이에 볼이 있는 스러스트 베어링이 있으며, 하중의 작용 방향이 다르다. 2개의 궤도륜 사이에 있는 볼은 서로 접촉이 일어나지 않을 정도의 충분한 거리를 두고 있으며, 구름접촉을 하므로 마찰손실이 작다. 구름 베어링은 볼의 모양에 따라서 볼 베어링(ball bearing)과 롤러 베어링(roller bearing)으로 구분한다.



<그림 III-28> 레이디얼 베어링과 스러스트 베어링

레이디얼 베어링은 구조가 간단하며 정밀도가 높으므로, 고속 회전용으로 많이 사용된다. 스러스트 베어링은 축 방향의 하중을 받는 곳에 사용되며, 고속 회전에는 부적합하다.

〈그림Ⅲ-29〉는 모터의 의한 회전운동을 직선운동으로 변환하는 1축 테이블의 사례를 나타낸다. 모터의 축과 볼나사의 축은 커플링에 의해 연결되고 있으며, 볼나사의 양쪽에는 베어링에 의해 지지되고 있다.



〈그림Ⅲ-29〉 베어링에 의한 축의 지지

(4) 안내 요소

〈그림Ⅲ-30〉는 직선운동을 필요로 하는 테이블의 안내요소인 직선 가이드를 나타내며, 안내는 주로 볼 베어링 혹은 롤러 베어링을 사용한다. 안내면에는 윤활유를 사용하여 마찰을 최소화시키며, 주기적으로 윤활유의 공급이 필요하다.



〈그림Ⅲ-30〉 직선 가이드

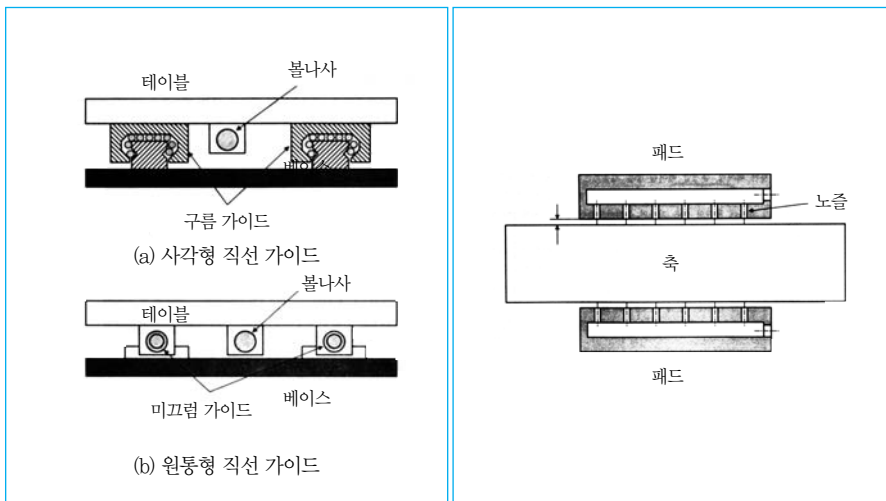
〈그림Ⅲ-31〉은 직선 가이드에 있어서 슬라이드부가 일정한 각을 가진 사각형 형태와 원통형으로 구분된다. 사각형은 볼 베어링에 의해 안내되는 점 접촉



형과 롤러 베어링에 의해 안내되는 선 접촉형이 있다. 원통형은 주로 미끄럼 안내를 하므로 면 접촉을 이루고 있다.

최근에는 안내요소의 마찰을 최소화하여 위치 정밀도를 향상시키기 위하여 공기 베어링, 자기 베어링, 유정압 베어링 등을 사용하는데, 이러한 베어링은 공기, 자력, 압유 등에 의해 안내면이 일정한 간격을 유지하고 있으므로 비접촉이다. 따라서 마찰이 없어서 약간의 힘을 가해도 움직이므로 초정밀 위치제어를 요하는 반도체 장비에 많이 채용되고 있다.

〈그림 Ⅲ-32〉은 공기 베어링의 구조를 나타낸다. 축과 패드는 일정한 간격을 유지하며, 노즐을 통하여 일정한 압력을 가진 공기압을 토출한다. 토출된 공기압이 축과 패드의 비접촉을 유지하므로 일정한 공기압을 계속하여 공급하여야 하는 단점을 가지고 있으나, 초정밀 위치제어에 많이 채용되고 있다.



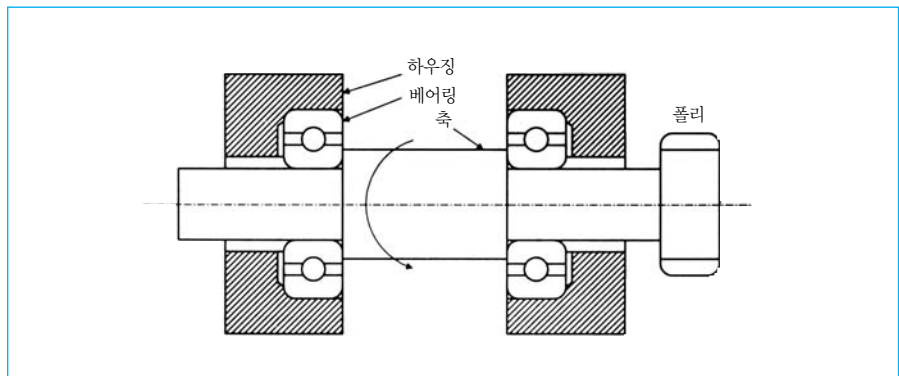
〈그림 Ⅲ-31〉 직선 가이드의 종류

〈그림 Ⅲ-32〉 공기 베어링

4 축과 축이음

(1) 축

회전운동을 전달하는 기계요소인 축은 하중이 작용하는 상태에 따라 굽힘 모멘트를 받는 축, 비틀림, 인장, 압축 등의 하중이 동시에 2개 이상 받는 축으로 분류한다.



〈그림Ⅲ-33〉 레이디얼 베어링에 의한 축의 지지

〈그림Ⅲ-33〉은 2개의 레이디얼 베어링으로 축을 지지하는 구조를 나타낸다. 축의 중간 부분에서는 축의 하중에 의한 처짐이 발생하고, 풀리에 의해 동력이 전달될 경우에는 비틀림 변형이 발생한다. 축을 설계할 때 고려해야 하는 주요 사항은 다음과 같다.

- ① 강도 : 하중의 종류에 따라 충분한 강도(strength)를 갖도록 한다. 특히, 축의 모서리 부분에는 응력이 집중되기 쉬우므로 모서리 부분에는 라운드(round)를 둔다.
- ② 강성 : 하중과 변형의 관계를 강성(stiffness)이라고 하며, 하중에 의한 변



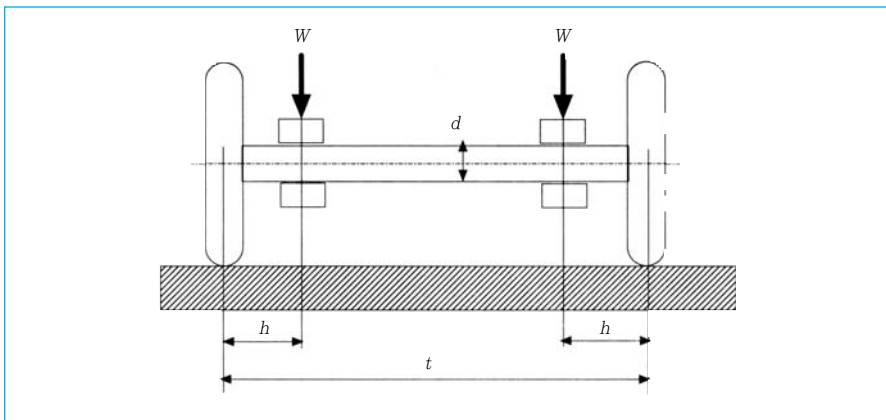
형이 일정한 한계 이하가 되도록 설계하여야 한다. 굽힘하중을 받은 경우에는 축의 처짐이 어느 한도를 넘으면 베어링 압력의 불균일, 기계적 정도(정밀도)의 저하를 발생시킨다.

- ③ 진동 : 축이 회전하면서 발생하는 축의 굽힘 또는 비틀림에 기인한 진동이 축의 고유진동과 공진할 경우에는 축이 파손될 우려가 있으므로, 회전속도와 하중을 충분히 고려하여 축을 설계한다.

1) 굽힘 모멘트를 받는 축

베어링으로 지지되고 있는 축에 있어서 양 끝단에는 <그림 Ⅲ-34>와 같이 차량의 타이어가 부착되어 있다. 베어링에는 하중 W 가 작용하고, 차량의 타이어와 베어링에 작용하는 하중 점과의 거리, 최대 굽힘응력을, 축의 단면계수를 Z 라고 하였을 경우에, 축에, 작용하는 최대 굽힘모멘트 M 은 다음과 같다.

$$M = \sigma_b Z \frac{\sigma_b \pi d^3}{32} [mm \cdot kgf] \quad \dots (III-14)$$



<그림 Ⅲ-34> 굽힘 모멘트가 작용하는 축의 지지

[예제Ⅲ-8] <그림Ⅲ-34>에서 하중 $W=4000[kgf]$ 이 작용하고 있다. 타이어와 하중 점의 거리 $l_1=200[mm]$, 타이어 사이의 거리는 $l=1120[mm]$, 허용 굽힘응력은 $\sigma_b=4.5[kgf/mm^2]$ 이다. 축의 지름 d 는 얼마인가?

(풀이) 최대 굽힘모멘트는 베어링의 사이에서 작용하며 크기는 Wl_1 이 된다.

$$M = Wl_1 = 4000 \times 200 = 8.0 \times 10^5 [mm^2 \cdot kgf]$$

이 된다. 위에서 구한 값을 최대 굽힘모멘트를 구하는 식(Ⅲ-5)에 대입하여 지름을 구한다.

$$d^3 = \frac{32M}{\sigma_b \pi} = \frac{32 \times 8.0 \times 10^5}{4.5\pi} = 5.1 \times 10^8 [mm^3]$$

따라서 지름은 $d = 122[mm]$ 이 된다.

3) 비틀림 모멘트를 받는 축

일반적으로 동력을 전달하는 전동축은 비틀림 모멘트를 받는다. 축에 작용하는 비틀림 모멘트 T 는

$$T = \tau Z_p = \frac{\pi d^3 \tau}{16} [mm \cdot kgf]$$

이다. 여기서, $\tau [kgf \cdot mm^2]$ 는 축의 비틀림 응력, $Z_p [mm^3]$ 는 축의 극 단면계수, $d [mm]$ 는 축의 직경이다.



[예제 Ⅲ-9] 비틀림 모멘트 $T = 400[m/kgf]$, 허용 비틀림 응력은 $\tau = 6[kgf \cdot mm^2]$ 일 경우에 축의 지름 $d[mm]$ 는 얼마인가?

(풀이) 축에 작용하는 비틀림 모멘트를 구하는 공식을 이용하여 축의 지름을 구한다.

$$d^3 = \frac{16T}{\pi\tau} = \frac{16 \times 400000}{\pi \times 6} = 3.4 \times 10^5 [mm^3]$$

따라서 지름은 $d = 69.8[mm]$ 이 된다.

(2) 축이음

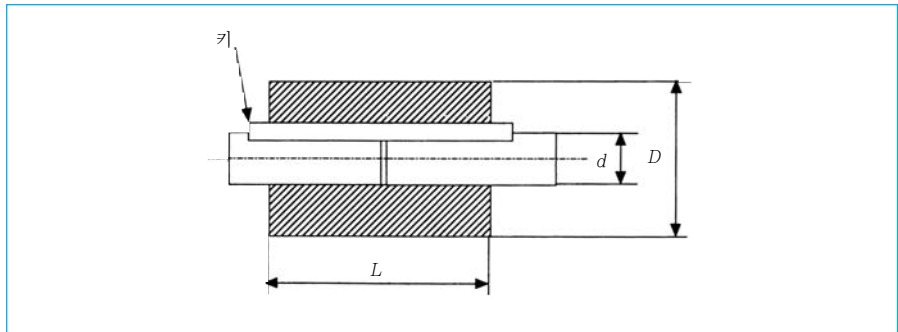
2개의 축을 연결하는 기계요소를 축이음이라고 하고, 회전 할 때 고정되어 회전운동의 전달과 단락의 조정이 불가능한 것은 커플링(coupling), 회전운동의 전달과 단락의 조정이 가능한 구조로 된 것을 클러치(clutch)라 한다.

1) 고정축 커플링

2개의 축 중심선이 일치하고, 축 방향의 이동이 없는 경우에 사용한다. 축의 중심이 어긋나면 회전 시에 굽힘하중이 반복적으로 작용하여 축의 수명을 감소시키며, 베어링에 부가하중으로 작용한다. <그림 Ⅲ-35>는 키(key)에 의한 2개의 축을 고정시키는 원통형 커플링을 나타내며 저속회전, 작은 토크의 전달에 사용하는 가장 간단한 커플링이다.

개략적인 설계치수는 다음과 같다.

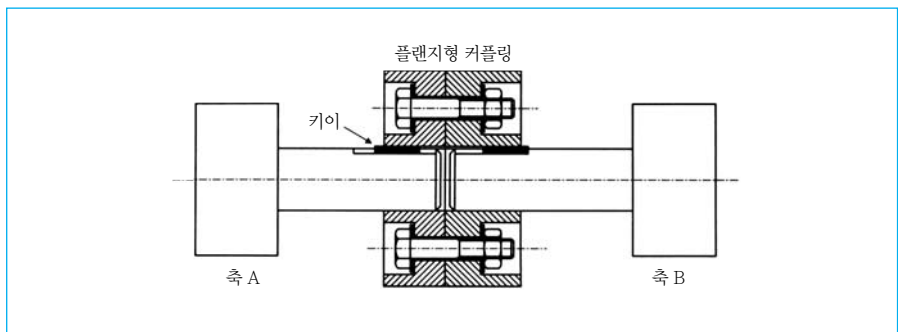
- L = 3d (축이음할 축의 직경이 큰 경우)
- L = 4d (비교적 직경이 작은 축이음의 경우)
- D = 1.8d + (10~20) [mm]



〈그림 Ⅲ-35〉 원통형 축이음

2) 플랜지 커플링

〈그림 Ⅲ-36〉처럼 2개의 축의 끝부분에 플랜지(flange)를 억지로 끼워 맞춘 후 키로 고정한다. 축에 고정되어 있는 플랜지는 볼트를 사용하여 결합함으로써 확실하게 동력을 전달하는 축이음이 된다. 축의 지름이 200[mm] 정도까지의 축이음에 사용되며, 큰 하중이나 고속회전용에 많이 사용한다.



〈그림 Ⅲ-36〉 플랜지 커플링



5 기구 응용

1 기구 응용

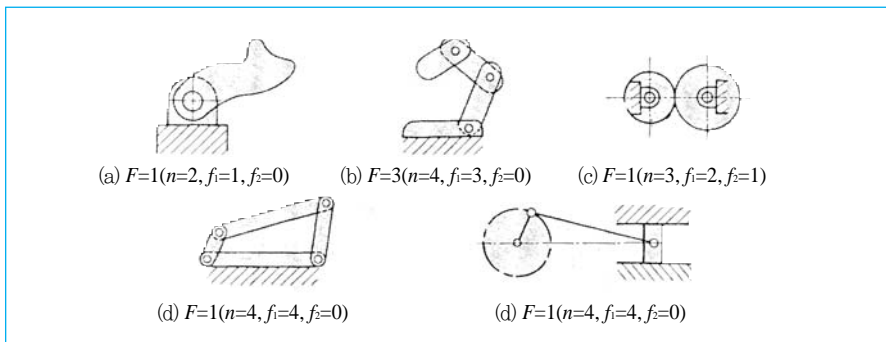
(1) 암(arm) 기구

1) 암의 자유도

● 자유도

물체를 특별한 위치로 움직이기 위하여 필요로 하는 최소한의 독립적인 입력 인자(변수)의 수를 자유도(degree of freedom)라 한다. 일반적으로, 평면 운동을 하는 기구에 대한 자유도는 다음의 쿠츠바흐(Kutzbach) 판별식으로 간단히 구할 수 있다.

그림은 여러 가지 자유도를 나타낸 것이다.



〈그림 III-37〉 여러 가지 기구의 자유도

$$F = 3 \cdot (n-1) - 2 \cdot f_1 - f_2$$

여기서, n : 링크의 수

f_1 : 1자유도의 상대 운동을 허용하는 짝의 수(미끄럼 접촉을 하는 짝
이나 구름 접촉을 하는 짝의 수)

f_2 : 2자유도의 상대 운동을 허용하는 짝의 수(미끄럼 접촉과 구름 접

축이 같이 생기는 짝의 수)

$F > 0$ 인 경우, 기구는 F 개의 자유도를 가진다. 만일 $F = 0$ 이면 운동이 불가능한 기구로서 구조물(structure)이 되며, $F = 1$ 이면 하나의 입력 운동에 의하여 구동할 수 있는 기구가 된다. $F \geq 1$ 이면 기구의 운동을 구속하기 위하여 2개 이상의 입력 운동이 필요하며, 그만큼 운동의 제어가 복잡해진다.

로봇의 경우처럼 3차원 공간상에서 필요로 하는 자세를 취하기 위해서는 6자유도가 필요하며, 이것들은 일반적으로 1자유도를 가지는 6개의 기구를 연결하여 구성된다. 기계를 구성하는 기구의 운동은 대부분 1개의 입력에 대하여 1개의 출력으로 전달되도록 한다. 따라서 1자유도를 가지는 기구가 가장 많이 사용되고 있다.

2) 관절배열에 의한 암의 분류

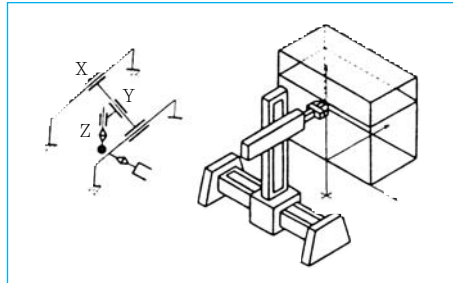
① 직교좌표형

a. 직교좌표 로봇

일명 XY 로봇이라고도 불리는 직교좌표 로봇(Cartesian coordinate robot)은 <그림 III-38>에서와 같이 직선 운동을 수행하는 축으로만 구성되어 있는 로봇이다.

일반적으로 X, Y, Z의 3축으로 구성되며, 경우에 따라서는 1개 또는 2개의 축만을 사용하기도 한다. 또, 잡은 물체의 방향을 바꿀 수 있도록 하기 위하여, 로봇의 선단에 회전축을 붙여 4축으로 구성한 것도 있다.

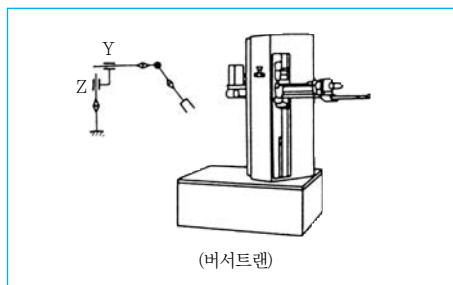
직교좌표 로봇은 기계적으로 튼튼하고 안정적이며 위치 정밀도가 우수하다. 또한, 인간에게 가장 익숙한 직교좌표계를 사용함으로써 이용자가 쉽게 사용할 수 있다는 장점을 가지고 있다. 그러나 동작 영역에 비하여 설치 면적이 큰 것이 단점이다.



〈그림 III-38 직교좌표 로봇〉

② 원통좌표형

원통좌표 로봇(cylindrical coordinate robot)은, 〈그림 III-39〉에서와 같이 원통의 길이와 반지름 방향으로 움직이는 2개의 직선축과 원주 방향으로 움직이는 하나의 회전축으로 구성된 로봇이다. 원통좌표 로봇은 작업 영역이 넓고, 설치 면적도 작으며, 위치 정밀도도 우수하다. 그러나 로봇의 엔드 이펙터(end effector)에 과중한 하중이 걸리는 작업이나 일감의 중량이 크면, 위치 정밀도가 떨어지는 단점을 가지고 있다.



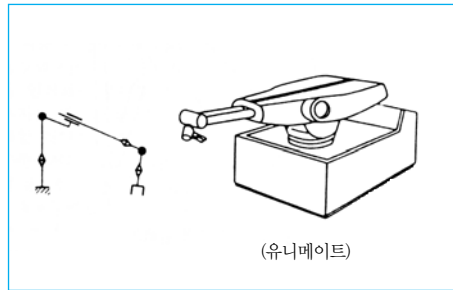
〈그림 III-39〉 원통좌표 로봇

③ 극좌표형

극좌표 로봇(polar coordinate robot)은 1개의 직선축과 2개의 회전축을, 〈그림 III-40〉에서와 같이 조합하여 구성한 로봇이다.

극좌표 로봇은 수직면에 대하여 상하 운동 특성이 우수하여 작업 영역이 넓

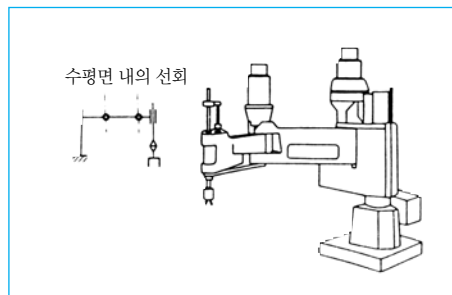
고 경사진 위치에서 작업을 수행할 수 있으므로, 용접 작업이나 도장 작업에 적합하다.



<그림 III-40> 극좌표 로봇

④ 다관절형

스카라 로봇(selective compliance assembly robot arm, SCARA)이라고도 불리는 수평 다관절 로봇(horizontal articulated robot)은, <그림 III-41>에서와 같이 3개의 회전축과 하나의 직선축으로 구성된 다관절 구조의 로봇이다. 1, 2축은 수평 회전 운동을 하고, 3축은 수직 상하 운동을 하며, 4축은 다시 회전 운동을 수행한다. 스카라 로봇은 수평면상의 운동 특성을 가지고 있고, 각 관절의 운동이 중력의 영향을 받지 않으므로 암을 간결하게 설계할 수 있는 강성 구조를 가지고 있기 때문에 조립 작업 또는 팔레타이징(palletizing) 작업에 적합하다.



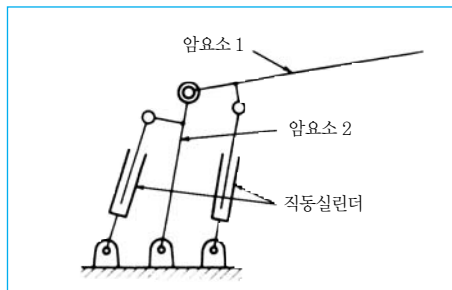
<그림 III-41> 다관절 로봇

3) 관절의 구동형태

① 직접구동형 관절

a. 회전·선회 관절

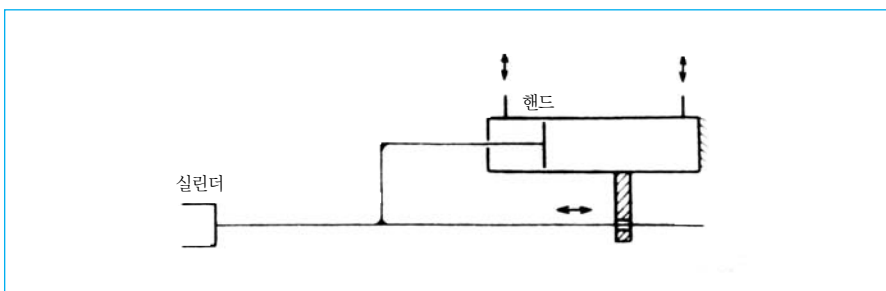
전동식 직접 구동 토크 모터, 유압식 직동실린더 요동모터가 액추에이터로서 이용된다



〈그림 Ⅲ-42〉 직동실린더에 의한 직접구동 선회 관절

b. 슬라이드 관절

유압식에서는 직동 실린더, 전동식에서는 리니어 토크 모터, 리니어 펄 모터 등이 이용된다.



〈그림 Ⅲ-43〉 직동실린더에 의한 직접구동 슬라이드 관절

② 간접구동형 관절

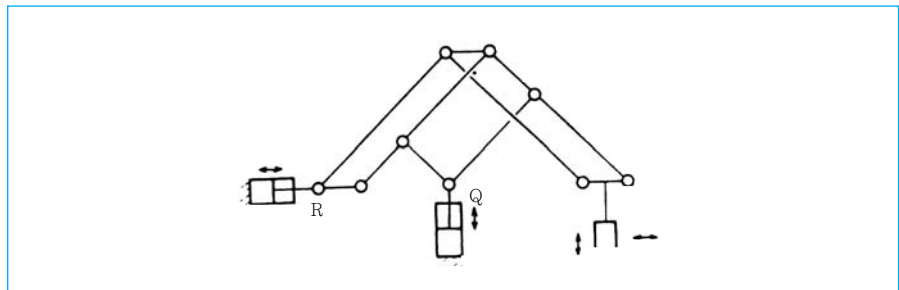
a. 감속기식

전동모터는 정격회전수가 높으며, 저속에서의 발생토크가 낮다. 그 때문에 기어를 이용하여 감속을 해주면 출력 토크가 증폭된다. 하모닉드라이브, 평기어 감속기를 주로 사용한다.

하모닉드라이브는 감속비가 높고 콤팩트하여 스칼라 로봇에서 많이 사용된다.

b. 링크식

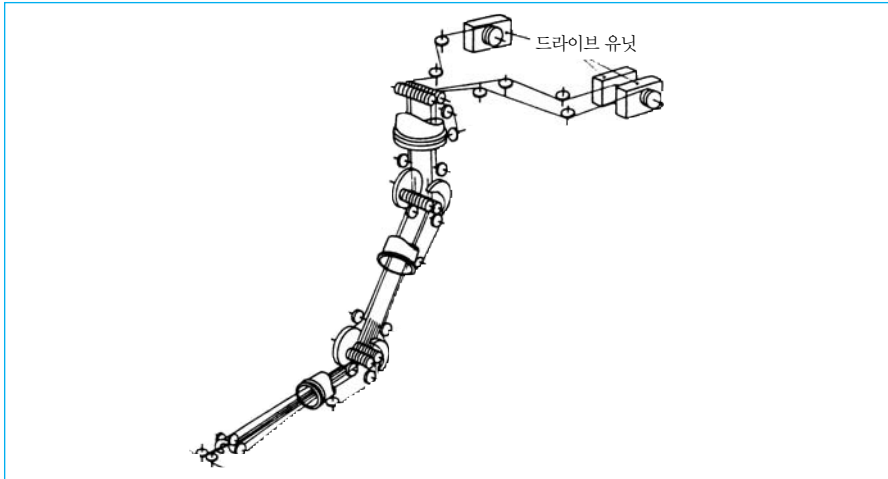
다관절형 암과 같이 직렬로 회전, 선회 관절이 접속되는 타입의 암에서는 핸드 부분에 가까운 관절의 액추에이터의 중량이 기동부에 가까운 관절의 액추에이터의 부하가 된다. 그 결과 기동 부분의 고출력 액추에이터를 필요로 하고, 암구조의 대형화를 초래한다. 이 문제를 해결하는 방법으로서, 주로 평행사변형 링크와 팬터그래프 기구를 많이 사용한다.



〈그림Ⅲ-44〉 팬터그래프 구조를 이용한 암

c. 케이블 풀리식

이 방식도 액추에이터를 기동부에 배치할 목적으로 도입되었다. 풀리로 케이블을 가이드해서, 기동부에 배치된 액추에이터(주로 전동식)의 출력을 각 변위의 관절에 전달한다.



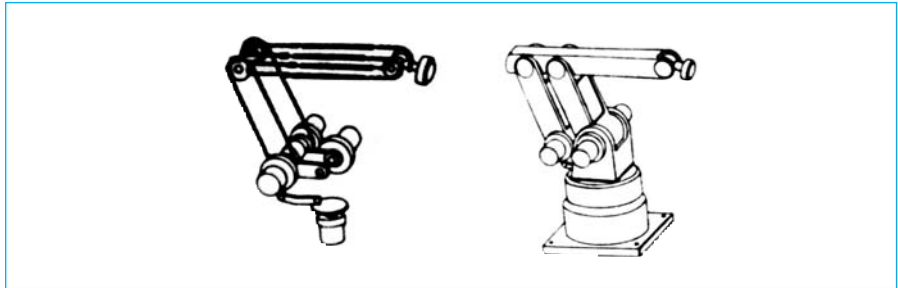
〈그림 III-45〉 케이블 풀리식

③ 간접구동형 관절-슬라이드 관절

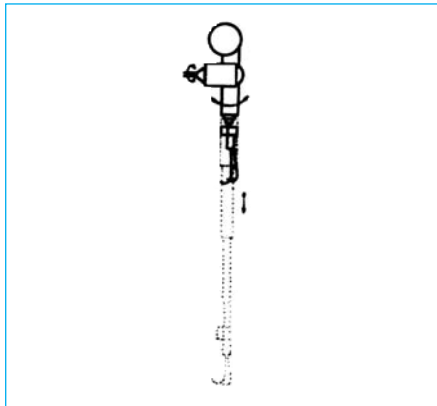
회전형 액추에이터로 직동운동을 얻는 타입이다. 가장 일반적인 것은 회전형 전동 모터의 출력을 볼나사로 직선운동으로 바꾸는 타입이다. 예로 〈그림 III-47〉의 텔레스코픽 암(telescopic arm)이 이용된다.

4) 손목 관절 기구(wrist joint mechanism)

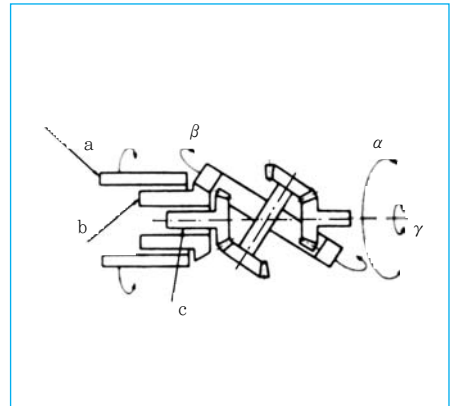
〈그림 III-48〉처럼 핸드에 가까운 3 관절(손목관절)에서 간결하게 설계하기 위해서는 특수한 기구가 채용되는 수가 있다. 〈그림 III-49〉는 어프세트형 손목 기구로 a로 전체를 회전시키면 γ , 축 b를 회전하면 베벨기어를 매개로 β , 축 c를 a회전하면 마찬가지로 베벨기어를 매개로 α 의 회전을 얻을 수 있다.



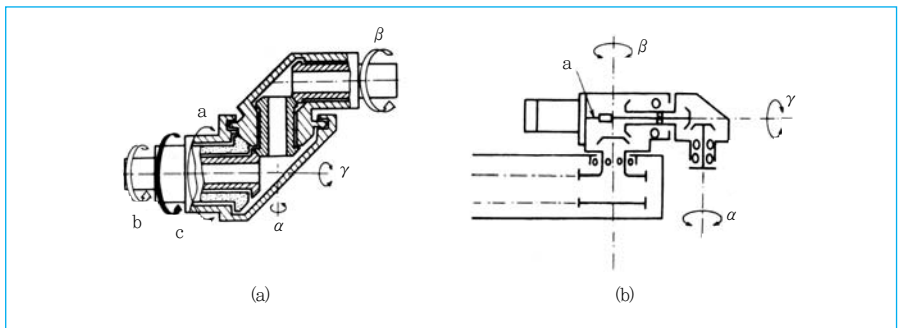
〈그림 III-46〉 체인 · 스포라딕에 의한 손목 관절의 구동



〈그림 III-47〉 텔레스코픽 암



〈그림 III-48〉 자유도 리스트 관절



〈그림 III-49〉 어프셋트형 3자유도 리스트 관절



5) 다자유도 암 기구(arm mechanism with-multidegree of freedom)

좁은 공간을 장애물을 피하면서 목표점에 도달해야만 할 때는 암 자신의 형상이 자유곡선을 그리는 것 같은 특성을 가지면 효과적이다. 기능을 실현하기 위해서는 다수의 자유도를 포함한 암 기구보다는 매끄럽게 휘는 관절 기구가 필요하다.

(2) 핸드 기구

1) 핸드의 분류

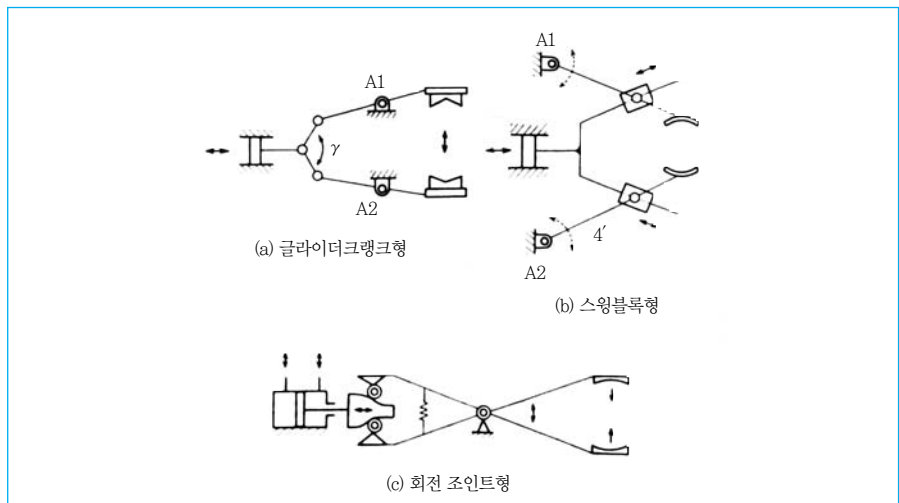
핸드는 암의 선단에 부착, 대상에 직접 작용을 미치는 장치이다. 엔드 이펙터(end effector)라고도 한다. 이 장치는 사람으로 말하면 손에 해당하나, 사람의 손의 모든 기능을 갖춘 핸드를 설계하는 것은 곤란하고 실용적 관점에서 필요한 기능을 추출하여 설계한다. 기능이 한정된 핸드는 특수핸드라고 불리며 전용 공구를 이용하는 타입이다. 예를 들면 용접 로봇의 용접건, 도장 로봇의 스프레이건을 이용한다. 평평한 금속면이나 유리판을 핸들링하는 자기흡착 장치나 진공흡착 장치, 조립용 로봇은 특수한 컴플라이언스 기구가 사용된다. 산업용으로 많이 이용되고 있는 특수공구 이외의 핸드는 메커니컬 핸드(machnical hand)라 불리는 것이다. 관절이 없는 2개 혹은 3개의 손가락으로 가위 기능을 가지며, 특정 형상의 것을 잡는 데 사용된다. 유니버설 핸드(universal hand)는 3개 혹은 5개의 손가락을 가지며 관절을 가진 손가락을 유연하게 제어함으로써 다양한 형상의 물체를 잡거나, 대상을 조종하는 기능을 실현할 수 있는 가능성이 있다.

2) 메커니컬 핸드

파지하는 형태로서는 대상을 바깥에서 파지하는 것과 안에서 파지하는 것이 있다. 기구적으로는 동일하므로 바깥에서 파지하는 것에 대하여 설명한다.

① 두손가락 형

두 개의 리지트한 손가락을 갖춘 것으로 실용례가 많다. 손가락이 상호 평행 상태를 유지해서 개폐되는 병진형 핸드(translational gripper)와 그 방향이 변화하는 스윙형 핸드(swinging gripper)로 분류된다. 스윙형은 기구가 간단하나, 대상은 안정되게 파지하는 점에서 병진형보다 떨어진다.

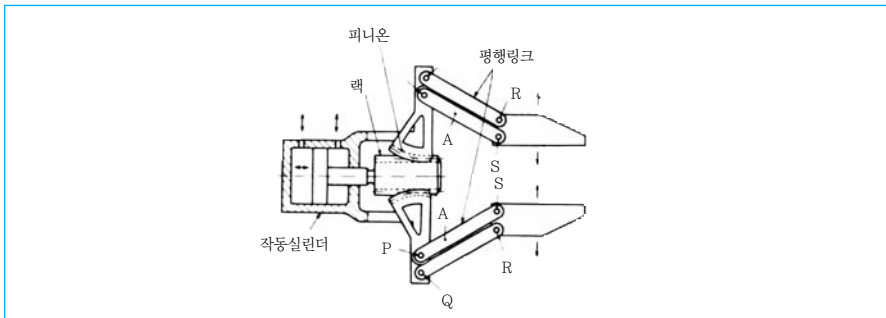


<그림 Ⅲ-50> 스윙형 핸드의 링크 기구

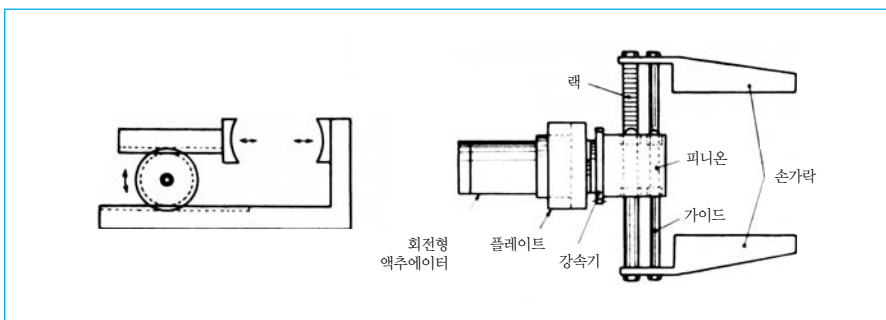
병진형에서는 평행링크, 랙·피니온, 나사이송 기구 등이 이용되고 있다. <그림 Ⅲ-51>은 평행링크 방식의 예로, 직동형 액추에이터의 운동을 랙과 피니온으로 회전운동으로 바꾸고, 그 피니온 운동으로 평행링크의 하나의 링크 A를 축 P의 주위에 회전시킴으로써 손가락의 개폐를 얻는다. 링크 PS와 링크 QR은 항상 평행으로 유지되기 때문에, 손가락의 물체 파지면의 자세도 개폐 동작 중 일정하게 유지된다. 직동형 대신에 회전형 액추에이터를 이용한 형식에서 평행링크를 회전시키도록 한 것도 있다. 평행링크를 이용한 형식은 개폐에 의해서 손가락 위치가 전·후진하는 결점이 있다. 이 문제를 해결하는 방법으로서 볼나사 기구나 랙·피니온 방식이 채용된다. <그림 Ⅲ-52>는 회전형 액추에이터로



피니온을 회전시킴으로써 개폐동작을 하는 핸드의 예이다. 볼나사 기구를 이용하는 경우에는 1대의 왼나사, 오른나사의 볼나사 기구 각각을 손가락에 부착, 나사를 회전시킴으로써 개폐동작을 실현한다.



〈그림 Ⅲ-51〉 병진형 핸드(평행링크형)

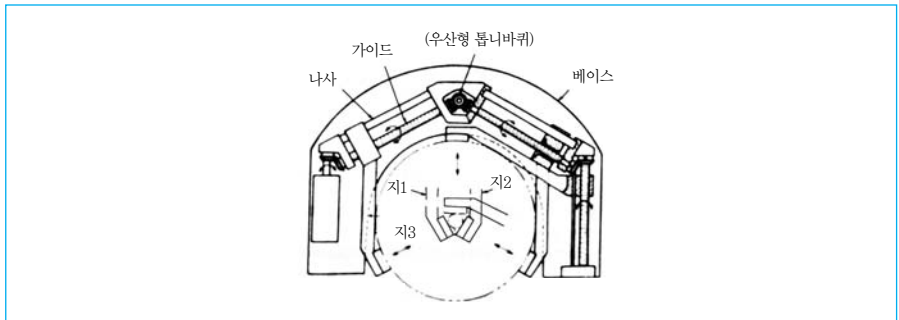


〈그림 Ⅲ-52〉 병진형 핸드의 예(랙, 피니온 형)

② 세손가락 형

세 손가락을 가진 핸드에서는 대상을 3점에서 유지할 수 있기 때문에 두 손가락으로 잡은 것보다 안정된 파지 동작을 할 수 있다. 일례를 〈그림 Ⅲ-53〉에 나타낸다. 이 핸드에는 베벨기어를 매개로 상호 각도를 주어서 결합한 3개의 볼나사를 하나의 모터로 구동해서 세 손가락을 동작시킴으로써, 이른바 3포인트 핀치(three point pinch)를 실현하는 것이다. 특히 원통 물체를 확실하게 잡아서

유지하는 데 적합하다. 이 외에 각 손가락을 개별 모터로 구동하도록 한 3지형 핸드도 개발되고 있으나, 기구가 복잡하며 중량도 증가하기 때문에 실용례는 적다.



〈그림 Ⅲ-53〉 세 손가락형 핸드의 예

③ 손가락 모양

단단한 손가락을 가진 핸드의 경우, 대상의 형상에 따라서는 안정되게 잡기가 곤란한 것도 있다. 이상적으로 물체를 잡으려면 대상에 감싸는 손가락 구조가 바람직하다. 단단한 손가락으로 근사적으로 실현하는 방법으로서 V자 홈을 손가락 면에 설치하는 것이 있다. 또한 잡히는 대상이 특정한 모양인 경우는, 대상의 외형에 대응하는 홈을 설치하는 것도 있다. 이러한 단단한 손가락에 대해서, 특히 대상의 외형에 대한 친밀함을 고려해서 설계된 손가락 기구도 연구되고 있다.

3) 유니버설 핸드

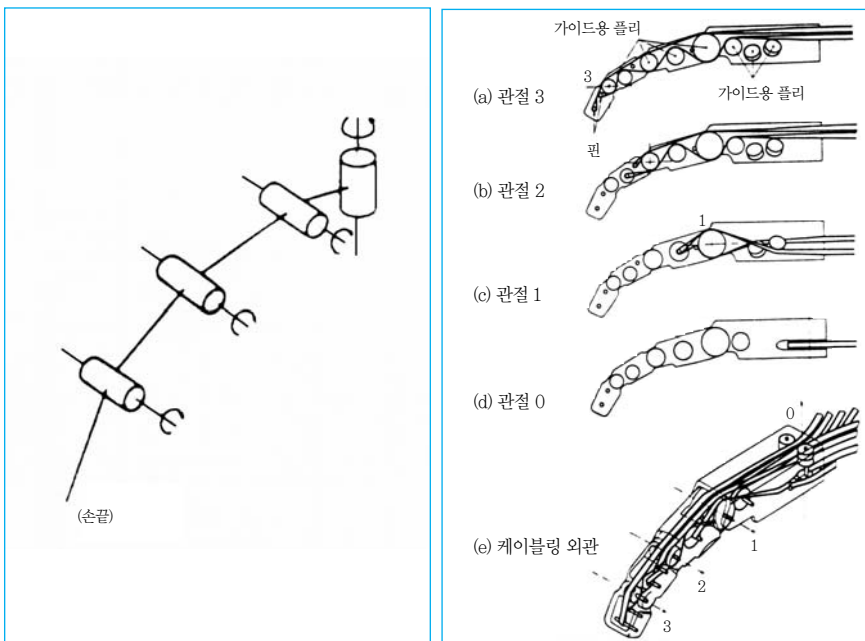
사람의 손에는 다섯 손가락이 있으나 각 손가락이 수행하는 기능에 주목하면 소지, 약지를 제외한 세 손가락으로 충분한 경우가 많다. 현재 연구되고 있는 유니버설 핸드(universal hand)에는 다섯 손가락형도 있으나 일반적으로는 세 손가락형이 많다. 어느 형태라도 손가락에 관절을 가진다. 각 손가락의 관절은 3내지 4이다. 〈그림 Ⅲ-54〉에 각 손가락에 자유도 구성을 나타냈다. 각 손가락



을 3자유도로 할 때는 일반적으로 손가락 선단에 가까운 관절이 생략된다.

이러한 손가락 관절의 구동에서는 각 관절 위치에 액추에이터를 배치하는 공간이 없기 때문에 케이블 구동방식이 채용된다.

〈그림 Ⅲ-55〉은 평 케이블을 이용하는 4자유도 손가락의 예를 나타낸다. 각 그림은 손가락 끝에 가까운 관절부터 순서대로 각각 네 개의 관절의 구동 기구에 대응하고 있으며, 어느 것이든 2줄의 케이블로 구동된다. 케이블의 선단은 핀으로 고정되어 있으며, 핀을 매개로 2중의 케이블로 관절을 잡아당김으로써 각 관절 주위에 토크가 발생된다. 핸드의 케이블링에서는 암의 경우에 비해서 사용가능한 공간이 좁기 때문에, 각 자유도에 간섭이 발생하지 않도록 하기 위한 방안이 필요하다. 〈그림 Ⅲ-55〉의 경우는, 그림(e)에서 보는 바와 같이 상호 겹치지 않도록 케이블을 배치함과 동시에 관절부에 배치된 가이드용 폴리의 바깥 기둥에 감아 붙이는 것으로, 간섭을 방지한다.

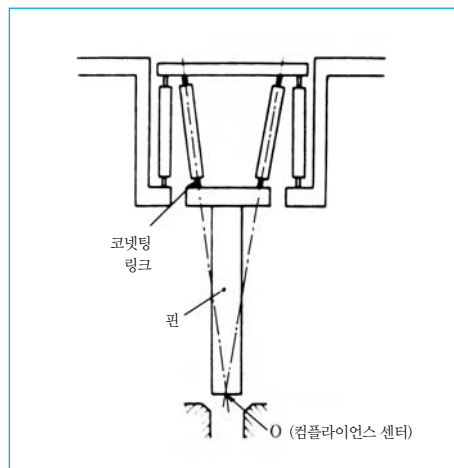


〈그림 Ⅲ-54〉 손가락의 대표적인 자유도 구성 〈그림 Ⅲ-55〉 유니버설 핸드의 손가락 기구의 일례

4) 컴플라이언스 기구(compliance mechanism)

핸드로 잡은 핀을 동그란 구멍에 삽입하는 작업을 효율적으로 하는 장치로, 리모트 센터 컴플라이언스(remote center compliance, RCC) 기구가 있다. <그림Ⅲ-56>에 기본구조를 나타내었다. 이 기구는 평행 링크와 대칭 형상 링크의 조합으로 이루어진다. 파지하고 있는 핀에 옆 방향에서 수직으로 힘이 작용하면 평행링크가 그것을 흡수하고, 회전력이 가해지면 대칭형 링크의 작용으로 점 O를 중심으로 회전해서 그 회전력을 흡수한다. 점 O를 컴플라이언스 센터(compliance center)라고 부른다.

이러한 기구를 이용해서 눌러 넣음과 동시에 핀은 옆 방향으로 힘을 받기 때문에 구멍의 중심 방향으로 이동한다. 또한 핀이 비스듬하게 동그란 구멍에 맞을 때는 구멍의 내벽으로부터의 반력에 의해 점 O를 중심으로 핀이 회전하고, 구멍의 중심선과 핀의 중심선을 일치시키는 운동이 발생한다. 따라서, 틈새가 적을 경우에도 간단한 제어로 매끄러운 끼우기 작업을 고속으로 로봇에 시킬 수 있다.



<그림Ⅲ-56> 리모트 센터 컴플라이언스(RCC) 기구



단원 학습 정리



- 1** 로봇의 구성요소는 시스템의 전체 골격을 유지하는 요소로서 구조물, 운동 발생 장치, 상태의 측정 장치로 구분할 수 있다.
- 2** 부품과 부품을 고정하기 위해서는 여러 가지 방법이 있으며 재료, 용도, 강도, 진동, 열팽창에 대한 영향 등을 충분히 고려하여 목적 달성에 가장 적합한 체결 방법을 사용하여야 한다.
- 3** 나사는 나선상의 홈을 이용하여 두 가지의 부품을 체결하는 것과 동력을 전달하는 것으로 구분할 수 있다. 체결용 나사는 산의 모양이 삼각형이고, 동력 전달용 나사의 경우는 산의 모양이 사각형 혹은 사다리꼴의 형태를 가진다.
- 4** 분해할 필요가 있는 부품을 체결할 경우에는 볼트를 이용하여 체결하거나 볼트와 너트를 사용하여 체결한다. 분해할 필요가 없는 경우에는 리벳을 사용하거나 용접 등과 같은 방법으로 두 부품을 결합한다.
- 5** 동력은 일반적으로는 회전운동의 전달을 나타내며, 회전운동을 전달하는 가장 기본적인 기계요소는 축이다. 모터 혹은 엔진 등에 의해 발생된 회전운동은 축에 의해 다른 동작 부분으로 전달되는데, 동력의 전달 방법으로서는 기어 및 벨트 등을 사용한다.
- 6** 동력을 확실하게 전달하는 방법으로서 기어를 많이 사용하는데, 시계와 같은 작은 기계에서 선박, 항공기 등과 같은 대형 기계에 이르기까지 사용하고 있다.

- 7 원동축과 종동축이 인접할 경우에는 기어를 사용하여 동력을 전달하는 것이 효율적이나, 축간거리가 상당한 거리를 유지하고 있을 경우에는 벨트를 사용하거나 체인을 사용하여 동력을 전달한다.
- 8 2개의 축을 연결하는 기계 요소를 축이음이라고 하고, 회전 할 때 고정되어 회전운동의 전달과 단락의 조정이 불가능한 것은 커플링(coupling), 회전운동의 전달과 단락의 조정이 가능한 구조로 된 것을 클러치(clutch)라 한다.
- 9 로봇에서 관절 배열에 의한 암의 분류는 직교좌표형, 원통좌표형, 극좌표형, 다관절형으로 분류된다.
- 10 구동 형태에 따른 관절의 종류로는 직접구동형관절, 간접구동형관절, 간접구동과 슬라이드형을 합친 관절이 있다.
- 11 핸드는 암의 끝에 부착되어 대상에 직접 작용을 미치는 장치로, 사람의 손의 기능을 한다.





단원 종합 문제

- ▶ **1** 나사의 피치가 $1.5[mm]$ 이고 2줄 나사를 가진 삼각나사의 리드는 얼마인가?
- ▶ **2** 직선운동 기구에 있어서 스테핑 모터의 분해능은 $1.8[^\circ / \text{펄스}]$, 피치가 $4[mm]$, 나사산의 줄 수가 1인 볼나사를 사용하며 안내면의 마찰력은 무시한다. 테이블의 최소이동거리(위치 정밀도)를 구하고, 스테핑 모터에 $400[\text{펄스}]$ 를 입력할 경우에 이동대가 이동하는 거리를 구하라.
- ▶ **3** 기어 전동의 특징을 설명한 것이다. 잘못 설명한 것은?
 - ① 속도비가 일정하다.
 - ② 큰 동력을 전달할 수 있다.
 - ③ 감속비가 크다.
 - ④ 충격에 강하다.
- ▶ **4** 한 쌍의 기어가 맞물려서 회전할 때 잇수가 작은 기어를 무엇이라 하는가?
 - ① 피니언 ② 큰 기어 ③ 전위 기어 ④ 스큐 기어
- ▶ **5** 원동축의 기어 잇수는 40, 기어의 외경이 $128[mm]$, 종동축의 기어 잇수가 20인 경우에 이의 크기와 종동축의 외경, 축간거리를 구하여라.
- ▶ **6** 회전속도가 $300[\text{rpm}]$ 인 축을 지지하는 저널 베어링에 하중이 작용하고 있다. 허용 베어링의 압력을 $P=0.2[\text{kg}/\text{mm}^2]$, 저널의 폭과 지름의 비가 $l/d=2$ 라고 할 경우에 저널 베어링의 폭과 지름을 구하라.



로봇설계의 실제

이 단원에서는 원하는 로봇을 설계하기 위하여 로봇을 실제로 어떻게 설계할 것인가에 대하여 알아보기로 한다.



학습목표

1. 로봇의 기능과 특성을 설명할 수 있다.
2. 로봇 제작 전에 로봇설계에 관한 계획을 세울 수 있다.
3. 어떤 사용목적에 적합한 로봇을 설계하는 데 필요한 마이크로프로세서와 기계·기구부, 센서부, 모터 제어부와의 인터페이스를 설계할 수 있는 능력을 기른다.
4. 설계 및 제작 과정 또는 제작 후에 자료의 문서화와 유지관리에 관하여 이해한다.
5. 로봇을 제어하는 데 필요한 프로그래밍 방법을 이해하고, 실습을 통하여 그 응용 능력을 기른다.



1 로봇의 기능 및 특성

로봇은 그 기능과 용도에 따라 크게 산업용 로봇, 서비스용 로봇, 기타 특수 목적을 수행하는 기능을 갖는 로봇 등으로 나눌 수 있다.

1 산업용 로봇의 기능 및 특성

산업용 로봇은 수치제어 로봇, 지능 로봇 등 여러 가지 종류의 산업용 로봇이 있으나, 사용 용도에 따라 각기 그 기능을 달리한다. 수치제어 로봇은 입력되는 수치제어 정보에 따라 기능을 수행하는 특성을 갖고 있다. 따라서 이러한 기능을 잘 수행할 수 있도록 설계해야 하며, 지능 로봇의 경우는 감각 및 인식 기능에 의하여 요구되는 작업을 스스로 결정할 수 있는 특성을 갖고 있으므로 사용 용도에 따라 설계 계획을 달리하여야 할 것이다.

이 밖에 수동 로봇은 사람의 조작에 의하여 그 기능을 수행하는 특성을 갖고 있으므로 이러한 기능을 잘 수행할 수 있도록 설계하여야 한다. 시퀀스 로봇은 미리 설정된 순서와 조건에 따라 동작하는 특성을 갖고 있으므로, 이러한 특성을 잘 수행할 수 있도록 설계한다. 플레이백 로봇은 사람이 로봇에 정보를 입력하고 기억시켜 두었다가 필요한 작업을 반복할 수 있는 기능을 갖고 있다.

한편 감각 정보를 사용하여 스스로 동작할 수 있는 기능을 갖고 있는 감각제어 로봇, 제어 특성 및 조건을 작업 환경에 따라 스스로 변화하여 그 기능을 수



행할 수 있는 적응제어 로봇 등도 그 기능과 특성에 따라 설계하여야 한다. 학습 제어 로봇의 경우는 이전에 작업했던 작업 경험을 바탕으로 다음 작업에서 요구되는 특성 및 기능을 수행할 수 있도록 설계되어야 할 것이다.

2 서비스용 로봇의 기능 및 특성

서비스용 로봇은 사람과 동일한 공간에서 사용될 수 있는 인간 공존형 로봇이다. 산업용 로봇이 특정 분야에서 사용 목적에 따라 작업하는 기능을 갖는 것과는 달리 다양한 종류의 서비스를 수행할 수 있는 로봇으로 그 분야가 다양하다. 예를 들면 우편안내용 로봇, 청소용 로봇 등이 그 예이다. 서비스용 로봇은 개인 서비스용의 로봇이 대부분 사용되고 있지만 국방, 의료, 우주, 보안 및 의료용의 전문 서비스용 로봇도 있다.

3 기타 로봇의 기능 및 특성

위에 설명한 산업용 또는 서비스용 로봇 외에도 인간의 지적 수준을 가지고 인간을 대신할 수 있는 휴머노이드 로봇과 인간의 문화생활을 영위하는 데 도움을 줄 수 있는 엔터테인먼트용 로봇, 축구 경기의 기능을 갖춘 축구 로봇 등 각각 사용 목적에 따라 기능과 특성이 다르도록 설계되어야 한다. 또한 군사용 및 지뢰제거용 로봇 등도 그 특수 목적에 따라 적절히 그 기능을 수행할 수 있도록 설계하여야 할 것이다. 특히 의료용 첨단 초소형 로봇의 경우는 인간의 생명과 질병을 다루는 특수 목적에 적합한 기능을 갖도록 설계, 개발되고 있다.

이상과 같이 로봇설계를 위하여 산업용, 서비스용 및 기타 휴머노이드 로봇 등을 살펴보았다. 로봇의 설계 및 제작을 위해서는 콘텐츠 디자인 기술, 네트워크 기술, 시스템 종합 기술, 알고리즘 기술, 액추에이터 기술, 센서기술 및 메커

엔터테인먼트

(Entertainment)

연예·오락·문화산업 등을 총칭하여 일컫는다.

휴머노이드 로봇
걸모습과 행동이 인간과
비슷한 로봇을 말한다.

니즘 기술 등 7개 분야의 기술이 필요하다. 이러한 첨단 핵심 기술들은 정보통신과 컴퓨터와 관련된 기술인 IT(정보통신 기술), 생물학 정보와 관련된 기술인 BT(바이오 기술), 초정밀 기술과 관련된 NT(나노 기술), 문화 영상 기술과 관련된 CT(문화 기술) 및 환경 및 에너지와 관련된 ET(환경 기술) 등과 융합된 종합 과학기술이 필요하다고 볼 수 있다.

2 설계 특성

로봇의 기능은 다양하다. 산업용, 서비스용 또는 특수한 목적에 따라 설계 계획을 달리해야 한다. 또한 기계 기구 시스템의 특성과 센서의 특성 등에 의하여 적절한 설계 계획을 세워야 한다. 기계기술, 전자기술 및 정보기술이 종합된 로봇은 요구 명령을 정확하게 해석하고, 주위 상황을 파악하여 기계 또는 기구를 움직이거나 센서를 작동시켜 주어진 일 또는 명령을 수행하는 전자기계이다. 설계할 때는 다음과 같은 특성을 고려하여야 한다.

1 기능 및 동작 설계 특성

- ① 기계적 또는 물리적 구성품이 프로그램에 의한 기능의 유연성을 갖는다.
- ② 비접촉화와 가동부의 전자화로 고신뢰성을 갖도록 한다.
- ③ 메카트로닉스의 소형 경량화 등이 이루어지도록 한다.
- ④ 고정밀화와 고속화가 가능하도록 한다.

2 기계 · 센서부 설계 특성

하나의 제품을 제작 또는 생산하기 위하여는 기획 → 사양 결정 → 설계 → 생



산 → 검사의 과정을 거쳐 제작 설계하여야 하며, 기계 메커니즘의 설계 순서는 대개 기구 방식의 선정 → 기본 조건의 설정 → 운전 조건의 설정 → 부하 조건의 설정 → 부하 계산 → 사용기구의 크기 선정의 순서로 이루어진다.

기계 메커니즘의 경우는 기계·기구 메커니즘의 구성 요소인 액추에이터, 액추에이터용 전원 및 부속기기, 회전축의 지지 요소, 회전운동을 직선운동으로 전환하기 위한 직선운동계의 구성 요소, 동력의 입력축과 출력축을 연결하기 위한 전동계의 구성 요소, 기계시스템의 동작상태를 파악하기 위한 검출기, 각종 계측기 및 기계시스템을 자동으로 제어하기 위한 마이크로컴퓨터 등의 제어 기기가 다음과 같이 적절히 설계 특성을 갖도록 하여야 한다.

센서의 경우는 메카트로닉스의 작동에 필요한 정보를 검출 및 변환하는 기능을 갖도록 설계하여야 한다. 광센서, 초음파 센서, 변위 센서, 압력 센서, 온도 센서, 습도 센서, 가스 센서, 미각 센서, 마이크로 스위치, 시각 센서 등은 각기 그 특성에 따라 정보를 입력받아 신호를 적절히 증폭하고, 제어가 필요한 적절한 신호로 변환 가능한 특성을 가져야 한다. 기계·기구의 설계 특성은 다음과 같다.

- ① 생산 환경의 측면에서, 사용 목적에 적합한 구조와 성능 특성을 갖는다.
- ② 제작자의 측면에서, 가공 및 조립이 쉬운 특성을 가져야 한다.
- ③ 수리자의 측면에서, 수리 및 정비가 용이한 구조 특성을 가져야 한다.
- ④ 사용자 측면에서, 가격이 저렴하고 사용의 편리성을 갖도록 설계하여야 한다.
- ⑤ 기계·기구 또는 물리적 구성품이 호환성을 갖는 특성을 지녀야 한다.
- ⑥ 소형이고 다루기 쉽고 디자인이 좋은 제품 특성을 가져야 한다.
- ⑦ 가능한 한 소음과 진동이 적은 특성을 가져야 한다.

3 제어부 설계 특성

로봇의 제어부는 컴퓨터 하드웨어 및 소프트웨어가 담당하며, 소프트웨어는 로봇의 제어부에 입력되는 요구 명령과 입력 정보들을 처리하고 그 결과를 저장하거나 출력함으로써, 로봇이 목적하는 기능을 제어하고 동작하도록 하는 역할을 하게 된다. 로봇의 제어부 설계 특성은 다음과 같다.

- ① 기계·기구가 목적하는 제어 및 동작 특성을 갖도록 하기 위하여 적절한 성능의 컴퓨터 하드웨어를 선정 또는 설계하여야 한다.
- ② 제어부 전원은 로봇의 기능과 동작에 충분한 용량의 특성을 가져야 한다.
- ③ 소프트웨어는 컴퓨터 하드웨어가 신속 정확한 처리를 할 수 있도록 적절하며 신뢰성 있고 검증된 특성을 지녀야 한다.
- ④ 컴퓨터 하드웨어 및 소프트웨어의 동작 시 전기적 또는 전자적 잡음(특히 모터 구동 시)에 견디는 특성을 가져야 하고, 소프트웨어는 가능한 한 프로그램 보안을 위하여 디버깅(Debugging) 기능을 가지는 것이 좋다.

4 인터페이스 설계 특성

인간은 눈, 귀, 코, 입, 혀 등 5감이라고 하는 감각기관을 통하여 외부로부터의 정보를 얻는다. 로봇의 경우는 대개 내부에 컴퓨터 기능을 내장하고 있어서 인간의 이 5감 중의 어느 하나 또는 복합적인 정보를 외부로부터 얻고, 외부로부터 얻은 정보는 컴퓨터가 로봇의 내부에서 처리하여 로봇의 기능과 특성에 따라 사용하게 된다. 이러한 경우 외부로부터 정보를 얻기 위하여 사용하는 하드웨어, 소프트웨어 및 네트워크 또는 연결 장치 등을 총칭하여 인터페이스(Interface)라고 부른다.

로봇의 인터페이스 중 대표적인 것으로는 발광 다이오드(LED), 센서 등과 같은 입·출력 장치와 데이터의 입·출력이 들어오고 나가는 마이크로프로세서

인터페이스 설계 (Interface Design)

접속 설계의 뜻으로 특별한 입출력 장치를 요구하는 중앙 컴퓨터 처리 온라인 시스템에 필요한 접속장치를 설계하는 것을 말한다.

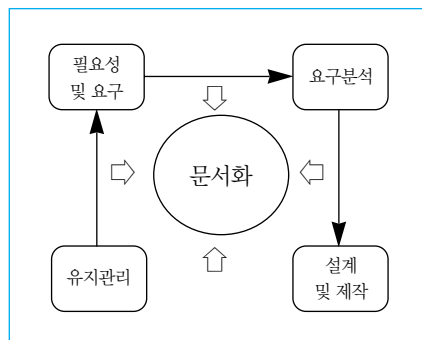


의 포트(Port)를 서로 연결하기 위한 것(장치, 연결선, 연결용 코넥터 등) 등이 있다. 인터페이스 설계 특성은 다음과 같다.

- ① 발광 다이오드, 센서 등의 입·출력 장치는 동작 점검을 위하여 제어부의 브레드보드(Breadboard)에 만들 수 있고, 인터페이스부를 통하여 선으로 적절히 연결할 수 있는 특성을 가져야 한다.
- ② 인터페이스부를 통하여 선으로 연결된 발광 다이오드, 센서 등을 마이크로프로세서 포트에 연결할 수 있는 특성이 있어야 한다.
- ③ 인터페이스용 연결선은 사용과 연결을 쉽게 할 수 있어서 가능한 한 헤드 소켓(Head Socket)을 사용할 수 있는 것이어야 한다.
- ④ 인터페이스부는 연결 시 신호의 연결 또는 접촉이 잘 되는 특성을 갖어야 한다.

5 문서화

로봇에 관한 설계와 제작이 완료되면, 설계에 필요한 각종 자료와 로봇의 기능 및 특성에 관한 자료 및 제작에 필요한 자료를 문서화(Documentation)하며 잘 보관함으로써 다음 설계 또는 제작자가 활용할 수 있도록 하여야 한다. 이러한 문서 자료를 바탕으로 보다 나은 기술로 발전 또는 개선시킬 수 있는 계기가 될 수 있으므로, 모든 자료는 가능한 한 컴퓨터에 의한 정보로 디지털 문서화하여 찾아볼 수 있도록 하며, 장기간 보관할 수 있도록 계획하여야 한다. <그림 IV-1>로봇 설계의 요구에서부터 설계 완료 후 문서화에 이르기까지의 과정을 나타낸다.



<그림 IV-1> 로봇설계 문서화 과정

브레드보드(Breadboard)
전자회로의 개발 또는 원형판을 위하여 회로 구성품과 신호선들을 납땀하지 않고 사용할 수 있는 기판을 말한다.



02 기계·기구 시스템 설계

1 기계·기구 시스템의 기능 및 구동 특성

1 산업용 로봇

산업용 로봇의 기계 장치에서 실행하는 작업 대상은, 실질적 작업을 위하여 3차원 공간에서 이동시키기 위한 로봇의 기구적인 부분이나 로봇 팔 또는 로봇 손에 잡힌 물체이다. 산업용 로봇의 기계 장치는 어떠한 형태의 로봇에서나 구조적으로 몸통, 팔, 손목, 그리고 손으로 구분된다.

3차원 공간에 있는 임의의 물체의 자세(Orientation) 및 위치(Position)를 결정하기 위해서는 위치에 3개, 자세에 3개, 합계 6개의 독립된 변수가 필요하다. 따라서 물체를 3차원 공간에서 이동시켜 요구되는 자세 및 위치를 결정하기 위하여, 로봇에는 6개의 변수 값을 자유로이 변경시킬 수 있는 자유도, 즉 6자유도가 필요하다. 특수한 경우, 수평 다관절 로봇과 같이 6자유도 미만의 자유도를 가지는 로봇이 사용되고 있다.

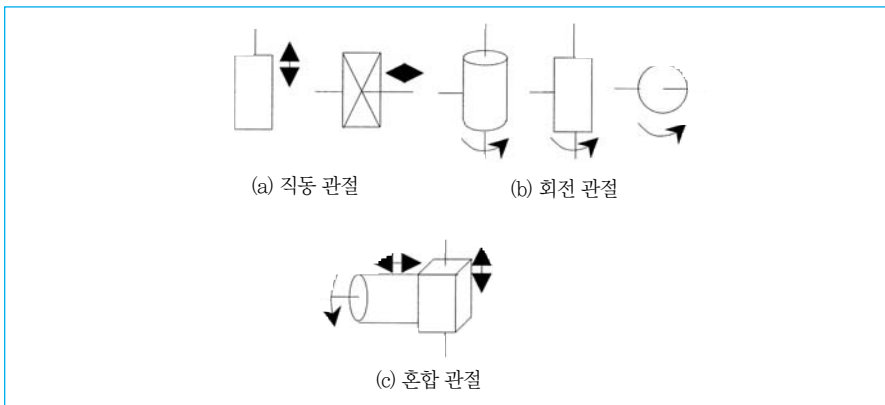
그러나 이들 로봇은 3차원 공간에 있는 임의의 위치에 임의의 자세로 도달할 수 없으며, 따라서 한정된 범위의 작업만을 수행할 수 있는 로봇이다. 예를 들어, 수평 다관절 로봇은 수평면 내에서의 운동만이 가능한 로봇이고 4자유도를 가지고 있다. 그러므로 수평 다관절 로봇은 수평면 내에서의 운동만을 사용하여 작업이 완수될 수 있는 형태의 작업에만 활용될 수 있다.

로봇의 자유도를 결정하는 것은 관절(Joint)이다. 로봇 기구의 기본은 2개의



강체, 즉 링크(Link)를 1개의 관절을 사용하여 직렬로 결합한 것으로, 마지막 링크에 작업용 도구 또는 손을 결합한 것이고, 정지 링크에 해당되는 몸통이 고정되어 있다. 6자유도 로봇 기구의 기본은, 정지 링크(몸통)로부터 팔에 해당되는 처음 3개의 관절은 위치를 결정하기 위하여 사용되고, 손목에 해당되는 나머지 3개의 관절은 로봇의 자세를 결정하기 위하여 사용된다.

로봇의 관절은, <그림Ⅳ-2>에서와 같이 직선운동을 하는 직동 관절(Prismatic Joint)과 축 방향을 중심으로 회전운동을 하는 회전 관절(Rotational Joint) 및 직선운동과 회전운동을 혼합한 혼합 관절 등이 있다.



<그림Ⅳ-2> 관절의 종류

1) 기계기구부 기능

① 몸통

몸통(Body)은 팔과 그 밖의 동작 기관을 지탱하기 위한 부분이다. 일정한 거리를 움직일 수 있는 이동형 로봇과 몸통이 고정되어 있는 고정형 로봇이 있다. 산업 현장에서는 대부분 고정형 로봇을 사용하고 있으나, 반도체 생산 공정 등에서와 같이 이동형 로봇이 사용되는 곳도 있다. 이동형 로봇은 고정형 로봇과 비교하여 활용성은 높으나, 이동을 위한 장치가 필요하여 구조가 복잡하기 때문에 제어의 어려움과 추가 비용이 요구되는 단점을 가지고 있다.

② 팔

팔은 링크와 관절로 구성된다. 팔의 주기능은 손과 손목을 작업 영역의 특정 위치로 보내기 위한 것이다. 팔은 구조와 동작 특성에 따라 원통형, 직각형, 다관절형 그리고 구형으로 구분한다. 원통형은 원통좌표 로봇에서 사용되는 팔이며, 직각형은 직교좌표 로봇에서 사용되는 형태의 팔이다. 다관절형은 수직 다관절 로봇이나 수평 다관절 로봇에서 사용되며, 구형은 극좌표 로봇에서 사용되는 팔이다.



〈그림Ⅳ-3〉 로봇 팔의 12가지 운동

③ 손

엔드 이펙터(End Effector) 또는 그리퍼(Gripper)라고 불리는 로봇의 손은 인간의 손에 해당하는 부분으로, 물건을 잡거나 요구되는 작업을 실제로 수행하는 부분이다. 엔드 이펙터는 작업 공구를 로봇 손의 위치에 부착하여 손 대신으로 사용할 때에 부르는 말이고, 그리퍼는 손가락 또는 손 전체로 물건을 잡는 일을 할 경우에 주로 사용되는 집게 모양의 로봇 손을 지칭하는 용어이다.

산업현장에서 사용되고 있는 로봇 손은 요구되는 작업의 종류에 따라 진공흡



착, 도장(칠하기), 용접, 포크 리프트(Fork Lift : 포크로 들어올리는 것과 같은 작업) 및 여러 가지 손가락과 같은 기능의 작업을 수행한다.

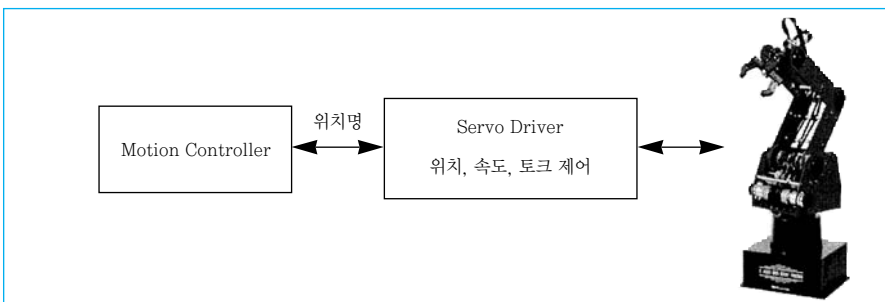
2) 제어부

① 로봇 제어기

로봇 제어기는 로봇에 작업을 명령하고, 각종 원하는 목적의 동작을 수행할 수 있도록 만들어진 제어장치이다. 초기의 로봇 제어기는 공작기계용 NC 장치를 응용한 것으로 8비트 CPU를 사용하여 간단한 작업을 수행하는 정도의 것이었다. 그 후 마이크로프로세서의 발전과 함께 좀 더 복잡한 기능을 구현할 수 있게 되었고, 최근에는 고도의 작업을 필요로 하는 외부 축 및 다축의 주변장치들과의 협조동작이나 2대의 조작기(Manipulator)의 협동작업과 같은 기능이 실현될 수 있게 되었다.

② 로봇 제어기의 종류와 특징

로봇 제어기는 가장 세부적으로 로봇 각 관절의 서보 모터를 제어하는 역할을 한다. 가장 중요한 역할은 로봇 제어기상의 메인 CPU로부터 위치명령을 따르도록 로봇 관절의 서보 모터를 구동하는 일이다. 이러한 일을 수행하기 위해서 로봇 제어기는 하드웨어적으로 모션 컨트롤러(Motion Controller)와 서보 드라이버(Servo Driver)로 구성된다.



모션 컨트롤러

로봇의 동작을 제어하는 것을 말한다.

〈그림Ⅳ-4〉 로봇 제어

모션 컨트롤러는 메인 CPU를 통하여 로봇의 다양한 기능 및 원하는 목적의 작업을 수행한다. 서보 드라이버는 각 관절 서보 모터의 위치, 속도, 토크를 제어하는데, 이는 상위 제어기로부터 명령을 받아 파워 변환을 통하여 이루어진다.

로봇 제어기는 모션 컨트롤러와 서보 드라이버와 같은 하드웨어적 구성과 이를 위한 소프트웨어의 구성을 어떻게 하느냐에 따라 구분된다.

a. 전용 제어기

로봇 메이커에서 그들의 로봇 몸체에 최적화시켜 나오는 제어기이다. 따라서 로봇 제어기의 하드웨어와 소프트웨어 부분에 대해서는 따로 신경 쓸 필요가 없으며, 로봇 제어기에 대한 특별한 전문지식이 없어도 로봇을 쉽게 구동시킬 수 있다.

전용 제어기의 한 예로 4자유도 이상의 로봇에서는 대부분 이러한 전용 제어를 로봇 메이커가 제공한다. 3자유도 이하의 로봇에서는 가격적인 고려가 중요하며 복잡한 계산이 모션 컨트롤러에 필요치 않게 된다. 따라서 다음과 같은 간단하고 소형화가 가능한 다음에 소개하는 조합형 제어기를 많이 사용한다.

b. 조합형 제어기

조합형 제어기는 보통 1, 2, 3축의 로봇에 많이 사용되는 저렴한 제어기이다. 일반적으로 조합형 제어기에서는 독립형 서보 드라이버를 사용하고 별도의 모션 컨트롤러로 구성된다. 이 중에 가장 간단한 것이 1축 제어기인데, 1축에서는 모션 컨트롤러와 서보 드라이버가 하나의 박스 안에 들어가는 것을 많이 사용한다.

조합형 제어기는 일반적으로 사용자가 로봇 제어기의 구성 요소들을 서로 조합하여 만든다. 로봇 제어기는 하드웨어적으로 크게 모션 컨트롤러부와 서보 드라이버부로 구성되기 때문에, 이 두 가지 제품을 완제품 형태로 따로 판매하

자유도

로봇의 움직임은 일반적으로 팔과 몸체 운동, 손목 관절 운동, 다리 및 발목 관절 운동 등으로 나눌 수 있다. 이러한 동작과 연관된 각각의 관절 운동을 자유도라 한다.



는 경우가 있다. 따라서 로봇 사용자는 자기의 로봇과 로봇의 작업 특성에 맞는 제품을 선택하여 제어기를 설정할 수 있다.

2 서비스용 로봇

서비스용 로봇은 오늘날 청소용 로봇이 실용화되고 있다. 청소용 로봇 중에는 수중 청소용으로 개발되고 있는 로봇도 있다. 청소로봇은 초창기 높은 가격으로 인해 실용성이 떨어진다는 지적이 있었지만, 요즘은 여러 기업에서 청소용 로봇을 판매하고 있고 가격도 적정 수준으로 판매되고 있다.

1) 청소용 로봇의 구성과 기능

청소용 로봇은 로봇이 자유롭게 움직일 수 있도록 하는 이동용 모터와 먼지를 흡입할 수 있도록 하는 흡입용 모터로 구성되어 있다.

이동용 모터는 그 속도가 일정해야 하며, 너무 빠르게 움직이면 가구나 가전 제품의 손상을 입게 할 수 있다. 먼지를 흡입하는 흡입용 모터는 먼지를 잘 흡입할 수 있도록 모터의 힘이 좋아야 하며, 소음을 최소화할 수 있는 모터로 구성된다.

이러한 청소용 로봇의 움직임은 센서에서의 감지된 정보를 기초로 움직이게 되는데, 청소 로봇은 정면에 여러 개의 적외선 센서를 부착하여 벽에 부딪치는 경우를 최소화한다. 또한 바닥을 감지하는 센서와 바퀴의 빠짐을 감지하는 센서가 부착되어 있다. 센서는 로봇의 이동경로와 아주 밀접한 관계를 가지고 있으므로 센서에 문제가 발생하면 로봇의 이동에도 문제가 발생하게 된다. 로봇의 정면에는 벽에 부딪쳤을 경우를 대비한 쿠션 역할을 하는 범퍼가 부착되어 있어서 보다 안전하게 청소를 할 수 있다.

서보(servo)

서보는 '추종한다, 따른다'는 의미이며, 명령을 따르는 모터를 서보모터라고 한다. 예를 들어, 프린터, DVD, 공작기계, CCTV 카메라, 캠코더 등에 사용되는 모터처럼, 명령에 따라 정확한 위치와 속도로 제어될 수 있는 모터를 서보모터라고 한다.

2) 제어기능

청소용로봇의 모든 동작을 제어할 수 있는 제어부는 로봇이 잘못된 동작을 하지 않게 하기 위해서 여러 개의 센서와 프로그래밍 된 정보를 잘 제어하여야 한다. 때문에 제어부에서는 센서와 모터 간의 입출력이 원활하고 정확히 이루어져야 한다.

청소용 로봇의 전원은 충전용 고성능, 고효율 리튬이온 배터리를 사용하여 로봇의 작동시간이 충전시간보다 길도록 되어 있다. 또한 배터리가 스스로 사용할 수 있는 용량의 한계를 기억하여 반전과 완전한 충전이 되지 않는 현상을 막을 수 있다.

3) 청소용 로봇의 제어 및 동작

청소용 로봇은 미리 정해진 청소 방법 프로그램을 통해 로봇의 동작과 이동을 제어할 수 있다. 청소 방법 프로그램에는 로봇이 벽을 만나면 일정한 각으로 방향을 전환하여 움직이는 방법이 있고, 계속 원을 그리며 일정한 각으로 이동하는 방법, 그리고 벽면을 따라서 이동하며 청소하는 방법 등이 있다. 이는 사용자가 방법을 설정할 수 있도록 되어 있기 때문에 청소를 시작하기 전에 설정한다.

2 기·계·기·구·시·스·템·설·계

1 설계 시 고려사항

1) 기·계·기·구·시·스·템·설·계·고·려·사·항

기계적인 설계를 위해서는 기계 시스템의 사용 환경에 제약되는 여러 가지 사양이 있다. 요구되는 부하의 크기, 속도, 운동의 정밀도, 기계 시스템의 치수와 중량 기계의 제작에 사용되는 예산 및 관리비용 등의 조건에 따라서 각종 센서,



액추에이터 및 기계요소를 최적으로 선택하여 기계시스템을 설계하여야 한다.

① 기구방식의 설계

요구하는 기구의 특성, 즉 부하의 크기, 이동거리, 이동시간, 이동의 방향성, 작동 특성 및 위치 결정 정밀도 등에 대응 가능한 기구의 방식을 선정하여 설계한다.

② 기본 조건의 설정

기구의 방식에서 요구되는 부하의 종류 및 이동의 방향성과 정밀도 등을 만족할 수 있는 기구를 구성하는 요소와 기기류의 종류를 결정하여 설계한다.

③ 운전 조건의 설정

이동량과 시간의 관계에 따른 가감속도, 모터의 회전 수와 감속비의 관계, 구동 기기와 이송기구 사이의 연결 방식 등을 고려하여 조건을 설정하고 설계한다.

④ 부하 조건의 설정

부하의 크기, 안내 요소의 마찰계수 및 예압의 하중, 가소시간, 굽힘 모멘트에 의한 좌굴 하중의 유무, 외부의 하중과 방향성 등을 고려하여 설계한다.

⑤ 부하계산

부하 조건과 운전 조건의 설정에 따라서 마찰에 의한 부하, 실제의 작업상에서 발생하는 부하, 관성에 의한 부하를 계산하고 오차를 최소화하여 설계한다.

⑥ 사용기기의 크기 선정

운전 조건의 부하계산과 설정조건에서 계산한 결과를 이용하여 적당한 크기를 선정하여 설계한다.

2 이동용 로봇 설계 고려사항

1) 라인트레이서의 센서

라인트레이서의 센서는 선을 잘 감지할 수 있는 적절한 기능의 적외선 센서

를 사용한다.

2) 이동용 로봇의 설계 및 제작

주로 대부분 많이 사용하는 이동용 로봇으로서 라인 트레이서에 관한 설계 및 제작 과정을 살펴보자. 라인 트레이서는 하얀 선이 그려져 있는 미로판에서 정해진 하얀 선을 따라 자율적으로 이동하는 로봇이다. 라인 트레이서의 센서는 적외선 센서를 이용하고, 모터는 일반적으로 DC 모터를 사용한다

① 주제어 기판

주제어 기판은 마이크로 컨트롤러와 전원부, 메모리 등이 포함되어 있다. 주제어 기판은 작성한 프로그램에 의해 움직이도록 제어하는 회로들로 이루어져 있다.

② 센서 기판

센서 기판은 라인 트레이서의 눈과 같은 역할을 하는 센서들이 부착된 기판으로 주제어 기판과 커넥터를 사용하여 연결시킨다.

이동을 통해 센서로 감지되는 각종 정보는 센서에서 제어 기판으로의 입력과 제어 기판에서 모터로의 제어가 이루어진다.

③ IC 부품과 커넥터

주제어 기판과 센서 기판에 장착될 부품으로서 분실하거나 손상되지 않도록 관리한다. 각 부품들은 센서와 제어부의 연결에 영향을 미치므로 정확히 설치하여야 한다.

④ 차체

라인 트레이서의 차체는 몸체와 바퀴 등으로 구성된다. 몸체에 설치될 모터 박스와 바퀴가 잘 작동할 수 있도록 제작한다. 모터는 라인 트레이서의 제어부에서 받은 정보로 로봇의 이동에 많은 영향을 준다.



⑤ 통신케이블 충전 어댑터

라인 트레이서 건전지를 이용할 수 없을 경우 전원 장치로 전원을 공급하며, 또한 제작한 프로그램을 통신 케이블을 연결하여 라인 트레이서에 다운로드할 수 있다.



03 센서

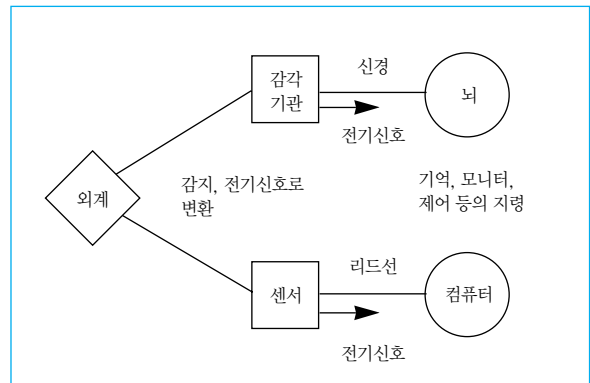
1 개요

센서는 1960년부터 사용된 이래 여러 가지 표현으로 불려 왔으며, 일반적으로 “측정 대상물로부터 해당 정보를 감지 또는 측정하여 그 측정량을 전기적인 신호로 변환하는 장치”로 정의된다.

한편 센서는 인간의 오감(시각, 청각, 후각, 미각, 촉각)의 역할을 대신하는 기기로 비유된다. 인간의 감각기능과 기계장치의 센서를 비교해 보면, 감각기관은 외부로부터 오감을 통하여 얻은 자극이 전기신호로 변환되고 이것이 신경을 통해 뇌에 전달되어 적절한 판단에 따라 손과 발을 제어하도록 정보화 하는 데 비해, 센서는 대상물로부터 검출한 신호를 센서에서 적당한 형태의 신호로 변환하고, 신호 전송로를 경유하여 제어기에 전달되며, 예정된 작업순서나 내용에 따라 처리되어 출력단을 구동하게 된다.

인간의 오감에 따른 센서를 정리해 보면 <표 IV-1>과 같다.

센서는 센스(sense)에서 유래되었으며, 비슷한 용어로 트랜스듀서(transducer)가 있다.



<그림 IV-5> 감각기관과 센서의 비교

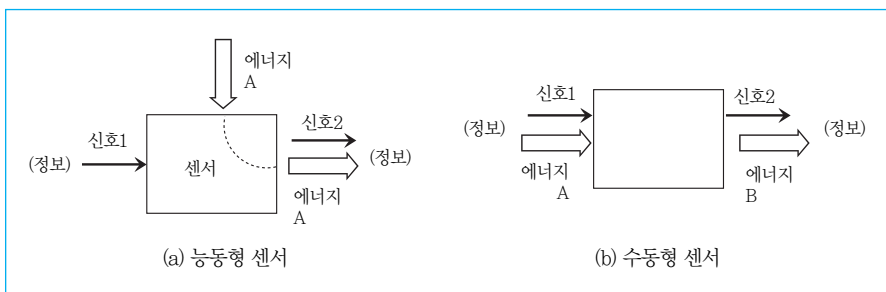


감각	기관	물리현상, 물리량	반도체 센서	
시각	눈	가시광 결상	광전 변환장치	광전도 장치, 포토 다이오드, CCD 이미지 센서
청각	귀	음파 진동	압력전기 변환장치	피어조 저항장치, 감압 다이오드
촉각	손가락, 피부	변위 압력	변위전기 변환장치	피어조 저항장치, 비틀림 게이지
후각	코	확산 흡착	가스 센서, 흡착 센서	
미각	혀	용해 흡착	이온 검출 FET	

〈표Ⅳ-1〉 인간의 감각과 반도체 센서

2 센서의 분류

센서는 종류가 다종다양하여 뚜렷이 분류하기가 어렵다. 대체로 구성별, 검출 신호별, 재료별 등으로 분류되는데, 일반적으로 대상물의 접촉 여부에 따라 접촉형, 비접촉형으로 분류되고, 또한 대상물로부터 정보를 획득하는 방법에 따라 능동형(active)센서와 수동형(passive) 센서로 분류되기도 한다.



〈그림Ⅳ-6〉 능동형 센서와 수동형 센서

능동형 센서 : 광전도 셀(CdS) 및 서미스터(thermistor)는 빛 혹은 온도에 의해 전기 저항값이 변하는 센서로, 출력 전력은 빛이나 열이 변환된 것이 아니

다. 출력 전력은 저항 측정을 위한 회로 전원으로부터 공급된 전력의 일부분일 뿐이다. 이처럼 검출 대상으로부터 검출된 신호(빛 또는 열)에 의해 전원에서 출력으로 에너지 흐름이 제어되는 센서가 능동형 센서이다.

수동형 센서 : 태양전지나 열전대에서 출력신호 전력은 센서가 흡수한 빛 혹은 열에너지의 일부이다. 이처럼 외부로부터 별도의 전원 공급이 필요하지 않고, 입력신호 에너지의 일부가 변환되어 출력되는 센서가 수동형 센서이다.

단 계	센서 구분
구성 방법	기본 센서, 조립 센서, 응용 센서
검출 신호	아날로그 센서, 디지털 센서
검출 기능	화학 센서, 물리 센서, 역한 센서, 시각 센서, 촉각 센서 등
제작 재료	반도체 센서, 세라믹 센서, 금속 센서 등
정보 획득	능동형 센서, 수동형 센서
접촉 여부	접촉형 센서, 비접촉형 센서

〈표Ⅳ-2〉 센서의 분류

여기서는 센서의 내용을 스위치에 국한하여 위의 “제어용 전기기기”의 “검출용 스위치”에서 제시되었던 내용을 기준으로 설명한다.

3 접촉형 스위치

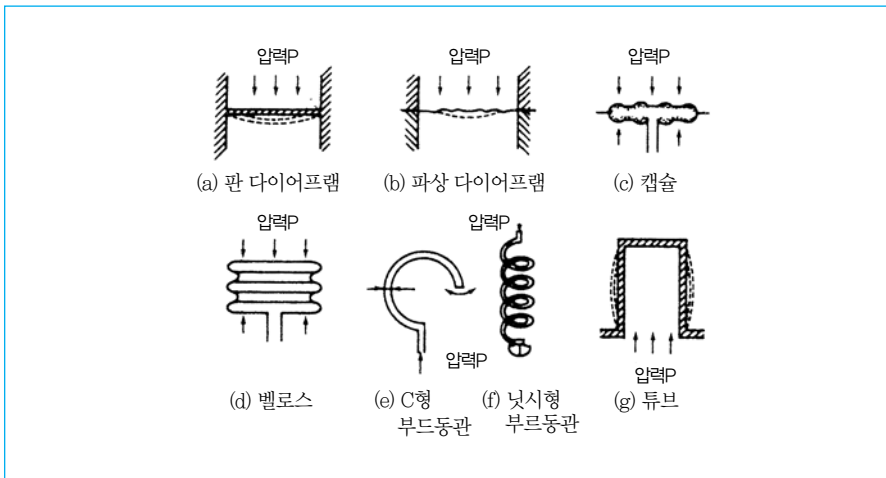
접촉형 스위치에는 리밋 스위치, 압력 스위치, 리드 스위치가 있는데, 리밋 스위치는 “검출용 스위치”에서 설명했으므로 여기서는 압력 스위치와 리드 스위치에 대해 설명한다.

1 압력 스위치(pressure switch)

압력 스위치는 명칭과 같이 일정한 압력에 이르면 스위치가 ON이나 OFF되는 스위치이다. 압력을 측정하는 원리는 변위, 변형, 진동수 등을 이용하는 것으로, 최근에는 반도체 기술의 발전으로 더욱 소형화, 복합화된 다기능의 센서가 출시되고 있다.

압력 센서의 종류는 크게 기계식, 전기식, 반도체식이 있으며, 다시 기계식은 부르동관(Bourdon tube), 다이어프램, 벨로스(bellows)가 있다. 이 중에서 부르동관이 가장 많이 쓰이며, 단면이 원상이고 개방된 고정단으로부터 측정 압력을 도입하면 다른 밀폐된 관의 선단이 이동하는 원리를 이용하고 있다. 또한 다이어프램은 압력 차에 비례한 원판의 휨 정도로, 벨로스는 원통 내부와 외부의 압력 차에 의해 주름상자가 신축하여 그 변위량이 압력 차에 비례한 것으로부터 각각 압력을 측정한다.

부르동관은 19세기 중엽 프랑스의 E.부르동(E.Bourdon)이 고안하였다.



〈그림Ⅳ-7〉 기계적 압력 변환요소

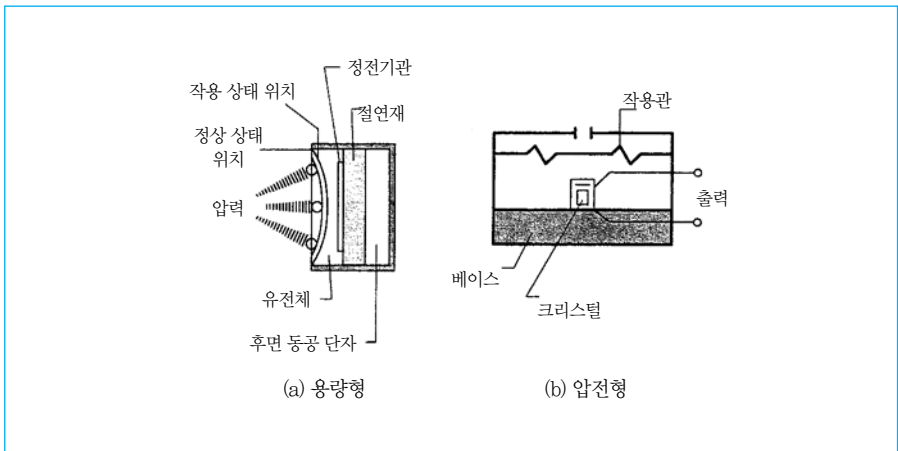
전기식은 대부분 기계적인 변위를 전기신호로 변환하는 부분이 기계식과 다를 뿐 기본적으로 기계식과 동일하다. 여기에는 두 개의 물체 간의 정전용량의 변화를 이용한 용량형, 스트레인 게이지를 이용한 압저항형, 유기 또는 무기압전 소자를 이용한 압전형 등이 있다.

반도체형은 최근에 실용화가 가속되고 있는 센서로 직선성이 좋고, 소형경량으로 진동에도 매우 강한 것이 특징이다. 또한 기계식보다 고감도, 고신뢰성이며 양산성이 좋다. 반도체형은 압력을 왜응력으로 변환하는 다이어프램과 다이어프램에서 발생하는 동력을 전기신호로 변환하는 두 부분으로 구성되어 있다.

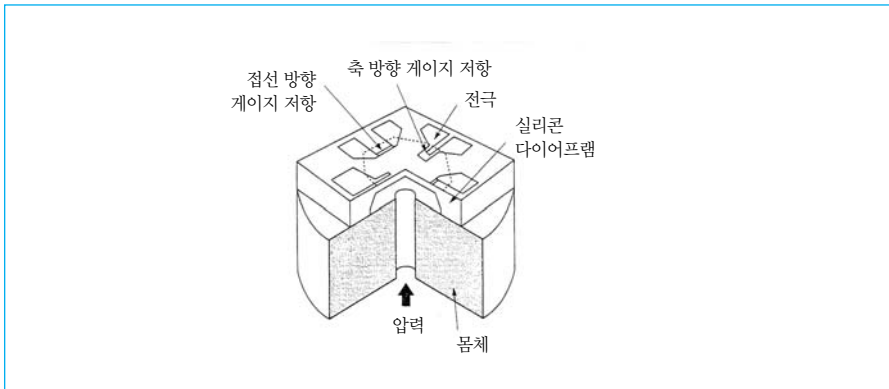
다이어프램은 단결정 실리콘을 화학적으로 에칭(etching)하여 형성하며, 다이어프램에서 발생하는 응력을 압저항식과 정전 용량식의 방법으로 전기적인 신호로 변환한다. 이 중 압저항식은 저항확산식 또는 확산식이라고도 하는데, 압저항소자를 형성시킬 때에 반도체의 불순물 확산 공정이 이용되기 때문이다.

에칭(etching)

반도체 표면의 부분을 산 따위를 써서 부식시켜 소거하는 방법으로, 반도체 직접 회로의 제조 공정 따위에 쓴다.



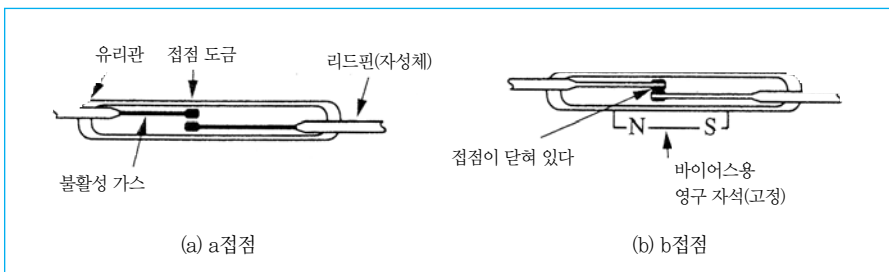
〈그림 IV-8〉 전기적 압력 변환요소



〈그림Ⅳ-9〉 반도체 압력센서

2 리드 스위치(reed switch)

리드 스위치는 자기 현상을 이용한 것으로 고빈도 개폐 사이클이 요구될 때 적절하며, 기계적 리밋 스위치를 설치할 공간이 없을 경우에 사용된다.

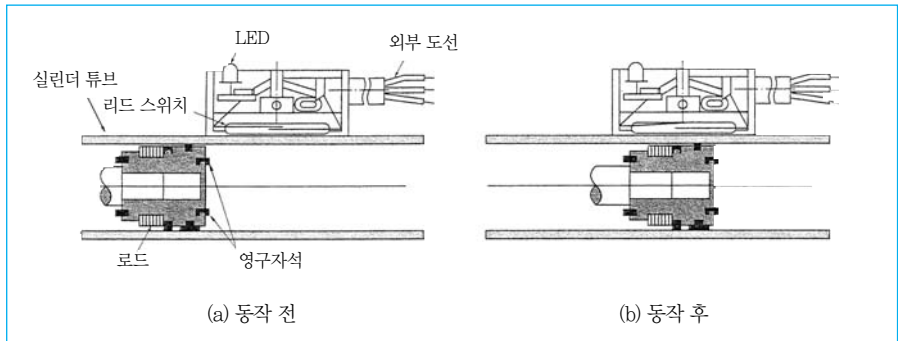


〈그림Ⅳ-10〉 리드 스위치의 구조

리드 스위치는 〈그림Ⅳ-10〉과 같은 구조로, 접점 부분이 비활성 가스를 충전한 유리관 속에 봉입되어 있다. 코일에 전류를 흘리면, 자력선은 자성체로 만들어진 리드 속을 좀 더 많이 통과하기 때문에 자기적 흡인력이 생겨 접점이 접촉하게 된다. 이때 코일에 흐르는 전류를 차단하면, 외부 자계가 없어져 기계적 탄성에 의해 최초 접점 상태로 돌아간다. b접점의 경우는 정상상태에서 바이어

스(bias)용 영구자석에 의해 접점이 붙어 있다가 코일에 전류가 흐르면 반발력에 의해 접점이 열린다.

〈그림Ⅳ-11〉은 실린더의 위치를 감지하는 리드 스위치의 동작을 보여준다.



〈그림Ⅳ-11〉 리드 스위치의 동작

4 비접촉형 스위치

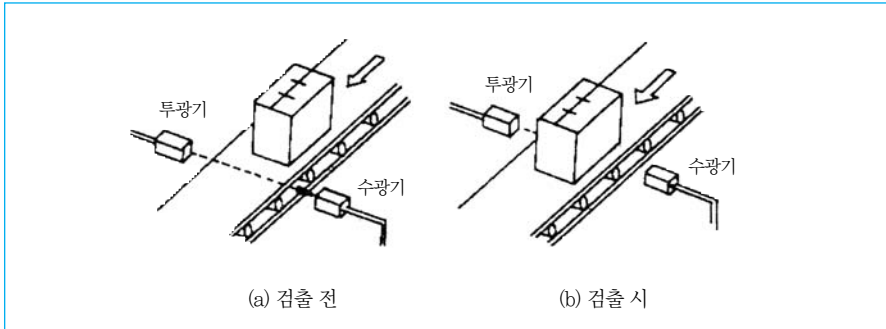
비접촉형 스위치는 광전 스위치, 근접 스위치, 초음파 스위치가 있다.

1 광전 스위치(photoelectric switch)

광전 스위치는 빛 자체를 정량적으로 검출하기보다는 빛을 이용하여 물체의 유무를 검출하는 것을 말하며, 빛을 방사하는 투광기와 광량의 변화를 전기신호로 변환하는 수광부, 수광부로부터 신호를 증폭하고 접점을 개폐하는 증폭기로 구성된다.

광전 스위치를 〈그림Ⅳ-12〉처럼 배치해 두면, 물체가 광로를 차단하였는가의 여부에 따라 스위치 접점을 ON, OFF 하므로 물체의 유무나 위치를 검출할

수 있다.



〈그림Ⅳ-12〉 광전 스위치의 사용예

〈그림Ⅳ-12〉의 (a)와 같이 물체가 광로 밖에 있으면, 투광기의 빛이 수광기에 들어가서 스위치는 동작하지 않는다. 그러나 (b)와 같이 물체가 광로를 차단하면, 투광기로부터의 빛이 차단되어 스위치가 동작하게 된다.

광전 스위치는 투광기와 수광기의 구성에 따라 다음과 같은 종류로 나눈다.

① 투과형(through beam)

투광기와 수광기를 동일 광축선상에 서로 마주보게 설치해 두고, 그 사이를 통과하는 물체에 의해 광량의 변화가 발생하는 것을 검출하여 동작된다.

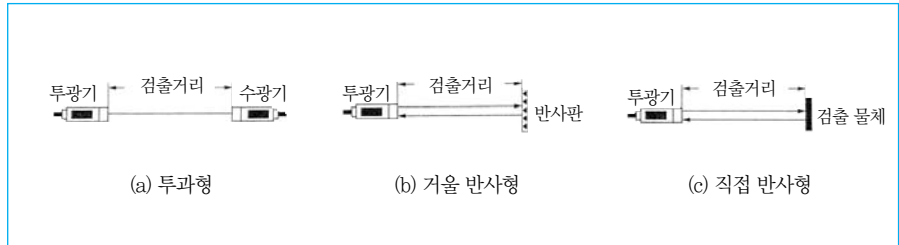
② 거울 반사형(retro-reflection)

투광기와 수광기를 동일한 케이스 내에 조립하여 이것을 반사거울을 마주보게 설치하고, 그 사이를 통과하는 물체에 의해 광량의 변화가 발생하는 것을 검출하여 동작된다.

③ 직접 반사형(diffusion)

직접 반사형 역시 거울 반사형과 마찬가지로 투광기와 수광기가 일체로 되어

있으며, 투광기로부터 투사된 빛이 검출 물체에 부딪혀서 반사된 광을 수광부가 직접 검출하여 동작된다.



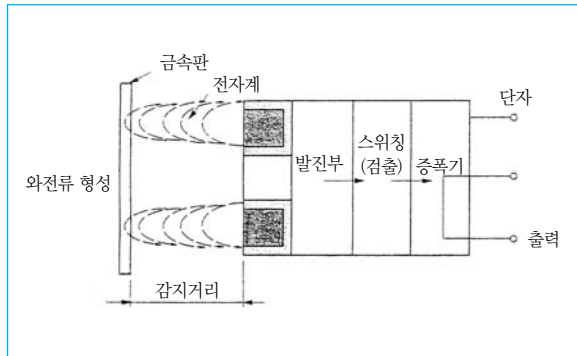
〈그림Ⅳ-13〉 광전 스위치의 종류

2 근접 스위치(proximity switch)

광전 스위치와 동일하게 물체에 접촉하지 아니하고 그 유무나 위치를 검출하는 스위치로서 구동 방식에 따라 크게 유도형(inductive)과 용량형(capacitive)으로 나뉜다.

먼저 유도형은 금속체에만 반응하는 것으로 고주파 전자계를 센서의 표면에 방출한다. 이때 검출 헤드(발진 코일) 가까이 금속체가 없으면 변화가 없고, 도체인 금속성 물체가 감지거리 이내에 들어오면 발진 코일은 금속체에 에너지를 빼앗겨 발진 진폭이 감소되어 금속체를 감지한다.

용량형은 절연된 도체를 전극으로 하여 다른 도체가 있는 유도체가 접근하면 전극이 정전용량 변화를 전기신호로 변환하여 물체의 접근을 검출한다. 유도형 센서는 와전류 형성을 위한 금속체만을 검출할 수 있는데 비해, 용량형은 분극 현상을 이용하므로 비금속 물체도 검출이 가능하다.



〈그림Ⅳ-14〉 유도형 센서의 구성

③ 초음파 스위치

초음파란 보통 인간의 귀로 들을 수 없는 20[Khz] 이상의 음파를 말하며, 물체의 유무 혹은 특정한 물체까지의 거리를 측정하는 데 이용되는 초음파 센서는 대략 30[Khz] - 300[Khz]의 주파수를 사용한다.

일반적으로 초음파의 발생이나 검출은 크게 전자유도 현상, 자왜 현상, 압전 현상 중 어느 하나를 이용하며, 전기신호를 초음파로 변환하는 송신기와 초음파를 전기신호로 변환하는 수신기로 구성된다.

초음파 센서의 형태는 일반적으로 송신기와 수신기가 한 물체로 되어 있으나, 〈그림Ⅳ-15〉에서와 같이 송신기와 수신기가 떨어져 있는 형태의 구조도 있다.


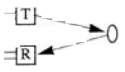

직접형은 송신기와 수신기를 서로 마주보게 배치하여 수신기에 초음파가 감지되면 물체가 존재하지 않고, 초음파가 감지되지 않으면 물체가 있는 것으로 인식한다. 반면에 확산형 즉, 반사 방식은 독립형과 겸용형이 있는데, 어느 것이나 반사파가 있을 때 물체가 존재하는 것으로 인식한다.

초음파 센서는 다양한 물체를 검출할 수 있다는 장점이 있지만 스위칭 주파

자왜(磁歪)현상

강자성의 물질을 자화할 때 그 물질에 탄성적 변형이 생기는 현상으로 자기 일그러짐 현상이라고도 한다.

수가 낮고 광센서에 비해 고가이다.

배치 방식	내용
직접형 	<ul style="list-style-type: none"> · 감지 감도를 자유로이 설정할 수 있어 설계가 쉽다. · 설치 장소가 2곳이 필요하다.
반사방식(독립형) 	<ul style="list-style-type: none"> · T에서 R로 직접 들어가는 것에 대한 대책 필요 · T와 R전용 센서를 사용할 수 있으며, 효율이 좋다. · 근거리(10cm 이하)인 경우가 많다.
반사방식(겸용형) 	<ul style="list-style-type: none"> · 송수신 전환 회로가 필요 · 근거리는 측정할 수 없다.

〈그림Ⅳ-15〉 초음파 센서의 형태



04 제어 · 센서부 설계

1 센서부 설계

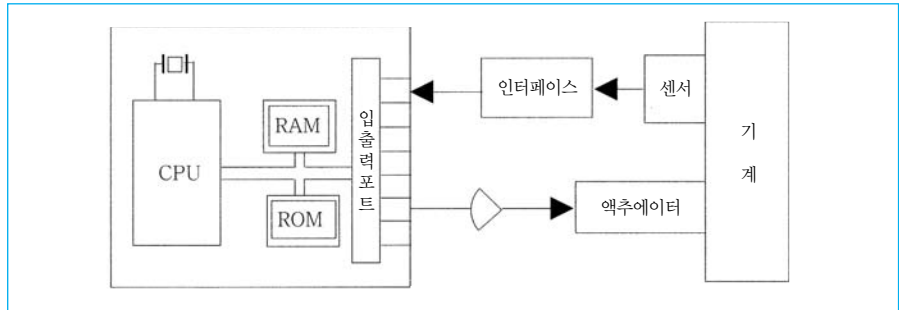
인간 한계를 뛰어넘어 우리를 대신하여 작업을 수행하고, 나아가 스스로 판단할 수 있는 지능을 가지고 업무를 처리하며 자동화 시스템의 제어를 독립적으로 담당하는 지능형 로봇에게 필요한 센서를 어떻게 설계할지 살펴보자.

로봇에서 센서는 사람의 감각과도 같은 중요한 요소로, 최근에 주로 연구되고 있는 센서들은 거리 측정 및 접촉 여부, 방향성, 변위 등을 로봇에게 전달할 수 있는 기능은 물론, 특히 비전 센서와 같이 많은 정보를 시스템에 전달하는 특성을 지니고 있다. 이 경우 CCD 카메라를 통해 들어온 이미지는 프레임 그래버(Frame Grabber)를 통해 컴퓨터로 영상이 전달된다.

이러한 자료는 인간형 로봇, 안내 로봇, 경비 로봇 등과 같이 이동하는 형태의 로봇에서뿐만 아니라, 카메라의 입력영상에 포착된 특정한 물체를 선택하고 카메라가 선택한 물체를 따라 이동하는 감시기능을 지닌 지능형 로봇 등의 센서들에 많이 사용된다.

특별한 응용에서 어떤 센서가 최적인지 또는 시스템에서 필요한 센서가 무엇인지를 결정하는 것이 바로 설계자가 담당해야 할 중요한 부분이다.

다음에 실제로 많이 사용되는 로봇용 센서 설계 시에 고려할 사항들에 관하여 살펴보자.



〈그림Ⅳ-16〉 센서 구동 구성도

1 인체 열 감지 센서 설계 고려사항

빛의 발생 근원을 찾아내고 인간이나 동물의 움직임을 검출할 수 있는 경비로봇과 재난 구조 로봇 등에 활용할 수 있다. 일반적으로 밝은 곳에서 양초의 불빛 정도를 검출할 수 있는 능력이 있으며, 검출하고자 하는 대상이 지닌 방사율에 따라 감도를 설정한다. 빛의 간섭 실험용 Fresnel 렌즈는 넓은 범위를 감지할 수 있게 감지 물체 쪽으로 초점을 맞추어 적합하게 설치되도록 설계하여야 한다.

2 자이로 센서 설계 고려사항

RC 비행기, RC 헬리콥터 등에 삽입되어 기능을 수행하기 위해 소형, 경량, 저전력으로 제작되고 있다.

1개의 자이로 센서를 이용하여 1축 로봇의 회전을 검출할 수 있도록 설계하여야 하며, 여러 개의 자이로 센서를 복합적으로 구성하는 경우는 다축의 회전을 검출할 수 있도록 설계하여야 한다

3 기울기 감지 센서 설계 고려사항

케이스에 장입된 수은에 의해 동작되는 것으로 직경 6mm, 길이 9mm 정도의



규격이며 $\pm 30^\circ$ 또는 $\pm 90^\circ$ 의 동작 각도 설정이 가능하도록 설계하여야 한다.

4 굽힘 센서 설계 고려사항

굽힘 각도에 따라서 적절히 저항 값이 변하도록 설계하여야 하며, 평상시 약 $10\text{k}\Omega$ 정도에서 90° 가량 휘었을 때는 $35\text{k}\Omega$ 정도까지 변화하는 특성을 갖는다.

5 각속도 센서 설계 고려사항

운동하고 있는 물체 또는 속도를 가지고 있는 물체가 회전하면, 그 속도 방향에 수직으로 “코리올리스”의 힘이 작용한다. 이러한 물리현상을 이용하여 각속도를 검출하는데, 일반적으로 아날로그 신호 출력 형태를 가진 자이로의 신호를 A/D를 거쳐 적절한 디지털 신호로 변환하여 제어부에서 처리할 수 있도록 설계하여야 한다. 이때 제어부에서 검출된 각속도를 적분 연산을 하게 되면 각도를 산출할 수 있다.

6 가속도 센서 설계 고려사항

물체의 운동 상태를 순서적으로 감지하거나 충격, 진동 등의 동적 힘을 감지하여 기울기 또는 각속도 등을 측정할 수 있도록 설계하여야 한다. 로봇 및 공장자동화 등의 제어 시스템에 있어서 필수적인 소자로서, 검출 방식에 따라서 크게 분류하면 자이로식, 관성식, 실리콘 반도체식 등이 있다.

7 영상 프로세싱 센서 설계 고려사항

원칩 마이크로 콘트롤러 계열인 PIC 칩 등과 직렬 (Serial)로 직접 인터페이스 되어 이동 로봇의 실시간 영상 처리가 가능하도록 설계하여야 한다. 이것은

로봇 머리에 이용되어 물체를 추적하거나 쉽게 물체를 탐지하는 역할을 수행하도록 하여야 한다.

③ 촉각 센서 설계 고려사항

1mm 정도의 공간 분해능을 가진 촉각 센서를 인간형 로봇의 표면에 인공 피부로 사용하여 다양한 힘과 온도를 감지할 수 있도록 설계하여야 하며, 인간형 로봇이 사람처럼 뜨거움과 차가움, 민감한 접촉 등을 구분할 수 있는 기능을 갖출 수 있어야 한다.

지금까지 개발된 촉각 센서는 자극 형태를 인지하고 약 80개 정도의 반응을 보이는 수준까지 이르렀는데, 이러한 촉각 센서는 로봇의 수요와 더불어 앞으로 그 개발이 크게 증가할 것으로 기대된다.

2 모터부 설계

모터는 로봇의 이동에 필수적인 역할을 담당한다. 로봇 내부의 컴퓨터와 모터 간에 인터페이스를 통하여 로봇 내부의 컴퓨터가 필요한 정보를 출력하면, 해당 모터의 코일(3상 또는 4상)에 여자전류가 흘러 모터를 구동하게 함으로써, 로봇이 필요한 동작을 하게 된다.

① DC 모터 설계 고려사항

DC 모터는 가장 쉽게 사용할 수 있는 모터로서 컴퓨터의 플로피디스크 드라이브, 어린이들의 장난감 또는 CD-ROM 드라이브 내에서 널리 쓰이고, 수명



은 비교적 짧다.

일반적으로 사용되는 전자회로는 간단하지만, 특성을 살려 원하는 동작을 위한 제어회로를 구축할 때에는 주변 회로 및 하드웨어가 많이 필요하다. 또 DC 모터는 시동, 가속토크를 임의로 선택할 수 있어 토크 효율이 좋지만, 정류자를 가진 구조이기 때문에 높은 전압 입력이나 고속화의 경우에 제한이 있다.

DC모터는 광범위하고 높은 정밀도의 속도 제어가 가능하도록 설계하여야 하며, 또한 여자 방식에 따라 다른 특성이 나타나기 때문에 부하에 대한 적응성, 시동 토크 등을 고려하여 설계하여야 한다. 시동 토크가 커서 가변속 제어나 큰 시동 토크가 요구되는 시스템의 경우에도 잘 작동될 수 있도록 설계하여야 한다.

2 기어드 DC 모터 설계 고려사항

DC모터의 특성을 가지지만 DC 모터와 함께 기어박스가 같이 부착되어 일반 DC 모터의 회전수를 감속시키거나 늘릴 수 있고, DC모터의 토크 또한 일정 범위 내에서 조절이 가능하다. 반대로 발전기에서 응용할 때 비교적 높은 효율의 발전기를 제작할 수도 있다. 기어드 DC모터는 기어를 이용하므로 기계적 특성에 따라 워기어, 일반 기어드 유성기어 등 다양하게 활용하여 설계할 수 있다.

3 서보모터 설계 고려사항

DC 서보 모터는 전기자 전류에 대하여 발생 토크의 관계가 직선성으로 우수하고, OA(사무 자동화)에서부터 FA(공장 자동화) 분야까지 광범위하게 사용되고 있다. AC 서보 모터는 주로 항공기 관계, 계측기에 사용되었으나 현재는 거의 사용되지 않는 실정이다.

속도 제어, 전류 제어, 전력 변환장치, 위치 제어 등의 기능을 가지므로 각 기능에 맞도록 제어할 수 있도록 설계한다.

4 스텝핑 모터 설계 고려사항

스텝핑 모터는 펄스 신호를 이용하여 보내진 펄스만큼 회전하며 펄스가 끝난 후 모터 자체적으로 정지 토크가 발생한다는 특징을 가지고 있고, 제어펄스의 속도에 따라 회전 속도가 결정이 된다. 그러나 DC모터에 비해 전력 소모가 많고 크기가 크며, 모터에 따라 특정 주파수 영역에서 공진과 함께 진동이 발생할 수도 있는 것이 단점이다.

3 인터페이스 설계

센서를 이용하여 정보를 읽고 모터를 움직이거나 스위치 입력을 받을 수 있으려면 반드시 입출력 장치를 통해야 한다. 즉, 입출력 장치가 마이크로프로세서와 외부세계를 연결하는 인터페이스 역할을 하는 것이다. 그러므로 인터페이스 설계는 하드웨어와 소프트웨어 사이의 정보를 정확히 주고받을 수 있으며 서로를 제어할 수 있도록 설계해야 한다.

1 제어부와 센서부 인터페이스

로봇에 설치되어 있는 여러 종류의 센서는 제어부와 입·출력 동작을 할 수 있도록 설계되어야 한다. 센서에 감지된 정보를 제어부에서 기계·기구부와 모터부 등으로 상황에 맞는 동작을 할 수 있도록 연결되어 있으므로 제어부가 동작을 제어할 수 있게 된다. 예를 들어 이동 로봇에서 센서가 벽을 감지한 경우 그 정보를 제어부에 보내어 다음 동작을 할 수 있도록 한다.

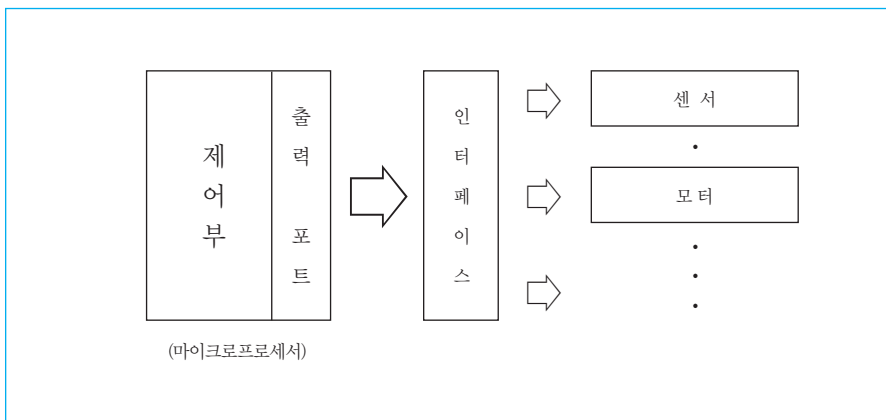
센서의 기능에 따라 제어부에 입력되는 정보가 다르기 때문에, 제어부는 각 센서들의 입력에 맞는 동작을 할 수 있도록 프로그램되어야 한다.



2 제어부와 모터부 인터페이스

외부로부터 받은 정보가 로봇의 이동에 대하여 제어가 필요할 경우 제어부에서는 모터의 상황에 맞는 동작을 할 수 있도록 제어 신호를 보내게 된다. 모터부는 로봇의 이동과 관련이 있기 때문에 모터부의 제어가 정확히 이루어지도록 제어부가 프로그램 되어야 한다.

3 입출력 인터페이스



〈그림Ⅳ-17〉 센서 및 모터 제어 구성도

모터를 제어하는 기능을 갖는 제어부는 센서로부터 얻은 제어 정보를 통해, 동작되고 있는 모터의 동작 환경에 대한 정보를 입력하여 모터가 감속할 것인지 계속 유지할 것인지 제어한다. 또한 로봇의 이동 방향의 정보를 모터에 보내어 로봇이 방향을 전환할 수 있도록 한다.

Flash Memory는 휴대가 가능하며 Read(읽기), Write(쓰기) 모두 자유로움을 말한다. 반면에 ROM(Read Only Memory)은 일반적으로 휴대가 불가능하며 Read(읽기)만 가능한 ROM을 말한다.



05

프로세서부 설계

1 마이크로프로세서의 기능 및 특성

MPU는 Micro Processing Unit, MCU는 Micro Control Unit 라는 뜻이고, 둘 다 마이크로프로세서 8051처럼 한 칩(Chip) 속에 I/O(입출력) 장치와 메모리 등을 작은 한 개의 칩 속에 모두 집적시켜 밥솥이나 냉장고, 여러 가지 산업용 장비 등에 응용하기 위한 용도로 사용하는 CPU(중앙처리장치: Central Processing Unit)를 말한다. 이러한 경우 CPU는 제어장치(Controller)의 기능을 갖는다. 일반 CPU와 다른 점은 MPU나 MCU는 한 개의 칩 속에 기계 제어를 위한 모든 장치(Device)가 집적되어 있다는 것이다. MPU라고 하면 일반 CPU도 포함되기도 한다. 한편, 마이크로프로세서(Microprocessor)는 소형의 CPU를 일컫는 말이었으나, 오늘날 마이크로프로세서는 CPU와 혼용해서 사용하는 것이 보통이다.

MPU는 CPU와 동의어로도 많이 쓰인다고 볼 수 있다. 그러나 일반적으로 한 칩(One Chip)으로 된 것은 원 칩 마이크로 제어장치라고 하거나 MCU라 한다.

1 89×51 마이크로프로세서

89C51은 4KB 롬(ROM) 용량의 원 칩 마이크로 컨트롤러로서, 프로그램 메모리가 Flash Memory(ROM) Type이며 전용 라이터(Writer)를 이용하여 프로그램 쓰기(Write)를 할 수 있고, 다시 쓰기(Rewrite)도 가능하다. 한편, 89C51의 구형 모델인 8051은 프로그램 메모리가 Mask ROM Type으로 되어 있어서 제조 시 프로그램을 탑재시키거나, 외부에 프로그램 메모리를 추가하여 사용하여야 한다.



한편, 89S51은 89C51처럼 프로그램 메모리가 Flash Memory이면서 전용 라이터를 이용하여 프로그램 쓰기를 할 수 있고, 다시 쓰기도 가능하다. 89C51과 89S51과의 차이점은 89C51은 CPU를 보드에서 분리하여 전용 라이터 장비에 장착하여야 하나, 89S51은 보드에 장착되어 있는 상태로도 가능하다. 이런 기능을 In-System Programming이라고 부른다.

2 89×52 마이크로프로세서

89S52는 8KB 롬 용량의 원 칩 마이크로컨트롤러이며, 가운데 S가 들어가 있는 모델은 롬라이터 없이 ISP방식으로 쓰기(Write)가 가능한 장치이다. C가 있는 모델은 롬라이터가 필요하다. 또한 51과 52의 차이는 롬 용량이 각각 4k와 8k, 내부 램 용량이 128바이트와 256바이트의 차이가 있으며, 타이머도 51은 2개, 52는 3개가 있다

마이크로프로세서명	롬(ROM) 기억용량	램(RAM) 기억용량	칩(Chip) 형태	롬 형태	롬 쓰기 장치
8051	0KB	0KB	CPU 한 개	마스카(Mask) 롬 사용 가능	별도 필요
89C51	4KB	128B	원 칩(One Chip)	플래시 메모리형 (Flash Memory)	별도 필요
89S51	4KB	256B	원 칩(One Chip)	플래시 메모리형 (Flash Memory)	별도 필요치 않음
89C52	8KB	128B	원 칩(One Chip)	플래시 메모리형 (Flash Memory)	별도 필요
89S52	8KB	256B	원 칩(One Chip)	플래시 메모리형 (Flash Memory)	별도 필요치 않음

〈표Ⅳ-3〉 8××5× 계열 마이크로프로세서의 차이점 비교

52를 사용한 시스템에 51을 사용하려면, 프로그램을 작성할 때 52에만 있는 타이머와 램을 사용하지 않고 51과 52의 공통인 부분만 사용하면 되며, 프로그램 크기가 4KB 이내로 하면 문제없다(예 : 타이머 2, 상위 128바이트 램 등). 이러한 제한적인 용도 범위 내에서 사용하면 89C52 사용 시 그대로 호환이 된다. 핀은 같으므로 별도의 하드웨어 조작용은 필요없다.

2 기타 로봇 설계에 많이 사용하는 마이크로프로세서

미국 인텔(INTEL)사의 80계열로 앞에서 설명한 89C51과 89S51 및 89C52와 89S52 외에 8031, 8032, 8051, 8052, 8751, 80196 등과 미국 아트멜(ATMEL)사의 AVR 계열, 미국 마이크로칩스(MICOCHIPS)사의 PIC16F64와 같은 PIC 계열 등이 있다. 8051은 인텔에서만든 8비트 마이크로 콘트롤러이다. 미국 인텔사의 8051을 보통 MCP라고도 부른다(Micro Control Processor). 8051은 하나의 칩 내에 한 개의 CPU와 롬(ROM), 램(RAM), I/O 포트, 타이머, 카운터, 시리얼 포트 및 클럭(Clock) 발생 부분을 포함하고 있어 적은 수의 외부 부품으로 동작이 가능하게 되어 있다. 8051의 내부 롬은 마스크 롬(Mask ROM)으로 되어 있다. 8031은 8051의 타입에서 내부 롬을 없앤 타입이며, 8751은 자신이 직접(프로그래머) 내부 롬을 구워 넣을 수 있는 타입을 말한다. MCS-51 계열의 CPU들 중에는 일반 CPU와 같은 기능을 하면서 전력을 더욱 적게 소비하는 CMOS(Complementary Metal Oxide Semiconductor) 형이 있다.

표기는 예를 들어, 8051, 8031, 8751의 CMOS 형일 경우 각각 80C31, 80C51, 87C51과 같이 가운데에 'C' 자를 넣어서 표시한다. In-System Programming 형일 경우 'S' 자를 넣어 89S91 등과 같이 나타낸다.



이들 외에 MCS-51 계열에는 약간의 기능을 추가시킨 다양한 CPU들이 있다. 8051 같은 경우 Keil C를 많이 사용하고, AVR 같은 경우 IAR 이나 AVR-GCC 같은 것을 많이 사용한다. 그리고 ISP 기능을 가진 8051 같은 경우 따로 롬 라이터가 필요 없다. AVR 같은 경우 자체 내에 ISP를 가지고 있기 때문에 그걸 프로그래밍해 주는 소프트웨어만 있으면 된다 즉, 둘 다 C언어 프로그램이 다르다. C 언어의 문법성은 같지만 그걸 번역해 주는 컴파일러는 틀리다는 것을 의미한다. 미국 ATMEL사에서 나온 요즘 CPU는 롬 라이터가 없어도 프로그램을 입력할 수 있어서 편리하고, 프로그램 쓰기를 하는 비용도 줄일 수 있다.

롬 라이터가 내장되어 있는 CPU는 51계열로 AT89S51, 89S52가 있고 AVR은 모두 가능하다. AVR은 RISC 머신이며, 이보다는 Z80이나 8051 68HC11 마이크로프로세서 등이 더 먼저 상용화된 마이크로프로세서로 AVR은 고급에 속한다. 일본에서는 마이크로프로세서 컴퓨터 시스템을 마이컴이라고도 부른다. ISP는 In-System Programming의 약자로 마이크로프로세서 시스템에 연결하여 프로그래밍 할 때 롬 쓰기가 가능하도록 한 인터페이스 장치라고 볼 수 있다.

2 CPU 동작과 로봇 제어

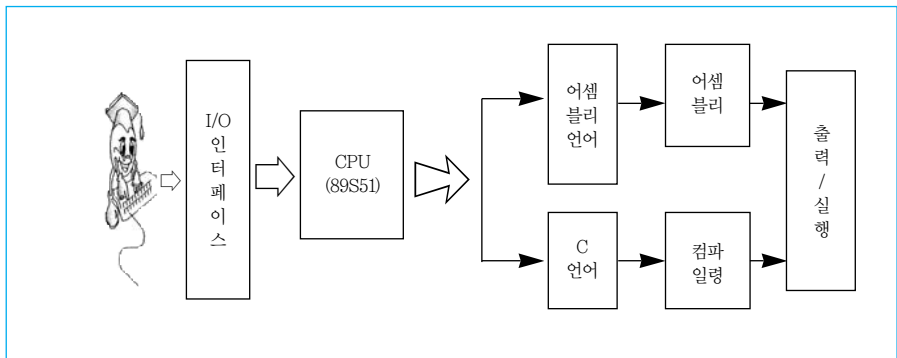
1 로봇 동작에 필요한 소프트웨어

로봇을 동작시키고 제어하기 위하여는 로봇 그 자체는 하드웨어(Hardware)적인 구성으로 이루어지기 때문에 소프트웨어(Software)가 필요하다. 소프트웨어에는 어셈블리 언어(Assembly Language), C 언어(C Language) 및 베이직 언어 (Basic Language) 등이 있다. 로봇을 움직이고 제어하기 위하여는 이

러한 컴퓨터 언어를 사용하여 작성하는 제어 또는 응용 프로그램이 주로 사용된다. 이들 컴퓨터 언어 중에서 어셈블리 언어는 처리속도가 빠르나 프로그래밍을 하는 데 다른 컴퓨터언어에 비하여 많은 시간이 걸린다. 따라서 초보자가 어셈블리 언어로 응용 프로그램을 개발하려면 많은 노력과 시간이 필요하다.

C 언어나 베이직 언어는 프로그래밍할 때 우리의 일상생활에서 사용하는 영어 또는 숫자 등을 사용하기 때문에 보다 프로그래밍이 쉽지만, 처리 속도는 어셈블리언어에 비하여 늦다. 베이직 언어는 간편하기 때문에 초보자라도 쉽게 프로그래밍할 수 있는 이점이 있다.

컴퓨터 언어 중에서 C 언어는 베이직 언어보다 기능이 많고 응용 프로그램이 많이 개발 되어 있다. 따라서 이 책에서도 C 언어를 사용하여 로봇을 제어하기 위한 기본 또는 응용 프로그램들을 다루고 있다.



〈그림Ⅳ-17〉 로봇에게 주어진 제어명령이 처리되는 과정

그러나 로봇에 내장되어 있는 CPU는 이러한 언어를 인식하지 못하기 때문에 다시 번역하여 로봇이 번역된 프로그램을 인식하도록 하여야 한다. 이 번역 프로그램을 어셈블러(Assembler) 또는 컴파일러(Compiler)라고 부른다. 어셈블러는 어셈블리 언어를 번역하는 기능을 갖고 있고, 컴파일러는 C 언어, 베이직



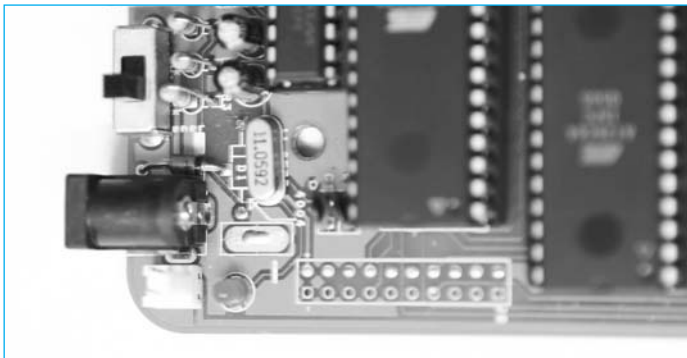
언어 등을 번역하는 기능을 갖고 있다. 로봇에게 제어명령이 주어지는 경우 소프트웨어의 처리 과정을 나타내면 <그림Ⅳ-17>과 같다.

2 로봇 동작에 필요한 하드웨어 모듈

로봇 동작과 제어에 필요한 주요 장치인 하드웨어 모듈에 관하여 알아보자. 하드웨어 모듈은 로봇을 구성하는 물리적인 구성 장치로 일정한 기능을 갖도록 관련 부품들로 구성된다.

1) 전원부

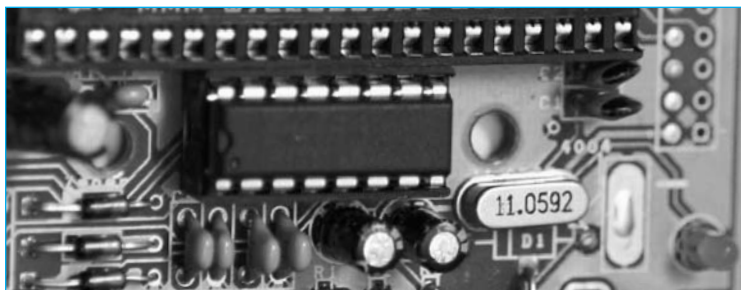
로봇에 전원을 공급하는 하드웨어 장치를 말한다. 건전지와 교류(AC : Alternative Current) 어댑터(Adaptor)를 모두 이용할 수 있도록 설계한다. 전원부는 전원 스위치, 가정의 220V 교류를 직류(DC :Direct Current)로 변환시키는 정류회로, 다시 로봇에 내장되어 있는 마이크로프로세서에 공급할 직류 +5V로 만들어주는 +5V 정전압 회로 등으로 구성된다. 건전지와 교류 어댑터를 동시에 사용하지 않는 것이 좋다.



<그림Ⅳ-18> 전원부

2) 직렬 통신부

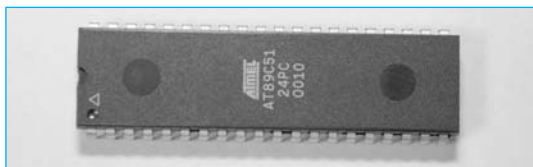
로봇과 개인용 컴퓨터(PC : Personal Computer) 또는 다른 로봇과 통신을 함으로써 데이터를 송신 또는 수신하며 로봇을 제어하고 동작시키기 위한 통신 장치이다. 직렬통신 장치는 개인용 컴퓨터에는 물론 대부분의 로봇에 설치되어 있다. 유선이나 무선으로 로봇과 통신이 가능하도록 설계되어야 한다.



〈그림Ⅳ-19〉 직렬 통신부

3) CPU 모듈

CPU(Central Processing Unit : 중앙처리장치)는 로봇제어장치의 핵심인 89S51 마이크로프로세서가 내장되어 로봇을 제어하게 된다. 주요 기능은 로봇의 동작명령이나 제어명령을 처리하고 로봇을 동작시킨다. 89S51 마이크로프로세서는 2K 바이트까지 프로그램을 저장하고 처리할 수 있다.

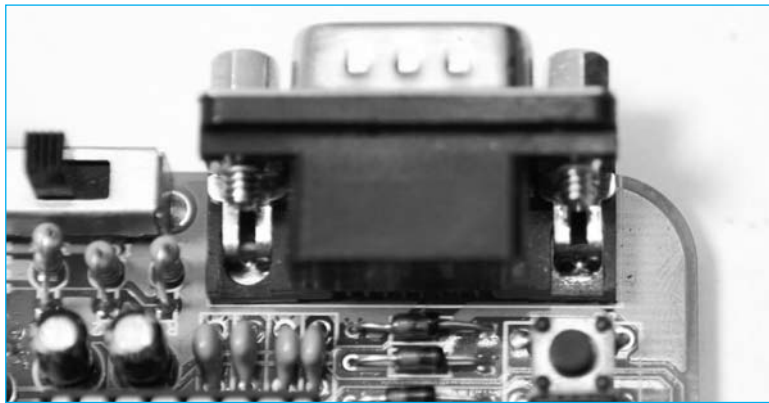


〈그림Ⅳ-20〉 CPU 모듈



4 프로그래머 소켓

CPU 모듈의 89S51 마이크로프로세서에 로봇을 제어하기 위한 프로그램을 쓰기(Write) 위한 소켓을 말한다. 대개 기본 실험장치에 설치된 소켓을 사용하는 것이 편리하다. 프로그래머 케이블은 연결이나 탈착 시 무리하게 힘을 주지 않도록 주의하여야 한다.

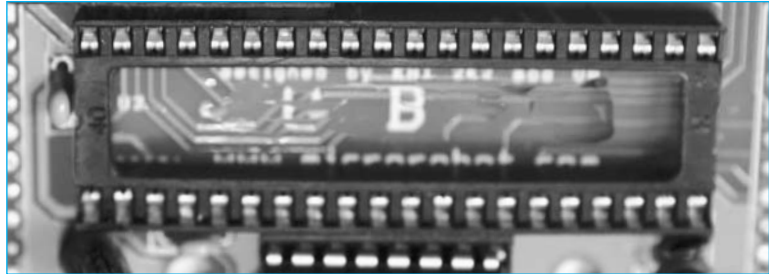


〈그림Ⅳ-21〉 프로그래머 소켓

5 인터페이스부

인터페이스부(Interface Unit)는 LED(발광 다이오드), 스위치(Switch) 및 센서(Sensor) 등과 같은 주변 입·출력 장치와 마이크로프로세서의 입출력 포트(Port)를 서로 연결하기 위하여 사용한다.

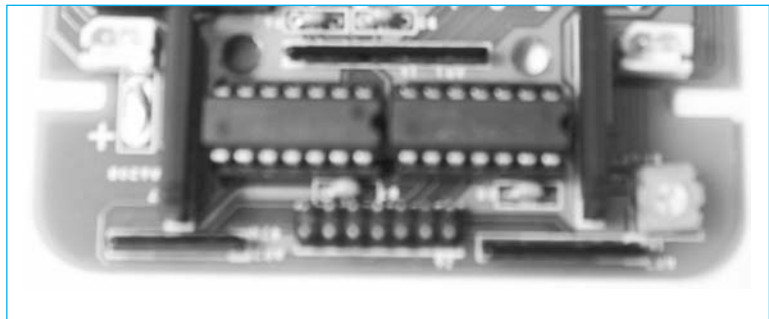
입·출력용 포트는 입력 또는 출력 데이터를 송·수신하기 위한 것으로 CPU와 입·출력용 I/O 장치(Input and Output Device)를 연결하기 위한 것이다. 일반적으로 연결을 편리하게 하기 위하여 소켓을 사용한다.



〈그림Ⅳ-22〉 인터페이스부

⑥ 모터 제어부

모터 제어를 위하여 모터 구동용 회로가 설치된 장치를 말한다. 모터는 로봇이 움직일 때 로봇의 정지 및 이동과, 이동시 할 때 이동하는 방향과 속도 등을 제어하는 경우에 사용한다.



〈그림Ⅳ-23〉 모터 제어부



06

운영 시스템 설계

1 C언어와 컴파일러

1 C 언어의 이해

C 언어는 ALGOL60이라는 프로그래밍 언어를 기원으로 하고 있다. 많은 사람들이 많은 노력 및 시간을 통해 프로그래밍에 적절한 언어를 만들어 사용하다가 1972년 전화기로 유명한 미국의 벨연구소(Bell Laboratory)에서 Brain W. Kernighan과 Demni's M. Ritchie라는 사람에 의해 만들어졌다. 이후 여러 종류의 C 언어들이 만들어지자, C 언어의 표준화 작업을 하여 ANSI C라는 C 언어의 표준을 제정하여 권고함으로써 현재의 C 언어가 생겨나게 되었다. 많은 프로그래머들이 여러 가지의 개발 언어 중에서도 C 언어를 즐겨 사용하는 데에는 몇 가지 이유가 있다.

1) C 언어의 특징

- 1 다양한 운영체제에서도 프로그래밍이 가능하여 호환성이 높다.
- 2 폭넓은 응용 프로그램을 구현할 수 있다.
- 3 시스템 프로그램과 일반 응용 프로그램을 모두 수행할 수 있다.
- 4 효율적이며 실행 파일의 실행 속도가 비교적 빠르다.
- 5 프로그램 작업이 쉽다.

2 C언어의 기본 형태

C언어는 실행 시 이를 번역할 수 있는 컴파일러가 필요한 언어로, 프로그래머가 작업한 C언어의 소스 파일을 컴파일러를 통해 목적 파일(*.obj)을 생성하고, 컴파일러가 제공하는 함수들이 있는 파일과 연결(*.lnk)하여 최종 파일인 실행 파일을 생성한다.

함수의 시작은 열린 중괄호({})를, 함수의 끝은 닫힌 중괄호(})를 사용하고, 한 문장의 끝은 세미콜론(;)을 사용한다.

1) 헤더파일 (Header File)

로봇을 움직이는 CPU에 대한 주요 기능을 미리 선언하여 프로그램을 쉽게 할 수 있도록 도와주는 파일이다(헤더파일은 컴파일러 판매 회사 또는 마이크로프로세서 제조회사에서 제공되며, 사용하는 마이크로프로세서에 맞는 헤더 파일을 사용하여야 한다.).

2) 주함수

이 곳에서부터 프로그램이 시작되어진다. C언어에서는 main() 함수가 항상 있어야 한다.

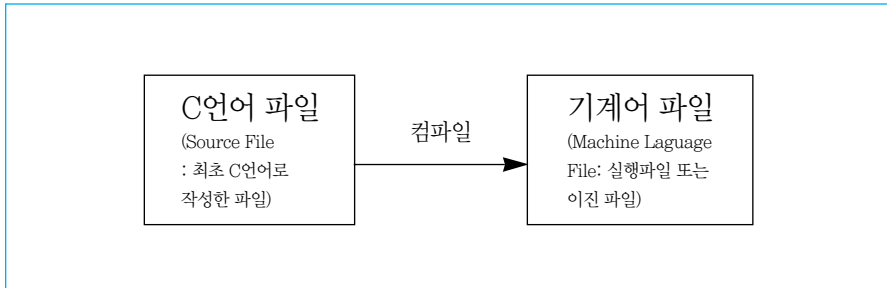
3 컴파일과 컴파일러의 이해

1) 컴파일(Compile)

프로그램을 작성한 소스 파일을 컴퓨터가 알아볼 수 있도록 기계어로 번역하여 이진(0 또는 1) 형태의 실행 파일을 만드는 것이다. 최초로 C언어로 작성한 파일을 소스 파일(Source File)이라고 하며, 고급언어 파일이라고도 부른다. 이렇게 최초로 작성한 C언어 파일을 컴파일하여 최종적으로 로봇 내부에 있는 컴퓨터(마이크로프로세서 시스템)가 인식할 수 있는 실행 파일로 변환하게 되



면, 로봇은 이 실행 파일의 내용에 따라 필요한 작업을 실행하게 된다.



〈그림Ⅳ-24〉 C 언어의 컴파일

2) 컴파일러(Compiler)

소스 파일을 기계어로 바꾸고 이것을 다시 이진 형태의 실행 파일을 만들어 주는 컴파일 과정을 도와주는 프로그램이다.

컴파일러는 마이크로 로봇을 제작하는 경우에는 케일 C, IAR C 등의 컴파일러를 사용한다.

4 변수와 상수

1) 변수

어떤 종류의 데이터는 프로그램이 사용되기 전에 값이 미리 정해지며, 프로그램이 실행되는 동안 그 값이 변하지 않고 유지된다. 그러한 데이터를 상수라고 부른다. 이와는 반대로 프로그램이 실행되는 동안 값이 변하거나 값이 대입되는 데이터를 변수라 한다.

2) 변수 선언

변수를 선언한다는 것은 컴퓨터 메모리의 일부분을 사용하겠다고 컴퓨터에게 알려주는 것이며, 어느 정도 크기의 메모리를 사용할 것인가 또는 데이터가

사용되는 장소의 이름을 지어주는 것이다.

5 변수의 데이터형

1) char형 데이터

char형은 데이터형 또는 문자형이라고 불리기도 하며, 숫자에 대응하는 문자로도 표현할 수 있다. 음수 부분은 제외되고 1byte의 크기를 사용하게 된다. 여기서 1byte는 8bit를 나타내며 1bit는 0 또는 1의 숫자(데이터라고도 함)를 나타낸다.

2) signed와 unsigned형 데이터

데이터의 크기들이 나타내는 범위를 음수와 양수 모두를 포함할 것인지 양수로만 표현할 것인지를 결정하도록 사용 전에 미리 지정해 놓은 예약어로서, signed는 음수, 양수를 나타내겠다는 의미이며, unsigned는 양수로만 나타내겠다는 의미이다.

3) int형 데이터

정수형(integer type) 데이터 형태는 2byte의 크기를 가지며 char형보다 훨씬 큰 수를 저장할 수 있다. 부호 있는 정수형(양수 또는 음수)은 int형이라고 하며, 부호 없는 정수형(양수)의 경우 unsigned int라고 한다.

4) 그 외의 데이터형

데이터형에는 문자형과 정수형 데이터형 외에도 long(긴 정수형), float(실수형), double(긴 실수형)형의 데이터 형태가 있다.

long형과 float형은 4byte의 크기를 가지며 double형은 8byte의 크기를 가진다. 다음은 여러 가지 데이터형을 보여주는 예이다.



```

char 'a'; // char형 변수 선언, 문자의 표현은 홑따옴표를 이용하여 표현
int x; // 정수형 변수 선언
signed int y; // 부호 있는 정수형 변수 선언, 생략 가능
unsigned int z; // 양수만 표현하는 정수형 변수 선언, 생략 불가능
long i; // 긴 정수형 변수 선언
float j; // 실수형 변수 선언
double k; // 긴 실수형 변수 선언

```

6 연산자

C는 데이터를 처리하는 풍부한 연산자를 제공한다. 값 비교, 변수 수정, 산술 계산 관계들의 논리적 결합 등에 사용할 수 있다.

우선순위	연산자	기호	연산순서
1	일차연산자	() . ->	→
2	단항연산자(증감연산자)	++ -- * & ~ ! sizeof (+ -)	←
3	곱셈, 나눗셈, 나머지	* / %	→
4	덧셈, 뺄셈	+ -	→
5	shift(비트 이동)	<< >>	→
6	이 항 연 산 자	관계	< > <= >=
7		등가	== !=
8		비트 곱	&
9		비트 배타적 합	^
10		비트 합	
11	논리 곱	&&	→
12	논리 합		→
13	삼항연산자(조건연산자)	? :	←
14	대입연산자	= += -= *= /=	←
15	순차연산자	,	→

〈표Ⅳ-4〉 연산자의 종류와 우선순위

2 순서도

어떤 데이터를 가지고 원하는 결과를 얻도록 프로그램을 작성하려면 명령의 처리 절차를 정확하게 해야 한다.

프로그램 명령의 순서를 정확히 하기 위해서 일련의 과정을 그림으로 나타낸 것이 순서도(Flowchart)이다. 즉, 실행되는 명령의 종류(입력, 조건, 반복, 출력 등)에 따라 각기 다른 모양의 기호를 써서 그려가게 되므로, 시각적으로 논리의 흐름을 알기 쉽게 하는 것이다.

1 순서도의 종류

1) 시스템 순서도

시스템 순서도에서는 작업의 흐름에 따라 데이터 파일의 생산, 병합, 검색, 오류 검사를 명시하고, 작업별로 각종 보고서의 출력 현황을 나타낸다. 또한 처리공정 순서도로서의 역할도 겸한다.

2) 개략 순서도

복잡한 프로그램에서는 처리해야 할 중요한 사항들의 개략적인 내용을 순서도로 나타내는 경우가 있다. 이것은 프로그램을 작성하기 위한 상세한 순서도의 전 작업으로서, 완벽한 프로그램을 만드는 데 효율적이며 필요한 과정이다.

3) 상세 순서도

상세 순서도는 프로그램의 정확한 작성을 위하여 명령과 일대일의 대응관계가 되도록 그리는 것이다.



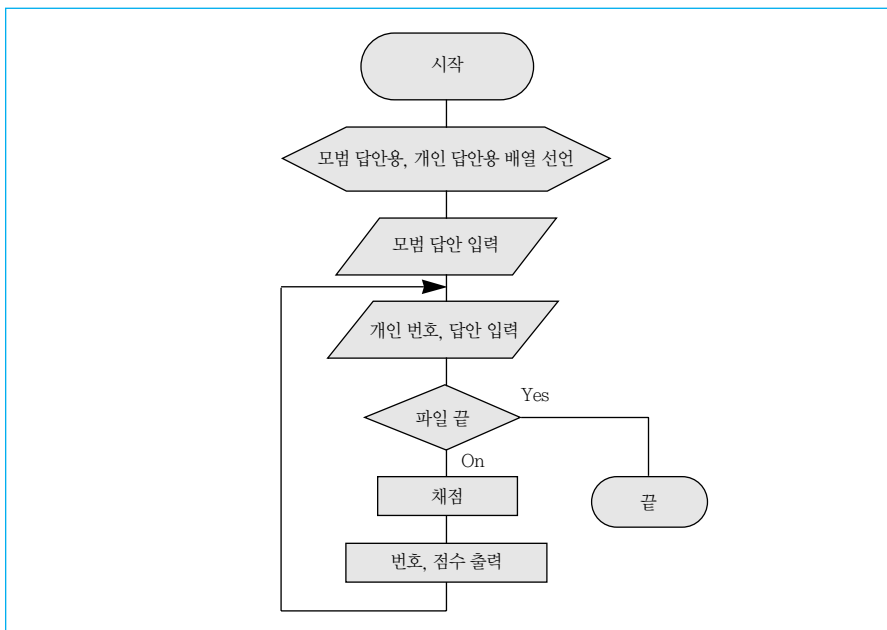
2 순서도의 기호

기호	명칭	사용 용도	기호	명칭	사용 용도
	처리	각종 연산, 데이터 이동 등의 처리		터미널	인터넷의 시작과 끝 표시
	연결자	흐름이 다른 곳과 연결되는 입출구를 나타냄		천공카드	천공카드의 입출력
	입출력	데이터의 입력과 출력		서류	서류를 매체로 하는 입출력 표시
	흐름선	처리의 흐름과 기호를 연결하는 기능		수동 입력	콘솔에 의한 입력
	준비	기억 장소, 초기값 등 작업의 준비 과정을 나타냄		카드 파일	천공카드로 구성된 파일
	미리 정의된 처리	미리 정의된 처리로 옮길 때 사용		디스플레이	결과를 모니터로 나타냄

〈표Ⅳ-5〉 순서도 기호

3 순서도의 예

〈성적 처리 업무〉



〈그림Ⅳ-25〉 성적 처리 순서도의 예

3 프로그래밍 설계

1 제어문

프로그램의 반복적 과정을 간소화하고, 조건을 찾아 동작할 수 있도록 할 수 있는 것이 제어문이다.

1) 비교 제어문

조건을 비교하여 프로그램의 흐름을 제어한다. 비교 제어문은 주어지는 조건식이 참인 경우와 거짓인 경우, 각각 다른 실행문을 실행한다.

〈비교 제어문 형식〉

```
if( 조건식 ) //조건식이 참일 경우
{
    실행문;
}
else //조건식이 거짓일 경우
{
    실행문;
}
```

〈비교 제어문 예〉

```
#include <io51.h>
#define RobotLED (*(unsigned char *)0x018000)
void main(void)
{
    unsigned char i;
    i = 100;
    if(x > 100)
    {
```



```

    RobotLED = 0x01;
}
else
{
    RobotLED = 0x02;
}
for(;;);
}

```

〈결과〉

로봇의 LED(발광 다이오드)는 0x018000 번지로 할당되어 있으므로, 로봇에 전원을 연결하고 프로그램을 실행하면 RobotLED의 값에 의하여 LED가 점등된다.

실행 결과는 다음과 같다.

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
○	○	○	○	○	○	●	○

② 반복 제어문 (for문, do~while문)

조건식을 비교하여 조건식이 참이면 중괄호 안의 실행문을 실행하는 데, 조건식이 거짓이 될때 까지 계속 반복 실행하게 된다.

① for 반복 제어문

초기값부터 조건식이 거짓일 때까지 증감연산자에 의해 반복 작업을 수행한다.

〈for 반복 제어문 형식〉

```
for(초기값 ; 조건식 ; 증감연산자
```

<for 반복 제어문 예>

```
#include <io51.h>
#define RobotLED (*(unsigned char *)0x01800)
void main(void)
{
    unsigned char x;
    unsigned int y;

    for(x = 1; x <=10; x++)
    {
        RobotLED = x;
        for(y = 0; y <30000; y++);
    }
    for(;;);
}
```

<실행 결과>

· x = 1일 때

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
○	○	○	○	○	○	○	●

· x = 2일 때

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
○	○	○	○	○	○	●	○

· x = 3일 때

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
○	○	○	○	○	○	●	●

· x = 4일 때

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
○	○	○	○	○	●	○	○



· x = 5일 때

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
○	○	○	○	○	●	○	●

· x = 6일 때

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
○	○	○	○	○	●	●	○

· x = 7일 때

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
○	○	○	○	○	●	●	●

· x = 8일 때

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
○	○	○	○	●	○	○	○

· x = 9일 때

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
○	○	○	○	●	○	○	●

· x = 10일 때

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
○	○	○	○	●	○	●	○

② do~while 반복문

먼저 실행을 한 다음 조건식을 비교하므로, 조건이 맞지 않더라도 꼭 한번은 실행이 된다.

〈do~while 반복문의 형식〉

```
do {
    실행문;
}while(조건식);
```

<do~while 반복문의 예>

```
#include <io51.h>
#define RobotLED (*(unsigned char *)0x01800)
void main(void)
{
    unsigned char x;
    unsigned int y;

    i = 1;
    do{
        RobotLED = x;
        for(y = 0; y < 30000; y++);
        x++;
    }while(x <= 10);
    for(;;);
}
```

<실행 결과>

for 반복제어문의 실행 결과와 같다.

3 선택 제어문

선택 제어문은 비교할 대상을 선택하게 함으로써 좀 더 효율적인 프로그램을 할 수 있도록 한다.



〈선택 제어문의 형식〉

```
switch( 변수 )
{
    case 상수1: 실행문; //상수1이 변수와 일치할 경우 실행될 실행문
                break;
    case 상수2: 실행문;
                break;
    case default: 실행문; //변수와 일치하는 상수가 없는 경우 실행될 실행문
                break;
}
```

〈선택 제어문의 예〉

```
#include <io51.h>
#define RobotLED (*(unsigned char *)0x01800)
void main(void)
{
    unsigned char x;
    x = 0x01;
    switch(x)
    {
        case 0x01 : RobotLED = 0x01;
        case 0x02 : RobotLED = 0x02; break;
        case 0x04 : RobotLED = 0x04; break;
        default : RobotLED = 0x08; break;
    }
    for(;;);
}
```

① break : switch case 제어문에서는 break 문장을 꼭 써주어야 한다.

〈실행 결과〉

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
○	○	○	○	○	○	●	○

2 함수와 배열

1) 함수

함수는 하나의 특정 작업을 수행하도록 독립적으로 설계된 프로그램 코드의 한 단위이다. 함수는 그 형태를 선언하여 실행한 결과를 되돌려 주거나 특정 값을 전달할 수도 있다. 또한 여러 프로그램을 작성할 경우, 중복되는 기능들을 다시 작성하지 않고 호출하여 사용할 수 있는 장점이 있다.

2) 함수의 기본 형태

```
void main(void)
```

① void

함수의 형태를 나타내는 것으로 함수의 실행 결과를 되돌려 줄 데이터의 형태를 나타낸다. 일반적인 함수의 형태를 나타낼 때 void형으로 함수 형태를 지정한다.

② main

함수의 이름을 나타낸다. 사용자가 새로 작성한 함수의 경우 함수의 이름을 직접 정하게 된다. C 언어에서는 main 함수가 항상 있어야 한다.

③ (void)

전달되는 데이터가 있는 경우, 전달되는 데이터를 받을 변수를 선언하게 된다. 전달될 데이터가 없을 때는 void 또는 아무것도 작성하지 않아도 된다.



3) 함수 사용의 예

```
#include <io51.h>
#define RobotLED (*(unsigned char *)0x01800)
void sum(unsigned char x, unsigned char y)
{
    unsigned char z;
    z = x + y;
    RobotLed = z;
}
void main(void)
{
    unsigned char x, y;
    x = 3;
    y = 5;
    sum(x, y);
    for(;;);
}
```

<실행 결과>

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
○	○	○	○	●	○	○	○

3) 배열

배열이란 같은 데이터형의 같은 변수의 이름으로 방을 여러 개 만들어 사용하는 것이다. 선언된 변수의 첫 번째 메모리 위치에서 선언된 방의 개수만큼 메모리를 연속으로 사용하는 것을 말한다.

① 배열의 선언

배열은 변수의 선언과 같이 사용될 데이터의 형태와 변수의 이름을 선언한 다음, 변수의 이름 옆에 선언할 방의 개수를 선언하는 것이다.

```
int array [5];
```

a. int

선언된 변수의 데이터 형을 나타낸다.

b. array

선언할 배열의 변수명을 나타낸다.

c. [5]

배열을 나타낸다. 배열의 선언에서 방의 개수는 '[' 와 ']' 를 사용하여 나타낸다.

② 배열의 메모리 번지

int array[5]; 라는 배열을 선언하였을 때 배열의 메모리 위치를 보면

방번호	array[0]	array[1]	array[2]	array[3]	array[4]
번지	0x00	0x02	0x04	0x06	0x08
값	1	2	3	4	5

* array[5];의 번지는 임의의 번지임.

위의 표를 보면 메모리 번지는 2씩 증가(변수 array가 int형으로 선언되었기 때문)하고, 선언된 배열의 순서는 0부터 시작되는 것을 알 수 있다.

〈배열의 선언과 초기화 예〉

```
#include <io51.h>
#define RobotLED
void main(void)
{
    int array[5] = {0x00, 0x02, 0x04, 0x08, 0x10};
    int i, j;
    for(i = 0; i < 5; i++)
```



```

{
  RobotLED = array[i];
  for(j = 0; j < 30000; j++)
    for(j = 0; j < 30000; j++)
  }
  for(;;);
}

```

〈실행 결과〉

· array[0]의 값이 출력된 경우

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
○	○	○	○	○	○	○	●

· array[1]의 값이 출력된 경우

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
○	○	○	○	○	○	●	○

· array[2]의 값이 출력된 경우

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
○	○	○	○	○	●	○	○

· array[3]의 값이 출력된 경우

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
○	○	○	○	●	○	○	○

· array[4]의 값이 출력된 경우

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
○	○	○	●	○	○	○	○

4 포인터(Pointer)

프로그램에서 사용되는 변수들은 버퍼라는 컴퓨터의 특정 기억장소에 저장

된다. C언어는 기억 장소에 대한 직접적인 참조를 가능하게 하여 기계어 수준의 데이터 처리를 가능하게 해준다. 때문에 C언어는 새로운 자료형을 제공하며, 이는 C언어의 가장 큰 특징이다.

① 포인터의 선언

포인터의 선언이란 포인터가 갖는 값은 실제적인 데이터가 아니라 데이터가 저장되어 있는 메모리의 주소를 갖는 것을 말한다. 포인터는 메모리상의 주소를 조작하므로 메모리를 동적으로 관리할 수 있다.

```
#include <stdio.h>
{
    int a; // 정수형 변수 a를 선언
    int *p; // 정수형으로 선언한 포인터 변수 p
    p = &a; // 포인터 변수 p에 변수 a의 주소값을 대입
}
```

- a. 포인터 변수의 명칭은 p이다.
- b. 포인터 변수 p의 형은 int가 아니고 p에서 참조할 데이터 형이 int이다.
- c. 포인터 값을 조작할 때는 *을 붙이지 않는다.
- d. 데이터를 참조할 경우 *를 붙인다.

② 포인터 변수와 배열

배열과 포인터는 밀접한 관계를 가지며 배열의 침찰 조작되는 모든 배열의 처리는 포인터로도 처리가 가능하다. 배열로 처리하는 것보다 포인터를 이용한 배열 조작이 더 빠르기 때문에, C언어 컴파일러는 거의 대부분의 배열 연산을 내부적으로 포인터 연산으로 변환하여 처리하고 있다.

가. 배열의 이름 그 자체는 주소 값이고 포인터 상수이다.

나. a를 배열이라고 할 때 a 자체에 다른 값을 대입할 수 없다. 왜냐하면 a는



주소 상수이기 때문이다.

③ 포인터 연산

포인터 연산에서의 포인터에 1을 증가시키는 연산은 주소값이 1만큼 증가하는 것을 의미하는 것이 아니라 포인터가 가리키는 데이터형의 바이트 수만큼 증가한다.

〈데이터 형과 포인터 연산의 관계〉

선언	처리	의미
char *p	++p;	1번지 증가
short *p	++p;	2번지 증가
int *p	++p;	DOS : 2번지 증가 UNIX : 4번지 증가
long *p	++p;	4번지 증가
float *p	++p;	4번지 증가
double	++p;	8번지 증가

〈포인터 연산의 의미〉

선언	처리	의미
*p+1	(*p)+1	P가 가리키는 곳의 내용을 1증가시킨다.
*(p+1)	*(p)+1	P+1번지의 내용을 읽어온다.
*p	*p	P의 내용을 읽어온다.
(*p)--	(*p)--	P가 가리키는 곳의 내용을 1감소시킨다.
*++p	*(++p)	P를 1번지 증가시킨 후 그 내용을 읽는다.
++*p	++(*p)	P가 가리키는 곳의 내용을 1 증가시킨다.

포인터 간의 덧셈, 뺄셈은 허용되나 곱셈과 나눗셈, 포인터 간의 덧셈, 뺄셈은 허용되지 않는다.

④ 포인터 배열

포인터 배열이란 동일한 데이터형을 갖는 포인터 변수들이 하나의 이름으로 모여 있는 형태를 말한다.

<포인터 배열의 예>

```
#include <stdio.h>
{
    char *name[5];
}
```

- a. char : 데이터형을 나타낸다.
- b. *name : 포인터 배열명을 나타낸다.
- c. [5] : 원소의 수를 나타낸다.

<배열의 포인터 예>

```
#include <io51.h>
#define RobotLED (*(unsigned char *)0x01800)
void main(void)
{
    int array[5] = {0x00, 0x02, 0x04, 0x08, 0x10};
    int x, y, *p;

    p = &array[0];
    RobotLED = *(p + 2);
    for(;;);
}
```

<실행 결과>

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
○	○	○	○	●	○	○	○



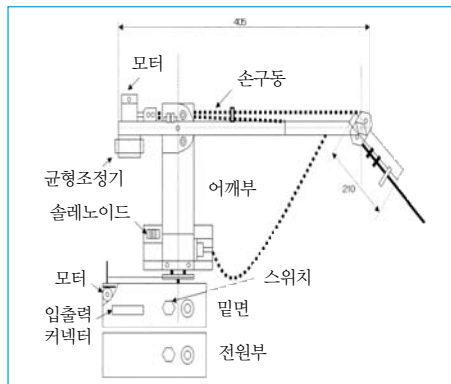
07

제작도면 작성

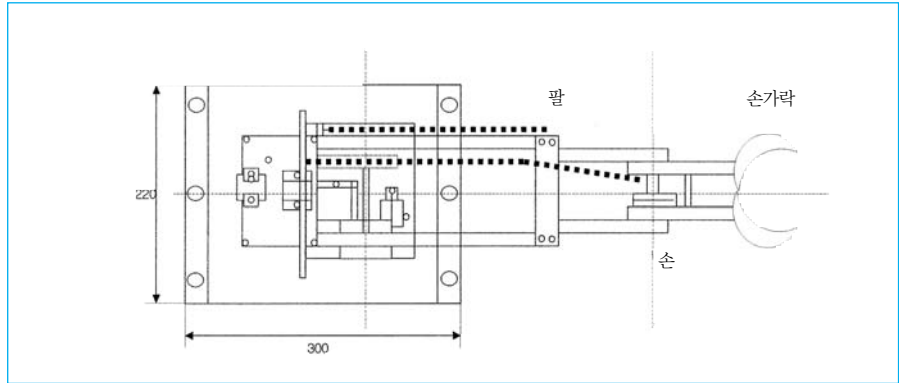
로봇 설계에 관한 제작 도면은 각 구성 부분별로 정리하여 디지털 문서화하고 설계 시, 개편 작업 시 또는 기능 개선 작업 시 활용할 수 있도록 잘 보관해 두어야 한다. 이는 다음 작업자 또는 개발자에게 작업 정보를 제공하고 기능 또는 특성을 개선하여 보다 나은 기능성과 편리성, 안정성 등을 유지하며 더 훌륭한 로봇 개발과 신제품 개발에 밑거름이 되는 계기가 될 것이다. 특히 개발된 첨단 로봇 제품은 기계, 컴퓨터의 종합 학문의 결정체이기 때문에 보안 유지에 신경을 써야 하며, 이러한 설계에 관한 제작 도면 관리에도 각별히 주의하여야 할 것이다.

1 기계 · 기구부

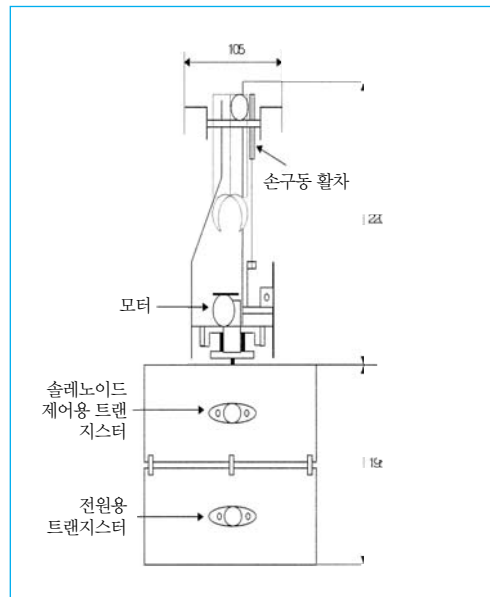
다음 <그림Ⅳ-26>, <그림Ⅳ-27> 및 <그림Ⅳ-28>은 기계 · 기구부 중 로봇 팔(Arm)에 관한 도면을 보여준다.



<그림Ⅳ-26> 로봇 팔의 외형도



〈그림 IV-27〉 로봇 팔의 평면도

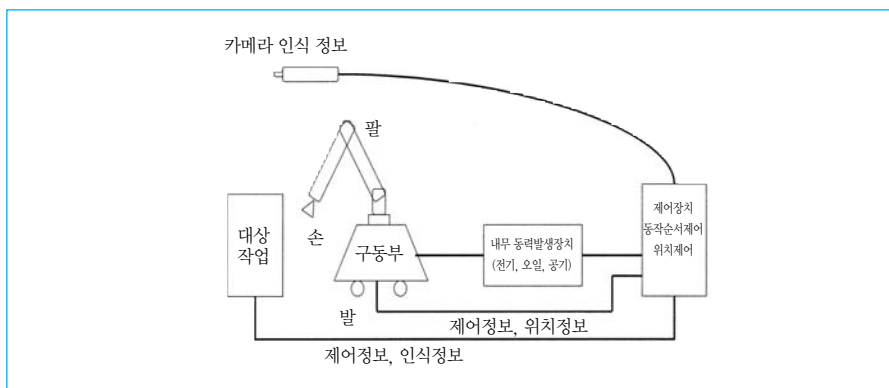


〈그림 IV-28〉 로봇 팔의 정면도



2 센서 · 제어부

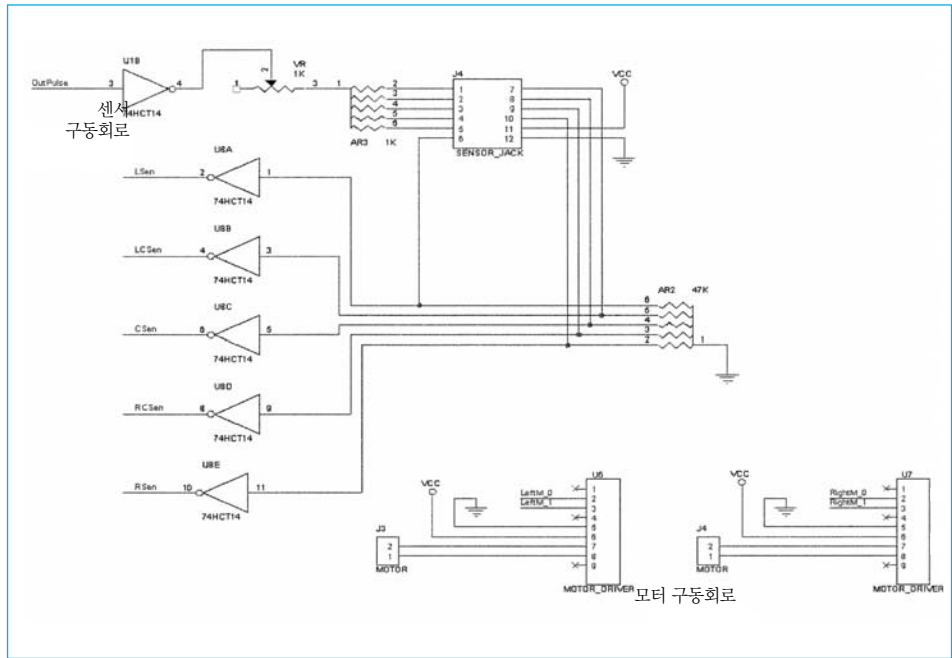
센서 · 제어부는 로봇이 인간의 5감에 해당하는 정보 또는 신호를 디지털 정보 또는 디지털 신호로 입력하고 제어하여, 로봇으로 하여금 원하는 작업을 수행하게 하고 이동하게 하는 기능을 담당하게 하는 장치이다. <그림Ⅳ-29>는 센서 · 제어부의 구성 블록도이다.



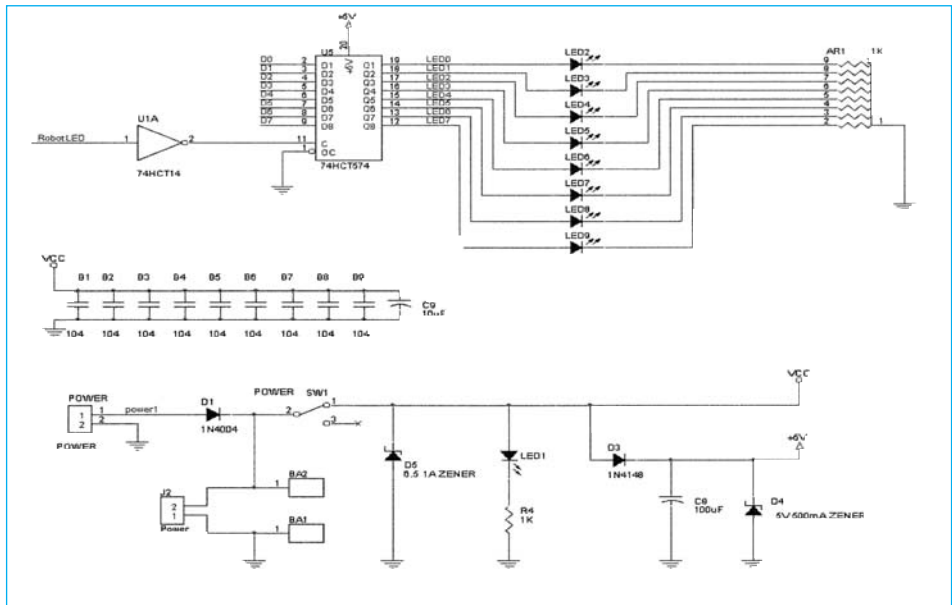
<그림Ⅳ-29> 센서 · 제어부의 구성 블록도

3 전자회로도

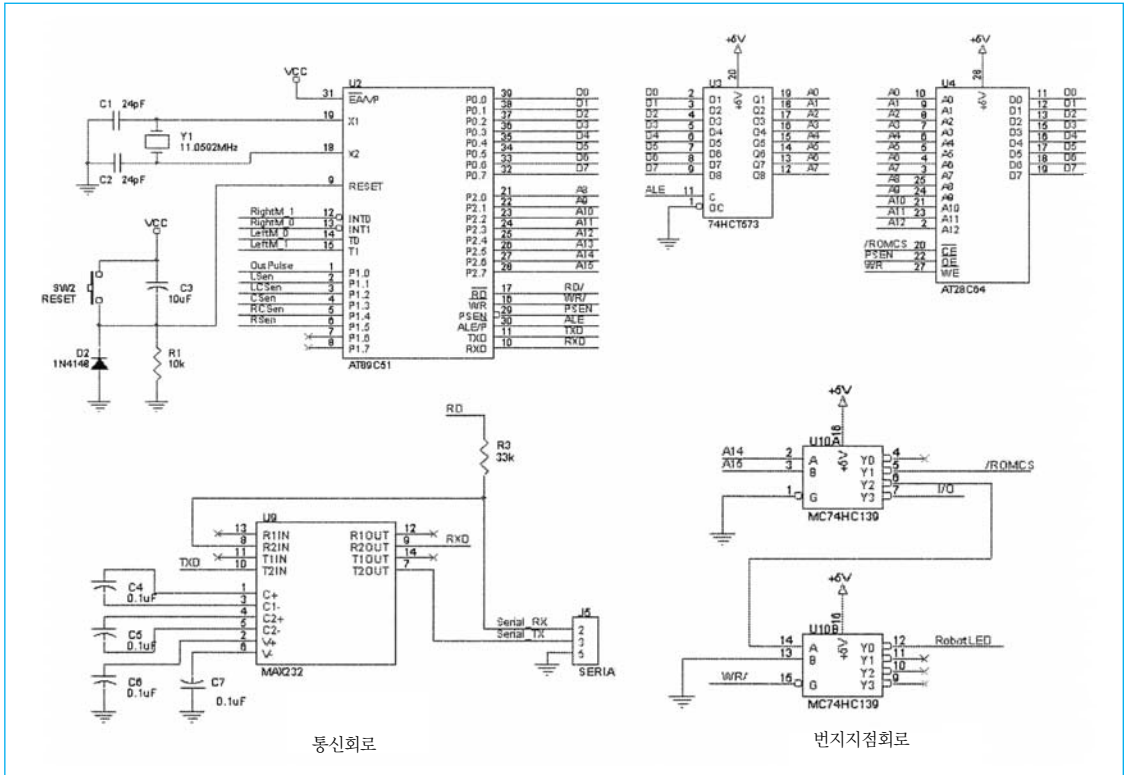
다음 <그림Ⅳ-30>, <그림Ⅳ-31> 및 <그림Ⅳ-32>은 각각 로봇 모터 구동회로, 로봇의 LED 구동회로 및 로봇의 통신회로를 나타낸다.



〈그림IV-30〉 로봇 모터 구동회로



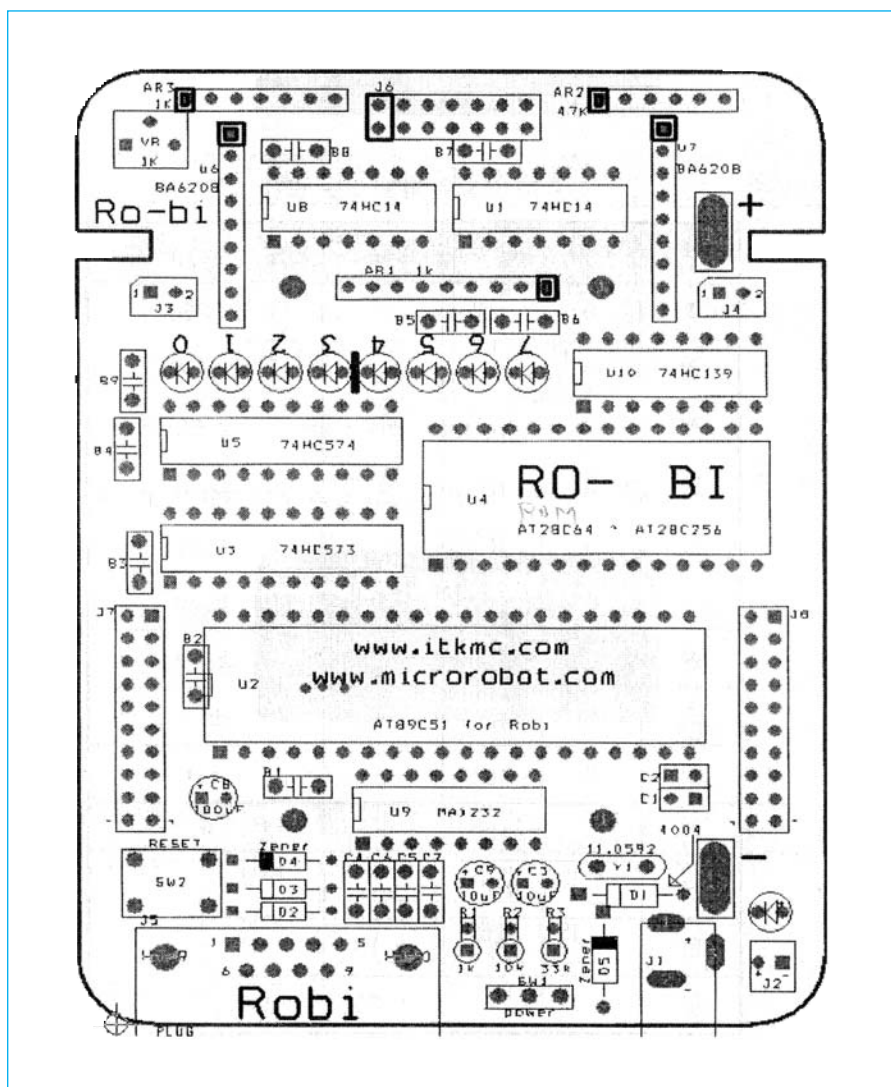
〈그림IV-31〉 로봇의 LED 구동회로



〈그림 IV-32〉 로봇의 통신회로

4 주기판(Main Board) 구성도

다음은 전체 시스템의 대부분을 포함하고 있는 로봇의 주기판 구성도의 예이다.



〈그림 IV-33〉 로봇의 주기판 구성도



단원 학습 정리



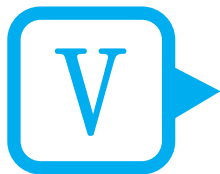
- 1** 로봇은 크게 산업용 로봇, 서비스용 로봇 및 기타 특수 목적을 수행하는 기능을 갖는 로봇 등으로 분류할 수 있다.
- 2** 로봇의 기능 및 동작 설계 특성으로는, 기계적 또는 물리적 구성품의 프로그램에 의한 기능의 유연성, 비접촉화 및 가동부 전자화의 고신뢰성, 메카트로닉스의 소형 경량화 및 고정밀화와 고속화등을 들 수 있다.
- 3** 외부로부터 정보를 얻기 위하여 사용하는 하드웨어, 소프트웨어 및 네트워크 및 연결 장치 등을 총칭하여 인터페이스라고 부른다.
- 4** 로봇에 관한 설계와 제작이 완료되면, 설계에 필요한 각종 자료와 로봇의 기능 및 특성에 관한 자료 및 제작에 필요한 자료를 문서화한다.
- 5** 기계기구 시스템 설계 고려사항은 기구 방식, 기본 조건의 설정, 운전 조건의 설정, 부하 조건의 설정, 부하 계산 및 사용기기의 크기 선정 등을 들 수 있다.
- 6** 로봇의 센서는 거리 측정 및 접촉 여부, 방향성, 변위 등을 로봇에게 전달할 수 있는 기능은 물론, 특히 비전 센서와 같이 많은 정보를 시스템에 전달하는 특성을 지니고 있다.
- 7** 모터를 제어하는 기능을 갖는 제어부는 센서로부터 얻은 제어 정보를 통해 동작되고 있는 모터의 동작 환경에 대한 정보를 입력하여 모터의 속도 및 이동 방향 등을 제어한다.

단원 종합 문제

- ▶ **1** 산업용 로봇과 서비스용 로봇의 주요 차이점은 무엇인가?
- ▶ **2** 특수 목적용 로봇에는 어떤 종류의 로봇들이 있나?
- ▶ **3** 로봇 설계 및 개발에 필요한 기술들은 무엇이라고 생각하는가?
- ▶ **4** 로봇의 일반적인 기계 · 센서부 설계 특성은 무엇인가?
- ▶ **5** 로봇의 일반적인 제어부 설계 특성은 무엇인가?
- ▶ **6** 이동용 로봇 등의 인터페이스 설계 특성은 무엇인가?
- ▶ **7** 로봇 설계의 요구에서부터 설계 및 제작과 문서화에 이르기까지의 과정을 간략히 설명하라.
- ▶ **8** 산업용 로봇 기계 장치의 네 부분으로 이루어진 기본 구조는 무엇 무엇이라고 생각하는가?
- ▶ **9** 기계 · 기구 시스템 설계시 고려사항은 무엇인가?
- ▶ **10** 이동용 로봇의 설계 및 제작에 필요한 주요 부품들은 무엇인가?
- ▶ **11** 영상 프로세싱 센서 설계의 고려사항은 무엇인가?



- ▶ **12** 이동용 로봇 등에 가장 많이 사용하는 스테핑 모터의 설계시 고려 사항은 무엇인가?
- ▶ **13** 모터와 센서를 제어할 수 있는 제어부는 일반적으로 어떤 제어 기능을 갖고 있는가?
- ▶ **14** MPU와 CPU의 큰 차이점은 무엇이라고 생각하는가?
- ▶ **15** 원 칩 마이크로프로세서에 관하여 설명하라.
- ▶ **16** 이동용 로봇 등에 많이 사용되는 89C51과 89S51과의 차이점을 설명하라. 또한, ISP에 관하여 설명하라.
- ▶ **17** 로봇 구동을 위하여 컴파일러가 필요한 이유를 설명하라.
- ▶ **18** 이동용 로봇 제작에 필요한 구성품들은 무엇인가?
- ▶ **19** 이동용 로봇에서 만일 직렬 통신부가 없다면, 로봇이 구동할 때 어떤 일이 벌어질 것인가?
- ▶ **20** C언어 변수에 어떤 데이터 형들을 사용하여야 하는가?
- ▶ **21** 3가지 종류의 제어문의 차이점을 설명하라.



로봇 설계 실습 모듈

로봇의 종류에는 크게 서비스 로봇, 산업용 로봇이 있다. 이 단원에서는 기초적인 산업용 로봇을 설계해 보고 여러 종류의 로봇에 응용되는 라인트레이서를 8951을 이용해 하드웨어를 설계하고, 프로그래밍을 통해 제어 원리를 터득한다.



학습 목표

1. 로봇의 기능과 특성을 설명할 수 있다.
2. 로봇 제작 전에 로봇 설계에 관한 계획을 세울 수 있다.
3. 기초적인 산업용 로봇을 설계할 수 있다.
4. 어떤 사용 목적에 적합한 로봇을 설계하는 데 필요한 마이크로프로세서와 기계·기구부, 센서부, 모터 제어부와 인터페이스를 설계할 수 있는 능력을 기른다.
5. 설계 및 제작 과정 또는 제작 후에 자료의 문서화와 유지 관리에 관하여 이해한다.
6. 로봇을 제어하는 데 필요한 프로그래밍 방법을 이해하고, 실습을 통하여 그 응용 능력을 기른다.



01 전기 공압제어 방식에 의한 스탬핑 장치 제어 설계

1 전기 공압제어의 제어회로 설계

1 제어회로 설계

전기공압 제어회로에는 여기에서 배우는 제어회로의 설계 방법만으로는, 현장에서 사용되는 공장 자동화 설비를 제어하기는 어렵다. 왜냐하면, 공압 시스템 설계 시 제어회로뿐만 아니라, 실린더에 걸리는 부하의 종류와 부하의 중량을 알아야 하고, 부하의 중량에 따른 실린더의 직경을 선정하며, 적정값 이상의 압력을 갖는 압축공기의 생산을 위해 압축기 및 탱크 용량 등이 고려되어야 하기 때문이다. 따라서, 공압 시스템이 원활히 작동하기 위해서는 이러한 모든 제한 요소들이 고려되어 설계되어야 한다.

다음은 공압 시스템의 설계 순서를 요약한 것이다.

단 계	내 용
1	가공품 또는 반송품의 형상과 동작을 검토하여 장치의 구조와 설계 사양을 작성한다.
2	장치의 구조와 설계 사양을 기초로 변위단계선도를 작성한다.
3	장치의 구조와 설계 사양을 기초로 액추에이터에 걸리는 부하를 계산한다.
4	장치의 구조와 변위단계선도를 고려하여 공압 장치의 회로도도를 작성한다.
5	부하를 검토하여 실린더 직경을 구한다.
6	밸브의 크기 및 종류, 공압 필터 등 실린더 제어기기를 선정한다.
7	공기 압축기 및 배관 계획을 선정한다.
8	제어회로를 작성한다.

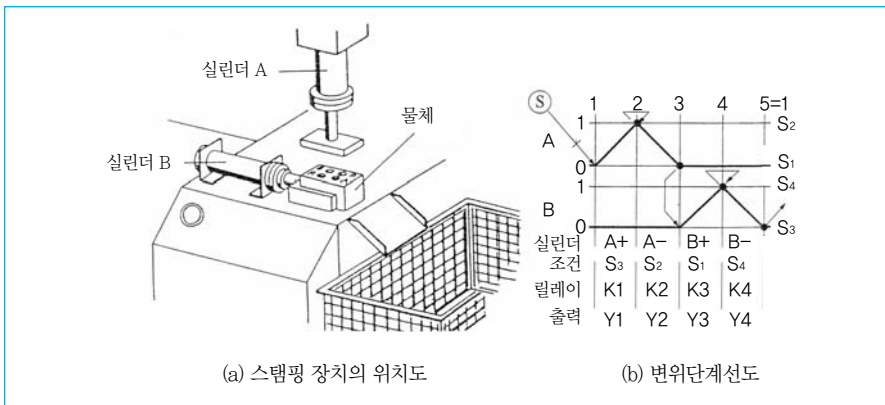
부하의 종류에는 실린더와 부하가 붙어 동작하는 관성부하와 떨어져 동작하는 저항부하가 있다.



여기서는 제어회로의 설계만을 다룰 것이므로, 그에 관련된 단계인 2단계, 4 단계, 8단계만을 알아본다.

1) 스텝퍼(stepper) 방식에 의한 회로 설계

다음 <그림V-1>과 같은 스탬핑 장치를 이용하여 물체에 문자를 새기려 한다. 물체는 수동으로 삽입하고 실린더 A를 이용하여 작업을 수행하게 된다. 작업이 끝나면 실린더 B가 물체를 밀어내게 된다.



<그림V-1> 스탬핑 장치

① 양 솔레노이드 밸브에 의한 회로 설계

이전 단계가 완료되어야만 다음 단계가 작동되도록 하는 회로 설계 방법으로 다음 단계에 따른다.

a. 변위단계선도 작성

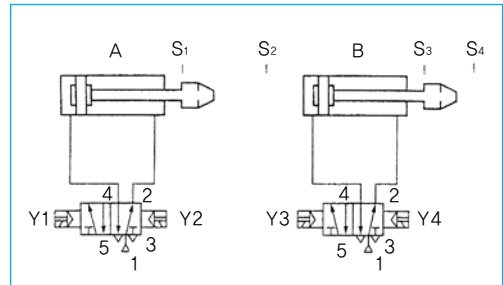
주어진 작업을 분석하여 실린더의 동작을 변위단계선도로 나타낸다. 여기서는 <그림V-1>의 (b)에 나타내었는데, 위의 변위단계선도에서 A+, A-는 각각 실린더 A의 전진, 후진을 나타내며, 화살표는 전 단계에서 실린더의 동작이 완료되어 도달 센서, 즉 리밋 스위치에 이르렀을 때 다음 단계의 동작이 이루어짐을 나타낸다.

b. 공압 회로도 작성

실린더와 전자 밸브 및 리밋 스위치의 위치를 정하고 공압 회로도를 그린다.

c. 제어회로 설계

아래의 공식에 따라 제어회로를 설계하고, 변위단계선도의 출력을 고려하여 솔레노이드 작동 부분 즉, 출력 부분을 첨가한다.



〈그림V-2〉 공압 회로도

〈변위단계선도에서 각 단계의 릴레이가 ON되는 조건의 공식〉

㉠ 첫 단계의 릴레이가 ON되는 조건식

$$K_1 = [(start \cdot 조건) \cdot K_{last} + K_1] \cdot \overline{K_2}$$

㉡ 첫 단계와 마지막 단계를 제외한 일반 릴레이가 ON되는 조건식

$$K_n = [(조건) \cdot K_{n-1} + K_n] \cdot \overline{K_{n+1}}$$

㉢ 마지막 단계의 최종 릴레이가 ON되는 조건식

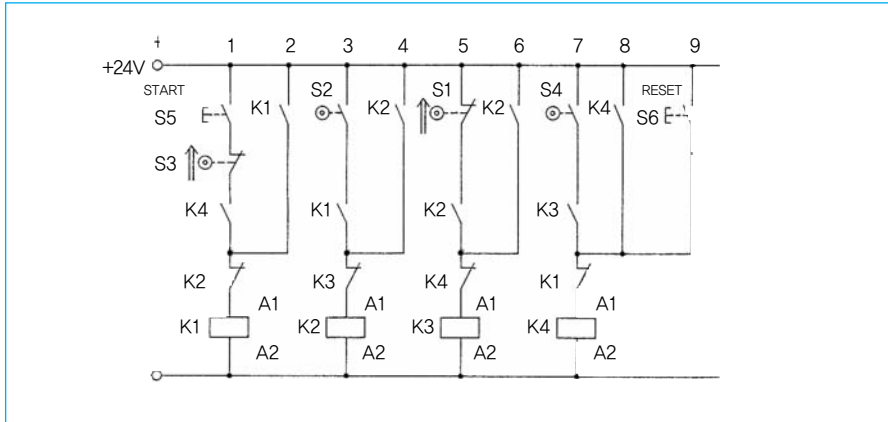
$$K_{last} = [(조건) \cdot K_{last-1} + K_{last} + Reset] \cdot \overline{K_1}$$

여기에서 start는 시작 푸시버튼 스위치를, K_n 은 a점점의 릴레이를, $\overline{K_n}$ 은 b점점의 릴레이를 나타낸다. 또, “·”는 직렬 연결을 “+”는 병렬 연결을 나타낸다. 공식 중에서 조건에 해당되는 것은 바로 전 단계의 도달 센서를 말하는데, 첫 단계의 릴레이가 ON되는 조건식의 조건은 마지막 단계의 도달 센서를 의미한다.

〈그림V-2〉의 (b)의 변위단계선도는 총 4단계로, 첫 단계의 릴레이는, 일반 릴레이는 이고, 최종 릴레이는 이다. 따라서 공식을 적용시켜 보면 다음과 같다.

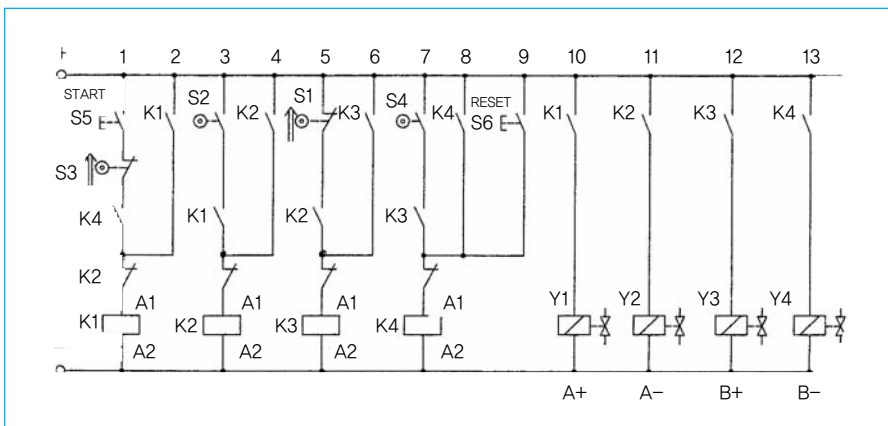


$K_n = [(start \cdot S_3) \cdot K_4 + K_1], \overline{K_2}, K_2 = [(S_2) \cdot K_1 + K_2] \cdot \overline{K_3}$
 $K_n = [(S_1) \cdot K_2 + K_3] \cdot \overline{K_4} \quad K_2 = [(S_4) \cdot K_3 + K_4 + Reset] \cdot \overline{K_1}$
 이들 공식을 기초로 제어회로를 그려보면 아래와 같다.



〈그림V-3〉 양 솔레노이드를 이용한 제어 회로도

위의 회로의 오른쪽에 솔레노이드 작동 부분을 첨가하면 아래와 같다.



〈그림V-4〉 완성된 제어 회로도

스테퍼 방식의 회로 설계는 양 솔레노이드 밸브 이용 시 다음 단계의 릴레이

가 ON되면, 바로 이전 단계의 릴레이가 OFF되어 상대 동작 금지(inter-lock) 되어 양 솔레노이드 밸브의 간섭현상이 없어진다. 또한, 리셋(reset) 스위치를 눌러서 K_{last} 릴레이의 접점을 ON한 후에 시동(start) 스위치로 작동이 가능하게 된다. 따라서, 오동작이 방지되어 안전한 시퀀스 작동이 이루어진다

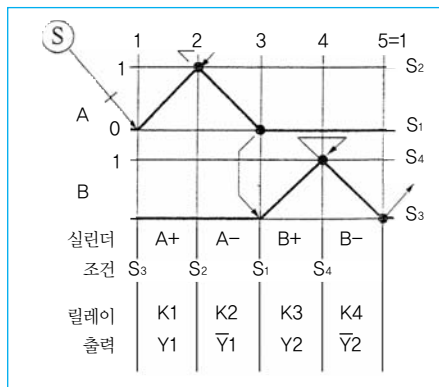
편 솔레노이드에 의한 회로 설계의 특징은 모든 릴레이가 순차적으로 자기 유지회로에 의해 ON되어 가고, 최종 릴레이가 ON되는 순간 모든 릴레이가 OFF된다.

① 편 솔레노이드 밸브에 의한 회로 설계

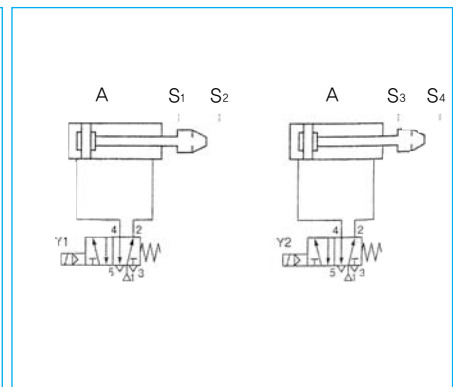
〈그림V-1〉의 스텝핑 장치를 이번에는 편 솔레노이드 밸브를 이용해 설계해 보자. 우선 편 솔레노이드 밸브를 사용하면, 방향 전환이 솔레노이드와 스프링에 의해 이루어진다. 따라서, 4단계의 변위단계선도를 갖지만 2개의 솔레노이드에 의해 동작되며, 전원이 OFF되면 스프링에 의해 복귀되는 위치로 밸브가 설치된다. 이 밸브를 사용한 회로 설계 순서는 다음과 같다.

a. 변위단계선도 작성

기본적으로 양 솔레노이드를 사용했을 때와 같다. 다만, 편 솔레노이드 밸브를 이용하기 때문에 스프링의 복원력을 이용하기 위해, 솔레노이드를 OFF시키는 동작이 필요한데 \bar{Y} 이를 로 나타내었다. 이는 솔레노이드 Y에 해당 릴레이의 b접점 스위치가 연결됨을 의미한다.



〈그림V-5〉 변위단계선도



〈그림V-6〉 공압 회로도



b. 공압 회로도 작성

실린더와 전자 밸브 및 리밋 스위치의 위치를 정하고 공압 회로도를 그린다.

c. 제어회로 설계

아래의 공식에 따라 제어회로를 설계하고, 변위단계선도의 출력을 고려하여 솔레노이드 작동 부분, 즉 출력 부분을 첨가한다.

〈변위단계선도에서 각 단계의 릴레이가 ON되는 조건의 공식〉

① 첫 단계의 릴레이가 ON되는 조건식

$$K_1 = [(start \cdot 조건) + K_1] \cdot \overline{K_{last}}$$

② 첫 단계와 마지막 단계를 제외한 일반 릴레이가 ON되는 조건식

$$K_n = [(조건) + K_n] \cdot K_{n-1}$$

③ 마지막 단계의 최종 릴레이가 ON되는 조건식

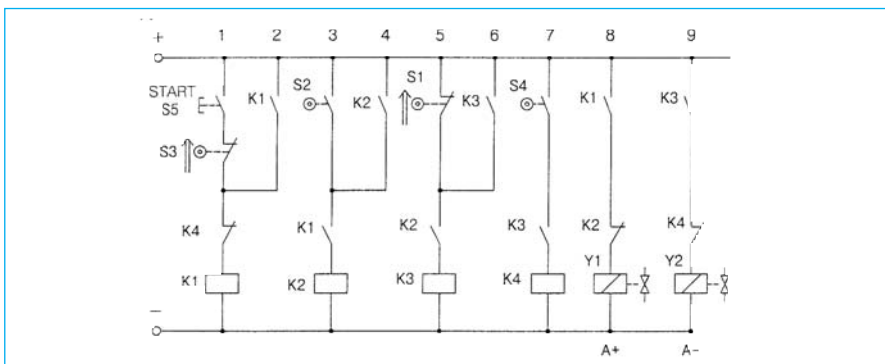
$$K_{last} = (조건) \cdot K_{last-1}$$

〈그림V-5〉의 변위단계선도에 공식을 적용시켜 보면 아래와 같다.

$$K_n = [(start \cdot S_3) + K_1] \cdot \overline{K_4}, \quad K_2 = [(S_2) + K_2] \cdot K_1$$

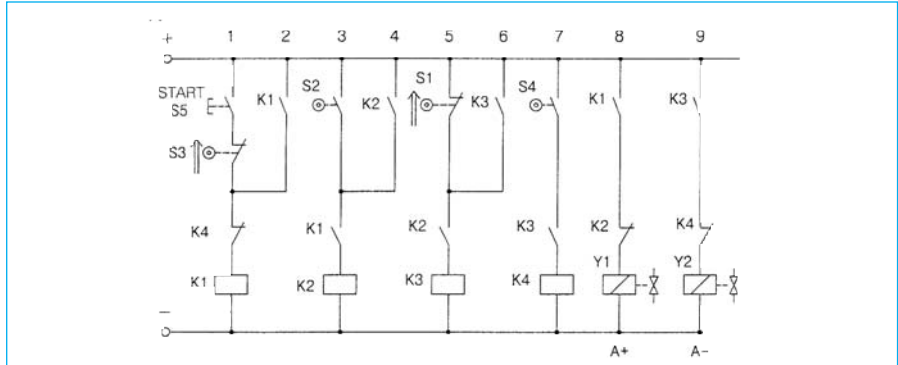
$$K_3 = [(S_1) + K_3] \cdot K_2 \quad K_4 = (S_4) \cdot K_3$$

이들 공식을 기초로 제어회로를 그려보면 아래와 같다.



〈그림V-7〉 편 솔레노이드를 이용한 제어 회로도

위의 회로의 오른쪽에 솔레노이드 작동 부분을 첨가하면 아래와 같다.



〈그림V-8〉 완성된 제어 회로도

캐스케이드 방식은 그룹 수 만큼의 릴레이가 필요하다.

2) 캐스케이드(cascade) 방식에 의한 회로 설계

제어 시스템에 신호가 중복될 경우 작동에 문제가 발생한다. 따라서 신호가 더 이상 필요 없을 경우에는 신호를 차단해 주어야 한다. 이와 같이 신호를 적절하게 차단하기 위해서 솔레노이드를 그룹화시켜 간섭현상을 방지하는 회로 설계 방법이 캐스케이드 방식으로 양 솔레노이드를 주로 사용한다.

앞의 스테퍼 방식은 현 단계의 동작시 이전 단계의 동작을 무조건 차단함으로써 양 솔레노이드의 간섭현상을 배제하였다. 그러나 캐스케이드 방식에서는 각 그룹별로 순차적으로 동작되므로 같은 그룹 내에 여러 개의 솔레노이드가 있을 경우 오동작의 위험이 존재한다. 그러나 스테퍼 방식보다 릴레이가 적게 들어가는 장점이 있다.

여기서도 〈그림V-8〉의 스탬핑 장치를 설계해 보자.

① 변위단계선도 작성

위의 스테퍼 방식의 양 솔레노이드를 사용하는 경우와 같다. 단지 간섭을 방지하기 위하여 한 그룹 내에 같은 실린더가 들어가지 않도록 그룹나누기가 필



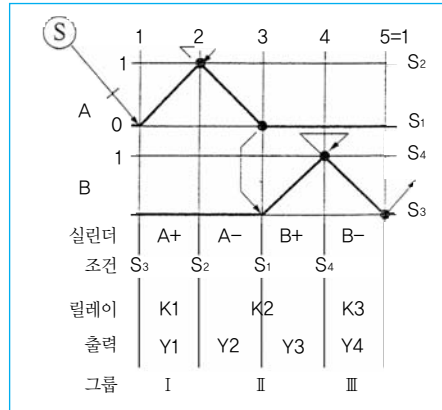
요하다. 캐스케이드 방식에서는 그룹 수와 같은 수의 릴레이만 필요하게 된다.

여기서는 아래와 같이 그룹나누기를 한다.

A+ / A- B+ / B-
 I 그룹 II 그룹 III 그룹

② 공압 회로도 작성

실린더와 전자 밸브 및 리밋 스위치의 위치를 정하고 공압 회로도를 그린다. <그림V-2>



<그림V-9> 변위단계선도

③ 제어회로 설계

아래의 공식에 의거 제어회로를 설계하고, 변위단계선도의 출력을 고려하여 솔레노이드 작동 부분, 즉 출력 부분을 첨가한다.

<변위단계선도에서 각 단계의 릴레이가 ON되는 조건의 공식>

a. 첫 단계의 릴레이가 ON되는 조건식

$$K_1 = [(start \cdot 조건) \cdot K_{last} + K_1] \cdot \overline{K_2}$$

b. 첫 단계와 마지막 단계를 제외한 일반 릴레이가 ON되는 조건식

$$K_n = [(조건) \cdot K_{n-1} + K_n] \cdot \overline{K_{n+1}}$$

c. 마지막 단계의 최종 릴레이가 ON되는 조건식

$$K_{last} = [(조건) + K_{last-1} + K_{last} + Reset] \cdot \overline{K_1}$$

위 조건식은 그룹이 3개 이상의 경우에 적용하며, 2개의 그룹일 경우는 릴레이 1개로 제어회로의 설계가 가능하다. 두 개의 그룹일 때 릴레이 이 ON되는 조건식은 아래와 같다.

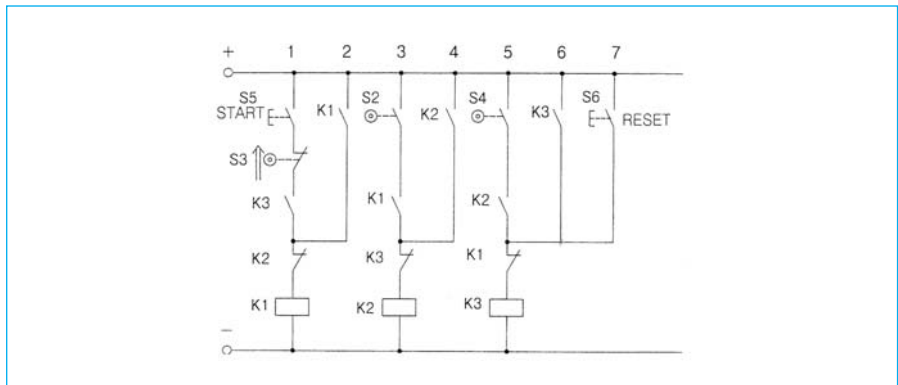
$$K_1 = [(start \cdot 조건) + K_1] \cdot (조건)$$

위의 공식을 <그림 V-9>의 변위단계선도에 적용시켜 보면 아래와 같다. 여기에서 릴레이 K_3 의 조건은 그룹 II의 마지막 도달 센서 가 된다.

$$K_1 = [(start \cdot S_3) \cdot K_3 + K_1], \overline{K_2}, K_2 = [(S_2) \cdot K_1 + K_2] \cdot \overline{K_3}$$

$$K_3 = [(S_3) \cdot K_2 \cdot K_3 + Reset] \cdot K_1$$

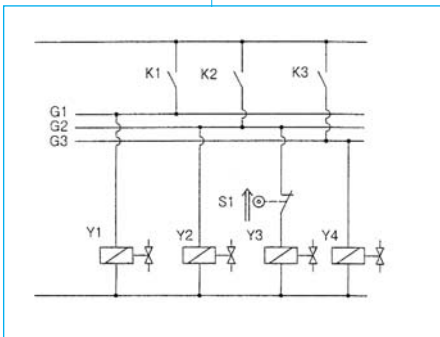
위의 공식을 이용하여 제어회로를 그려보면 아래의 그림과 같다.



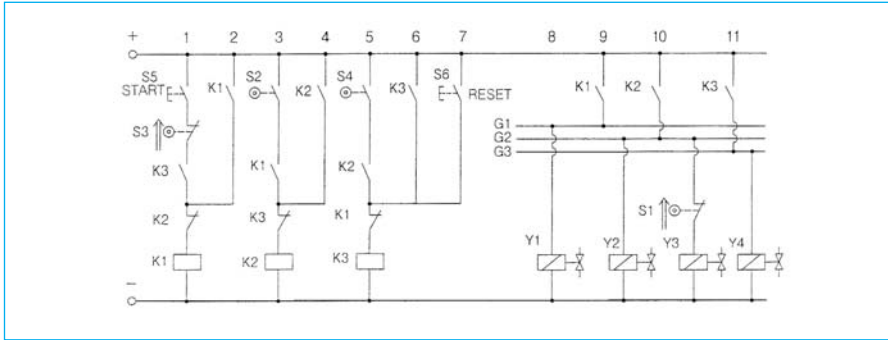
<그림V-10> 캐스케이드 방식에 의한 제어회로도

캐스케이드 방식은 솔레노이드 작동 회로의 작성이 스테퍼 방식보다 까다롭다. 일단, 제어회로도의 우측에 각각의 그룹라인을 전원선과 평행하게 긋고, 각 그룹의 릴레이 접점을 +전원선에서 해당 그룹라인에 연결한다. 또한 각각의 그룹라인에 소속된 솔레노이드를 그룹라인과 -전원선에 연결한다.

이때 주의할 점은 두 번째 그룹에는 솔레노이드 Y_2 와 Y_3 가 속하는데, 그룹 내 두 번째 단계의 솔레노이드인 Y_3 에 그룹 내 전 단계의 도달 센서인 S_1 을 직렬로 연결해야 한다.



<그림V-11> 솔레노이드 작동 회로도

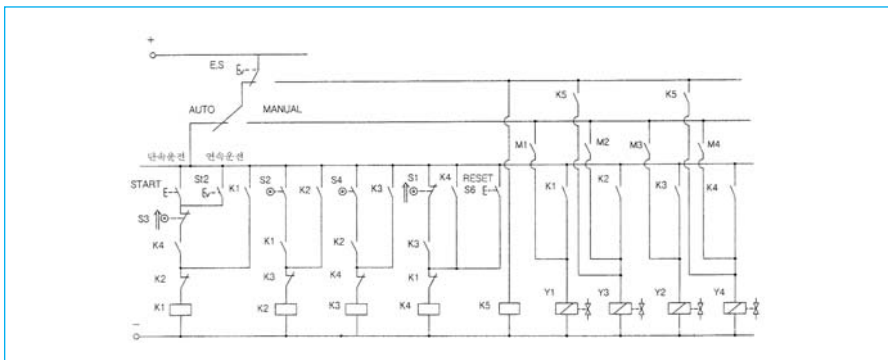


〈그림V-12〉 캐스케이드 방식의 완성된 회로도

3) 부가조건을 고려한 제어회로 설계

자동화 장치는 기본적인 제어회로와 더불어 작업자에게 좀더 쉬운 작업이 되도록 제공되는 조건이나, 또는 장치나 작업자가 예기치 않은 위험으로부터 안전하게 보호될 수 있는 조건이 필요하다. 이러한 조건들을 회로의 부가조건이라 하며, 이와 같은 조건은 통상 1사이클 동작의 기본 제어회로를 설계한 후 부여한다.

부가조건에는 장시간의 정지에 대비한 메인(main) 스위치의 ON/OFF, 자동 작업과 수동 작업의 선택, 단동과 연속 사이클의 선택, 비상 정지(emergency stop) 등의 기능이 있다.



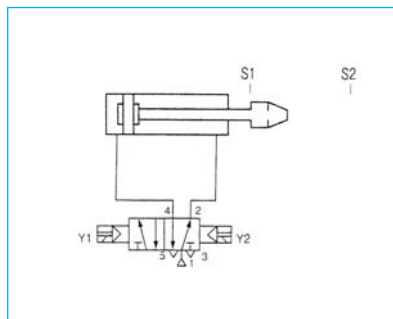
〈그림V-13〉 복합 기능의 제어회로

위의 기능은 필요에 따라 복합적으로 구성되는데, <그림V-13>에 실린더가 A+B+A-B-의 순으로 동작되는 제어회로에 단동 및 연속 사이클 선택, 비상 정지, 자동 및 수동 작업 선택 기능을 구현한 것이다.

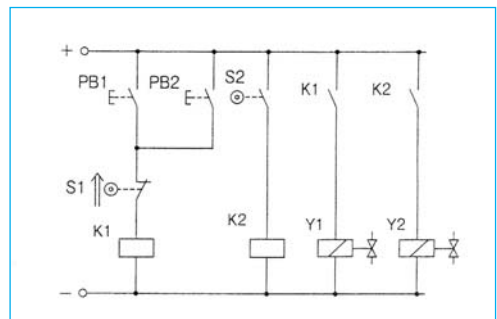
다음의 예를 통해 각각의 기능에 대해 보다 자세히 알아보자

① 단동 및 연속 사이클 선택 회로

<그림V-14>은 양 솔레노이드 밸브와 공압 복동 실린더를 구동하는 공압 회로이다.



<그림V-14> 공압 회로도



<그림V-15> 단속 및 연속 운전 회로

위의 <그림V-15>의 제어회로에서 PB1을 누르면 차례대로 K1과 Y1이 여자되어 실린더는 전진한다. 그 후 실린더의 전진이 완료되면 리밋 스위치 S2가 ON되어 차례대로 K2와 Y2가 여자되고, 실린더 전진 시 리밋 스위치 S1이 OFF 되므로, 실린더는 후진하고 1사이클이 종료된다.

만약 유지형 푸시버튼 스위치 PB2를 누르면 1사이클 종료 시 리밋 스위치 S1이 실린더에 의해 ON되므로 K1이 ON되어 연속 사이클 작업이 이루어진다.

② 비상 정지 기능 회로

공압 장치에서 비상 정지 스위치를 누르면, 장치는 모든 실린더를 초기 위치로 돌리고 정지되어야 한다. 그러나 비상 정지 해제 시 다시 시동될 수 있는 상태로



환원되도록 설계되어야 한다.

〈그림V-14〉의 회로에서 PB1을 누르면 실린더가 왕복 운동한다. 그러나 동작 중에 이상이 발생되어 비상 정지 스위치 PB2를 누르면 K3의 b점 접 스위치가 OFF되어 장치에 전원이 차단되고, 솔레노이드

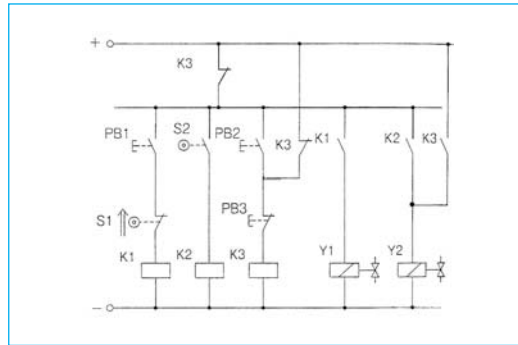
Y2가 직접 통전되어 여자되므로 실린더는 즉시 복귀된다.

다시 작업을 수행하기 위해 비상 정지 해제 스위치 PB3를 누르면 K3가 소자되어 장치가 초기 상태로 돌아간다.

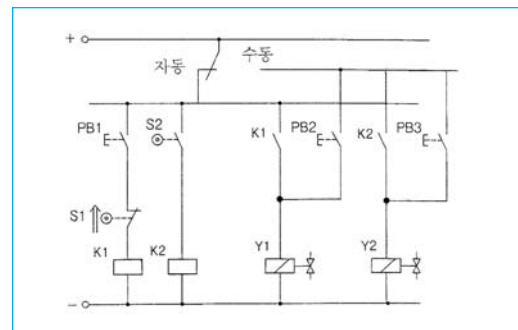
③ 자동 및 수동 작업 선택 회로

자동화된 장치에서 작업의 시작은 시동 스위치를 누르기 전에 자동 또는 수동 작업을 선택해야 한다. 수동으로 작업하는 경우에는 모든 작동 요소들은 정해진 스위치 의해 작업이 가능해야 한다. 그러므로 자동 및 수동 작업의 선택 기능이 부가된 회로는 전원선을 분리하여 각각을 구성한다. 이때 자동 및 수동 선택 스위치는 고정형 선택 스위치가 사용된다.

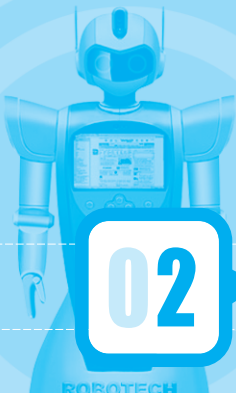
우측의 회로에서 선택 스위치가 수동인 경우 PB2는 실린더 전진용이고, PB3는 실린더 후진용이다.



〈그림V-16〉 비상 정지 회로



〈그림V-17〉 자동 및 수동 회로



PLC에 프로그램 입력 시 직접 컴퓨터에서 PLC에 프로그램을 전송하기도 한다.

1 PLC 제어의 제어회로 설계

1 제어회로 설계

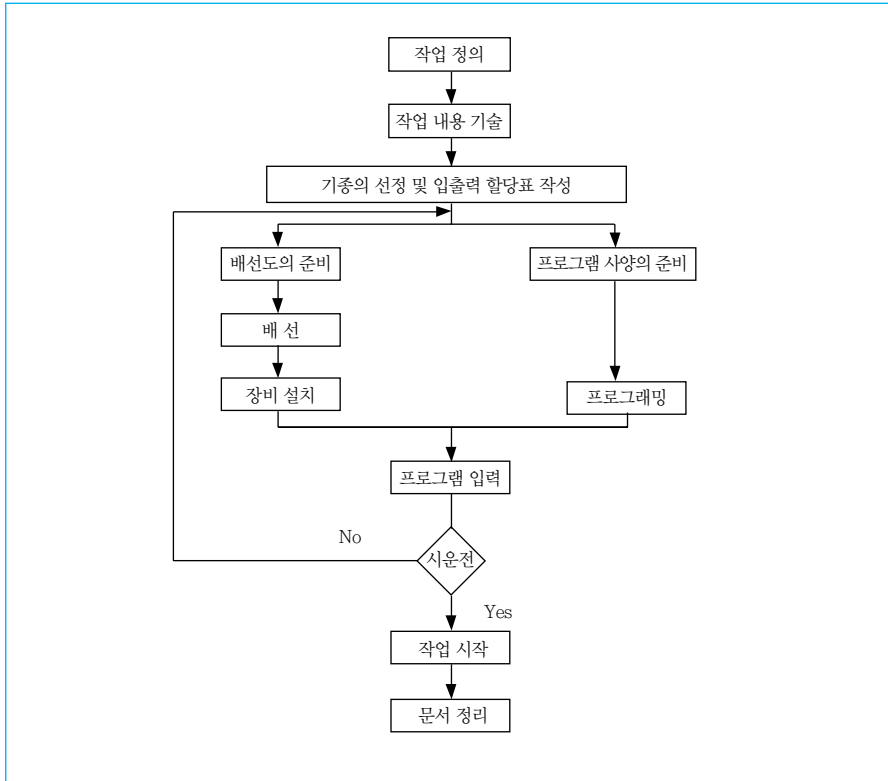
1) 제어시스템 설계 절차

제어 대상물의 자동화를 이루기 위해서는 먼저 자동화될 작업을 정의한다. 이때 기계적, 공압적, 전기적인 제반 수단으로부터 PLC의 제어 범위를 명확히 해야 한다. 작업이 정의되면 그것을 바탕으로 다음과 같은 순서로 설계가 진행된다.

다음으로 정의된 작업을 바탕으로 보다 자세한 작업 내용을 기술하는데, 이것은 전기공압 회로의 변위단계선도를 그리는 것에 해당된다. 다음으로 작업에 필요한 입·출력 점수, 메모리 용량 등을 고려하여 PLC의 기종을 선정하고, 입·출력 할당표를 작성한다.

다음에 입·출력 할당표에 의해 배선 및 장비를 설치하는 한편 프로그래밍을 한다. 프로그래밍 시에 릴레이 시퀀스 회로와 유사한 래더 회로를 사용할 경우에 전기공압 제어회로에서 소개한 공식을 이용하면 도움이 된다.

이렇게 작성된 프로그램을 PLC의 프로그래머(programmer)를 이용해 PLC에 입력한 후 시운전을 통해 잘못된 부분을 수정하고, 완벽한 프로그램이 완성되면 프로그램을 문서화(documentation)한 후 설계 절차를 마무리한다.



〈그림V-18〉 PLC 제어시스템의 설계 절차

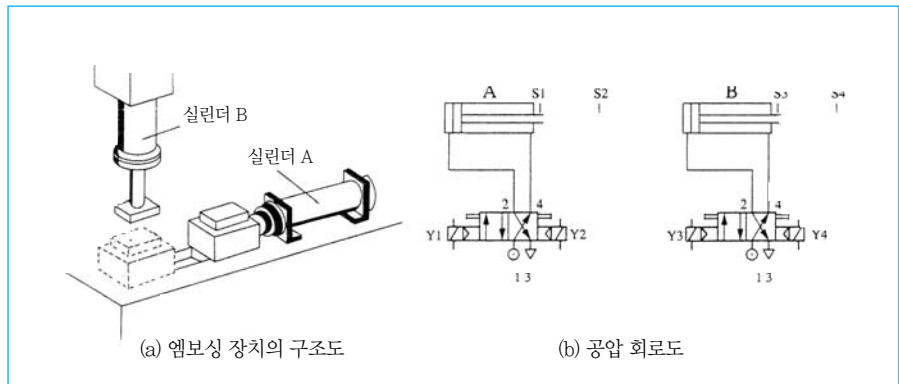
2) 설계의 실제 예

PLC는 유접점의 릴레이 제어방식을 무접점화한 제어 방식으로, 릴레이 제어 방식과 비교했을 때 신뢰성이나 확장성 등에 차이가 있지만, 릴레이를 이용한다는 면에서는 차이가 없다.

따라서, 전기공압 제어회로 설계 시 사용되었던 릴레이 제어 공식을 그대로 이용할 수 있다. 여기서는 〈그림V-19〉와 같은 엠보싱 작업기 제어회로를 설계해 본다.

① 작업 정의

2개의 실린더를 이용해 작업 대상물에 엠보싱 작업을 하는 장치로, 실린더 A가 작업 대상물을 작업 위치로 이송시켜 고정한다. 다음에 실린더 B가 엠보싱 작업을 수행한 후 후진하고, 뒤따라 실린더가 후진하여 작업을 완료한다. 이때 작업 대상물은 수동으로 공급된다.



〈그림V-19〉 엠보싱 작업기

② 작업내용 기술

a. 소재의 이송과 고정

작업 리셋 버튼과 시동 버튼이 눌러지고 작업이 시작되면, 실린더 A를 전진시킬 수 있는 솔레노이드 Y1에 신호가 존재한다.

b. 엠보싱 작업

리셋 스위치 S2가 작동되어 실린더 A의 전진 운동이 완료된 것이 확인되면, 실린더 B를 전진시킬 수 있는 솔레노이드 Y3에 신호가 존재한다.

c. 엠보싱 작업 실린더의 귀환

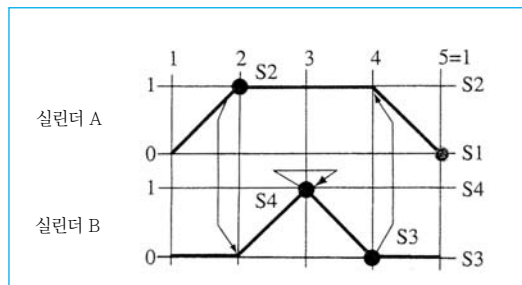
리셋 스위치 S4로 실린더 B의 전진 운동이 확인되면, 실린더 B를 후진시킬 수 있는 신호가 솔레노이드 Y4에 존재한다.



d. 소재 고정의 해제

실린더 B가 후진 운동을 완료하여 리밋 스위치 S3가 작동되면 실린더 A의 귀환 운동 신호가 솔레노이드 Y2에 존재한다.

이상의 작업 내용을 변위단계선도를 이용해 나타내면 <그림V-20>과 같다.



<그림V-20> 변위단계선도

③ 입·출력 할당표 작성

입력 측의 시작 버튼과 리밋 스위치와 출력 측의 실린더 A (Y1,Y2)와 실린더 B(Y3,Y4)를 각각 <표V-10>과 같이 할당한다.

입력		출력	
센서	PLC 입력	솔레노이드	PLC 출력
시동 버튼	I 0	솔레노이드 Y1	O1.0
리밋 스위치 S1	I 1	솔레노이드 Y2	O1.1
리밋 스위치 S1	I 2	솔레노이드 Y3	O1.2
리밋 스위치 S1	I 3	솔레노이드 Y4	O1.3
리밋 스위치 S1	I 4		
리셋 버튼	I 6		

<표V-10> 엠보싱 작업요소의 할당

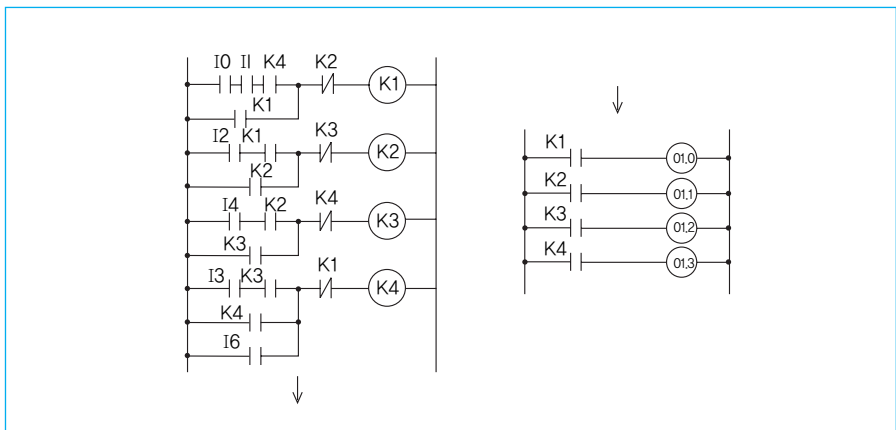
④ PLC 배선 및 프로그램 작성

이상의 과정이 완료되면 입·출력 할당표를 보고 PLC에 센서(input)와 액추에이터(output)를 연결하고, 동시에 프로그램을 작성한다. 이때 전기공압 제어 회로 설계 시 이용했던 공식을 이용하면 다음과 같은 식이 얻어진다.

$$K_1 = [(start \cdot S_3) \cdot K_4 + K_1], \bar{K}_2, K_2 = [(S_2) \cdot K_1 + K_2] \cdot \bar{K}_3$$

$$K_3 = [(S_4) \cdot K_2 + K_3] \cdot \bar{K}_4 \quad K_4 = [(S_3) \cdot K_3 + K_4 + Reset] \cdot \bar{K}_1$$

위 공식들을 입·출력 할당표를 이용해 래더 다이어그램을 그리면 <그림V-21> 과 같다.



<그림V-21> 엠보싱 작업기의 래더 다이어그램

⑤ 프로그램 입력, 시운전, 문서화 및 운전

위에서 완성된 래더 다이어그램을 PLC의 프로그램 입력장치인 프로그래머로 PLC로 입력하여 시동하고, 오류 발견 시 ④ PLC 배선 및 프로그램 작성으로 되돌아가 오류를 수정한다. 다음에 문서화를 거쳐 본격적인 운전 들어간다.



마이크로 로봇 CPU 89C51

1 89C51

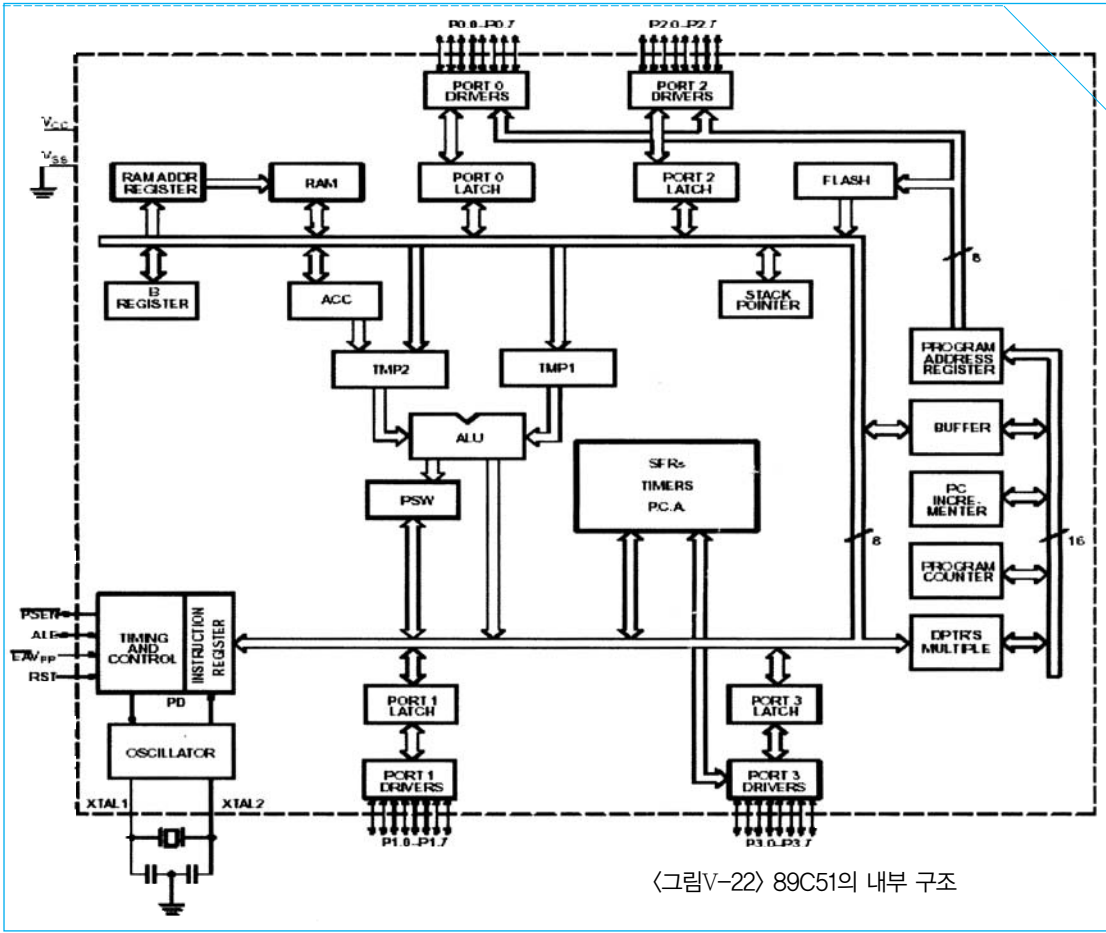
89C51이란 Intel사의 MCS-51 Family에 속하는 80XX 시리즈 중 가장 기본적인 프로세서라고 할 수 있다. 자동화 기기의 핵심 부품으로서 산업체 등에서 널리 사용되고 있는 프로세서이다. 89C51의 경우 내부에 4Kbyte의 플래시 롬이 들어 있어 초보자가 사용하기 편리하기 때문에 널리 활용되고 있다.

1 89C51의 내부 구조

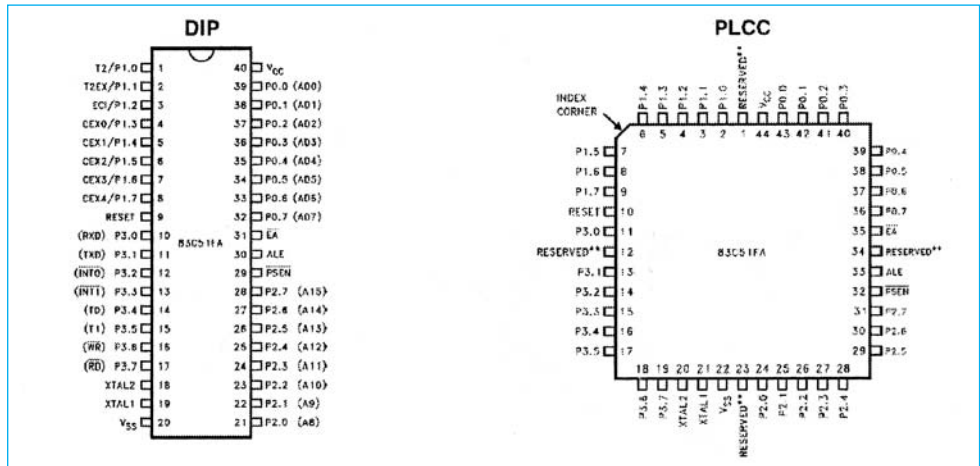
마이크로 로봇에서 사용되는 89C51 CPU는 다음과 같은 특성을 가지고 있다.

- 8비트 CPU
- 클럭 발진기
- 4바이트(32비트의 입출력 포트)
- 4개의 모드로 사용 가능한 16비트 타이머/카운터
- 1개의 시리얼 통신 포트
- 우선 순위 설정이 프로그램으로 가능한 5개의 인터럽트
- 4K바이트의 내부 프로그램 메모리
- 64K바이트의 외부 프로그램 메모리 확장 영역
- 불(Boolean) 명령 가능한 프로세서

다음의 <그림 V-22>은 89C51의 내부 구조를 나타내었다.



2 89C51 핀 기능



〈그림V-23〉 89C51의 핀 배치도



마이크로프로세서를 설계할 때 가장 중요한 것이 타이밍과 핀의 기능이다. 89C51의 각 핀의 기능은 다음과 같다.

1) 포트 0 : P0.0~P0.7

- ① 양방향 입출력 단자
- ② 각 비트는 독립적으로 입력 또는 출력으로 사용될 수 있고, 출력으로 사용 시 각 핀은 8개의 TTL을 구동할 수 있다.
- ③ 외부 메모리(ROM, RAM)를 연결할 때는 어드레스 신호와 데이터 신호가 함께 나오는 AD0~AD7 라인으로 사용된다.
- ④ 외부 메모리를 연결하지 않았을 때에는 입·출력 포트로서 사용될 수 있다.

2) 포트 1(P1.0~P1.7)

- ① 8비트 양방향 입·출력 단자이고, 출력 단자로 사용 시 4개의 LS Type의 TTL IC를 구동할 수 있다.
- ② 일반적인 입출력 단자로 사용될 수 있다.

3) 포트 2(P2.0~P2.7)

- ① 8비트 양방향 입출력 단자로서 4개의 LS Type의 TTL IC를 구동할 수 있다.
- ② 외부에 메모리 또는 I/O를 확장할 때 A8~A15까지로도 사용할 수 있다.
- ③ 내부 메모리(ROM)를 사용할 때는 일반적인 I/O로 사용할 수 있다.

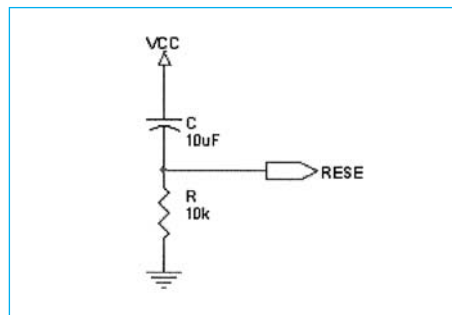
4) 포트 3(P3.0~P3.7)

- ① 8비트 양방향 입출력 단자이며 출력으로 사용 시 4개의 LS TTL을 구동할 수 있다. 이 포트는 입출력 단자 외에 다음 표와 같은 중요한 다른 기능을 가지고 있다.

P3 포트	기능
P 3.0	RDX : serial input port
P 3.1	TXD : serial output port
P 3.2	INT0 : external interrupt 0
P 3.3	INT1 : external interrupt 1
P 3.4	T0 : Timer / Counter 0 external input
P 3.5	T1 : Timer / Counter 1 external input
P 3.6	WR : external Data memory write strobe
P 3.7	RD : external Data memory read strobe

5) RESET

입력으로 액티브 HIGH 즉, RESET 입력단에 HIGH가 입력되면 마이크로프로세서는 리셋된다. 89C51을 설계할 때 인텔에서 추천하는 리셋 회로는 다음 <그림 V-24>과 같다.



<그림V-24> 리셋

6) ALE/PROG(Address Latch Enable output/Program pulse input)

- ① (ALE)는 이름 그대로 어드레스를 래치하기 위해 발생하는 신호이다. 즉, 74HC573의 11번 핀 ENABLE C단자로 입력되어 AD 라인, 즉 어드레스와 데이터가 같이 나오는 P0 포트의 신호선들을 어드레스와 데이터로 구분해주는 것이다. 인텔 계통의 일반적인 마이크로프로세서(8086, 8088,

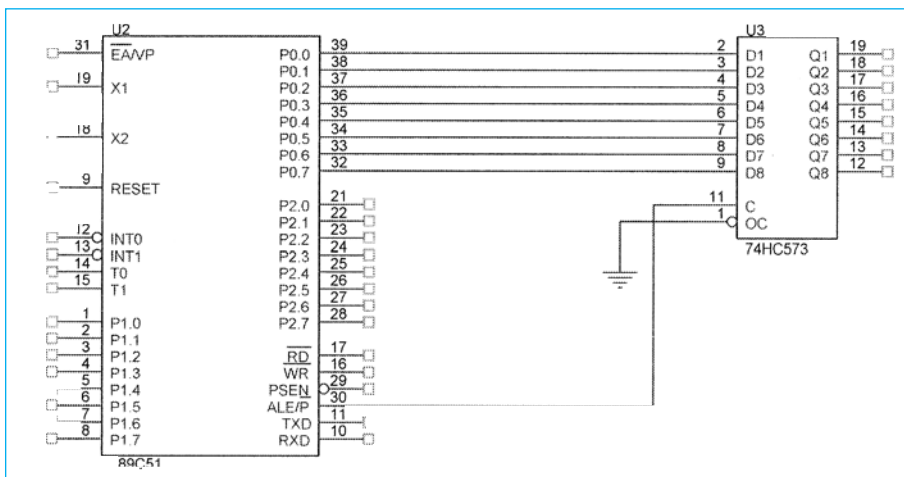
하이 임피던스(High Impedance)

이 신호의 의미는 전기적으로 끊어져 있다는 말입니다. 실제 회로도상에서는 이어져 있는 것처럼 보이지만 IC 소자의 동작상 전기적으로 끊어진 상태가 되어 있다는 것을 의미한다.



8051, 8031등)는 이런 ALE 신호를 이용하여 데이터와 어드레스 신호를 분리해 내고 있다. <그림 V-25>는 ALE 신호로 어떻게 데이터와 어드레스를 분리하는가를 보여주고 있다.

- ② (PROG)로 사용될 때는 내부 EP-ROM을 프로그램할 때이다. 즉, EP-ROM에 프로그램을 입력시키는 동안 프로그램 펄스가 입력이 된다.



<그림V-25> 어드레스 래치 회로

7) PSEN(Program strobe enable)

외부 프로그램 메모리를 읽어 들일 때 액티브 LOW신호. 각 머신 사이클에서 두 번 출력되며, ALE 하강 모서리에서 포트 0(AD0~AD7)로부터 나오는 값은 어드레스로 사용된다.

8) EA/Vpp(External access enable/ V peak-peak)

EA 입력은 사용자가 정의하게 되어 있다. 즉, 외부 ROM을 사용할 것인가 아니면 내부 ROM을 사용할 것인가를 EA 단자의 상태에 따라서 정의한다. 여기에서 설명하는 마이크로 로봇에서는 내부에 로봇의 기본적인 프로그램이 탑재되어 있으므로 +5V에 연결하여 준다. Vpp는 내부EP-ROM을 프로그램하는

액티브 로우(Active Low)

로우신호 '0' 신호가 타당한 신호라는 뜻이다.



액티브 하이(Active High)

하이신호 '1' 신호가 타당하다는 뜻이다.

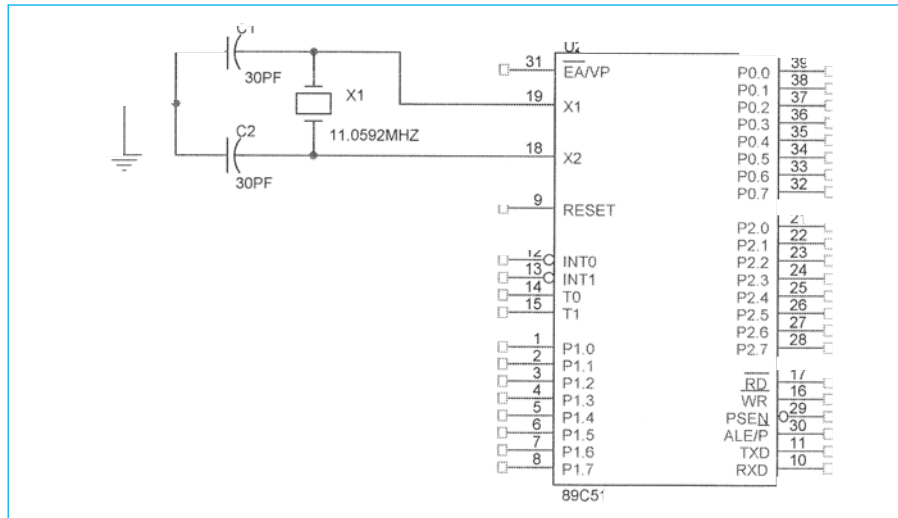


동안 전원을 입력시켜 주는 단자이다.

9) XTAL1

입력 단자이며 발진기 증폭기가 인버팅되어서 입력된다.

10) XTAL2



〈그림V-26〉 XTAL 접속 방법

초보자의 경우, 납땜을 하다보면 흔히 이 접지 단자와 전원 단자 부분을 빠뜨리는 경우가 있는데, 그럴 경우 IC는 완전히 동작하지 않는다는 점에 주의해야 된다.

11) VCC(5V)

전원 단자. +5V 전원에 연결한다.

12) VSS(GND)

접지 단자. GND에 연결한다.



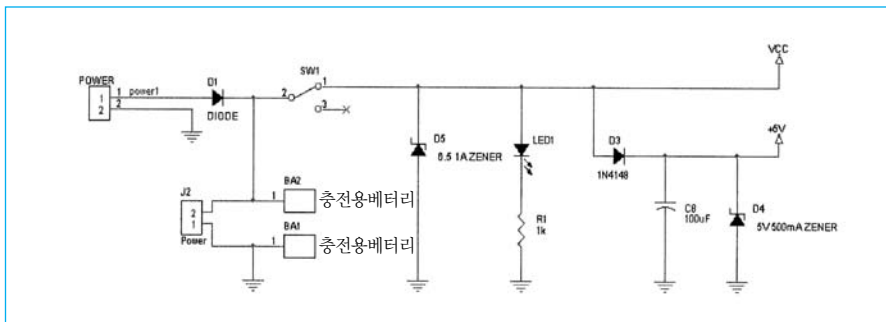
2 마이크로 로봇의 기본회로

1 전원

설계되는 마이크로 로봇의 경우, 전원은 DC 5V 어댑터 혹은 AA 건전지 4개를 이용하여 공급할 수 있다.

충전용 건전지와 어댑터를 함께 사용하면, 어댑터를 잭에 끼웠을 때는 배터리 충전을 하면서도 어댑터 전원으로 로봇을 사용하고, 어댑터를 끼우지 않았을 때는 충전용 건전지로 사용하게 된다. 단, 충전용 건전지를 사서 처음 충전할 경우에는 초기에 건전지가 어댑터 전원을 모두 사용하기 때문에 충전과 동시에 로봇을 동작시킬 수 없다.

이 경우, 로봇에 어댑터 잭을 끼워놓고 약 5시간 이내로 충전시키면 로봇을 어댑터 전원으로 사용할 수 있다.

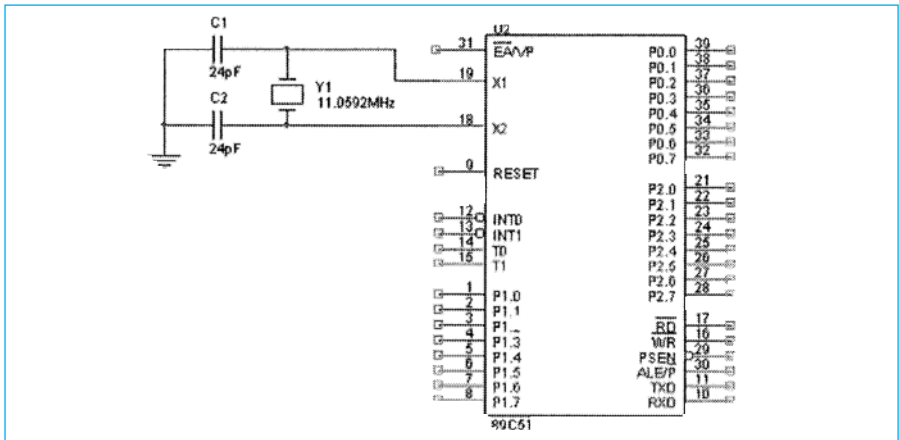


〈그림V-27〉 전원 회로부

위 그림에서 좌측의 'POWER' 라고 되어 있는 부분이 어댑터 입력 단자이다. D1은 어댑터 입력 전원이 배터리나 회로로는 들어갈 수 있으나, 배터리 입력 전원이 어댑터로 들어가는 것을 방지해 주는 다이오드이다. LED1은 전원이 투입되었음을 나타내 준다.

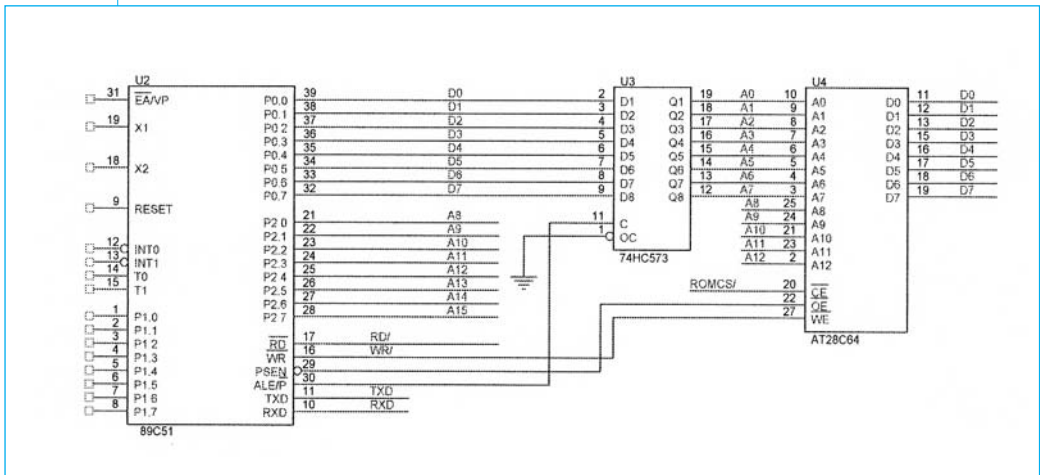
2 오실레이터

마이크로 로봇에서는 11.0592Mhz의 오실레이터를 사용하였다.



〈그림V-28〉 오실레이터

3 74HC573과 27C128(ROM) 또는 28C64(EEPROM)와의 접속



〈그림V-29〉 74HC573, 28C64(ROM) 접속

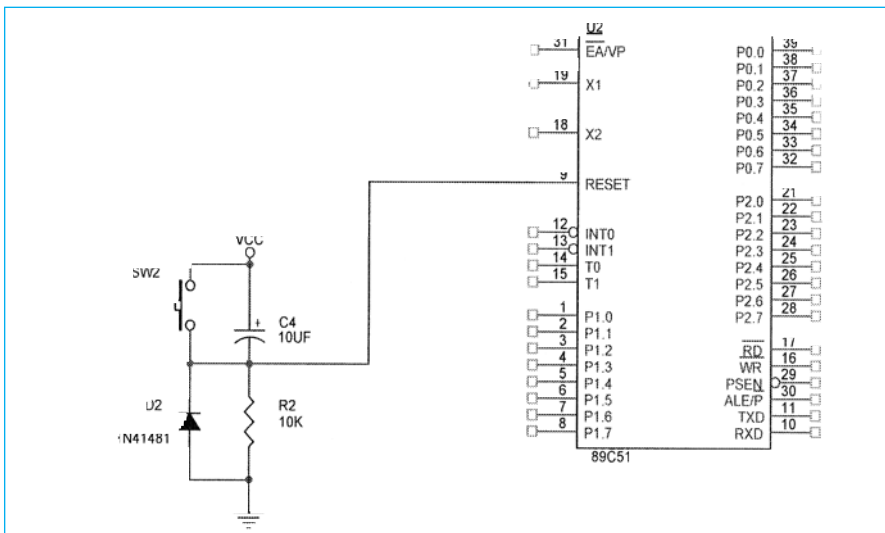


74HC573은 인텔 계통의 마이크로프로세서의 어드레스 래치용으로 가장 많이 쓰이는 TTL IC이다.

〈그림V-29〉과 같이 ALE가 HIGH일 때 어드레스가 출력된다. 따라서 ALE가 HIGH일 때를 이용하여 어드레스를 래치해야 될 것이다. 〈그림V-28〉은 74HC573을 이용하여 ALE가 HIGH가 될 때 어드레스 신호를 래치하도록 하여 28C64(EEPROM)의 어드레스 핀에 입력되도록 하였다. 데이터 라인은 AD핀과 바로 연결해 준다.

4 89C51 리셋 회로

앞 절에서 설명 리셋 회로에 리셋 스위치를 포함하여 완벽한 리셋 회로를 구성한다.



〈그림V-30〉 로봇 제어부 리셋 회로

5 모터제어 신호와 센서구동 신호

〈그림V-30〉에서와 같이 모터제어 신호 LeftM_0, LeftM_1, Right_M0, RightM1을 89C51에서 직접 출력하였다.

1) 모터제어 신호

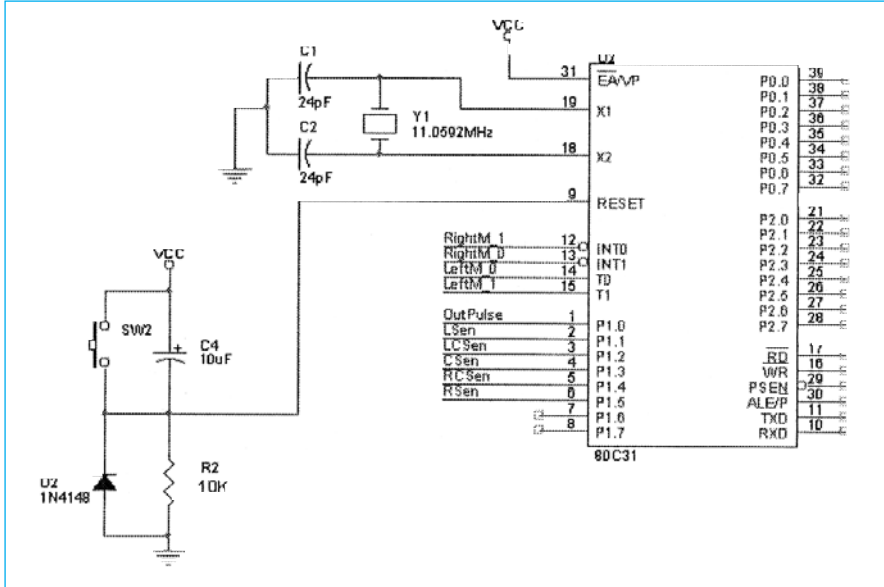
- ① 12번핀 INT0/P3.2핀(번지: 0xb2)에서는 우측 모터제어 신호인 RightM_1을 출력
 - ② 13번핀 INT1/P3.3핀(번지: 0xb3)에서는 우측 모터제어 신호인 RightM_0을 출력
 - ③ 14번핀 T0/P3.4핀(번지: 0xb4)에서는 좌측 모터제어 신호인 LeftM_0을 출력
 - ④ 15번핀 T1/P3.5핀(번지: 0xb5)에서는 좌측 모터제어 신호인 LeftM_1을 출력
- 센서 구동 신호의 경우 1번 핀 P1.0(번지:0x90)에서 '1' 신호를 출력하여 발광 센서들을 구동시킨 후, 수광 센서의 입력 값을 P1.1, P1.2, P1.3, P1.4, P1.5로 읽어 들인다.

2) 센서입력 신호

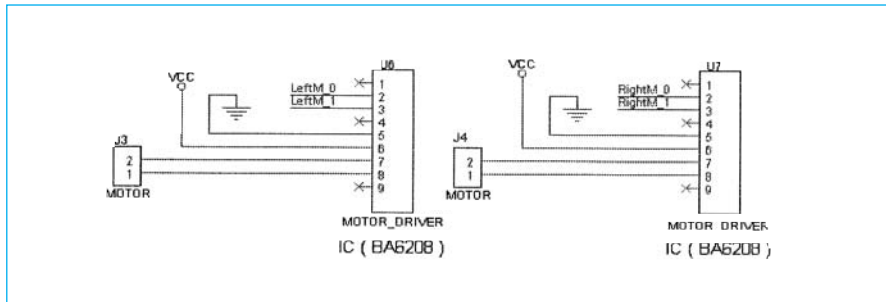
- ① 2번핀 P1.1(번지:0x91)은 왼쪽 끝에 위치한 센서입력 단자(LSen)
 - ② 3번핀 P1.2(번지:0x92)은 왼쪽 중간에 위치한 센서입력 단자(LCSen)
 - ③ 4번핀 P1.3(번지:0x93)은 중앙에 위치한 센서입력 단자(CSen)
 - ④ 5번핀 P1.4(번지:0x94)은 오른쪽 중간에 위치한 센서입력 단자(RCSen)
 - ⑤ 6번핀 P1.5(번지:0x95)은 오른쪽 끝에 위치한 센서입력 단자(RSen)
- 89C51에서 출력되는 모터제어 신호 LeftM_0, LeftM_1, RightM_0, RightM1은 모터드라이버 IC를 구동하는 신호다. 모터 드라이버 IC BA6208은



이 신호들을 입력받아 신호에 해당되는 동작을 모터에 지령하게 된다.

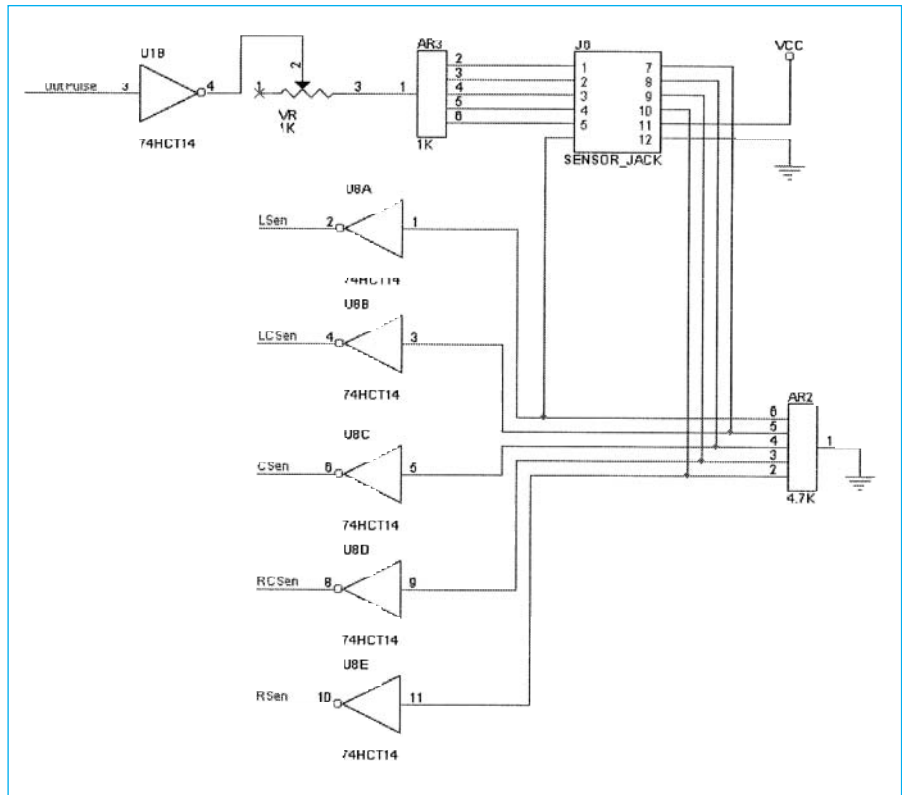


〈그림V-31〉 로봇 제어부의 모터와 센서구동 신호



〈그림V-32〉 모터 드라이브 회로부

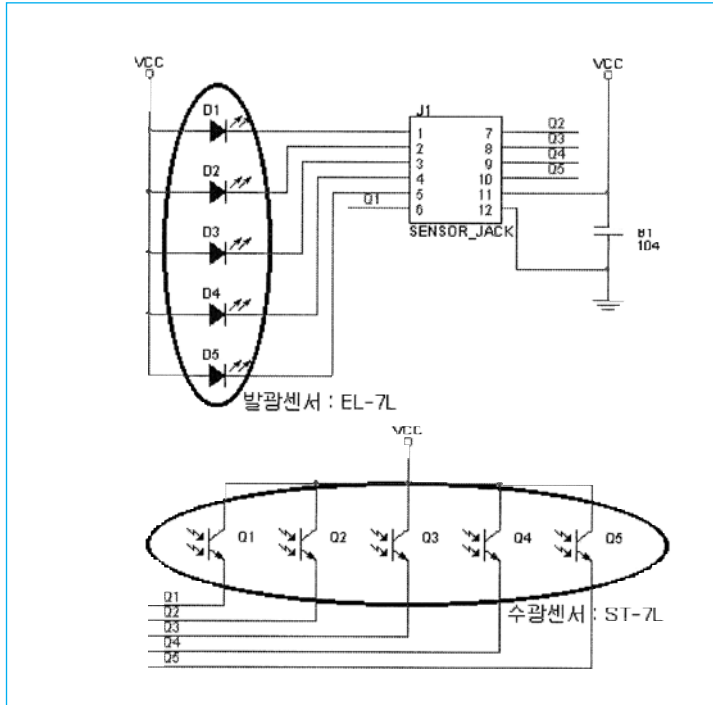
센서부의 출력은 J6 커넥터의 1번 핀부터 4번 핀까지 4개의 핀을 통하여 센서보드로 전달되고, 센서보드에서 입력받은 센싱 값은 J6 커넥터의 7번 핀부터 10번핀까지 4개의 핀을 이용하여 센서보드에서 89C51로 전달된다.



〈그림V-33〉 센서보드로 연결되는 센서 입력력 회로부

〈그림V-33〉은 센서보드에 출력하는 신호와 입력받는 신호의 체계를 나타내고 있다. 메인보드 J6와 연결되는 것은 센서보드의 J1 커넥터이다. 메인보드에서 OutPulse 신호를 '1'로 주면 센서보드의 발광부 다이오드가 ON 되고, 이때 센서 바닥이 흰색이면 빛을 반사하여 수광부가 '1'이 되고, 검정색이면 수광부가 '0'이 된다. 이 신호를 센서보드에서 메인보드로 전달한다. 또한 메인보드에서는 NOT 게이트를 사용하여 89C51에 전달된다.

따라서, LSen, LCSen, CSen, RCSen, RSen을 89C51에서 읽었을 때 '0'이면 바닥에 흰색을 감지한 것이다..



〈그림V-34〉 모터 드라이브 회로부

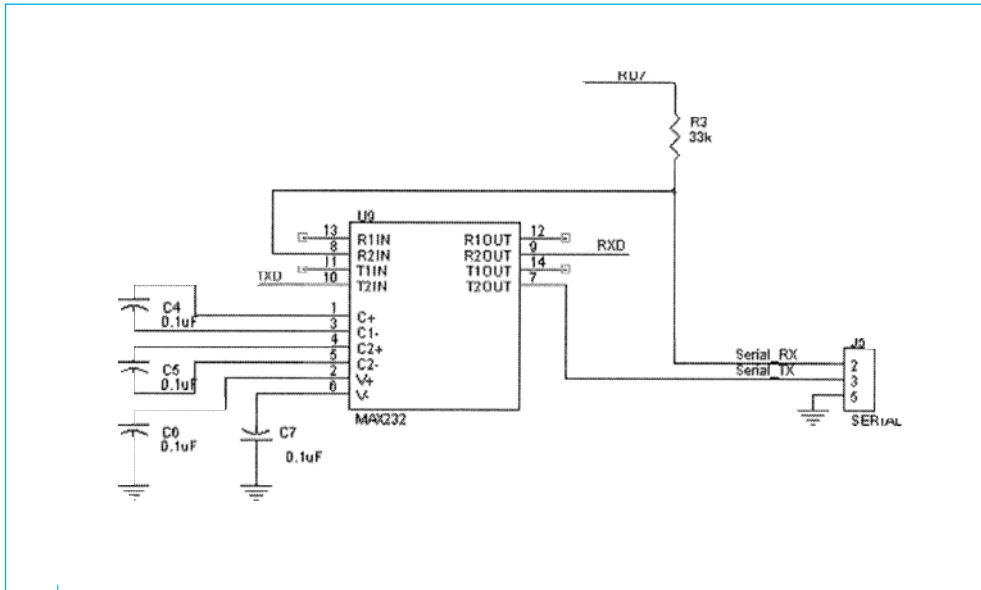
⑥ 통신 회로, 번지지정 회로, LED 구동 회로

1) 통신 회로

메인보드의 J5는 시리얼 신호들을 컴퓨터와 연결시켜 주는 부분이다.

로봇은 89C51의 11번 핀 P3.1 TXD에서 신호를 내어 통신용 IC인 MAX 232C를 거쳐 로봇의 시리얼 잭의 3번 핀 Serial_TX를 통하여 PC에 데이터를 송신하고, PC에서 데이터를 보낼 때는 시리얼 잭의 2번 핀 Serial_RX를 통해 MAX 232C를 거쳐 89C51의 10번 핀으로 데이터를 입력시킨다.

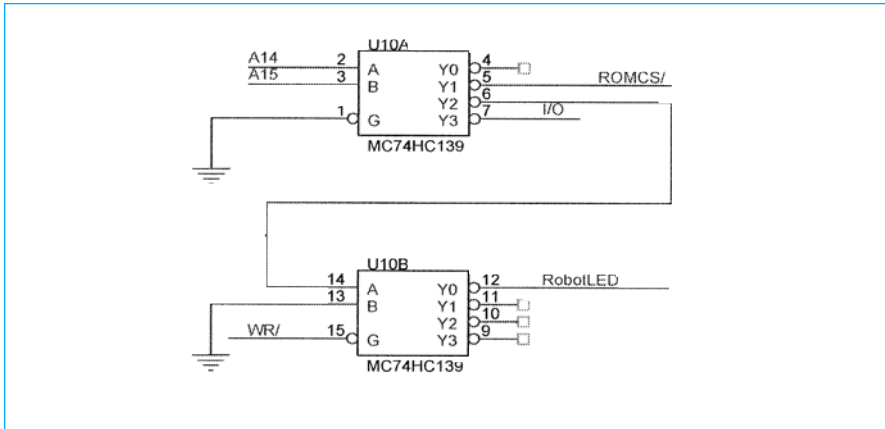
이런 과정을 통하여 PC에서 로봇에 필요한 프로그램을 다운로드하도록 되어 있다. 또한 89C51에서는 RD/신호를 읽어서 PC와 커넥터가 연결되었는지 연결되지 않았는지를 판단하게 된다.



〈그림V-35〉 RS-232C 전용 IC MAX232와 시리얼 잭

2) 번지지정 회로

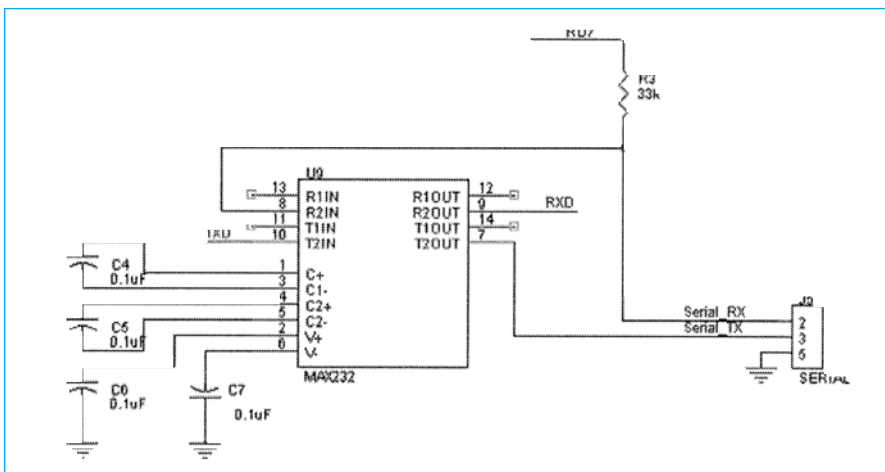
메인보드에서 입력된 A15, A14를 이용하여 번지지정 회로를 설계하였다. 〈그림V-36〉에서 알 수 있듯이 ROMCS/ 신호는 0x4000번지로 지정되어 있고, RobotLED 신호는 WR/ 신호와 동기되어 쓰기번지로 0x8000번지로 지정되어 있다. 또한, 확장 기능에 사용할 수 있도록 IO번지를 0xC0000번지로 지정하여 두었다.



〈그림V-36〉 번지지정 회로

3) LED 구동 회로

〈그림V-37〉은 앞에서 지정된 0x8000번지를 이용하여 74HC574 IC를 통해 LED를 구동하는 회로이다. 74HC574 IC의 11번 핀에 입력되는 번지 지정 신호가 액티브 High일 때, 2번 핀부터 9번 핀까지 입력되는 데이터 신호를 이용하여 출력단자인 12번 핀부터 19번 핀까지에 연결된 LED를 켜거나 끄도록 되어 있다.

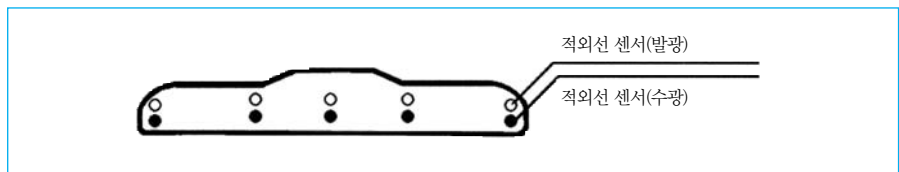


〈그림V-37〉 로봇의 LED 구동 회로

7 센서로 로봇 위치 감지

마이크로 로봇에서는 여러 가지 센서를 이용하여 벽의 감지, 선의 감지 등 위치와 장애물에 대한 판단을 하게 된다. 여기서는 적외선 센서를 이용하여 선을 감지하여 로봇의 움직일 방향을 선택하고, 로봇이 길을 잃지 않고 목적지까지 도달하도록 프로그램을 작성한다.

우선 마이크로 로봇에 부착된 센서를 응용하는 데 필요한 기본 함수들을 설명하도록 하겠다.



로봇의 눈 역할을 하고 있는 적외선 발광 센서에서 빛을 쏘아주면, 하얀색 테이프나 빛을 반사하는 물체를 통해 빛이 반사되어 적외선 수광 센서에 인식이 되어 동작을 하게 된다.

센서에서 읽어 들인 값을 눈으로 확인하기 위해서는 RobotLED를 활용하는 방법이 편리하다. 위 그림과 같이 센서의 방향을 놓고 보면, 맨 왼쪽에 있는 센서가 감지되면 LED7이, 왼쪽 중간에 있는 센서는 LED6, 한 가운데 있는 센서는 LED5, 오른쪽 중간에 있는 센서는 LED4, 맨 오른쪽 센서는 LED3에 할당하여, 센서에서 흰색 라인을 감지하면 LED에 불이 켜지도록 하였다. 이 실험은 검정 바닥에 백색 종이를 로봇 아래에 두고 움직이면서 하면 편리하며, 이 프로그램을 수행하면서 센서의 감도가 좋지 않으면 로봇 앞에 있는 가변저항을 돌리면서 센서의 감도를 향상시킬 수 있다.



예 제

센서를 읽어 로봇 LED에 표시하기

```
#include <io51.h>
#define RobotLED (*(unsigned char *) 0x018000)

bit OutPulse    = 0x90;
bit rLSen       = 0x91;
bit rLCSen      = 0x92;
bit rCSen       = 0x93;
bit rRCSen      = 0x94;
bit rRSen       = 0x95;
void delay(unsigned int t) {
long i ;
for(i = 0 ; i < t ; i ++ ) ;
}
void main(void) {
    unsigned char SenData ;
    for(;;) {
        SenData = 0 ;
        OutPulse = 1 ;
        delay(2);
        if(rLSen==0 ) SenData = SenData | 0x80 ;
        if(rLCSen==0 ) SenData = SenData | 0x40 ;
        if(rCSen==0 ) SenData = SenData | 0x20 ;
        if(rRCSen==0 ) SenData = SenData | 0x10 ;
        if(rRSen== 0) SenData = SenData | 0x08 ;
        OutPulse = 0 ;
        RobotLED = SenData;
    }
}
```

설명

```

☞ bit OutPulse = 0x90;          bit rLSen = 0x91;
   bit rLCSen = 0x92;          bit rCSen = 0x93;
   bit rRCSen = 0x94;          bit rRSen = 0x95 ;

```

센서를 구동하기 위한 신호 출력과 센서에서 값을 읽어오기 위한 신호 입력 위치를 지정한다.

```

☞ unsigned char SenData ;

```

센서에서 읽은 값을 저장할 SenData 변수를 부호 없는 문자형으로 선언한다. 5개의 센서에서 읽어오는 값은 최대 8비트이므로, 문자형 변수로 선언한다.

```

☞ OutPulse = 1 ;

```

센서를 구동(ON)하기 위해 신호 출력에 1을 출력한다.

```

☞ if(rLSen==0 ) SenData = SenData | 0x80 ;

```

만약 왼쪽 센서의 값이 0이면, 하얀 선이 있는 것으로 판단한다.
SenData 변수에 0x80을 OR 연산하여 SenData에 넣는다.

```

☞ if(rLCSen==0) SenData = SenData | 0x40 ;

```

만약 왼쪽 중앙 센서의 값이 0이면, 하얀 선이 있는 것으로 판단, SenData 변수에 0x40을 OR 연산하여 SenData에 넣는다.

```

☞ OutPulse = 0;

```

센서를 OFF한다.

```

☞ RobotLED = SenData;

```

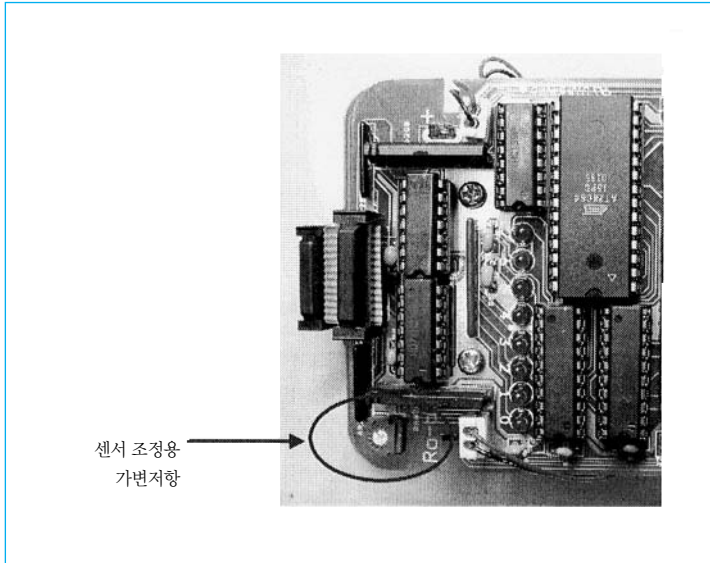
SenData를 RobotLED에 출력하여 LED에 표시합니다.

로봇의 센서는 고정된 방법 등 각각 특성에 따라서 약간씩 다르게 동작할 수 있다. 정확한 센서의 동작을 위해 저항 값을 조정할 수 있도록 하였다.

오른쪽의 그림처럼 로봇의 앞부분에 위치한 가변저항을 이용하여 센서의 감도를 조정하도록 하기 위해서는 예제 프로그램을 컴파일하여 다운로드하여야 한다.

◆ 로봇의 센서 감도 조정 방법 ◆

- ① 리셋 스위치를 누르면서 검정색 바닥 위에 센서를 위치시킨다.
- ② 드라이버를 이용하여 가변저항을 좌우로 돌려 LED에 불이 전부 켜지지 않도록 조정한다.



- ③ 리셋 스위치를 누르면서 하얀색 바닥 위에 센서를 위치시킨다.
- ④ 드라이버를 이용하여 가변저항을 좌우로 돌려 LED에 불이 전부 켜지도록 조정한다.
- ⑤ 리셋 스위치를 누르면서 하얀색 바닥 위에 센서를 위치시킨다.
- ⑥ 센서가 위로 약간 치켜 세워지게 로봇의 뒷부분을 약간 누른다.
- ⑦ 드라이버를 이용하여 가변저항을 좌우로 돌려 LED에 불이 전부 켜지도록 조정한다.

8 로봇의 모터 움직이기

이제부터 사용하게 될 마이크로 로봇에는 사람과 똑같이 눈과 다리의 역할을 하는 장치들이 있다. 먼저 여기서는 사람의 다리와 같이 로봇의 다리 역할을 하는 모터를 움직여 보도록 하자.

1) 모터를 움직이는 기본 함수들

모터를 움직일 때 앞으로 갈 것인지 뒤로 갈 것인지를 결정하기 위해서는 프로그램으로 작성하여 마이크로프로세서에 명령을 주어야 한다. 여기서는 왜 그런 명령을 주는지에 대한 하드웨어적인 설명은 생략하도록 하겠다.

◆ 왼쪽 모터를 움직이게 하는 명령

Left_M0	Left_M1	동 작
1	0	왼쪽 모터 전진 방향으로 회전
0	1	왼쪽 모터 후진 방향으로 회전
1	1	왼쪽 모터 정지
0	0	왼쪽 관성운전

◆ 오른쪽 모터 움직이게 하는 명령

Right_M0	Right_M1	동 작
1	0	오른쪽 모터 전진 방향으로 회전
0	1	오른쪽 모터 후진 방향으로 회전
1	1	오른쪽 모터 정지
0	0	오른쪽 관성운전

앞의 모터를 움직이는 신호가 정리되어 있는 표를 이용하여 왼쪽 모터, 오른쪽 모터를 차례로 움직여 보도록 하겠다.

예 제 왼쪽 바퀴만 전진하기

```
#include <io51.h>
// 모터 제어 신호선이 연결된 CPU 출력 포트를 매크로 상수로 선언
#define LeftM_0 T0
#define LeftM_1 T1

void main(void)
```




```
{
  for(;;) {
    LeftM_0 = 1 ; // 왼쪽 바퀴 전진
    LeftM_1 = 0 ;
  }
}
```

설명

```
☞ #define LeftM_0    T0
   #define LeftM_1    T1
```

왼쪽 모터 구동을 제어할 신호를 출력하는 CPU의 포트를 매크로 상수로 선언하였다. 하드웨어적인 자세한 설명은 생략하도록 하겠다.

```
☞ for(;;) {.....}
```

중괄호 안의 실행문을 무한 반복하여 실행한다.

```
☞ LeftM_0 = 1 ;      LeftM_1 = 0 ;
```

앞의 왼쪽 모터 전진을 위한 명령표에서 설명하였듯이 모터를 전진시키기 위해 각각 명령을 출력한다.

컴파일하여 생성된 hexa 파일을 ROBI STATION 프로그램으로 마이크로 로봇에 다운로드하여 로봇의 움직임을 살펴보면, 왼쪽 모터만 계속 전진하므로 시계 방향으로 로봇이 회전을 하게 된다.

다음은 왼쪽 모터 움직이는 명령 신호표의 다른 명령들을 통해 로봇을 움직여 보도록 하겠다.

예 제 왼쪽 바퀴 전진, 후진, 관성운전, 정지하기

이 예제에서는 왼쪽 모터를 움직이는 명령표의 내용을 전진→관성운전→정지→후진→관성운전→정지→전진...방식으로 운전해 보겠다.

관성운전이란 로봇에 연결된 모터의 특성에 의해 모터의 신호를 모두 끊어놓아도 어느 정도 동안은 계속 움직이게 된다. 이를 관성운전이라 한다.

```

// 왼쪽 바퀴 전진
LeftM_0 = 1 ;           LeftM_1 = 0 ;
for(i = 0 ; i < 30000; i++ ); // 약 1.5초 정도 대기
// 왼쪽 바퀴 관성운전
LeftM_0 = 0 ;           LeftM_1 = 0 ;
for(i = 0 ; i < 30000; i++ );
// 왼쪽 바퀴 정지
LeftM_0 = 1 ;           LeftM_1 = 1 ;
for(i = 0 ; i < 30000; i++ );
// 왼쪽 바퀴 후진
LeftM_0 = 0 ;           LeftM_1 = 1 ;
for(i = 0 ; i < 30000; i++);
// 왼쪽 바퀴 관성운전
LeftM_0 = 0 ;           LeftM_1 = 0 ;
for(i = 0 ; i < 30000; i++ );
// 왼쪽 바퀴 정지
LeftM_0 = 1 ;           LeftM_1 = 1 ;
for(i = 0 ; i < 30000; i++ );
    
```

위 프로그램을 작성하여 컴파일을 통해 생성된 헥사(HEX) 파일을 다운로드 하면 왼쪽 바퀴가 전진 → 관성운전 → 정지 → 후진 → 관성운전 → 정지 → 전진 → ...을 반복하여 움직이게 된다.

예 제 오른쪽 바퀴만 전진, 후진, 관성운전, 정지하기

여기에서는 앞 예제의 왼쪽 바퀴 움직이는 방법과 같이 오른쪽 바퀴를 전진→관성운전→정지→후진→관성운전→정지→전진...방식으로 운전해 보겠다.

```

RightM_0 = 1 ;           // 오른쪽 바퀴 전진
RightM_1 = 0 ;
for(i = 0 ; i < 30000; i++ ); // 약 1.5초 정도 대기

RightM_0 = 0 ;           // 오른쪽 바퀴 관성운전
RightM_1 = 0 ;
for(i = 0 ; i < 30000; i++ );
    
```



```

RightM_0 = 1 ;           // 오른쪽 바퀴 정지
RightM_1 = 1 ;
for(i = 0 ; i < 30000; i++ ) ;

RightM_0 = 0;           // 오른쪽 바퀴 후진
RightM_1 = 1;
for(i = 0; i < 30000; i++);

RightM_0 = 0 ;           // 오른쪽 바퀴 관성 운전
RightM_1 = 0 ;
for(i = 0 ; i < 30000; i++ ) ;

RightM_0 = 1 ;           // 오른쪽 바퀴 정지
RightM_1 = 1 ;
for(i = 0 ; i < 30000; i++ ) ;

```

왼쪽과 오른쪽 바퀴를 따로따로 움직여 보았다. 이것을 응용하여 다음 예제에서는 왼쪽 바퀴와 오른쪽 바퀴를 같이 움직여 로봇이 앞으로 움직이는 프로그램을 작성하도록 하겠다.

예 제 로봇 움직이기 - 전진, 후진

```

//로봇 전진하기
LeftM_0 = 1;           // 왼쪽 바퀴 전진
LeftM_1 = 0;
RightM_0 = 1;          // 오른쪽 바퀴 전진
RightM_1 = 0 ;
for(i = 0 ; i < 30000; i++ ) ;           // 약 1.5초 정도 대기

LeftM_0 = 0;           // 왼쪽 바퀴 관성운전
LeftM_1 = 0;
RightM_0 = 0 ;         // 오른쪽 바퀴 관성운전
RightM_1 = 0 ;
for(i = 0 ; i < 30000; i++ ) ;

LeftM_0 = 1;           // 왼쪽 바퀴 정지
LeftM_1 = 1;

```

```

RightM_0 = 1 ;                // 오른쪽 바퀴 정지
RightM_1 = 1 ;
for(i = 0 ; i < 30000; i++ ) ;

LeftM_0 = 0;                  // 왼쪽 바퀴 후진
LeftM_1 = 1;
RightM_0 = 0;                 // 오른쪽 바퀴 후진
RightM_1 = 1;
for(i = 0; i < 30000; i++);
LeftM_0 = 0;                  // 왼쪽 바퀴 관성운전
LeftM_1 = 0;
RightM_0 = 0;                 // 오른쪽 바퀴 관성운전
RightM_1 = 0;
for(i = 0; i < 30000; i++);

LeftM_0 = 1;                  // 왼쪽 바퀴 정지
LeftM_1 = 1;
RightM_0 = 1;                 // 오른쪽 바퀴 정지
RightM_1 = 1;
for(i = 0; i < 30000; i++);
    
```

예 제 로 봇 움 직 이 기 - 회 전

예제에서는 왼쪽 바퀴와 오른쪽 바퀴를 서로 반대 방향으로 구동하여 로봇이 자리에서 빙글빙글 회전하는 프로그램을 작성해 보겠다.

```

LeftM_0 = 1 ;                // 왼쪽 바퀴 전진
LeftM_1 = 0 ;
RightM_0 = 0 ;                //오른쪽 바퀴 후진
RightM_1 = 1 ;
for(i = 0 ; i < 30000; i++ ) ;

LeftM_0 = 0 ;                // 왼쪽 바퀴 관성운전
LeftM_1 = 0 ;
RightM_0 = 0 ;                // 오른쪽 바퀴 관성운전
RightM_1 = 0 ;
    
```



```

for(i = 0 ; i < 30000; i++ ) ;

LeftM_0 = 1 ;           // 왼쪽 바퀴 정지
LeftM_1 = 1 ;
RightM_0 = 1 ;         // 오른쪽 바퀴 정지
RightM_1 = 1 ;
for(i = 0 ; i < 30000; i++ ) ;

LeftM_0 = 0 ;         // 왼쪽 바퀴 후진
LeftM_1 = 1 ;
RightM_0 = 1 ;       //오른쪽 바퀴 전진
RightM_1 = 0 ;
for(i = 0 ; i < 30000; i++ ) ;

LeftM_0 = 0 ;         // 왼쪽 바퀴 관성운전
LeftM_1 = 0 ;
RightM_0 = 0 ;       // 오른쪽 바퀴 관성운전
RightM_1 = 0 ;
for(i = 0 ; i < 30000; i++ ) ;

LeftM_0 = 1 ;         // 왼쪽 바퀴 정지
LeftM_1 = 1 ;
RightM_0 = 1 ;       // 오른쪽 바퀴 정지
RightM_1 = 1 ;
for(i = 0 ; i < 30000; i++ ) ;

```

2) 모터의 이동 거리와 속도

로봇의 이동하는 거리는 다음 함수를 이용하여 실행하게 된다.

```

// 움직일 거리와 속도를 전달받아 모터를 움직이는 함수
void MotorGo( long Dist, long Speed ) {
    long i;
    // 움직일 거리만큼 반복
    for(i = 0 ; i < Dist ; i++ ) {
        LeftM_0 = 1;    LeftM_1 = 0;    // 로봇 전진
        Right_0 = 1;    Right_1 = 0;
        delay(Speed);    // 계속 전진
    }
}

```



```

LeftM_0 = 0;   LeftM_1 = 0;       // 관성운전
Right_0 = 0;   Right_1 = 0;
delay(15);

LeftM_0 = 1;   LeftM_1 = 1;       // 로봇 정지
Right_0 = 1;   Right_1 = 1;
delay(100-Speed);           // 계속 정지
}
}

```

위 함수를 이용하여 로봇이 움직일 위치와 속도를 정하게 된다.
다음 예제를 이용하여 로봇을 앞으로 움직여 보도록 하겠다.

예 제

모터 거리와 속도를 지정하여 로봇 움직이기

```

// Dist는 모터 이동 거리를 지정하고 Speed는 모터의 속도를 지정한다.
void MotorGo( long Dist, long Speed ) {
    long i;
    for( i = 0 ; i < Dist; i++ ){           // 움직일 거리만큼 반복
        // 로봇 전진시키기
        LeftM_0 = 1 ;   LeftM_1 = 0 ;
        RightM_0 = 1 ;   RightM_1 = 0 ;
        delay(Speed);           // 계속 전진시키기
        // 로봇 관성운전
        LeftM_0 = 0 ;   LeftM_1 = 0 ;
        RightM_0 = 0 ;   RightM_1 = 0 ;
        delay(15);
        // 로봇 정지시키기
        LeftM_0 = 1;   LeftM_1 = 1;
        RightM_0 = 1;   RightM_1 = 1;
        delay(100-Speed);           // 계속 정지하기
    }
}

void main(void) {
    for( ; ) {
        // 모터의 속도를 눈으로 확인하기 위해 LED를 ON

```



```

RobotLED = 0x01;
MotorGo(500,30);
delay(30000);
}
}

```

MotorGo 함수의 Speed값은 0부터 100까지 지정할 수 있다.

예 제 여러 가지의 속도를 지정하여 로봇 움직이기

예제에서 공부한 모터의 이동 거리와 모터의 속도를 제어하는 함수와 프로그램을 이용하여 여러 가지 속도로 로봇을 움직이는 프로그램을 작성해 보도록 하겠다.

```

RobotLED = 0x01; // 모터의 속도를 표시하기 위해 LED 사용
MotorGo(500,10);
delay(30000);
RobotLED = 0x02; // 모터의 속도를 표시하기 위해 LED 사용
MotorGo(500,20);
delay(30000);
RobotLED = 0x03; // 모터의 속도를 표시하기 위해 LED 사용
MotorGo(500,30);
delay(30000);
RobotLED = 0x04; // 모터의 속도를 표시하기 위해 LED 사용
MotorGo(500,40);
delay(30000);
RobotLED = 0x05; // 모터의 속도를 표시하기 위해 LED 사용
MotorGo(500,50);
delay(30000);
RobotLED = 0x06;
MotorGo(500,60);
delay(30000);
RobotLED = 0x07;
MotorGo(500,70);
delay(30000);
RobotLED = 0x08;

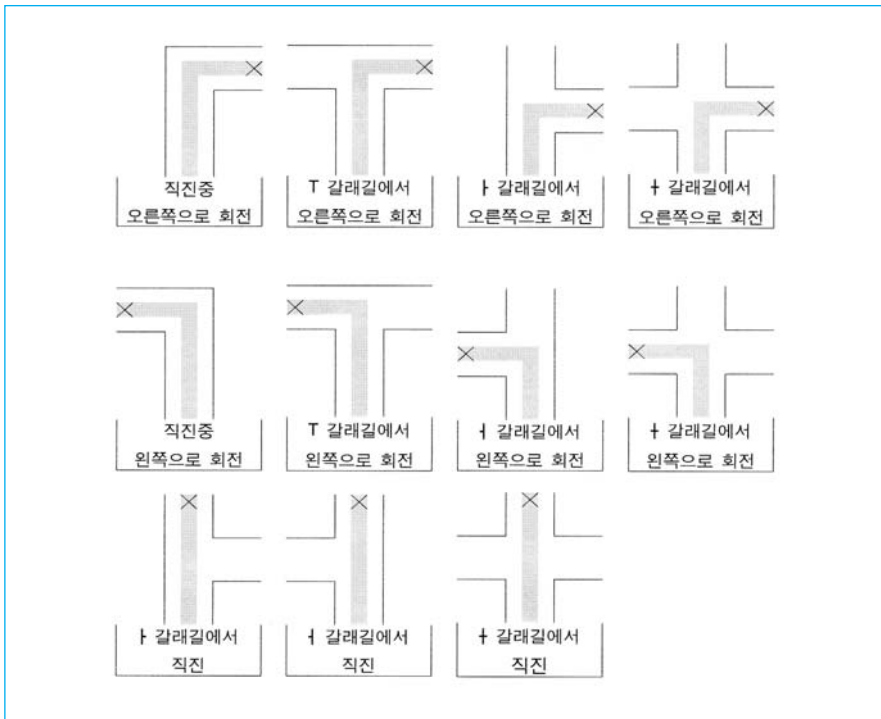
```




이때, 검정색 밑판은 무광택이어야 하며, 하얀색 선은 광택이어야 한다.

여기서 사용하는 로봇의 센서는 빛의 반사를 이용하여 빛이 반사되는 하얀색과 반사되지 않는 검정색을 비교하여 길을 찾아가게 되므로, 미로를 제작할 경우 꼭 지켜야 한다.

로봇이 길을 찾아가다 보면 전진 중 오른쪽으로 회전할 경우가 있고, 세 갈래의 길에서 왼쪽으로 회전할 경우, 네 갈래 길에서 왼쪽으로 회전할 경우 등 여러 가지의 경우가 있을 것이다.



약 13가지의 경우가 나오게 되는데 여기서는 U턴의 경우를 제외한, 각각의 경우를 실험할 수 있도록 실험용 미로판을 고안해 보자.

준비된 미로판을 이용하여 로봇이 미로판의 하얀색 선을 따라 이동하는 예제부터 알아보자.

예제 로봇 이동하기 - 직진

```
#include <io51.h>

#define RobotLED (*(unsigned char *) 0x018000)

#define RightM_0 INT1
#define RightM_1 INTO
#define LeftM_0 T0
#define LeftM_1 T1

bit OutPulse = 0x90;
bit rLSen = 0x91;
bit rLCSen = 0x92;
bit rCSen = 0x93;
bit rRCSen = 0x94;
bit rRSen = 0x95;

#define LSensor 0x80 // 왼쪽 센서 동작
#define LCSensor 0x40 // 왼쪽 중앙 센서 동작
#define CSensor 0x20 // 정중앙 센서 동작
#define RCSensor 0x10 // 오른쪽 중앙 센서 동작
#define RSensor 0x08 // 오른쪽 센서 동작

#define Speed 5 // 로봇의 속도 조정

void delay(unsigned int t) {
    long i ;
    for(i = 0 ; i < t ; i ++ ) ;
}

// 센서 상태 체크
unsigned char SensorScan(void)
{
    unsigned char ScanData = 0;
    OutPulse = 1; // Sensor ON
    delay(14);
    if(rLSen == 0) ScanData = ScanData | 0x80;
    if(rLCSen == 0) ScanData = ScanData | 0x40;
```



```

if(rCSen == 0)          ScanData = ScanData | 0x20;
if(rRCSen == 0)        ScanData = ScanData | 0x10;
if(rRSen == 0)         ScanData = ScanData | 0x08;
OutPulse = 0;          // Sensor OFF

return (ScanData);
}

void main(void) {
    unsigned char ReadData;

    for(;;){
        // 센서 체크
        ReadData = SensorScan();
        RobotLED = ReadData;          //센서 값을 LED에 표시

        // 왼쪽 바퀴 전진, 오른쪽 바퀴 전진
        LeftM_0 = 1;                  LeftM_1 = 0;
        RightM_0 = 1;                 RightM_1 = 0;

        switch(ReadData) {
            // 가운데 센서만 들어오면
            // 왼쪽바퀴 : 전진, 오른쪽 바퀴 : 전진
            case CSensor :
                LeftM_0 = 1;           LeftM_1 = 0;
                RightM_0 = 1;          RightM_1 = 0;
                delay(Speed);
                break;

            // 왼쪽으로 로봇의 방향이 틀어진 경우
            // 왼쪽바퀴 : 전진, 오른쪽 바퀴 : 정지
            case CSensor | RCSensor :
            case RCSensor | RSensor :
            case RSensor :
            case RCSensor :
                LeftM_0 = 1;           LeftM_1 = 0;
                RightM_0 = 1;          RightM_1 = 1;
                delay(Speed+10);
                break;

            // 오른쪽으로 로봇의 방향이 틀어진 경우
            // 오른쪽 바퀴 : 전진, 왼쪽 바퀴 : 정지

```

```

case CSensor | LCSensor :
case LCSensor | LSensor :
case LSensor :
case LCSensor :
    LeftM_0 = 1;           LeftM_1 = 1;
    RightM_0 = 1;         RightM_1 = 0;
    delay(Speed+10);
    break;
// 로봇 정지
case 0x00 :
    LeftM_0 = 0;           LeftM_1 = 0;           //관성운전
    RightM_0 = 0;         RightM_1 = 0;
    delay(15)
    LeftM_0 = 1;           LeftM_1 = 1;
    RightM_0 = 1;         RightM_1 = 1;
    break;
}
LeftM_0 = 0;           LeftM_1 = 0;           //관성운전
RightM_0 = 0;         RightM_1 = 0;
delay(15)
LeftM_0 = 1;           LeftM_1 = 1;
RightM_0 = 1;         RightM_1 = 1;
delay(38);
}
}

```

설명

위 예제는 로봇이 미로의 하얀선을 따라 이동하면서 조금씩 하얀선을 벗어나는 경우, 로봇의 한가운데 센서가 하얀선의 중앙으로 오도록 조정하며 로봇이 직진하는 예제이다. 로봇을 움직이다 보면 한쪽 방향으로 치우치는 경우가 이때 로봇이 오른쪽으로 치우치는 경우와 왼쪽으로 치우치는 경우가 발생하는데, 이때의 로봇 LED의 표시를 보면

- ① 로봇의 위치가 오른쪽으로 치우친 경우 LED 표시



(센서와 매칭되는 LED는 LED3~LED7까지이다. 따라서 LED0, LED1, LED2는 사용하지 않는다.)

하얀선 위에 한가운데 센서와 왼쪽 중앙 센서가 있는 경우

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
○	●	●	○	○	○	○	○

하얀선 위에 왼쪽 중앙 센서만 있는 경우 (LED7~LED3만 사용)

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
○	●	○	○	○	○	○	○

하얀선 위에 왼쪽 센서만 있는 경우 (LED7~LED3만 사용)

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
●	○	○	○	○	○	○	○

하얀선 위에 왼쪽 센서와 왼쪽 중앙 센서가 있는 경우 (LED7~LED3만 사용)

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
●	●	○	○	○	○	○	○

로봇의 방향을 왼쪽으로 전진하여 로봇의 위치를 하얀색 선의 정중앙으로 교정해 주어야 한다. 로봇을 왼쪽 방향으로 전진시키려면, 왼쪽 바퀴는 정지시키고 오른쪽 바퀴는 전진하면, 로봇이 위치를 왼쪽으로 옮겨 하얀색 선 위로 오게 된다.

```
LeftM_0 = 1;           LeftM_1 = 1;
RightM_0 = 1;          RightM_1 = 0;
```

이때 로봇의 속도를 약간 느리게 움직여 정확하게 중앙에 오도록 한다.

② 로봇의 위치가 왼쪽으로 치우친 경우 LED 표시 (LED7~LED3만 사용)

하얀선 위에 한가운데 센서와 오른쪽 중앙 센서가 있는 경우

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
○	○	●	●	○	○	○	○

하얀선 위에 오른쪽 중앙 센서만 있는 경우 (LED7~LED3만 사용)

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
○	○	○	●	○	○	○	○

하얀선 위에 오른쪽 센서만 있는 경우 (LED7~LED3만 사용)

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
○	○	○	○	●	○	○	○

하얀선 위에 오른쪽 센서와 오른쪽 중앙 센서가 있는 경우 (LED7~LED3만 사용)

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
○	○	○	●	●	○	○	○

로봇의 방향을 오른쪽으로 전진하여 로봇의 위치를 하얀색 선의 정중앙으로 교정해 주어야 한다. 로봇을 오른쪽 방향으로 전진시키려면, 오른쪽 바퀴는 정지시키고 왼쪽 바퀴는 전진하면, 로봇이 위치를 오른쪽으로 옮겨 하얀색 선 위로 오게 된다.

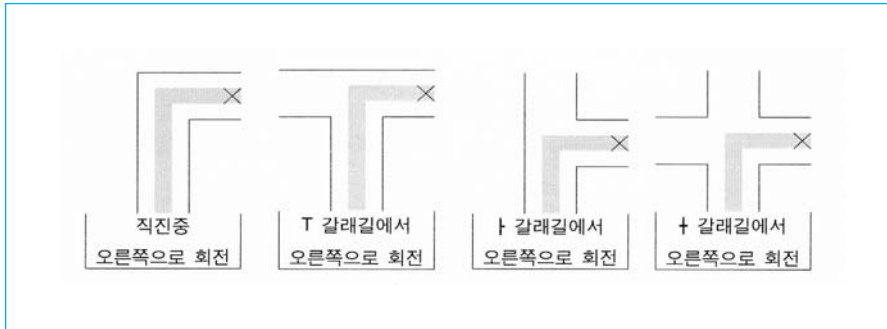
```

LeftM_0 = 1;           LeftM_1 = 0;
RightM_0 = 1;          RightM_1 = 1;
    
```

이때 로봇의 속도를 약간 느리게 움직여 정확하게 중앙에 오도록 한다.

③ 로봇의 회전 - 갈림길에서의 오른쪽으로 회전

로봇이 오른쪽으로 회전하는 경우는 아래 그림과 같은 4가지의 경우이다.



다음은 로봇이 회전하기 직전 회전 모서리에서 라인을 읽었을 때 나타나는 센서 값을 LED로 표시한 것이다.

직진 중 오른쪽으로 회전하는 경우 (LED7~LED3만 사용)

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
○	○	●	○	○	○	○	○

T갈래길에서 오른쪽으로 회전하는 경우 (LED7~LED3만 사용)

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
●	●	●	●	●	○	○	○

⊥ 갈래길에서 오른쪽으로 회전하는 경우 (LED7~LED3만 사용)

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
○	○	●	●	●	○	○	○

† 갈래길에서 오른쪽으로 회전하는 경우 (LED7~LED3만 사용)

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
●	●	●	●	●	○	○	○

예 제

로봇 이동하기 - 오른쪽으로 회전

```

#include <io51.h>

#define RobotLED (*(unsigned char *) 0x018000)

#define RightM_0 INT1
#define RightM_1 INTO
#define LeftM_0 T0
#define LeftM_1 T1

bit OutPulse = 0x90;
bit rLSen = 0x91;
bit rLCSen = 0x92;
bit rCSen = 0x93;
bit rRCSen = 0x94;
bit rRSen = 0x95;

#define LSensor 0x80 // 왼쪽 센서 동작
#define LCSensor0x40 // 왼쪽 중앙 센서 동작
#define CSensor 0x20 // 정중앙 센서 동작
#define RCSensor 0x10 // 오른쪽 중앙 센서 동작
#define RSensor 0x08 // 오른쪽 센서 동작

#define Speed 30 // 로봇의 속도 조정
#define RightAngle 90 // 오른쪽으로 회전할 각도
#define RightTurnSpeed (Speed + 10) // 회전 속도

void delay(unsigned int t){
    long i ;
    for(i = 0 ; i < t ; i ++ ) ;
}

// 센서 상태 체크
unsigned char SensorScan(void){
    unsigned char ScanData = 0;
    OutPulse = 1; // Sensor ON
    delay(14);
    if(rLSen == 0) ScanData = ScanData | 0x80;
    if(rLCSen == 0) ScanData = ScanData | 0x40;

```




```

if(rCSen == 0)      ScanData = ScanData | 0x20;
if(rRCSen == 0)    ScanData = ScanData | 0x10;
if(rRSen == 0)     ScanData = ScanData | 0x08;
OutPulse = 0;      // Sensor OFF
return (ScanData);
}

void SlowGo(long dest){
    long i;

    for(i=0; i<dest; i++) {
        LeftM_0 = 1;      LeftM_1 = 0;      //왼쪽, 오른쪽 전진
        RightM_0 = 1;     RightM_1 = 0;
        delay(Speed);

        LeftM_0 = 0;      LeftM_1 = 0;      //관성운전
        RightM_0 = 0;     RightM_1 = 0;
        delay(15);

        LeftM_0 = 1;      LeftM_1 = 1;      // 왼쪽, 오른쪽 정지
        RightM_0 = 1;     RightM_1 = 1;
        delay(100-Speed);
    }
}

// 오른쪽 바퀴 정지, 왼쪽 바퀴는 전진, RightAngle만큼 회전
void RotateRight(long rotangle){
    long i;

    for(i=0; i<rotangle; i++) {
        LeftM_0 = 1;      LeftM_1 = 0;      // 왼쪽 전진
        RightM_0 = 1;     RightM_1 = 1;     // 오른쪽 정지
        delay(RightTurnSpeed);
        LeftM_0 = 0;      LeftM_1 = 0;      //관성운전
        RightM_0 = 0;     RightM_1 = 0;
        delay(15);
        LeftM_0 = 1;      LeftM_1 = 1;
        RightM_0 = 1;     RightM_1 = 1;
        delay(100-RightTurnSpeed);
    }
}
}

```

```

void RightTurn(void) {
    SlowGo(40); // 약간 앞으로 이동
    RotateRight(RightAngle); // 오른쪽으로 회전
    while(!(SensorScan() & 0x40)) RotateRight(1);
}

void main(void) {
    unsigned char ReadData;

    for(;;){
        // 센서 체크
        ReadData = SensorScan();
        RobotLED = ReadData; //센서 값을 LED에 표시

        // 왼쪽 바퀴 전진, 오른쪽 바퀴 전진
        LeftM_0 = 1; LeftM_1 = 0;
        RightM_0 = 1; RightM_1 = 0;
        switch(ReadData) {
            // 가운데 센서만 들어오면
            // 왼쪽 바퀴 : 전진, 오른쪽 바퀴 : 전진
            case CSensor :
                LeftM_0 = 1; LeftM_1 = 0;
                RightM_0 = 1; RightM_1 = 0;
                delay(Speed);
                break;
            // 왼쪽으로 로봇의 방향이 틀어진 경우
            // 왼쪽 바퀴 : 전진, 오른쪽 바퀴 : 정지
            case CSensor | RCSensor :
            case RCSensor | RSensor :
            case RSensor :
            case RCSensor :
                LeftM_0 = 1; LeftM_1 = 0;
                RightM_0 = 1; RightM_1 = 1;
                delay(Speed+10);
                break;
            // 오른쪽으로 로봇의 방향이 틀어진 경우
            // 오른쪽 바퀴 : 전진, 왼쪽 바퀴 : 정지
            case CSensor | LCSensor :
            case LCSensor | LSensor :
            case LSensor :

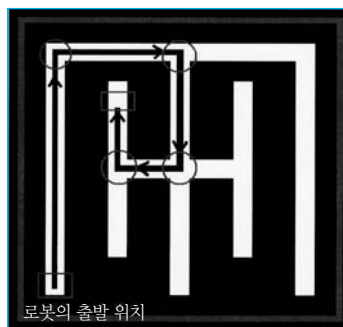
```



```

case LCSensor :
    LeftM_0 = 1;           LeftM_1 = 1;
    RightM_0 = 1;          RightM_1 = 0;
    delay(Speed+10);
    break;
// 장애물이 있는 경우 일단 로봇을 정지 시킨다.
case CSensor | LCSensor | RCSensor :
    LeftM_0 = 0;           LeftM_1 = 0;      // 관성운전
    RightM_0 = 0;          RightM_1 = 0;
    delay(15);
    LeftM_0 = 1;           LeftM_1 = 1;      // 정지
    RightM_0 = 1;          RightM_1 = 1;
    delay(Speed);
    break;
// 한가운데 센서, 오른쪽 중앙 센서, 오른쪽 센서 동작
// ? 모퉁이인 경우 오른쪽으로 회전
case CSensor | RCSensor | RSensor :
// ? 갈림길인 경우 오른쪽으로 회전
case CSensor | RCSensor | RSensor | LCSensor :
// ?, T 갈림길인 경우 오른쪽으로 회전
case CSensor | LCSensor | LSensor | RCSensor | RSensor :
    RightTurn();
    break;
case 0x00 : // 모든 센서가 동작하지 않은 경우
    LeftM_0 = 0;           LeftM_1 = 0;      //관성운전
    RightM_0 = 0;          RightM_1 = 0;
    delay(15);
    LeftM_0 = 1;           LeftM_1 = 1;
    RightM_0 = 1;          RightM_1 = 1;
    break;
}
LeftM_0 = 0;           LeftM_1 = 0;      //관성운전
RightM_0 = 0;          RightM_1 = 0;
delay(15);
LeftM_0 = 1;           LeftM_1 = 1;
RightM_0 = 1;          RightM_1 = 1;
delay(38);
}
}

```



[실험용 미로판]

위 예제 프로그램을 이용하여 실험용 미로판의 갈림길에서 오른쪽으로 회전하는지 확인하여 보자. 로봇을 움직였을 때, 로봇의 모터가 가지고 있는 고유 특성과 건전지의 남은 전류량에 따라 조금씩 다르게 동작할 수 있다. 이러한 경우에는 아래 표와 같은 세 가지의 매크로 상수 값을 이용하여 로봇이 정상적인 운전을 하도록 조정하게 되어 있다.

변수명	조정되는 내용	조정 범위
Speed	로봇이 움직이는 속도 조정	1 - 100
RightAngle	로봇이 회전하는 각도 조정	1 - 120
RightTurnSpeed	로봇이 회전할 때의 속도 조정	1 - 100

위의 값들을 조정하면서 속도를 너무 빠르게 하면 갈림길을 인식하지 못하고 그냥 지나칠 수 있으므로, 갈림길을 정확하게 인식하도록 적당한 속도를 지정하여 운전하여야 한다. 또한, 갈림길에서 회전을 할 때 회전하는 각도 값이 너무 작거나 너무 크면, 로봇이 길에서 이탈하여 동작하므로 정상적인 운전을 하지 못할 수 있다.

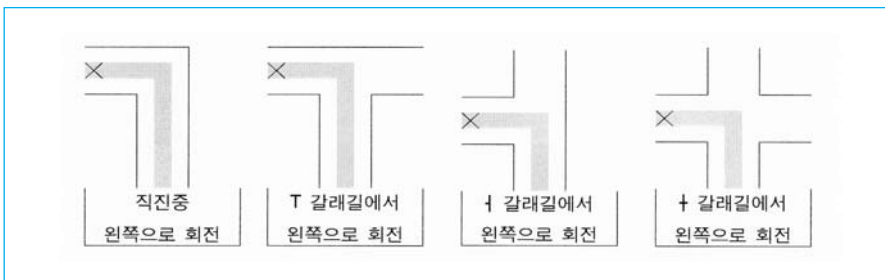
로봇의 회전하는 반경이 너무 적으면 값을 크게 하고, 회전하는 반경이 너무 크면 값을 작게 조정하여야 한다. 로봇이 회전하는 속도는 모터의 특성에 따라 로봇마다 다르게 속도가 적용되므로, 여러 번의 실험을 통해서 로봇에게 가장



적당한 값을 찾아 로봇을 움직이도록 하여야 한다.

④ 로봇의 회전 - 갈림길에서의 왼쪽으로 회전

로봇이 왼쪽으로 회전하는 경우는 오른쪽으로의 회전과 마찬가지로 아래 그림과 같은 4가지의 경우이다. 센서의 동작 상태는 오른쪽으로 회전하는 경우와 반대로 센서가 동작한다. 단, T자 갈림길과 +갈림길에서는 똑같이 동작을 하게 되지만, 이때는 어느 한 방향으로만 진행을 하도록 프로그램을 하면 된다.



각각의 회전하는 경우가 발생할 때 센서의 동작을 보면,

직진 중 왼쪽으로 회전하는 경우 (LED7~LED3만 사용)

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
●	●	●	○	○	○	○	○

T갈래길에서 왼쪽으로 회전하는 경우 (LED7~LED3만 사용)

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
●	●	●	●	●	○	○	○

┣ 갈래길에서 왼쪽으로 회전하는 경우 (LED7~LED3만 사용)

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
●	●	●	○	○	○	○	○

↑갈래길에서 왼쪽으로 회전하는 경우 (LED7~LED3만 사용)

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
●	●	●	●	●	○	○	○

예 제 로봇 이동하기 - 왼쪽으로 회전

```
#include <io51.h>

#define RobotLED  (*(unsigned char *) 0x018000)

#define RightM_0  INT1
#define RightM_1  INT0
#define LeftM_0   T0
#define LeftM_1   T1

bit OutPulse     = 0x90;
bit rLSen        = 0x91;
bit rLCSen       = 0x92;
bit rCSen        = 0x93;
bit rRCSen       = 0x94;
bit rRSen        = 0x95;

#define           LSensor           0x80    // 왼쪽 센서 동작
#define           LCSensor0x40      // 왼쪽 중앙 센서 동작
#define           CSensor           0x20    // 정중앙 센서 동작
#define           RCSensor          0x10    // 오른쪽 중앙 센서 동작
#define           RSensor           0x08    // 오른쪽 센서 동작

#define           Speed              30     // 로봇의 속도 조정
#define           LeftAngle90        // 왼쪽으로 회전할 각도
#define           LeftTurnSpeed      (Speed+10) //회전 속도

void delay(unsigned int t) {
    long i ;
    for(i = 0 ; i < t ; i ++ ) ;
}

// 센서 상태 체크
```



```

unsigned char SensorScan(void) {
    unsigned char ScanData = 0;

    OutPulse = 1;           // Sensor ON
    delay(14);
    if(rLSen == 0)          ScanData = ScanData | 0x80;
    if(rLCSen == 0)        ScanData = ScanData | 0x40;
    if(rCSen == 0)         ScanData = ScanData | 0x20;
    if(rRCSen == 0)        ScanData = ScanData | 0x10;
    if(rRSen == 0)         ScanData = ScanData | 0x08;
    OutPulse = 0;           // Sensor OFF
    return (ScanData);
}

void SlowGo(long dest) {
    long i;

    for(i=0;i<dest;i++)
    {
        LeftM_0 = 1;        LeftM_1 = 0;    //왼쪽, 오른쪽 전진
        RightM_0 = 1;       RightM_1 = 0;
        delay(Speed);
        LeftM_0 = 0;        LeftM_1 = 0;    //관성운전
        RightM_0 = 0;       RightM_1 = 0;
        delay(15);
        LeftM_0 = 1;        LeftM_1 = 1;    // 왼쪽, 오른쪽 정지
        RightM_0 = 1;       RightM_1 = 1;
        delay(100-Speed);
    }
}

// 왼쪽 바퀴 후진, 오른쪽 바퀴 전진, LeftAngle만큼 회전
void RotateLeft(long rotangle)
{
    longg i;
    for(i=0; i<rotangle;i++) {
        LeftM_0 = 1;        LeftM_1 = 1;    //왼쪽으로 회전
        RightM_0 = 1;       RightM_1 = 0;
        delay(LeftTurnSpeed);
        LeftM_0 = 0;        LeftM_1 = 0;    //관성운전
    }
}

```

```

        RightM_0 = 0;           RightM_1 = 0;
        delay(15);
        LeftM_0 = 1;           LeftM_1 = 1;    //정지
        RightM_0 = 1;         RightM_1 = 1;
        delay(100-LeftTurnSpeed);
    }
}

void LeftTurn(void) {
    SlowGo(40);
    RotateLeft(LeftAngle);
    while(!(SensorScan() & 0x10))    RotateLeft(1);
}

void main(void) {
    unsigned char ReadData;

    for(;;){
        // 센서 체크
        ReadData = SensorScan();
        RobotLED = ReadData;          //센서 값을 LED에 표시

        // 왼쪽 바퀴 전진, 오른쪽 바퀴 전진
        LeftM_0 = 1;                 LeftM_1 = 0;
        RightM_0 = 1;                RightM_1 = 0;
        switch(ReadData) {
            // 가운데 센서만 들어오면
            // 왼쪽 바퀴 : 전진, 오른쪽 바퀴 : 전진
            case CSensor :
                LeftM_0 = 1;         LeftM_1 = 0;
                RightM_0 = 1;        RightM_1 = 0;
                delay(Speed);
                break;
            // 왼쪽으로 로봇의 방향이 틀어진 경우
            // 왼쪽 바퀴 : 전진, 오른쪽 바퀴 : 정지
            case CSensor | RCSensor :
            case RCSensor | RSensor :
            case RSensor :
            case RCSensor :

```




```

LeftM_0 = 1;          LeftM_1 = 0;
RightM_0 = 1;         RightM_1 = 1;
delay(Speed+10);
break;
// 오른쪽으로 로봇의 방향이 틀어진 경우
// 오른쪽 바퀴 : 전진, 왼쪽 바퀴 : 정지
case CSensor | LCSensor :
case LCSensor | LSensor :
case LSensor :
case LCSensor :
    LeftM_0 = 1;          LeftM_1 = 1;
    RightM_0 = 1;         RightM_1 = 0;
    delay(Speed+10);
    break;
// 장애물이 있는 경우 일단 로봇을 정지 시킨다.
case CSensor | LCSensor | RCSensor :
    LeftM_0 = 0;          LeftM_1 = 0;          //관성운전
    RightM_0 = 0;         RightM_1 = 0;
    delay(15);
    LeftM_0 = 1;          LeftM_1 = 1;
    RightM_0 = 1;         RightM_1 = 1;
    delay(Speed+10);
    break;
// 한가운데 센서, 왼쪽 중앙 센서, 왼쪽 센서 동작
// ? 모퉁이인 경우 왼쪽으로 회전
case CSensor | LCSensor | LSensor :
// ? 갈림길인 경우 왼쪽으로 회전
case CSensor | LCSensor | LSensor | RCSensor :
// ?, T 갈림길인 경우 왼쪽으로 90도 회전
case CSensor | LCSensor | LSensor | RCSensor | RSensor :
    LeftTurn();
    break;
case 0x00 :          // 모든 센서가 동작하지 않은 경우
    LeftM_0 = 0;          LeftM_1 = 0;          //관성운전
    RightM_0 = 0;         RightM_1 = 0;
    delay(15);
    LeftM_0 = 1;          LeftM_1 = 1;
    RightM_0 = 1;         RightM_1 = 1;
    break;

```



```

    }

    LeftM_0 = 0;           LeftM_1 = 0;       //관성운전
    RightM_0 = 0;          RightM_1 = 0;
    delay(15);
    LeftM_0 = 1;           LeftM_1 = 1;
    RightM_0 = 1;          RightM_1 = 1;
    delay(38);
  }
}

```

위 예제 프로그램을 작성하여 로봇을 움직여 보도록 하겠다.

이번에는 왼쪽으로만 회전하므로 그림과 같이 로봇이 움직이게 된다.

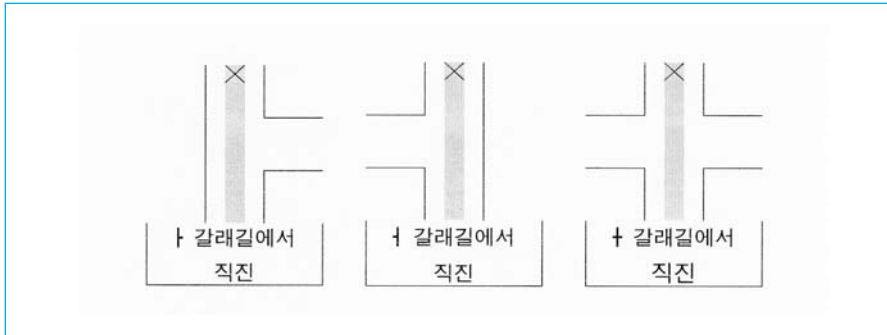


로봇이 움직이는 속도 또는 각도, 회전속도를 조정하려면 예제에서와 같이 Speed 매크로 상수의 값과 LeftAngle, LeftTurnSpeed 매크로 상수의 값을 조절한다.

⑤ 로봇의 이동 - 갈림길에서의 직진 이동

앞에서는 갈림길에서 로봇을 오른쪽 또는 왼쪽으로 회전하는 방법을 예제를 통해 알아보았다.

이번에는 갈림길에서 로봇이 직진을 하는 경우에 대해 알아보겠다.



로봇이 갈림길에서 직진하는 경우는 아래 그림과 같이 세 가지의 경우가 있다. 각 경우에 센서의 동작 상태를 보면,

┆ 갈림길에서의 센서 동작 (LED7~LED3만 사용)

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
○	○	●	●	●	○	○	○

┆ 갈림길에서 왼쪽으로 회전하는 경우 (LED7~LED3만 사용)

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
●	●	●	○	○	○	○	○

┆ 갈림길에서 오른쪽으로 회전하는 경우 (LED7~LED3만 사용)

LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
●	●	●	●	●	○	○	○

예 제

로봇 이동하기 - 갈림길에서 직진

```
#include <io51.h>

#define RobotLED (*(unsigned char *) 0x018000)

#define RightM_0 INT1
#define RightM_1 INT0
#define LeftM_0 T0
```

```

#define LeftM_1 T1

bit OutPulse = 0x90;
bit rLSen = 0x91;
bit rLCSen = 0x92;
bit rCSen = 0x93;
bit rRCSen = 0x94;
bit rRSen = 0x95;

#define LSensor 0x80 // 왼쪽 센서 동작
#define LCSensor 0x40 // 왼쪽 중앙 센서 동작
#define CSensor 0x20 // 정중앙 센서 동작
#define RCSensor 0x10 // 오른쪽 중앙 센서 동작
#define RSensor 0x08 // 오른쪽 센서 동작
#define Speed 30 // 로봇의 속도 조정

void delay(unsigned int t){
    long i ;
    for(i = 0 ; i < t ; i ++ ) ;
}

// 센서 상태 체크
unsigned char SensorScan(void){
    unsigned char ScanData = 0;

    OutPulse = 1; // Sensor ON
    delay(14);
    if(rLSen == 0) ScanData = ScanData | 0x80;
    if(rLCSen == 0) ScanData = ScanData | 0x40;
    if(rCSen == 0) ScanData = ScanData | 0x20;
    if(rRCSen == 0) ScanData = ScanData | 0x10;
    if(rRSen == 0) ScanData = ScanData | 0x08;
    OutPulse = 0; // Sensor OFF
    return (ScanData);
}

void main(void)
{
    unsigned char ReadData;
    for(;;){

```



```

// 센서 체크
ReadData = SensorScan();
RobotLED = ReadData;           //센서값을 LED에 표시

// 왼쪽 바퀴 전진, 오른쪽 바퀴 전진
LeftM_0 = 1;                   LeftM_1 = 0;
RightM_0 = 1;                  RightM_1 = 0;

switch(ReadData)
{
    // 가운데 센서만 들어오면 왼쪽 바퀴 : 전진, 오른쪽 바퀴 : 전진
    case CSensor :
        // ?모퉁이인 경우 직진
        case CSensor | RCSensor | RSensor :
            // ?모퉁이인 경우 직진
            case CSensor | LCSensor | LSensor :
                // ? 모퉁이인 경우 직진
                case CSensor | RCSensor | RSensor | LCSensor :
                    // ? 갈림길인 경우 직진
                    case CSensor | LCSensor | LSensor | RCSensor :
                        // ?, ? 갈림길인 경우 직진
                        case CSensor | LCSensor | LSensor | RCSensor | RSensor :
                            LeftM_0 = 1;           LeftM_1 = 0;
                            RightM_0 = 1; RightM_1 = 0;
                            delay(Speed);
                            break;
                // 왼쪽으로 로봇의 방향이 틀어진 경우
                // 왼쪽 바퀴 : 전진, 오른쪽 바퀴 : 정지
                case CSensor | RCSensor :
                case RCSensor | RSensor :
                case RSensor :
                case RCSensor :
                    LeftM_0 = 1;           LeftM_1 = 0;
                    RightM_0 = 1;         RightM_1 = 1;
                    delay(Speed+10);
                    break;
                // 오른쪽으로 로봇의 방향이 틀어진 경우
                // 오른쪽 바퀴 : 전진, 왼쪽 바퀴 : 정지
                case CSensor | LCSensor :
                case LCSensor | LSensor :

```

```

case LSensor :
case LCSensor :
    LeftM_0 = 1; LeftM_1 = 1;
    RightM_0 = 1;      RightM_1 = 0;
    delay(Speed-3);
    break;
// 장애물이 있는 경우 일단 로봇을 정지시킨다.
case CSensor | LCSensor | RCSensor :
    LeftM_0 = 0; LeftM_1 = 0;
    RightM_0 = 0;      RightM_1 = 0;
    delay(15);
    LeftM_0 = 1; LeftM_1 = 1;
    RightM_0 = 1;      RightM_1 = 1;
    delay(Speed);
    break;
case 0x00 : // 모든 센서가 동작하지 않은 경우
    LeftM_0 = 0; LeftM_1 = 0;
    RightM_0 = 0;      RightM_1 = 0;
    delay(15);
    LeftM_0 = 1; LeftM_1 = 1;
    RightM_0 = 1;      RightM_1 = 1;
    break;
}
LeftM_0 = 0;      LeftM_1 = 0;
RightM_0 = 0;      RightM_1 = 0;
delay(15);
LeftM_0 = 1;      LeftM_1 = 1;
RightM_0 = 1;      RightM_1 = 1;
delay(38);
}
}

```

위 예제 프로그램을 작성하여 로봇에 다운로드한 뒤 미로판에서 움직여 보도록 하자.

이렇게 해서 미로판을 이용하여 로봇이 길을 찾아가기 위한 기본 프로그램들



을 작성해 보았다. 이제부터 여러 가지의 경우에 대한 미로를 만들어 여러분들이 작성한 프로그램을 이용하여 로봇이 미로를 탈출하도록 프로그램하여 보자.

1 산업용 로봇의 형태별 분석 및 사용 예

여기에서는 산업용 로봇이 형태별로 사용되는 예를 들어 봄으로써 산업용 로봇을 설계하거나 제어할 때 어떤 요소들이 고려되어야 하는지에 알아보자.

1 머니플레이터들의 각 형상의 장단점 및 전형적인 응용

(1) 직교 좌표형(Cartesian)

장 점	단 점
좌우 운동축을 따라 주행하므로 넓은 작업 영역을 얻음	유지 보수에 어려움
간단한 제어로 직선 이동이 가능	한번에 한 방향으로 이동
높은 기계적 강선, 정확도와 정밀도	로봇 자체 앞에만 접근 가능
무거운 하중을 운반	설치 공간이 커야 함
오프라인 프로그래밍이 용이	밀봉(seal)이 어려운 축

1) 작업 영역

로봇의 형상과 손목의 관절, 링크(link)의 크기에 따라서 작업 영역이라고 불리는 점들이 집합에 도달이 가능하다. 각각의 로봇마다 작업 영역의 모양새는 로봇의 특성에 따라서 달라진다. 작업 영역은 각 관절의 한계운동 영역을 포함하여 로봇의 관절과 링크를 정의하는 수학적 운동방정식으로 생성할 수 있다. 또한 작업영역은 각 관절을 운동영역 안에서 움직이거나 도달할 수 있는 영역



을 모두 포함하고, 도달하지 못하는 영역은 삭제하는 방식으로 실험을 함으로
도 생성이 가능하다.

2) 응용

- 픽-플레이스(pick-and-place) 작동
- 표면 마무리 가공작업
- 핵 재료 취급
- 물분사(waterjet) 절단
- 점착성의 응용
- 원격작동 오염 제거
- 조립과 부조립
- 일반적인 기계작업
- CNC 선반과 밀링의 자동 장착
- 정밀검사
- 로봇을 이용한 X-선과 중성자 방사선
사진 촬영
- 용접

(2) 원통 좌표형(Cylindrical)

장 점	단 점
수직 구조는 층 공간을 유지함	회전 축이 직교 좌표형 로봇의 강성보다 낮음
원거리 범위 작업에 유용	반복 정밀도와 정확도가 회전 이동의 방향에서 낮음
큰 적재하중 능력	로봇 자체 위에 접근 불가능
로봇 주위에 접근 가능	제어 시스템이 복잡
밀봉이 용이한 회전축	밀봉이 어려운 직선축

1) 작업 영역

직교 좌표형 로봇보다 더 큰 작업 영역을 가지며, 2개의 실린더 사이의 공간
이 원통 좌표형 로봇이 차지하는 작업 영역이다.

2) 응용

- 픽-플레이스 작업
- 고기포장

- 부품의 장착과 탈착
- 가공물 주조
- 주물과 단조 응용
- 일반적인 물류 이송
- 컨베이어 팻릿(pallet) 운송
- 정밀 주조
- 사출 성형
- 다이 캐스팅
- 코팅 응용
- 조립

(3) 구 좌표형(Spherical)

장 점	단 점
직교형, 원통형보다 더 큰 작업 영역	작업 영역의 중심부는 작업 불가능한 영역
디자인이 단순	수직 방향 이동이 매우 제한
좋은 적재 하중 능력	높이가 낮고 길이가 긴 기하학적 구조
1개 직선, 2개 회전운동으로 공간 활용 극대	장애물 주위에 접근 불가능
어떤 방향이라도 로봇이 위치함	
로봇의 끝점이 지정된 위치에 직접 도달	

1) 작업 영역

극 좌표형 로봇의 작업 영역은 두 개의 반구 사이의 공간이다.

2) 응용

- 다이캐스팅
- 딥 코팅(dip coating)-말단에 gripper
- 유리 공정 처리
- 공작기계 공구 장착
- 물류 이송
- 부품 청소
- 단조
- 열처리
- 사출 성형
- 프레스 장착
- 적재와 하역



(4) 수직 관절형(Articulated)

장 점	단 점
3개의 회전축	복잡한 지점 도달 방법
다양한 방향으로 빠른 속도로 이동	복잡한 구조의 머니플레이터
장애물 상하에 접근 가능	
최소 공간에 최대 작업 영역	
주어진 위치에 다양한 각도로 접근	

1) 작업 영역

작업 영역은 로봇을 위에서 보았을 때 원형이고, 측면에서 보았을 때 관절의 운동 한계로 인하여 내부의 부채꼴 모양을 가진 형태를 한다.

2) 응용

- 자동 조립
- 재료 절단
- 다이 캐스팅
- 재료 절삭
- 공정 검사
- 적재
- 기계 비전
- 열코팅
- 부품 및 장치의 장착과 탈착
- 용접
- 도장 및 접착제 도포 공정에 응용

(5) 수평 관절형(SCARA)

장 점	단 점
가장 유연한 형상	정교한 제어기 필요
큰 작업 영역	오프라인 프로그래밍의 어려움
최소의 설치 공간 차지	복잡한 구조의 팔
넓은 수평적 도달 범위	정확도, 하중 운송 용량, 동적 거동, 움직임의
강성 구조의 높이 축	반복 정밀도에 다른 값 가질 수 있음
장애물 주위에 접근 가능	



1) 작업 영역

장애물 주위에 도달할 수 있는 두 개의 실린더 사이의 공간

2) 응용

- 정밀조립
- 장착, 탈착
- 미세부품 납땜
- 사출물 삽입, 검사, 운반, 게이징

2 3가지 동력 공급장치의 경제성, 신뢰도, 하중 능력 등의 장단점 비교.

(1) 전기 공급장치(Electric)

모든 로봇 시스템들은 주 에너지원으로 전기를 사용한다. 전기는 유압과 공압을 제공하는 펌프를 회전시킨다. 또한 로봇 제어기, 모든 전자 부품과 주변 장치에 동력을 공급한다. 모든 전기 로봇에서는 제어기뿐만 아니라 액츄에이터에 전기로 동력이 공급된다. 대부분의 전기 로봇은 다축으로 움직이기 위해 서보모터를 사용하지만, 몇 개의 개루프 로봇 시스템은 스테핑 모터를 이용한다. 대다수의 로봇들이 현재 직류 서보모터를 갖추고 있지만, 교류 서보모터의 신뢰성, 소형화, 고성능으로 인하여 결국에는 교류 서보모터로 바뀔 것이다.

전기 로봇은 유압동력 장치를 필요로하지 않기 때문에, 공장의 장치 설치 영역과 소음을 감소시킨다. 직접구동 모델은 매우 빠른 응답을 제공한다. 전력이 구동축의 액츄에이터에 직접 적용되기 때문에 감속장치를 이용하는 에너지 전환은 필요하지 않다. 전기 액츄에이터는 작업하중 능력이 300lb 이하로 한정되어 있고, 폭발성이 있는 주위 환경에서의 작업이 문제를 야기한다.



장 점	단 점
모든 크기의 로봇에 유리	강성도가 낮음
제어가 쉽고 고정밀도 로봇에 유리	감속기어가 필요하며, 백래시, 가격, 무게 등이 커짐
유압보다 순응도가 높음	모터에 전원이 공급되지 않을 때 제동 장치가 필요함
모터의 관성을 줄이기 위한 감속기어가 필요함	- 그렇지 않으면 로봇팔이 처짐
누수가 없으며 청결한 환경에 사용	
신뢰도가 높으며 유지 보수가 낮음	
방전이 없고 폭발 위험이 있는 곳에서도 사용 가능	

(2) 유압 공급장치(Hydraulic)

유압 시스템과 구동기는 높은 동력 대 중량비를 제공하고, 저속에서도 큰 힘을 낼 수 있으며 마이크로프로세서와 전자제어 시스템과도 호환이 가능할 뿐 아니라, 극도의 위험한 환경에서도 견딜 수 있는 내구력이 있다. 지난 10년 동안 Cincinnati Milacron™ T3 유압로봇이나 다른 회사에서 제작한 유사한 로봇은 주로 자동차 생산 분야에 사용되었다. T3 로봇은 높이가 7ft이며 적재중량이 220lb를 넘는 혁신적인 한계 수치를 갖는 로봇이었다. 그러나 유압 시스템의 고질적인 누설 문제와 고가이며 무거운 구동기의 문제로 더 이상 사용되지 않는다. 오늘날 대부분의 로봇은 전자로봇이나 여전히 산업체에서는 유압로봇이 많이 사용되고 있다. 게다가 매우 큰 로봇, 공공 서비스 로봇과 같은 특수용에는 유압 구동기가 합리적인 선택이 될 수 있다.

장 점	단 점
대형이나 큰 하중을 들 수 있는 로봇에 유리함	누수가 있을 수 있음 - 청결이 필요한 곳은 적합하지 않음
동력/무게비가 가장 큼	펌프, 저수조, 모터, 호스가 필요함
강선 시스템, 고정밀도, 향상된 응답 속도	매우 고가이며 잡음이 심함
감속 기어가 필요 없음	유지 보수가 필요
넓은 범위의 속도로 작업이 손쉬움	유체 점성이 온도에 따라 변함
손실 없이 위치를 유지 가능	유체에 이물질이 있어도 사용 가능
뛰어난 제어 및 조절성	순응도가 낮음
정확한 작동과 반전 및 열 방출성	구동기의 토크가 세고 압력이 높으며 관성이 큼
과부하에 대한 안전성과 시동 가능	

(3) 공압 공급장치(Pneumatic)

공압장치는 대체로 유압장치와 매우 유사하다. 압축된 공기가 동력원으로 사용되거나, 솔레노이드 밸브에 의해서 수동적 또는 전기적으로 제어되는 선형 및 회전 실린더 장치를 구동한다. 압축공기가 동작하는 구동기로부터 분리되어 있으므로, 이러한 시스템들은 관성부하가 매우 낮다. 그러나 공압장치는 아주 낮은 공압(100~120psi 정도)에서 동작하므로, 동력 대 중량비는 유압장치보다 훨씬 더 낮다.

공압장치의 중요한 문제는 부하가 작용할 때의 압축과 변형이다. 따라서 공압 실린더는 단지 구동기 앞, 뒤로는 움직이는 부품 삽입용이나 1/2자유도 관절에만 사용된다. 그렇지 않은 경우에는 공압 실린더의 정확한 위치제어가 매우 어렵다. 공압 실린더의 변위를 제어하는 한 방법으로 디퍼렌셜 디더링(differential dithering)이 있다. 이 시스템은 피스톤의 정확한 위치를 선형 엔코더(linear encoder)나 퍼텐쇼미터(potentiometer)와 같은 되먹임 센서를 통해 측정한다. 이 정보는 정밀한 위치를 제어하기 위해 서보밸브를 사용하는 실린더의 양단에서 공압을 제어하는 제어기에서 사용한다.



장 점	단 점
많은 요소들에 사용 가능	소음이 있음. 공압과 필터 등이 필요함
신뢰도 있는 요소	선형 위치제어가 필요함
누수와 방전이 없음	하중이 실리면 변형됨
저가이며 단순함	강성도가 매우 낮고 부정확한 응답성
압력이 유압보다 낮음	동력/무게비가 가장 낮음
부품 집기나 배치 작업에 유리	운전 비용이 고가임
순응적 시스템	압축성임
속도가 빠름	역학적 사용한계가 있다.
안정성이 있음	
저장이 용이함	

- 각 동력 공급장치의 경제성, 신뢰도, 하중 능력 비교

		전기 공급장치	유압 공급장치	공압 공급장치
경제성	사용 용이성	주변기가 갖추어진 검사도 용이	작동유의 관리, 관로의 플래싱, 필터 관리에 주의	공급 공기의 수분 제거, 윤활성의 부가에 주의를 요하지만 유압보다 용이
	수명	사이리스터의 솔리드스테이트 구동으로 개선됨	기름에 윤활성이 있어서 기기의 수명이 김	공기에 윤활성이 없어 유압, 전기에 비해 뒤떨어짐
	가격	보통	가격, 운영비 높음	저가격
신뢰성	속응성	(중간) 저관성 서보 전동기의 개발도 좋아지고 중소 출력인 것에서는 유압에 가까워짐	(크다) 토크-관성비가 크고 고속 응답을 쉽게 얻음	(작다) 고속 응답이 곤란. 단, 배관계의 손실이 작아서 단순한 동작일 때 유압보다 빠름
	안전성	과부하에 약함 방폭을 생각할 필요가 있음 기타 안전성이 높음	상당히 발열함 과부하에 강함 화재의 위험이 있음	과부하에 가장 강함 발열은 없음 인체에 위험이 적음
하중 능력	조작력	(소)에서 (중) 정도의 조작력. 통상 회전력으로 얻음	대단히 큰 힘을 얻음 회전력으로 직선 운동력도 얻음	큰 힘은 없다 통상 작동 운동력으로 얻음
	크기/중량	프린트 모터 등에 의해 상당히 개선되었고, 넓은 범위의 사이즈를 얻음	중량, 크기/출력이 아주 높음 유압 파워유닛이 상당한 공간을 차지	유압에 비해 뒤떨어짐 소형, 저출력인 것에 이용 가치가 있음

③ 4가지의 경로 제어 및 응용의 관점에서 비교

1) 제한 시퀀스 제어(limited-sequence controller)

시퀀스 제어란 미리 정해진 순서에 따라 제어의 각 단계를 하나씩 수행해 나가게 하는 제어 방식이다. 시퀀스 제어에서는 다음 단계에서 이루어져야 할 제어 동작이 미리 정해져 있다. 그러나 현 단계에서 제어 동작을 완료한 후 다음 동작으로 넘어갈 때에 현 단계의 제어 결과에 따라 다음 동작을 결정해서 다음 단계로 넘어가는 경우도 있다. 시퀀스 제어는 가장 널리 보급되어 있는 작업순서 제어기술이다.

로봇을 시동시키고 작업할 수 있는 상태로 준비하는 것을 원점조정(calibration)이라고 한다. 로봇의 각 부분이 작업을 할 수 있는 상태로 안정된 후에 작업 요구가 있을 때까지 대기한다. 대기 상태에서부터 시퀀스 제어가 필요하게 된다. 시동 직후에 바로 작업할 수 있는 안정된 상태가 되어야 하는데, 그러지 못할 때 외부에서 작업 요구가 왔을 때에는 작업 불능이란 신호를 외부측에 보내어 기다리게 하고, 로봇의 동력원에서부터 모든 부분의 상태를 검사해서 작업 가능한 상태가 된 뒤에 다시 외부로 작업 가능의 신호를 보낸다. 이것이 시퀀스 제어 시스템의 스텝 지연이다.

그 다음 순서는 작업이다. 외부에서 요구한 내용의 작업을 수행할 때 요구된 내용의 작업을 요구된 횟수만큼 반복 작업하기 위해서는 시퀀스 제어 시스템의 조합 논리 회로와 내장된 프로그램이 필요하다. 작업 공간 내에 위치한 여러 개의 작업 점을 순차적으로 작업해 나가는 PTP제어 방식과 작업 공간 내에 설정한 연속적인 경로 중의 하나를 따라서 작업하는 CP 제어 방식이 있다.

각 동작의 자유도에 대해서 작업 점의 위치에 대응하는 값이 설정되고, 그 설정치가 서보기구에 주어진다. 작업 점의 위치는 로봇이 사용하고 있는 좌표계에서의 위치를 의미한다.



로봇의 팔과 손이 위치 결정점에 도달한 것을 확인하기 위하여 접촉 정지 센서를 그곳에 넣고, 그 센서의 신호로 위치 결정 시퀀스의 완료를 확인한 뒤 다음 작업으로 넘어간다.

엄밀히 정의하자면 작업 점과 위치 결정점이 항상 동일하다고는 할 수 없다. 시퀀스 제어 시스템을 설계하는 데 있어서 플로 차트와 시퀀스 차트의 표현에 익숙해야 한다. 플로 차트는 작업의 흐름을 순서에 따라 그림 부호와 간단한 요약 설명으로 만든 것으로서, 각 부분 마다의 동작 상태를 세밀하게 표현하는 것이 아니라 전체의 작업 진행을 한 눈에 알아볼 수 있도록 표현한 것이다. 시퀀스 차트는 전체가 아니라 각 부분마다의 세부적인 작업을 면밀히 그림으로 설명하는 것이다. 따라서 어떤 작업에 대한 시퀀스 제어를 설계할 때에 이 두 가지 차트를 동시에 사용해야 한다.

제어계의 구체적 표기 단계에서는 제어 대상의 파악과 제어 작업의 구체적인 표현이 중요하다. 플로 차트와 시퀀스 차트에 의한 작업 순서 및 방법이 나중에 논리 회로로 이루어질 제어 장치의 구체적인 구성에 크게 영향을 미치므로, 항상 디버깅(debugging)에 신경을 써야 한다. 제어 대상을 파악하고 표현할 때에도 어떤 것을 입력 조건으로 해야 할지 등을 명확하게 제시해야 한다.

2) PTP 제어 (point-to-point control)

작업 공간 내에 흩어져 있는 예정된 작업 점들의 위치를 미리 정해진 순서대로 통과하게 하는 제어 방식이다. 고정된 순서로 유한 개의 작업 점을 통과하도록 지정하는 제어일 때는 순차적인 위치 결정 제어(sequential positioning control)라고도 부른다.

3) CP 제어(continuous path control)

작업 공간 내의 작업 점들을 통과하는 경로가 직선 또는 곡선으로 지정되어

있어서, 그 지정된 경로를 따라 연속적으로 위치 및 방향 결정을 하면서 작업을 하도록 하는 제어 방식이다. 통과 점들로 이루어진 모든 경로가 지정되어 있는 경로 제어이며, 보간 방식(interpolation)에 의한 의사 경로제어(pseudo-continuous path control)도 이에 포함된다.

4) 제어경로(controlled path control)

지점 간 제어가 더욱 정밀하게 제어되는 전문적인 제어 방법이다. 제어경로 로봇은 로봇이 2개의 학습된 지점들 사이에서 적합한 세그먼트를 기술한다. 로봇은 2개의 학습된 지점들 간을 직선을 따라 움직인다. 제어경로 로봇의 관절 액추에이터는 제어경로 제어기에 의하여 생성된 각 관절의 해당 속도로 움직이는 반면에, 지점간 비제어 로봇의 모든 관절 액추에이터는 동시에 별도로 움직인다.

※ 비교

PTP 제어와 CP 제어가 다른 점은, PTP 제어는 순서가 결정된 작업 점들을 차례대로 통과만 하면 된다는 것이다. 다시 말해서, 한 작업 점에서 다음 작업 점으로 갈 때 직선으로 가던 곡선으로 가던 상관없이 다음 작업 점에 도착만 하면 된다. 그러나 CP 제어의 경우는 작업 점 통과 순서뿐만 아니라 지나는 경로까지도 지정되어 있다. PTP 제어는 작업 점에서 작업이 있건 없건 일단 정지한다.

또한 통과 경로만 지정되는 CP 제어가 있고, 경로와 통과 시각이 지정되는 CP 제어가 있다. 후자에서는 앞에서 설명한 대로 속도 제어가 요구되며, 그에 따른 속도 서보계의 역할이 필요하다.

PTP 제어와 CP 제어 중에 어느 것이 필요한가를 알아보기 위해 예를 하나 들어보자. 어느 한 작업 점에서 작업을 완료한 뒤 다음 작업 점에서 작업을 하기 위해 이동해야 하는데, 이동을 하는 중간에 여러 개의 작업 점을 통과해야 한다.



그 작업 점들에서는 통과만 할 뿐 작업이 이루어지지 않는 것이다. 그런데 처음 시작 작업 점과 목적지 작업 점 사이에 장애물이 있어서 그것을 피해 가야 한다고 할 때, PTP 제어와 CP 제어 중에 어느 것을 선택해야 좋을지를 결정하는 문제이다.

이 경우에는 PTP 제어의 적용이 안전하다. CP 제어는 두 작업 점 사이에 최적의 경로를 만들려고 한다. 최적 경로란 가장 짧은 시간에 가장 짧은 거리를 갈 수 있도록 된 경로이다. 각 관절의 회전 각도, 팔 길이, 취할 수 있는 자세 등의 로봇 구조가 허용하는 한도 내에서 두 작업 점 간에 최적 경로를 만들려고 한다. 따라서 장애물이 있는 좁거나 구불구불한 공간을 지나가야 할 때에 최적 경로를 고집하므로 장애물과의 충돌을 일으키기가 쉽다.

PTP 제어는 짧게 짧게 끊어준 통과점들을 거의 직선 운동에 가까운 동작으로 움직일 수 있으므로 장애물을 교묘히 피해나갈 수가 있다. 그러나 PTP 제어의 한 가지 단점은 속도가 CP 제어에 비해 떨어지는 것이다. 통과점들을 지날 때 CP 제어는 연속 동작으로 지나가게 되지만, PTP 제어에서는 아주 짧은 시간이라는 하지만 일단 정지하므로 통과점 들의 수가 많을수록 통과 소요 시간이 비례적으로 길어진다.

산업 로봇에서 아마도 영원히 풀 수 없는 숙제는 속도와 정밀도 간의 조절이다. 정밀도를 높이다 보면 속도가 떨어지고, 속도를 높이려면 정밀도가 떨어지게 된다. 속도가 미치는 가장 큰 영향은 관성에 의해 생기는 탄력이며 그것이 위치 결정의 정밀도를 떨어뜨린다.

PTP 제어로 작업 점을 이동할 때에 한 개의 자유도씩 작동시키는 것과 여러 개의 복수 자유도를 동시에 작동시키는 것 두 가지 방법이 있다.

복수 자유도의 동작을 수행시키기 위해서는 여러 개의 액추에이터들을 동시에 작동시켜야 하므로 일시에 많은 동력이 소모된다. 동력원의 부담뿐 아니라 여러 개의 자유도를 동시에 제어하기도 쉽지 않으므로 동작 시간이 길어지더라도 1개의 자유도씩만으로 동작을 하는 방식이 많이 쓰여 왔다.

산업용 로봇의 단위 동작 속도는 빠르고 복합 동작은 느리다. CP 제어의 경우에는 직선 단위 동작만으로는 움직임이 이루어질 수가 없다. 따라서 각각의 자유도 사이의 상호 간섭도 경우에 따라서는 문제가 되며, 특히 동력원 측에 부담이 크다. 동력원 내부 임피던스의 영향에 의해서 공급 동력의 수준이 저하된다.

초기의 CP 제어 산업용 로봇에서는 경로 정보를 제어 장치 내부에 내장시키는 것이 어려웠다. 신뢰성 높은 메모리 칩들은 매우 고가였기 때문에 다소 신뢰성이 낮은 자기 테이프를 주로 사용했다. 때로는 신뢰성을 높이기 위해 코어 메모리를 사용하기도 했으나 그것 역시 작동 비용에 영향을 크게 미쳤다. 그러나 마이크로 프로세서의 발달로 제어 장치에 경로 정보를 내장하기 위한 비용이 저하되고 제어계의 신뢰도가 향상되어, CP 제어의 산업용 로봇 제작이 예전보다 훨씬 쉬어졌다.

PTP 제어 방식대로 위치 결정점(또는 작업점)들의 간격을 매우 작게 잘라서 정밀도를 높게 유지하면서도 CP 제어의 기능을 구현하려고 하는 것이 의사(pseudo) CP 제어의 목적이다. 모든 동작 자유도가 동시에 작동하는 PTP 제어에서 작업 점의 간격을 짧게 자르고, 또한 시작과 끝을 제외한 모든 작업 점에서 일시 정지하지 않아도 된다고 하면, 그것이 바로 의사 CP 제어이다. PTP 제어의 작업 점들을 결정할 때에 일정한 시간 간격마다의 위치로 하는 방법도 있다.

이산적인 작업 점들을 끊어진 동작으로 제어하는 PTP 제어의 경우에도 필요에 따라서 작업 점들 사이를 연속적인 동작으로 움직이게 하는 CP 제어의 성격을 가져야 한다. 그것을 위해 의사 CP 제어가 필요한 것이다. 예를 들어 단순히 목표물을 손으로 잡아서 들고 이동하여 원하는 위치에 내려놓는 산업용 로봇의 동작(pick-and-pick)을 생각해 보자.

PTP 제어로서는 파악(잡) - 상승 - 이동 - 하강 - 개방(뺐)의 다섯 단계로 시간이 많이 걸리지만, CP 제어를 도입하면 파악 - 원호 궤도의 이동 - 개방의



매끄러운(smooth) 동작 세 단계로 고속화될 수 있다. 의사 CP 제어가 발전됨으로서 PTP 제어와 CP 제어의 구별이 점점 모호해지고 있으며, 양 제어의 성격을 모두 갖춘 의사 CP 제어가 일반적으로 작업점들 간의 경로 제어를 대표하고 있다.



단원 학습 정리



- 1 전기공압 제어방식에 의한 전기공압회로 설계 방식에는 스테퍼 (stepper)방식에 의한 회로 설계 방식과 캐스케이드(cascade)방식에 의한 회로 설계 방식이 있다.
- 2 전기공압회로 설계시 부가조건에는 단동 및 연속 사이클 선택, 비상 정지 기능, 자동 및 수동 작업 선택 기능이 있다.
- 3 PLC 제어방식에 의한 제어회로 설계순서는 작업정의, 작업내용기술, PLC 기종 선정 및 입출력 할당표 작성, 배선 장비 설치 및 프로그래밍, 프로그램 입력 및 시운전의 순서로 진행 된다.
- 4 89C51이란 Intel사의 MCS-51 Family에 속하는 80XX 시리즈 중 가장 기본적인 프로세서라고 할 수 있다. 자동화기기의 핵심 부품으로서 산업체 등에서 널리 사용되고 있는 프로세서이다. 89C51의 경우 내부에 4Kbyte의 플래시 롬이 들어있어 초보자들이 사용하기 상당히 편리하기 때문에 널리 활용되고 있다.



단원 종합 문제

- ▶ **1** 스텝 방식에 의한 양솔레노이드 변위단계선도에서 각 단계의 릴레이가 ON되는 조건의 공식을 써보시오.
- ▶ **2** A, B 두 개의 공압실린더가 차례대로 $A+ \rightarrow B+ \rightarrow A- \rightarrow B-$ 의 순으로 동작되는 전기공압 제어회로를 설계해보시오.
- ▶ **3** 위와 같은 조건의 PLC 제어회로를 설계하여 이를 래더 다이어그램으로 구현해 보시오.
- ▶ **4** 인터럽트란 무엇인지 설명하시오.
- ▶ **5** 다음이 설명하는 것은?
 마이크로프로세서의 동작 중에서 어떠한 데이터가 마이크로프로세서를 거치지 않고 직접 메모리를 통해 사용되어짐으로서 속도의 향상을 가져올 수 있는 기능이다.
 ① DMA ② Timer/Counter ③ Interrupt ④ Reset
- ▶ **6** 함수의 형태를 나타낼 때 되돌려줄 결과값이 아무것도 없을 때 사용하는 예약어는 무엇입니까?
 ① int ② char ③ void ④ unsigned

- ▶ **7** 배열을 선언할 때 배열의 크기를 나타내는 문자는 무엇입니까?
 ① { 와 } ② (와) ③ [와] ④ “ 와 ”
- ▶ **8** 1부터 10까지의 수를 합을 구하면서 더해지는 과정을 모두 LED에 표시하는 프로그램을 작성하십시오. 이때 사용자 정의 함수를 사용하여 되돌려받은 결과값으로 출력하십시오.
- ▶ **9** 두 개의 변수에 각각 값을 넣어 사용자 정의 함수에서 사칙 연산을 수행한 뒤 결과값을 되돌려 받아 1초에 한번씩 덧셈 결과부터 차례대로 결과를 LED에 표시하는 프로그램을 작성하십시오.
- ▶ **10** 변수 x의 값이 34일 때, 상위 LED에는 첫 번째 자리의 수 '3' 을 표시하고, 하위 LED에는 두 번째 자리의 수 '4' 를 표시하십시오.
- ▶ **11** 다음 중에서 직교좌표로봇의 장점이 아닌 것은?
 ① 좌우 운동축을 따라 주행하므로 넓은 작업 영역을 얻음
 ② 간단한 제어로 직선이동이 가능
 ③ 높은 기계적 강성, 정확도와 정밀도
 ④ 온라인 프로그래밍이 용이
- ▶ **12** 구좌표형 로봇의 단점으로 지적되어지고 있는 것은 무엇인가?
 ① 직교형, 원통형보다 더 큰 작업영역
 ② 좋은 적재 하중 능력

- ③ 작업영역의 중심부는 작업 불가능한 영역
- ④ 로봇의 끝점이 지정된 위치에 직접도달

▶ **13** 산업용 로봇중 가장 유연한 형상을 가지고 있으며 최소설치공간을 차지하고 있음에도 큰 작업영역을 가지고 있어 가장 널리 사용되어지고 있는 로봇은 어떤 로봇인가?

- ① 직교 좌표형 로봇
- ② 수평 관절형 로봇
- ③ 수직 관절형 로봇
- ④ 원통 좌표형 로봇

○ 찾아보기

ㄱ

가감속기	95
개루프제어	48
검출용스위치	52
고정형 로봇	141
공기 베어링	109
공압공급장치	282
관절.....	42, 140
광전 스위치	156
구름 베어링	102
그리퍼	142
기구학	30
기어	89

ㄴ

논리회로	59
능동형 센서	151

ㄷ

다리식	44
다이어프램	153, 154
대수방정식	72
동기구학	32
동역학	34
되먹임	47
디버깅	138

ㄹ

래더다이아그램	72
레이디얼	102
레이디얼베어링.....	110
리드	81
리밋스위치	52
리벳	87

ㄴ

메모리부	67
메인스위치	217
모듈	91
모션 컨트롤러	144
모터	55
문서화	220
미끄럼 베어링	102

ㄷ

배열	190, 194
번지지정회로	237
베어링.....	101
벨로스	149
변수	179
변위단계선도	209
병진형 핸드	124
불나사	100
불베어링	102
분해능	83

비상정지 218
 비접촉형 151

스

서보 드라이버 144
 센서 150
 솔레노이드 55
 수동형 센서 151
 수평 다관절로봇 118
 순기구학 31
 순서도 182
 스러스트 저널 103
 스투드 보드 139
 스윙형 핸드 124
 스카라로봇 118
 스테퍼방식 209
 스포로킷 98
 스프링와셔 86
 시간지연회로 63

오

아이들러 99
 암 41
 액추에이터 46, 47
 엔드 이펙터 142
 엔코더 80
 역기구학 31
 오실레이터 232
 완적기구 40

용량형 158
 우선회로 62
 원주 피치 91
 유니버설 핸드 123, 126
 유도형 158
 유압공급장치 281
 이동기구 40
 이동형 로봇 141
 인터록회로 63
 인터페이스 138
 입력부 68

즈

자기유지회로 61
 자유도 41
 자이로 센서 162
 저널 102, 98
 전기공급장치 280
 전자밸브 56
 접촉형 151
 제어경로 286
 제어문 184
 제어신호 233
 제어용릴레이 53
 제한시퀀스 제어 284
 조합형 제어기 144
 지능형 로봇 11
 지름피치 90
 직교좌표로봇 116

직선가이드 108

츠

차륜형 43

초음파 159

출력부 69

크

카운터 58

캐스케이드 방식 214

커플링 108, 113

컴파일러 179

컴플라이언스 128

코리오리스 163

크롤러형 44

클러치 113

트

타이머 54

타이밍벨트 98

타임차트 72

탭 85

텔레스코픽 압 121

통신회로 236

투광기 152

팬더그래프 120

편 솔레노이드 밸브 212

평벨트 98

페루프제어 48

포인터 194

포트 227

푸시버튼스위치 51

프로그래머 71, 220

프로세서 67

중

한시동작타이머 54

한시복귀타이머 55

할당표 220

함수 190

핸드 41

영문

ALE/PROG 228

CP 제어 285

CPU 168

EA 229

IFR 12

MPU 168

PLC 49

PSEN 229

PTP 제어 285

RESET 228

XTAL 230

▶▶참고문헌◀◀

1. 메카트로닉스 개론, 진상호, 박정규 공저, 북두출판사, 2005.
2. 한국사전연구소, 기계공학 편람사전, 1997.
3. 공유압일반, 산업인력공단, 2002.
4. 자동화 시스템제어, 산업인력공단, 2002.
5. 전기공압과 PLC 제어기초, 이상준, 양호언, 오병욱 공저, 남두도서 2005.
6. 자동화를 위한 센서공학 김준식, 김원희 공저 성안당 2005.
7. 로봇공학, 김정하, 염영일 공역, 사이텍미디어, 2002.
8. 마이크로 로봇바이블, 윤지녕, 성안당, 2000.
9. 메카트로닉스 개론, 진상호, 박정규 공저, 북두출판사, 2003.
10. 로봇 스테디I, 이재창, 심광렬, 송경화 공저, 동일출판사, 2003.
11. 마이크로 로봇, 정기철, 문장현 공저, 북두출판사, 2004.
12. 지능형 로봇 구조 및 응용, 남상엽외 2인, 상학당, 2006.