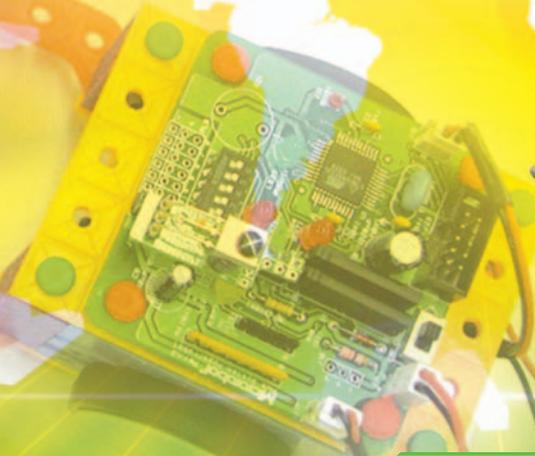


로봇제작

고등학교

서울특별시교육감 인정

2007-000 | 2007. 02 .



로봇 제작

김경채·김진오·장종문·김흥락 저

서울특별시교육청

머 . 리 . 말 .



로봇 산업은 대한민국 차세대 및 차차세대 동력산업으로서, 미래 산업의 주역으로 부각되고 있다. 첨단 산업과 관련한 각종 국제 대회 및 행사에서 로봇 제품들이 여타 다른 IT제품들을 제치고 대한민국 대표상품으로 등극한 것은 그 좋은 본보기라 하겠다. 현재 우리나라의 로봇 산업 현황을 보면 제조업 로봇 분야는 대기업이, 서비스 로봇 분야는 중소, 벤처 기업이 주도하고 있다. 이러한 상황에서 우리가 이미 보유하고 있는 선진 IT와 대기업, 중소, 벤처 기업에서 보유하고 있는 로봇기술을 결합시키면 로봇의 부품생산이나 제작 면에서 충분한 경쟁력을 지닐 수 있을 것이라 전망된다.

또한 정부에서는 로봇 산업의 발전에 따라 7년 이내에 현재의 10배에 해당하는 로봇 인력을 양성한다는 계획을 수립하고 있다. 이러한 사업 계획의 일환으로 로봇 산업에 이바지할 인력 양성에 큰 역량을 쏟고있다. 따라서 본 <로봇 제작> 교과서에서는 로봇의 기초가 되는 직교로봇과 스카라로봇의 동작 원리와 주요 부품의 구성과 이해, 로봇제작에 따른 계획 수립, 하드웨어 설계와 소프트웨어 설계 등을 학습할 수 있도록 하였다. 이러한 과정을 거쳐 이동로봇과 창작로봇을 제작할 수 있도록 함으로써 학습자 스스로 로봇 제작에 대한 지식과 실력을 갖출 수 있도록 구성하였다.

1단원은 로봇에 대한 전반적인 이해를 도울 수 있도록 구성하였으며,

2, 3단원은 로봇 제작에 필요한 기본적인 요소를 설명하고 있다.

4단원은 직교로봇의 제작을 위한 설계 방법 및 제작 실습을 다루고 있으며,

5단원과 6단원은 각각 SCARA 로봇과 이동로봇의 제작에 대한 내용을 익힐 수 있도록 하였다.

마지막 7단원은 위에서 배운 내용을 기반으로 한 창작로봇을 제작하는 방법을 익힐 수 있도록 구성하였다.

끝으로, 본 교과서를 학습하여 로봇에 대한 기본적인 개념을 이해하고, 로봇 제작을 할 수 있는 실력을 배양함으로써 로봇산업의 미래를 이끌어갈 일꾼이 되기를 바란다.



차례

contents

I. 로봇 개요

1. 로봇 역사 / 10
2. 로봇 분류 / 13
3. 로봇의 미래 / 18

II. 로봇 제작

1. 제작 로봇 선정 및 진행 일정 수립 / 24
2. 현재 나와 있는 로봇 조사 / 25
3. 로봇 기구부 설계 / 26
4. 로봇 회로 설계 및 부품 선정 / 28
5. 로봇 조립 및 전장 / 34
6. 로봇 소프트웨어 설계 / 34
7. 로봇 구동 실험 및 결과 보고 / 35

III. 로봇 제작 기초

1. 로봇의 구성 이해 / 40
2. 로봇의 동작원리 이해 / 49
3. 로봇의 주요 구성 부품 이해 / 62
4. 로봇 제작 계획 수립(Process) / 76
5. 로봇 H/W 설계 기초 / 83
6. 로봇 S/W 설계 기초 / 90

IV. 직교 로봇 제작

1. 직교 로봇 하드웨어 상세 설계 / 100
2. 직교 로봇 소프트웨어 상세 설계 / 101
3. 직교 로봇 조립 / 108
4. 직교 로봇 프로그래밍 / 111
5. 직교 로봇 구동시험 및 예제 실습 / 120



V. 스카라(SCARA) 로봇 제작

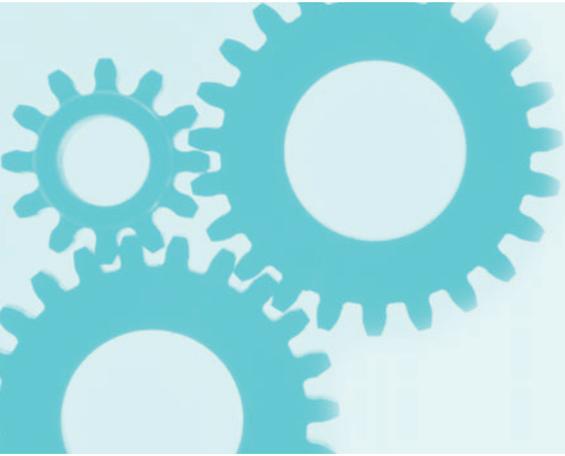
1. SCARA 로봇 하드웨어 설계 / 130
2. SCARA 로봇 소프트웨어 상세 설계 / 131
3. SCARA 로봇 조립 / 137
4. SCARA 로봇 프로그래밍 / 142

VI. 이동 로봇 제작

1. 이동 로봇 하드웨어 상세 설계 / 156
2. 이동 로봇 소프트웨어 상세 설계 / 163
3. 이동 로봇 조립 / 169
4. 이동 로봇 프로그래밍 / 186
5. 이동 로봇 구동 시험 및 예제 실습 / 198

VII. 창작 로봇 제작

1. 뱀 로봇 하드웨어 상세 설계 / 208
2. 뱀 로봇 소프트웨어 상세 설계 / 211
3. 뱀 로봇 조립 / 224
4. 뱀 로봇 구동실험 및 예제 실습 / 240



I

로봇 개요

로봇은 인간의 삶 주변에 한 층 가깝게 다가오고 있다. 청소용 로봇, 제조용 로봇, 엔터테인먼트 로봇 등의 제품들은 이미 제품으로 다양한 경로를 통하여 일반 판매되고 있다.

본 단원에서는 로봇에 대한 전체적인 이해를 돕기 위하여 로봇의 역사, 로봇의 분류 그리고 로봇의 미래 발전 방향에 대해서 학습한다.



학습목표

1. 로봇의 역사에 대해 이해할 수 있다.
2. 로봇의 분류에 대해 설명할 수 있다.
3. 로봇의 미래에 대해 설명할 수 있다.



I

Robot Robot

로봇 개요

1 로봇 개요

1 로봇 역사

사람이나 동물을 닮은 인조물에 대한 발상은 기원전부터 신화나 전설 등의 이야기 속에서 전해져 내려오고 있다. 사람들이 만든 상상 속의 인조 인간이나 인조 생물에 대한 꿈은 과학 기술이 발전하면서 기구나 장치들을 이용한 자동 인형으로 만들어졌고, 점차 실물로 등장하게 되었다.

기원전 1세기에는 그리스의 '박카스' 신전에 움직이는 조각상으로 '박카스' 신과 '여신상'이 촛대에 촛불이 켜짐과 동시에 회전하는 형태의 조각상이 만들어졌다.

13세기에는 점심 시간이 되면 날개를 펴고 시간을 알려 주는 '날개 펼치는 닭'이 있었고, 15세기에는 해가 뜨고 동시에 소리를 내는 '에티오피마오' 왕의 동상이 만들어졌다.

18세기 자동인형의 출현은 움직이는 정밀한 기계를 만들기 시작했다는 큰 의미가 있다. 자동 인형이 만들어지고 발달된 이유는 근대에 이르러 빠른 속도로 발달된 시계 기술의 덕분이다.

1737년 프랑스의 자크드 보캉송이 자동 인형 '오리'를 만들었다.

'오리'는 마치 살아있는 것처럼 수영을 하고 퍽퍽 소리를 내며, 물도 마시고,



공무니로 배설도 했다고 한다.

물론 당시의 지식 수준으로 내장기관을 만들어 소화나 배설과 같은 생화학적 작용까지 시뮬레이션할 수는 없었다. 나중에 알고 보니 속임수였다. 소화는 가짜였고, 오리의 배설물도 빵 부스러기를 푸른색으로 염색한 것이었다.

1773년 스위스의 드로스가 만든 ‘글쓰는 인형’도 태엽과 톱니 바퀴를 이용하여 머리와 손을 움직이고, 눈을 깜박거렸으며 종이 위에 글씨나 개의 그림을 그릴 수 있는 정교한 자동 인형이다.

유럽에서만 자동 인형을 만든 것은 아니었다. 일본에서도 시대(1603~1867)에 ‘카라쿠리’라는 자동인형을 만들었다. 그 중에서 대표적인 것이 차 나르는 인형이다. 주인이 태엽을 감아 인형의 손 위에 찻잔을 올려놓으면, 인형은 손님을 향해 나아간다. 손님이 찻잔을 들면 그 자리에 멈추고, 다 마신 찻잔을 손 위에 올려놓으면 뒤로 돌아 주인에게 되돌아온다.

로봇의 어원

로봇이란 용어는 체코의 극작가 카렐차펙이 1920년에 쓴 희곡「R.U.R.」(Rossum's Universal Robots)에서 처음 소개되었다.

로봇의 어원은 Robot로서 체코어로 강제적 노동 또는 노예를 뜻한다.



〈그림 I-1〉 자동 인형

산업 혁명 초기 증기의 힘을 이용하는 증기 엔진이 새로운 동력원으로 등장하였다.

1764년 영국의 방직공 제임스 하그리브스는 딸이 쓰러뜨린 방직 기계를 보고 물레바퀴가 수직으로 세워져도 여전히 작동한다는 사실을 알게 되었다. 많은

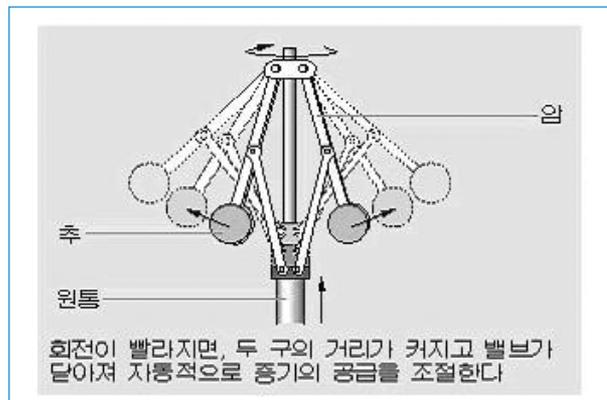
연구 끝에 하그리브스는 8개의 실을 동시에 짤 수 있는 기계를 만들었고, 자신의 딸의 이름을 붙여서 '실짜는 제니' 라고 명명했다.

과학 기술이 발전하면서, 곧 부분적으로 기계를 제어할 수 있는 제어기가 발명되었으며, 스코틀랜드의 제임스 와트가 '뜯은 공 조절기' 라는 제어기를 발명하였다.

엔진이 점점 빨리 돌면서 속도가 빨라지고 양쪽 옆의 공이 원심력으로 인해 점점 공중에 뜨게 된다.그런 다음 증기 밸브가 점점 조이게 되는 원리를 이용하였다. 엔진 속도가 느려지게 할 수 있는 제어 장치였던 것이다.

다시 엔진 속도가 느려지면 공은 천천히 내려오게 되고 엔진 속도를 빨라지게 하였다. 제임스 와트의 '뜯은 공 조절기' 의 제어 장치는 최초로 피드백을 할 수 있는 장치라 할 수 있다.

지금까지 살펴본 것처럼 로봇은 우연히 우리 곁에 다가온 것이 아니다. 전설과 우화 그리고 생활 속에서 생겨나 발전해 온 것이다. 고대에서 근대로, 근대에서 현재로 시간이 흐르면서 과학기술이 발전하듯 로봇도 함께 발전해 온 것이다.



<그림 I-2> 제임스 와트의 뜯은 공 조절기

2 로봇 분류

(1) 구동 방식(Drive Technologies)에 따른 로봇의 분류

1) 전기 구동 로봇(Electric-drive Robot)

구동 수단으로 전기 서보 모터(Servo Motor)나 스텝 모터(Step Motor)를 사용하는 로봇을 말한다.

2) 유압 구동 로봇(Hydraulic-drive Robot)

구동 수단으로 유압 장치(Hydraulic Equipments)를 사용하는 로봇을 말한다. 이 외에도 공압이 사용되는데 압축 공기의 압축성과 그로 인한 제어의 어려움 때문에 구동 장치보다는 엔드 이펙트, 특히 그립퍼 등에 제한적으로 사용된다. 전기 구동과 유압 구동 방식의 상대적 특성은 <표 I-1>과 같다.

단 계	전기구동	유압구동
구 조	간단하다	복잡하다
가 격	저렴하다	고가이다
출 력	소출력이다	대출력이다
청 정 도	깨끗하다	오염이 있다
안 전 성	과부하에 약함	과부하에 강함
응 답 성	보통(저관성 서보 개발로 양호해 지고 있음)	좋음(토크-관성비가 크다)
기 타	정확도가 높다	정확도가 높다
	반복성이 좋다	잡음이 있다
	관리에 편하다	유지, 보수가 필요
	설계가 쉽다	설치에 넓은 면적 필요
	저속 구동	고속 구동

<표 I-1> 동력원의 특징

스텝핑 모터

스텝모터, 펄스모터 등으로 불리며, 다른 모터에 비해 가격이 싸고 정확한 각도 제어에 유리하여 널리 쓰인다.

(2) 기하학적 작업 궤적(Work-envelope Geometries)에 따른 로봇의 분류

기하학적 작업 궤적 즉, 엔드 이펙터의 작동 궤적에 따라 직각 좌표계(Cartesian coordinate, Rectangular-Coordinate) 로봇, 원통 좌표계(Cylindrical-Coordinate) 로봇, 구면 좌표계(Spherical-Coordinate) 로봇, SCARA(Selective Compliance Assembly Robot Arm) 로봇, 다관절(Articulated, Revolute, Anthropomorphic) 로봇 등이 있다.

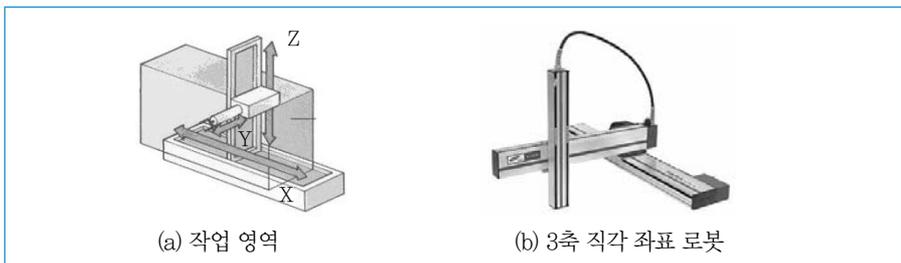
스카라(SCARA) 로봇을 수평 다관절 로봇이라 부르기도 하며, 특수한 경우의 원통 좌표계 로봇이다. 아티큘레이티드 로봇을 수직 다관절 로봇이라 부르기도 하며, 특수한 경우의 구면 좌표계 로봇이기도 하다. 또한 구면 좌표계 로봇을 극 좌표계 로봇이라 부르기도 한다

티칭 펜던트

로봇에게 명령을 내릴 수 있도록 하는 입력장치의 일종으로서, 텔레비전의 리모컨이나 구성조종차의 조정기와 유사한 역할을 한다.

1) 직각 좌표계 로봇

직각 좌표계 로봇을 직교 로봇이라 부르기도 하며, 개조용 로봇 중에서 가장 간단한 구조를 갖는 것으로 로봇의 구조적 동작 특성이 직각 좌표계를 이루기 때문에 XY로봇이라고도 불리운다. 직각 좌표계 로봇은 각 축들이 직선 운동을 하기 때문에 로봇의 작업 영역은 구성하는 자유도의 수에 따라 직선, 직사각형, 직육면체가 된다. 직각 좌표계 로봇은 그 구성에 있어 로봇 몸체 부분과 로봇 제어기 부분으로 구성되어 있으며 제어기에는 티칭 펜던트를 달아 로봇을 제어할 수 있다.



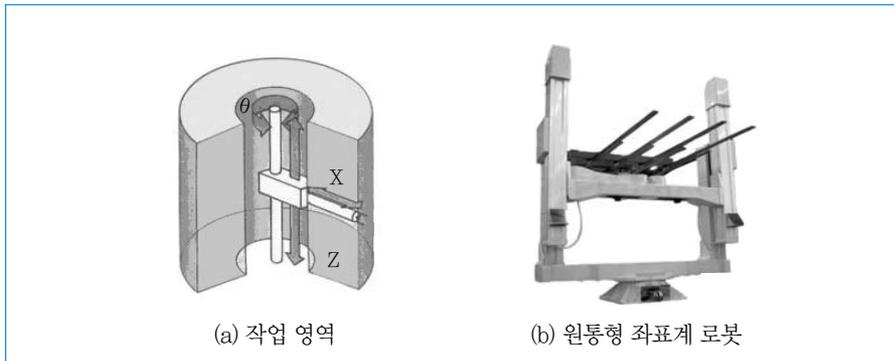
〈그림 I-3〉 직각 좌표계 로봇 및 작업 영역

2) 원통 좌표계 로봇

원통 좌표계 로봇은 엔드 이펙트의 동작 범위가 원통 모양을 가지므로 원통 좌표계 로봇이라 하며, 구조는 베이스에 필러(Pillar)가 있고 필러에 연결된 암이 상하 운동을 하고 암 자체는 암의 중심축 방향으로 직선운동을 하며, 암의 선단에 엔드 이펙트가 부착되어 있는 형식이다.

원통 좌표계 로봇은 신뢰성이 상대적으로 높아서 공작 대상물의 로딩과 언로딩에 많이 사용된다. 물론 팔레타이징(Palletizing)의 로딩과 언로딩에도 특성이 좋다. 또한 기계와 기계사이, 컨베이어의 물품을 입·출력 시키는 용도로도 사용된다.

〈그림 I-4〉은 원통 좌표계 로봇의 형상과 작업 영역을 보여준다.



필러(Pillar)

로봇의 바닥면과 로봇의 팔을 연결시켜 기능역할을 하는 부분이다.

〈그림 I-4〉 원통형 로봇 및 작업 영역

3) 극 좌표계 로봇

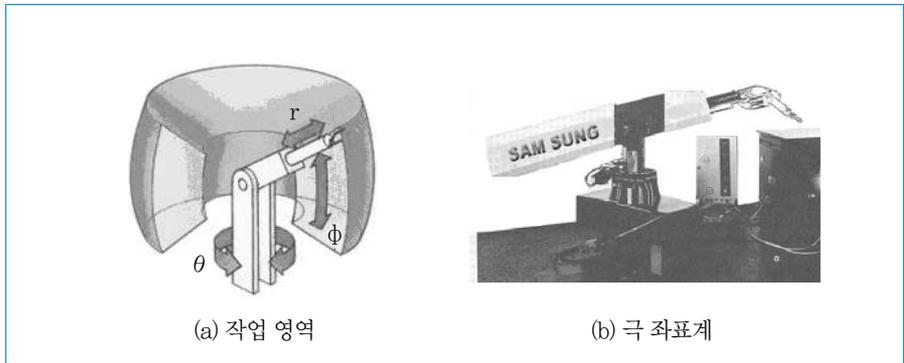
극 좌표계(Spherical-Coordinate) 로봇은 제조용 로봇의 최초 방식으로 오늘날 광범위하게 사용되고 있다. 최초의 실용 산업용 로봇인 유니메이션 사의 유니메이트도 이 방식이었다. 다만 이 로봇이 갖는 높은 강성과 신뢰성에도 불구하고 다관절 로봇에 비해 상대적으로 동작상의 유연성(Flexibility)이 떨어진다. 또한 직선 운동을 할 때 복잡 제어(Complex Control) 또는 오프라인 제어

팔레타이징

물체를 팔레트에 차곡차곡 쌓는 작업을 의미한다. 박스의 적재나 물체의 이동 등에 응용되는 작업이다.

(Off-line)가 요구된다.

이 로봇의 작업 영역은 부분적인 구면 궤적을 가지며 주로 스포트 용접, 팔레타이징(Palletizing), 중량물의 취급에 사용된다. <그림 I-5>는 극 좌표계 로봇의 형상과 작업 영역을 보여준다.



<그림 I-5> 극 좌표계 로봇 및 작업 영역

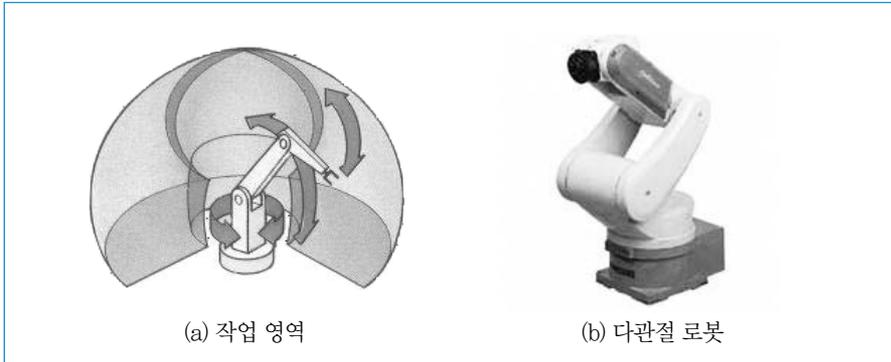
4) 다관절 로봇

다관절(Articulated) 로봇은 회전(Revolute) 로봇, 인간 유사(Anthropomorphic) 로봇, 수평축 암 결합(Jointed-Arm Horizontal-Axes) 로봇 등으로 불리며, 국내에서는 SCARA 로봇과 상대적인 의미로 수직 다관절 로봇으로 불리기도 한다. 지면에 수직으로 서있는 회전 허리에 2~3개의 암이 수평축으로 연결되어 있다.

인간의 팔과 가장 많이 닮은 형상을 하고 있으며, 동작도 유사하다. 그러므로 엔드 이펙트의 동작도 가장 다양하게 구현할 수 있어 여러 가지 작업에 사용되고 있다. 예를 들면 도색(Paint spraying), 용접(Welding), 접착 작업(Gluing), 조립(Assembly), 중량물 취급(Heavy Materials Handling)등이다.

이 로봇에서 직선 운동을 하기 위해서는 여러 개의 축들이 동시에 움직여야 하므로 극 좌표계 로봇에서와 마찬가지로 복합 제어 프로그램이 필요하다. <그

림 I-6)은 다관절 로봇의 형상과 작업 영역을 보여준다.



〈그림 I-6〉 다관절 로봇 및 작업 영역

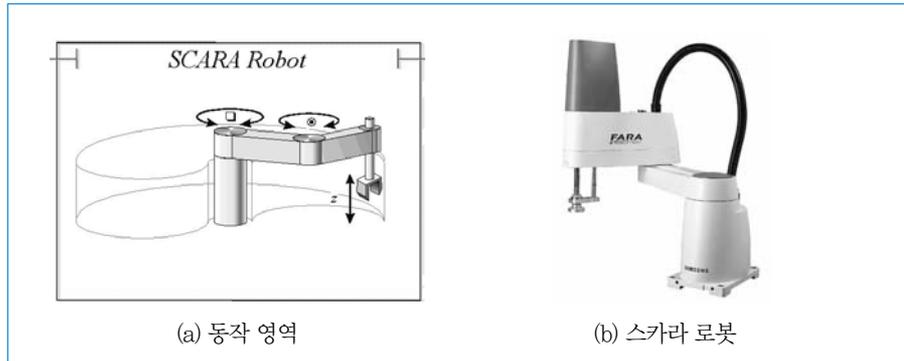
5) 스카라 로봇

스카라(SCARA: Selective Compliance Assembly Robot Arm: 선택적 유연 조립 로봇 팔) 로봇은 다관절 로봇과 달리 조인트 부분의 축 중심선이 지면에 수직으로 되어 있어 수직축 암 결합(Jointed-Arm Vertical-Axis) 로봇 또는 암들의 동작이 수평으로 이루어져 수평 다관절 로봇이라 부르기도 한다.

주로 전자제품 조립에 사용하며 1979년 일본의 야마니시 대학에서 최초로 개발하였다. 직교 로봇에 비해 설치 면적에 대한 동작 범위가 크고 동작 속도가 빠르다는 특징을 갖고 있다. 스카라 로봇에 사용되는 모터는 대부분 AC서보 모터이며 인코더는 절대값 인코더(Absolute Encoder)를 많이 사용한다. 왜냐하면 DC서보 모터와는 달리 브러시(Brush)의 교환이 없어 유지 관리가 편하고, 모터 구동에 따른 분진의 발생이 적어 전자제품 조립 현장의 특성에 유리하기 때문이다. 또한 전원이 끊어졌을 때 원점 복귀를 할 필요가 없기에 절대값 인코더를 사용한다

인코더

모터의 회전속도 및 회전 위치를 파악하기 위해서 모터에 부착하는 장치이다. 인토더를 이용하면 DC/AC 모터를 원하는 속도 및 회전량만큼 정확히 제어할 수 있다.



〈그림 1-7〉 스카라 로봇 및 작업 영역

3 로봇의 미래

로봇은 인간을 대신해서 일(work, task)을 하는, 인간이 창조한 일종의 기계이다. 그러나 카렐 차פק이 지적했듯이 로봇에 의해 인간이 피해를 당하는 경우도 가정할 수 있다. 그렇다면 미래의 로봇은 어떤 방향으로 발전할 것인가?

국내의 산업용 로봇시장 규모는 전세계의 약 6위이다. 통계치를 차치하고라도 우리 주변에서 이제 로봇은 낯선 존재가 아니다. 제조용 로봇뿐만 아니라 일상 생활에도 로봇들이 등장하고 있다. 완구의 성격을 지나면서도 기술적으로는 상당한 수준의 로봇들도 출하되고 있다.

전문가들은 일부 산업현장에서만 활약해 오던 로봇이 곧 회사, 병원, 학교, 가정 등에까지 보급될 것이라 믿고 있다. 이러한 예상은 최근 일본의 혼다에서 키120cm, 몸무게 43kg의 ASIMO(Advanced Step in Innovative Mobility)라는 인간과 같이 직립보행하고 자연스럽게 방향을 바꿀 수 있는 로봇을 출시한 기술적 발전에서 가시화되고 있다. 그 외에도 소니의 SDR(Sony Dream Robot)-3X는 키50cm, 몸무게 5kg으로 색을 구별하고, 춤까지 추는 로봇을 발표했다. 가격은 자동차 한 대값이라 한다.

결국 언젠가는 80여년 전에 카렐 차팩이 발표했던 희곡의 내용처럼 될 것인



가? 여기서 우리는 러시아 출신 과학자, 정확히는 화학자 아시모프의 1950년의 소설 “나는 로봇”의 내용에 주목하게 된다. 이 소설에서 아시모프는 로봇의 3원칙을 발표했다. 로봇이 인간을 해치는 것을 막기 위해 로봇의 성격을 규정한 것이었다. 그 내용은 다음과 같다.

제1조 - 로봇은 인간에게 해를 주어서는 안 된다. 또 인간의 위험을 간과해 버림으로써 인간에 위해가 미치게 해서는 안 된다.

제2조 - 로봇은 인간의 명령에 따르지 않으면 안 된다. 그러나 그 명령이 제1조에 어긋날 때에는 따르지 않아도 좋다.

제3조 - 로봇은 제1조와 제2조에 위배하는 염려가 없는 한 자기를 지키지 않으면 안 된다.

결국 컴퓨터나 기타 기술의 발달로 로봇도 빠른 속도로 발전할 것이다. 따라서 휴머노이드(Humanoid) 로봇이 등장하게 될 것이다. 로봇의 발전은 결국 메카트로닉스의 발전에 의존하게 될 것이고 메카트로닉스는 기계 분야, 전기전자 분야, 컴퓨터 분야의 결합이므로 이 분야들의 미래를 예측함으로써 로봇의 미래도 충분히 예측할 수 있을 것이다. 다만 아시모프의 선언을 잊지는 말아야 할 것이다.

휴머노이드

휴머노이드는 ‘외모가 인간처럼 생겼다.’ 따라서 로봇뿐만이 아니라 외계인이나 기타 정체불명의 어떤 것이든간에 겉모습이 사람처럼 두 팔, 두 다리가 있다면 ‘휴머노이드’ 타입이라고 말한다.

단원 학습 정리



- 1** 전기 구동 로봇(Electric-Drive Robot)은 구동 수단으로 전기 서보 모터(Servo Motor)나 스텝핑 모터(Stepping Motor)를 사용하는 한다.
- 2** 기하학적 작업 궤적은 직각 좌표계(Cartesian coordinate, Rectangular-Coordinate)로봇, 원통 좌표계(Cylindrical-coordinate)로봇, 구면 좌표계(Spherical-Coordinate) 로봇, 스카라(Selective Compliance Assembly Robot Arm) 로봇, 다관절(Articulated, Revolute, Anthropomorphic) 로봇 등이 있다.
- 3** 직각 좌표계 로봇을 직교 로봇이라 부르기도 하며, 제조용 로봇 중에서 가장 간단한 구조를 갖는 것으로 로봇의 구조적 동작 특성이 직각 좌표계를 이루기 때문에 XY 로봇이라고도 한다.
- 4** 원통 좌표계 구조는 베이스에 필러(Pillar)가 있고 필러에 연결된 암이 상하 운동을 하고 암 자체는 암의 중심축 방향으로 직선운동을 하며, 암의 선단에 엔드 이펙트가 부착되어 있는 형식이다.
- 5** 극 좌표계(Spherical-Coordinate) 로봇은 제조용 로봇의 최초 방식으로 오늘날에 광범위하게 사용되고 있다.

6 다관절(Articulated) 로봇은 회전(Revolute) 로봇, 인간 유사(Anthropomorphic) 로봇, 수평축 암 결합(Jointed-Arm Horizontal-Axes) 로봇 등으로 불리며, 국내에서는 스카라(SCARA: Selective Compliance Assembly Robot Arm: 선택적 유연 조립 로봇 팔)로봇과 상대적인 의미로 수직 다관절 로봇으로 불리기도 한다. 지면에 수직으로 서 있는 회전 허리에 2~3개의 암이 수평축으로 연결되어 있다.

7 스카라(SCARA: Selective Compliance Assembly Robot Arm: 선택적 유연 조립 로봇 팔)로봇은 다관절 로봇과 달리 조인트 부분의 축 중심선이 지면에 수직으로 되어 있어 수직축 암결합(Jointed-Arm Vertical-Axis) 로봇 또는 암들의 동작이 수평으로 이루어져 수평 다관절 로봇이라 부르기도 한다.





II

로봇 제작

로봇을 제작하는 일은 많은 인력과 기술, 그리고 자본을 필요로 한다. 그리고, 정확한 과정과 체계를 요구한다. 본 단원에서는 로봇 제작 과정 전체를 이해하고, 로봇 제작에 필요한 하드웨어와 소프트웨어 등의 전반적인 내용을 학습한다.



학습목표

1. 로봇 제작 과정을 설명할 수 있다.
2. 로봇 제작에 필요한 부품을 설명할 수 있다.
3. 로봇을 동작시킬 수 있는 소프트웨어를 설명할 수 있다.

II

Robot Robot

로봇 제작 과정

배틀로봇

싸움을 위한 로봇으로서 각각의 로봇은 상대방을 공격할 수 있는 무기를 지니고 있으며, 원격조정으로 로봇의 움직임을 제어할 수 있도록 되어 있다.

로봇을 제작하기 위해서는 여러 가지 과정을 거쳐야 한다.
아래 과정을 수행하면서 로봇에 필요한 과정을 배울 수 있다.

1 제작 로봇 선정 및 진행 일정 수립

로봇을 제작하기 위해서 여러 형태의 로봇 중 제작하고자 하는 로봇을 선정해야 된다.

바퀴형 로봇으로는 라인트레이서, 마이크로 마우스, 배틀 로봇이 있고, 관절형 로봇으로는 4족 로봇, 6족 로봇, 뱀 로봇 등이 있다.



라인트레이서

마이크로 마우스

배틀 로봇

〈그림 II-1〉 바퀴형 로봇



4족 로봇

6족 로봇

뱀 로봇

〈그림 II-2〉 관절형 로봇

다양한 로봇 형태 중 선정된 로봇을 제작하기 위해서는 팀원간에 역할 분담과 로봇 일정 관리가 필요하다.

아래 〈그림 II-3〉처럼 해야 할 일을 열거하고 각 수행 단계에 맞게 수행 일정을 정하면 된다.

단계별 수행작업	수행 일정						
	10/1 주	10/2 주	10/3 주	10/4 주	11/1 주	11/2 주	11/3 주
제작 일정 계획	■						
기구부 설계		■	■				
회로 설계			■	■			
프로그램 코딩 및 디버깅				■	■	■	
구동 및 결과 보고							■

〈그림 II-3〉 수행 일정 계획 예제

2 현재 나와 있는 로봇 조사

로봇 제작 과정에서 좀더 효과적으로 과정을 수행하기 위해서는 현재 제작되어 나와 있는 로봇을 조사하면 각 로봇의 장점 및 제작하고자 하는 로봇의 형태

를 계획하는 데 많은 도움이 될 뿐만 아니라 제작 시간 및 실수를 줄일 수 있다.

〈표 II-1〉는 현재 국내 및 해외 로봇 관련 사이트이다. 아래의 주소 외 많은 로봇 관련 카페나 블로그를 통해서 설계하기 전에 정보를 수집하고 수집된 정보에 대해 토의해 보자.

	협회 및 대회	사이트 주소
1	ROBO-ONE KOREA	http://www.robo-one.or.kr/
2	일본 로보원 위원회	http://www.robo-one.com/
3	해외 배틀 로봇 홈페이지	http://www.robotcombat.com/
4	한국 마이크로로봇 경연대회	http://microrobot.snu.ac.kr/
5	전국 라인트레이서 로봇 경연 대회	http://zetin.uos.ac.kr/
6	대한 로봇축구 협회	http://www.krso.org/
7	국제 로봇 올림피아드	http://www.iroc.org/

〈표 II-1〉 로봇 협회 및 대회

③ 로봇 기구부 설계

위 단계는 로봇 제작을 위한 사전 조사로 제작 로봇을 선정하고 구성원간의 일 분담 및 제작 일정을 정한다. 기존에 나와 있는 제품을 통해서 제작할 로봇을 구상했다면 지금부터는 실제로 로봇의 기구부 설계를 하기 위한 단계이다.

로봇 하드웨어를 설계 하기 위해서는 다양한 CAD/CAM 프로그램을 이용해 제작이 가능하다.

CAD는 Computer Aided Design의 약칭으로 컴퓨터를 이용하여 설계하는 것을 뜻하며, CAM은 Computer Aided Manufacturing의 약칭으로 컴퓨터를 이용하여 생산하는 것을 뜻한다.

CAD는 도면을 제도대 위에서 1장씩 그리는 것이 아니라, 데이터베이스의 정

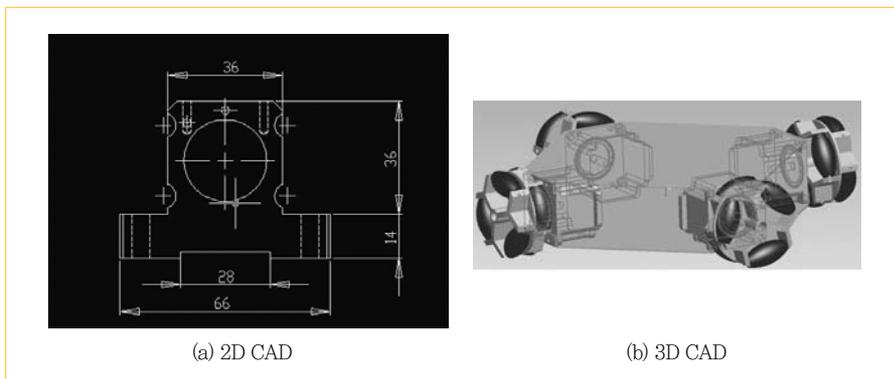


보를 컴퓨터 모니터를 보고 합성하면서 설계하기 때문에 작업의 생산성 및 고속화 등이 가능하다.

그리고 CAD에 의하여 설계된 내용은 바로 CAM으로 연결되고, CAM을 통해 NC(수치제어) 공작기계에 정확한 작업 동작 지시를 하게 되며 생산, 가공, 조립, 검사 등의 제조 과정을 컴퓨터로 관리하여 작업의 신속성과 제품의 정밀성을 기하게 된다. 근래에는 설계 도면을 입체적으로 나타낼 수 있는 3차원 CAD·CAM 시스템이 개발되어, 입체 형상을 컴퓨터 화면에 완벽하게 재현시켜 줄 뿐만 아니라, 그 대상 입체의 겉넓이·부피·무게·강도 등 물리적 성질까지 자동으로 계산해서 적합한 형태로 설계해 준다.

형식	이름	용도
2D	Autocad , uniCAD	산업은 전범위
3D	Unigraphics, Catia, I-DEAS, PRO/E	항공기, 자동차 등
	Solidworks, 인벤트	단품 및 전자 제품 등

〈표 II-2〉 CAD 종류



〈그림 II-4〉 2D 및 3D CAD 예제

4 로봇 회로 설계 및 부품 선정

하드웨어 설계가 끝나면 로봇을 제어하기 위한 회로 설계 및 전자 부품 선정을 해야 된다.

로봇 제어에 필요한 부품으로는 머리에 해당하는 마이크로 프로세서, 손과 발에 해당되는 모터, 음식에 해당되는 배터리와 감각 기관에 해당하는 센서, 원거리 제어를 위한 장치 등으로 구성된다.

1) 마이크로 프로세서

마이크로 프로세서는 CISC와 RISC 형식으로 분류 된다.

CISC란 명령어의 갯수가 많고 복잡하며, 각 명령어들마다 실행되는 사이클 수도 틀리고 복잡한 만큼 실행 속도도 RISC 타입보다는 느리지만 호환성이 좋다.

- Z-80: 미국 자일로그 사가 제조하는 8비트 마이크로 프로세서로, 인텔의 8080 마이크로 프로세서의 개량형이라고 할 수 있는 제품이다. 80년대 중순까지는 활발하게 사용되었으나 현재는 거의 사용되지 않고 있다.
- 8051: 대표적인 8비트 마이크로 프로세서로 인텔에서 제조했으며, 8051의 코어를 사용하는 수많은 MCU들이 아직도 출시되고 있다. 수많은 어플리케이션이 존재하는 교과서적인 MCU이다.
- 80196: 인텔 16비트 마이크로 프로세서로 여러모로 강력한 기능들을 가지고 있지만 프로그램 메모리가 없고, 개발환경이 불편하다는 단점을 가지고 있다.

MCU

Micro Controller Unit 특정 시스템을 제어하기 위한 전용프로세서로서 롬(ROM)과 램(RAM)회로 까지 내장하고 있어 사실상 초소형 컴퓨터의 역할을 한다
원칩(one chip)컴퓨터 또는 '마이크'로 불리기도 한다.



RISC: 명령어가 간단하고 수도 적다. 간단한 만큼 실행 속도가 빠르지만, 복잡한 명령어를 사용하려면 많은 명령어를 실행해야 하는 단점이 있다.

근래 출시되는 모델은 거의 RISC 타입을 채택하는 편이다.

- PIC: 미국 마이크로 칩사에서 생산하는 마이크로 프로세서를 일컫는 말이다. 매우 적은 명령어 수를 가지고 있다.
- AVR: ATMEL 사에서 제조한 8bit 마이크로 프로세서로 내부 메모리도 크고, 내부 분주도 하지 않아 속도가 빠르다. 또한 C언어에 최적화되었고 ISP를 이용하므로 개발에 용이하다.

2) 모터

로봇 제작에 사용되는 모터는 DC 모터, 스텝 모터, RC서보 모터 등이 있다.

바퀴형 로봇에는 DC모터나 스텝 모터를 많이 사용하고, 관절형 로봇에는 RC서보 모터를 많이 사용하고 있다.

- DC 모터 : DC 모터는 높은 회전수를 얻을 수 있으며 비교적싼 값에 큰 출력을 얻을 수 있는 장점이 있다. 그러나 모터의 수명이 짧으며, 속도를 일정하게 유지한다든지 일정한 거리를 정확히 이동하게 한다든지 하는 일을 시키기 위해선 복잡한 부가 장비들을 붙여야 한다는 단점이 있다.
- 스텝 모터 : 스텝 모터는 입력되는 펄스의 수에 비례하여 모터가 회전한다. 회전각 검출을 위한 피드백이 필요 없고, 제어기가 간단하다는 장점을 가지고 있지만, 고속 운전 시 탈조되기 쉽고 토크가 DC 모터에 비해서 효율이 떨어진다는 단점을 가지고 있다.

ISP

In system Programing의 약자로서, 롬라이더를 이용하지 않고서 MCU에 직접 기계어 코드를 이식할 수 있도록 구성된 방식이다.

- RC 서보 모터 : RC 서보 모터는 무선 모형 제품 즉 RC 자동차, RC 비행기의 방향 조정에 사용된 모터인데 크기에 비해 큰 토크와 주변회로가 간단하다는 장점을 가지고 있다. 단점은 제품에 따라 다르지만 180도 정도의 제한 각도를 가지고 있고, 많은 전류를 소모한다.



〈그림 II-5〉 모터 종류

메모리 효과(Memory Effect)

충전지 중 Ni-cd 및 Ni-MH 배터리에 있는 특성으로 배터리가 사용했던 용량을 기억하여 배터리 수명을 단축시킨다.

이것은 배터리가 100% 용량 중 10%만 자주 충전할 경우 배터리가 자기용량을 10%만로만 기억하게 된다.

이것을 해결하기 위해서는 가끔씩 완전방전을 시켜주어야 한다.

3) 배터리

로봇 제작에 사용되는 배터리는 니켈 수소, 니켈 카드뮴, 리튬 폴리머, 납축 전지 등을 사용하고 있다.

- 니켈 카드뮴 : 내부 저항이 작고 순간 방전율이 뛰어나며 열악한 조건(저온, 과방전, 과충전)등에 잘 견디나, '메모리 효과'라는 현상 때문에 완전히 방전하지 않고 충전하면 용량이 줄어드는 현상이 일어난다.
- 니켈 수소 : 니켈 카드뮴 전지에 비하여 2배의 용량과 '메모리 효과'가 개선되었지만 많은 전류를 한 번에 내보내지 못한다.
- 리튬 폴리머 : 방전율이 매우 우수하며, 경량 고용량 고전압의 출력을 가지



지만, 메모리 효과가 없으며, 다른 배터리에 비해 비싼 편이다. 충전이 올
바르지 못하면 폭발 위험성을 가지고 있다.

- 납축전지 : 일반적으로 배틀 로봇에 많이 사용된다. 가격에 비해서 많은 전
류량을 가지고 있다는 장점을 가지고 있지만 무겁고 방전율이 낮다는 단
점을 가지고 있다.

	니켈 카드뮴	니켈 수소	리튬 폴리머	납축전지
실물				
방전율	좋음	낮음	매우 좋음	낮음
장점	순간 방전율이 좋다.	과방전 과충전 급속 충방전에 유리하며, 니켈 카드뮴의 동일크기 대비 고용량	경량, 고전압 고용량 및 메모리 효과가 없다.	가격이 저렴하다 편한 관리
단점	용량이 작다. 메모리 효과	메모리효과	비싸다. 폭발위험성	무겁다

〈표 II-3〉 배터리 특징

4) 센서

- 초음파 거리 감지 센서

초음파를 이용하여 거리 측정이 가능한데 초음파 센서에서 발사된 초음파가
벽이나 장애물의 표면에 반사되어 되돌아올 때까지의 지연 시간을 측정하여 장
애물까지의 거리를 계산한다.

- 적외선 거리 감지 센서

적외선을 이용한 방식은 발광부와 수광부 한쌍으로 발광부에서 내보낸 적외선이 벽이나 장애물의 표면에 반사되어 되돌아 온 적외선의 양을 수광부에서 받아 0.2V ~ 3.2V의 변하는 출력 값을 이용하여 거리를 계산한다.

- 압력 센서

압력에 따라 전기적 저항 값이 변하는 센서로서, 압력이 없을 경우 높은 저항 값을 출력하고 압력이 가해지면 5kΩ 정도까지 저항이 내려간다. 전자 저울이나 로봇의 발에 부착하여 로봇의 균형을 잡는데 사용된다.

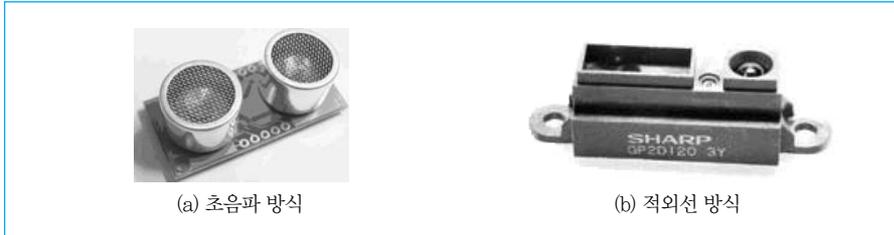
- 광량 센서(Cds)

빛의 양에 따라 전기적 저항값이 변화하는 센서로 빛이 밝을 때 약 30kΩ 정도의 저항값을 가지고 어두울 경우 5MΩ의 저항값을 가지게 된다.

회로를 추가하여 ON/OFF방식의 자동 전등기 제작이 가능하다.

- 온도 센서(서미스터)

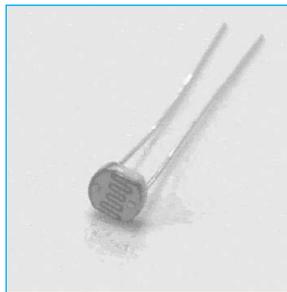
온도의 변화에 따라 전기 저항값이 변하는 센서이다. 물론 모든 물질은 온도에 따라 전기 저항이 변하지만, 서미스터는 그 현상을 공업적으로 이용해서 만든 소자를 통칭하는데, 일반적으로 NTC 서미스터와 PTC 서미스터로 나눌 수 있다. NTC(Negative Thermal Coefficient)는 온도가 올라갈수록 저항이 감소하는 물질이고, PTC (Positive Thermal Coefficient)는 반대로 온도가 올라가면 저항이 증가하는 물질이다. 보통 금속은 온도가 올라갈수록 저항이 증가하고 세라믹 재료는 저항이 감소한다. 하지만 일반적인 서미스터하면 세라믹 계통의 소자를 지칭한다.



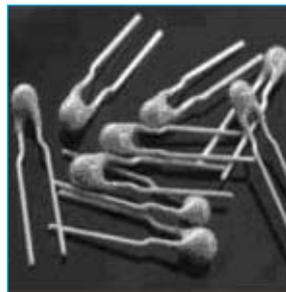
〈그림 II-6〉 거리 측정 센서



〈그림 II-7〉 압력 센서



〈그림 II-8〉 광량 센서



〈그림 II-9〉 온도 센서

5) 무선 통신

무선 통신을 하는 방법은 RC, 블루투스, RF 방식이 있다.

	RC	블루투스	RF
통신 거리	10 ~100M	10 ~ 50M	10~2KM
통신 속도	20ms	최대 1Mbps	최대 250Kbps
주파수 대역	27 ~ 75MHz	2.4GHz	418MHz~433MHz
채널수	많음	많음	2개
추가 프로그램	없음	필요	필요
추가 회로	없음	필요	필요
장 점	별도의 제작이 필요 없음	장비간 간섭이 없음 보안성 및 신뢰도 높음 양방향 통신	장거리 송수신 양방향 통신
단 점	느린 전송속도, 단방향 통신	재 연결시 많은 시간이 소요	낮은 신뢰도

〈표 II-4〉 무선 통신 비교

〈표 II-5〉은 국내의 전자 부품 쇼핑몰과 모터를 판매하는 인터넷 사이트들이다. 쇼핑몰에는 다양한 센서와 모터, 기타 전자 모듈 등을 팔고 있다. 이러한 제품들을 보면서 새로운 형태의 로봇을 제작해 보자.

	이름	인터넷 주소	판매 제품
1	디바이스 마트	http://www.devicemart.co.kr/	전자 부품
2	엘레파츠	http://www.eleparts.co.k	전자 부품
3	ICBank	http://www.icbank.com	전자 부품
4	미니 로봇	http://www.minirobot.co.kr/	전자 부품
5	스카이 홀릭	http://www.hobbytech.co.kr	RC 모터, 배터리
6	모터 뱅크	http://motorbank.co.kr/	각종 모터

〈표 II-5〉 부품 인터넷 쇼핑몰

5 로봇 조립 및 전장

회로 설계 및 부품 선정이 끝나면 로봇의 기구부에 대한 조립을 시작한다.

CAD에서 제작한 도면과 실제 제작된 제품과의 차이는 있는지 실제 조립했을 때에 문제는 없는지, 아니면 관절형 로봇인 경우 관절이 접히는 각도는 예상대로 나왔는지 등을 확인한다.

만능 기판에 회로를 구성하고 전원을 넣기 전에 회로시험기로 전원이 합선되지 않았는지 전선은 잘 연결되었는지를 확인한다.

회로 기판에 아무 이상이 없다면 로봇 동작에 영향이 미치지 않는 곳에 회로 기판과 배터리를 고정시킨다.

6 로봇 소프트웨어 설계

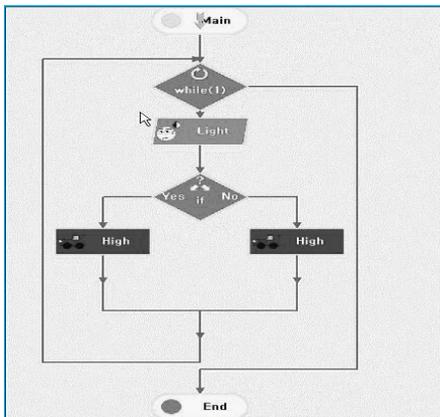
회로 및 기구부가 완성되면 이제부터 로봇을 제어하기 위한 프로그램을 작성해야 된다.



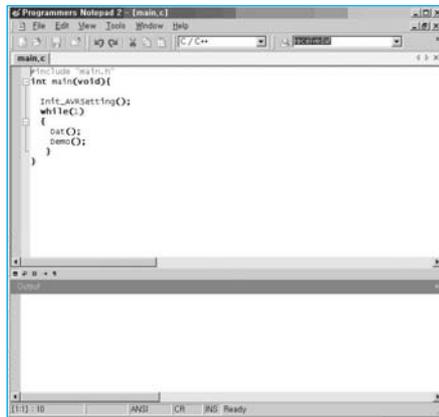
현재 많이 사용하고 있는 ATMEL 사의 AVR을 기준으로 진행해 보자.
 AVR은 어셈블러, C언어, BASIC을 지원하는 컴파일러가 시중에 나와 있다.
 앞에서 설명했듯이 AVR은 C언어를 이용한 개발에 적합하기 때문에 C언어를 공부해서 개발하는 것을 추천한다.
 C언어로 개발하기 전에 해야 할 일은 로봇에 어떠한 기능을 넣을 것인가 이다. 센서들의 인·아웃 관계를 미리 확인하면서 플로 차트를 그려보고 시작을 해야 실제로 코딩 했을 때 발생하는 오류를 많이 줄일 수 있다.

AVR C 컴파일러의 종류

- Atmel 사의 AVRStudio
- IAR사의 IAR Workbench for AVR · HP Infotech의 Codevison
- Image craft사의 Imagecraft(ICC)
- GNU의 GCC-AVR



〈그림 II-10〉 플로 차트 작성



〈그림 II-11〉 C언어 코딩

7 로봇 구동 실험 및 결과 보고

프로그램이 완성되면 로봇을 구동 시험해 본다. 프로그램대로 로봇이 동작하는지 확인을 하고, 센서의 반응 정도 및 배터리의 지속 시간을 확인해 그 결과를 토대로 결과 보고서를 작성한다.

결과 보고서

※각 항목에 반드시 검토 및 토의 내용을 작성하여 제출.

	실험조(조)		공동 제작자		공동 제작자	
	담당	기구 설계	담당	회로 설계	담당	프로그래밍
	학번	성명	학번	성명	학번	성명

1. 수행 일정 계획

단계별 수행작업	수행 일정						
	10/1 주	10/2 주	10/3 주	10/4 주	11/1 주	11/2 주	11/3 주
제작 일정 계획							
기구부 설계							
회로 설계							
프로그램 코딩 및 디버깅							
구동 및 결과 보고							

3. 사용한 각 부품 설명 간단하게 요약

4. 제작 과정 요약

5. 프로그램 흐름에 대한 설명

6. 검토 및 결론

〈그림 II-12〉 결과 보고서 예제



단원 학습 정리



- 1 여러 형태의 로봇 중 제작하고자 하는 로봇을 선정해야 된다.
- 2 로봇 제작 과정에서 좀더 효과적으로 과정을 수행하기 위해서는 현재 제작되어 나와 있는 로봇을 조사하면 각 로봇의 장점 및 제작하고자 하는 로봇의 형태를 계획하는 데 많은 도움이 될 뿐만 아니라 제작 시간 및 실수를 줄일 수 있다.
- 3 로봇 하드웨어를 설계하기 위해서는 다양한 CAD/CAM 프로그램을 이용해서 제작이 가능하다.
- 4 하드웨어 설계가 끝나면 로봇을 제어하기 위한 회로 설계 및 전자 부품을 선정해야 된다.
- 5 회로 설계 및 부품 선정이 끝나면 로봇의 기구부에 대한 조립을 시작한다. CAD에서 제작한 도면과 실제 제작된 제품과의 차이는 있는지 실제 조립했을 때에 문제는 없는지, 아니면 관절형 로봇인 경우 관절이 접히는 각도는 예상대로 나왔는지 등을 확인한다.
- 6 회로 및 기구부가 완성되면 이제부터 로봇을 제어하기 위한 프로그램을 작성해야 된다
- 7 프로그램이 완성 되면 로봇을 구동 시험해 본다.



III

로봇 제작 기초

로봇을 제작하기 위해서는 많은 기술이 요구된다. 로봇에 대한 기본 이론뿐만 아니라 각 부품의 원리 및 전체적인 구성형태를 이해하고 설계 방법 대한 노하우도 지녀야 한다.

본 단원에서는 로봇의 구성과 동작원리를 이해하고, 로봇의 하드웨어와 소프트웨어의 설계 순서 및 방법을 학습한다.



학습목표

1. 로봇의 구성과 동작 원리를 이해한다.
2. 로봇의 주요 구성 부품들의 원리를 이해한다.
3. 로봇 제작에 필요한 단계별 계획서 작성을 배운다.
4. 로봇의 H/W와 S/W의 설계 순서 및 방법을 이해한다.



1 로봇의 구성 이해

1 직교 로봇, 스카라(SCARA) 로봇

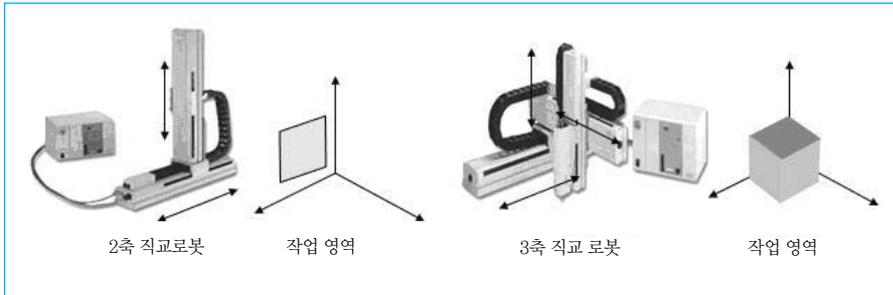
1) 직교 로봇의 정의

일명 XY 로봇이라고 불리는 직교 로봇(Cartesian Coordinate Robot)은 <그림 III-1>에서와 같이 직선 운동을 수행하는 축으로만 구성되어 있는 로봇이다.

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

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

직교 로봇의 작업 영역은 구성하는 자유도의 수에 따라 직선, 직사각형, 직육면체가 되는데 이것은 인간에게 가장 익숙한 좌표계가 바로 90도씩 분할된 직각 좌표계이기 때문에 일반 로봇 이용자들이 쉽게 배워서 사용하고 있으며, 또한 산업현장에서 널리 이용되고 있는 로봇들 중의 하나이다.



〈그림 III-1〉 2축과 3축 직교 로봇 작업 영역

직교 로봇은 1축의 직교 로봇을 기본 모듈로 하여 조합하기에 따라 여러 가지 형태를 만들어 낼 수 있다.

아래축의 슬라이더(Slider)에 위의 축을 연결하기도 하며 또 위의 축의 슬라이더를 아래축의 슬라이더에 직접 연결한 경우도 있다.

이렇게 다양한 형태의 조합이 가능한데 주어진 작업에 가장 적합한 것을 선택하는 것이 중요하다.

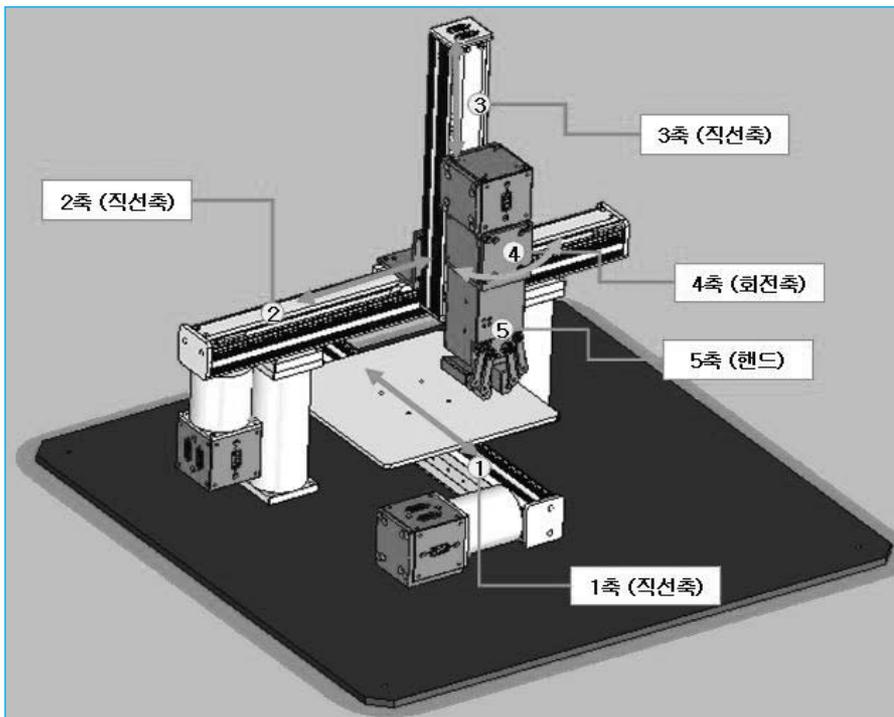
● 보충 설명

슬라이더(Slider)란? 볼스크류의 너트와 LM-가이드의 블록을 연결하여 직선 운동을 가능케 하며, 로봇 사용자에게 메카니컬 인터페이스(Mechanical Interface)를 제공한다.

2) 직교 로봇의 구성

완성된 직교 로봇은 일반적으로 〈그림 III-2〉와 같은 구성 사양을 가진다. 각 축의 배치는 기계적인 특성 및 구동 제어적인 특성을 고려하여 배치하였다.

1축은 베이스(Base)판에 장착되어 있으며, 1축의 엔드 이펙터(End-effector)에는 슬라이더가 장착되어 작업대상물을 옮길 수 있는 구조이다. 2축의 슬라이더는 3축 직선 축이장착되어 있으며 회전축과 핸드모듈이 장착되어 있다.



〈그림Ⅲ-2〉 직교로봇의 구성 및 명칭

보충설명

가반중량(Payload)이란? 로봇 사양 중에 가반중량의 의미를 잘 이해하여야 로봇을 올바르게 사용할 수 있다. 로봇이 허용하는 가반중량은 엔드 이펙터에 부착된 핸드모듈이나 툴 무게에 들어 옮기는 작업 대상물의 무게를 더한 전체 중량(작업 중량)이 가반중량 이하가 되어야 한다는 것을 의미한다.

작업하중(Payload) ≥ 엔드 이펙터(End Effector)하중 + 작업 대상물(Workpiece)하중

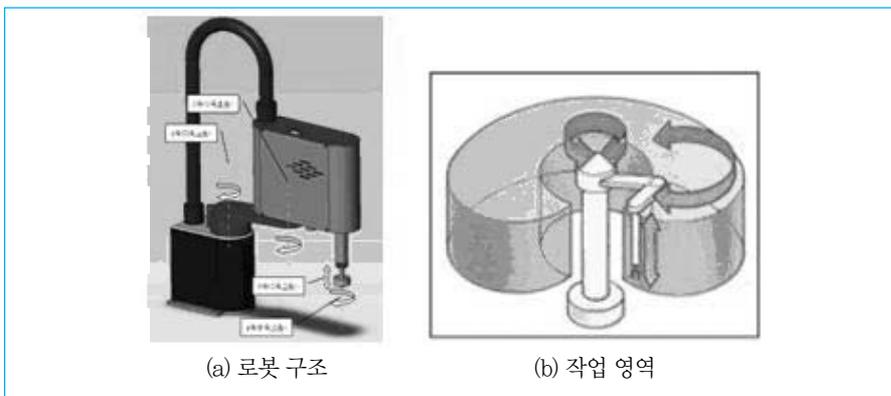


3) 스카라 로봇의 정의

스카라 로봇(Selective Compliance Assembly Robot Arm, SCARA)이라고도 불리는 수평 다관절 로봇(Horizontal Articulated Robot)은, <그림 III-3>에 서와 같이 2개의 회전축과 하나의 직선축으로 구성된 로봇이다.

1,2축은 수평 회전 운동을 하고 3축은 수직 상하 운동을 하며, 3축은 다시 회전운동을 수행한다. 스카라 로봇은 수평면상의 운동 특성을 가지고 있으므로 조립 작업 또는 팔레타이징(Palletizing) 작업에 적합하다.

전자부품 조립에서 평면상의 부품은 X,Y 그리고 q(회전)의 3 자유도를 갖는데 여기에 Z방향으로 들어서 옮겨야 하기 때문에 보통 4개 자유도를 가지고 움직이게 된다. 이런 작업을 테이블-탑 오퍼레이션(Table-Top Operation)이라고 하는데, 빠른 속도가 필요하며 설치 면적이 협소한 경우에는 수평다관절 로봇이 안성맞춤이다.



<그림 III-3> 스카라 로봇 구조와 작업 영역

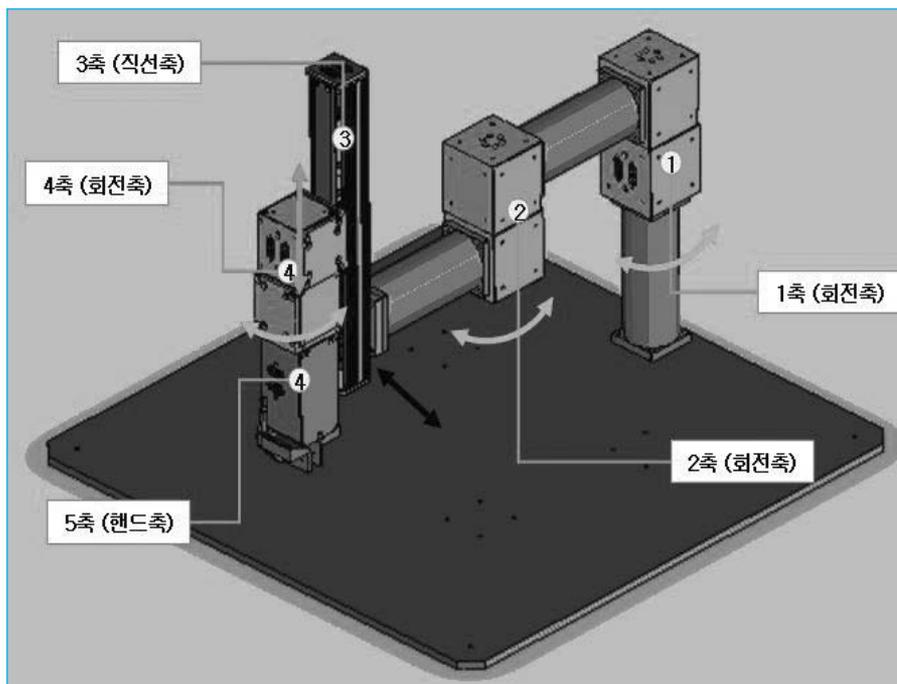
● 보충설명

팔레타이징(Palletizing)은 로봇이 물건을 적재할 때 일정한 크기의 물건을 지정한 위치에 순차적으로 적재할 수 있는 기능을 일컫는다. 물론 전체 위치를 알려줘야 할 필요없이 최소한의 정보로 로봇이 자동으로 계산하여 물건을 적재하는 것이다.

4) 스카라 로봇의 구성

완성된 스카라 로봇은 일반적으로 <그림Ⅲ-4>와 같은 구성 사양을 가진다. 각 축의 배치는 기계적인 특성 및 구동제어적인 특성을 고려하여 배치하였다.

1축 회전축과 2축 회전축은 용량이 큰 모터가 장착되어 있으며, 3축은 직선축으로 구성된다. 4축 회전축과 핸드모듈이 장착되어 작업이 가능한 구조이다.



<그림Ⅲ-4> SCARA 로봇의 구성 및 명칭

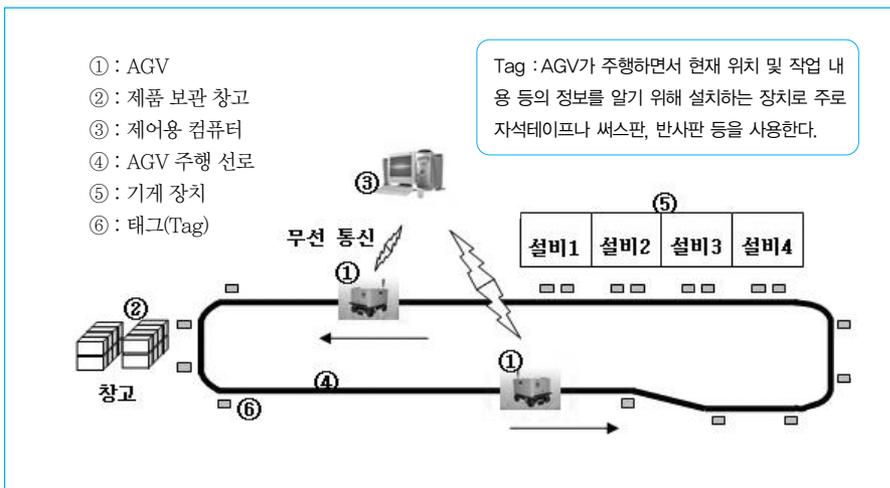
2 이동(Mobile) 로봇

다양한 형태로 상용화되고 있는 이동 로봇은 분류 범위가 너무 넓고 다양하



기 때문에 본 과정에서는 공장에서 사용되고 있는 이동 로봇인 AGV(Automatic Guided Vehicle)에 대해 알아보려고 한다.

일반적으로 AGV(①)는 <그림 III-5>와 같이 전체 공장을 제어하는 컴퓨터(③)로부터 주행 정보를 받아 선로(④)를 주행하게 된다. AGV의 위치를 인식하는 센서가 장착된 경우는 AGV의 위치(번지)를 인식하기 위한 Tag(⑥)가 필요하지 않지만, 일반적인 AGV의 경우 아래 그림과 같이 특정한 기능을 구현할 위치에 Tag를 부착하고, AGV에 장착된 센서를 이용해서 이 Tag를 인식하고, 해당 Tag 위치에서 할 일을 수행하게 된다. 컴퓨터는 설비(⑤)에서 필요로 하는 부품의 유무 상태를 파악하여 창고에서 해당 설비로 부품을 보낼 수 있도록 주변 기기들을 제어하게 된다. 부품을 운송하는 물류 시스템을 간단하게 구성할 경우는 상위 컴퓨터 없이 AGV만을 이용하여 구성할 수도 있다.



<그림 III-5> AGV 설치 예

1) AGV의 형태

〈그림Ⅲ-6〉과 같이 AGV가 하는 일의 종류에 따라 다양한 형태의 AGV가 있다. 이동할 대상을 실은 대차를 끌고 다니는 대차 및 탑재형 구조와 지게차처럼 선반에 적재된 물건을 이동하는데 편리한 포크형, AGV에 간단한 컨베이어를 탑재하여 물건을 자동으로 이송할 수 있는 구조의 컨베이어형, AGV에 로봇을 탑재하여 반도체 공장에서 다양한 작업을 할 수 있도록 만든 로봇 탑재형 등이 있다.



〈그림Ⅲ-6〉 AGV의 기능에 따른 형태

2) 가이드(Guide) 방법

AGV가 자동으로 길을 찾아갈 수 있도록 하기 위한 보조적인 센서들로 사용하는 환경이나 정밀도, 복잡도, 가격 등에 따라 선택하여 사용하면 된다. 다음 방법들은 AGV에서 주로 사용되고 있는 가이드 방법들이다.



- ① 적외선(IR) 센서, 자기 테이프, 자기 유도 방식 : 이 방법은 바닥에 AGV에 장착된 센서가 인식할 수 있는 재질이나 방법으로 주행할 선로를 미리 설치하고, AGV에 장착된 센서를 이용하여 길을 인식하면서 주행한다.
- ② 레이저(Laser)를 이용한 방법: AGV 몸체에 레이저 스캐너를 설치하고, AGV가 주행할 통로에 레이저 빛을 반사시키는 반사판(알고 있는 위치)을 설치한다. 그 후 레이저 스캐너를 360° 회전시키면서 반사판에서 반사된 빛을 이용하여 자기 위치를 인식하는 방법이다.

3 창작 로봇

뱀 로봇의 동작 원리는 일반적인 로봇과는 많은 차이가 있다. 2족 로봇이나, 4족 로봇과 같은 걷는 형태의 로봇은 다리관절을 이용하여 이동을 한다. 반면 뱀 로봇은 서보 모터를 좌, 우로 움직여서 발생한 힘을 이용하여 움직인다. 그러나 모든 서보 모터들을 단순히 좌, 우로 움직인다고 해서 로봇이 움직이지는 않는다. 로봇이 움직이는 데에는 일정한 규칙에 의해서 만들어진 힘을 이용해야만 한다. 즉, 로봇이 전진 하기 위해서는 필요한 힘을 발생시키는 방법을 알아야 한다.

1) 생물학적인 뱀의 움직임

뱀 로봇은 생물학적인 뱀의 모습과 움직임을 모방한 형태의 로봇이므로 생물학적인 뱀의 움직임을 파악해야 구동 원리를 찾을 수 있다. 다음은 지성사 자연사박물관 '뱀' 편에 수록되어 있는 뱀의 이동 방법으로 다음 4가지가 있다.

① 측선 물결 운동

뱀은 몸통을 좌우로 물결 모양으로 굽혀 그 굽힌 바깥쪽을 어떤 돌출물에 밀

어붙여 전진한다. 매끄러운 표면에 뱀을 놓아두면 앞으로 가지 못하고 꿈틀거리기만 하는데, 이런 상황에서 몸을 압착시킬 수 있는 돌이나 막대기 등의 돌출물을 놓으면 뱀은 그 물체에 몸통을 밀어 붙여 앞으로 나아가게 된다. 모래사장 에서 이 방법을 사용할 경우에는 돌출물이 없으므로 몸을 굽힌 바깥쪽에는 작은 모래 산이 세워지게 된다. 이때 몸통은 지면을 흐르는 것같이 보이고 물결 모양의 길을 달리는 것이다. 이 방법은 거의 대부분의 뱀이 사용하며, 이 방법으로 물 위에서 헤엄을 칠 수가 있다.

② 직선 운동

이 방법은 말 그대로 일자로 가는 방법으로 보면 된다. 몸통이 굽은 뱀(예 : 보아뱀, 살모사, 비단뱀 등)들이 사용하는 방법이다. 몸통이 가는 뱀은 이 방법을 사용하지 않는다. 이 방법을 쓸 때 뱀의 피부는 근육과 밀착되어 있지 않으므로 피부를 지면의 한 점과 붙이고 피부 안의 몸통을 밀어낸다. 먼저 반듯하게 몸을 뻗어 배 비늘을 지면의 작은 돌기에 걸리게 한 후 피부 안의 몸통을 잡아 당기고 몸통이 나아간 곳까지 피부를 끌어당긴다. 이런 방식을 계속하여 앞으로 나아가게 되며, 지네가 기어가는 것 같이 보인다. 이 방법은 느리고 조용해서 주로 뱀이 먹이를 잡을 때 사용한다.

③ 아코디언식 운동

이 방법은 미끄러운 표면 또는 측선 물결 운동이 되지 않는 좁은 장소를 지날 때 쓰인다. 뱀이 이 방법을 쓸 때에는 몸통을 지그재그로 잡고 몸통 앞 부분을 밀어서 몸을 고정시킨 다음에 몸통 뒷 부분을 잡아당긴다. 이런 방식을 계속하여 아코디언이 퍼졌다 쭈그러졌다 하는 것처럼 전진하게 된다.

④ 사이드와 인딩



이 방법은 아주 적은 종류의 뱀(예 : 유럽 초원뱀, 캘리포니아 대님뱀, 사이드 와인더 방울뱀 등)이 사용하며 주로 사막에 사는 뱀들이 사용하는 방법이다. 진행되는 것을 보면 용수철이 옆으로 굴러가는 것과 비슷하게 보인다. 쉽게 말해 몸을 S자로 굽힌 뒤 옆으로 계속해서 밀어낸다. 즉, 앞으로 가는 것이 아니라 옆으로 미끄러져 가는 것이다.

대부분의 뱀은 위에서처럼 세 가지 방법을 사용한다. 따라서 뱀이 앞으로 갈 때에는 꼭 S자로 몸을 굽혀서 가는 것만은 아니라는 것이다. 뱀은 네 번째 방식을 제외하면 앞으로 갈 수 있으며 뒤쪽으로는 절대로 갈 수가 없다. 그 이유는 배 비늘이 앞으로 갈 수 있도록 방향이 나 있기 때문이다.

뱀 로봇에 적용된 이동 방식은 사인(Sine)파 곡선 형태의 이동 방식이다. 일반적인 뱀의 움직임이 S자형의 움직임 즉, 사인파 형태의 움직임이기 때문이다.

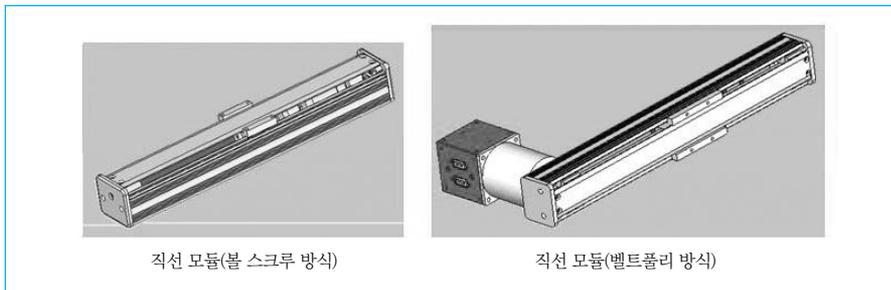
벨트 풀리

로프를 걸어서 회전시키는 바퀴로서, 도르레의 동작과 유사하다.

2 로봇의 동작원리 이해

1 직교 로봇, 스카라 로봇

1) 직교 로봇의 동작 원리 이해

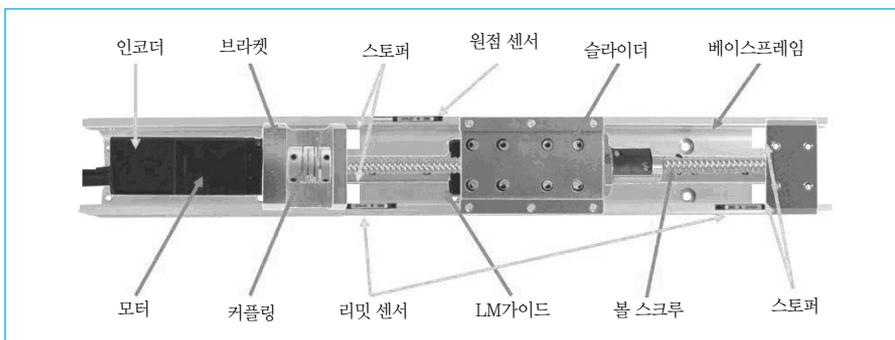


직선 모듈(볼 스크루 방식)

직선 모듈(벨트풀리 방식)

〈그림 III-7〉 직선축 구동 전달 방식

직교 로봇에서는 모든 기구적 부품들이 장착되는 알루미늄 압출 소재의 베이스 프레임 (Base Frame)과 그 내부의 공간에 회전 동력을 발생시키는 액추에이터 (즉, Servo Motor), 그리고 회전 동력을 직선 운동으로 변환하여 주는 볼스크루(Ball Screw)가 핵심 구성품이다.



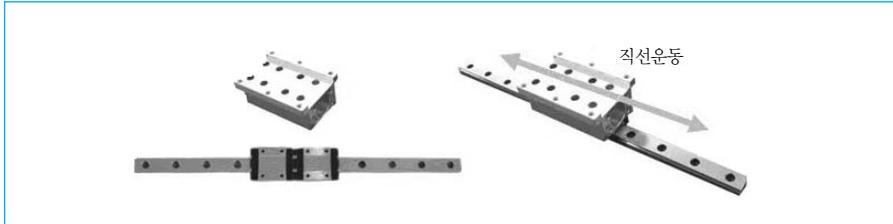
〈그림Ⅲ-8〉 직선축 구동 전달 방식의 핵심 구성품.

로봇의 작동 원리를 살펴보면, 우선 모터가 회전을 할 경우 모터 축과 볼스크루 사이에 있는 커플링(Coupling)을 통하여 볼스크루에 회전 운동이 전달된다.



〈그림Ⅲ-9〉 볼스크루의 특징

볼스크루의 회전 운동은 LM 가이드(LM Guide)와 함께 볼스크루 너트에 고정된 슬라이더를 직선 운동을 만든다. 로봇의 사용자들은 이 슬라이더의 취부면에 핸드(Hand) 또는 툴(Tool) 등의 엔드이펙터(End-Effector)를 달아 사용하게 된다.

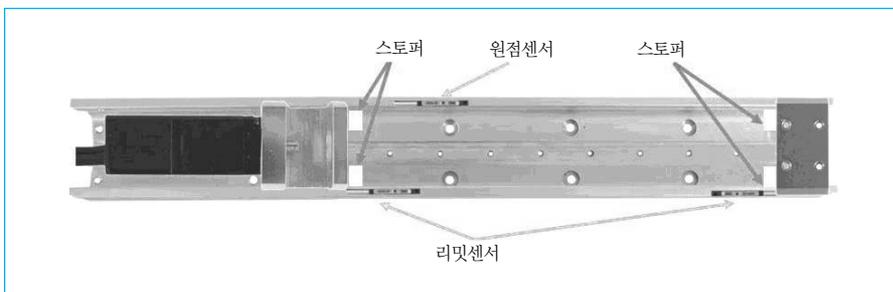


〈그림 III-10〉 LM가이드

센서들은 로봇의 원점복귀나 하드웨어적 제한(Limit)을 위해 사용된다.

먼저 〈그림 III-11〉의 가운데에 위치한 센서는 단축 로봇의 원점을 잡기 위한 것으로 로봇이 제어기의 서보 온(Servo on)이후 원점 복귀(Zero Return)을 명령하면 이 센서를 기준으로 슬라이더가 이동하여 항상 같은 위치에서 정지한다. 절대값 인코더(Absolute encoder)를 사용하는 모터의 경우 이 원점 복귀 센서를 사용하지 않아도 되지만 보다 안전하게 로봇의 준비(시작) 위치를 잡기 위해 사용하기도 한다. 그리고 양쪽의 바깥쪽에 위치한 센서는 로봇의 최대 동작범위를 제한하는 리밋(Limit)센서에 해당한다.

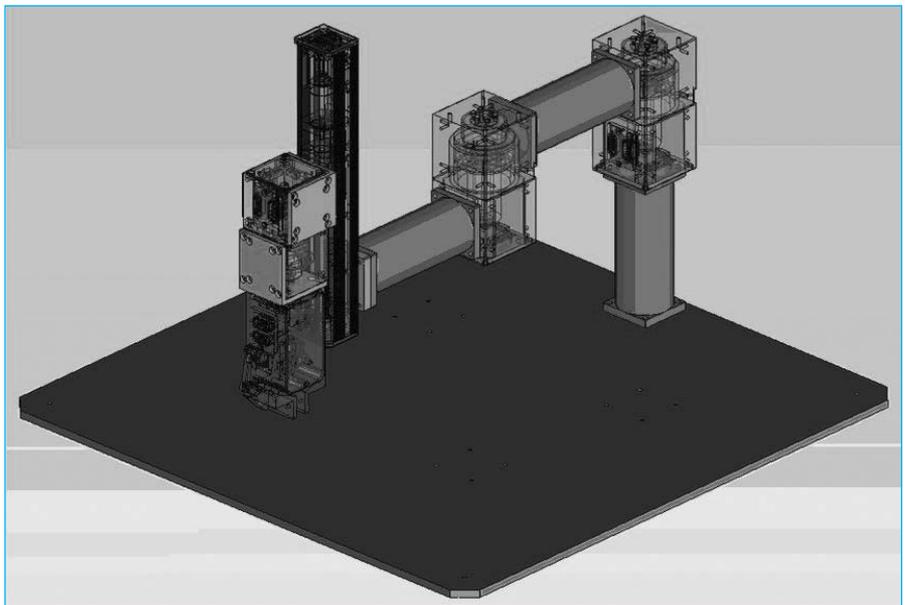
일반적으로 이 최대 동작 범위보다 안쪽에 소프트웨어적으로 리밋(Limit)을 설정해 놓기 때문에 이상 상태가 아니면 이 센서를 사용하지 않는다. 그리고 센서의 바로 바깥쪽에 스톱퍼(Stopper)가 설치되어 이 센서를 통과해도 더 이상 기구적으로 움직일 수 없도록 되어 있다.



〈그림 III-11〉 센서 및 스톱퍼

이와 같이 로봇의 이상 동작에 대비한 이중 또는 삼중의 안전 장치가 붙어 있다. 점점 제어기의 신뢰성이 향상됨에 따라 이 리밋 센서(Limit Sensor)를 사용하지 않는 로봇도 많이 등장하고 있는 것이 현재의 모습이다.

2) 스카라 로봇의 동작 원리 이해



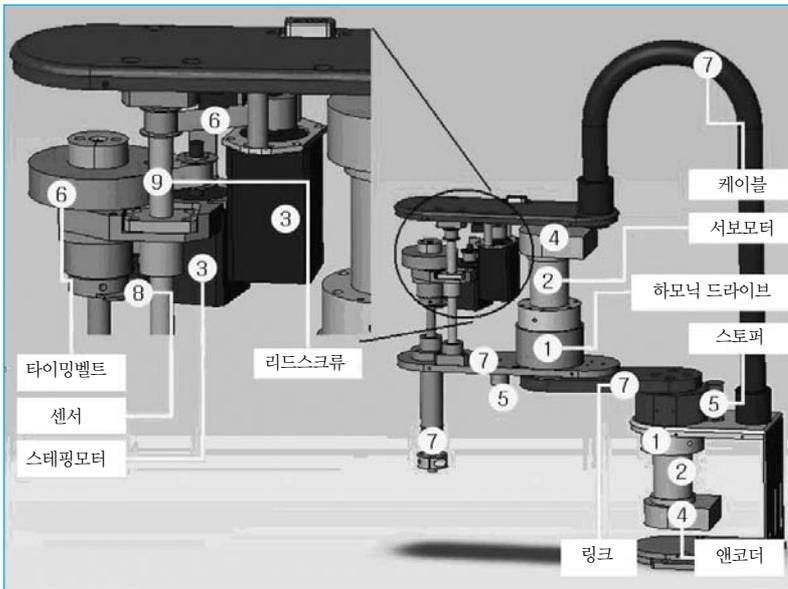
〈그림Ⅲ-12〉 스카라 로봇

수평 다관절 로봇에 사용되는 모터는 대부분 서보 모터이다. 최근에는 서보 모터 중에서 AC 서보 모터가 주류이며 절대값 인코더(Absolute Encoder)를 가진 것이 많아지고 있다. 이것은 DC 서보모터와 달리 브러시 교환이 없어 관리가 편리하고, 전원이 다운되었을 때 원점 복귀를 할 필요가 없기 때문에 많이 사용되고 있는 추세이다.

각각의 모터는 감속기, 볼 스크루 등의 동력 전달 기구와 함께 하나의 관절을 구성하는데 로봇 전체가 어떻게 동력이 전달되어 움직이게 되는지 그 예를 전



체 조립도에 나타내었다.



〈그림 III-13〉 스카라 로봇의 내부 구조

직교 로봇과는 달리 1, 2축이 중력 방향으로 평행하고 3축이 Z방향으로만 움직인다. 따라서 처음의 두 회전축에 직각인 수평면에서의 운동은 매우 빠르다는 장점을 갖고 있다.

직교 로봇처럼 4축으로 만들어진 경우에 테이블탑 오퍼레이션(Table-top Operation)에 강점을 갖는 로봇이다.

직교보다는 처음 두 회전축에 직각인 평면에서의 운동이 더 빠르므로 사이클 타임이 짧아 더 높은 생산성을 가진다.

설치공간 면에서도 유리하다. 1,2축이 중력 방향의 회전축을 가지므로 두 개의 모터는 중력의 영향으로부터 자유롭다. 따라서 중력 방향으로 큰 강성을 가지고 가반 중량을 크게 만들기가 용이하다.

가반 중량 60Kg이상 그리고 빠른 속도가 요구되는 팔레타이징 로봇도 이러

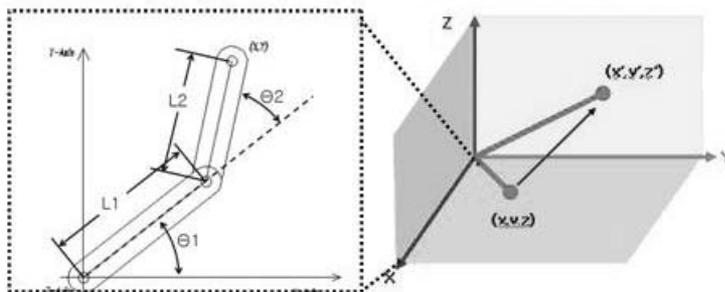
한 구조를 가진다. 다만 Z축이 베이스부에 바로 연결되어 1축이 되고 Z방향으로 큰 동작 범위를 가지는 점이 다르다.

보충설명

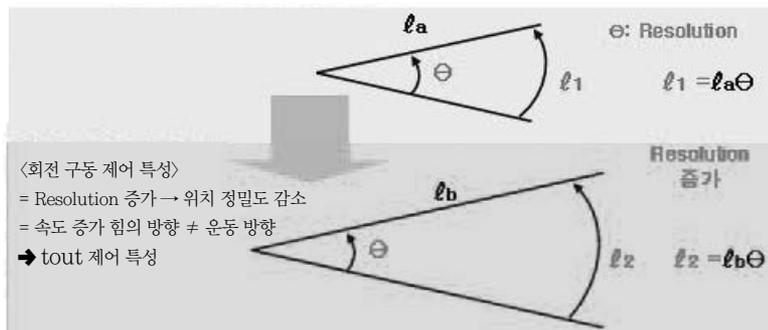
스카라 로봇의 이해

● 실제 산업용 로봇의 예

◎ SCARA 기구 해석 : 1축, 2축 회전 운동의 조합에 의하여 XY 평면상에 위치시킨다.



- ◆ 특징 : - 1축, 2축은 수평 방향으로 회전 모듈 3축은 수직 방향으로 직선 모듈로 구성된다.
- 로봇에 비해 작업 영역이 크다.





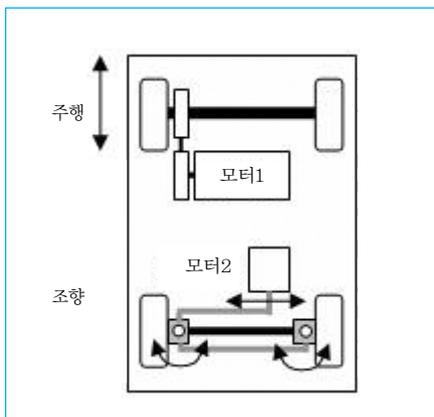
2 이동 로봇

1) 이동 원리

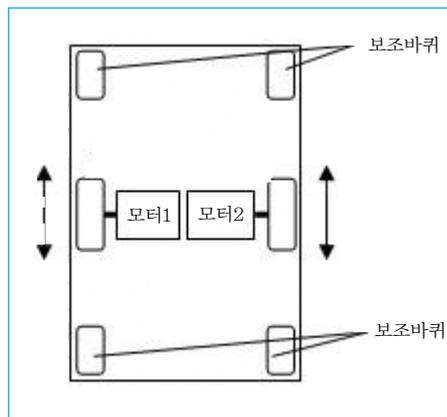
모바일 로봇의 움직임은 전/후로 움직이기 위한 주행 부분과 좌/우로 회전하기 위한 조향 부분으로 구분할 수 있다. 이러한 기능을 구현하기 위해 크게 다음과 같은 방법이 사용된다.

〈그림 III-14〉의 방법은 일반 자동차 구조와 같이 앞 바퀴로 회전, 뒷 바퀴로 주행을 할 수 있도록 모터를 장착한 방법이다. 본 교재에 사용될 이동 로봇의 구조도 이 방법을 사용하였다. 이 방법은 자동차에 사용되는 주요 부품을 사용할 수 있어 싸고 좋은 로봇을 만들 수 있으나, 후진 특성이 떨어지는 단점이 있다.

〈그림 III-15〉의 방법(두 바퀴의 속도차)은 차체의 앞·뒤에 보조 바퀴를 부착하고, 차체의 중앙 좌·우에 2개의 모터를 장착하여 두 모터의 속도차를 이용해서 회전을 한다. 이러한 속도차 방법은 두 모터의 평균 속도로 주행을 제어할 수 있는 가장 보편적인 형태이다.



〈그림 III-14〉 자동차형 구조



〈그림 III-15〉 속도차 구조

적외선

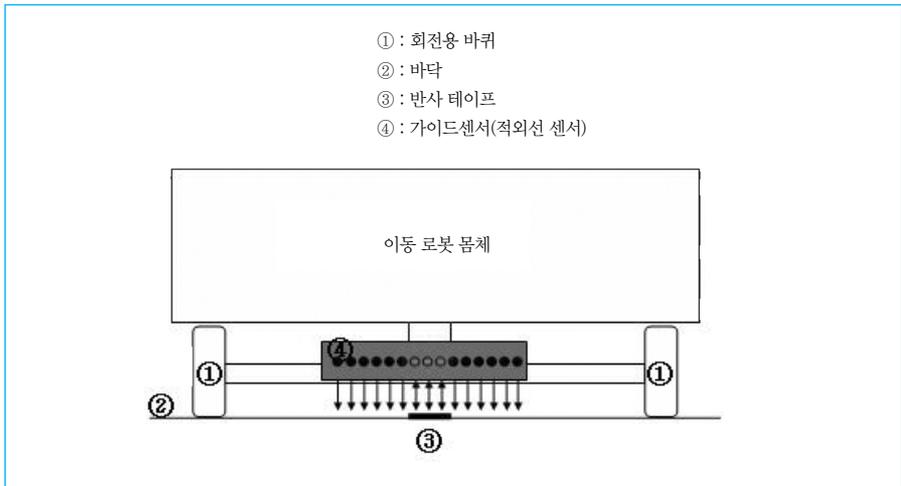
햇빛이 방출하는 빛을 스펙트럼으로 분산시켜 보았을 때 적색 스펙트럼의 끝보다 더 바깥쪽에 있는 전자기파를 적외선이라 한다. (f)보라색 바깥쪽에 있는 전자기파는 자외선이라 한다.

2) 길을 찾는 원리

본 교재에서 사용하는 이동 로봇은 <그림Ⅲ-16>과 같이 바닥에 붙인 반사 테이프를 적외선(가이드 센서) 센서가 감지하여 길을 찾아간다. 이러한 방법으로 이동 로봇이 길을 인식하는 원리는 다음과 같다.

적외선 센서에서 적외선(눈에 보이지 않음)을 바닥에 쏘면 바닥면에서는 이 빛을 흡수하거나 난반사시킨다. 이때 돌아오는 빛의 세기가 작아 센서가 동작하지 않는다. 하지만 선로용으로 사용되는 반사 테이프에서 반사되는 빛의 세기가 커서 센서가 동작하게 되는 것이다. 만일 바닥면이 빛을 반사시키는 특성이 우수하면 반사 테이프 주변에 검정색 테이프를 설치하여 바닥에서 빛이 반사되지 않도록 한다.

이러한 센서를 15개 탑재한 AGV용 가이드 센서로 로봇의 이동 방향을 알고 방향을 조정하는 조향 모터의 회전 방향 및 속도를 계산하여 동작시킨다.

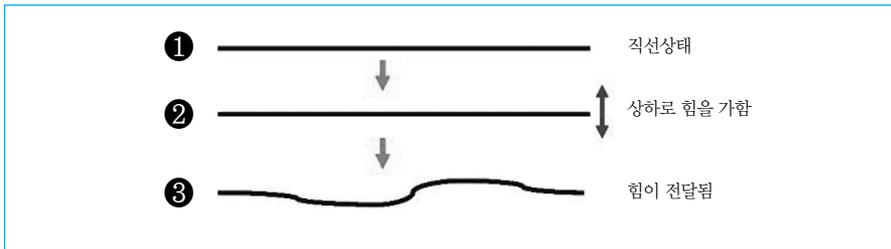


<그림Ⅲ-16> 가이드 센서 동작 원리



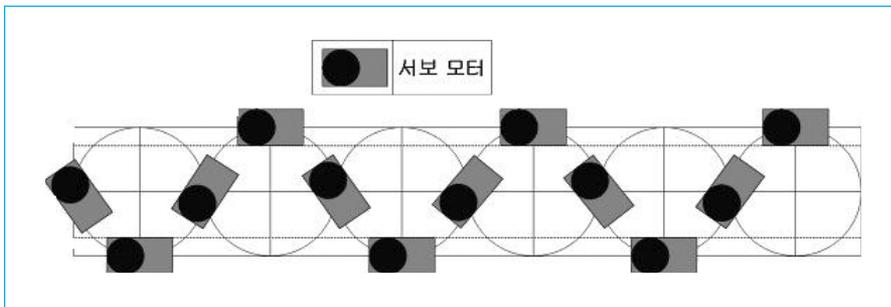
3 창작 로봇

1) 뱀 로봇이 전진할 수 있는 이유



〈그림 III-17〉 직선에서의 힘 전달

〈그림 III-17〉은 뱀 로봇의 전진 원리를 쉽게 이해 할 수 있는 그림이다. 직선 상태의 줄이 외력에 의해 상하로 힘을 받게되면 줄은 파형을 그리며 줄의 처음 부터 끝부분까지 힘이 전달되는 것이다. 이때 힘의 전달 방향을 유심히 봐야 한다. 이러한 원리를 로봇의 몸체 안에 들어 있는 서보모터의 연속적인 상태로 놓고 사인과 곡선을 만들어 보자.



〈그림 III-18〉 서보 모터를 이용한 사인파

모터가 작동할 때 각각의 모터는 고유 넘버를 가진다. 1~12번까지의 모터는 각각의 타이밍에 맞게 순차적으로 구동이 되어야 한다. 1번 모터가 지정된 각도

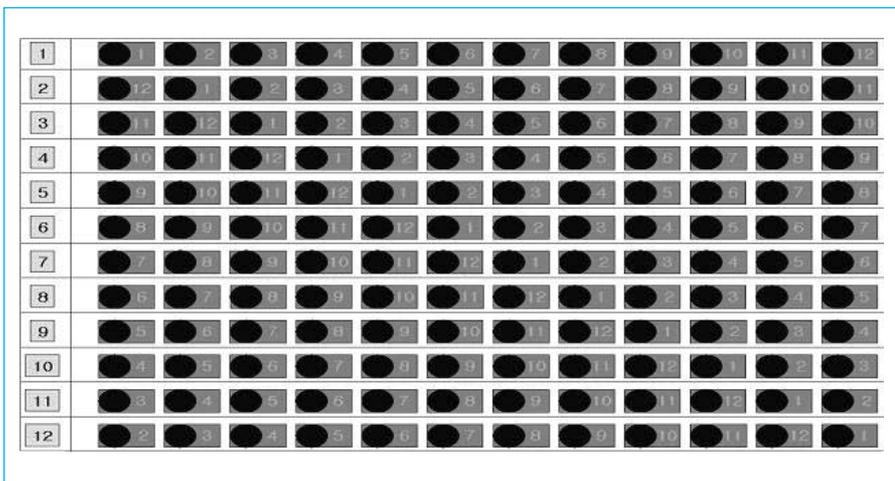
로 이동이 되면, 2번 모터가 뒤를 이어 해당하는 각도로 이동한다. 다음은 3번 모터가 이동하고, 순번대로 이동하여 최종적으로는 12번 모터가 이동이 된다. 즉 1번 모터가 최초 움직이고 순차적으로 움직여 마지막으로 12번 모터까지 움직이는 형태이다. 이 패턴을 기본으로 하여 로봇이 움직인다.

이제 모터 값의 변화를 포함한 모터의 구동을 알아보자.



〈그림Ⅲ-19〉 모터 배열

모터 배열에서 각 모터들은 고유의 모터 값을 가진다. 이 고유의 모터 값을 순차적으로 넘겨주면 로봇은 동작하게 된다. 그림을 기본으로 알아보자.



〈그림Ⅲ-20〉 모터 값의 이동

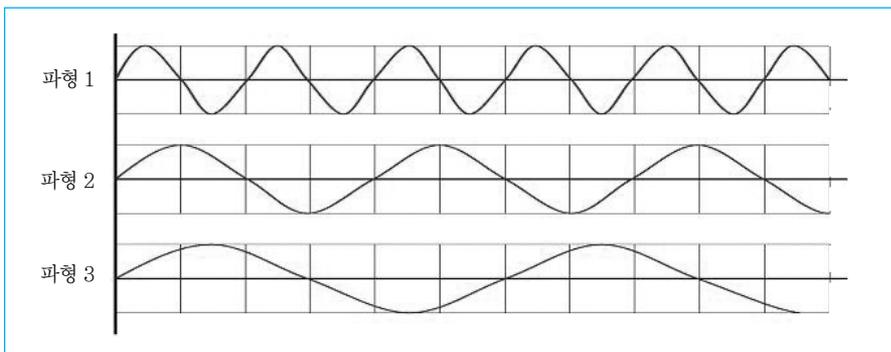
〈그림Ⅲ-20〉을 보면 첫 번째 순서에서 모든 모터들은 각 번호에 맞는 모터의 값을 가지고 있다. 최초 1번에서 12번까지의 모터 값이 인가되었다. 두 번째 순서는 1번에서 12번까지 모터의 값이 우측으로 한 칸씩 이동하였다. 12번 모터



값은 1번 모터의 위치 값이 되었다. 세 번째 순서는 다시 우측으로 한 칸씩 모터의 값이 이동한다. 이런 식으로 모터의 값이 다음 번의 모터 값으로 이동하여 로봇이 구동된다. 모터는 항상 한 번에 1개씩 움직이며, 전체적인 형태는 사인파 형태로 나타난다.

2) 사인파 구하기

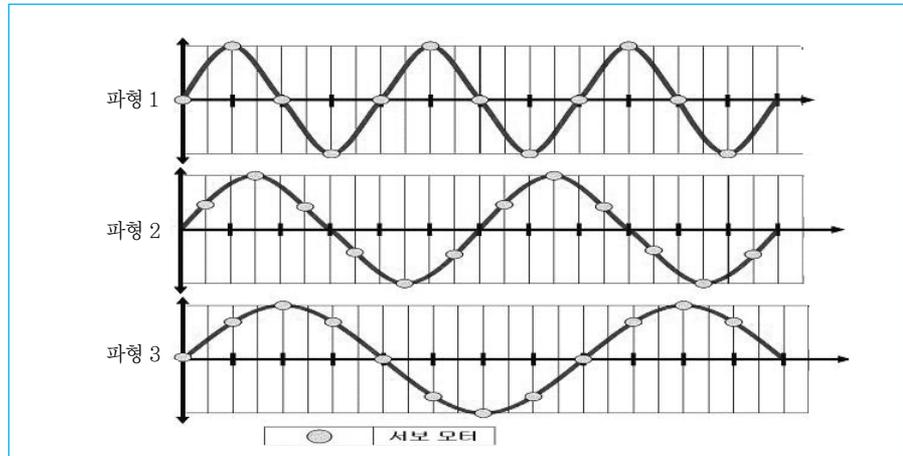
뱀 로봇을 구동하기 위해서는 파형이 필요하다. 기본적인 파형은 사인파를 이용한다. 사인파를 어떻게 넣어 주느냐가 로봇의 움직임 및 이동 속도를 결정하게 된다. 파형의 크기가 커지면 로봇이 이동하는데 좌, 우로 움직이는 거리가 많아지므로 앞으로 전진하는 것이 느려지게 된다. 파형의 크기가 작아지면 좌, 우로 움직이는 거리가 줄어들므로 전진하는 것이 빨라진다. 단 파형이 너무 작으면 로봇이 움직이지 않는다.



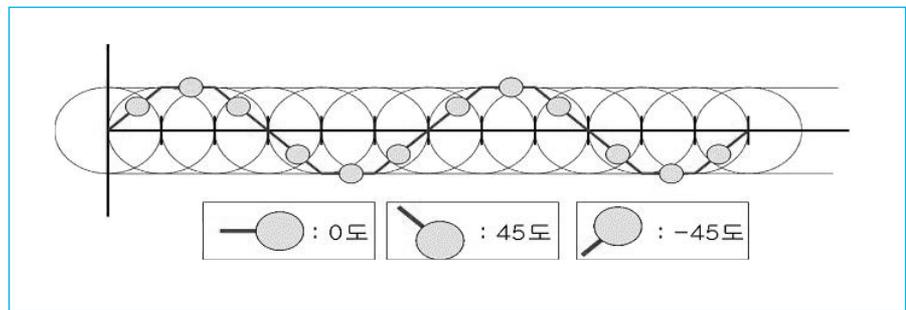
〈그림 III-21〉 사이클 수에 의한 파형

〈그림 III-21〉은 사이클 수에 의한 파형이다. 모터의 배치를 어떻게 하느냐에 따라서 뱀 로봇이 동작 할 때 어떠한 파형을 그리며 움직이는지 알 수가 있다. 뱀이 움직일 때도 한 가지의 S자형이 아니라 S자가 작을 때도 있고 클 때도 있기 때문에 사인파형을 사이클 수에 따라서 분류하고, 이에 맞는 모터의 배치를

함으로써 다양한 형태의 이동이 가능하다. <그림Ⅲ-22>에서 볼 수 있듯이 파형의 사이클 수에 따라서 각기 다른 모터의 배치를 볼 수 있다. 모터의 수는 파형의 주기에 따라서 달라진다. 이유는 그 파형 주기에 맞는 형태를 찾아 사인 모양을 표현하기 위해서이다.



<그림Ⅲ-22> 사이클 수에 따른 모터 배치



<그림Ⅲ-23> 파형2에 대한 모터 값 산출

12개의 모터에 들어갈 모터 값을 산출하자. 산출 방법은 모터를 기준으로 하였을 때 모터를 지나가는 선의 형태로서 알아낼 수 있다. 여기서는 맨 처음의 모터를 지나가는 선의 형태가 -45도를 나타내는 형태와 같으므로 처음의 모터가

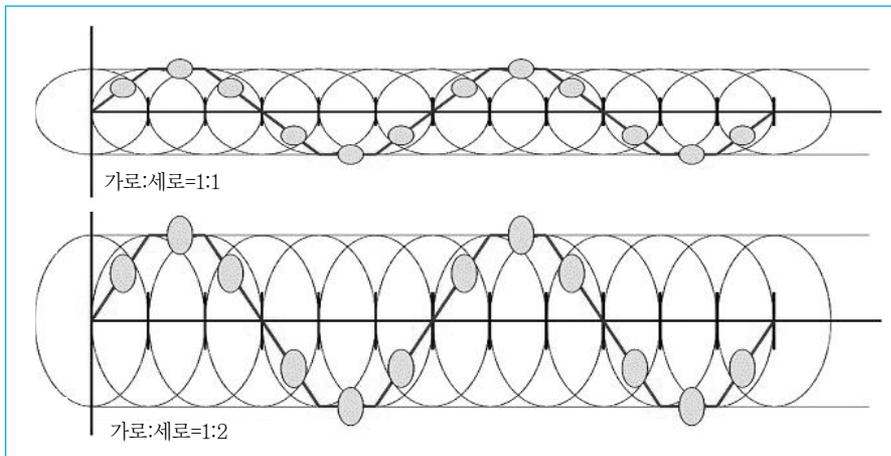


나타내는 각도를 알 수가 있다. 다른 모터들도 이 방법을 이용하여 모두 산출할 수가 있다. 이제 산출한 모든 모터의 값을 표로 확인해 보자.

모터	1번	2번	3번	4번	5번	6번
각도	-45	45	45	45	0	-45
모터	7번	9번	10번	11번	9번	12번
각도	-45	45	45	45	0	-45

〈표Ⅲ-1〉 산출된 모터 값

다음은 비율에 의해서 각 모터의 값이 변화할 수 있다는 것을 보여 준다.



〈그림Ⅲ-24〉 비율에 따른 값 변화

즉, 하나의 파형으로 여러 가지 형태의 값을 얻을 수 있다는 것을 보여 준다. 모터를 기준으로 하였을 때 각 모터의 값이 변화하기 때문이다.

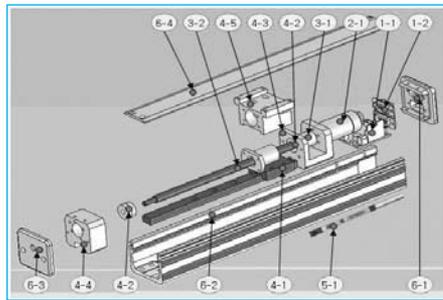
3 로봇의 주요 구성 부품 이해

1 직교 로봇 및 스카라 로봇에 사용하는 모듈

직교 로봇의 구성품은 <그림 III-25>의 볼 스크루 방식의 직선축 모듈과 <그림 III-26>벨트 풀리 방식의 직선축 모듈로 구성된다.

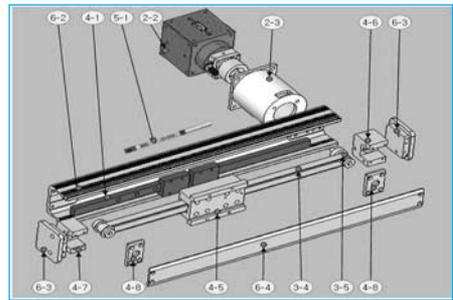
스카라 로봇의 구성품은 <그림 III-25>의 볼 스크루 방식의 직선축 모듈과 <그림 III-27>회전 모듈로 구성된다.

각 모듈에 대한 구성요소에 대한 명칭과 기능을 설명하면 다음과 같다.



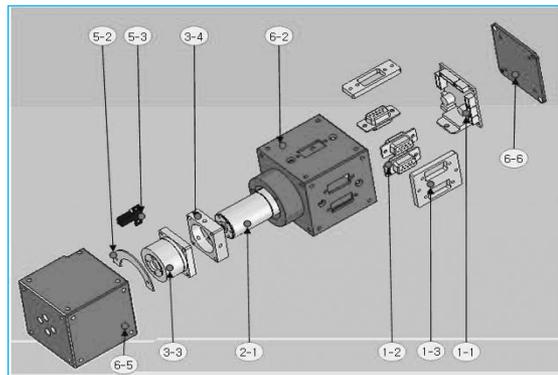
<그림 III-25>

볼 스크루 방식 직선 모듈 요소품 분해도



<그림 III-26>

벨트 풀리 방식 직선 모듈 요소품 분해도



<그림 III-27> 회전 모듈 요소품 분해도



번호	부품명	기능 설명
	제어부	모터를 움직이게 하기 위하여 모터에 전기 신호인 Pulse 전류를 발생시키며, 모터의 위치를 감지하여 피드백 연산을 담당한다.
1-1	1축 제어기	외부의 명령 신호를 CAN통신 방식에 따라서 PC로부터 수신하여 명령에 따라 동작하는 CPU 및 프로토콜을 저장한다.
1-2	커넥터	커넥터를 통하여 로봇 구동에 필요한 신호의 입출력과 전원을 공급받는다.
1-3	커넥터 브라켓	커넥터를 본체와 연결하기 위한 기구물 구조로 일반적으로 커넥터 브라켓이라고 통일된 명칭을 사용한다.
	액추에이터부	전기적인 에너지를 기계적인 회전 운동으로 전하시켜 주는 장치로 일반적으로 모터가 보편적인 장치로 쓰인다.
2-1	모터	로봇의 운동을 일으키는 핵심 동력원으로 속도 및 힘 제어할 수 있는 DC서보 모터로 모터의 위치 및 속도값을 감지하기 위하여 회전 엔코더 센서가 장착되어 있다.
2-2	회전 모듈 소	모터, 센서, 제어기, 감속기 일체형의 회전 모듈
2-3	연결 브라켓	벨트 풀리 방식의 직선 모듈과 회전 모듈 소를 연결하기 위한 연결 기구물
	구동 전달부	액추에이터부에서 발생된 회전 운동을 감속하거나 직선 운동으로 변환시켜주는 장치로, 마찰에 의한 에너지 손실을 최소화하며 백래쉬가 없는 구조가 동력 전달에 유리하다.
3-1	커플링	모터와 볼 스크루를 연결하여 회전 동력을 전달하는 매개체로 모터샤프트와 볼 스크루 사이의 제한된 편심량을 허용하여 동력을 전달할 수 있다.
3-2	볼 스크루	몸체에 나사 형태의 홈이 있어, 산을 따라 마찰을 감소시키기 위한 볼이 내장된 너트에 의하여, 스크루의 샤프트를 회전시킬 경우 볼 스크루 너트의 직선 운동을 얻을 수 있다.

3-3	하모닉 드라이브	소형 정밀 제어용 수평 다관절과 같이 소형 고속 동작용으로 많이 사용하는 감속기로 다음과 같이 구성되어 있다. 유성치차 감속기 대비 백래쉬가 없고, 고회전 정도, 고속 회전이 특징이다.
3-4	타이밍 벨트	미끄럼이 없는 정확한 치형에 의한 높은 전달 효율과 최소한의 백래쉬로 고속으로 동력전달이 가능한 구조
3-5	타이밍 풀리	
	가이드부	가이드부는 직선 운동과 회전 운동이 외부의 페이로드의 하중을 지탱하면서 정해진 정밀도 범위 내에서 운동이 가능하도록 하는 기구 시스템이다.
4-1	직선 불가이드(LM Guide)	직선 운동을 하는 리니어 가이드 역할을 하며, 페이로드의 하중을 지탱한다.
4-2	지지 베어링(Support Bearing)	볼 스크루의 양단을 고정하고 로봇의 베이스 프레임부에 취부되어 볼 스크루의 안정된 회전 운동을 가능케 함. 일반적으로 볼 스크루와 함께 제공된다.
4-3	모터 브라켓	모터를 고정시키기 위한 구조물
4-4	지지 브라켓	지지 베어링을 감싸는 케이스 구조물로 조립 정밀도를 위하여 정밀 가공된다.
4-5	슬라이더(Slider)	볼 스크루의 너트와 LM-가이드의 블록을 연결하여 직선 운동을 가능하게 하며, 로봇 사용자에게 Mechanical Interface를 제공한다
	센서(Sensor)부	로봇의 원점 복귀를 위하여 사용되는 센서이다. 또한 하드웨어적 리미트로 사용되기도 한다. 일반적으로 포토 센서가 많이 사용된다.
5-1	근접 센서	자력의 원리에 의하여 상대물이 근접하였을 때, 감지되어 ON/OFF 제어된다.
5-2	센서 도그	센서를 감지 시키기 위한 기구물
5-3	포토 센서	일반적으로 광학 센서에 의한 빛의 통과 유무에 의하여 ON/OFF 제어된다.
	몸체부	로봇의 몸체는 경량화를 위하여 알루미늄 재질로 구성하였으며 프레임(Frame)



6-1	커버	커넥터가 위치하는 커버
6-2	베이스 프레임(Base Frame)	로봇 구동부의 부품들이 취부되는 알루미늄 소재의 부품이며, 일반적으로 압출물을 사용한다.
6-3	커버	외관 커버
6-4	판넬 커버	외관 커버
6-5	회전 커버	회전 모듈에 적용되며 탈부착이 가능하다. 벨트 풀리 방식의 직선 모듈에서 회전 모듈소의 회전 커버를 제거한 다음 직선 모듈에 조립이 가능한 구조로 되어 있다.

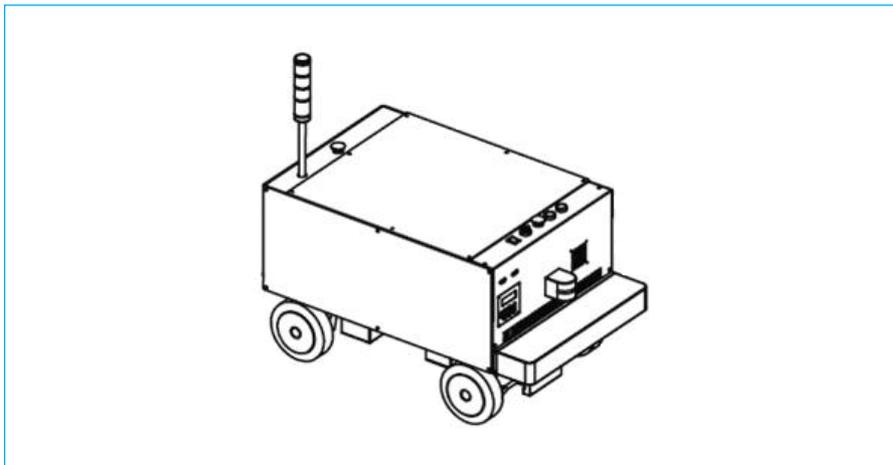
〈표 III-2〉 구성 요소에 대한 기능과 명칭

2 이동 로봇

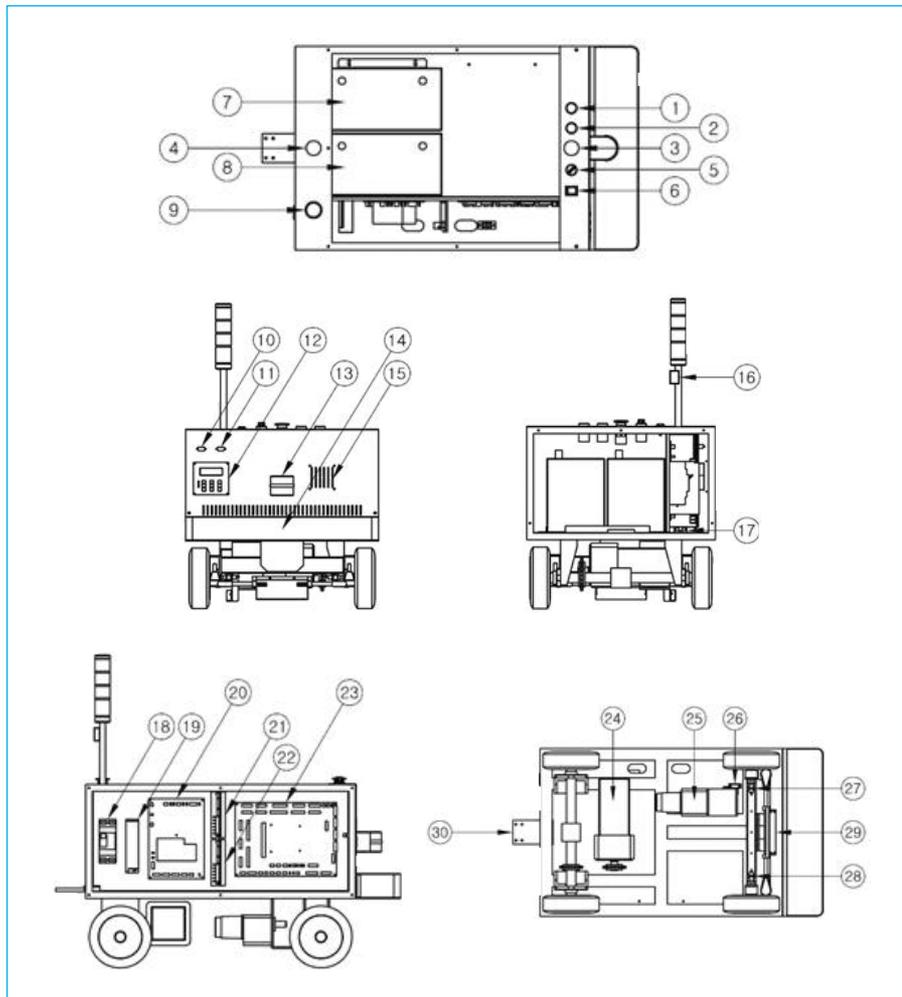
〈그림 III-28〉은 본 교재에서 다룰 AGV의 외관도이며, 〈그림 III-29〉는 주요 부품에 대한 배치도이다. 이 배치도에 표기된 부품들의 목록은 〈표 III-3〉에 정리되어 있다. 각 부품에 대한 세부적인 기능은 다음 장에서 설명한다.

AGV

무인운반차
Automated Guided Vehicle



〈그림 III-28〉 전체 외관도



<그림 Ⅲ-29> 부품 배치도

번호	품명 및 기능	번호	품명 및 기능
1	정지(Stop) 버튼	2	시작(Start) 버튼
3	비상 정지 스위치	4	비상 정지 스위치
5	수동/자동 절환 스위치	6	전원 스위치
7	납축전지(배터리)	8	납축전지(배터리)
9	확장 기능용	10	수동 조작 기기 연결 단자



11	PC 연결 단자	12	LCD 모듈
13	장애물 감지 센서	14	범퍼
15	스피커	16	타워 램프
17	충전 단자	18	전원 차단기
19	제어기 파워 공급 장치	20	파워 제어 보드
21	주행 모터 드라이버	22	조향 모터 드라이버
23	DSP 보드 및 I/O 제어 보드	24	주행 모터
25	조향 모터	26	번지 카운터용 센서
27	오른쪽 끝 감지 센서	28	왼쪽 끝 센서
29	가이드 센서	30	대차 연결부

〈표 III-3〉 부품 리스트

3 창작 로봇

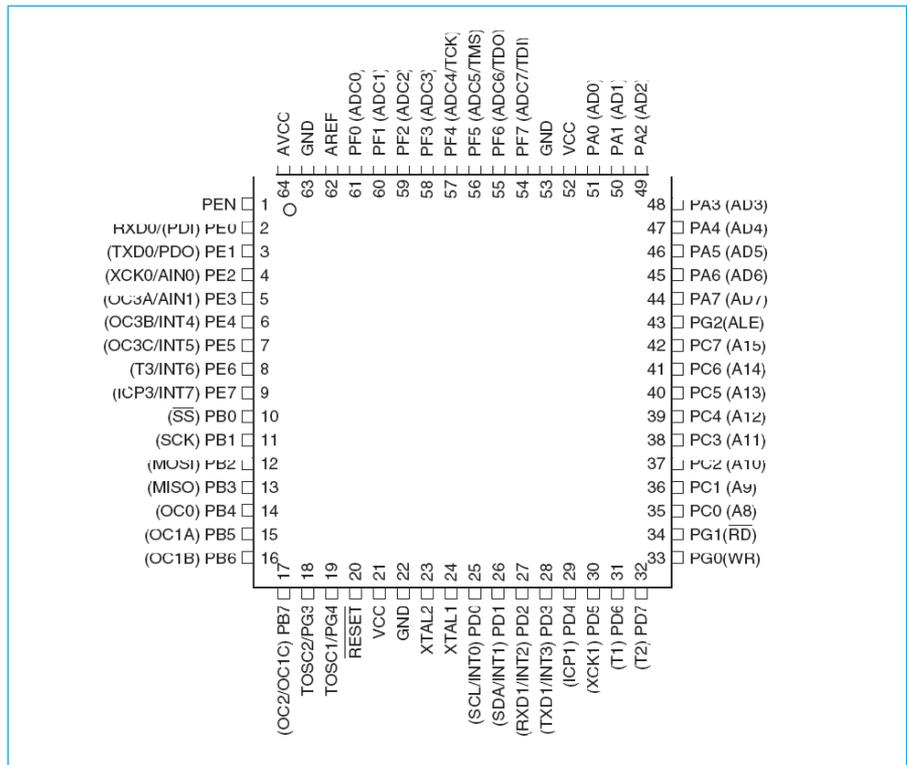
1) AVR

AVR 마이크로 컨트롤러의 탄생 및 특징

AVR 시리즈 마이크로 컨트롤러는 1984년에 창립된 미국의 Atmel사에서 1997년에 처음 발표한 8비트 제어용 마이크로프로세서이다. AVR은 Atmel사의 장점인 플래시 메모리를 내장하고, 여기에 사용자 프로그램을 쉽게 다운로드할 수 있는 ISP(In-System Programming) 방식을 적용하였다. 이러한 장점 때문에 그 당시 시장을 상당 부분 지배하고 있던 8051 시리즈나 PIC 시리즈의 마이크로 컨트롤러를 능가하는 인기를 단기간 내에 얻게 되었다.

- 8-비트 RISC(Reduced Instruction Set Computer) 구조로 명령어가 간단하며 동작 속도가 빠르다
- 1MHz당 약 1MIPS(Million Instruction Per Second)의 성능을 보인다.
- 소비 전력이 적다.

- 10 비트 ADC를 내장하고 있다.(일부 모델에 한함)
- 다른 마이크로 컨트롤러에 비해 큰 SRAM을 가지고 있다.
- 플래시 메모리의 내장으로 프로그래밍이 용이하다.
- EEPROM을 내장하고 있어서 데이터 백업이 가능하다
- C언어에 최적화된 설계
- 8비트 및 16비트 타이머를 내장하고 있다.
- UART, SPI(Serial Pheripheral Interface), PWM(Pulse Width Modulation) 등을 내장하고 있다.
- 소비전력이 적다.



<그림 Ⅲ-30> ATmega128의 핀 구조



2) RC 서보 모터

RC 서보 모터의 특징

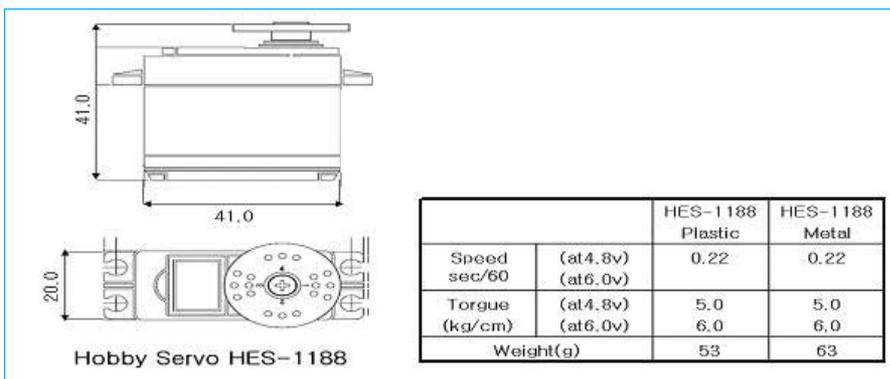
내부에 DC모터를 탑재하고 있으나 일반 DC모터와 달리 펄스 길이(PWM : Pulse Width Modulation)로 회전각을 조절한다. 제품마다 약간의 차이가 있으나 일반적으로 모터의 동작 범위(각도)는 +90도 ~ -90도(180도)이다. 동작 속도는 각각의 모터의 기종에 따라 다르며, 동작 전압은 4.8V ~ 6V 사이 전압을 일반적으로 인가한다.

HES-1188의 특징

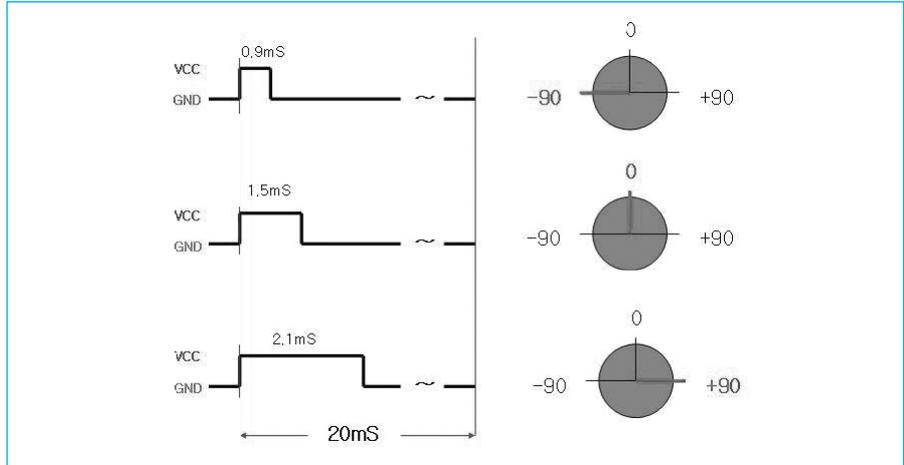
- 제어 방식 : PWM 제어
- 동작 전압 : 4.8V ~ 6.0V
- 동작 온도 : -20 ~ 60°C
- 토크 : 4.8V에서는 5.0Kg, 6.0V에서는 6.0Kg
- 동작 범위 : -90° ~ +90° (180°)
- 무게 : 53.0g(Plastic), 63.0g(Metal)
- 모터의 선은 총 3가닥이며, 선의 색깔 별 역할은 주황색-신호(Signal), 빨강색-전원(VCC), 갈색-접지(GND)이다.

HES-1188

하비전자에서 제작한 RC 서보모터로서 RC모형, 소형관절로봇의 제작 등에 많이 사용되고 있다.



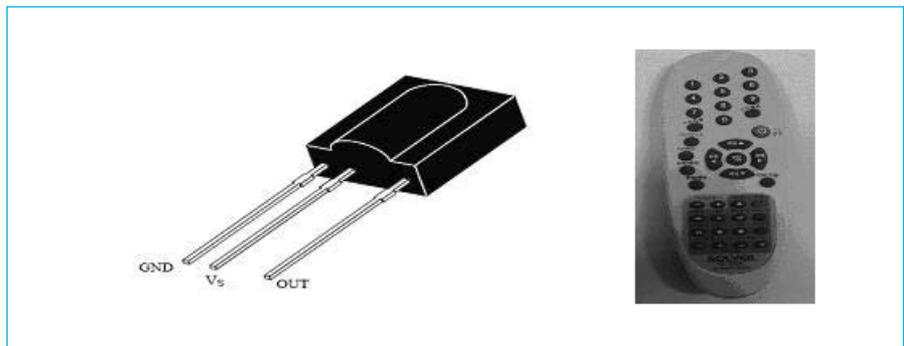
(그림 III-31) HES-118의 외형



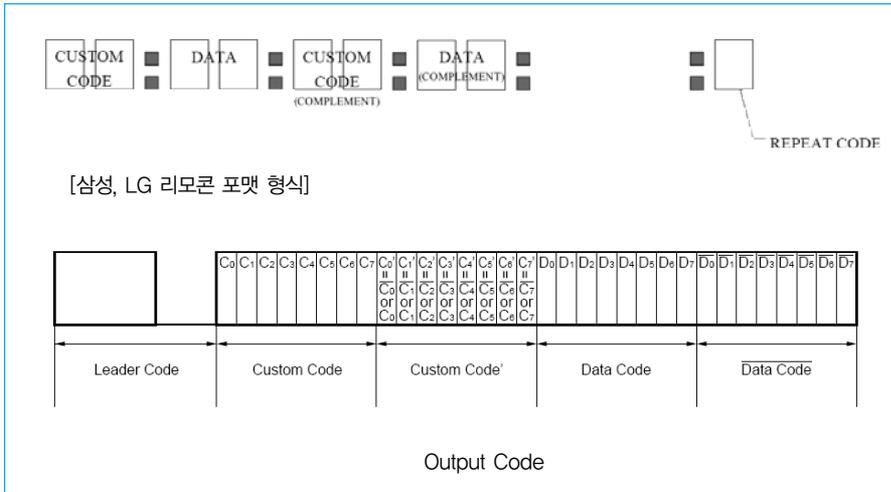
〈그림Ⅲ-31〉 서보 모터의 동작

3) 리모트 컨트롤

일반 TV 리모컨의 적외선 통신을 잘만 응용한다면 매우 효과적인 로봇 명령 송수신기로 사용할 수 있다. 이 절에서는 이러한 리모컨의 송신/수신 방법을 이해하고 로봇에 적용하여 보자.



〈그림Ⅲ-32〉 (좌) TSOP1738(IR 수신 센서), (우) 리모컨



〈그림 III-33〉 리모컨 출력 형식

TSOP1738 수신 출력부에는 리더 코드(Leader Code), 커스텀 코드(Custom Code), 데이터 코드(Data Code) 이렇게 3개의 코드가 출력된다. 여기서 리더 코드는 제외하고, 커스텀 코드는 포트를 제어하는 데이터로서는 쓸모가 없다. 커스텀 코드는 장치의 종류를 분별하기 위해서 보내는 정보 즉 TV, 비디오, 오디오, 프로젝터 등의 종류를 나타내기 때문이다. 따라서 본 로봇에 사용할 때는 필요가 없다. 데이터 코드는 Data code, 반전된 Data code 로 출력되어 출력된 데이터가 바른 신호인지를 구별할 수 있다. 이렇게 출력된 데이터 코드를 가지고 본 로봇을 제어한다.

4) 초음파 센서

① 초음파를 이용한 거리 측정의 원리

초음파란 일반적으로 인간의 귀에 들리지 않는 높은 주파수의 음(약14KHz이상)을 통칭하며 일컫는다. 그러나 인간의 귀로 듣는 것을 목적으로 하지 않는 공업 분야에 적용하는 가청음을 초음파라 부르기도 한다. 따라서 초음파 센서

란 “음향 에너지 중에서 비교적 주파수가 높은 영역을 검출하기 위한 센서로서 20KHz 이상의 음향 에너지를 검출하는 소자” 라고 정의할 수 있다. 이 종류의 센서 재료는 압전의 정효과 및 부효과를 동시에 이용하는 압전 세라믹 소자를 이용하는 경우가 많다. 초음파 센서에 적용되는 주파수는 그 사용 용도에 따라 결정되는 것이 원칙이나 일반적으로 공기 중 물체 감지에 적용되는 초음파 센서는 9KHz 에서부터 50KHz 정도의 주파수를 사용한다. 그 이유는 이 범위의 주파수가 강력한 초음파 펄스를 발생하기 쉽고 지향 특성을 얻기가 용이하기 때문이다. 초음파 센서의 측정 원리는 초음파 센서에서 발사된 초음파 펄스가 피 측정물의 표면에서 반사되어 다시 초음파 센서로 되돌아올 때까지의 시간을 측정하여 측정 대상에 따른 정보를 얻는 방법을 사용한다. 즉 초음파 송신 후 되돌아 올 때까지의 지연 시간을 측정하고 공기 중에서 초음파의 온도에 따른 음속을 보상하여 거리를 산출하는 방법을 사용한다. 일반적으로 공기 중에서의 음속은

$$C \approx 20\sqrt{T} \approx 331.5 + 0.6\theta [m/s]$$

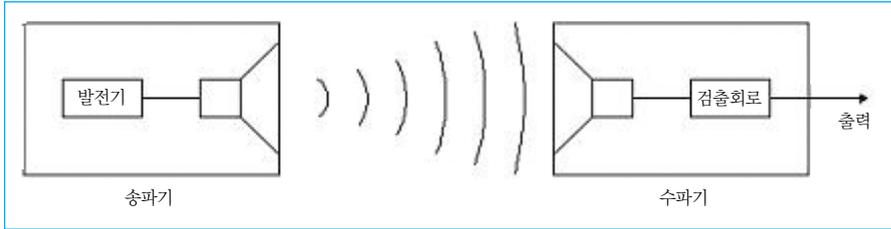
로 산출하며 여기서 θ 는 온도[$^{\circ}C$]이다. 위 식에서 공기 중에서는 온도가 1[$^{\circ}C$] 변화하면 음속은 0.17% 변화하기 때문에 기체의 온도를 측정하여 음속을 보상하는 것이 필요하다. 거리측정은 다음의 식으로 간단히 구할 수 있다.

$$H = (u \times t) \div 2$$

H : 초음파 센서에서 측정 물체까지 거리, u : 기체 음속(공기), t : 소요 시간[sec]

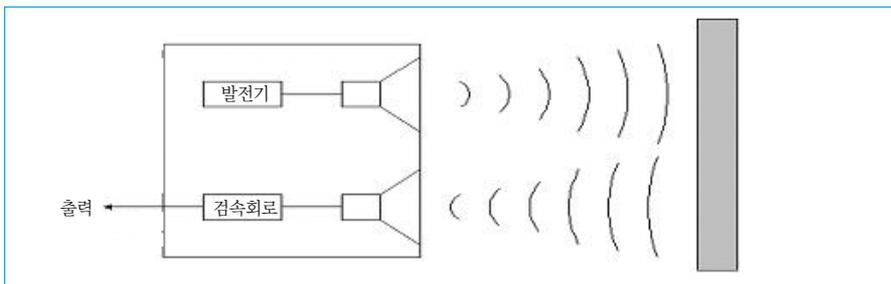
② 초음파 센서의 종류

일반적으로 초음파 센서의 종류에는 <그림 Ⅲ-34>과 같은 대향형과 <그림 Ⅲ-35> 같은 반사형이 있다. 대향형은 광전 스위치의 투과형과 같이 송파기와 수파기를 마주 보게 하고, 송파기로부터 방출된 초음파가 물체에 의해 차단될 때 수파기측의 신호 변화를 검출하는 방식이다.



〈그림 III-34〉 대향형 초음파 센서

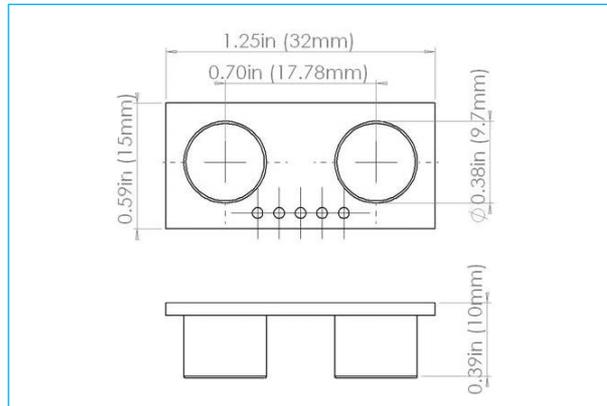
반사형은 〈그림 III-35〉 같이 한 쌍의 송파기와 수파기를 동일 유니트로 구성한 것이 반사형이다. 송파기에서 방출된 초음파는 공기보다 고밀도의 물체 즉, 대부분의 물체에서 반사되기 때문에 이 반사파를 수파기로 검출한다. 또, 이 반사기는 단순히 물체의 유무를 검출할 뿐만 아니라 물체까지의 거리 측정이 가능하다. 초음파를 발생시키고 이와 동시에 숫자 세기를 시작하여, 초음파가 돌아온 것이 감지되면 세는 것을 멈춘다. 이 때 세어진 값을 읽어 초음파가 되돌아오는 데 걸린 시간을 계산하여 거리를 측정하는 것이다. 이것은 일종의 스톱 워치(stop watch)로 시간을 잰다고 생각하면 된다. 숫자를 세는 데 걸리는 시간이 1이라면, n번 세는 동안에 초음파가 되돌아 왔을 때 걸린 시간은 n이 된다. 소리의 속도는 340m/sec(기온 15℃ 기준)이므로 거리는 속도×시간으로 구할 수 있다. 초음파는 벽과 센서 사이의 왕복 거리이므로 벽까지의 거리는 $(340 \times n \div 2)[m]$ 가 된다. 이때 마지막에 나누기 2를 한 것은 측정할 것이 왕복 시간이기 때문이다.



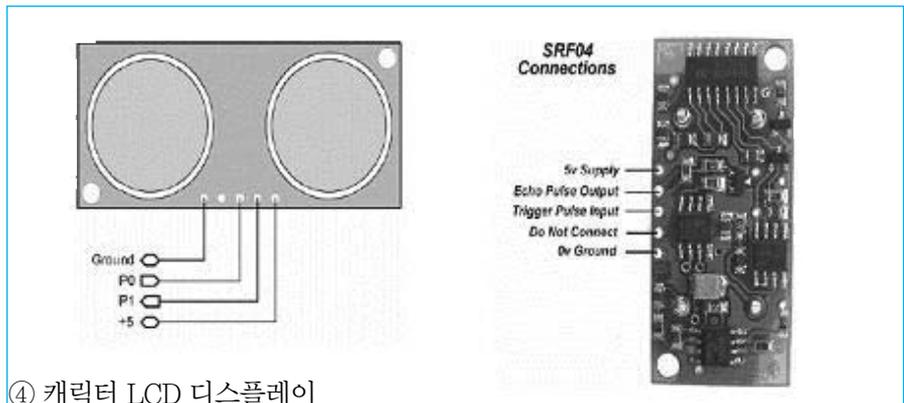
〈그림 III-35〉 반사형 초음파 센서

만약, 숫자 세기에 사용한 카운트 변수가 16비트라면 클럭이 1[MHz]이므로 최대 카운트가 가능한 시간은 65,535msec가 되고, 최대 측정 거리는 $340 \times (65.535 \times 0.001) / 2 = 11.14095\text{m}$ 가 된다. 1msec의 시간 간격으로 카운트 되므로 $11.14095 / 65535 = 0.00017\text{m}$, 즉, 0.17mm의 정밀도를 갖는다.

③ SRF-04의 외부구조 및 사양



〈그림 Ⅲ-36〉 SRF-04 구조(외부크기)

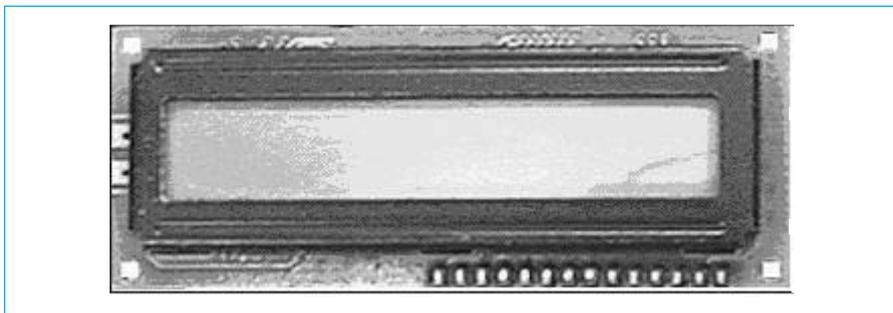


④ 캐릭터 LCD 디스플레이

〈그림 Ⅲ-37〉 SRF-04 사양(외부 연결핀)



LCD(Liquid Crystal Display)란 디스플레이 액정으로 각종 전자제품 표시창, 공장 자동화 시스템의 상태 디스플레이 부품으로서 가장 널리 쓰이는 장치이다. 이러한 액정을 이용하여 화소에 도달하는 빛을 선택적으로 투과시키거나 차단시켜 문자나 화상 등을 표시할 수 있는 시각적인 전달 장치이다.



〈그림 III-38〉 캐릭터 LCD (2×16) 디스플레이

- 4비트, 8비트의 마이크로프로세서와 인터페이스 가능
- 5×8 도트, 5×10 도트의 디스플레이 가능
- 80×8 비트의 디스플레이 RAM(D.D RAM : 최대 80 글자까지)
- 240 글자 폰트를 위한 9,920비트 글자 발생기 ROM(C.G ROM)
 - ≫ 208 글자 폰트(5×8 도트)
 - ≫ 32 글자 폰트(5×10 도트)
- 64 × 8 비트 글자 발생기 RAM(C.G RAM)
 - ≫ 8 글자폰트(5×8 도트)
 - ≫ 4 글자폰트(5×10 도트)
- +5V단일 전원

4 로봇 제작 계획 수립(Process)

1 직교 로봇, 스카라 로봇

로봇이 사용될 작업 환경에 따라 로봇의 사양을 결정하는 기본 요건인 제어 축수, 가반하중, 작업 영역, 반복 정밀도, 제어 방법 등이 달라진다.

1) 로봇의 축 수(Number of Axis)란?

① 로봇의 축 수는 로봇의 용도에 따라 정해지며, 로봇의 가능 작업을 결정한다.

② 주어진 작업은 그 고유의 자유도를 갖는다. 이 자유도에 대한 이해가 바로 작업 분석의 시작이자 로봇 선정의 시작이라고 볼 수 있다. 3차원상의 집기(Pick) & 놓기(Place) 작업의 경우 XY 평면상에 위치하기 위한 2자유도와 Z방향으로 물건을 들기 위한 1자유도가 필요하다. 그리고 로봇 작업 대상물의 형태에 따라서는 손목부에서 회전이 필요하므로 회전 1자유도가 필요하다. 또한 물건을 잡기 위해서는 핸드(Hand)에 의한 그립(Grip) 방향으로 1자유가 필요하다. 그러므로 총 5축이 필요하게 된다.

* 예시 *

Pick & Place의 경우 직교 로봇의 경우 XYZ 공간에서 3자유가 필요하며 각각 3방향에서 직선으로 만나는 직선 모듈이 필요하다. 그리고 엔드 이펙트 부위에 물체의 방향에 따른 회전을 하기 위한 회전 모듈 1축과 그리퍼 모듈(Gripper Module) 1축이 필요하다.



스카라 로봇의 경우 XY 평면상에서는 회전 모듈 2축의 조합에 의하여 위치 점에 도달하게 되면 Z축 방향으로 직선 모듈 1축이 필요하게 된다. 그리고 엔드 이펙트 부위에 물체의 방향에 따른 회전을 하기 위한 회전 모듈 1축과 그리퍼 모듈 1축이 필요하다.

2) 로봇의 가반하중(Maximum Load Capacity)이란

① 로봇의 가반하중은 엔드 이펙터에 달 수 있는 핸드 및 작업물을 포함한 중량을 나타낸다.

예시

직선 3축의 엔드 이펙터에 달 수 있는 회전 모듈과 그리퍼 모듈 및 작업 대상물 하중의 합이 직교 3축 로봇의 가반하중이 된다.

스카라 로봇의 엔드 이펙터에 달 수 있는 회전 모듈과 그리퍼 모듈 및 작업 대상물 하중의 합이 스카라 로봇의 가반하중이 된다.

3) 로봇의 작업 영역(Motion Range)이란?

① 로봇이 작업을 할 수 있는 가상의 공간이다.

예시

직선 로봇과 스카라 로봇의 그리퍼 모듈 끝에 펜을 장착하여 그림을 그려보면, 최대한 도달할 수 있는 XY 평면상의 공간에 차이가 발생한다. 이 결과물의 차이는 로봇의 구조적 특징에 나타난다.

그리퍼(Gripper)

로봇 엔드 이펙터 중의 일종으로서 물체를 집기 위한 기구 물이다. 대부분 모터나 공압/유압 장치를 이용하여 구현된다.

4) 로봇의 반복 정밀도(Repeatability)란?

- ① 로봇의 명령점과 실질적으로 도달된 점과의 반복된 정도차로 사용자의 작업 종류에 따라 로봇의 사양을 결정할 수 있다.

* 예시 *

로봇의 반복 정밀도는 정밀한 측정이 필요하다. 산업용에 준하는 모듈 부품의 적용 시 시각적으로는 판단할 수 없다. 그러므로 그리퍼 끝에 인디케이터를 장착하여 일정거리를 왕복 운동시켜서 약 30회 반복 운동으로 측정되는 데이터의 평균에 대하여 편차를 계산하면 로봇의 반복 정밀도를 측정할 수 있다.

5) 로봇의 최대 속도(Max speed)란?

- ▶ 로봇 엔드 이펙터의 최대 합성속도를 나타내며, 로봇의 작업 Tack Time 결정에 영향을 미친다. 또한 가반하중이나 로봇의 감속비 결정 등과 같은 설계상의 문제점들과 밀접한 관련이 있다.

* 예시 *

로봇의 최대 속도가 빠르다고 반드시 좋은 것은 아니다. 그러나 로봇의 사양 결정시 최대 속도를 측정해 봄으로써 로봇이 정해진 시간 내에 할 수 있는 작업횟수가 결정된다.

6) 로봇 제어 방법이란?



- ▶ 목적에 맞는 로봇을 구동시키기 위해서는 작업 방법에 대한 순서도가 필요하다. 그리고 제어기 구동을 위한 적절한 변수 설정과 프로그래밍을 하게 된다.

예시

로봇의 작업 순서는 Pick & Place의 경우 프로그래밍이 필요하다. 프로그래밍의 순서에 의하여 로봇 작업의 효율이 달라진다. 뿐만 아니라 로봇의 수명에도 많은 영향을 미친다.

2 이동 로봇

1) 역할 분담

이동 로봇 제작에 필요한 일들을 구분하면 크게 아래와 같다. 인력 상황 및 진행 일정에 따라 역할을 구분하여 진행하여야 한다.

- ① 전기 배선도 : 전체 결선도를 이해하고 잘못 배선되지 않도록 특별한 주의가 필요함. 잘못 결선된 경우 부품 파손 및 화재 등 위험한 상황이 발생할 수 있다.
- ② 기구부 : 센서 배치 및 고정, 센서 감도 조정 등 부품을 조립하는 역할을 담당한다.
- ③ 루프 및 태그 설치 : AGV가 주행할 루프 및 태그를 규격에 맞게 설치하며, 이 루프 및 태그의 설치 상태에 따라 AGV의 주행 성능에 차이가 있을 수 있다.
- ④ 제어 및 프로그램 : 조립된 AGV를 조작하고 프로그램하는 역할을 한다.

2) 준비물

- ① AGV 관련 부품 : 배터리(Battery) 외 18개 품목
- ② 조립 공구 : 스패너, 드라이버 세트, 육각 렌치 세트 등

- ③ 기타 소모품 : 케이블 타이, 절연 테이프
- ④ 기술 자료 : 전체 결선도, 전원 계통도, AGV 사용 설명서, 기구 조립도, 부품 리스트
- ⑤ 프로그램 개발 툴 : Windows XP, 시리얼 통신 케이블, 캔탑스에서 제공한 프로그래밍 툴인 ctAGV를 설치

3) 점검표

〈표Ⅲ-4〉는 AGV 제작에 필요한 부품 및 제작 단계에서 각 업무별로 중점적으로 확인해야 할 사항에 대해 정리하였다.

항목	점검 내용	결과
부품 준비	1. Battery 2개 2. Bumper 1개 3. Tower Lamp 1개 4. 스피커 1개 5. LCD 모듈 1개 6. 장애물 센서 1개 7. 광센서 1개 8. Power Control 보드 1장 9. 모터 드라이버 2개 10. I/O Control 보드 1장 11. DSP 보드 1장 12. 주행 및 조향 모터 13. 리미트 센서 2개 14. 가이드 센서 1개 15. Tag 리드 센서 16. Joystic 17. 써스 테이프 : 직선 20M, 회전(R 1100) 24장, Tag용 10장 18. 온도 센서 19. 케이스 커버:상판, 좌판, 우판	



부품 조립	<ol style="list-style-type: none"> 1. 준비된 부품의 조립 상태 2. 케이블의 고정 상태 	
전장 배선	<ol style="list-style-type: none"> 1. Battery 극성 체크 2. Power Control용 결선 체크 3. ICon 및 DSP 보드 결선 4. 모터 드라이버 결선 	
센서 튜닝	<ol style="list-style-type: none"> 1. Tag 센서의 설치 위치 및 각도 2. 가이드 센서의 설치 위치 및 각도 3. 로봇 동작시 케이블 간섭이 없도록 고정 	
루프 및 Tag 설치	<ol style="list-style-type: none"> 1. 회전 반경(1100mm) 확인 2. 루프가 끊김 없이 잘 만들어졌는가? 3. 직선 구간에서의 선로가 일직선인가? 4. Tag의 위치가 규정에 맞는가? 	
동작 시험	<ol style="list-style-type: none"> 1. 전원 투입 : 과전류 2. 수동 조작으로 전/후/좌/우 동작 3. 프로그램 선택 후 자동 동작 	

〈표 III-4〉 부품 및 제작 단계

3 창작 로봇

1) 역할 분담

뱀 로봇 제작 실습에서는 아래의 로봇 제작 순서에 따라 각 단계에 맞춰 인원 배치 및 역할 분담을 한다.

2) 로봇 제작 순서

① 기구부 조립

기구부는 크게 2가지로 로봇 프레임 부분과 마이크로 컨트롤러 조립부분으로 나눌 수 있다. 로봇 프레임 부분은 모터와 로봇 몸통 프레임과 조립 바퀴 몸통 프레임들과의 연결 과정을 거치고 마이크로 컨트롤러 조립 부분은 상하 마이크

로 컨트롤러 조립 초음파 센서, LCD 조립 등으로 나눌 수 있다.

② 배선

뱀 로봇의 조립이 완성되면 모터와 마이크로 컨트롤러와 연결하고 뱀 로봇을 움직이기 위한 충전기 홀더와 충전기를 조립하여 로봇의 안쪽에 넣는다.

③ 센서 테스트

기구부 및 배선 조립이 완료되면 시연을 위한 프로그램을 다운로드시켜 초음파 센서와 리모컨 센서의 동작을 확인하고 초음파 센서의 결과가 LCD에 출력되는지 확인한다.

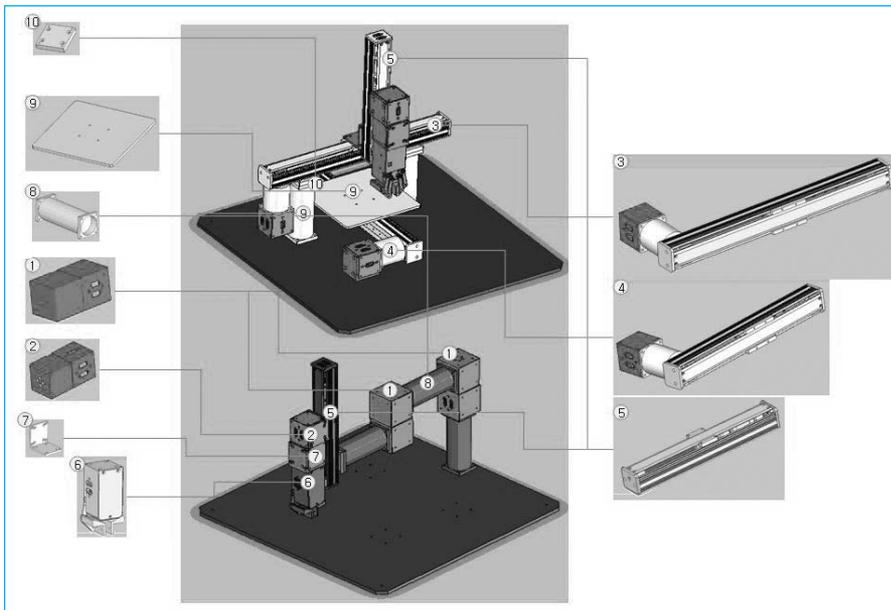
④ 에뮬레이터 실습 및 C언어 실습

에뮬레이터는 시리얼 통신을 이용해서 실시간으로 각 모터를 제어할 수 있는 버튼식 방법으로 뱀 로봇에 프로그래밍할 수 있다. 또한 C언어로도 로봇을 제어할 수 있다. 로봇을 통해서 C언어의 개념과 응용을 배울 수 있는 것이다.



5 로봇 H/W 설계 기초

1 직교 로봇, 스카라 로봇



〈그림 III-39〉 직교 로봇 및 스카라 로봇 외관도

1) 부품 목록 및 재고 체크

부 품	수량	부 품	수량
① 회전 모듈 대	2	⑦ 연결 브라켓	2
② 회전 모듈 소	1	⑧ 연결 링크	3
③ 직선 모듈 대	1	⑨ Moving Plate	1
④ 직선 모듈 중	1	⑩ 브라켓	3
⑤ 직선 모듈 소	1		
⑥ 그리퍼 모듈	1		

〈표 III-5〉 직교 로봇 및 스카라 로봇 부품 목록

2) 조립 도면 작성 및 주의 사항

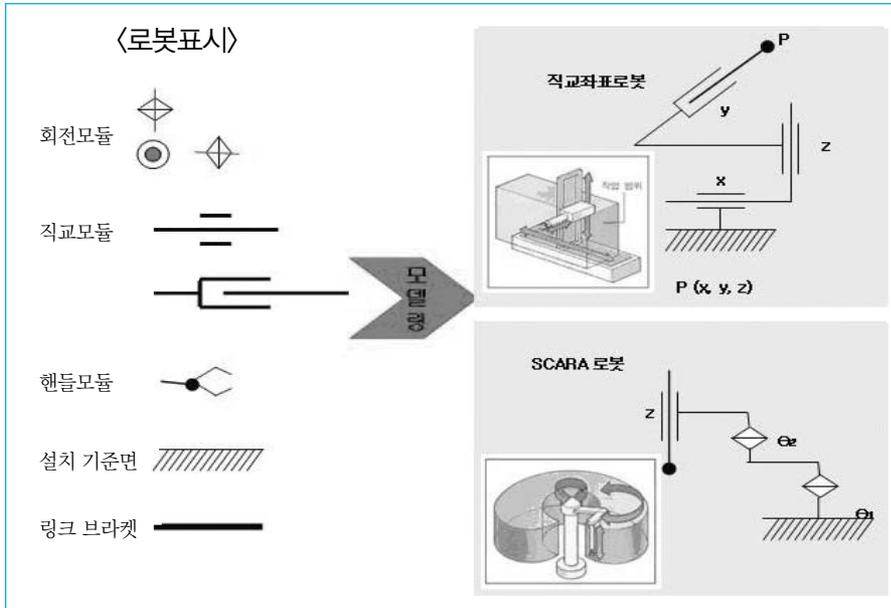
- ▶ ①회전 모듈 대는 1축과 2축 용으로 사용 가능하다.
- ▶ ②회전 모듈 소는 3축 이상에서 사용 가능하다.
- ▶ ③④⑤직선 모듈과 ⑧연결 링크 조립 시 ⑦연결 브라켓을 이용하여 조립.
- ▶ ②회전 모듈 소와 ⑥그리퍼 모듈 조립 시 ⑦연결 브라켓을 이용하여 조립
- ▶ ①②③④⑤모듈간의 케이블 연결은 In, Out 커넥터 구분이 없다.
- ▶ 볼트는 동일 사양으로 M4 렌치머리 볼트이다. 볼트 조립 시 스프링와셔와 평와셔를 사용하여 조립한다.

3) 로봇 설계 사양 정의

- ① 로봇의 용도를 정의한다.
- ② 로봇의 용도에 맞는 축 수를 결정한다.
- ③ 축 수 및 용도에 맞는 재고를 가지고 있는지 파악한다.
- ④ 로봇 모델링 설계를 한다.
- ⑤ 축에 대한 사양 계산을 한다.
- ⑥ 구체적인 사양을 정한다.
- ⑦ 구체적인 사양에 맞는 조립도면을 작성한다.
- ⑧ 로봇의 조립 방법을 정의한다.
- ⑨ 로봇의 제어 파라미터를 셋팅한다.
- ⑩ 로봇의 작업 프로그래밍을 한다.
- ⑪ 실제로 구동해 봄으로써 처음 의도와의 차이점을 확인한다.



4) 로봇 조립도 구성



<그림 III-40> 로봇 조립 모델링

모델링

구현하고자 하는 시스템을 기호나 수식등을 이용하여 표현하는 것으로서, 시스템을 이해해 해석하는데 있어서 중요한 과정이다.

2 이동 로봇

이동 로봇 설계에 사용되는 주요 부품 및 회로와 AGV 사양 가운데 중요한 용어에 대해 설명하면 다음과 같다.

단 계	내 용
비상 정지 (E-Stop)	이상 동작 시 이 버튼을 누르면 일정 시간 후 모터에 공급되는 전원을 차단하는 기능이다. 프로그램에서 먼저 감속 및 정지를 수행하고, 일정 시간 지연 후 하드웨어적으로 전원을 차단한다.
범퍼회로	AGV 전면에 부착되며 충돌 시 감지되며, 사용 방법에 따라 2선식(범퍼 동작만 검출)과 4선식(범퍼 동작과 결선 이상 상태도 검출 가능)이 있다.
자동 전원 차단 기능	이동 로봇의 경우 배터리가 과방전되는 경우 배터리 수명이 떨어지기 때문에 일정 전압 이하로 배터리 전압이 떨어지거나 치명적인 문제가 있는 경우 자동으로 전원을 차단해 주는 기능이다.

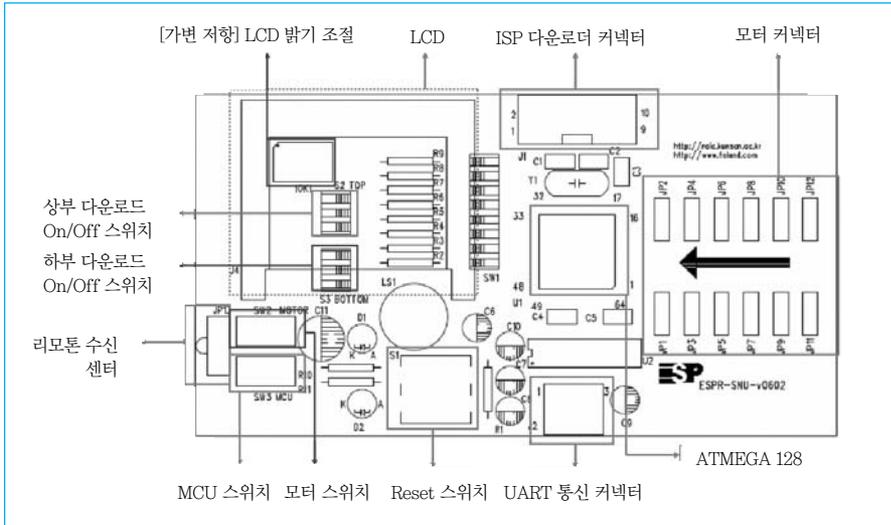
장애물 감지 센서	범퍼가 동작하기 전에 AGV 전면에 장애물의 유무를 판단하며, 센서 중에는 1차, 2차 감지 거리를 설정할 수 있어 장애물에 따른 AGV의 속도를 부드럽게 제어할 수 있다.
시스템 이상 체크 기능 (Watchdog)	프로세서가 탑재된 전자 회로에서는 외부 노이즈나 환경에 따라 오동작할 가능성이 있다. 이러한 오동작에 대비하기 위해 프로세서의 이상 유무를 판단하여 이상이 있는 경우 모터에 공급되는 전원을 차단시키는 기능이다.
노이즈 및 정전기	이동 로봇의 경우 노이즈 발생원은 크게 모터 구동부에서 발생하는 전기적인 노이즈와 이동 로봇이 움직이면서 발생하는 정전기이다. 이 영향을 최소화하기 위한 회로 설계 기술이 필요하다.
배터리 및 충전 관련	사용되는 배터리는 납축전지이며 최대 50% 이하로 과방전 되는 경우 수명이 떨어질 수 있다. 이를 위해서는 정밀하게 전압을 측정하는 회로가 필요하며 항상 충전율을 체크하여 배터리가 과방전되지 않도록 한다.
음성 출력 및 주행음	안전을 위하여 이동 로봇이 움직일 때 주변 사람들이 로봇의 유무를 알 수 있도록 노래나 소리를 내는 기능이다.
배터리 전압 및 전류, 온도 측정	배터리를 보다 안정적으로 오래 사용하기 위해서는 현재 전압, 사용 전류, 온도를 알아야 하며, 이를 위해서 아날로그 신호를 디지털 신호로 변환해 주는 ADC 변환기가 필요하다.
가이드센서 및 Stop 센서	이동 로봇의 길과 주행하고 있는 위치(어드레스)를 알 수 있도록 사용하는 센서이다. 본 교재의 AGV에는 IR 방식의 센서를 사용한다.
시리얼 통신 (Update)	RS-232C 통신을 통해 윈도 프로그램에서 설계된 AGV 프로그램을 다운로드하거나 AGV 상태 정보를 PC에 보낼 수 있다.
수동 조작기 (TP)	아래 그림과 같이 일반 게임기용 조이스틱(Joystick)을 개조하여 AGV를 전/후/좌/우로 원하는 위치로 이동할 수 있다. 동작 방향은 AGV 차체를 기준으로 한다. 수동 조작은 Joystick의 전면면에 있는 D_M(Dead_Man) 버튼을 항상 On시킨 상태에서만 가능하다.
모터 제어 회로	주행 및 조향 모터의 속도를 제어하기 위해 프로세서에서 디지털로 값을 주면 이를 아날로그로 변환해 주는 DAC 변환기가 사용된다.
리미트 센서	조향 모터를 제어할 때 기구적으로 회전할 수 있는 한계임을 알 수 있게 하여 모터에 과전류가 흐르지 않도록 한다.

〈표Ⅲ-6〉 이동 로봇 하드웨어 기술

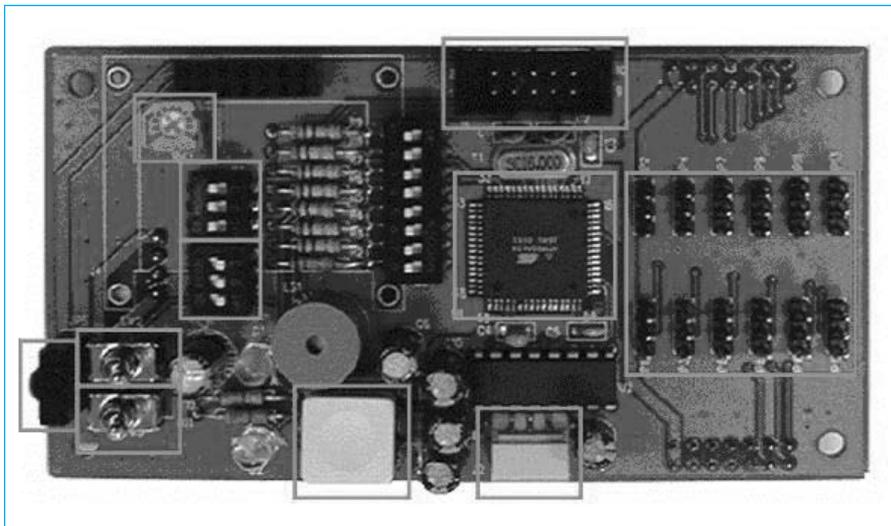


3 창작 로봇

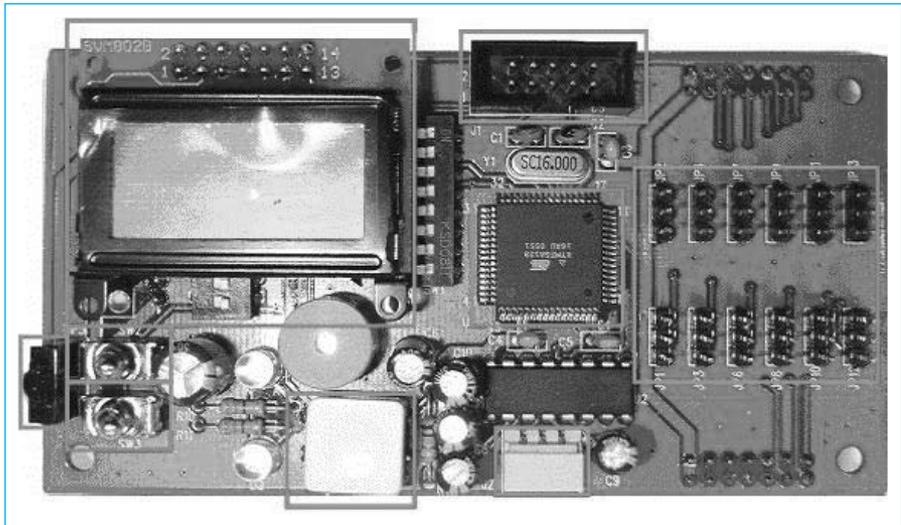
1) 상부 보드 구성



〈그림 III-41〉 상부보드 설명

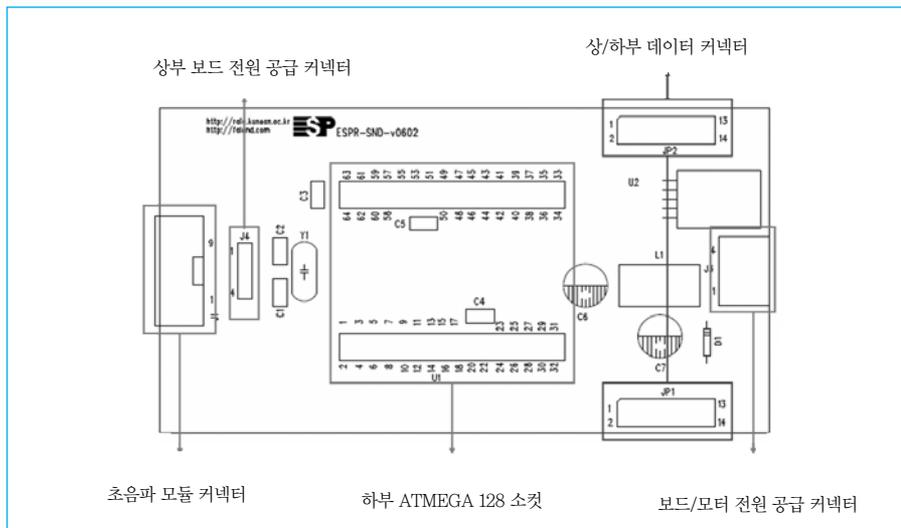


〈그림 III-42〉 상부 보드 실물도

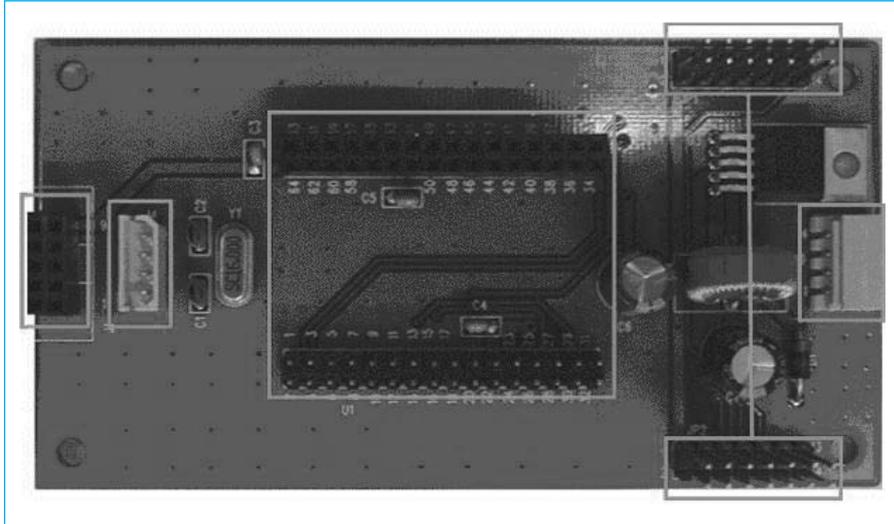


<그림 III-43> 상부 보드 실물도2

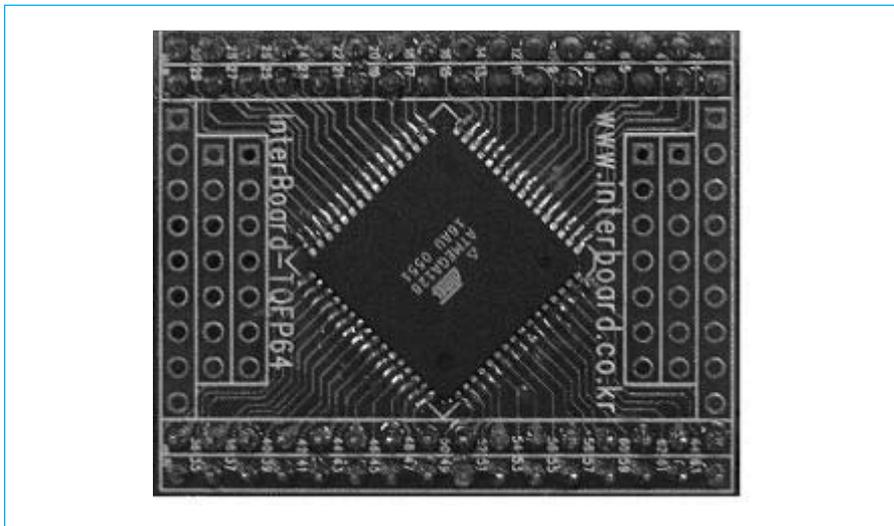
2) 하부 보드 구성



<그림 III-44> 하부 보드 설명



〈그림 Ⅲ-45〉 하부 보드 실물도

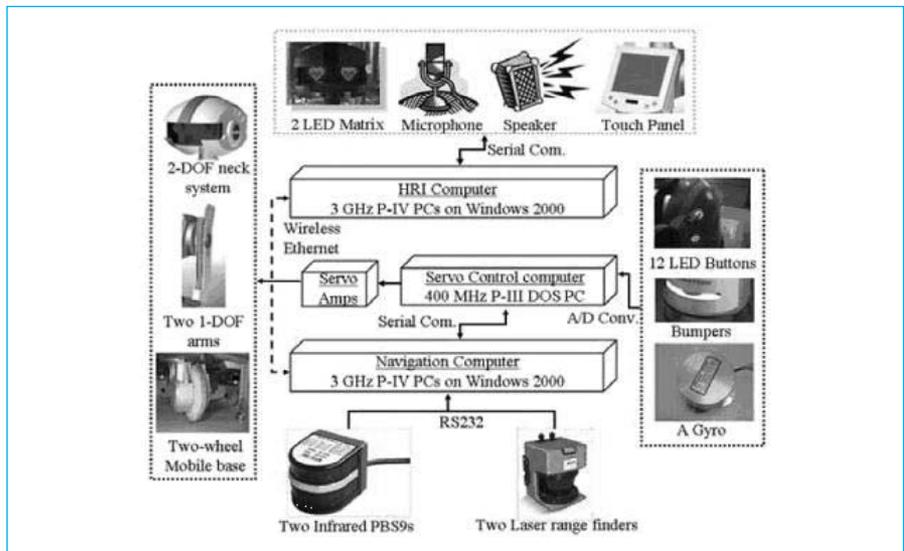


〈그림 Ⅲ-46〉 하부 마이크로 컨트롤러

6 로봇 S/W 설계 기초

1 직교 로봇, 스카라 로봇

1) 블록 다이어그램 및 흐름도의 이해

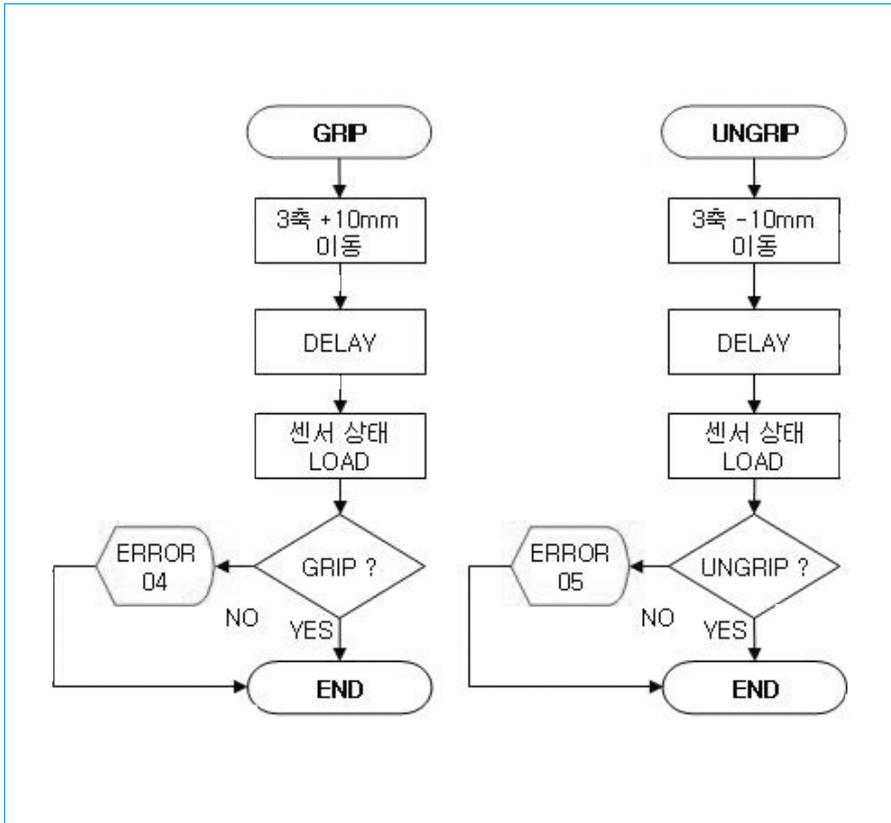


〈그림Ⅲ-47〉 로봇 동작 블록 다이어그램

블록 다이어그램이란 시스템 전체의 동작 관계를 블록 형태로 전개해 놓은 것이다. 블록 다이어그램만 보아도 전체 시스템이 어떻게 연결되어 있는지 확인할 수 있다.

각 블록은 특수한 목적과 관계되어 있어서 특정 장비 또는 특정 부품, 특정 프로그램으로 나눌 수 있으며 대부분은 입력에 의한 결과값의 변화가 나타나는 부분을 블록으로 만들어서 작업을 한다.

또, 신호의 흐름을 나타내기 위해서 선으로 연결된 부분에 화살표를 표시해서 신호의 입출력 방향을 나타내기도 한다.



〈그림 III-48〉 흐름도 예

블록 다이어그램이 전체 시스템의 상호 연관성을 나타내는 것이라면 흐름도는 시스템이 특정 동작을 할 때 초기 단계부터 시간의 흐름에 따라 어떤 동작을 하며 어떤 영향을 주는지를 표현한 것이다.

흐름도에서 사용하는 도형은 〈표 III-7〉과 같이 의미를 갖고 있다.

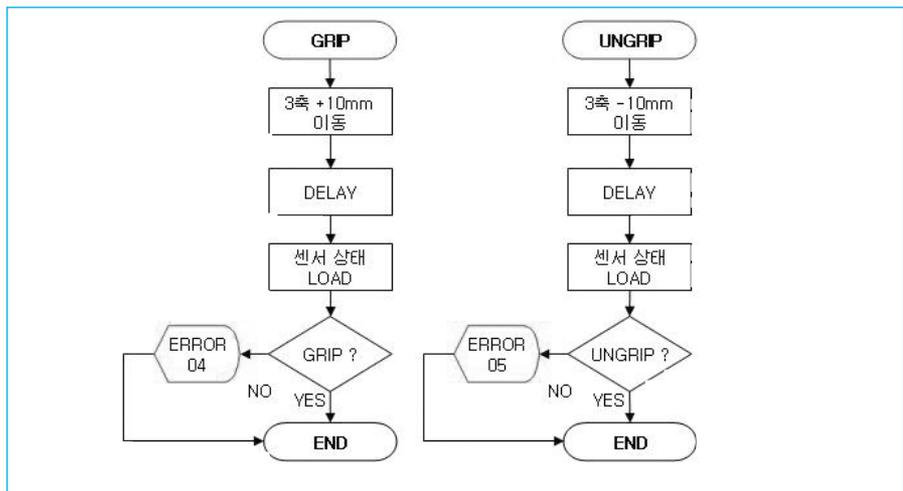
도형	내용	도형	내용
	수행의 시작/종료: 처리 과정의 시작과 끝을 나타낸다.		연결자: 순서도 내에서 상호 참조하거나 다른 처리 과정으로 안내하는 선이다.
	처리: 처리 과정의 한 단계이다.		데이터: 받는 정보나 유포하는 정보와 같은 모든 종류의 입력 또는 출력이다.
	중속 처리: 이미 알려졌거나 이해한 처리 과정으로 순서도에서는 자세히 설명하지 않는다.		문서: 인쇄물처럼 사람들이 읽을 수 있도록 만들어진 것을 의미한다.
	판단: 처리 과정에서 판단을 내려야 하는 지점을 나타낸다.		지연: 처리 과정에서 기다리는 것을 의미한다.
	준비: 준비 단계를 의미한다		

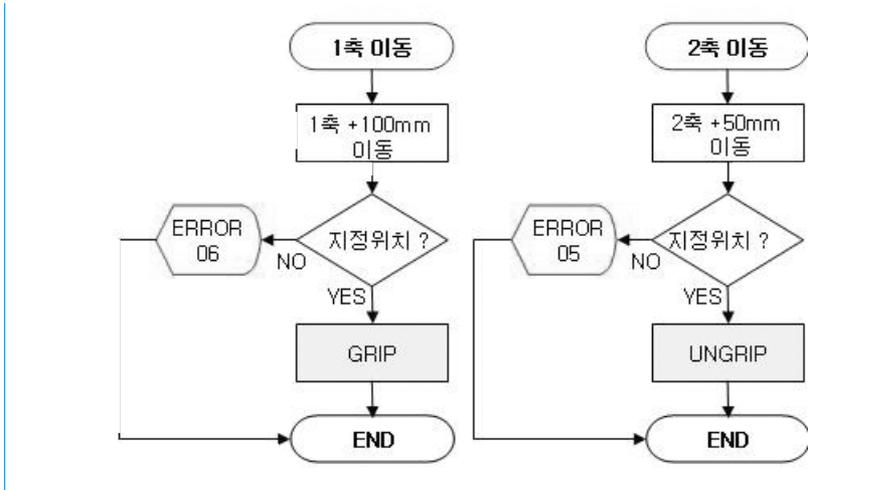
〈표Ⅲ-7〉 흐름도 사용 도형

2) 구동 별 프로그램

구동 별 프로그램은 로봇의 동작을 최소 단위로 구분해서 〈그림Ⅲ-49〉와 같이 GRIP/UNGRIP으로 나누어 하나의 모터가 한 가지 동작을 수행하는 단위를 말한다.

또, 아래와 같이 1축 및 2축 이동 동작 프로그램에 GRIP와 UNGRIP에 대한 동작을 포함할 수 있다.



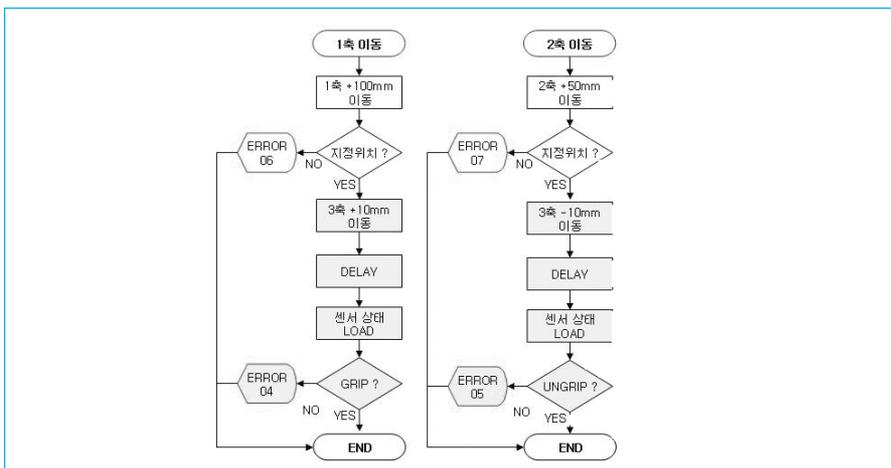


〈그림 III-49〉 모듈별 흐름도

3) 전체 작업 프로그램

전체 작업 프로그램은 구동 별 프로그램을 순차적으로 나열하면 전체 작업 프로그램이 된다.

GRIP와 UNGRIP의 최소 동작은 1축 이동과 2축 이동이라는 상위 단계에 포함되기 때문에 다음과 같은 전체 작업 흐름도를 그릴 수 있다.



〈그림 III-50〉 전체 작업 흐름도

4) 구동 프로그램 작성

구동 프로그램을 작성하기 위해서는 전체 프로그램의 순서도를 보고 작성한다.

순서도를 보고 작성을 하게 되면 전체 내용을 이해할 수 있기 때문에 순서도 없이 작성을 할 때보다 매우 빠른 속도로 프로그램을 작성할 수 있으며, 중간에 엉뚱한 결과가 나왔을 경우 어느 부분에서 잘못되었는지 쉽게 판단할 수 있다.

또한, 내용을 수정할 경우 프로그램 전체에 미치는 영향을 알 수 있기 때문에 프로그램을 수정하지 않고도 결과를 미리 예측할 수 있다.

2 모바일 로봇

이동 로봇의 소프트웨어 기능 구현에 필요한 기본적인 기술에 대해 알아본다.

1) DSP

본 교재에서 사용되는 프로세서는 TI사이, TMS320C6713이라는 DSP (Digital Signal Processor)로 일반적인 프로세서에 비해 I/O 기능은 부족하지만 계산 및 데이터 처리 속도가 빨라서 영상 신호 처리, 음성 신호 처리, 로봇 제어 등에 널리 사용되고 있다.

2) 타이머 인터럽트

프로세서에서 사용하는 타이머는 <그림 III-51>과 같이 프로세서 외부에 연결된 클럭을 이용하여 필요로 하는 특정 시간 펄스를 만들기 위한 기능이다.

예를 들면 25MHz(1펄스의 시간은 0.00000004초)의 클럭을 사용하여 0.02초 단위로 세려면(타이머) 500,000개의 클럭이 필요하게 된다. 이 기능을 프로세서에서 사용하기 위해서는 타이머 설정 레지스터(A)에 500,000을 써 주면

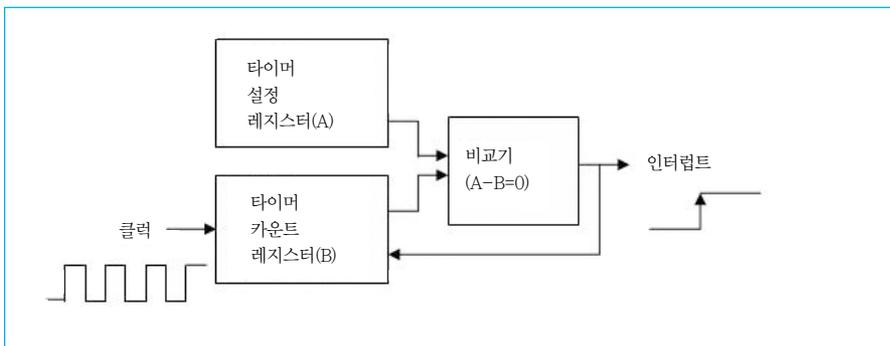


비교기에서 타이머 카운트 레지스터(B) 값이 같아지는 500,000이 될 때 인터럽트 펄스를 발생하게 된다. 또한 이 펄스가 발생할 때 타이머 카운트 레지스터(B)는 0으로 다시 초기화하게 된다. 타이머 카운트 레지스터(B)는 입력 클럭이 들어오면 1씩 증가하는 카운터이다.

이렇게 만들어진 펄스(0.02초)를 DSP의 인터럽트 신호에 연결하여 사용하면 DSP는 0.02초마다 동일한 일을 정확한 시간에 반복적으로 일을 할 수 있다.

3) Watchdog 기능

〈그림 III-51〉에서 만들어진 인터럽트 처리 프로그램의 경우 어떤 일이 있어도 DSP가 항상 동작하는 프로그램 영역이 된다. 만일 DSP가 외부 영향으로 이 프로그램이 정상적으로 실행되지 않는다면 치명적인 오동작이 발생하게 된다. 이러한 상황에서 AGV의 이상 동작을 최소화하기 위해 구동부인 모터의 전원을 차단하게 된다. 이렇게 DSP의 동작 상태를 외부에서 확인하여 처리하는 기능을 Watchdog 기능이라 한다.

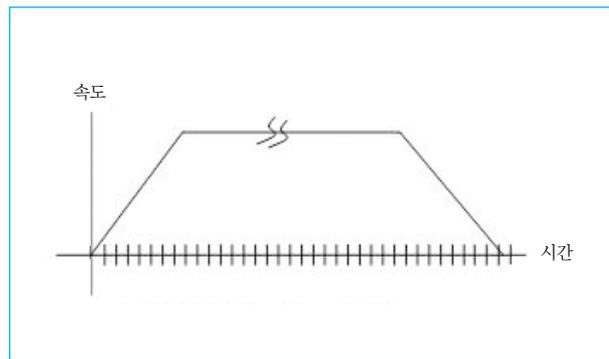


〈그림 III-51〉 타이머 인터럽트

4) 속도 제어

이동 로봇의 주행 속도를 제어하는 방법은 〈그림 III-52〉와 같이 사다리꼴 형

태의 속도 궤적을 만들어서 사용한다. DSP에서 만들어진 0.02초 인터럽트 루틴에서 현재 출력할 속도 명령을 계산하여 모터 드라이브의 속도 명령으로 출력된다.



〈그림Ⅲ-52〉 속도 궤적

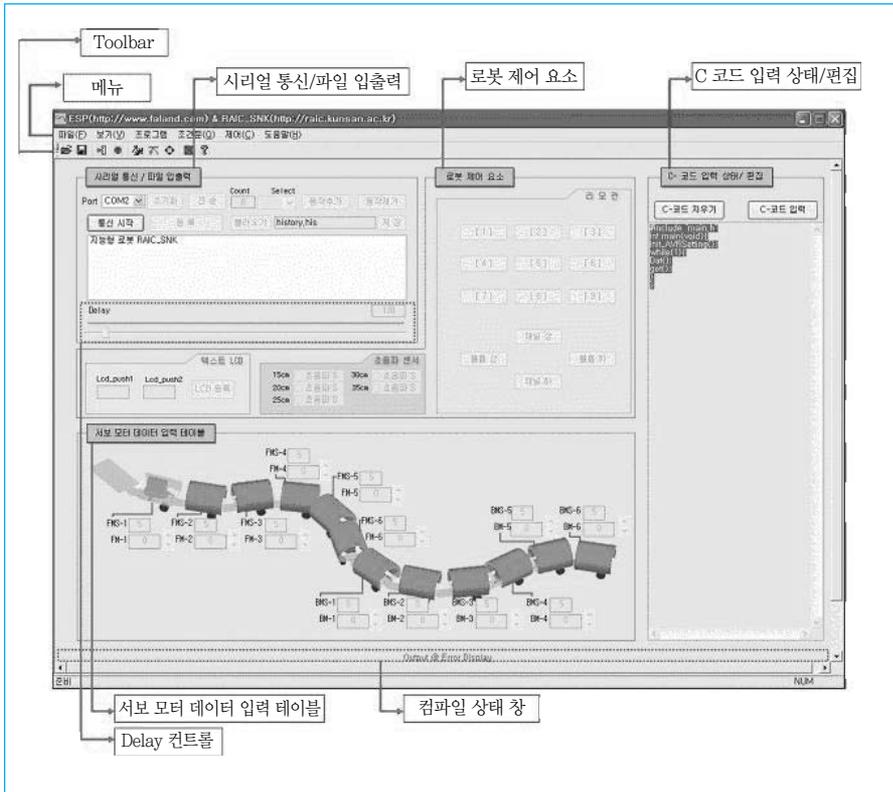
5) 시스템 상태 체크

이동 로봇이 주행할 때 안전에 필요한 센서 및 스위치, 배터리 전압, 온도 등을 주기적으로 확인하여 이상이 있는 경우 로봇을 정지시킨 후 에러 코드 및 에러음을 발생한다.



3 창작 로봇

1) 에뮬레이터



〈그림 III-53〉 에뮬레이터의 통합 환경

특징

- 시리얼 통신을 이용해서 각 모터를 실시간으로 제어할 수 있다.
- 각 센서와 LCD를 버튼 클릭만으로 프로그래밍할 수 있다.
- 고급과정으로 C언어를 함수를 이용해서 프로그래밍할 수 있다.

단원 학습 정리



- ❶ 직교 로봇은 직선 운동을 수행하는 축으로만 구성되어 있는 로봇이다.
- ❷ 직교 로봇의 장점은 기계적으로 튼튼하고 안정적이며 위치 정밀도가 우수하고, 이용자가 쉽게 기술을 습득할 수 있다.
- ❸ 직교 로봇의 작업 영역은 자유도의 수에 따라 직선, 직사각형, 직육면체가 된다.
- ❹ 스카라 로봇은 수평 다관절 로봇이라고도 하며 2개의 회전축과 하나의 직선 축으로 구성된다.
- ❺ 스카라 로봇의 1축과 2축은 수평 회전 운동을 하고 3축은 수직 상하 운동과 회전 운동을 한다.
- ❻ 스카라 로봇은 4개의 자유도를 가지는 작업에 사용된다.
- ❼ 스카라 로봇의 1축과 2축은 회전축으로서 용량이 큰 모터가 장착되어 있으며, 3축은 직선 축 그리고 4축은 핸들 모듈로 구성된다.
- ❽ 직교 로봇은 베이스 프레임, 액츄에이터, 그리고 볼 스크루가 핵심 구성품이다.

9 직교 로봇에서 센서들은 원점 복귀나 하드웨어적 제한을 위해 사용된다.

10 스카라 로봇에 사용되는 모터는 대부분 서보 모터이다.





IV

직교 로봇 제작

직교로봇은 제조업 분야에서 널리 사용되고 있는 로봇 중의 하나이다. 본 단원에서는 직교로봇의 하드웨어와 소프트웨어를 이해하고, 직교 로봇의 조립 방법을 학습한다.



학습목표

1. 직교 로봇의 하드웨어 설계 능력을 배양한다.
2. 직교 로봇의 소프트웨어 설계 능력을 배양한다.
3. 직교 로봇의 조립 능력을 배양한다.

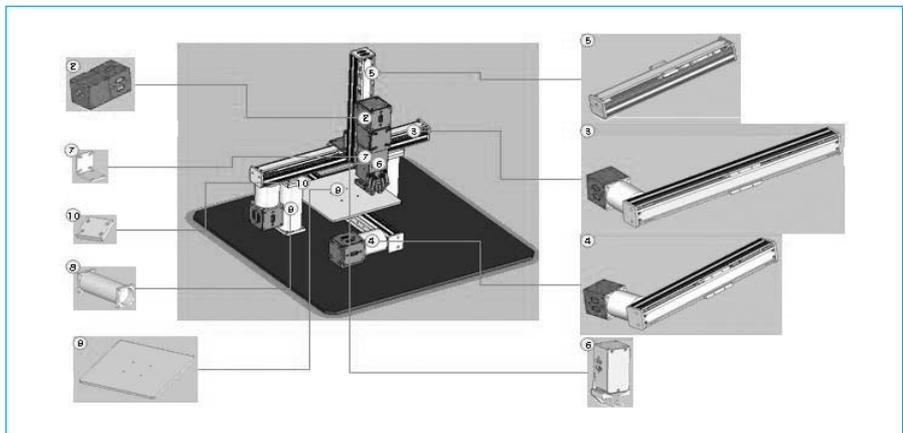


IV

직교 로봇 제작

1 직교 로봇 하드웨어 상세 설계

1 부품 리스트 및 부품 재고 체크



〈그림Ⅳ-1〉 직교 로봇 4축 구성

2 조립 도면 작성 및 주의 사항

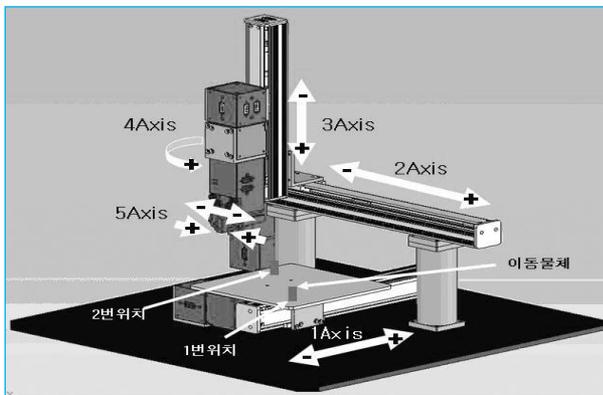
- ③직선 모듈(Module)대(大), ④직선 모듈 중(中), ⑤직선 모듈 소(小)는 1,2,3축에 사용 가능하다.
- ⑤직선 모듈 소(小)는 Z축 방향으로 짧은 스트로크(Stroke) 사용에 적합하다.
- ③④⑤직선 모듈과 ⑧연결 링크 조립 시 ⑦연결 브라켓(Bracket)을 이용하여 조립한다.



- ②회전 모듈 소(小)와 ⑥그리퍼(Gripper) 모듈 조립 시 ⑦연결 브라켓을 이용하여 조립한다.
- ②③④⑤⑥모듈간의 케이블 연결은 In, Out 커넥터(Connector) 구분이 없다.

2 직교 로봇 소프트웨어 상세 설계

1 작업 분석



〈그림Ⅳ-2〉 직교 로봇 5축 구성

〈그림Ⅳ-2〉와 같이 로봇이 물체 1번 위치의 물건을 2번 위치로 이동하는 경우 프로그램을 어떻게 설계하여 작성할 것인가에 대해 생각해 보자.

일단, 1번 위치에서 2번 위치로 블록을 옮길 경우, 사용하는 축이 총 5축이므로 상당히 많은 경우의 수가 발생한다.

어느 축을 먼저 움직일 것인가에 대한 내용은 로봇이 움직일 때 주변의 다른 물체와 충돌을 하지 않는지 또는 이동시킬 물체와 충돌하지 않는지와 같은 경우에 필요하다.

일단 여기서는 어떤 축을 움직일지에 대해서는 프로그램 작성자가 스스로 주



변 환경을 파악하여 결정해야 한다.

어떤 순서로 모터를 구동시킬 것인지 결정되면 순서에 따라 흐름도를 작성한다. 작성된 흐름도를 기본으로 해서 각 구동 별 프로그램을 작성한다.

1. 모든 축의 원점 복귀(Origin)를 수행한다.
2. 1축을 1번 위치로 이동한다.
3. 2축을 1번 물체의 중간 위치로 이동한다.
4. 3번 축을 4축 그리퍼가 물체를 집을 수 있도록 높이를 이동한다.
5. 4축 그리퍼를 이용해서 물체를 집는다.
6. 3번 축을 위로 이동하여 물체를 들어 올린다.
7. 1축을 이동하여 2번 위치로 이동한다.
8. 2축을 2번 위치의 중간 위치로 이동한다.
9. 3축을 4축 그리퍼가 물체를 놓을 수 있도록 최소 높이로 이동한다.
10. 4축 그리퍼를 이용해서 물체를 놓는다.
11. 3축을 이용해서 그리퍼를 위로 올린다.
12. 모든 축을 원위치한다.

보충설명

원점 복귀(Origin)는 로봇을 구동하기 전에 반드시 수행해야 하는 부분으로 증가형 인코더를 사용하는 모든 시스템에 적용된다.

하지만, 원점복귀(Origin)를 수행하지 않고 로봇을 구동할 수 있는 시스템이 있는데, 이런 시스템에는 절대치 인코더를 사용한다.

절대치 인코더는 전원이 차단되어도 자신의 위치를 기억하고 있는 형태의 인코더이다.



2 구동 별 프로그램

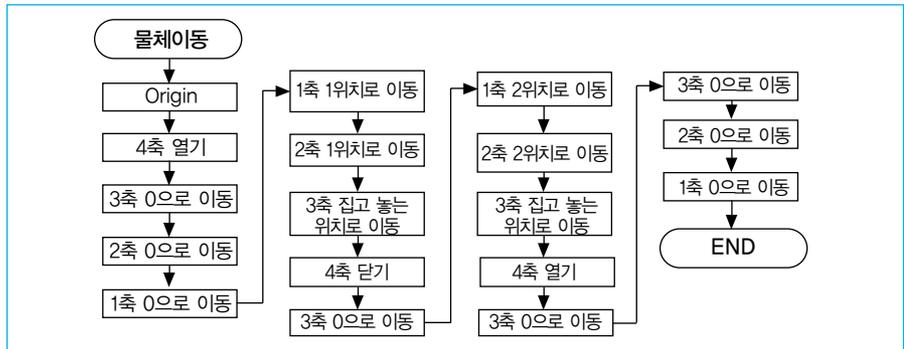
구동 별 프로그램은 위의 작업 순서에 따라 해당 모터가 구동할 수 있는 최소 단위로 분리하는 것이다.

<p>그리퍼를 연다</p> <pre> graph TD A(4축열기) --> B[4축 +30mm 위치로 이동] B --> C(END) </pre>	<p>그리퍼를 닫는다.</p> <pre> graph TD A(4축닫기) --> B[4축 0mm 위치로 이동] B --> C(END) </pre>	<p>1축을 초기 위치로 이동한다.</p> <pre> graph TD A(1축 0으로 이동) --> B[좌표 0mm 위치로 이동] B --> C(END) </pre>
<p>1축을 1번 위치로 이동한다.</p> <pre> graph TD A(1축 1위치로 이동) --> B[1축 +200mm 위치로 이동] B --> C(END) </pre>	<p>1축을 2번 위치로 이동한다.</p> <pre> graph TD A(1축 2위치로 이동) --> B[1축 +10mm 위치로 이동] B --> C(END) </pre>	<p>2축을 초기 위치로 이동한다.</p> <pre> graph TD A(2축 0으로 이동) --> B[좌표 0mm 위치로 이동] B --> C(END) </pre>
<p>2축을 1번 위치로 이동한다.</p> <pre> graph TD A(2축 1위치로 이동) --> B[2축 +100mm 위치로 이동] B --> C(END) </pre>	<p>2축을 2번 위치로 이동한다.</p> <pre> graph TD A(2축 2위치로 이동) --> B[2축 -50mm 위치로 이동] B --> C(END) </pre>	<p>3축을 초기 위치로 이동한다.</p> <pre> graph TD A(3축 0으로 이동) --> B[좌표 0mm 위치로 이동] B --> C(END) </pre>
<p>3축을 집고 놓기 위한 위치로 이동한다.</p> <pre> graph TD A(3축 집고 놓는 위치로 이동) --> B[3축 +80mm 위치로 이동] B --> C(END) </pre>	<p>모든 축의 원점 복귀(Origin)를 수행한다.</p> <pre> graph TD A(Origin) --> B[4축 Origin] B --> C[3축 Origin] C --> D[2축 Origin] D --> E[1축 Origin] E --> F(END) </pre>	

〈표IV-1〉 직교로봇 구동 별 흐름도

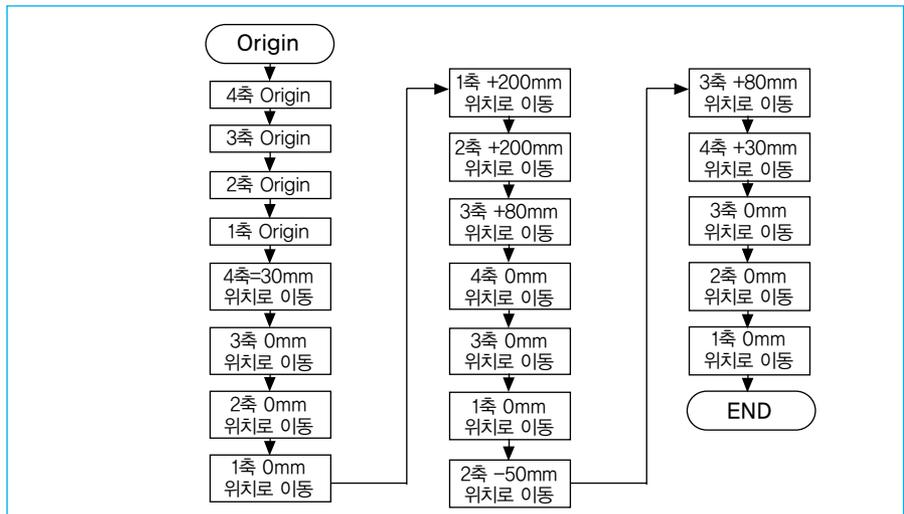
3 전체 작업 프로그램

전체 작업 프로그램을 각 구동 별 프로그램으로 구성을 하면 다음과 같이 흐름도를 구성할 수 있다. 구성된 프로그램은 구체적이지 않고 약간은 추상적인 동작 상태를 나타내고 있다.



〈그림Ⅳ-3〉 직교 로봇 전체 흐름도

〈그림Ⅳ-4〉와 같은 흐름도는 위에서 나타난 흐름도보다 동작에 대한 상태를 자세하게 나타내고 있다. 흐름도만으로 전체 프로그램을 쉽게 작성할 수 있다.



〈그림Ⅳ-4〉 직교 로봇 상세 작업 흐름도



4 구동 프로그램 작성

구동 프로그램은 전체 프로그램을 바탕으로 순차적으로 구성하면 된다.

```
ORG

MOVEJ 4 30 MOVE_JOINT
MOVEJ 3 0 MOVE_JOINT
MOVEJ 2 0 MOVE_JOINT
MOVEJ 1 0 MOVE_JOINT

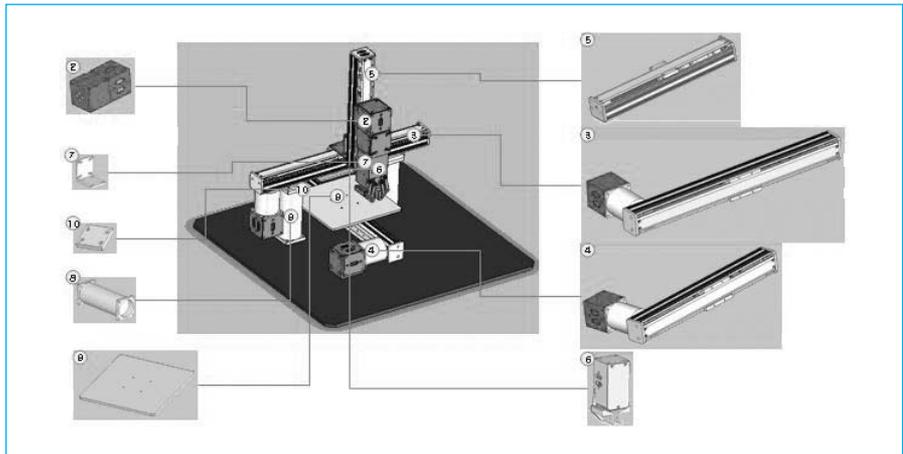
MOVEJ 1 200 MOVE_JOINT
MOVEJ 2 100 MOVE_JOINT
MOVEJ 3 80 MOVE_JOINT
MOVEJ 4 0 MOVE_JOINT
MOVEJ 3 0 MOVE_JOINT

MOVEJ 1 10 MOVE_JOINT
MOVEJ 2 50 MOVE_JOINT
MOVEJ 3 80 MOVE_JOINT
MOVEJ 4 30 MOVE_JOINT
MOVEJ 3 0 MOVE_JOINT

MOVEJ 2 0 MOVE_JOINT
MOVEJ 1 0 MOVE_JOINT
```

〈표IV-2〉 직교 로봇 전체 프로그램

3 직교 로봇 조립



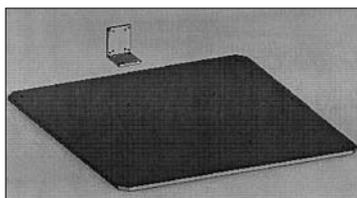
〈그림Ⅳ-5〉 직교 로봇 구성품 명칭

부 품	수 량	부 품	수 량
① 회전 모듈 대	2	⑦ 연결 브라켓	2
② 회전 모듈 소	1	⑧ 연결 링크	3
③ 직선 모듈 대	1	⑨ Moving Plate	1
④ 직선 모듈 중	1	⑩ 브라켓	3
⑤ 직선 모듈 소	1		
⑥ 그리퍼 모듈	1		

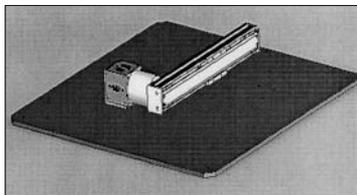
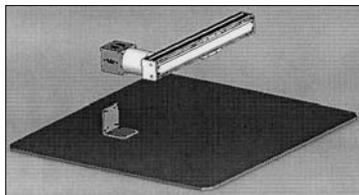
〈표Ⅳ-3〉 구성품 목록

〈조립 순서〉

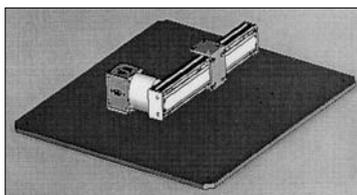
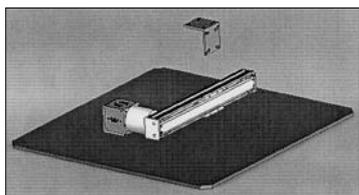
Step 1 • Base 판에 ⑧ 연결 링크 조립



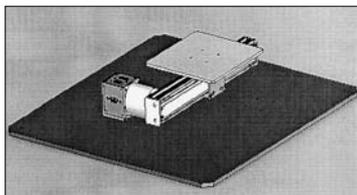
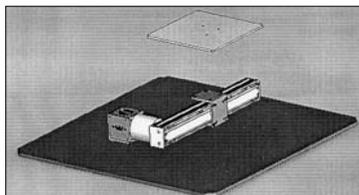
Step 2 • ④ 직선 모듈 중(中) 연결



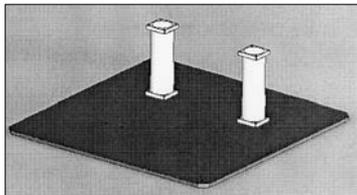
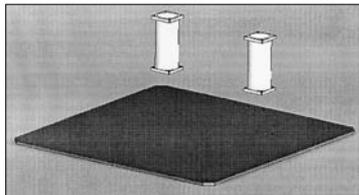
Step 3 • ④ 직선 모듈 중(中)의 엔드이펙터에 ⑦ 연결 브라켓 조립



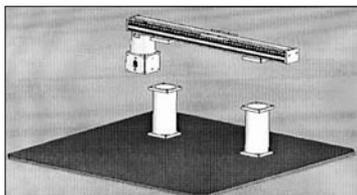
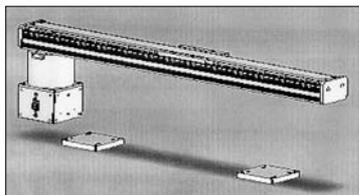
Step 4 • ⑦ 연결 브라켓에 ⑨ 무빙 플레이트(Moving Plate) 조립



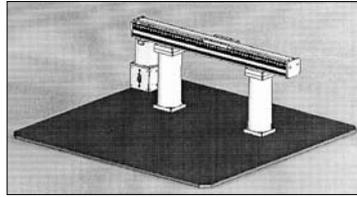
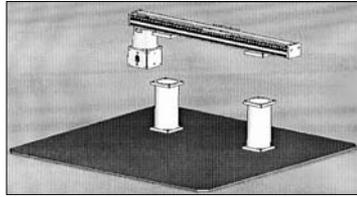
Step 5 • Base에 ⑧ 연결 링크 조립



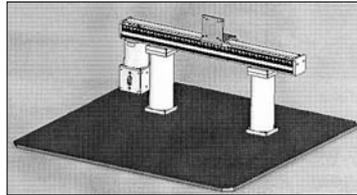
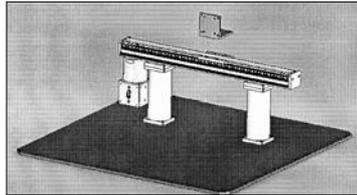
Step 6 • ③ 직선 모듈 대(大)에 ⑩ 브라켓 조립



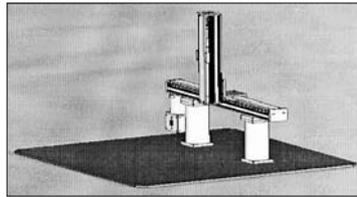
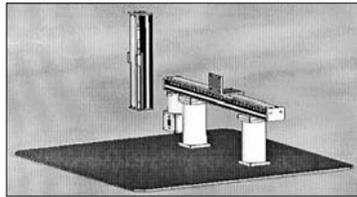
Step 7 ● ⑧ 연결 링크에 ③ 직선 모듈 대 조립



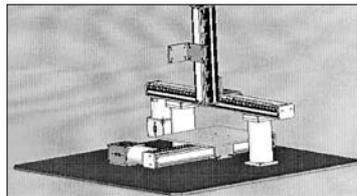
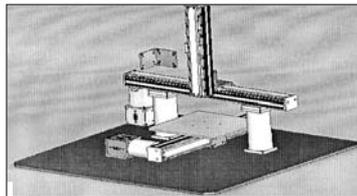
Step 8 ● ③ 직선 모듈 대(大) 에 ⑦ 연결 브라켓 조립



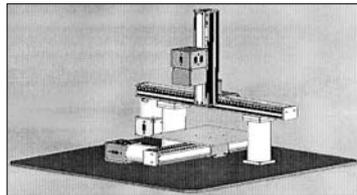
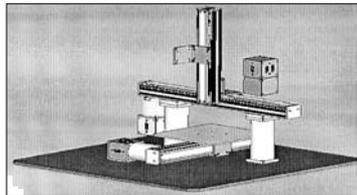
Step 9 ● ⑦ 연결 브라켓에 ⑤ 직선 모듈 소 조립



Step 10 ● ⑤ 직선 모듈 소(小) 의 엔드이펙터 부에 ⑦ 연결 브라켓 조립

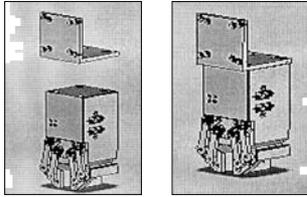


Step 11 ● ⑦ 연결 브라켓에 ② 회전 모듈 소 조립

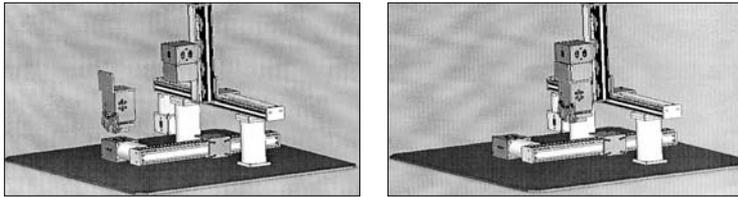




Step 12 ● ⑥ 그리퍼 모듈에 ⑦ 연결 브라켓 조립



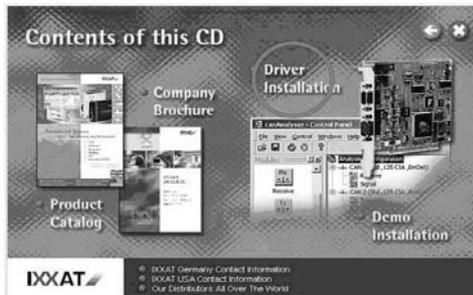
Step 13 ● ② 회전 모듈 소(小) 에 ⑥ 그리퍼 모듈 조립



4 직교 로봇 프로그래밍

⇒ S/W 설치





데모(DEMO) 프로그램

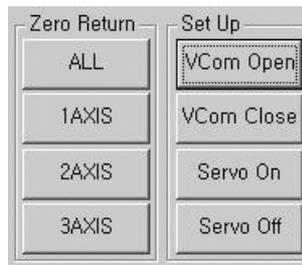
CD안의 “파워 모듈 데모 프로그램” 디렉토리에 있는 CartesianDemo.exe 파일을 실행한다.





데모 프로그램을 실행시키면 다음과 같은 화면을 볼 수 있다.

여기서 로봇을 구동하기 위해서는 “VCom Open”을 클릭하여 USB to CAN과 연결한다.



연결이 완료된 후 로봇을 구동하기 전 반드시 Zero Return의 “ALL” 버튼을 클릭하여 원점 복귀(Zero Return)를 수행한 후 로봇을 구동하여야 한다.

보충설명

한 번 Zero Return(원점 복귀) 되면 전원을 OFF하기 전까지는 다시 원점 복귀 하지 않아도 된다.

로봇의 동작 형태는 Joint Move에서 절대 모드로 선택하면 된다.

◆ 보충설명

절대 모드는 원점 복귀한 후 절대값으로 바로 이동하는 모드이다.

만일 10mm이동하라는 명령을 내렸을 경우 로봇은 10mm만 이동하게 된다. 그 후 또 10mm 이동하라는 명령을 로봇에게 보내면 현재 10mm이동한 상태 이므로 움직이지 않고 정지하게 된다.

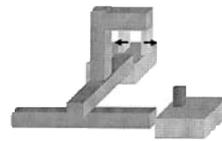
상대 모드는 현재 위치에서 입력한 값만큼 이동하는 모드이다.

현재 위치가 5mm이고 10mm 이동하라는 명령을 보내면 10mm 이동하여 최초 위치에서 15mm를 이동하게 된다.

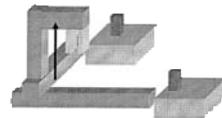
앞서 작성한 흐름도를 기반으로 해서 프로그램을 작성해 보도록 하자.

일단 작업에 의해 필요한 위치 포인트(Point)의 값은 다음과 같이 총 11개의 포인트가 필요하다.

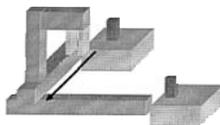
1. 5축의 +30mm 위치 포인트 값을 저장한다. (Point 1)



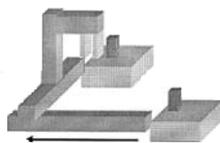
2. 3축의 0mm 위치 포인트 값을 저장한다.



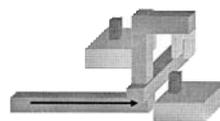
3. 2축의 0mm 위치 포인트 값을 저장한다.



4. 1축의 0mm 위치 포인트 값을 저장한다.



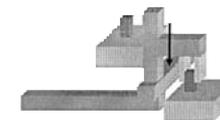
5. 1축의 +200mm 위치 포인트 값을 저장한다.



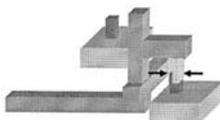
6. 2축의 +100mm 위치 포인트 값을 저장한다.



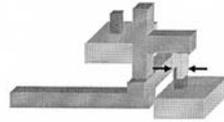
7. 3축의 +80mm 위치 포인트 값을 저장한다.



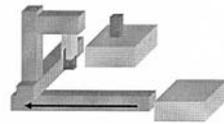
8. 5축의 +0mm 위치 포인트 값을 저장한다.



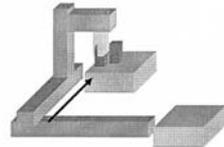
9. 1축의 +10mm 위치 포인트 값을 저장한다.



10. 2축의 +50mm 위치 포인트 값을 저장한다.

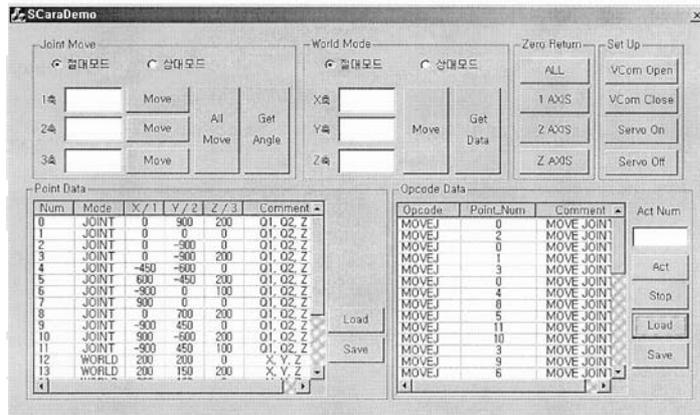


11. 3축의 +80mm 위치 포인트 값을 저장한다.



이렇게 11개의 위치 포인트를 저장하고 난 후 저장된 포인트를 이용하여 실제 로봇 프로그램을 작성한다.

- 필요한 위치의 포인트가 전부 저장되었나 다시 한번 흐름도를 확인해 본다.
- 저장하지 않은 부분은 로봇을 이동시켜 특정 위치의 포인트를 저장한다.
- 저장된 포인트를 이용해 다음과 같이 프로그램을 작성한다.



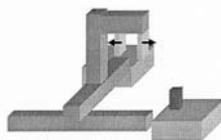
구동 프로그램은 전체 프로그램을 바탕으로 순차적으로 구성하면 된다.

1. 전체 축이 원점복귀를 수행한다.



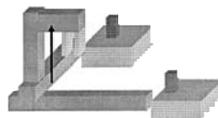
[ORIGIN]

2. 3축을 30mm 위치로 이동한다.



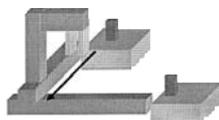
[MOVEJ 4 30 MOVE_JOINT]

3. 2축의 0mm 위치로 이동한다.



[MOVEJ 3 0 MOVE_JOINT]

4. 1축의 0mm 위치로 이동한다.



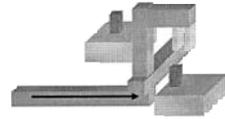
[MOVEJ 2 0 MOVE_JOINT]

5. 5축의 0mm 위치로 이동한다.



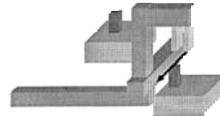
[MOVEJ 1 0 MOVE_JOINT]

6. 1축의 +200mm 위치로 이동한다.



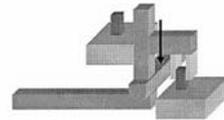
[MOVEJ 1 200 MOVE_JOINT]

7. 2축의 +250mm 위치로 이동한다.



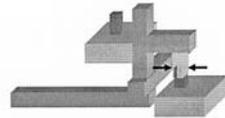
[MOVEJ 2 100 MOVE_JOINT]

8. 3축의 +50mm 위치로 이동한다.



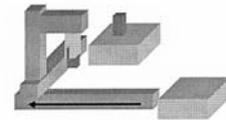
[MOVEJ 3 80 MOVE_JOINT]

9. 4축의 +30mm 위치로 이동한다.



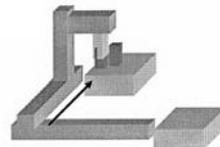
[MOVEJ 4 0 MOVE_JOINT]

10. 3축의 0mm 위치로 이동한다.



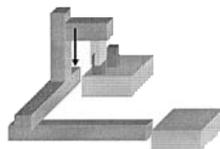
[MOVEJ 1 10 MOVE_JOINT]

11. 2축의 +50mm 위치로 이동한다.



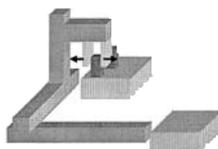
[MOVEJ 2 50 MOVE_JOINT]

12. 1축의 +30mm 위치로 이동한다.



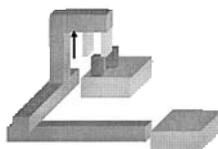
[MOVEJ 3 80 MOVE_JOINT]

13. 3축의 +80mm 위치로 이동한다.



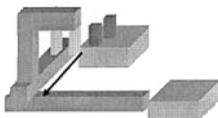
[MOVEJ 4 0 MOVE_JOINT]

14. 3축의 0mm 위치로 이동한다.



[MOVEJ 4 0 MOVE_JOINT]

15. 3축의 0mm 위치로 이동한다.



[MOVEJ 4 0 MOVE_JOINT]

이렇게 전체 로봇 동작에 대한 프로그램을 작성하면 실제 로봇의 동작이 흐름도와 같다는 것을 확인 할수 있다. 흐름도를 잘 작성하면 프로그램을 쉽게 작성할 수 있다. <표IV-4>는 전체 프로그램을 작성한 것이다.

직교 로봇 흐름도에 따른 전체 프로그램

```
MOVEJ 4 30 MOVE_JOINT
MOVEJ 3 0 MOVE_JOINT
MOVEJ 2 0 MOVE_JOINT
MOVEJ 1 0 MOVE_JOINT
```

```

MOVEJ 1 200 MOVE_JOINT
MOVEJ 2 100 MOVE_JOINT
MOVEJ 3 80 MOVE_JOINT
MOVEJ 4 0 MOVE_JOINT
MOVEJ 3 0 MOVE_JOINT

MOVEJ 1 10 MOVE_JOINT
MOVEJ 2 50 MOVE_JOINT
MOVEJ 3 80 MOVE_JOINT
MOVEJ 4 30 MOVE_JOINT
MOVEJ 3 0 MOVE_JOINT

MOVEJ 2 0 MOVE_JOINT
MOVEJ 1 0 MOVE_JOINT

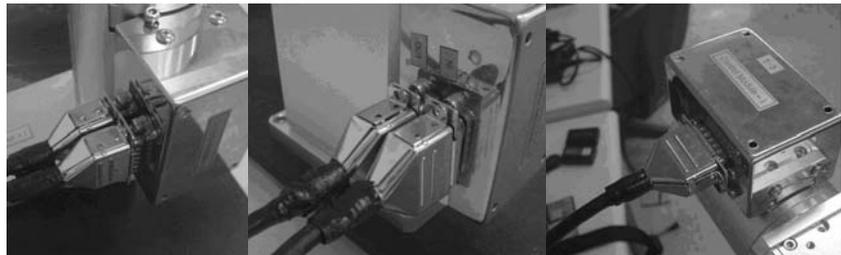
```

5 직교 로봇 구동시험 및 예제 실습

케이블 연결

케이블은 정확히 연결되어야 한다. 만약 케이블 연결이 잘못되면 로봇의 운동을 예측하지 못하는 상황이 발생할 수도 있고, 경우에 따라서는 심각한 부상과 손상을 초래할 수도 있다.

케이블이 연결되는 로봇 몸체 측과 제어기 측의 커넥터 판넬에 각각 나타나 있다. 케이블의 라벨을 확인하고 커넥터에 무리한 힘이 가해지지 않도록 주의하며 연결해야 한다.

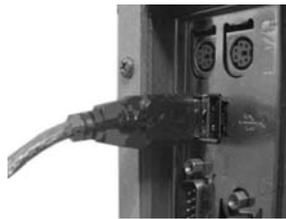




통신선과 전원선을 연결한다.



USB to CAN 을 PC의 USB 단자에 연결한다.



동작 방법

로봇과 제어기의 연결 상태를 확인한다.

로봇 제어기의 전원을 켜기 전에 다음 주의 사항을 확인한다.

* 주의!

제어기 전원 케이블의 연결 상태 및 전원의 정상 여부

로봇과 제어기 사이의 케이블 연결 상태

로봇이 프로그램 오류나 조작상의 실수로 인해서 의도한 동작과 다르게 움직일 경우, 또는 동작 범위 내의 사람이나 장애물과 충돌이 예상될 경우에는 로봇을 신속히 정지시켜야 한다. 이와 같이 로봇을 비상 정지시키기 위해서는 빨간색 버튼의 비상 정지 스위치를 눌러야 한다. 비상정지 스위치는 두 군데에 위치하고 있다.

CAN

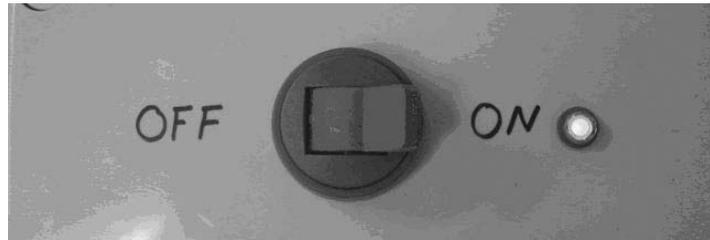
CAN (Controller Area Network)는 초기에 자동차 산업분야에 적용하기 위해 고안된 시리얼 네트워크 통신방식이다.

CAN은 여러가지 장치들을 2개의 선으로 제어할 수 있고 외부 노이즈에 강하다는 장점때문에 로봇 등의 산업에 점차 확대 적용되고 있다.

*** 주의!**

비상정지 스위치를 누르게 되면 로봇은 정지하고, 서보 전원도 꺼지게 된다. 비상정지가 작동하면 모든 모터의 전원 공급이 중단되므로 팔이 자유낙하하여 기구물에 부딪히지 않도록 주의해야 한다.

비상 정지 상태를 해제하려면 아래의 사진에 나타난 것처럼 비상정지 스위치를 시계 방향으로 돌려서 원 상태로 풀어주면 된다. 로봇의 주 전원을 끄지 않았다면 원점 복귀를 다시 수행할 필요는 없으며, 서보 전원만을 다시 투입하면 구동 가능 상태로 된다. 그러나 주 전원 스위치를 껐다면 로봇의 원점 복귀를 다시 실행해야 한다. 또한 모터 전원을 인가하면 로봇을 바르게 동작시킬 수 있다.

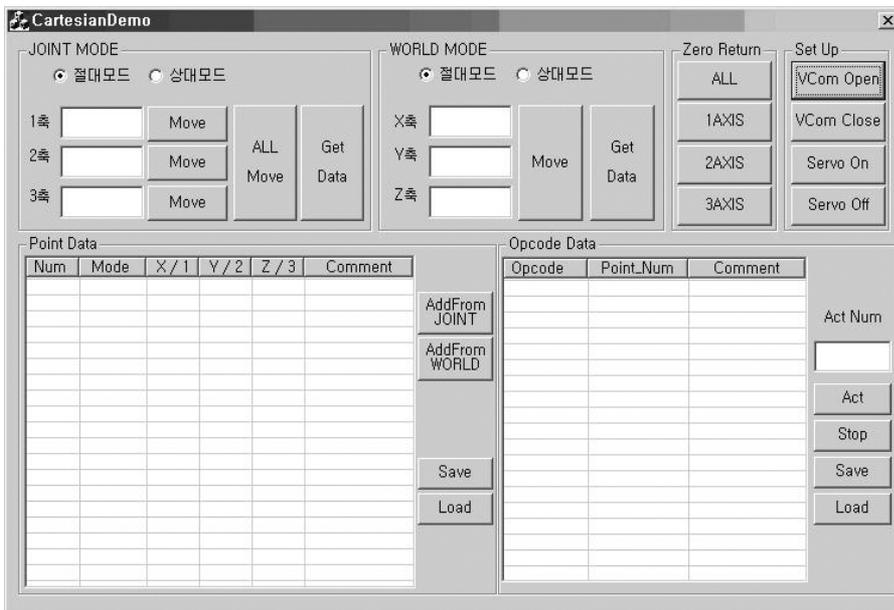


스위치를 ON 하게 되면 불이 들어오면서 24V 전원공급기가 동작하는 소리가 들린다.

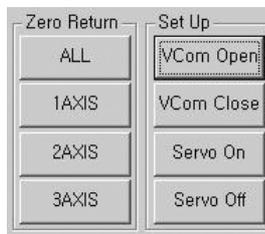
현재 E-STOP 버튼이 눌러져 있는 상태에서 E-STOP 버튼을 풀면 “딸깍” 하는 소리와 함께 팔이 약간 움직이는 것을 볼 수 있다.

데모(DEMO) 프로그램

CD안의 “파워모듈 데모 프로그램” 디렉토리에 있는 CartesianDemo.exe 파일을 실행한다.



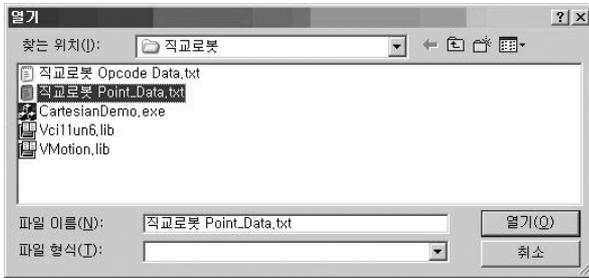
데모 프로그램을 실행 시키면 다음과 같은 화면을 볼 수 있다. 여기서 로봇을 구동하기 위해서는 “VCom Open”을 클릭하여 USB to CAN과 연결한다.



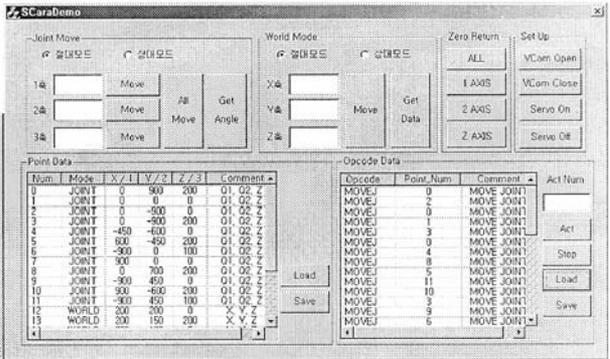
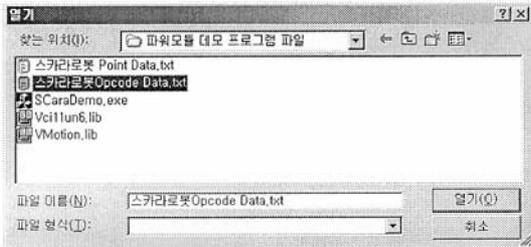
연결이 완료된후 로봇을 구동하기 전 반드시 Zero Return의 “ALL” 버튼을 클릭하여 원점복귀(Zero Return)를 수행한 후 로봇을 구동해야 한다.



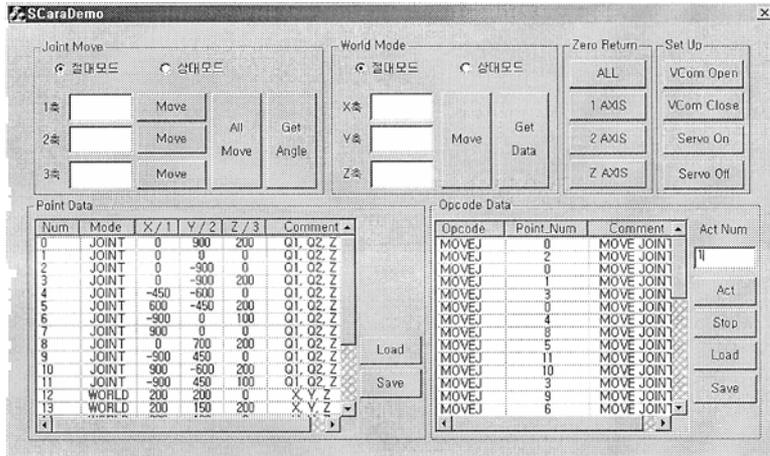
데모 프로그램을 실행하기 위해서는 다음과 같이 수행해야 한다.
Point Data 부분에서 Load를 눌러 해당 Point Data를 불러들인다.



Opocode 부분에서 LOAD를 눌러 해당 Opocode Data를 불러들인다.



실행하기 위한 모든 설정이 완료된다.



마지막으로 화면 오른쪽에 보이는 Act Num칸에 반복 수행 횟수를 입력한다. (1~999)

실행이 모두 끝나야 정지하므로 너무 많은 숫자를 입력하지 않는다. 설정된 프로그램을 실행하기 위해서는 “Act” 버튼을 클릭하면 설정된 순서로 동작하게 된다.

단원 학습 정리

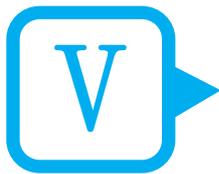


- 1** 전체 작업 프로그램은 각 부분별 프로그램으로 흐름도를 구성한다. 흐름도로 작성하면 프로그램을 쉽게 작성할 수 있다.
- 2** 구동 프로그램은 전체 프로그램을 바탕으로 순차적으로 구성한다.
- 3** 케이블 연결은 매우 중요한 부분이다. 잘못 연결된 케이블은 로봇의 운동을 예측할 수 없게 하며 오동작으로 인체에 치명적인 손상을 가져올 수도 있다.
- 4** 케이블은 라벨을 확인하고 커넥터에 무리한 힘이 가해지지 않도록 주의한다.
- 5** 프로그램의 오류나 조작상의 실수로 인해서 의도한 동작과 다르게 동작할 경우 로봇을 신속히 정지시켜야 한다.
- 6** 비상 정지 상태를 해제하려면 주위의 모든 상황을 점검하고 이상이 없을 때 비상정지 스위치를 시계방향으로 돌려서 원 상태로 복귀시킨다.

7 직교 좌표 로봇을 구동시키기 위해서 스위치를 동작시키면 전원이 동작하는 소리가 들린다.

8 E-STOP 버튼이 눌러져 있는 상태에서 E-STOP 버튼을 풀면 “딸깍” 하는 소리와 함께 팔이 약간 움직이며 동작한다.





SCARA 로봇 제작

스카라 로봇은 제조업 분야에서 직교로봇과 함께 널리 사용되고 있는 로봇 중의 하나이다.

본 단원에서는 스카라 로봇의 하드웨어와 소프트웨어를 이해하고, 스카라 로봇이 조립 방법을 학습한다.



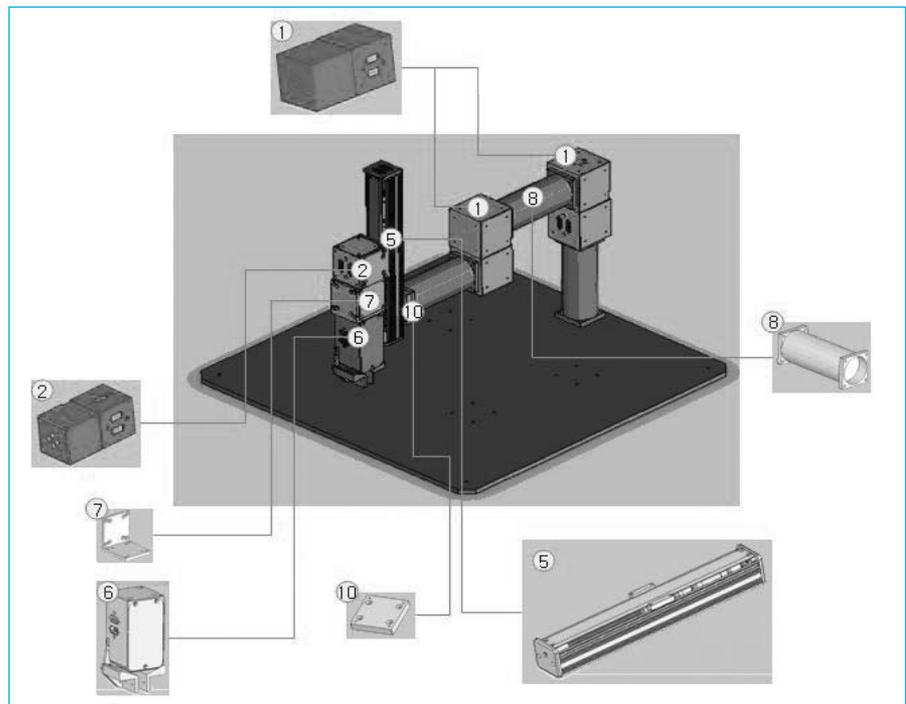
학습목표

1. 스카라 로봇의 하드웨어 설계 능력을 배양한다.
2. 스카라 로봇의 소프트웨어 설계 능력을 배양한다.
3. 스카라 로봇의 조립능력을 배양한다.



1 SCARA 로봇 하드웨어 설계

1 부품 리스트 및 부품 재고 체크



<그림V-1> 스카라 로봇의 하드웨어 구성

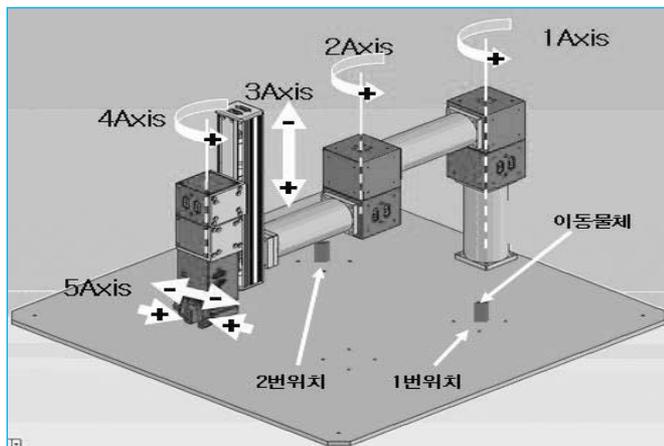


2 조립 도면 작성 및 주의 사항

- ①회전 모듈 대는 1축과 2축 용으로 사용 가능하다.
- ②회전 모듈 소는 3축 이상에서 사용 가능하다.
- ③④⑤직선 모듈과 ⑧연결링크 조립시 ⑦연결브라켓을 이용하여 조립한다.
- ②회전 모듈소와 ⑥그리퍼 모듈 조립시 ⑦연결 브라켓을 이용하여 조립한다.
- ①②③④⑤모듈 간의 케이블 연결은 In, Out 커넥터 구분이 없다.

2 SCARA 로봇 소프트웨어 상세 설계

1 작업 분석



〈그림V-2〉 스카라 로봇 5축 구성 시

〈그림 V-2〉와 같이 로봇이 물체 1번 위치의 물건을 2번 위치로 이동하는 경우 프로그램을 어떻게 설계하여 작성할 것인가에 대해 생각해 보도록 한다.

일단, 1번 위치에서 2번 위치로 블록을 옮길 경우, 사용하는 축이 총 4축이므



로 상당히 많은 경우의 수가 발생하게 된다.

어느 축을 먼저 움직일 것인가에 대한 내용은 로봇이 움직일 때 주변의 다른 물체와 충돌을 하지 않는지 또는 이동시킬 물체와 충돌하지 않는지와 같은 경우에 필요하다.

일단 여기서는 어떤 축을 움직일지에 대해서는 프로그램 작성자가 스스로 주변 환경을 파악하여 결정해야 한다.

어떤 순서로 모터를 구동시킬 것인지 결정되면 순서에 따라 흐름도로 작성한다. 작성된 흐름도를 기본으로 해서 각 구동 별 프로그램을 작성한다.

1. 모든 축의 원점 복귀(Origin)를 수행한다.
2. 1축을 1번 위치로 이동한다.
3. 2축을 1번 물체의 중간 위치로 이동한다.
4. 3축을 5축 그리퍼(Gripper)가 물체를 집을 수 있도록 높이를 이동한다.
5. 5축을 그리퍼를 이용해서 물체를 집는다.
6. 3축을 위로 이동하여 물체를 들어 올린다.
7. 2축을 초기 위치로 이동한다.
8. 1축을 이동하여 2번 위치로 이동한다.
9. 2축을 2번 위치의 중간 위치로 이동한다.
10. 3축을 5축 그리퍼가 물체를 놓을 수 있도록 최소 높이로 이동한다..
11. 5축을 그리퍼를 이용해서 물체를 놓는다.
12. 3축을 초기 위치로 이동시킨다.
13. 2축을 초기 위치로 이동시킨다.
14. 1축을 초기 위치로 이동시킨다.



원점복귀는 로봇을 구동하기 전에 반드시 수행해야 하는 부분으로 증가형 인코더를 사용하는 모든 시스템에 적용된다.

하지만, 원점복귀를 수행하지 않고 로봇을 구동할 수 있는 시스템이 있는데, 이런 시스템에는 절대형 인코더를 사용하고 있다.

절대형 인코더는 전원이 차단되어도 자신의 위치를 기억하고 있는 형태의 인코더이다.

구동별 프로그램은 위의 작업 순서에 따라 해당 모터가 구동할 수 있는 최소 단위로 분리하는 것이다.

2 구동 별 프로그램

구동 별 프로그램은 위의 작업 순서에 따라 해당 모터가 구동할 수 있는 최소 단위로 분리하는 것이다.

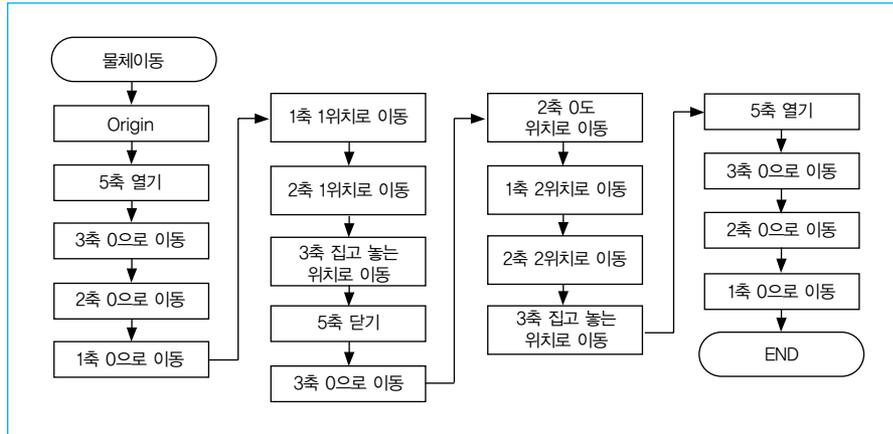
그리퍼를 연다	그리퍼를 닫는다.	1축을 초기 위치로 이동한다.
<pre> graph TD A(4축열기) --> B[4축 +30mm 위치로 이동] B --> C(END) </pre>	<pre> graph TD A(4축닫기) --> B[4축 0mm 위치로 이동] B --> C(END) </pre>	<pre> graph TD A(1축 0으로 이동) --> B[좌표 0mm 위치로 이동] B --> C(END) </pre>
1축을 1번 위치로 이동한다.	1축을 2번 위치로 이동한다.	2축을 초기 위치로 이동한다.
	<pre> graph TD A(1축 2위치로 이동) --> B[1축 +10mm 위치로 이동] B --> C(END) </pre>	<pre> graph TD A(2축 0으로 이동) --> B[좌표 0mm 위치로 이동] B --> C(END) </pre>

2축을 1번 위치로 이동한다.	2축을 2번 위치로 이동한다.	3축을 초기 위치로 이동한다.
<pre> graph TD A([2축 1위치로 이동]) --> B[2축 +100mm 위치로 이동] B --> C([END]) </pre>	<pre> graph TD A([2축 2위치로 이동]) --> B[2축 -50mm 위치로 이동] B --> C([END]) </pre>	<pre> graph TD A([3축 0으로 이동]) --> B[좌표 0mm 위치로 이동] B --> C([END]) </pre>
3축을 집고놓기 위한 위치로 이동한다.	모든 축을 Origin을 수행한다.	
<pre> graph TD A([3축 집고 놓는 위치로 이동]) --> B[3축 +80mm 위치로 이동] B --> C([END]) </pre>	<pre> graph TD A([Origin]) --> B[4축 Origin] B --> C[3축 Origin] C --> D[2축 Origin] D --> E[1축 Origin] E --> F([END]) </pre>	

〈표V-1〉 스카라 로봇 구동 별 흐름도

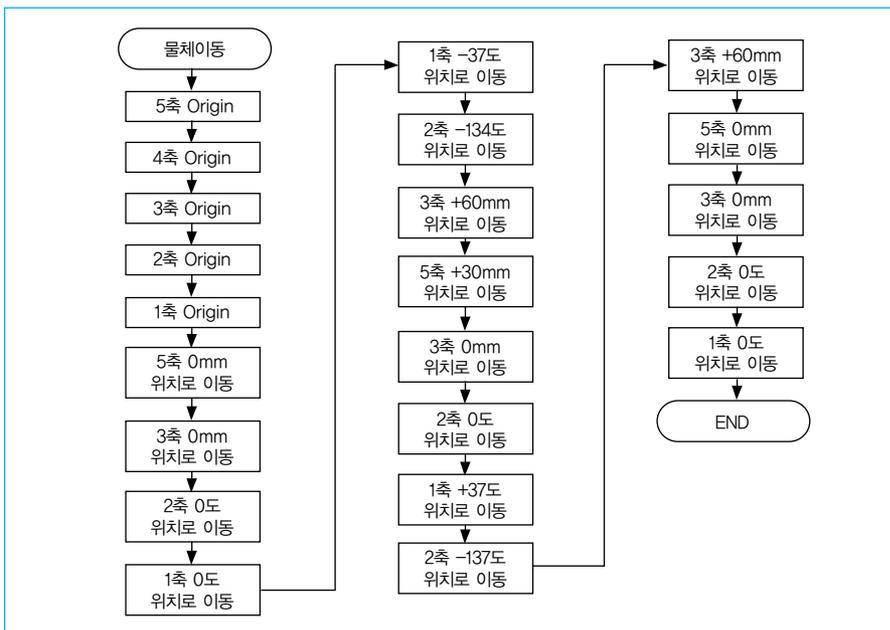
③ 전체 작업 프로그램

전체 작업 프로그램을 각 구동 별 프로그램으로 구성을 하면 다음과 같이 흐름도를 구성할 수 있다. 구성된 프로그램은 구체적이지 않고 약간은 추상적인 동작 상태를 나타내고 있다.



〈그림 V-3〉 스카라 로봇 전체 흐름도

〈그림 V-4〉와 같은 흐름도는 위에서 나타낸 흐름도보다 동작에 대한 상태를 자세하게 나타내고 있다. 흐름도만으로 전체 프로그램을 쉽게 작성할 수 있다.



〈그림 V-4〉 스카라 로봇 상세 작업 흐름도

4 구동 프로그램 작성

구동 프로그램은 전체 프로그램을 바탕으로 순차적으로 구성하면 된다.

```
ORG

MOVEJ 4 30 MOVE_JOINT
MOVEJ 3 0 MOVE_JOINT
MOVEJ 2 0 MOVE_JOINT
MOVEJ 1 0 MOVE_JOINT

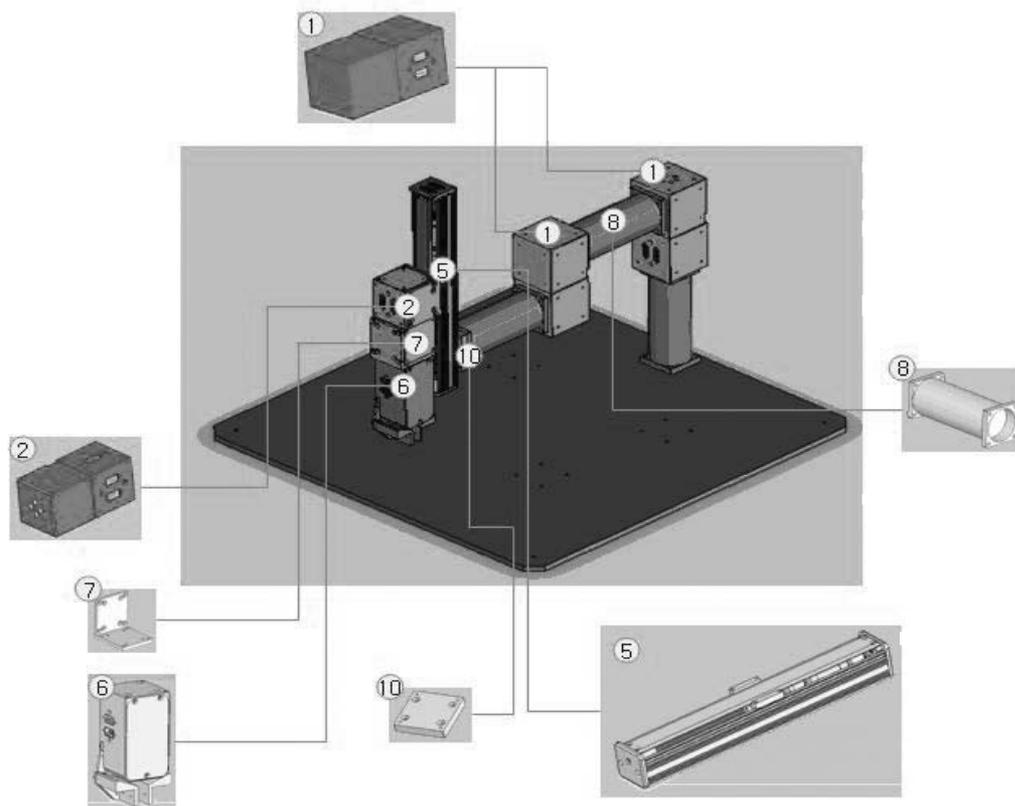
MOVEJ 1 200 MOVE_JOINT
MOVEJ 2 100 MOVE_JOINT
MOVEJ 3 80 MOVE_JOINT
MOVEJ 4 0 MOVE_JOINT
MOVEJ 3 0 MOVE_JOINT

MOVEJ 1 10 MOVE_JOINT
MOVEJ 2 50 MOVE_JOINT
MOVEJ 3 80 MOVE_JOINT
MOVEJ 4 30 MOVE_JOINT
MOVEJ 3 0 MOVE_JOINT

MOVEJ 2 0 MOVE_JOINT
MOVEJ 1 0 MOVE_JOINT
```

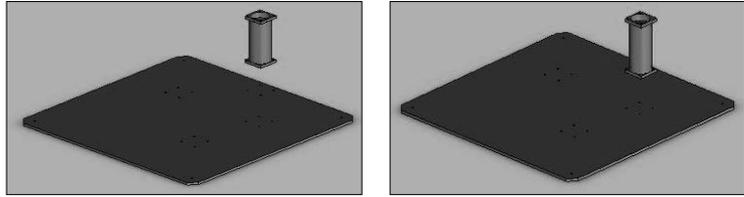
〈표V-2〉 스카라 로봇 전체 프로그램

3 SCARA 로봇 조립

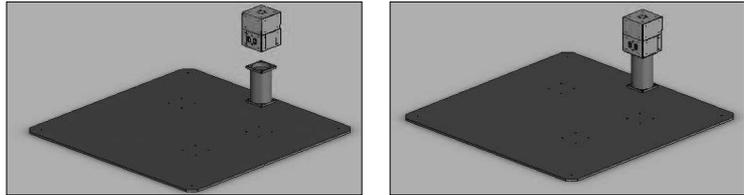


부 품	수량	부 품	수량
① 회전 모듈 대(大)	2	⑦ 연결 브라켓	2
② 회전 모듈 소(小)	1	⑧ 연결 링크	3
⑤ 직선 모듈 소(小)	1	⑩ 브라켓	3
⑥ 그리퍼 모듈	1		

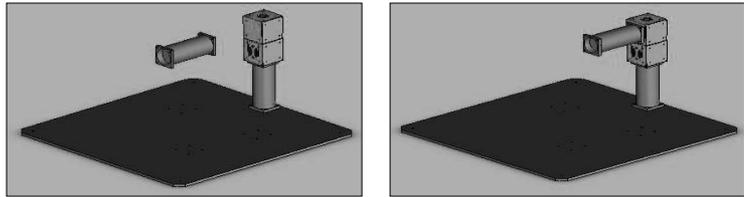
Step 1 • Base 에 ⑧ 연결 링크 조립



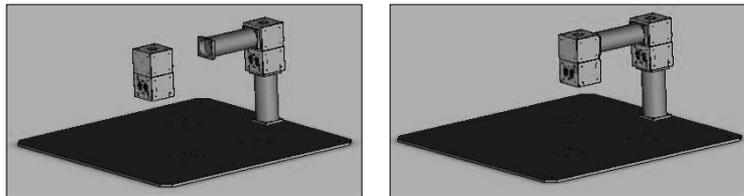
Step 2 • ⑧ 연결 링크에 ① 회전 모듈 대(大) 조립



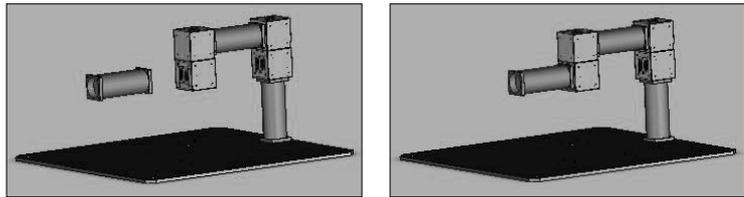
Step 3 • ① 회전 모듈 대(大)에 ⑧ 연결 링크 조립



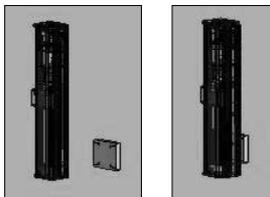
Step 4 • ⑧ 연결 링크에 ① 회전 모듈 중(中) 조립



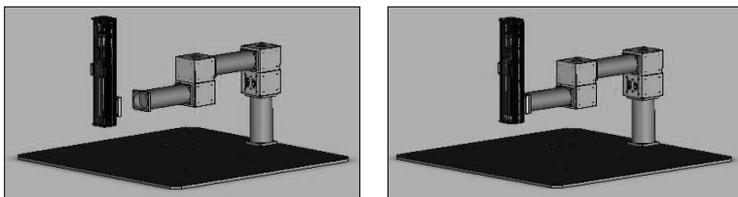
Step 5 • ① 회전 모듈 중(中)에 ⑧ 연결 링크 조립



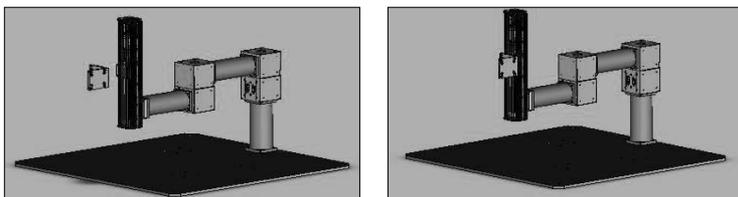
Step 6 ● ⑤ 직선 모듈 소(小)에 ⑩ 브라켓 조립



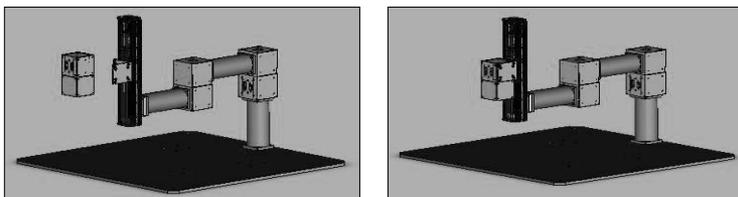
Step 7 ● ⑧ 연결 링크에 ⑤ 직선 모듈 소(小) 조립



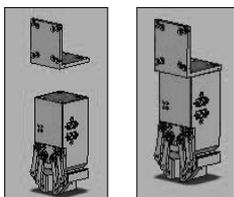
Step 8 ● ⑤ 직선 모듈 소(小)에 ⑦ 연결 브라켓 조립



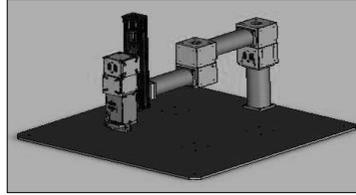
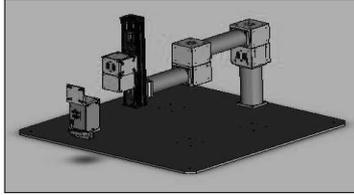
Step 9 ● ⑦ 연결 브라켓에 ② 회전 모듈 소(小) 조립



Step 10 ● ⑥ 그리퍼 모듈에 ⑦ 연결 브라켓 조립



Step 11 ● ⑥ 그리퍼 모듈 을 ② 회전 모듈 소(小)에 조립



<SCARA 로봇 실제 사진 자료 요약>



① 베이스판에 연결 브라켓 대 결합

M4-25볼트, 너트를 사용하여 결합한다.

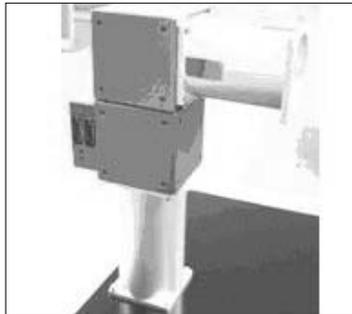


② 회전 모듈 대 연결

M4-25볼트를 사용하여 결합한다.

주의할 점

1. 모듈을 손으로 돌려서 회전 반경을 확인한 후 원하는 위치에 있는지 살펴 본 다음 결합한다.



③ 연결 브라켓 대 연결

M4-25볼트를 사용하여 결합한다.

주의할 점

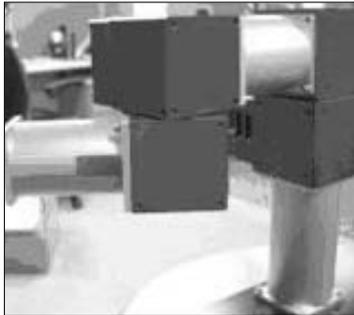
1. 볼트는 최소 2개 이상을 대각선 방향으로 대칭으로 결합한다.



④ 연결 브라켓 대에 회전모듈 중 결합

M4-25 볼트, 너트를 사용하여 결합한다.

1. 모듈을 돌려서 회전 반경을 확인 후, 원하는 위치에 있는지 살펴 본 다음 결합한다.



⑤ 연결 브라켓 소 연결

M4-25 볼트를 사용하여 결합한다.



⑥ 직선 모듈 소 결합

M4-25볼트를 사용하여 결합한다.

주의할 점

1. 엔드이펙트에 해당하는 부분과 연결 브라켓을 M4-20볼트를 먼저 연결한다. 이때 4개의 볼트를 먼저 가조립한 후 정조립한다.



⑦ 연결선 결합

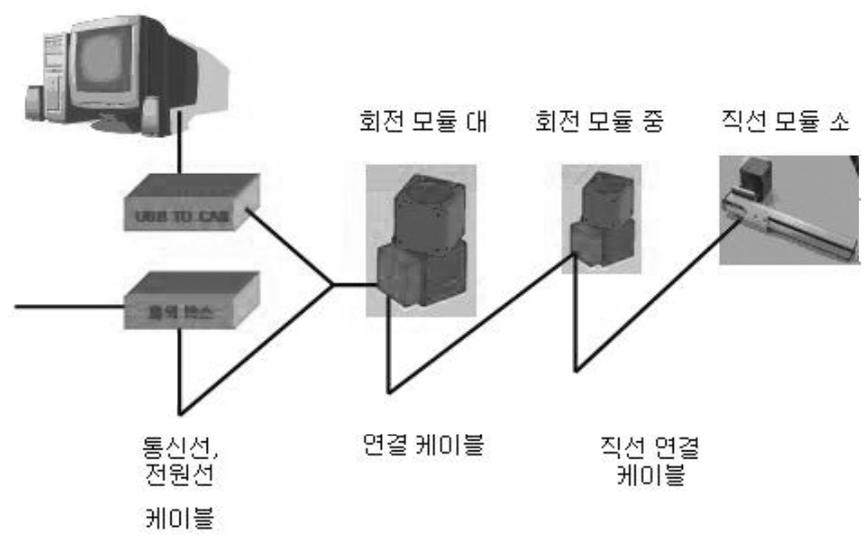
연결선을 제어기에 연결한다.

주의할 점

1. 연결선은 IN-OUT 구분이 없다.
2. 직선 연결 케이블의 경우 외부 센서를 연결한다.

	<p>㉘ 외부 통신선 전원선 연결</p> <p>연결선을 제어기에 연결한다.</p> <p>주의할 점</p> <ol style="list-style-type: none"> 1. 연결선은 IN-OUT 구분이 없다. 2. 직선 연결 케이블의 경우 외부 센서를 연결한다.
--	--

〈시스템 연결도〉





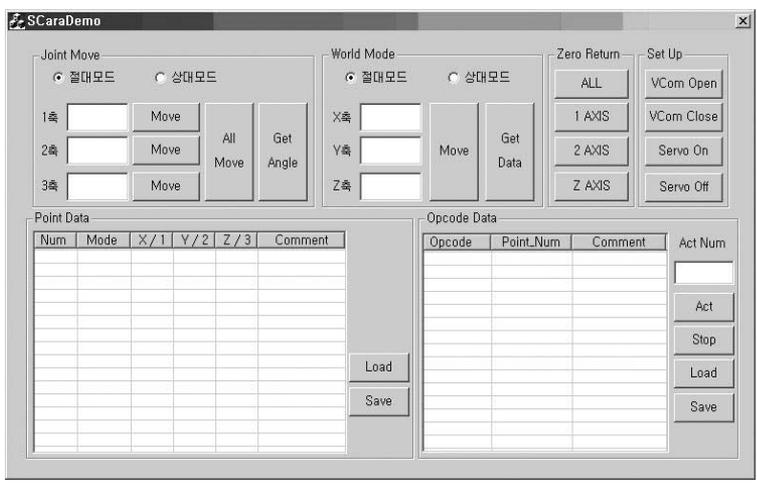
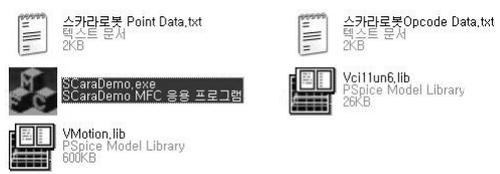
4 SCARA 로봇 프로그래밍

S/W 설치

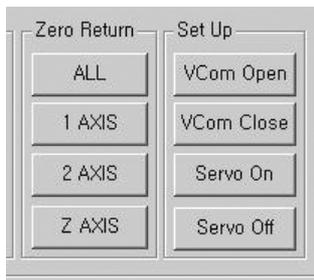


데모(DEMO) 프로그램

CD 안의 “파워 모듈 데모 프로그램” 디렉토리에 있는 SCaraDemo.exe 파일을 실행하면 된다.



데모 프로그램을 실행시키면 다음과 같은 화면을 볼 수 있다.
 여기서 로봇을 구동하기 위해서는 “VCom Open”을 클릭하여 USB to CAN과 연결한다.





연결이 완료된 후 로봇을 구동하기 전 반드시 Zero Return의 “ALL” 버튼을 클릭하여 원점복귀(Zero Return)를 수행한 후 로봇을 구동해야 한다.

보충설명

한 번 원점 복귀 되면 전원을 OFF하기 전까지는 다시 원점 복귀 하지 않아도 된다.

로봇의 동작 형태는 Joint Move에서 절대모드로 선택하면 된다.

보충설명

절대모드는 원점 복귀한 후 절대값으로 바로 이동하는 모드이다.

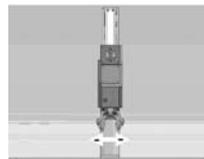
만일 10mm이동하라는 명령을 내렸을 경우 로봇은 10mm만 이동하게 된다. 그 후 또 10mm이동하라는 명령을 로봇에게 보내면 현재 10mm이동한 상태이므로 움직이지 않고 정지하게 된다. 상대모드는 현재 위치에서 입력한 값만큼 이동하는 모드이다.

현재 위치가 5mm이고 10mm이동하라는 명령을 보내면 10mm이동하여 최초 위치는 15mm를 이동하게 된다.

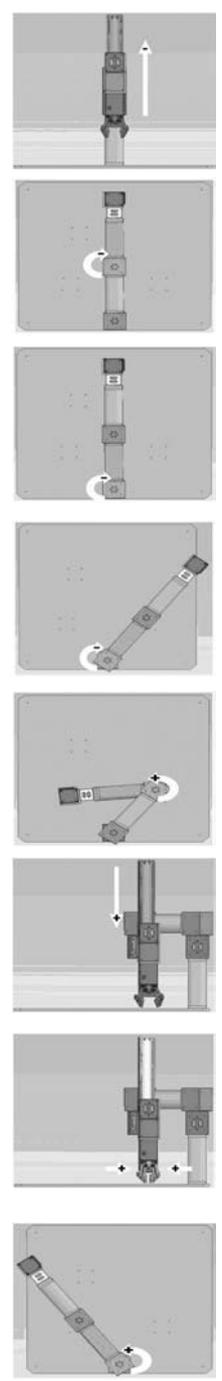
앞서 작성한 흐름도를 기반으로 해서 프로그램을 작성해 보도록 하자.

일단 작업에 의해 필요한 위치 포인트(Point)의 값은 다음과 같이 총 11개의 포인트가 필요하다.

1. 5축의 0mm 위치 포인트 값을 저장한다. (Point 1)

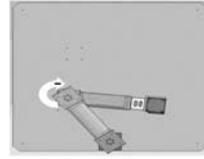


- 2. 3축의 0mm 위치 포인트 값을 저장한다.
- 3. 2축의 0도 위치 포인트 값을 저장한다.
- 4. 1축의 0도 위치 포인트 값을 저장한다.
- 5. 1축의 -37도 위치 포인트 값을 저장한다.
- 6. 2축의 +134도 위치 포인트 값을 저장한다.
- 7. 3축의 +60mm 위치 포인트 값을 저장한다.
- 8. 5축의 +30mm 위치 포인트 값을 저장한다.
- 9. 1축의 37도 위치 포인트 값을 저장한다.





10. 2축의 -134 위치 포인트 값을 저장한다.



이렇게 11개의 위치 포인트를 저장하고 난 후 저장된 포인트를 이용하여 실제 로봇 프로그램을 작성한다.

필요한 위치의 포인트가 전부 저장되었는지 다시 한번 흐름도를 확인해 본다.
저장하지 않은 부분이 있으면 로봇을 이동시켜 특정 위치의 포인트를 저장한다.

저장된 포인트를 이용해 다음과 같이 프로그램을 작성한다.

The screenshot shows the SCaraDemo software interface. It includes sections for Joint Move, World Mode, Zero Return, and Set Up. The Point Data table is as follows:

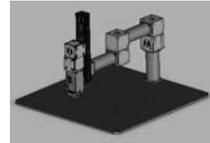
Num	Mode	X / 1	Y / 2	Z / 3	Comment
0	JOINT	0	900	200	01, Q2, Z
1	JOINT	0	0	0	01, Q2, Z
2	JOINT	0	-900	0	01, Q2, Z
3	JOINT	0	-900	200	01, Q2, Z
4	JOINT	-450	-600	0	01, Q2, Z
5	JOINT	600	-450	200	01, Q2, Z
6	JOINT	-900	0	100	01, Q2, Z
7	JOINT	900	0	0	01, Q2, Z
8	JOINT	0	700	200	01, Q2, Z
9	JOINT	-900	450	0	01, Q2, Z
10	JOINT	900	-600	200	01, Q2, Z
11	JOINT	-900	450	100	01, Q2, Z
12	WORLD	200	200	0	X, Y, Z
13	WORLD	200	150	200	X, Y, Z

The Opcode Data table is as follows:

Opcode	Point_Num	Comment	Act Num
MOVEJ	0	MOVE JOINT	
MOVEJ	2	MOVE JOINT	
MOVEJ	0	MOVE JOINT	
MOVEJ	1	MOVE JOINT	
MOVEJ	3	MOVE JOINT	
MOVEJ	0	MOVE JOINT	
MOVEJ	4	MOVE JOINT	
MOVEJ	8	MOVE JOINT	
MOVEJ	5	MOVE JOINT	
MOVEJ	11	MOVE JOINT	
MOVEJ	10	MOVE JOINT	
MOVEJ	3	MOVE JOINT	
MOVEJ	9	MOVE JOINT	
MOVEJ	6	MOVE JOINT	

구동 프로그램은 전체 프로그램을 바탕으로 순차적으로 구성하면 된다.

1. 전체 축의 원점복귀를 수행한다.



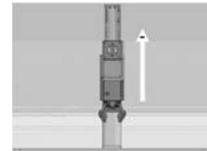
[ORIGIN]

2. 4축을 30mm 위치로 이동한다.



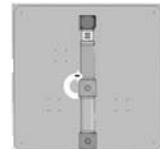
[MOVEJ 5 0 MOVE_JOINT]

3. 3축의 0mm 위치로 이동한다.



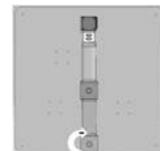
[MOVEJ 3 0 MOVE_JOINT]

4. 2축의 0도 위치로 이동한다.



[MOVEJ 2 0 MOVE_JOINT]

5. 1축의 0도 위치로 이동한다.



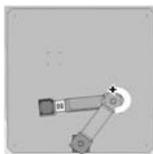
[MOVEJ 1 0 MOVE_JOINT]

6. 1축의 -37도 위치로 이동한다.



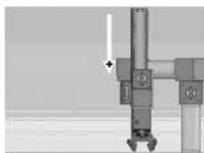
[MOVEJ 1 -37 MOVE_JOINT]

7. 2축의 +134도 위치로 이동한다.



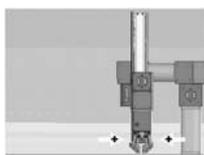
[MOVEJ 2 134 MOVE_JOINT]

8. 3축의 +60mm 위치로 이동한다.



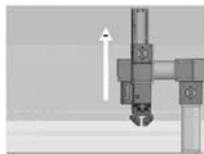
[MOVEJ 3 60 MOVE_JOINT]

9. 5축의 +30mm 위치로 이동한다.



[MOVEJ 5 30 MOVE_JOINT]

10. 3축의 0mm 위치로 이동한다.



[MOVEJ 3 0 MOVE_JOINT]

11. 2축의 0도 위치로 이동한다.



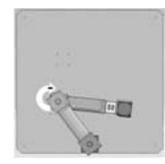
[MOVEJ 2 0 MOVE_JOINT]

12. 1축의 37도 위치로 이동한다.



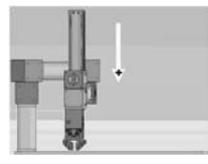
[MOVEJ 1 37 MOVE_JOINT]

13. 2축의 -134도 위치로 이동한다.



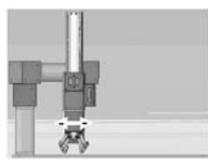
[MOVEJ 2 134 MOVE_JOINT]

14. 3축의 60mm 위치로 이동한다.



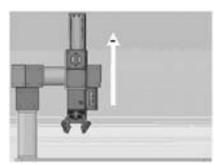
[MOVEJ 3 60 MOVE_JOINT]

15. 5축의 0mm 위치로 이동한다.



[MOVEJ 5 0 MOVE_JOINT]

16. 3축의 0mm 위치로 이동한다.



[MOVEJ 3 0 MOVE_JOINT]

17. 2축의 0도 위치로 이동한다.



[MOVEJ 2 0 MOVE_JOINT]



18. 1축의 0도 위치로 이동한다.



[MOVEJ 1 0 MOVE_JOINT]

이렇게 전체 로봇 동작에 대한 프로그램을 작성하면 실제 로봇의 동작이 흐름도와 같다는 것을 확인 할 수 있다. 흐름도를 잘 작성하면 프로그램을 쉽게 작성할 수 있다.

〈표 V-4〉는 전체 프로그램을 작성한 것이다.

```
[ORIGIN] // 전체 축을 Origin을 수행한다.

[MOVEJ 5 0 MOVE_JOINT] // 4축을 30mm 위치로 이동한다.
[MOVEJ 3 0 MOVE_JOINT] // 3축을 0mm 위치로 이동한다.
[MOVEJ 2 0 MOVE_JOINT] // 2축의 0도 위치로 이동한다.
[MOVEJ 1 0 MOVE_JOINT] // 1축의 0도 위치로 이동한다.
[MOVEJ 1-37 MOVE_JOINT] // 1축의 -37도 위치로 이동한다.
[MOVEJ 2 134 MOVE_JOINT] // 2축의 +134도 위치로 이동한다.
[MOVEJ 3 60 MOVE_JOINT] // 3축의 +60mm 위치로 이동한다.
[MOVEJ 5 30 MOVE_JOINT] // 5축의 +30mm 위치로 이동한다.
[MOVEJ 3 0 MOVE_JOINT] // 3축의 0mm 위치로 이동한다.
[MOVEJ 2 0 MOVE_JOINT] // 2축의 0도 위치로 이동한다.
[MOVEJ 1 37 MOVE_JOINT] // 1축의 37도 위치로 이동한다.
[MOVEJ 2 134 MOVE_JOINT] // 2축의 -134도 위치로 이동한다.
[MOVEJ 3 60 MOVE_JOINT] // 3축의 60mm 위치로 이동한다.
[MOVEJ 5 0 MOVE_JOINT] // 5축의 0mm 위치로 이동한다.
[MOVEJ 3 0 MOVE_JOINT] // 3축의 0mm 위치로 이동한다.
[MOVEJ 2 0 MOVE_JOINT] // 2축의 0도 위치로 이동한다.
[MOVEJ 1 0 MOVE_JOINT] // 1축의 0도 위치로 이동한다.
```

〈표 V-4〉 스카라 로봇 흐름도에 따른 전체 프로그램

단원 학습 정리



- 1 작업 분석에서 어느 축을 먼저 움직일 것인가에 대한 내용은 로봇이 움직일 때 주변의 다른 물체와 충돌하지 않도록 한다.
- 2 작성자가 어떤 순서로 모터를 구동시킬 것인지 결정되면 순서에 따라 흐름도를 작성하며 작성된 흐름도를 기본으로 각 구동 별 프로그램을 작성한다.
- 3 스카라 로봇의 베이스판에 연결 브라켓 대 결합은 M4-25볼트, 너트를 사용하여 결합한다.
- 4 회전 모듈 대 연결은 모듈을 손으로 돌려서 회전 반경을 확인한 후 원하는 위치에 있는지 살펴 본 다음 결합한다.
- 5 연결 브라켓 대 결합을 할 때 볼트는 최소 2개 이상을 대각선 방향으로 대칭으로 결합한다.
- 6 연결 브라켓 대에 회전 모듈 중 결합은 모듈을 돌려서 회전 반경을 확인하고 원하는 위치에 있는지 살펴 본 다음 결합한다.



- 7 직선 모듈 소 결합을 할 때 앤드이펙트에 해당하는 부분과 연결 브라켓을 M4-20볼트를 이용하여 연결한다.
- 8 연결선을 제어기에 연결할 때 주의할 점은 연결선은 IN-OUT 구분이 없으며 직선 연결 케이블의 경우 외부 센서를 연결한다.
- 9 외부 통신선 전원선 연결도 연결선 결합과 동일한 방법으로 연결한다.
- 10 데모 프로그램은 파워 모듈 데모 프로그램 디렉토리에 있는 SCarademo.exe 파일을 실행한다.





이동 로봇 제작

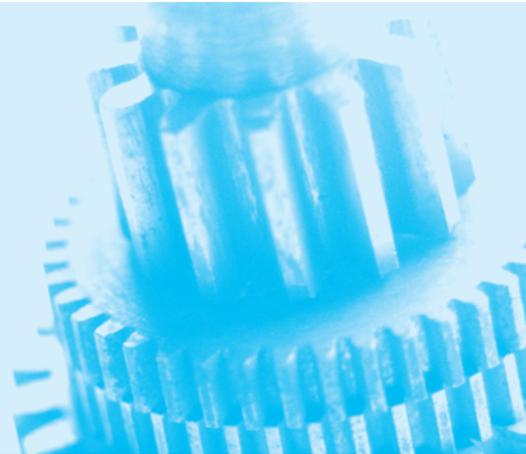
산업 현장에서 많은 물건들을 한 곳에서 다른 곳으로 이동해야 하는 일은 수 없이 많이 발생한다. 이러한 일을 사람이 직접 한다면 많은 인적 자원과 비용이 요구되고 정확성과 신뢰성 또한 많이 결여될 것이다. 이러한 현장에서 이동 로봇을 활용하면 상당한 생산 효율을 증대시킬 수 있을 것이다.

본 단원에서는 이동 로봇의 하드웨어와 소프트웨어를 이해하고, 이동 로봇의 조립 방법을 학습한다.



학습목표

1. 이동 로봇의 제작 설계 능력을 배양한다.
2. 이동 로봇의 소프트웨어 설계 능력을 배양한다.
3. 이동 로봇의 조립 능력을 배양한다.



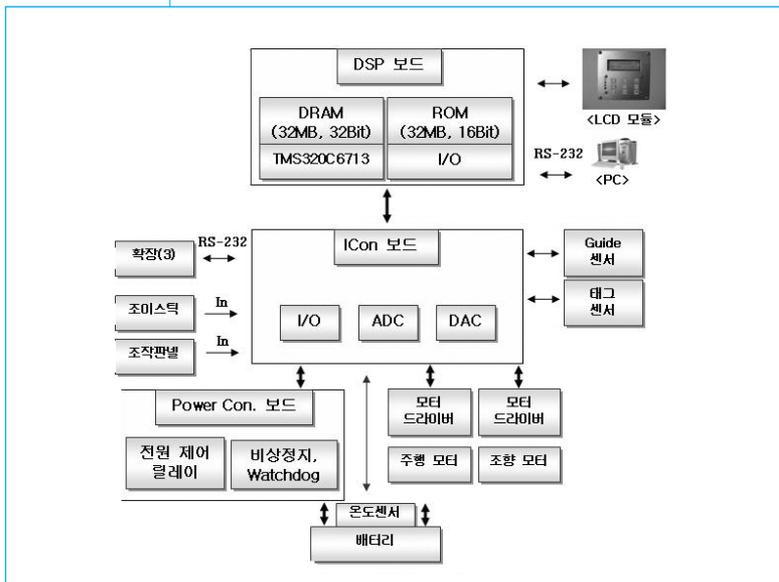
VI

이동 로봇 제작

1 이동 로봇 하드웨어 상세 설계

1 시스템 구조

이동 로봇용 제어기의 구성도는 <그림 VI-1>과 같이 제어기인 DSP 보드와 주변 I/O를 연결해주는 IO 제어보드(이하 ICon 보드), AGV(Automated Guided Vehicle), 내부 전원을 제어해 주는 보드(Power Control 보드), AGV 상태 등을 표시하거나 프로그램을 편집할 수 있는 LCD 모듈, 센서 및 스위치 등으로 구성되어



<그림 VI-1> 제어기 구성도

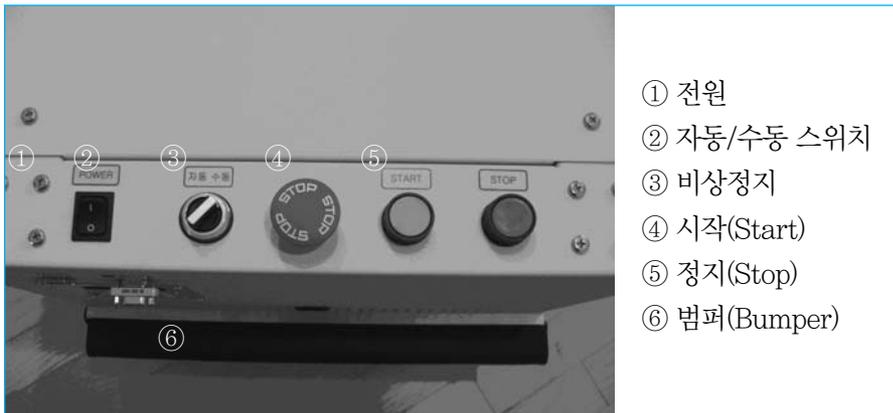
어 있다. 온도 센서는 충전하거나 시스템에 이상이 발생하여 일정 온도 이상 올라가면 자동으로 전원을 차단하기 위해 사용된다. <그림 VI-2>는 본 교재에서 다룰 이동 로봇의 외형 사진이다. <그림 VI-4>는 배터리 및 충전 단자에서 공급되는 전원이 각 보드에 공급되는 상태를 나

타내는 전원 계통도이다. <그림 VI-5>는 ICon 보드를 중심으로 센서 및 출력 기능들의 연결 상태를 나타낸 회로도이다.

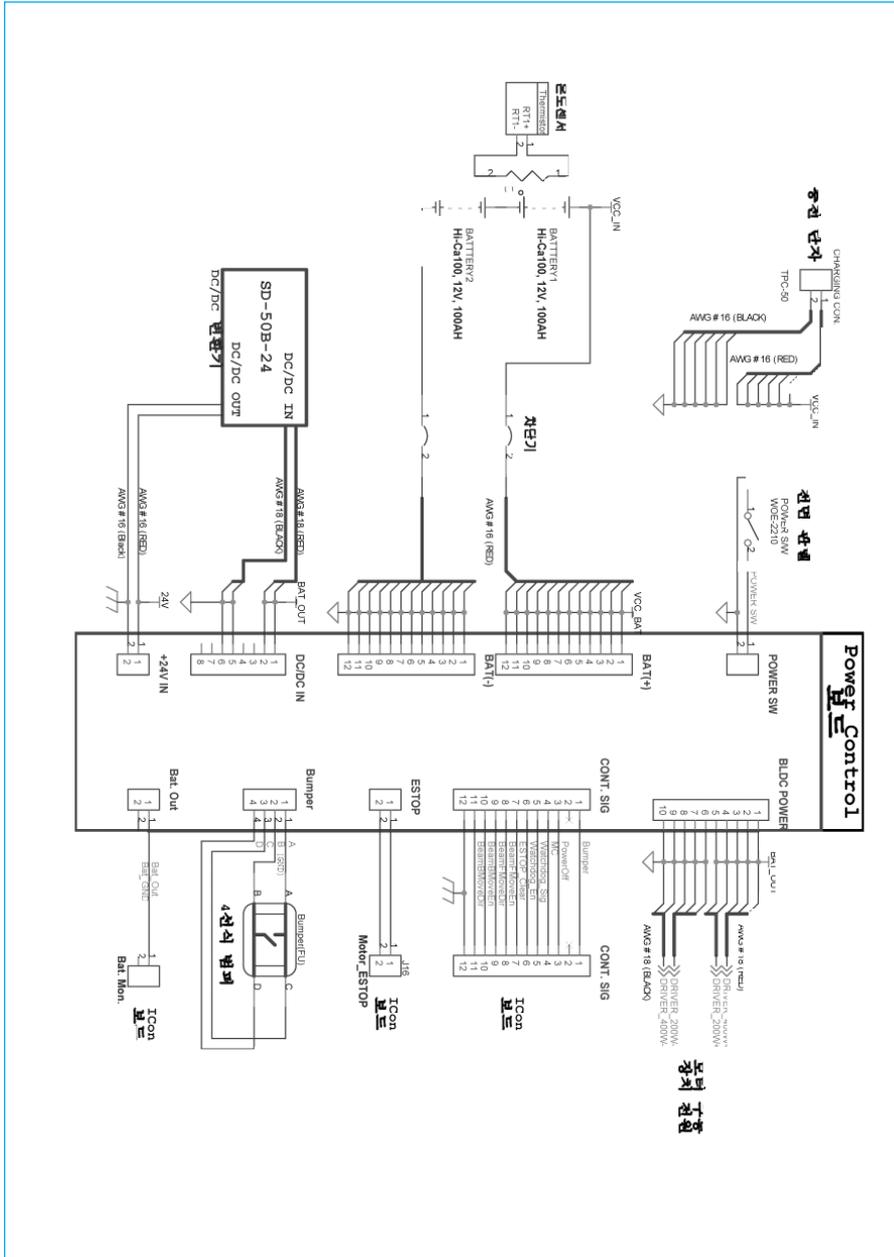


<그림 VI-2> 제작할 이동 로봇 사진

(1) 전면 조작부 설계



<그림 VI-3> 전면 조작 패널



〈그림Ⅵ-5〉 전장 배선도

1) 전원 스위치 (Power)

AGV의 전원을 On/Off 하기 위한 스위치이다. 장기적으로 AGV를 사용하지 않는 경우는 반드시 이 전원 스위치 및 내부 제어반에 있는 주 전원 차단기도 Off시켜 준다.

2) 자동/수동 선택 스위치

AGV의 동작 모드를 선택하기 위한 스위치이다. 자동의 경우 Start, Stop, 태그(Tag) 센서에 의한 가감속 등 자동 주행이 가능하다. 수동의 경우 전/후/좌/우의 4방향으로 운전이 가능하다. 자동 모드에서 Start 버튼을 1회 누르면 저속으로 AGV가 기동하며, 다시 한번 더 누르면 설정된 최고 속도로 가속하게 된다.

자동 모드가 선택되면 타워 램프에 초록색 램프가 켜지고 수동 모드가 선택되면 노랑색 램프가 켜지게 된다.

3) 비상 정지 스위치

이 스위치는 로봇이 오동작하거나 위험한 상황이 발생한 경우 누르는 버튼이다. 버튼이 동작하면 가장 먼저 프로그램에서 감속 및 정지를 시키며, 2차 안전장치로 Power Control 보드의 타이머에 설정된 시간 이후에 모터 전원을 자동으로 차단하여 AGV가 정지하게 된다. 이 기능을 Delayed E-Stop이라 하며, 반드시 하드웨어적으로 구현되어야 한다. 이 지연 시간은 타이머 값을 조정하여 1 ~ 10초까지 변경이 가능하다.

4) Start 버튼

자동 모드에서 AGV를 기동시키기 위한 버튼이다. 에러가 발생한 상황에서는 정지(Stop) 버튼을 이용하여 먼저 에러를 해제한 후 다시 버튼을 눌러 준다.



5) Stop 버튼

자동 주행 모드로 AGV가 이동하는 중간에 이 버튼을 누르면 AGV가 감속 후 정지하게 된다. 정지 상태나 에러가 발생한 상황에서 이 버튼을 다시 누르면 에러 상황 해제 및 AGV 컨트롤러의 각종 상태 및 I/O가 초기화된다.

6) 범퍼 스위치

로봇 이동 속도가 빠르고 차체의 중량 때문에 충돌에 의한 영향을 최소화하기 위해 스폰지 범퍼를 사용한다. 이 범퍼가 동작하면 프로그램에서는 부드럽게 감속을 하고, 일정 시간 이후에는 모터 드라이버에 공급되는 전원을 회로 (Power Control 보드의 기능)에서 자동으로 차단한다.

(2) 센서 연결 회로

이동 로봇의 안전 및 길을 찾기 위한 센서 등은 다음과 같은 기능을 갖는 센서가 사용된다. <그림 VI-6>은 가이드센서 및 태그(Tag) 인식 센서와 테이프의 배치에 대한 그림이며, 태그 인식용 테이프의 위치는 가이드 센서와의 간섭이 없는 위치에 배치하는 것이 필요하다.



- ① 전방 장애물 센서
- ② 태그용 테이프
- ③ 가이드 센서
- ④ 선로용 테이프
- ⑤ 태그 인식 센서

<그림 VI-6> 가이드 센서 및 태그인식 센서와 테이프 배치

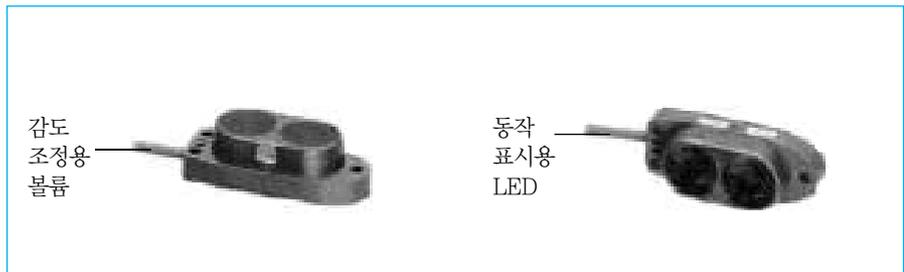
모터드라이버

MCU를 이용하여 모터를 제외하고자 할 때에는 모터 드라이버를 사용하여야 한다.

모터는 큰 전류를 필요로 하지만 MCU에서 보낼 수 있는 전류는 20mA에 불과하기 때문에 전류를 증폭시켜주는 기능을 하는 모터드라이버가 사용되어야 한다.

1) 전방 장애물 센서

적외선 방식의 반사형 센서로 검출 거리는 최대 2M이며, 센서 사용 여부는 LCD 모듈의 키를 이용하여 선택이 가능하도록 되어 있다. 이 센서는 단순히 전방 물체 유무를 검출하기 위한 용도이며, 사람이나 물건이 주행 경로에 자주 있는 경우는 보다 정밀하고 감도가 좋은 센서를 사용하는 것이 좋다. 센서 감지 거리는 <그림 VI-7>과 같이 센서 윗면에 있는 가변 저항기를 돌리면 거리를 변경할 수 있다.

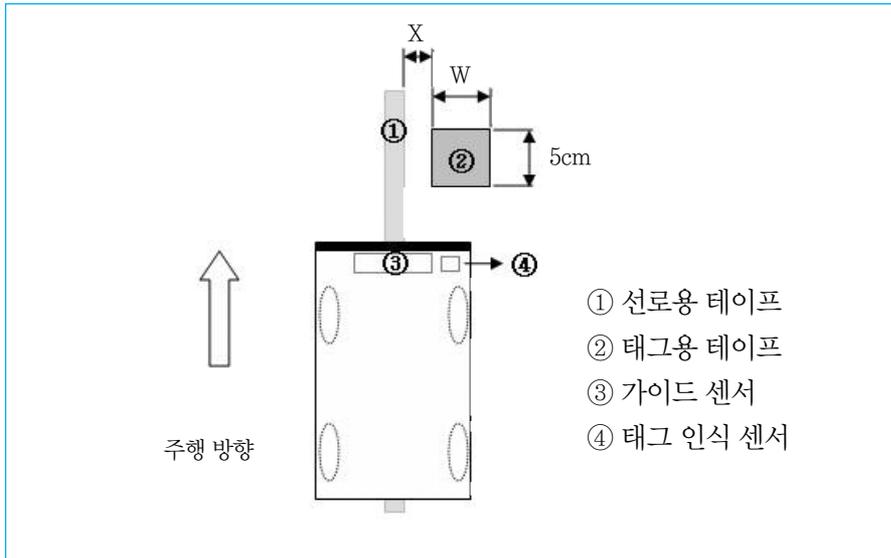


<그림 VI-7> 장애물 센서

2) 태그 인식 센서

이 센서는 AGV의 현재 주행 구간을 알 수 있도록 하기 위한 것으로 <그림 VI-8>과 같이 설치한다. 태그(Tag)용 재질이 주행 선로와 동일한 써스를 사용하는 관계로 상호 간섭이 없도록 설치하는 것이 필요하다.

테이프의 폭(W : 최소 10cm)은 가이드 센서가 인식하지 않는 범위에서 최대 폭으로 설치하며, 가이드용 테이프에서 태그용 테이프까지의 거리(x)는 7cm 정도에서 상호 간섭이 없도록 설치한다.



〈그림VI-8〉 테이프 설치 위치

3) 가이드 센서

빛을 반사하는 테이프 인식용 적외선 센서 모듈로 15비트 디지털 출력 타입이다. 감지 거리는 20~40mm이며 인식할 테이프와는 반드시 수직을 유지하도록 한다. 센서의 밑면에 이물질이 묻거나 흠집이 가는 경우 센서 감도가 떨어질 수 있으니 주의가 필요하다. 그리고 설치된 높이를 조정하면 인식 감도를 조정할 수 있다.

2 이동 로봇 소프트웨어 상세 설계

이동 로봇을 움직이는 방법에 따라 수동 모드와 자동 모드로 구분하며, 각 모드에서 구현할 기능은 다음과 같다.

1 수동 모드

1) 이동 관련 기능 설계

이동 로봇 전면에 있는 TP(Teaching Pendant) 단자에 수동 조작기를 연결한다.

〈그림Ⅶ-9〉와 같이 일반 게임기용 조이스틱(Joystic)을 개조하여 AGV를 전/후/좌/우로 원하는 위치로 이동할 수 있다. 동작 방향은 AGV 차체를 기준으로 한다. 수동 조작은 조이스틱의 전면에 있는 D_M(Dead_Man) 버튼을 항상 On 시킨 상태에서만 가능하다.

D_M 버튼이란 사람이 위험한 상황에서 무의식적으로 버튼을 놓기 때문에 이 버튼이 동작할 때만 움직이도록 한다면 보다 안정적인 로봇이 될 수 있기 때문에 수동 조작기에 필수적으로 들어가는 버튼이다.

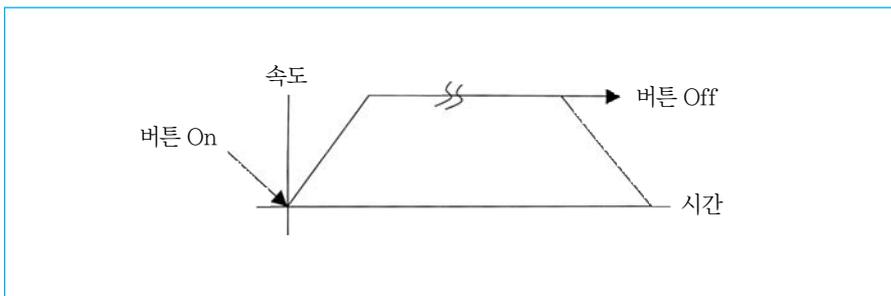
수동 조작 모드에서의 사용 편리성을 강화하기 위하여 D_M 버튼에 따라 주행 모터의 다이내믹 브레이크 기능(모터 3상을 쇼트(Short)시켜 모터를 회전하지 못하도록 하는 기능)을 선택적으로 사용할 수 있도록 하였다. D_M 버튼이 On되면 다이내믹 브레이크가 동작하고, Off되면 모터가 자유롭게 회전하는 자유주행 상태가 된다.



〈그림Ⅶ-9〉 수동 조작기(TP) 외형



수동 모드에서의 주행 및 조향 속도는 안전을 위하여 자동 모드의 70% 정도로 설계되어 있다. 주행 및 조향 부분의 속도 출력 방법은 <그림Ⅵ-10>에서 보는 바와 같이 같은 버튼을 계속 누른 경우는 최고 속도까지 가속을 한 후 등속을 유지하고, 버튼이 Off되거나 반대편 버튼이 동작한 경우는 바로 감속하도록 하였다. 또한 수동 모드에서는 배터리 전압을 체크하지 않도록 하여 배터리가 과방전 상태라도 이동할 수 있도록 되어 있다.



<그림Ⅵ-10> 수동 모드 속도 궤적

배터리 과방전 시 운행

자동모드에서는 배터리라 방전 시 오동작의 가능성이 있기 때문에, 일정수준 이상 방전되었을 때에는 안전을 위하여 동작이 자동으로 멈추도록 되어 있다.

2) 안전 장치 처리 방법

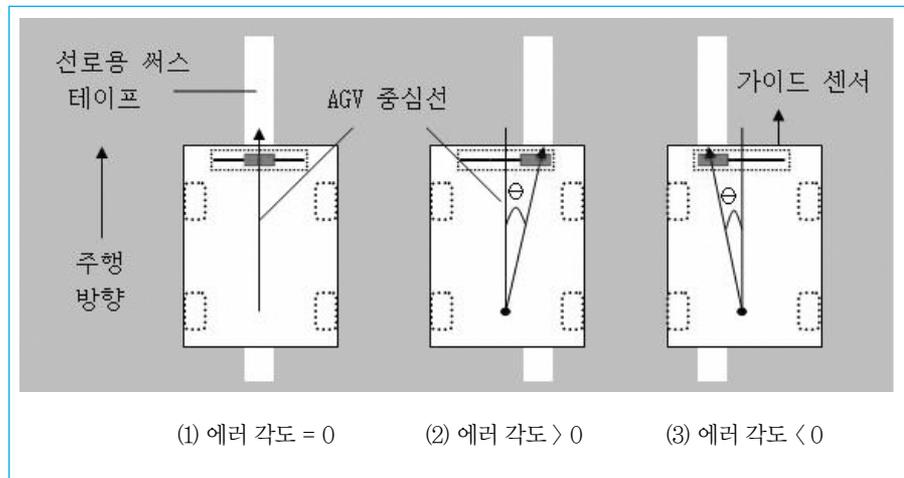
AGV가 충돌로 범퍼가 고장난 경우 잘못 하면 AGV를 이동할 수 없게 된다. 이러한 상황에서도 최소한 수동으로 AGV를 조작할 수 있도록 범퍼의 사용 여부를 선택할 수 있게 되어 있다. 그리고 수동 모드에서는 장애물 센서를 사용하지 않도록 되어 있다.

2 자동 모드

1) 조향 제어 방법

가이드 센서의 동작 중심을 구해서 <그림Ⅵ-11>의 (1)과 같이 예러 각도가 0인 경우 계속 똑바로 움직이도록 조향각 명령을 0으로 출력하며, <그림Ⅵ-11>

의 (2)와 같이 AGV가 오른쪽으로 치우쳐 주행하고 있는 경우는 조향 모터를 오른쪽으로 돌려 센서의 중심을 찾도록 한다. <그림 VI-11>의 (3)의 경우는 AGV가 왼쪽으로 치우쳐 주행하고 있는 상태로 조향 모터를 왼쪽으로 돌리면 AGV가 테이프의 중심으로 이동할 수 있다.



<그림 VI-11> 조향 제어 방법

(2) 주행 제어 방법

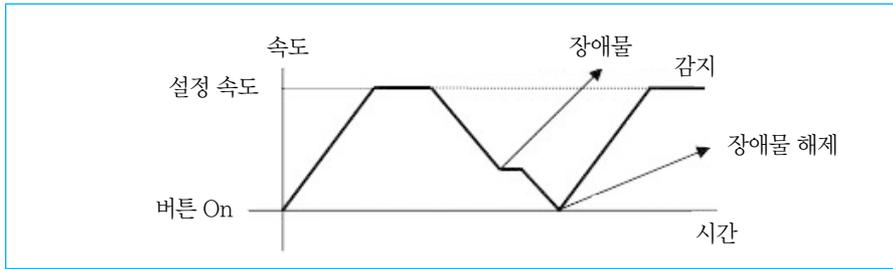
1) 주행 속도에 영향을 미치는 부품

이동 로봇을 주행하면서 정지 위치에서 정지하거나 부착된 센서의 동작에 의해 감속시킬 수 있다. 감속 단계는 2단계로 구분하였으며 최고 속도의 40% 수준과 20%로 감속하게 된다. 40%는 일반적인 회전 모드에서 사용하는 속도이고, 20%는 이동 로봇이 정지하기 전에 정밀한 정지를 위한 속도로 충분히 낮추기 위해 사용된다.

- ① 장애물 센서 : 이동 로봇의 전면에 부착된 장애물 센서가 동작하는 경우 감속(40%, 약 8cm 감속 주행)해서 정지하고, 해제되면 다시 가속하여 이전 속도로 주행하도록 한다. 이 가감속 프로그램은 타이머 인터럽트 루틴



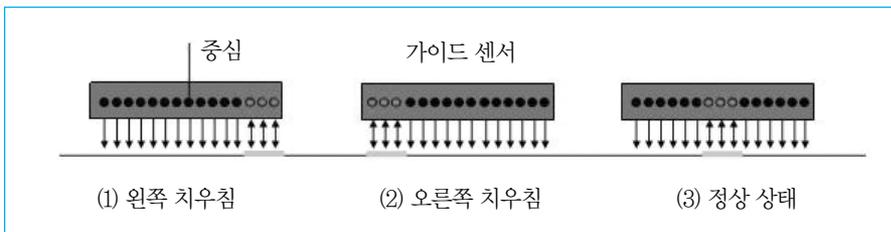
에서 주기적으로 실행된다. <그림 VI-12>는 장애물 센서가 동작하였을 때의 AGB 주행속도 궤적을 나타낸 그림이다.



<그림 VI-12> 장애물 센서에 의한 속도 궤적

② 감속 명령 : 프로그램에 의해 설정된 속도로 감속하는 것으로 10~100% 범위에서 임의로 설정이 가능하다. AGV가 회전하는 경우는 40% 정도의 값이 적당하며 정지하기 전에는 20%이하로 감속하면 정지 오차를 줄일 수 있다. 이동 로봇의 사용 조건에 따라 너무 감속을 하게 되면 움직이지 못하는 경우도 발생할 수 있다. 이런 상황에서는 속도를 조금 높여준다.

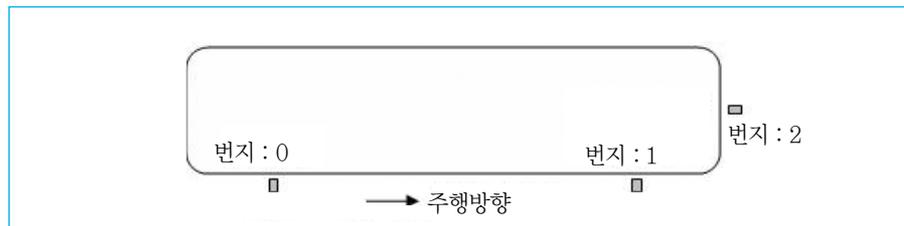
③ 가이드 센서 : 선로를 인식하는 중요한 센서로 <그림 VI-13>과 같이 중심으로부터 3개 이상 센서가 감지되지 않는 경우엔 감속하게 되고, 재가속은 1초 이상 센서의 중심이 동작하는 안정된 경우에서 수행하도록 하였다. 방향의 기준은 이동 로봇을 중심으로 한다.



<그림 VI-13> 가이드 센서에 의한 가감속 조건

2) 주행 제어 모드

이동 로봇의 주행 알고리즘은 직진 구간을 고속으로 주행하는 직선 주행 모드와 회전 구간에서 저속 회전 모드로 구분해서 설계되어 있다. 이러한 주행 모드를 선택하여 사용하기 위해서는 속도를 명령해야 한다. 예를 들어 40% 이하로 속도를 명령하면 회전 모드로 알고리즘이 선택되고, 40%보다 크게 되면 직진 모드로 동작하게 되는 것이다. <그림 VI-14>의 선로를 AGV가 주행할 때 바람직한 속도 설정 방법은 번지 0에서 1로 이동할 때 고속으로 주행할 수 있도록 주변 환경을 고려하여 최고 속도로 설정한다. 그런 다음 1번지에서는 회전 속도로 주행할 수 있도록 40%를, 2번지에서는 다시 최고 속도로 설정한다. 최고 속도는 로봇 주변이 충분히 안전한 상태에서만 시험해야 한다.

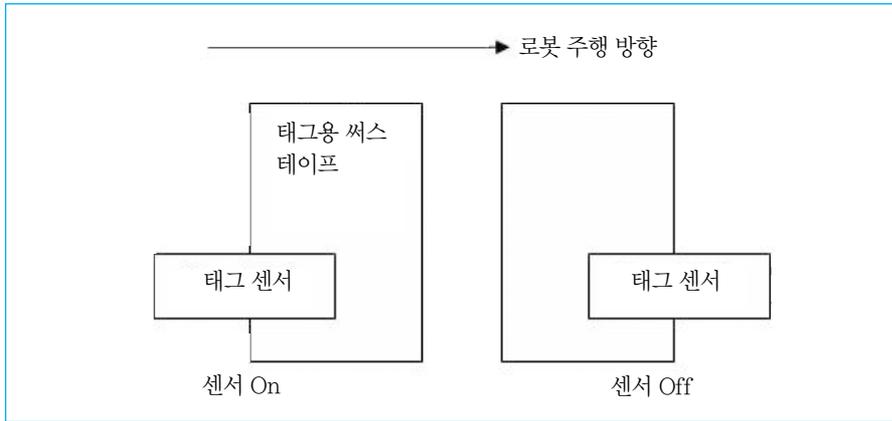


<그림 VI-14> 속도 설정 보기

3) 태그 인식 방법

번지를 인식하기 위한 태그는 이동 로봇의 주행 방향의 오른쪽에 설치되며 크기는 5x10cm인 알루미늄 반사 테이프를 사용한다. 이 테이프의 길이를 인식하는 방법은 <그림 VI-15>와 같이 이동 로봇이 주행하면서 태그 센서가 On 되는 시점에서 Off되는 시점까지의 주행 거리가 태그의 길이가 된다. 현재 설계된 모터의 인코더 펄스당 직선 주행 거리는 0.43cm이다. 따라서 태그 크기는 아래 식과 같이 계산을 하면 된다. 5cm±2.5 이내의 크기는 정상적인 태그로 인식하고 기타 크기의 태그는 무시하게 된다.

$$\text{태그 사이즈} = 0.43 \times (\text{On 시점에서의 모터 펄스 수})$$



〈그림 VI-15〉 태그 길이 인식 방법

3 이동 로봇 조립

1 구성품

로봇 조립에 사용될 부품은 다음과 같다.

1) 배터리

- 제품명 : Hi-Ca 100
- 제조사 : Delco
- 특징 : 12V, 100AH, (최대 55%까지 사용 가능)



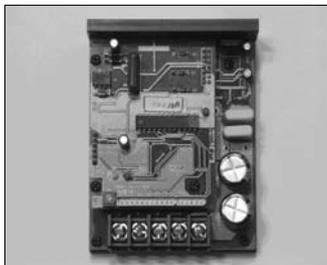


2) 배터리 연결 케이블 1개



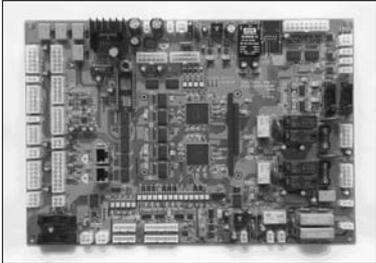
3) 모터 드라이버

- 제품명 : SBDM-25
- 제조사 : 세우산전
- 특징 : 구동 전류 최대 25A용, 주행 및 조향 드라이버가 바뀌지 않도록 주의한다.
- 수량 : 2개(주행용 4극 모터, 조향용 4극 모터, 조향용 8극 모터)



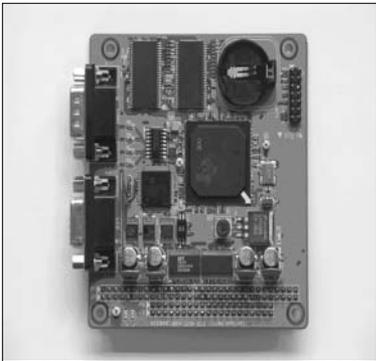
4) ICon 보드

- 제품명 : CTS-ICON-EA01
- 제조사 : 캔탑스
- 특징 : 이동 로봇의 모든 Input/Output을 DSP 보드와 연결시켜 주는 보드
- 수량 : 1개



5) DSP 보드

- 제품명 : CTS-AGVC-AA01
- 제조사 : 캔탐스
- 특징 : 이동 로봇을 제어하는 보드
- 수량 : 1개



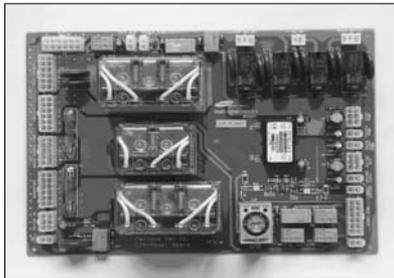
6) LCD 모듈

- 제품명 : CTS-AGVL-BA01
- 제조사 : 캔탐스
- 특징 : 2x16 문자의 LCD 및 12개의 버튼을 시리얼로 연결하는 모듈
- 수량 : 1개



7) Power Control 보드

- 제품명 : CTS-PCON-XA03
- 제조사 : 캔탑스
- 특징 : 이동 로봇의 전원을 제어하는 보드
- 수량 : 1개



8) 스피커 1개

- 수량 : 1개



9) 장애물 센서

- 제품명 : BA2M-DDT
- 제조사 : Autonics
- 특징 : 이동 로봇의 전방 장애물을 감지하는 IR 센서
- 수량 : 1개~13개



10) 범퍼

- 제품명 : BS-568-A
- 제조사 : 삼신센서
- 특징 : 스폰지형 범퍼, 4선식 출력
- 수량 : 1개



11) 좌/우측 리미트 센서

- 제품명 : PR12-4DN
- 제조사 : Autonics

- 특징 : 조향 모터의 과전류를 방지하기 위한 리미트(Limit) 센서
- 수량 : 1개



12) 가이드 센서

- 제품명 : CTS-IRGS-LF01
- 제조사 : 켈탐스
- 특징 : 선로(테이프)의 중심을 찾기 위한 15Bit IR 센서
- 수량 : 1개



13) 번지 카운터(Tag)용 센서

- 제품명 : CTS-IRGS-LD01
- 제조사 : 켈탐스
- 특징 : 어드레스를 카운트하기 위한 써스 테이프 인식 센서
- 수량 : 1개



2 조립

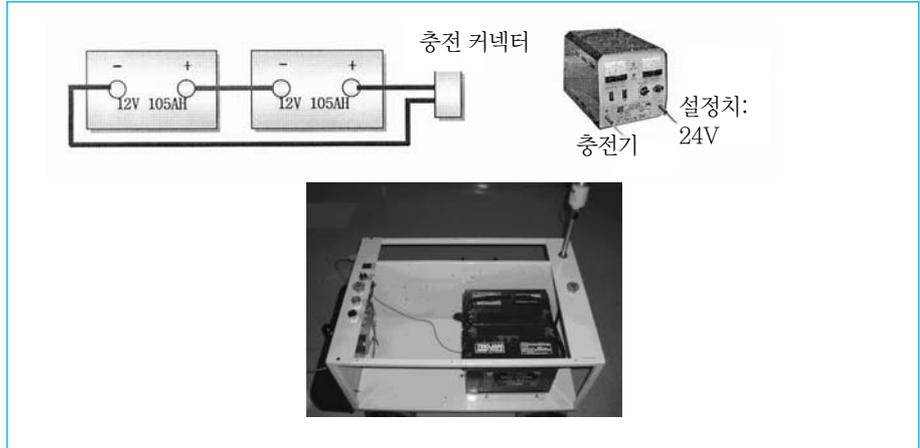
케이블을 잘못 꼽는 경우 부품 파손 및 화재로 사람이 다칠 수 있기 때문에 철저한 확인과 안전 작업이 필수적이다.

1) 배터리(Battery) 조립

〈그림 VI-16〉과 같이 배터리의 극성을 확인하고 케이블(그림의 적색 케이블)을 이용하여 배터리를 연결한다. 그리고 차체에 연결된 +극성 케이블과 -극성 케이블을 각각 +단자 및 -단자에 연결한다.

방전이 된 경우에는 AGV 전원을 Off시킨 상태에서 〈그림 VI-16〉의 충전기를 24[V]로 설정하고, 전류는 0[A]로 설정한 후 충전 단자에 연결하고 충전기 전원을 켜다. 전류를 증가시키면서 최대 10[A]로 설정한다. 과방전이 아닌 상태에서는 전류가 10[A] 이하로 흐를 수 있다. 너무 무리하게 전류를 높이지 않는 것이 좋다.

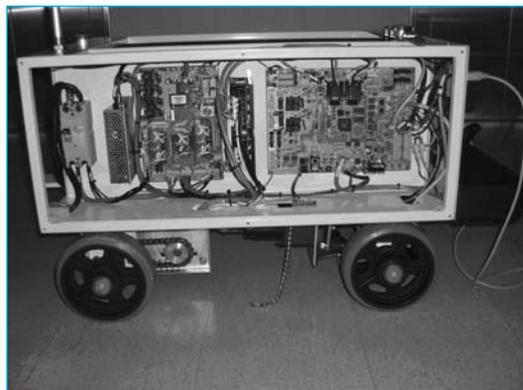
배터리 극성을 반대로 연결하는 경우 보드의 부품 파손 및 화재 위험이 있기 때문에 각별한 주의를 요한다.



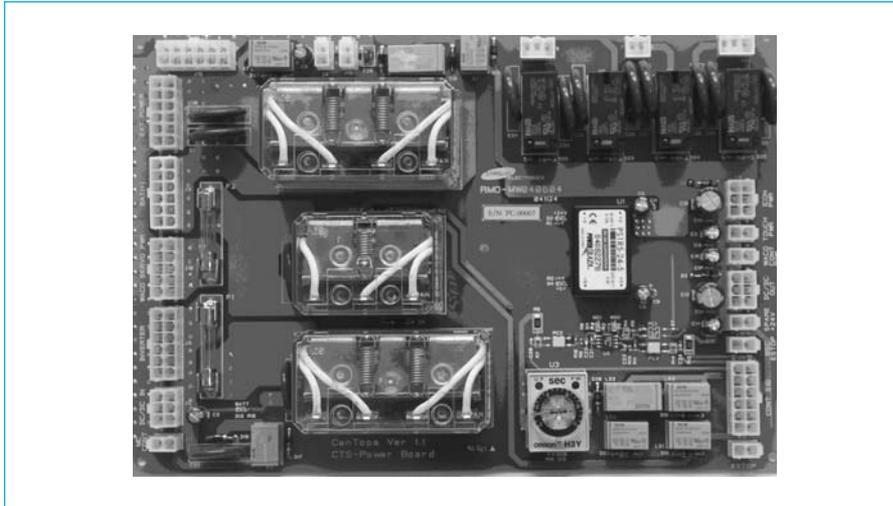
<그림 VI-16> 충전기 및 배터리 연결도

2) Power Control 보드 조립

먼저 보드를 <그림 VI-17>의 전장부 조립 사진과 같이 Power Control 보드의 방향을 확인한 후 조립하고, <그림 VI-18>과 같이 보드에 인쇄된 커넥터 이름과 케이블 이름이 일치하도록 커넥터를 연결한다. 케이블 커넥터의 모양 및 크기가 같을 수 있기 때문에 잘못 삽입할 경우 부품 파손 및 화재의 위험이 있으므로 각별한 주의가 필요하다.



<그림 VI-17> 전장부 조립 사진



〈그림 VI-18〉 Power Control 보드 커넥터 배치

① DC/DC IN

〈그림 VI-17〉의 흰색 타원으로 표시된 부품이 DC/DC 변환기이며, 이 변환기는 이동 로봇 제어기에 사용되는 24V 전원을 공급하는 기능을 한다. 이 DC/DC 변환기를 이용하여 배터리 전원과 제어기 전원이 절연될 수 있도록 하였다. 이 단자는 DC/DC 변환기의 입력 전원 단자와 연결된다.

② WACO Power

주행 및 조향용 모터 드라이버에 필요한 전원을 공급하는 단자로 모터 드라이버의 전원 단자와 연결된다.

③ BAT+, BAT-

배터리의 + 및 - 단자와 연결되는 커넥터이다. 거꾸로 삽입되지 않도록 주의한다.

배터리 역산 주의

배터리를 거꾸로 연결하여 삽입할 경우 로봇회로부의 파손 및 과열에 의한 화재도 발생할 수 있기 때문에 각별히 주의하여야 한다.

④ BAT OUT

ICon 보드에서 배터리 전압을 측정할 수 있도록 출력한 배터리 전압 신호이다.

⑤ Power S/W

전면 판넬의 전원 On/Off 스위치의 신호선이 연결된다.

⑥ DC/DC OUT

Power Control 보드에 필요한 24V 전원을 공급하기 위한 단자이다.

⑦ Cont Sig.

Power Control 보드에 범퍼 동작 신호를 입력하기 위한 단자이다.

⑧ 비상 정지

이동 로봇에 장착된 2개의 비상 정지 스위치의 상태를 Power Control 보드의 타이머 회로를 동작시키기 위한 입력 신호이다.

⑨ 타이머

비상 정지가 동작하고 하드웨어적으로 모터 드라이버에 공급되는 전원을 자동 차단하는 시간을 설정하는 기능을 한다. 1~2초 정도의 설정이 적당하다.

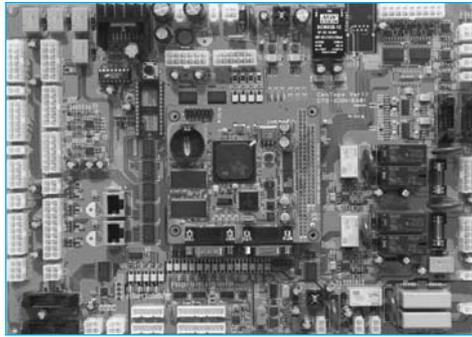
3) DSP 보드 조립

〈그림 VI-19〉와 같이 ICon 보드 위에 DSP 보드를 연결 커넥터의 핀 번호를 확인한 후 삽입한다. 삽입 후 보드간 연결 커넥터가 정상적으로 꼽혔는지 확인이 필요하다. 이상이 없으면 나사 4개를 이용하여 DSP 보드와 ICon 보드를 결합시킨다.

DSP 보드에 있는 두 채널의 시리얼 포트는 LCD 모듈 및 Upgrade 포트에



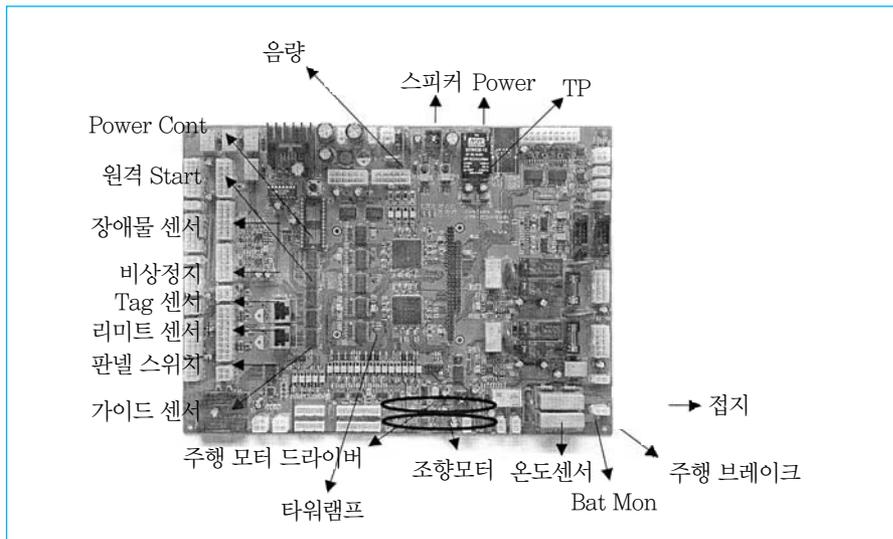
연결한다.



〈그림 VI-19〉 ICon 보드 외관

4) ICon 보드 조립

〈그림 VI-17〉과 같이 보드를 차체에 고정하고 연결 케이블의 이름과 보드상에 인쇄된 이름을 반드시 확인하면서 조립한다. 유사한 커넥터가 많이 있기 때문에 결선 상태를 모두 확인하고 다음 단계를 진행한다.

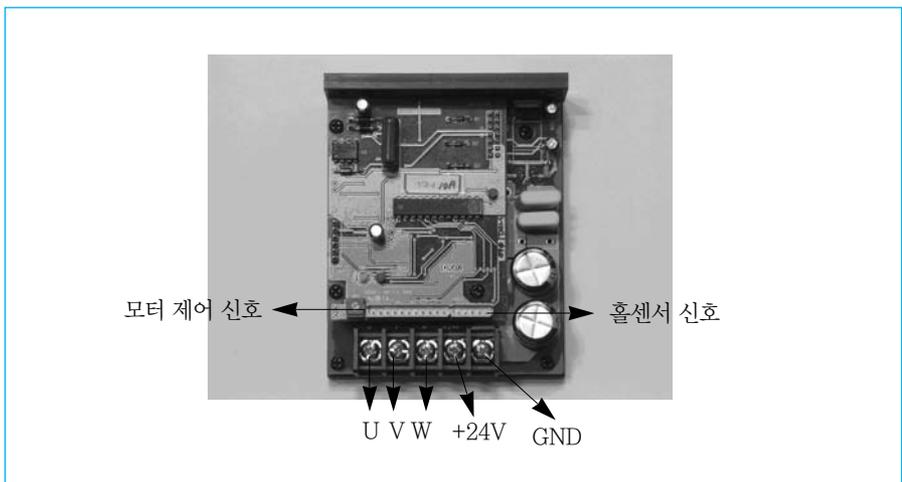


〈그림 VI-20〉 ICon 보드 커넥터 배치



5) 모터 드라이버 조립

〈그림Ⅵ-17〉과 같은 위치에 모터 드라이버를 고정한다. 위에 있는 모터 드라이버가 주행용 모터 드라이버이고, 아래는 조향용 모터 드라이버이다. 두 모터 드라이버의 특성이 전혀 다르기 때문에 위치가 잘못 조립된 경우 부품이 파손될 수 있다.



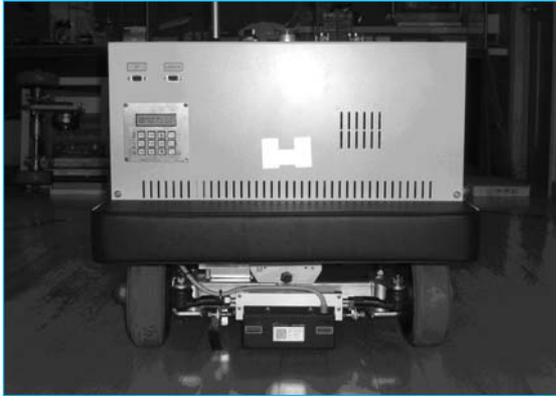
〈그림Ⅵ-21〉 모터 드라이버 커넥터 배치

U, V, W 신호 및 홀센서 신호는 모터 모터와 직접 연결되는 케이블이며, 모터 제어 신호는 ICon 보드와 연결되는 커넥터이다.

+24V와 GND는 Power Control 보드에서 공급되는 모터 드라이버의 전원선이다. 이 전압은 배터리 전압을 그대로 이용하며 범퍼나 비상 정지 스위치가 동작하면 전원은 자동으로 차단된다. 이 전원선이 반대로 연결 되면 내부 부품 파손 및 화재의 위험이 있다. 위에 있는 드라이버가 주행용 모터(4극 15A)이며, 아래에 있는 모터가 회전을 제어하는 조향용 모터(8극 10A)다.



6) 가이드 센서 조립



〈그림 VI-22〉 가이드센서 고정 위치

〈그림 VI-22〉와 같이 가이드 센서의 고정판에 가이드 센서를 고정한다. 가이드 센서의 케이블은 ICon보드와 연결되며, 이 케이블은 로봇이 동작할 때 파손되지 않도록 케이블 타이를 이용하여 고정한다. 그리고 센서는 반드시 바닥면과 수직인 상태가 써스 테이프를 가장 잘 인식할 수 있다. 각도를 확인하면서 조립한다.

7) 좌/우 리미트(Limit) 센서 조립

좌/우 리미트 센서는 조향 모터의 과전류를 방지하기 위해 기구의 좌/우측 끝전에 이 센서를 부착하여 기구적인 충돌을 막기 위한 것이다. 이 센서는 매우 감지거리가 짧고 정밀하기 때문에 조립 후 수동 모드에서 TP로 좌/우 리미트 센서가 정상 동작하는지 확인이 필요하다. 또한 좌측과 우측 리미트 센서가 서로 바뀌면 전혀 다른 동작을 하기 때문에 결선 시 주의가 필요하다. 케이블이 동작 부위와 간섭이 없도록 고정 후 로봇을 동작시킨다.



〈그림Ⅵ-23〉 조향 후 리미트 센서 배치

8) 태그(번지 카운터용) 센서



〈그림Ⅵ-24〉 태그 센서 고정 위치

태그 센서도 가이드 센서와 같이 지면과 수직일 때가 가장 좋은 성능을 낼 수 있다. 고정할 때 수직이 되도록 조립하며, 바닥면의 반사 특성이 좋아 바닥에서도 센서가 동작하는 경우엔 센서를 위로 이동하면서 바닥이 감지되지 않도록 조정이 필요하다. 이 케이블은 ICon 보드의 B_Stop이라는 커넥터와 연결된다.

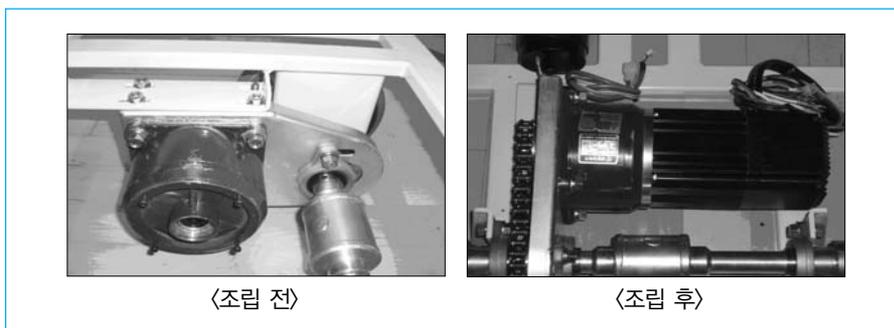
9) 범퍼 조립



〈그림Ⅵ-25〉 범퍼 조립도

〈그림Ⅵ-25〉와 같이 범퍼를 조립하고 케이블은 Power Control 보드의 확장 연결 케이블과 연결한다.

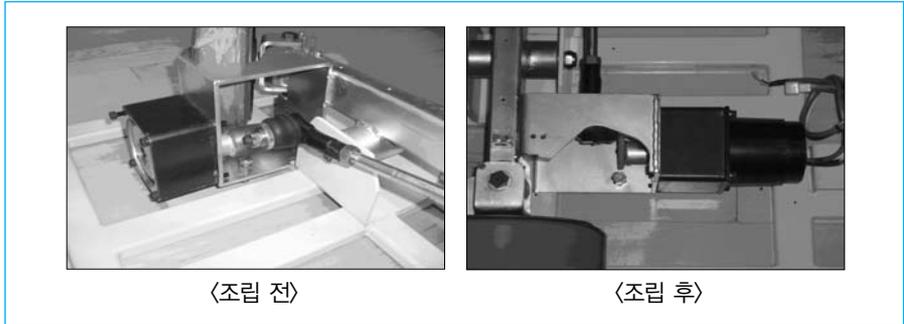
10) 주행 모터 조립



〈그림Ⅵ-26〉 주행 모터 조립도

〈그림Ⅵ-26〉과 같이 주행용 400W 모터를 감속기의 고정홀과 일치시킨 후 조립한다.

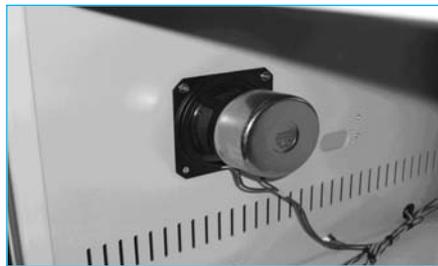
11) 조향 모터 조립



〈그림 VI-27〉 조향 모터 조립도

〈그림 VI-27〉과 같이 주행용 250W 모터를 감속기의 고정홀과 일치시킨 후 조립한다.

12) 스피커 조립



〈그림 VI-28〉 스피커 조립도

전면판에 너트로 고정하고, 케이블은 ICon 보드의 스피커 단자에 연결하면 된다. 소리 크기는 주변 상황에 따라 ICon 보드의 음량 조절 볼륨을 조정하면 된다. 이 소리 및 타워 램프에서 발생하는 부저(Buzzer) 소리를 끄는 기능은 전면 LCD 패널(Pannel)에서 설정이 가능하다.

13) 온도 센서 연결



〈그림Ⅵ-29〉 온도 센서 배치

온도 센서는 배터리의 과열을 검출해서 이상 과열 상태를 방지하기 위한 기능을 한다. 온도 센서는 두 배터리 사이의 온도를 측정할 수 있도록 배터리 사이에 넣고, 케이블은 ICon 보드에 연결한다.

14) LCD 모듈 조립



〈그림Ⅵ-30〉 LCD 모듈 조립도

이동 로봇의 전면에서 LCD 모듈을 삽입하고 4개의 고정 나사로 조립한다. 시리얼(Serial) 연결 케이블은 DSP 보드와 연결한다.



4 이동 로봇 프로그래밍

1 명령어 사용법

1) 용어 정의

AGV를 현장 상황에 따라 다양한 방법으로 이용할 수 있도록 간단한 구조의 명령어를 이용하여 프로그램을 할 수 있다.

- ① 프로그램 번호 : 최대 16개의 프로그램을 저장할 수 있으며, AGV가 수행할 일련의 기능들을 정의한 프로그램 이름(번호)을 의미한다.
- ② 번지(Address) : AGV가 주행하면서 태그(Tag)를 인식할 때마다 1씩 증가하며 최대 32개까지 태그를 사용할 수 있다. 이 태그 번지 영역에서 수행할 명령은 최대 16개까지 입력이 가능하다. 그리고 태그를 인식하는 센서는 AGV 주행 방향의 오른쪽 아래에 배치되어 있다. 따라서 태그용 반사 테이프의 위치는 주행 방향의 오른쪽에 설치한다.
- ③ 명령어 : AGV가 해당 번지에 가서 실행할 기능을 말하며, 세부 내용은 아래 명령어 리스트를 참조한다.

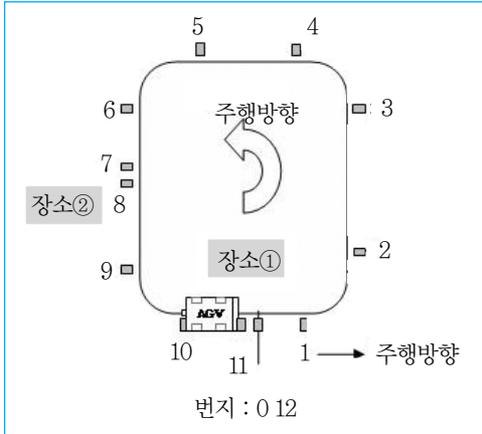
2) 명령어 리스트

분류	세부 기능
속도(Speed) S:10~100	<p>AGV의 주행 속도를 제어하기 위한 것으로 10~100 사이의 속도를 설정할 수 있으며, 100은 약 100cm/초를 의미한다. 참고로 적절한 주행 속도는 아래와 같다.</p> <ul style="list-style-type: none"> - 정지 속도(Low) : AGV를 정확한 위치에 정지하기 위해서는 가능한 속도를 낮추어 주행하는 것이 바람직하며 10 정도가 적당하다. 정지 정밀도가 중요하지 않은 경우 속도를 높게 사용하여 주행 시간을 단축시킬 수 있다. - 회전 속도(Middle) : AGV가 회전 구간을 주행할 때 필요한 최고 속도를 의미하며, 35cm/초의 속도로 회전 가능하다. 이 속도보다 빠른 상태에서 회전하게 되면 AGV가 탈선할 확률이 높아진다. 회전 속도를 높이려면 회전 반경을 크게 하면 된다. - 주행 속도(High) : AGV가 직선 구간에서 주행할 속도를 설정한다.



<p>I/O 제어 (IO Cont.) : O:01~23</p>	<ul style="list-style-type: none"> - 주행음 On/Off (01, 02) : AGV가 주행하는 것을 알리기 위한 소리를 켜줌/꺼줌. - 주행등 On/Off(03, 04) : AGV가 주행하는 것을 알리기 위한 램프를 켜줌/꺼줌. - 1차 장애물 감지 센서 On/Off(05, 06) : 장애물 센서의 1차 감지 레벨의 동작을 선택하기 위한 기능 - 2차 장애물 감지 센서 On/Off(07, 08) : 장애물 센서의 2차 감지 레벨의 동작을 선택하기 위한 기능 - 장애물 센서 동작 영역 선택(10~14) : 장애물 센서의 동작 영역을 선택하기 위한 기능, Option - 주행음 선택 기능(16~21) : 5종의 소리를 선택하기 위한 기능 - 에러음 On/Off(22, 23) : 에러가 발생한 경우 에러음을 켜줌/꺼줌
<p>Wait 기능 : W:1~10</p>	<p>최대 10초까지 1초 단위로 AGV가 일시 정지하였다가 다시 출발하는 기능</p>
<p>자동 충전 C:</p>	<p>충전 기능이 있는 AGV의 경우 자동 충전을 하기 위한 기능 (향후 기능, 제외)</p>
<p>컨베이어 기능: L:01~04</p>	<p>AGV에 컨베이어가 탑재된 경우 AGV가 설비 앞에 도착하여 할 일을 정의하기 위한 명령어(향후 기능, 제외)</p> <ul style="list-style-type: none"> - 왼쪽에서 적재>Loading) : 01 - 왼쪽에서 이재(Unloading) : 02 - 오른쪽에서 적재>Loading) : 03 - 오른쪽에서 이재(Unloading) : 04
<p>주행 방향 : D:01~04</p>	<p>양방향 이동형 AGV의 경우만 후진 기능이 구현되며, AGV의 주행 방향을 선택할 수 있다.</p> <ul style="list-style-type: none"> - 전진 : 01 - 후진 : 02(향후 기능, 제외) - 좌회전 : 03 - 우회전 : 04
<p>동작 관련: E:01~02</p>	<p>AGV가 정지한 후 다음 명령을 수행하는 방법을 알려주기 위한 기능</p> <ul style="list-style-type: none"> - 일시 정지(Pause, 01) : 외부의 시작 버튼이나 센서에 의한 시작 명령을 주면 다음 명령어 라인부터 프로그램이 실행된다. - 정지(Stop(02)) : 현재 선택된 프로그램을 완전히 끝내고 다음 명령을 기다린다.
<p>프로그램 관련: P:01~16</p>	<ul style="list-style-type: none"> - 프로그램 변경(03) : 지정된 프로그램 번호를 새로 시작하게 된다.

〈표 Ⅵ-1〉 명령어 리스트



〈그림Ⅶ-31〉 AGV 태그 배치 예1

3) 예제 프로그램 1

아래 프로그램은 장소 ①(0번지 또는 11번지)에서 장소 ②(4번지)로 물건을 이동시키기 위한 예제 프로그램이다. 프로그램 번호는 1번으로 한다. AGV를 빠르고 안정적으로 이동시키기 위한 속도 및 I/O 등을 조정하면서 LCD 모듈에 표기된 AGV의 최고 속도 및 주행 시간 정보를 활용하면 AGV의 최고 성능을 발휘할 수 있는 설정치를 얻을 수 있다.

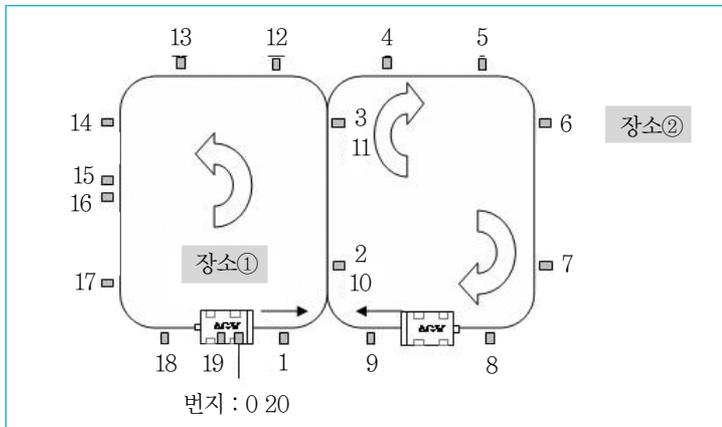
프로그램 번호	번지	명령어	기능설명
1	1	S : 35	회전하기 위한 감속
	2	S : 100	직선 구간 주행을 위한 속도 설정
	3	S : 15	장소②에서 정지하기 위한 급감속 설정
	4	E : 01	장소②에서 일시 정지, Start 입력이 들어오기 전까지 대기 상태에 들어간다.
	5	S : 35	회전하기 위한 감속
		0 : 06	장애물 센서를 사용하지 않는 모드
	6	S : 100	직선 구간 주행을 위한 속도 설정
		0 : 05	장애물 센서 사용하는 모드로 설정
	7	S : 35	회전하기 위한 감속
	8	S : 100	직선 구간 주행을 위한 속도 설정
	9	S : 35	회전하기 위한 감속
	10	S : 100	직선 구간 주행을 위한 속도 설정
	11	S : 15	장소①에 정지하기 위한 급감속 설정
12또는 0	E : 01	장소①에서 일시 정지, Start 입력이 들어오기 전까지 대기 상태에 들어간다.	
	P : 01	새로 시작할 프로그램 번호1	

〈표Ⅶ-2〉 예제 프로그램1



4) 예제 프로그램 2

아래 프로그램은 장소 ①(0번지 또는 20번지)에서 장소 ②(6번지)로 물건을 이동시키고 장소 ①로 돌아오기 위한 예제 프로그램이다. AGV가 주행하는 경로는 아래 표의 번지 순서대로 수행하게 된다. 프로그램 번호는 2번으로 한다.



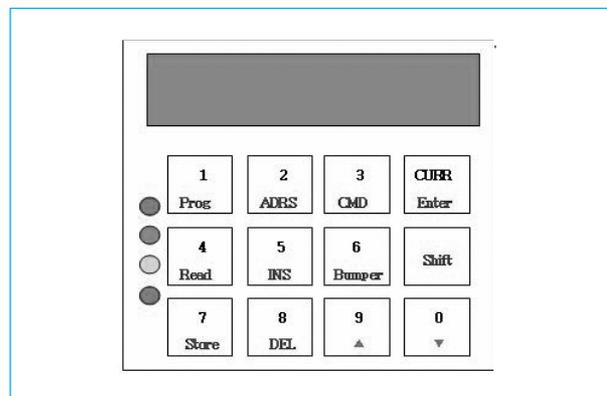
〈그림 VI-32〉 AGV 태그 배치 예2

프로그램 번호	번지	명령어	기능설명
2	1	S : 35	회전하기 위한 감속
	2	S : 100	직선 구간 주행을 위한 속도 설정
	3	S : 15	회전하기 위한 감속
		D : 04	분기점에서 회전 방향을 우측으로 설정
	4	S : 100	직선 구간 주행을 위한 속도 설정
	5	S : 35	회전하기 위한 감속
		O : 06	장애물 센서를 사용하지 않는 모드
	6	E : 01	장소 ②에서 일시 정지, Start 입력이 들어오기 전까지 대기 상태에 들어간다.
		O : 05	장애물 센서 사용하는 모드로 설정
	7	S : 35	회전하기 위한 감속
8	S : 100	직선 구간 주행을 위한 속도 설정	
9	S : 35	회전하기 위한 감속	
10	S : 100	직선 구간 주행을 위한 속도 설정	

11	S : 35	회전하기 위한 감속
	D : 03	분기점에서 회전 방향을 좌측으로 설정
12	S : 100	직선 구간 주행을 위한 속도 설정
13	S : 35	회전하기 위한 감속
14	S : 100	직선 구간 주행을 위한 속도 설정
15	O : 01	주행음을 끈다
	S : 15	16번지에서 잠시 정지하기 위해 급감속 설정
16	O : 02	주행음을 끈다
	W : 05	5초간 정지 후 다시 움직인다.
	S : 100	직선 구간 주행을 위한 속도 설정
17	S : 35	회전하기 위한 감속
18	S : 100	직선 구간 주행을 위한 속도 설정
19	S : 15	장소 ①에 정지하기 위한 급감속 설정
20또는 0	E : 01	장소 ①에서 일시 정지, Start 입력이 들어오기 전까지 대기 상태에 들어간다.
	P : 01	새로 시작할 프로그램 번호1

〈표 VI-3〉 예제 프로그램2

2 LCD 모듈을 이용한 프로그램



〈그림 VI-33〉 LCD 모듈 키 배치

〈그림 VI-33〉의 그림 컬러로 사용하기 위해서는 Shift 키를 계속 누른 상태에

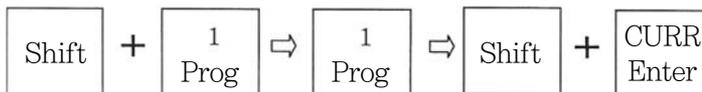


서 Key를 누르면 된다. 예를 들어 prog 버튼을 사용하려면 Shift Key와 1/prog Key를 함께 누르면 된다. 이렇게 Shift key와 함께 버튼을 눌러야 하는 경우는 Shift+prog라고 표현한다.

(1) 예제 프로그램 1 입력 하기

1) 프로그램 번호 입력

아래 그림과 같은 순서로 프로그램(Prog) 키와 프로그램 번호 1을 누른 후 Enter 키를 입력하면 된다.



2) 번지 입력

아래 그림과 같은 순서로 번지(ADRS) 키와 숫자 1을 누른다. 명령어 입력이 끝나기 전까지 번지 키를 다시 누르지 않으면 해당 번지에 계속해서 명령을 입력할 수 있다. 다른 어드레스에 해당하는 명령어도 동일한 방법으로 반복하면 된다.



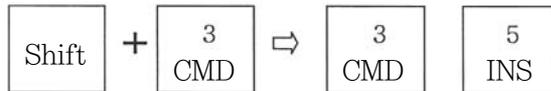
3) 명령어 입력

명령(CMD) 키를 눌러서 원하는 명령어가 LCD에 표시될 때까지 반복한다. 표시되는 명령어 순서는 속도(S:), I/O 제어(O:), Wait(W:), 주행 방향(D:), 동작 관련(E:), 프로그램 관련(P:) 명령어 순으로 반복된다. 명령어가 선택된 후 해당하는 숫자 키를 입력하면 명령어 입력이 종료된다. 계속해서 새로운 명령

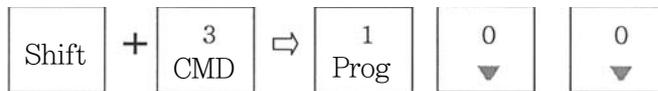
어를 입력하려면 위 순서를 다시 반복하면 된다.

해당 어드레스에서 명령어 입력을 종료하려면 Enter 키를 입력하면 된다. Enter 키를 입력하면 어드레스 카운터는 자동으로 1씩 증가하여 별도로 입력하지 않아도 다음 번지에서 실행할 명령만 입력하면 된다.

예제 프로그램의 S:35를 입력하는 방법은 CMD 키를 누르면 최초로 표시되는 S: 명령어가 표시되며, 속도를 지정하기 위한 35를 입력하면 첫번째 명령인 S:35가 입력되었다.



2번지에서의 S:100 명령은 아래와 같은 순서로 반복하면 된다.



4) 프로그램 입력 종료

명령어를 입력하고 Store키를 누르면 자동으로 프로그램 편집 모드가 종료된다.

5) 기존 프로그램 편집하기

Read 키를 입력한 후 편집할 프로그램 번호를 입력하면 된다. 어드레스 및 명령어 추가는 기존 방법과 동일하게 하며 추가나 삭제는 INS 및 DEL 키를 이용한다.

6) 선택된 프로그램 실행 방법

Prog 키와 숫자 키를 이용하여 프로그램을 선택하고, 이동 로봇의 조작 판넬



에 있는 Start 버튼을 누르면 실행된다.

7) 현재 위치(어드레스) 입력 방법

이동 로봇이 에러가 발생하여 정지하였거나, 전원을 새로 켜었을 경우 이동 로봇의 현재 위치(어드레스)를 알려주는 방법으로 아래와 같은 순서(11번인 경우)로 입력하면 된다.

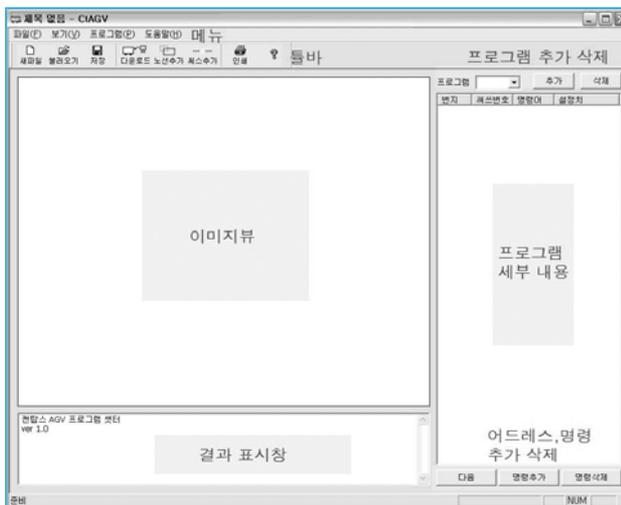


3 PC를 이용한 프로그램

1) 프로그램 설치

제공된 CD의 프로그램을 컴퓨터에 복사하고 Setup.exe를 실행한 후 Next만 클릭하면 설치 완료된다.

2) 프로그램 편집 화면



〈그림 VI-34〉 프로그램 편집 화면

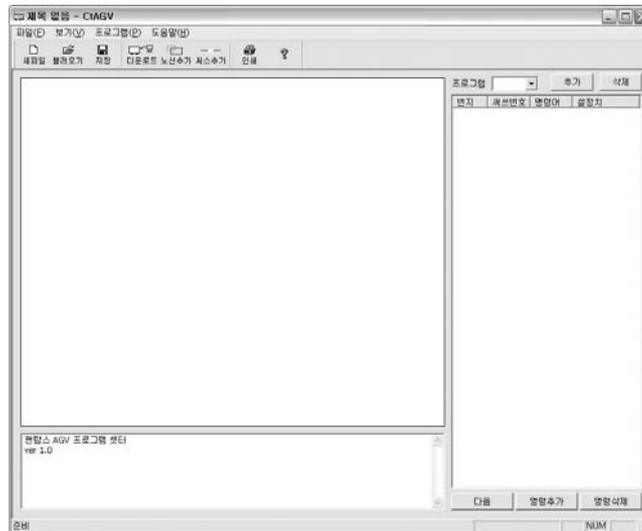


프로그램은 <그림 VI-34>처럼 메뉴, 툴바, 이미지 뷰창, 결과 표시창, 프로그램 추가 삭제, 프로그램 세부 내용창, 어드레스/명령 추가 삭제 부분으로 나누어진다. 툴바와 버튼의 기능은 메뉴에서도 동작이 가능하다.

- ① 이미지 뷰창은 노선(선로)과 테이프의 구성을 그래픽으로 확인하고 설계하는 부분이다.
- ② 결과 표시창은 지금 실행한 기능의 결과와 프로그램의 상태를 표시한다.
- ③ 프로그램 세부 내용창은 프로그램의 내용을 표시하며 추가/수정/삭제가 가능하다.

3) 프로그램 따라하기

- ① 프로그램 실행.



<첫화면>

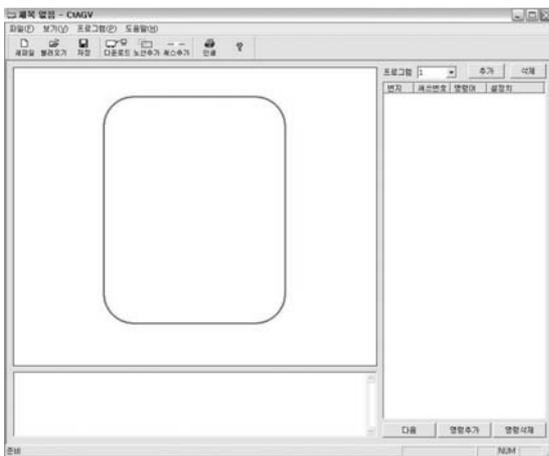


② 툴바의 새파일을 클릭.

AGV의 방향을 물어보면 시계방향인지 반시계 방향인지 결정해 준다.



〈방향을 물어보는 다이얼로그〉



〈새파일 실행 후 화면〉

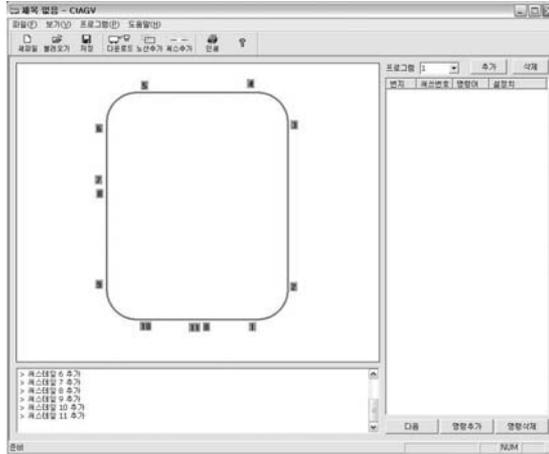
③ 씨스 태그 추가

툴바의 씨스 태그 추가를 클릭하고 이미지 뷰에 커서를 가져가면 아래와 같은 모양으로 변한다.



〈태그 추가시 커서의 모양〉

이때 원하는 위치에 클릭하면 태그가 추가 된다.



〈써스를 모두 추가한 후의 이미지〉

④ 프로그램 하기

어드레스/명령 추가 제거 부분에서 다음을 누르면 번지를 추가할 수 있다.

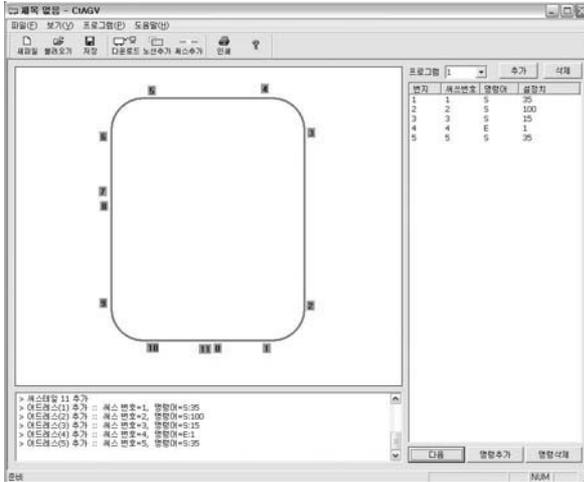
다음과 같은 창이 뜨면 해당 어드레스의 써스 번호, 명령, 설정치를 지정해 주고 확인을 누른다.



〈어드레스 추가 다이얼로그〉



〈명령어 콤보박스의 내용〉



〈어드레스 추가 후의 이미지〉

한 어드레스에 여러 가지 명령을 사용하고 싶을 때 명령 추가를 이용한다.
어드레스 추가와 비슷하며 명령어와 설정치만 지정해 주면 된다.



〈명령 추가 다이얼로그〉



〈프로그램을 끝낸 후의 이미지〉



⑤ 마무리

툴바의 저장을 클릭해서 지금까지의 내용을 저장해 둔다.

AGV에 적용하기 위해서는 PC와 AGV를 연결하고 툐바의 다운로드를 클릭 해주면 자동으로 다운로드 되어 프로그램이 적용된다.

5 이동 로봇 구동 시험 및 예제 실습

AGV 부품이 조립되면 아래와 같은 단계로 시험을 진행한다.

1 조립 상태 점검

- ① 전원선(배터리) 연결 상태를 확인한다.
- ② 센서의 고정 상태 및 케이블과 기구적인 간섭이 없는지 확인한다.
- ③ 차단기는 Off시킨 상태에서 테스트를 저항 측정 모드로 하고 제어기의 24V 전원단의 저항을 측정하여 수MΩ 이상이 나오면 정상이다.
- ④ 차단기 입력 라인에 배터리 전압(+24.5~+26.5V) 및 극성을 확인한다.
- ⑤ 전원 및 전장 결선도를 참조하여 케이블의 삽입 상태 및 오배선을 확인한다.

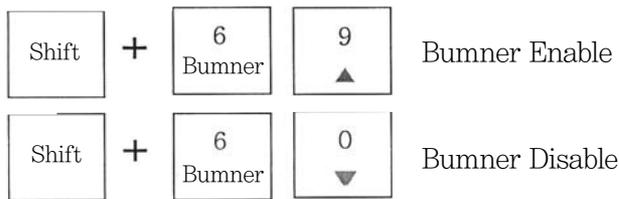
2 수동 모드에서의 동작 확인

- ① 모드를 수동으로 하고 비상정지 버튼을 On 시킨 상태에서 주 차단기를 On 시켜 전원을 공급한다.
- ② ICon 보드에 입력되는 +24V 전원을 확인한 후 육안으로 회로 내의 부품 파손 및 다른 이상이 없는지 확인한다.
- ③ LCD 모듈에 에러 코드(비상 정지, 모터 드라이버 전원 Off)가 표시되면서,



타워 램프의 부저와 스피커에서 에러음이 나오는 것을 확인하면 정상이다.

- ④ 비상 정지 버튼을 해제하고 이동 로봇 상부에 있는 Stop 버튼을 눌러서 에러를 해제시킨다.
- ⑤ 범퍼를 인에이블(Enable) 상태로 만들고 범퍼에 충격을 가했을 때 동작하는지 확인한다.



- ⑥ 센서 동작 상태 파악(가이드, 태그 센서) : LCD 모듈의 Up(▲), Down(▼) 버튼을 누르면 LCD 모듈에 이동 로봇의 다양한 정보가 표시되는데, 그 중 센서 정보가 다음의 그림과 같은 상태로 표시될 수 있도록 조작한다.

LCD 표시 내용	함께 사용하는 버튼
Adrs Counter:0 0001111111100009	없음

- Adrs Counter:0 => 씨스 테이프를 태그 센서를 동작시키면서 이 수가 동작시킨 횟수와 일치하는지 확인한다.
- 000111111110000 => 차체의 전방 하단에 붙어 있는 가이드 센서의 동작 상태를 표시하는 것으로 주기적으로 선로(씨스 테이프)의 오염 및 파손을 확인하는 용도로 사용하면 좋다. 0은 Off 상태, 1은 On 상태를 의미한다. 이 표시된 내용과 씨스 테이프를 가이드 센서를 감지시켰을 때의 내용과 일치하는 것을 확인한다.
- 9 => 이동 로봇이 주행하면서 선로가 끊어진 것을 모두 카운트(선로 이상



카운터)하여 표시하는 기능으로 주기적으로 이 값을 확인하여 0이 되도록 선로를 관리한다면 최적의 상태에서 로봇을 사용할 수 있다. ASCII 코드를 이용하여 최대 80개까지 표현할 수 있다.

- ⑦ 조향의 좌우 리미트 동작 상태 점검 : 수동으로 조향 바퀴를 좌로 끝까지 움직여 좌측 리미트 센서에 동작 LED가 켜지는 것을 확인한다. 다시 우로 끝까지 움직여 우측 리미트 센서가 감지되는 것을 확인한다. 센서와 감지용 볼트와의 거리는 2~3mm가 적당하며, 감지되지 않거나 항상 동작하는등 인식에 문제가 있을 경우 리미트 센서의 고정 볼트 및 너트를 조정하여 감지용 볼트와의 거리를 바꾸면서 설정한다.
- ⑧ 주행 모터 브레이크 점검 : 수동 모드에서 전원을 넣고 에러가 없는 상황에서 자동으로 모터 브레이크가 풀려야 하기 때문에 손으로 밀면 이동 로봇이 움직이게 된다. 움직이지 않은 경우엔 모터 브레이크 선의 연결 상태를 확인한다.
- ⑨ TP를 이용해서 전/후/좌/우로 이동 : TP를 연결해서 버튼을 이용하여 로봇을 움직여 본다. 동작하지 않으면 TP 케이블, 모터 케이블 등을 확인한다.
- ⑩ 배터리 동작 상태 파악 : LCD 모듈의 Up(▲), Down(▼)버튼을 누르면 LCD 모듈에 배터리 전압, 온도, 충전율, 사용할 비율(배터리 저전압 체크 레벨), 총 배터리 사용 시간이 표시된다. Up/Down 버튼을 이용하여 배터리 저전압 체크 레벨을 변경할 수 있으며, 80~50사이의 범위에서 조정이 가능하다. 이 기능을 이용하여 배터리 전압 및 온도에 이상이 없는 것을 확인한다.

LCD 표시 내용	함께 사용하는 버튼
V:24.67 Temp:24 C:100, 50 Hr:67.8	▲(저전압 검출 레벨 증가) ▼(저전압 레벨 레벨 감소)

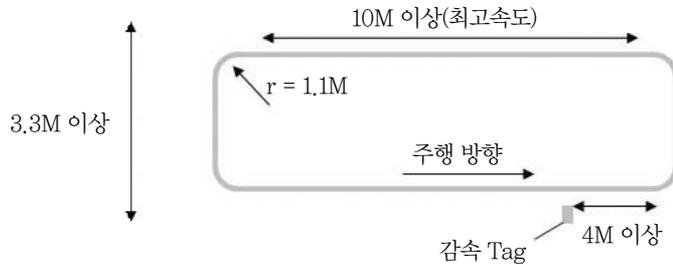


- V:24.67 => 현재 배터리 전압을 표시
- Temp:24 => 현재 배터리 온도를 표시하며, 60도 이상이면 에러로 처리한다.
- C:100, 50 => 충전율과 저전압 검출 레벨을 표시한다. 배터리 전압은 AGV가 주행할 때와 정지 상태에서 차이가 있을 수 있다.
- Hr:67.8 => 전원을 넣고 사용된 배터리의 총 시간을 표시하며, 최대 시간은 99.9 시간이다.(표시 단위는 시간임.)

3 주행 선로 만들기

주행 선로는 시험 공간 및 부품 재고 상황에 따라 임의의 크기로 만드는 것이 가능하나, 이동 로봇의 주행 성능은 고려하여 만들면 최고의 성능이 발휘될 수 있다.

- ① 감속 거리 : 물건이 없는 이동 로봇 자체의 경우 최고 속도(1M/초)로 주행할 경우 약 4M의 감속 거리가 필요하다.
- ② 가속 거리 : 최고 속도로 가속하는 데 필요한 거리는 약 7M이다.
- ③ 회전 반경 : 회전 반경은 이동 로봇의 속도와 관련이 있다. 속도가 빠르면 회전 반경이 커지고, 속도가 낮으면 회전 반경은 작아질 수 있다. 속도 40% 수준으로 회전을 한다면 최소 회전 반경은 1.1M 정도이다. 공간이 협소한 경우 회전 속도를 낮추면 반경(0.9M까지)을 줄일 수 있다.
- ④ 선로 만들기 : 써스 테이프의 접착력이 강해서 바닥에 붙이면 재활용하기가 어렵다. 따라서 시험 가동할 때에는 임시로 써스 테이프를 바닥에 놓고 일반 셀로판 테이프로 써스 테이프 위에서 바닥과 고정하는 방법이 좋다.



- ⑤ 어드레스 카운터 : 이동 로봇이 회전할 때에는 속도에 따른 감속 거리를 계산해서 회전 커브를 만나기 전 미리 감속이 필요하며, 정지하는 기능도 동일한 방법으로 충분한 거리 여유를 두고 감속할 수 있도록 태그를 설치한다. 가속하기 위한 태그는 선로가 직선 구간으로 가속할 수 있는 영역에서 설치한다.

4 자동 모드에서의 동작 확인

- ① 프로그램 : PC 프로그램이나 LCD 모듈에서 만들어진 선로에 대한 프로그램을 작성한다. 해당 프로그램 및 현재 위치(0인 위치에서는 입력할 필요 없음)를 입력하고 Start 버튼을 누른다.
- ② 동작 상태 파악 : LCD 모듈에 설정된 속도, 현재 속도, 최대 속도, 주행 누적 시간, 거리 합계에 대한 내용이 표시된다. 표시된 내용과 실제 주행 상태를 파악하여 이상 상태를 파악한다.

LCD 표시 내용	함께 사용하는 버튼
S:100 1.00 1.01 T:78.9 Mi:567.8	없음

- S:100 => 현재 설정된 속도의 백분율로 표시되며, 100으로 설정된 경우 약



1M/초의 속도로 이동 로봇이 주행하게 된다.

- 1.00 => 이동 로봇이 주행하는 현재 속도를 의미하며, 단위는 M/초이다.
- 1.01 => 이동 로봇의 주행 최대 속도를 의미하며, 단위는 M/초이다.
- T:78.9 => 이동 로봇이 정지 시간을 제외하고 주행한 총 시간을 의미하며, 단위는 분이므로 LCD의 표시 한계 때문에 99.9분이 초과하면 ASCII 코드를 이용하여 시간을 표시한다.

③ 에러 표시 내용 확인 : LCD 모듈의 Up(▲), Down(▼) 버튼을 누르면 LCD 모듈에 현재 에러 상태가 표시된다.

LCD 표시 내용	함께 사용하는 버튼
E.CNT=004 Cd=63 ESTOP On	▲(이전 에러) ▼(다음 에러)

- E.CNT=004 => 현재 발생된 에러의 총 수, 최대 30개까지 카운트 할 수 있다.
- Cd=63 => 에러 코드가 발생하였음을 보여주며, 상세 내용은 아래 줄에 표시된다.
- ESTOP On => 위 에러 코드에 대한 구체적인 내용이며, 각 에러에 대한 원인을 파악하여 조치한 후 다시 동작시킨다.

④ 태그 위치 및 선로 수정 : 이동 로봇의 주행 상태를 보고 직선 구간에서의 흔들림이 많은 경우 선로를 확인하여 직선이 되도록 수정하며, 회전 구간에서 탈선이 있는 경우 속도를 낮추거나 회전 반경을 키운다.



단원 학습 정리



- 1** 이동 로봇의 움직임은 전/후로 움직이기 위한 주행 부분과 좌/우로 회전하기 위한 조향 부분으로 구분한다.
- 2** IR 센서, 자기 테이프, 자기 유도 방식은 바닥에 AGV가 주행할 선로를 미리 설치하고 AGV에 장착된 센서를 이용하여 길을 인식하여 주행한다.
- 3** 레이저(Laser)를 이용한 방법은 AGV 몸체에 레이저 스캐너를 설치하고, AGV가 주행할 통로에 레이저 빛을 반사시키는 반사판을 설치한 후 레이저 스캐너를 360° 회전시키면서 반사판에서 반사된 빛을 이용하여 자기 위치를 인식하는 방법이다.
- 4** 주행 속도에 영향을 미치는 부품은 이동 로봇이 주행하면서 정지 위치에서 멈추게 하거나 부착된 센서의 동작에 의해 감속시킬 수 있다.
- 5** 감속 단계는 2단계로 구분하였으며, 최고 속도의 40% 수준과 20%로 감속하게 된다. 40%는 일반적인 회전 모드에서 사용하는 속도이고, 20%는 이동 로봇이 멈추기 전에 정밀한 정지를 위해 속도를 충분히 낮춰주기 위해 사용한다.



- ❖6 주행 선로는 시험 공간 및 부품 재고 상황에 따라 임의의 크기로 만드는 것이 가능하나, 이동 로봇의 주행 성능은 고려하여 만들면 최고의 성능이 발휘될 수 있다.
- ❖7 감속 거리는 물건이 없는 이동 로봇 자체의 경우 최고 속도(1M/초)로 주행할 경우 약 4M의 감속 거리가 필요하다.
- ❖8 가속 거리는 최고 속도로 가속하는 데 필요한 거리는 약 7M이다.
- ❖9 회전 반경은 이동 로봇의 속도와 관련이 있다. 속도가 빠르면 회전 반경이 커지고, 속도가 낮으면 회전 반경이 작아질 수 있다.
- ❖10 선로 만들기는 써스 테이프의 접착력이 강해서 바닥에 붙이는 경우 재 활용하기가 어렵다. 따라서 시험 가동할 때에는 간이로 써스 테이프를 바닥에 놓고 일반 셀로판 테이프로 써스 테이프 위에서 바닥과 고정하는 방법이 좋다.





창작 로봇 제작

로봇은 산업 현장뿐만 아니라 일상 생활의 다양한 분야에서 응용될 수 있다. 집안을 청소하는 것부터 시작해서, 하수도의 배관을 점검하는 일까지 인간의 상상력으로 무한한 형태의 로봇을 만들어서 활용할 수 있다.

본 단원에서는 창작로봇으로서 뱀 로봇의 하드웨어와 소프트웨어의 설계와 조립 방법을 학습한다.



학습목표

1. 뱀 로봇의 하드웨어 설계 능력을 배양한다.
2. 뱀 로봇의 소프트웨어 설계 능력을 배양한다.
3. 뱀 로봇의 조립 능력을 배양한다.

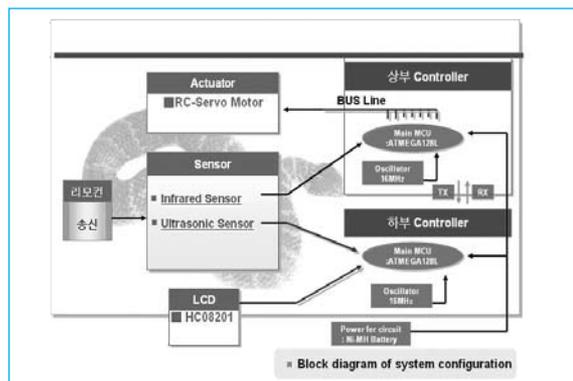


1 뱀 로봇 하드웨어 상세 설계

1) 시스템 구조

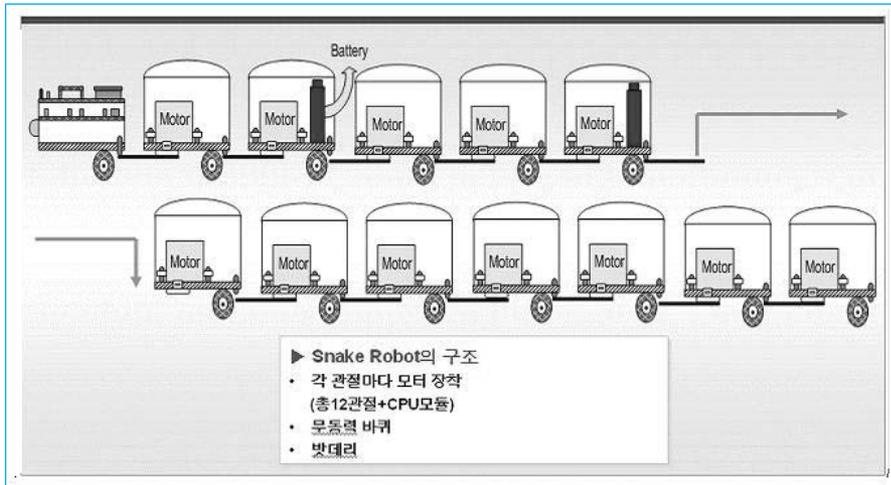
뱀 로봇의 시스템 구조는 2개의 컨트롤러(Controller) 보드가 USART(Universal Synchronous Asynchronous Receiver Transceiver) 통신을 통해서 데이터를 주고받는 것을 기반으로 한다. 상부 컨트롤러 보드에서는 12개의 RC-서보(Servo)모터와 리모컨 센서를 제어하고, 하부 컨트롤러 보드는 장애물 감지를 위한 초음파 센서와 초음파 센서의 거리 값을 표시하는 LCD를 제어한다.

12개의 RC-서보모터는 6V 니켈-수소(Ni-MH) 충전지를 직접 사용하고, 2개의 컨트롤러 보드는 9V 충전기와 레귤레이터(Regulator)를 이용하여 5V의 전압을 사용한다.



〈그림VII-1〉 뱀 로봇의 시스템 구조

2) 뱀 로봇의 구조

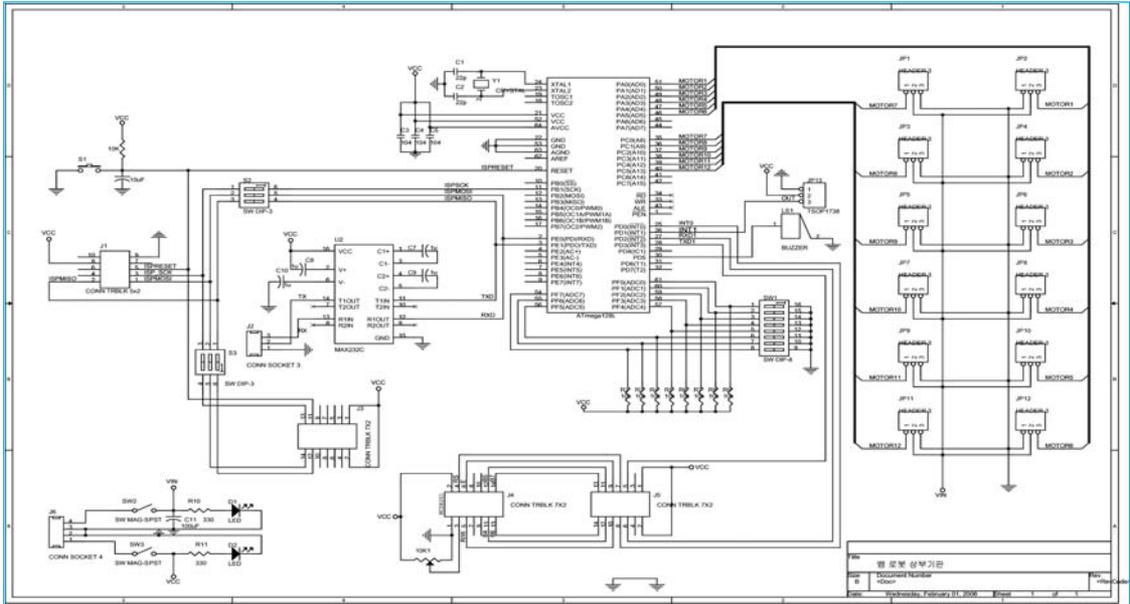


〈그림Ⅶ-2〉 뱀 로봇의 구조

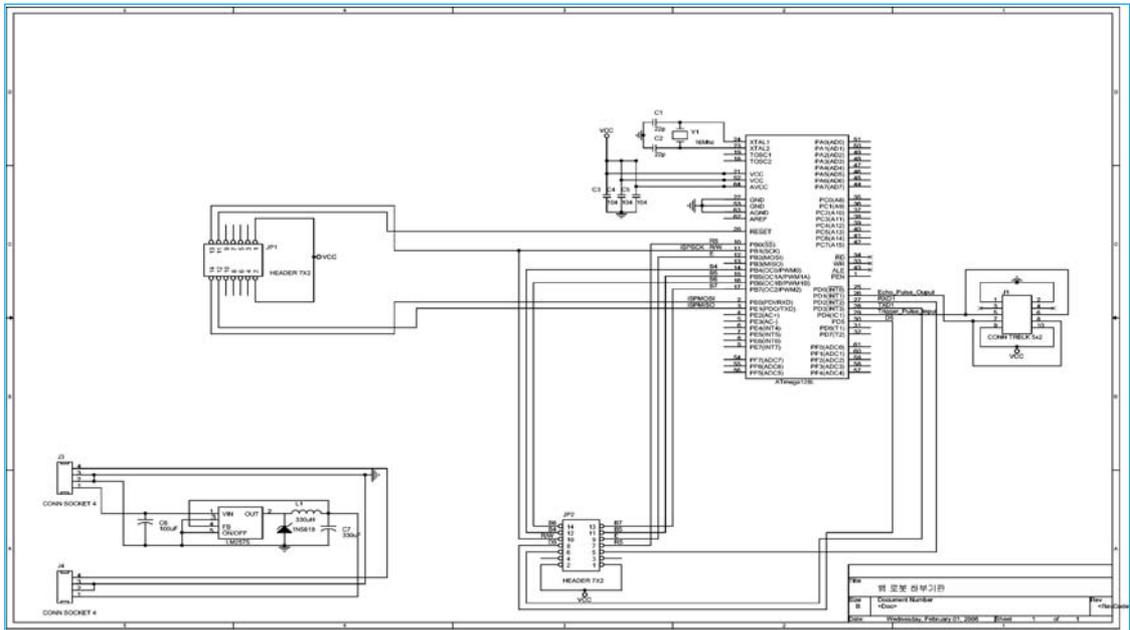


〈그림Ⅶ-3〉 뱀 로봇 실물도

3) 뱀 로봇 회로도



〈그림Ⅶ-4〉 뱀 로봇 상부 보드 회로도



〈그림Ⅶ-5〉 뱀 로봇 하부 보드 회로도



2 로봇 소프트웨어 상세 설계

1 에뮬레이터 프로그램 설치

로봇을 제어하기 위한 에뮬레이터 프로그램을 설치해 보자.



- ① setup.exe 파일을 실행시킨다.



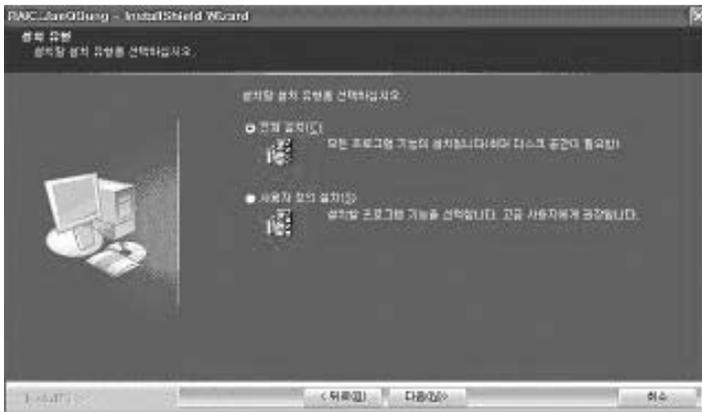
- ② 다음을 클릭(Click)한다.



- ③ 위 그림과 같이 사용권 계약의 조항에 동의한다(A) 를 지정 후 다음을 클릭한다.



④ 이름, 회사(학교)를 작성한 후 다음을 클릭한다.



⑤ 기존의 작성된 상태로 다음을 클릭해도 된다.



⑥ 이 부분에서는 설정 없이 전체 설치 그대로 하여 다음을 클릭한다.

⑦ 마지막으로 설치 버튼을 클릭하여 설치한다.

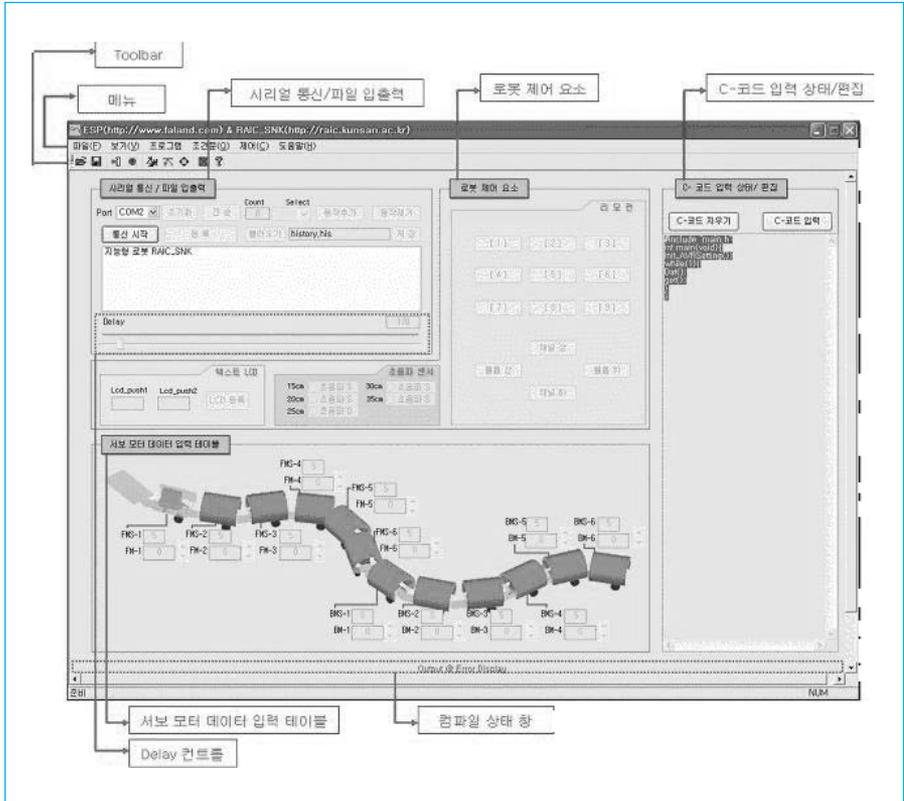


⑧ 위 그림과 같이 프로그램 설치 목록을 확인할 수 있다.



⑨ 완료 버튼을 클릭하여 프로그램을 종료한다.

2 에뮬레이터의 통합 환경 설명



〈그림VII-6〉 에뮬레이터의 통합 환경

에뮬레이터의 통합 환경의 구성은 〈그림VII-6〉과 같으며, 각 구성 요소에 대한 기능은 다음과 같다.

(1) 메뉴

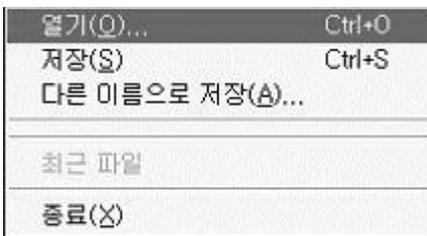


〈그림VII-7〉 메뉴



메뉴는 에뮬레이터의 기능을 사용할 수 있으며 상태 표시줄 표시, 컴파일 다운로드, 조건문, 제어, 에뮬레이터 정보 보기의 과정을 수행할 수 있다. 각 메뉴는 단축키 사용이 가능하다.

1) 파일

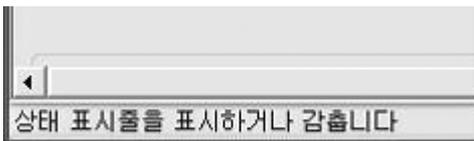


- ① 열기 : 이전에 저장했던 파일을 불러온다.
- ② 저장 : 현재 작업 중인 C코드(code)를 저장한다.
- ③ 다른 이름으로 저장 : 다른 이름으로 저장하고자 할 때 사용한다.
- ④ 종료 : 에뮬레이터를 종료시킨다.

2) 보기

상태 표시줄(D)

- ① 상태 표시줄 : 에뮬레이터 화면 하단에 현재의 작업 상태를 나타내는 부분을 활성화시킨다.



3) 프로그램

컴파일다운로드(F2)

- ① 컴파일 다운로드 : 사용자가 완성한 C 코드를 컴파일하여 로봇에 다운로드 시킨다.

4) 조건문

조건에 따른 프로그램의 선택적 실행을 하기 위한 기능이다.

Switch(S) ▶	리모트 제어(F6)
if(I) ▶	초음파센서(F7)

- ① Switch : 리모트 제어를 위한 요소의 창을 활성화시킨다.
- ② if : 초음파 센서를 위한 요소의 창을 활성화시킨다.

5) 제어

모터제어(F3)
모터 테스트(F4)
LCD 제어(F5)

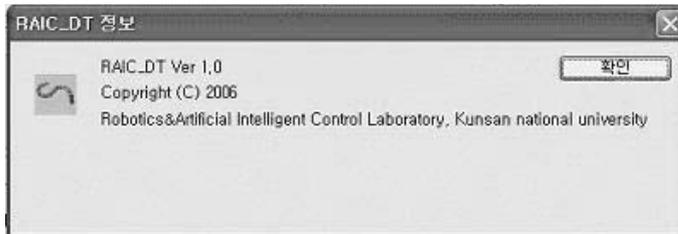
- ① 모터 제어 : 모터 제어를 위한 【서보 모터 데이터 입력 테이블】을 활성화시킨다.
- ② 모터 테스트 : 모터의 각도 변화를 테스트 하기 위한 기능이다. 시리얼 통신이 활성화 되어 있을 때에만 사용이 가능하다.



- LCD제어 : LCD제어를 위한 【TEXT LCD】을 활성화시킨다.

6) 도움말

에뮬레이터의 버전 정보를 확인할 수 있다.



(2) 툴바(Toolbar)



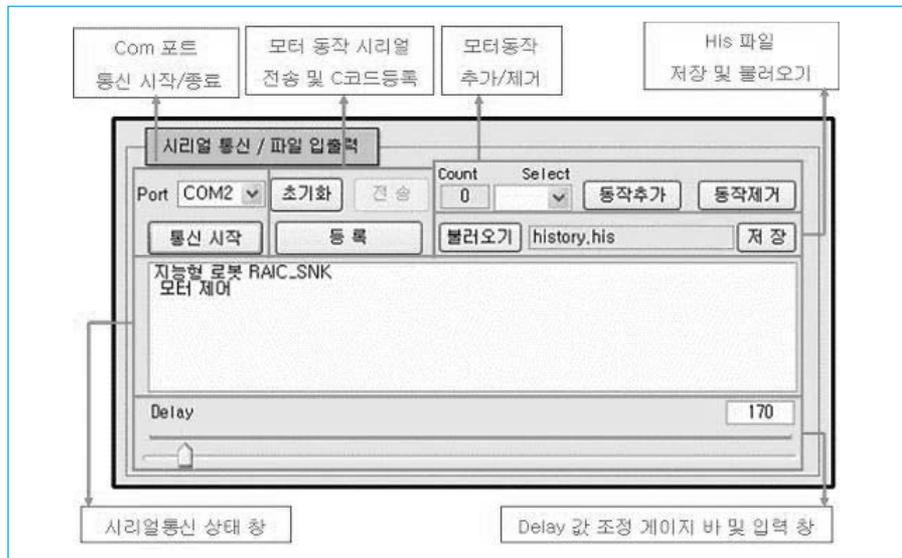
〈그림VII-8〉 툴바(Toolbar)

자주 사용하는 메뉴를 모아 놓은 부분으로 각각의 기능은 다음과 같다.

메뉴	기능	메뉴	기능
	파일 열기		파일 저장
	컴파일 및 다운로드		모터 테스트
	모터		초음파제어
	리모트제어		종료

〈표Ⅶ-1〉 툴바 기능

(3) 시리얼 통신/파일 입출력



〈그림Ⅶ-9〉 시리얼 통신/파일 입출력

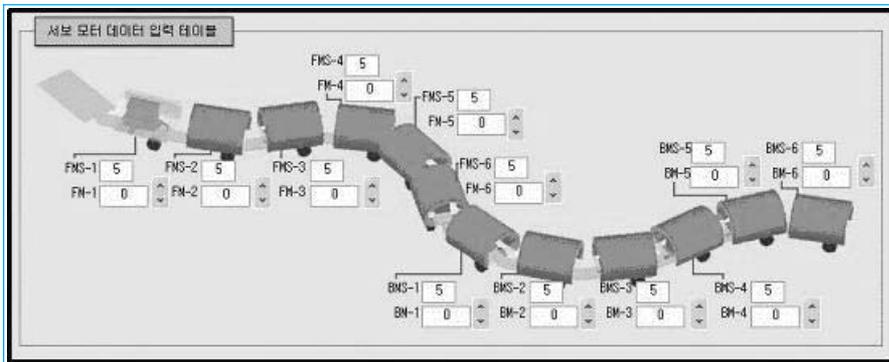
- ① Port : 시리얼 통신을 사용할 포트 선택
- ② 통신 시작 : 시리얼 통신 시작/종료
- ③ 초기화 : 서보 모터 입력 테이블을 초기화한다.
- ④ 전송 : 시리얼 통신을 통해 모터의 현재 설정 값을 전송하여 확인할 수 있다.
- ⑤ 등록 : C코드로 현재의 모터값을 등록한다.
- ⑥ 동작 추가 : 현재 모터 설정값을 동작으로 잠시 저장한다.



- ⑦ 동작 제거 : 잠시 저장된 모터 설정값을 선택적으로 제거한다.
- ⑧ Count : 잠시 저장된 모터의 설정값에 번호를 부여한다.
- ⑨ Select : 잠시 저장된 모터 설정값의 번호를 선택한다.
- ⑩ 불러오기 : 이전에 저장된 모터 설정값 파일 (확장자 his)을 불러온다.
- ⑪ 저장 : 잠시 저장된 모터 설정값을 파일(확장자 his)로 저장한다.
- ⑫ 시리얼 통신 상태창 : 시리얼 통신 송수신 및 연결/종료 상태를 보여준다.
- ⑬ Delay : 딜레이(Delay)값 설정을 위한 게이지 바(Gauge Bar) 및 입력 창

(4) 서보 모터 데이터 입력 테이블

서보 모터 데이터 입력 테이블은 【메뉴】->【제어】->【모터제어】클릭 시 활성화 되며, 툴바  아이콘은 마우스 클릭으로 사용 가능하다.



〈그림VII-10〉 서보 모터 데이터 입력 테이블

FMS-1 (Front Motor Speed 1) : 머리를 기준으로 첫 번째 모터 속도 입력창

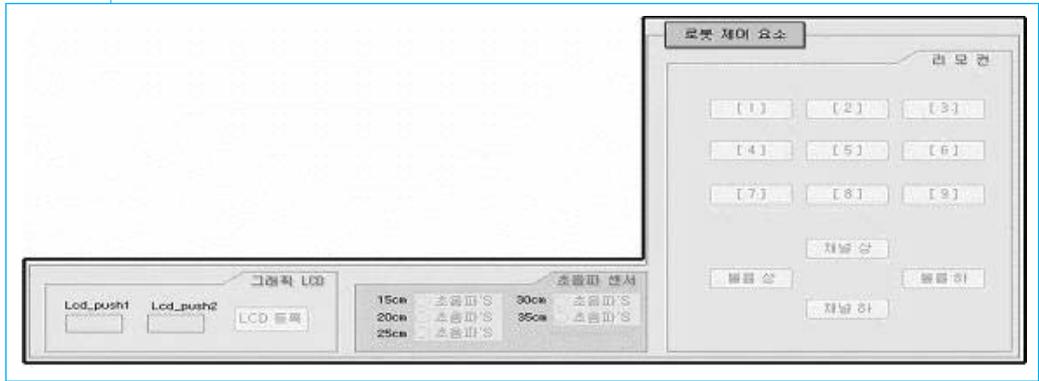
FM-1 (Front Motor 1) : 머리를 기준으로 첫 번째 모터값 입력창

BMS-1 (Back Motor Speed 1) : 앞에서 일곱 번째 모터 속도 입력창

BM (Back Motor 1) : 앞에서 일곱 번째 모터값 입력창

(5) 로봇 제어 요소

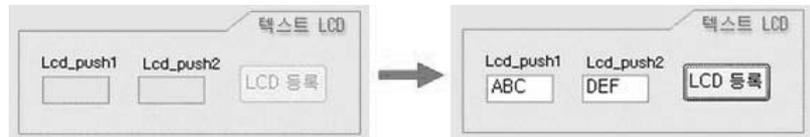
LCD의 글자 입력, 초음파 센서의 거리 선택, 리모컨의 코드 선택을 할 수 있는 부분으로 로봇의 제어에 추가적인 조건들을 제어하는 부분이다. 이는 마우스의 클릭으로 사용 가능하다.



〈그림VII-11〉 로봇 제어 요소

1) LCD

LCD 제어는 【메뉴】->【제어】->【LCD제어】클릭 시 활성화된다. 입력창은 첫 번째 줄과 두 번째 줄로 나누어져 있다. LCD 제어는 아래 그림과 같이 창이 활성화 되면 입력하고자 하는 글자를 입력 후 【LCD 등록】 버튼을 누르면 된다.



2) 초음파 센서

초음파 센서는 【메뉴】->【조건문】->【if】->【초음파 센서】 클릭 시 활성화되며,

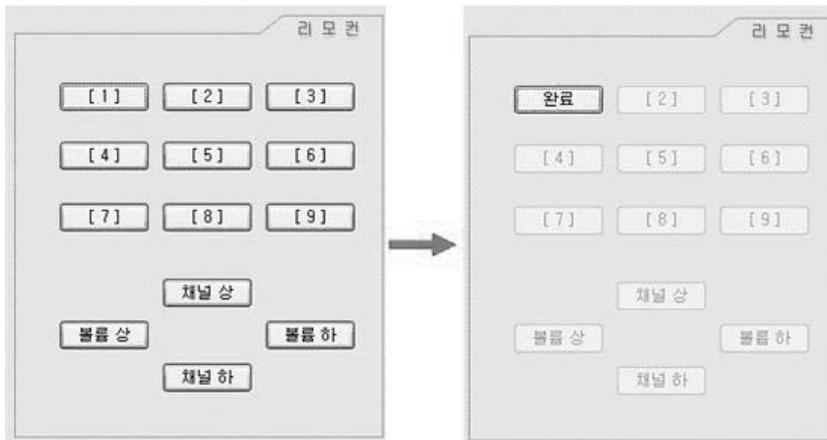


툴바에서 선택시 활성화된다. 그런 다음 정하고 싶은 거리를 정하면 된다. 단, 한번 거리를 선택하고 수정하면 안 된다.



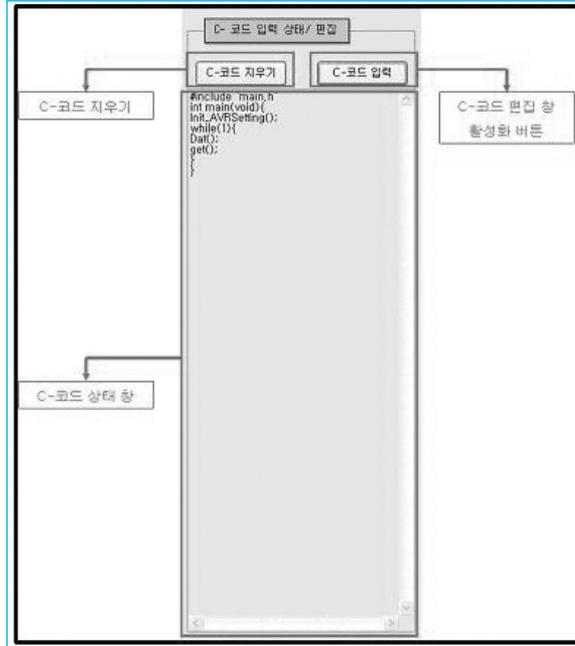
3) 리모컨

리모컨은 【메뉴】->【조건문】->【Switch】->【리모트 제어】 클릭 시 활성화되며, 툴바에서 선택 시 활성화된다. 리모컨을 이용한 제어에서 사용할 데이터 값이다. 총 13개의 리모컨 버튼을 사용할 수 있다. 해당 버튼을 클릭한 뒤 원하는 모터값 또는 데이터를 입력 후 【완료】 버튼을 클릭하면 된다.



(6) C 코드 입력 상태/편집

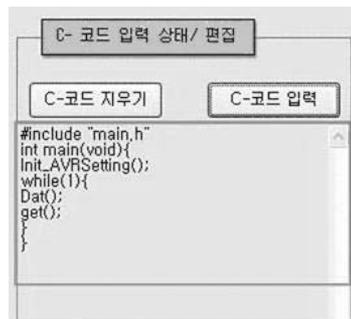
C 언어 문법을 이용하여 로봇 제어를 할 수 있으며 【C 코드 입력】 버튼을 사용해 텍스트 창을 활성화할 수 있고, 【C 코드 지우기】를 통해서 입력된 C 코드를 지울 수 있다. C 언어 문법의 조건문을 통해서 로봇의 다양한 제어가 가능하다.



〈그림VII-12〉 C 코드 입력 상태/편집

4) C 코드 지우기

클릭 시 현재 C 코드 상태 창에 나타난 C 코드를 초기 값으로 되돌린다.



5) C 코드 입력

직접 C 프로그래밍을 구현하기 위한 C 코드 편집 창을 활성화시키는 버튼이



다. C 코드 작성을 완료 후 **【확인】** 버튼을 클릭하면 된다.



6) C 코드 상태창

컴파일 할 현재의 C 코드 상태를 보여주는 창으로서 버튼으로 클릭한 입력상태를 직접 눈으로 확인할 수 있다.

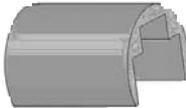
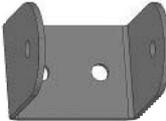
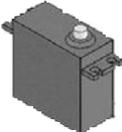
컴파일 상태 창은 **【메뉴】->【프로그램】->【컴파일 다운로드】** 클릭 시, 또는 툴바의  선택시 활성화된다. 컴파일 에러 및 다운로드 완료 상태를 확인할 수 있다.

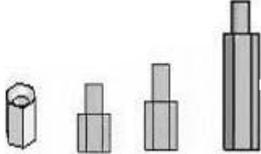
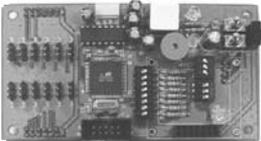
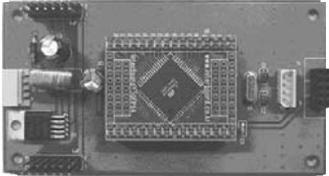


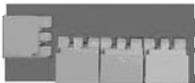
〈그림VII-13〉 C 코드 상태창

3 뱀 로봇 조립

1 구성 부품

부품명 및 개수	외형그림
머리판 (1개)	
몸체 프레임(大 : 10개)(小 : 2개)	
로봇 덮개(大: 10개)(小: 2개)	
연결판(12개)	
바퀴 프레임(13개)	
부품명 및 개수	외형그림
[바퀴세트] 샤프트 : 1개바퀴 : 1쌍 (13개)	
서보 모터HES-1188(12개)	

<p>모터혼 모터혼 나사(각 12개)</p>	
<p>볼트 (3ϕ 5mm) 너트 (3ϕ 2mm) 와셔 (3ϕ)</p>	
<p>[서포트] 10mm(F) (4개) 5mm(M) (48개) 7mm(M) (30개) 20mm(M) (4개)</p>	
부품명 및 개수	외형 그림
<p>상부 보드(1개)</p>	
<p>하부 보드(1개)</p>	
<p>초음파 모듈(1개)</p>	
<p>LCD 모듈(1개)</p>	

부품명 및 개수	외형 그림
뱀 로봇 연결선(1개)	
충전지 연결판(1개)	
건전지 전원선(1개)	
충전지 홀더 1AA(1개) 2AA(2개)	
1.2V 충전지(5개) 9V 충전지(1개)	
부품명 및 개수	외형그림
용로우즈(1개)	
+드라이버(1개)	

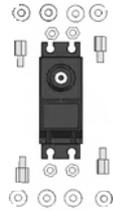
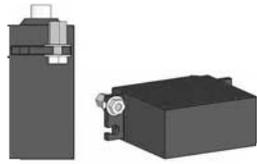


2 로봇 조립

1) 몸체프레임 조립

<p style="text-align: center;">1단계</p> <p>우측의 그림과 같이 부품을 준비한다. 몸체 프레임 : 1개 서포트 7mm : 2개 와셔 : 2개 너트 : 2개</p>	
<p style="text-align: center;">2단계</p> <p>몸체 프레임에 서포트와 와셔 너트를 조립한다. 조립 결과는 우측 사진과 같다.</p>	
<p style="text-align: center;">3단계</p> <p>다른 서포트도 상기와 같은 방법으로 조립한다. 조립 결과는 우측 사진과 같다. 1~3단계를 거쳐 12개의 프레임 조립을 완성한다.</p>	

2) 모터 조립

<p style="text-align: center;">1단계</p> <p>우측의 그림과 같이 부품을 준비한다. 모터 : 1개 와셔 : 8개 너트 : 4개 서포트 5mm : 4개</p>	
<p style="text-align: center;">2단계</p> <p>각 부품을 우측 사진과 같이 위치시키고 조립한다.</p>	
<p style="text-align: center;">3단계</p> <p>조립 결과는 우측의 사진과 같다.</p>	
<p style="text-align: center;">4단계</p> <p>1~3단계를 거쳐 나머지 부품들을 조립한다. 완성된 사진은 우측과 같다. 상기와 같은 방법으로 12개의 모터 조립을 완성한다.</p>	



3) 몸체와 모터 조립

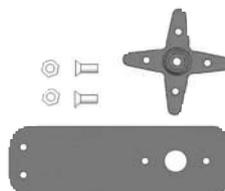
<p style="text-align: center;">1단계</p> <p>우측의 그림과 같이 부품을 준비한다. 1번과 2번의 조립 완성된 부분 몸체 프레임 : 1개 바퀴 프레임 : 1개 모터 : 1개, 볼트 : 4개</p>	
<p style="text-align: center;">2단계</p> <p>볼트 2개를 사용하여 모터와 몸체 프레임을 조립한다.</p>	
<p style="text-align: center;">3단계</p> <p>볼트 2개를 이용하여 몸체 프레임과 바퀴프레임, 모터를 조립 완성한다.</p>	
<p style="text-align: center;">4단계</p> <p>완성된 몸체를 확인한다. 1~4단계를 거쳐 12개의 몸체를 완성한다.</p>	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> 밑면 </div> <div style="text-align: center;"> 안쪽 </div> </div>

4) 연결판 조립

1단계

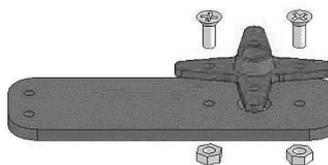
우측의 그림과 같이 부품을 준비한다.

연결판 : 1개
 모터 혼 : 1개
 너트 : 2개
 볼트 : 2개



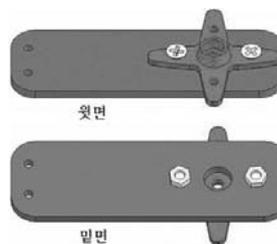
2단계

볼트와 너트를 이용하여 연결판에
 모터 혼을 조립한다.



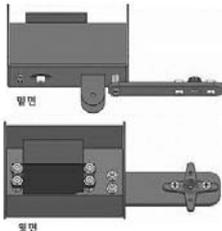
3단계

완성된 연결판을 확인한다. 1~3
 단계를 거쳐 12개의 연결판을 완
 성한다.

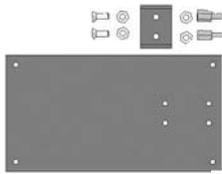
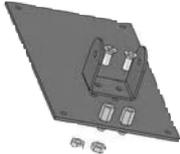


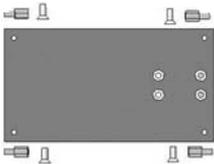
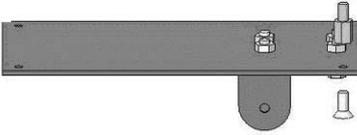
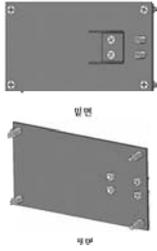


5) 몸체와 연결판 조립

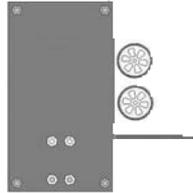
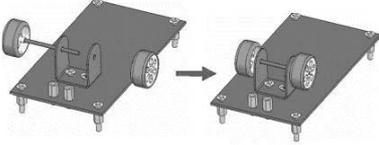
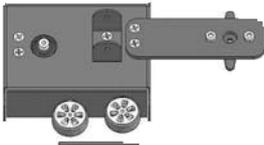
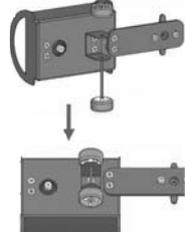
<p style="text-align: center;">1단계</p> <p>우측의 그림과 같이 부품을 준비한다. 연결판 : 1개 몸체 : 1개 볼트 : 2개</p>	
<p style="text-align: center;">2단계</p> <p>볼트 2대를 이용하여 몸체와 연결판의 조립을 완성한다.</p>	
<p style="text-align: center;">3단계</p> <p>완성된 몸체와 연결판의 조립을 확인한다. 1~3단계를 거쳐 11개를 완성한다. 작은 몸체 1개는 제외한다.</p>	

6) 머리판 조립

<p style="text-align: center;">1단계</p> <p>우측의 그림과 같이 부품을 준비한다. 머리판 : 1개 바퀴 프레임 : 1개 볼트 : 2개 너트 : 4개, 서포트 7mm : 2개</p>	
<p style="text-align: center;">2단계</p> <p>머리판에 서포트를 조립한다.</p>	
<p style="text-align: center;">3단계</p> <p>2단계 조립을 확인한다.</p>	
<p style="text-align: center;">4단계</p> <p>바퀴판을 머리판에 조립한다.</p>	

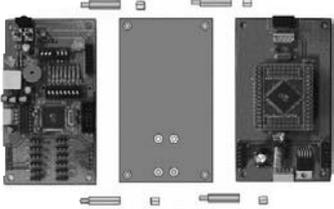
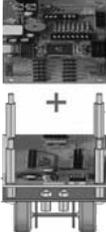
<p>5단계</p>	 <p>앞면</p> <p>뒷면</p>
<p>6단계</p> <p>우측의 그림과 같이 부품을 준비한다. 머리판 : 1개 볼트 : 4개 서포트 7mm : 4개</p>	
<p>7단계</p> <p>머리판에 서포트를 조립한다.</p>	
<p>8단계</p> <p>7단계 조립을 확인한다.</p>	 <p>앞면</p> <p>뒷면</p>
<p>9단계</p> <p>7단계와 같이 다른 부분도 조립을 완료하고 확인한다.</p>	 <p>앞면</p> <p>뒷면</p>

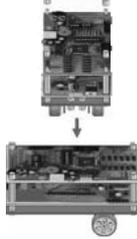
7) 바퀴 조립

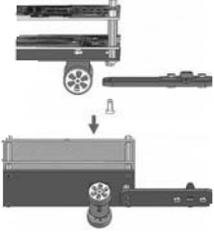
<p style="text-align: center;">1단계</p> <p>우측의 그림과 같이 부품을 준비한다. 머리판 : 1개 바퀴 세트 : 1개 즉, 바퀴 1쌍 샤프트 1개</p>	
<p style="text-align: center;">2단계</p> <p>머리판에 바퀴 세트를 조립한다.</p>	
<p style="text-align: center;">3단계</p> <p>우측의 그림과 같이 부품을 준비한다. 몸체 프레임 : 1개 바퀴 세트 : 1개</p>	
<p style="text-align: center;">4단계</p> <p>몸체에 바퀴 세트를 조립한다. 나머지 몸체에도 모두 바퀴를 조립한다.</p>	



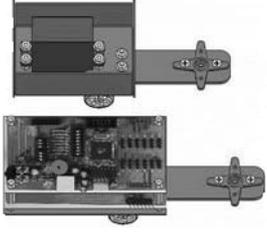
8) 머리 조립

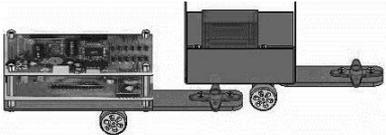
<p style="text-align: center;">1단계</p> <p>우측의 그림과 같이 부품을 준비한다. 머리판 : 1개 상부보드 : 1개 하부보드 : 1개 서포트 20mm : 4개 서포트 10mm : 4개</p>	
<p style="text-align: center;">2단계</p> <p>머리판의 서포트 홈에 하부 보드를 올린다. 화살표가 있는 곳이 바퀴가 위치한 곳이다.</p>	
<p style="text-align: center;">3단계</p> <p>하부보드가 올라간 곳을 서포트 (20mm)로 고정한다.</p>	
<p style="text-align: center;">4단계</p> <p>상부 보드를 올린다.</p>	

5단계	<p>상부보드가 올라간 곳을 서포트 (10mm)로 고정한다.</p>	
-----	---------------------------------------	--

6단계	<p>머리와 연결판을 볼트를 사용하여 조립한다.</p>	
-----	--------------------------------	--

9) 머리와 몸체 조립

1단계	<p>머리와 몸체를 준비한다.</p>	
-----	----------------------	--

2단계	<p>머리의 모터 혼에 몸체의 모터를 연결한다.(일렬로 조립 한다.) 이때 모터 혼의 나사는 조립하지 않는다.</p>	
-----	---	--



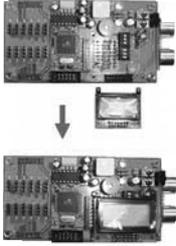
3단계	
조립 결과를 확인한다.	

4단계	
각각의 몸체와 몸체를 연결한다. (일렬로 연결한다.) 이때 혼의 나사를 잠그지 않는다.	

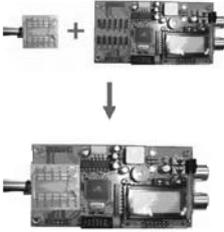
5단계	
머리와 몸체, 몸체와 몸체 연결을 확인한다.	

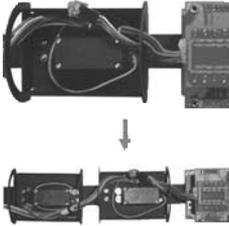
10) 초음파센서와 LCD 조립

1단계	
머리의 하부 보드에 초음파 모듈을 장착한다.	

2단계	<p>머리의 상부보드에 LCD 모듈을 장착한다.</p>	
-----	--------------------------------	--

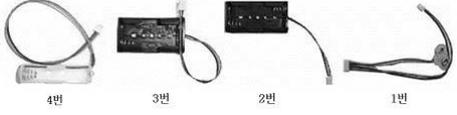
11) 연결선 조립

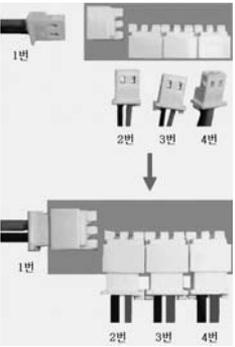
1단계	<p>연결선의 기판을 상부 보드에 연결한다. 이때 커넥터의 구멍을 잘 맞추어 끼운다.</p>	
-----	---	--

2단계	<p>연결선의 선들은 그림과 같이 몸체의 옆면 구멍을 통해서 지나가도록 처리한다. 꼬리부분까지 연결선이 위치하도록 한다.</p>	
-----	---	--

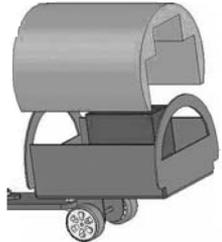
3단계	<p>모터선과 연결선을 연결한다. 이때 모터선의 색을 주의 한다. 모터 갈색 : GND, 적색 : VCC, 주황색 : SIGNAL 연결선 - 검정 : GND, 적색 : VCC GND <—> GND VCC <—> VCC SIGNAL <-> SIGNAL</p>	
-----	---	--

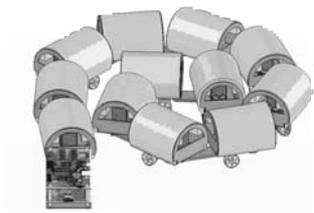
12) 충전지 조립

1단계	
<p>머리를 기준으로 각 몸체의 순서에 맞게 넣는다.</p>	

2단계	
<p>2번 몸체에 충전지 연결판을 놓고 각 몸체에 있는 건전지 선을 번호에 맞게 끼운다.</p>	

13) 덮개 조립

1단계	
<p>로봇의 몸체에 맞는 덮개를 덮는다.</p>	

2단계	
<p>덮개를 모두 덮고 확인한다.</p>	

4 뱀 로봇 구동실험 및 예제 실습

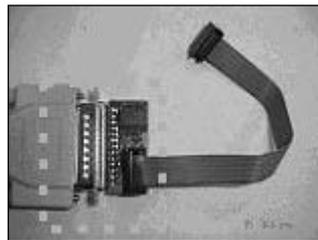
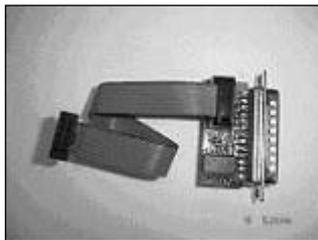
1 【실습 1】 로봇과 시리얼 통신 접속하기

※ 실습에 앞서 에뮬레이터를 실행시킨다.

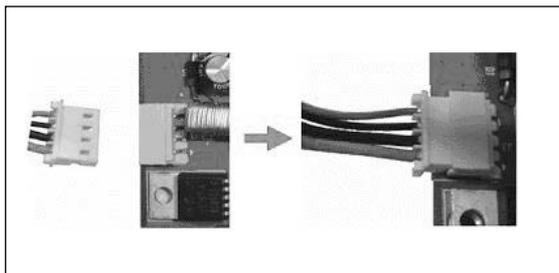
- ① 먼저 시리얼 통신을 위해서는 ISP(In System Programming) 다운로더로 (Down Loader) 시리얼 통신을 위한 프로그램을 로봇에 입력시켜야 한다.
- ② ISP 확장 케이블을 컴퓨터의 프린트 포트(25PIN)에 연결한다.



- ③ ISP 다운로더를 ISP 확장 케이블의 반대편에 연결한다.

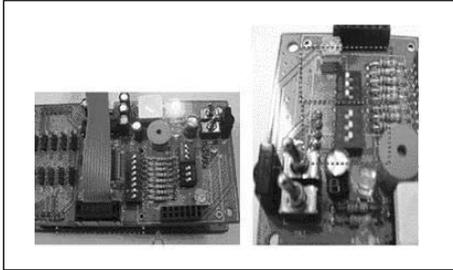


- ④ 보드 전원 커넥터를 연결한다.

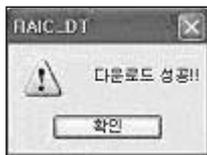




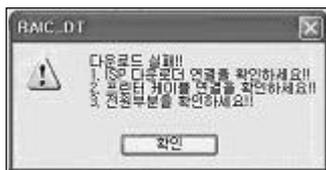
- ⑤ ISP 다운로드의 10PIN 박스 커넥터(Box Connector)를 보드의 A에 연결하고 B의 스위치 위치를 확인한다. MCU전원을 넣는다.(모터 전원은 넣지 않는다.)



- ⑥ 툴바의  킴파일 다운로드 아이콘을 클릭한다.
- ⑦ 잠시 기다리면 아래와 같은 메시지가 나오면서 다운로드가 완료되었음을 알린다.

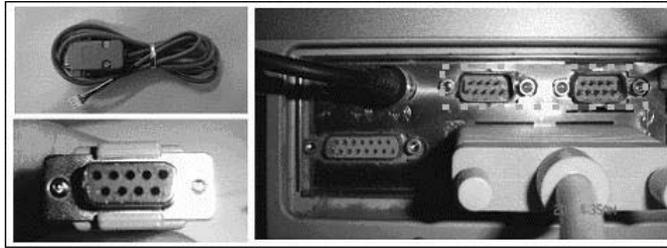


- ⑧ 다운로드 실패 시 다음의 메시지가 뜨며, 아래의 목록을 확인한다.

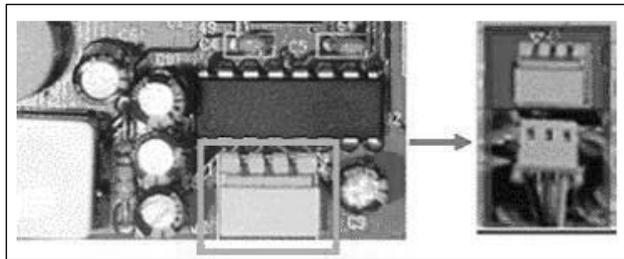


- ▶ 보드 전원을 제대로 연결하고 스위치를 ON으로 하였는지 확인한다.
- ▶ ISP 다운로드 10PIN을 보드에 제대로 연결하였는지 확인한다.
- ▶ 컴퓨터의 프린트 포트와 확장 케이블, 확장 케이블과 ISP 다운로드가 제대로 연결되었는지 확인한다.

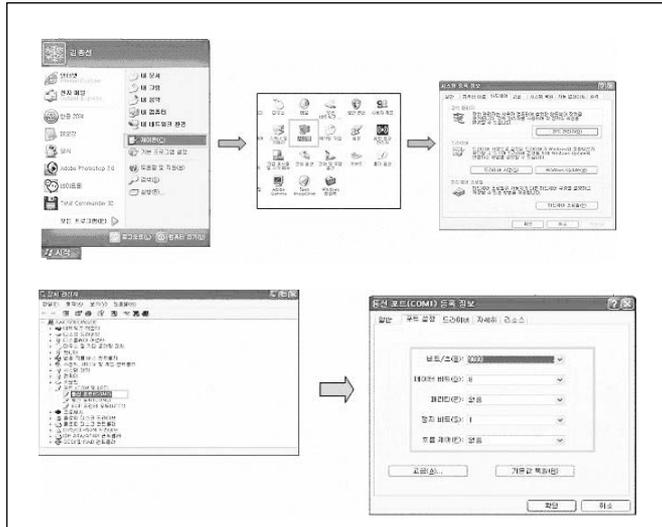
- ⑨ 메시지의 확인을 누른 후 전원을 OFF한 뒤, ISP 다운로더의 10PIN 커넥터를 보드에서 제거한다.
- ⑩ 시리얼 케이블을 준비하여 컴퓨터 본체 뒷면의 시리얼 포트 COM1, COM2 중 1곳에 케이블을 연결한다.



- ⑪ 케이블의 반대편은 로봇 컨트롤 보드의 아래와 같은 위치 (10PIN 커넥터 반대편 J2)에 연결한다.



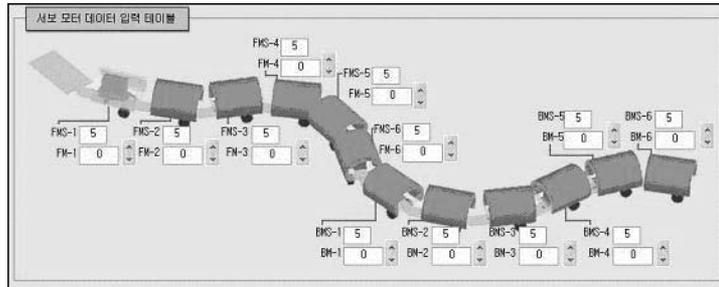
- ⑫ 아래와 같이 사용자가 선택한 컴퓨터 시리얼 통신 포트(1 또는 2)의 설정을 (여기에서는 포트 1을 설정) 9600, 8, 없음, 1, 없음으로 변경한다.



- ⑬ 이제 시리얼 통신을 위한 준비가 끝났으며 보드의 전원을 ON한다. 에뮬레이터에서 통신 시작을 누른다.
- ⑭ 보드 전원을 ON하면 에뮬레이터의 [시리얼 통신 상태 창]에 아래와 같은 문구(사각형)가 뜨면서 로봇과 연결되었음을 알려준다.



- ⑮ 로봇의 바닥을 손으로 잡아들어 올린 후 로봇 컨트롤 보드의 모터 전원을 ON 한다.
 - 모터 컨트롤 창이 활성화(입력 가능 상태)로 전환되는 것을 확인한다.



② 【실습 2】 시리얼 통신을 이용한 간단한 모터 제어

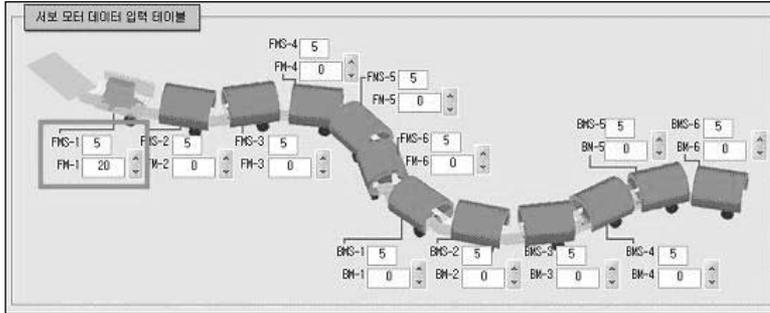
이미 앞장에서 시리얼 통신 접속을 위한 환경을 구축했다. 시리얼 통신을 통해 모터를 제어하기 위해서는 실습 1의 과정이 항상 반복되어야 한다는 것을 잊지 않도록 하자.

이제, 로봇과 접속된 시리얼 통신을 이용하여 모터를 직접 제어해 보기로 한다. 시리얼 통신으로 모터를 제어한다는 것은 직접 눈으로 주어진 모터 값을 확인할 수 있다는 이점이 있어서 모터의 동작을 만드는 데 매우 편리한 기능이다. 또한, 주어진 모터 값을 his 파일로 연속적으로 저장할 수 있기 때문에 또다시 같은 동작을 일일이 기억 할 필요가 없으므로 매우 유용한 기능이다.

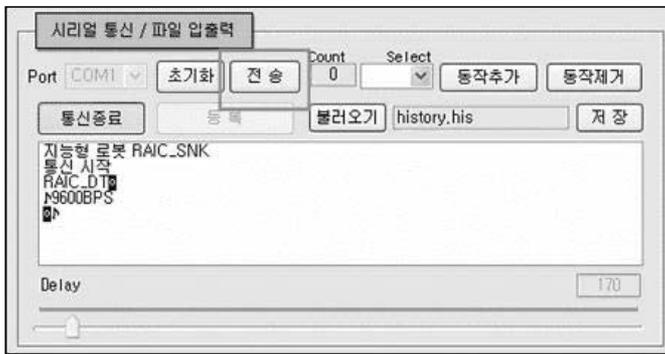
실습에 들어가기 전에, 실습 1의 과정에 따라 로봇과 시리얼 통신 접속을 하고 보드 전원 및 모터 전원을 ON 한 상태로 둔다.

본 실습의 목적은 시리얼 통신을 이용해 모터의 각도 및 속도를 제어하는 것이다.

① 서보 모터 데이터 입력 테이블의 모터 중 FM-1의 값을 그림과 같이 변경한다.



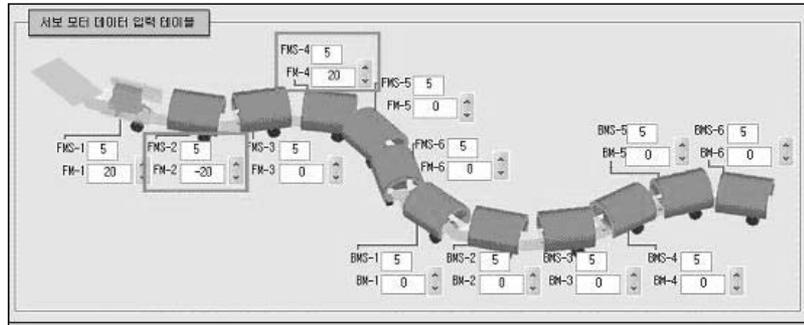
② 전송을 클릭한다. 해당 모터 값이 변하는 것을 확인할 수 있다.



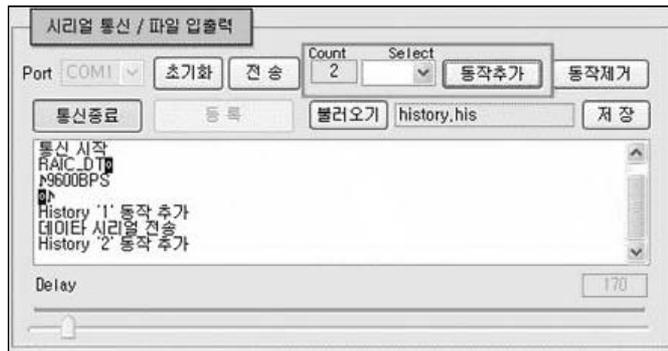
③ 현재 해당하는 모터 값의 동작을 추가한다. 카운트(Count) 값이 증가되는 것을 볼 수 있다.



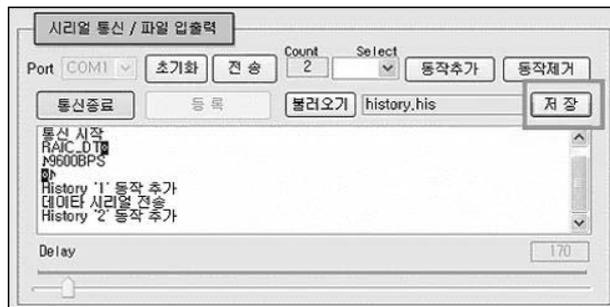
④ 다른 위치의 모터값을 변경한다.



⑤ 전송을 클릭하면 해당하는 모터가 이동하는 것을 확인할 수가 있다.



⑥ 2가지 로봇에 대한 모터 값이 임시 저장되었다. 이 동작을 his 파일로 저장하기 위하여 저장을 클릭한다.

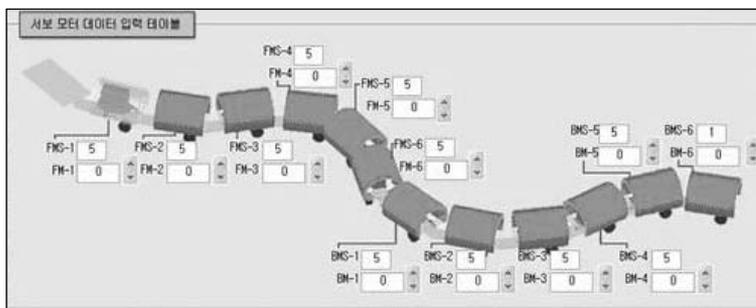
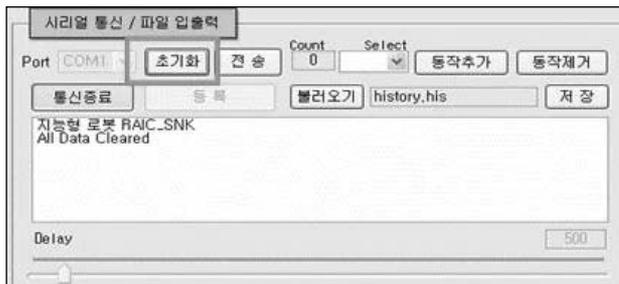




⑦ 파일 이름을 “동작1” 이라고 저장한다.



⑧ 초기화를 클릭한다. (현재 저장된 모든 임시동작 파일들이 지워진다) 모터테이블의 값도 초기 상태로 돌아간다.



⑨ 전송을 클릭하면 초기 상태의 값이 들어가므로 로봇의 상태가 최초 상태로

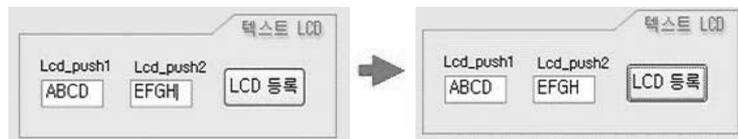
되돌아간다.

이와 같은 방법을 이용하여 모터의 값들을 계속 변화시킬 수가 있다. 여기서 동작1을 저장한 이유는 모터 값들을 다시 불러올 수 있도록 하기 위함이며, 저장한 파일에는 2개의 동작이 저장되어 있다.

③ 【실습 3】 텍스트 LCD에 글자쓰기

텍스트 LCD를 이용하여 글자를 표시하는 실습을 한다. 로봇에 장착되어 있는 LCD는 8×2를 표현하는 LCD이다. 즉 8개의 글자를 한 줄에 표시할 수 있으며 총 16글자를 표시할 수 있다. 그러나 본 LCD는 한글 표시가 불가능하다.

① 【메뉴】->【제어】->【LCD제어】를 클릭한다. 이때 텍스트 LCD의 에뮬레이터가 활성화 된다. 이제 자신이 기록하고자 하는 문자를 적으면 된다.



② 원하는 문자를 적었다면 LCD 등록 버튼을 클릭한다. 등록을 마치면 우측의 C 코드 상태 창에는 아래와 같은 문장들이 출력 된다.

```

C- 코드 입력 상태 / 편집
C-코드 지우기 C-코드 입력
#include "main.h"
int main(void){
Init_AVRSetting();
while(1){
Dat();
Lcd_push1("ABCD");
Lcd_push2("EFGH");
}
}

```



- ③ 상태 창에 표시가 되고 【실습1】의 1~7단계 과정을 거치면 즉 컴파일 및 다운로드 과정이 완료되었다면 LCD의 화면에 우리가 쓰고자 했던 글자들이 출력되는 것을 눈으로 확인할 수 있다.

4 【실습 4】 초음파 센서 테스트

초음파 센서를 이용하여 자신이 지정한 거리 값 안에 물체가 감지되면 그 거리를 LCD 화면에 출력하고, 동시에 물체가 감지되었다고 알리는 “삐” 소리를 내는 실습이다.

- ① 【메뉴】→【조건문】→【if】→【초음파 센서】를 클릭한다. 또는 툴바의  클릭한다. 선택을 하면 에뮬레이터에서 초음파 센서 부분이 활성화 된다. 자신이 물체를 감지하고 싶은 거리를 체크한다. 단, 한번 체크된 거리는 수정할 수 없다. 이미 체크 한 값을 바꾸려 하면 경고창이 뜬다.

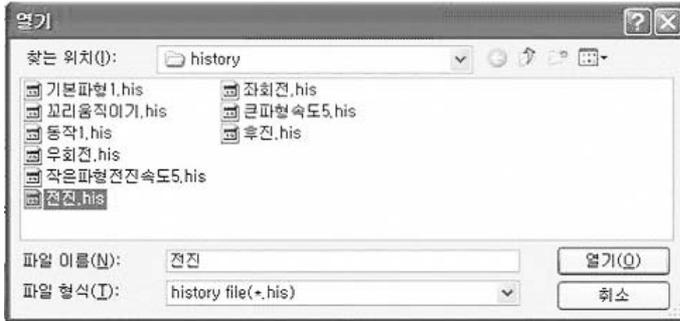


- ② 감지하고자 하는 거리를 결정하여 체크하고 【실습1】1~7단계를 거치면, 즉 컴파일 및 다운로드 과정이 완료되었다면 이제 초음파센서 근처에 자신의 손을 조금씩 가까이 하여 보자. 물체가 지정된 거리 안에서 감지 되면, “삐”하는 소리와 함께 LCD의 창에 거리가 측정되는 것을 확인할 수 있다.

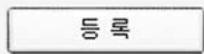
5 【실습 5】 초음파 센서와 로봇의 동작

본 실습에서는 초음파 센서를 사용하여 로봇이 주행 중에 설정한 범위 안의

장애물을 감지하면 다른 명령을 수행하는 실습을 한다.



① 【메뉴】->【제어】->【모터제어】를 클릭한다. 또는 툴바의 를 클릭한다. 모터제어 창이 활성화 된다. 【불러오기】를 클릭한 후, 【전진.his】를 열고 【등록】을 클릭한다.



② 【메뉴】->【조건문】->【if】->【초음파 센서】를 클릭한다. 또는 툴바의 를 클릭하면 에뮬레이터에서 초음파센서 부분이 활성화 된다. 자신이 물체를 감지하고 싶은 거리를 체크한다. 단, 한번 체크된 거리는 수정할 수 없다. 【불러오기】를 클릭한 후, 【후진.his】를 열고 【등록】을 클릭한다.



③ 【실습1】 1~7단계를 거치면, 즉 컴파일 및 다운로드 과정이 완료되었다면 다운로드 케이블을 제거하고 로봇의 구동을 확인한다. 전진 주행을 하다가 물체가 감지되면 후진을 한다. 단, 초음파 센서의 특성상 오차가 발생할 수 있

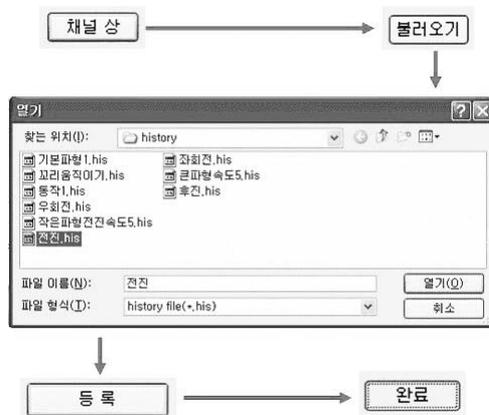


고 음파가 벽 또는 물체에 부딪혀 돌아올 때 왜곡될 수도 있다.
이는 초음파 센서의 특징이므로 잘못된 동작이 아니다.

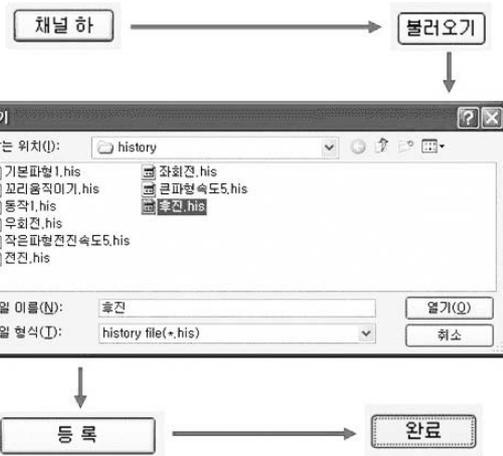
6 【실습 6】 로봇 제어를 위한 리모트 제어와 초음파 센서

본 실습에서는 로봇 제어를 위하여 리모트 제어와 초음파 센서 를 활용한 실습을 한다.

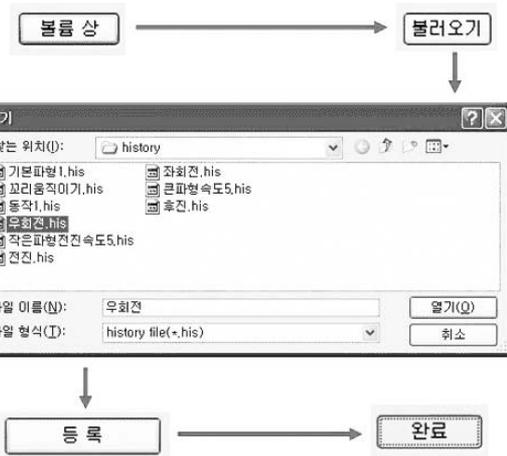
- ① 【조건문】->【Switch】->【리모트 제어】를 클릭한다. 또는 툴바의  를 클릭한다. 리모컨의 상태 창이 활성화 되었다면, 【채널상】을 클릭하고 【불러오기】를 누른후, 【전진.his】를 열고 【등록】을 클릭하여, 등록을 마쳤다면 리모컨의 【완료】를 클릭한다. 이 부분이 뱀 로봇의 전진에 대한 명령 및 데이터 등록을 마친 것이다. 그림으로 확인하여 보자.



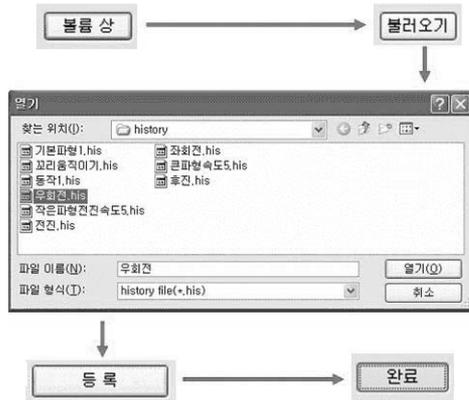
- ② 다음으로 【채널하】를 클릭하고 【불러오기】를 누른후, 【후진.his】를 열고 【등록】을 클릭하여, 등록을 마쳤다면 리모컨의 【완료】를 클릭한다. 이 부분이 뱀 로봇의 후진에 대한 명령 및 데이터 등록을 마친 것이다. 그림으로 확인하여 보자.



③ **【불륨상】**을 클릭하고 **【불러오기】**를 누른후, **【우회전.his】**를 열고 **【등록】**을 클릭하여, 등록을 마쳤다면 리모컨의 **【완료】**를 클릭한다. 이 부분이 뱀 로봇의 후진에 대한 명령 및 데이터 등록을 마친 것이다. 그림으로 확인하여 보자.



④ **【볼륨하】**를 클릭하고 **【불러오기】**를 누른후, **【좌회전.his】**를 열고 **【등록】**을 클릭하여, 등록을 마쳤다면 리모컨의 **【완료】**를 클릭한다. 이 부분이 뱀 로봇의 후진에 대한 명령 및 데이터 등록을 마친 것이다. 그림으로 확인하여 보자.



4가지 동작을 모두 등록하였다.

- ⑤ 【메뉴】->【조건문】->【if】->【초음파 센서】를 클릭한다. 또는 툴바의 를 클릭하면 에뮬레이터에서 초음파 센서 부분이 활성화 된다. 자신이 물체를 감지하고 싶은 거리를 체크한다. 단, 한번 체크된 거리는 수정할 수가 없다. 【불러오기】를 클릭한 후, 【후진.his】를 열고 【등록】을 누른다.



- ⑥ 이제 초음파 센서에 관한 것도 포함시켰다. 【실습1】1~7단계를 거치면 즉 킴파일 및 다운로드 과정이 완료되었다면, 다운로드 케이블을 제거하고 로봇의 구동을 확인한다. 로봇이 대기 상태에 있다가 지정된 리모컨의 버튼을 누르면 동작을 하게 되고, 주행 중 설정된 범위 안에 물체가 감지되면 후진을 실행한다.

5 뱀 로봇 프로그래밍

1 C언어의 이해를 위한 기초 지식

```

main( )
{
    int motor_1=0;           // 주석문 //
    motor_10;
    FMotor(motor_1 , 0, 0, 0, 0, 0);
}
    
```

main()함수

main()함수는 하나의 프로그램에 반드시 한 개가 존재하여야 한다.

1) main()함수

C 프로그램은 하나 이상의 함수들로 구성되며, 반드시 main() 함수로 시작된다. 위 프로그램에서 “{”는 main()함수의 시작을 나타내고 “}”는 main()함수의 끝을 나타낸다.

즉, main()에서 ()는 함수를 나타내며 “{”와 “}”사이의 내용으로 구성된다. 이것은 곧 “프로그램이 main() 함수의 { }로서 둘러싸인 부분만을 실행한다.”라는 의미이다.

2) 주석문(comment)

주석문이란 “프로그램이 어떠한 목적으로 누구에 의해서 작성되었는가” 또는 “각 루틴이나 문장의 논리에 대한 설명을 기술하는 것”을 말한다. 프로그램을 문서화 할 목적으로 많이 사용된다. 주석문은 /*로 시작해서 */로 끝나는 문자들의 집단을 의미하며 그 사이에는 어떠한 문자라도 사용이 가능하고 프로그램 중 여백 문자가 들어갈 수 있는 곳이면 어디든지 기술할 수 있다.



【보기】 주석문의 예

```
// 테마 로봇
/* 테마 로봇
*/
```

주석문

//는 한줄 만을 주석으로
처리 할 때 사용
/*~*/는 여러 줄을 주석
으로 처리할 때 사용

3) 문장(statement)

문장이란 프로그램을 구성하는 기본적인 요소로 문장의 끝을 나타내는 세미 콜론(:)을 반드시 포함하고 있다. 즉, 프로그램이란 어떠한 기능들을 수행하는 문장들로 구성된다고 말할 수 있다.

【보기】 문장의 예

```
int motor_1;
FMotor(motor_1 , 0, 0, 0, 0, 0);
```

2 자료(Data)와 자료형(Data Type)

1) 변수와 상수

컴퓨터 프로그램이 인간의 자료를 처리하기 위해서는 먼저 자료의 형태를 구분지어야 하고, 구분되어진 자료는 변수(Variable)와 상수(Constant)로 표현된다.

상수란 프로그램이 실행되는 동안 변하지 않는 데이터를 말하며, 변수란 프로그램이 실행되는 도중에 자료의 크기 형태 등이 변하는 데이터를 말한다.

이들 명령, 기호적 명칭, 상수 값 사이에는 띄어쓰기(키보드의 스페이스바)를 통해 구분 짓는다.

1) 자료형의 유형

① 정수형

소수부가 없는 데이터형을 말한다. signed는 양수 값과 음수 값을 모두 표현할 수 있으며 unsigned는 양수 값만 표현할 수 있다. signed는 음수 값이 들어갈 수 없고, 양수 값만 저장될 수 있는 변수를 선언할 경우 사용한다.

자료 형	비트 수	범위
int	16,32 비트	-32768 ~ 32767
short	16 비트	-32768 ~ 32767
long	32 비트	-2147483648~2147483647
unsigned int	32 비트	0~65535
unsigned short	16 비트	0~65535
unsigned long	32 비트	0~4294967295

〈표VII-2〉 정수형 크기

② 실수형

소수부가 있는 자료형을 말하며 매우 큰 값의 표현이나 아주 작은 값들을 처리하기에 적당하다.

자료 형	비트 수	범위
float	32 비트	소숫점 이하 6자리 표현
double	64 비트	소숫점 이하 14자리 표현
Long double	80 비트	소숫점 이하 19자리 표현

〈표VII-3〉 실수형의 크기

③ 문자형

C 에서 문자 데이터는 하나의 문자를 의미하며 정수 데이터 유형의 경우와



같이 unsigned 가 붙어 있는 것과 그렇지 않은 것이 있다. 문자를 컴퓨터에 표현하려면 특정 코드를 사용해야 하는데, C에서는 ASCII라는 이름의 코드를 사용한다. ASCII 코드는 7bit로 문자를 표현하는데, 메모리의 기본단위가 1byte 이기 때문에 8bit로 한 문자를 나타내게 한다. 그런데 이 코드라는 것이 2진수 이기 때문에 겉으로 볼 때 정수와 전혀 차이가 없어 C에서 아예 이들의 구분을 없애 버렸다. 단지 문자 데이터 유형은 1byte의 크기를 갖는 정수 데이터 유형으로 간주하게 된 것이다. 즉, C에서의 문자 데이터 유형은 크기가 1byte인 정수 데이터 유형으로 생각하면 된다.

ASCII(American Standard Code for Information Interchange)

1963년 미국표준협의회에 의해 결정된 미국표준 부호로서, 128개의 가능한 문자조합을 제공하는 7bit는 부호이다.

자료 형	비트 수	범위
char	8비트	-128 ~ 127
unsigned char	8비트	0~255

〈표Ⅶ-4〉 문자형의 크기

3 제어문

여러분이 좋은 프로그램을 만들려고 한다면, 필연적으로 프로그램의 흐름(Flow)을 제어(Control)할 수 있어야 한다. 따라서 프로그램을 작성하는 언어는 다음 3가지 정도의 조건을 충족할 수 있어야 한다.

첫째, 연속된 일련의 명령들을 수행할 수 있어야 한다.

둘째, 어떤 조건이 만족될 때까지 일정한 부분을 반복할 수 있어야 한다.

셋째, 조건에 따라 명령문을 선택할 수 있어야 한다.

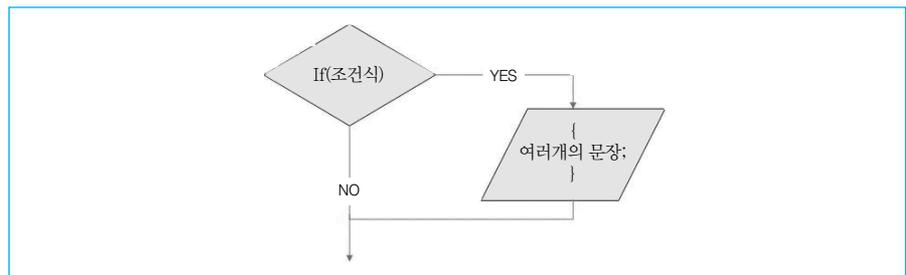
우리는 이미 연산자를 통해 첫째 조건을 배웠다. 이제 둘째와 셋째를 배울 차례이다. 프로그램에서 선택, 반복할 수 있는 구문을 제어문이라 하며 그 첫 번째는 선택문으로 if, else, switch case문을, 두 번째는 반복문으로서 while, for 문 등을 배울 것이다.

1) if 문

형식 :

```
if(조건식)
    여러 개의 문장;
```

조건식이 참이면 다음에 나오는 문장을 수행하고 거짓이면 수행하지 않는다. 이때 실행문장은 단일 명령문이거나 또는 여러 문장들이 될 수 있는데, 실행문장이 두 문장 이상인 경우는 반드시 블록으로 묶어야 한다. 조건식에는 거의 관계 표현식이 사용된다.



〈그림Ⅶ-14〉 if문의 기본 구조

●● 예제 프로그램 1 ●● if문을 이용한 리모컨 제어

이 예제에서는 remote 변수 즉 리모컨의 수신 데이터의 값을 저장하고 있는 변수를 if문을 이용하여 비교 판단하여 해당하는 동작을 수행한다.

```
#include "main.h"
```

```
int main(void)
{
```



```
Init_AVRSetting(); // 초기화
Dat(); // 센서 값을 읽어옴
while(1) // 무한 반복
{
    if(remote == channel_up)
    { //리모컨 값 비교
        FRONT_LS1();
        Delay_ms(500);
    }
    if(remote == channel_down)
    {
        BACK_LS1();
        Delay_ms(500);
    }
    if(remote == volume_up)
    {
        RIGHT_S1();
        Delay_ms(500);
    }

    if(remote == volume_down)
    {
        LEFT_S1();
        Delay_ms(500);
    }
}
```

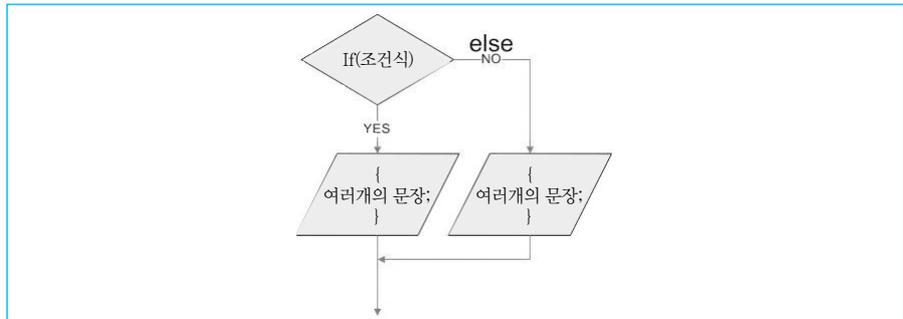
}

결과

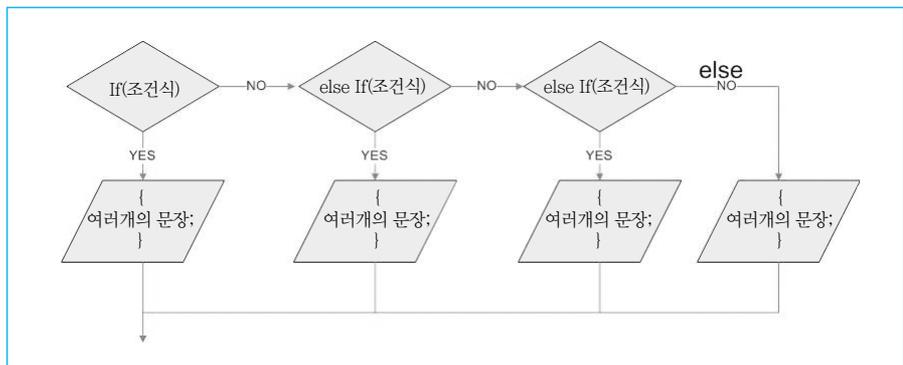
리모컨의 채널 업 버튼을 누르면 앞으로 가는 함수의 동작을 수행하며 채널 다운 버튼을 누르면 뒤로 가는 함수를 수행한다. 그리고 볼륨 업 버튼을 누르면 오른쪽으로 회전 하고 볼륨 다운 버튼을 누르면 왼쪽으로 회전 한다.

2) if else 문과 그 확장

if 문은 연속적인 질문의 확장이 가능하다. 먼저 그 첫 번째 확장인 if - else 의 구조를 알아보자.



<그림VII-15> if-else문의 기본 구조



<그림VII-16> if-else문의 확장



else if문은 if else 문의 연속이라 생각하면 매우 간단하다. else 라는 말이 if 앞에 붙은 이유는 해당 else if 문 바로전의 if, 혹은 else if 문의 거짓(No)으로서 또 다른 선택을 위해 if를 붙여야 하기 때문이다. 그렇지만 여기서 주의해야 할 것은 조건식이 만족하는 if 또는 else if문을 만나면 그 하위 else if 문은 실행되지 않는다. 아래 표의 프로그램은 같은 구문을 if에서 if else로 변환한 것이다.

if	if else
<pre>while(1) { if(remote==4) { FMSpeed(1 , 0, 0, 0, 0, 0); FMotor(-24 , 0, 0, 0, 0, 0); Delay_ms(1000); } if(remote==5) { FMSpeed(2 , 0, 0, 0, 0, 0);</pre>	<pre>while(1) { if(remote==4) { FMSpeed(1 , 0, 0, 0, 0, 0); FMotor(-24 , 0, 0, 0, 0, 0); Delay_ms(1000); } else if(remote==5) { FMSpeed(2 , 0, 0, 0, 0, 0);</pre>
<pre> FMotor(0 , 0, 0, 0, 0, 0); Delay_ms(1000); } FMotor(8 , 0, 0, 0, 0, 0); Delay_ms(1000); }</pre>	<pre> FMotor(0 , 0, 0, 0, 0, 0); Delay_ms(1000); } FMotor(8 , 0, 0, 0, 0, 0); Delay_ms(1000); }</pre>
<p>두 개의 if 구문은 프로그램 실행 시 모두 조건식을 검사하게 된다. 또한 맨 아래의 문장 FMotor(8) Delay_ms(1000) 는 프로그램 실행 시 항상 실행된다.</p>	<p>좌측의 구문을 똑같이 if else 를 이용하여 구현한 것이다. 여기서 첫 번째 if(remote==4) 구문이 참이 될 경우 else if(remote==5)구문은 실행되지 않고 바로 맨아래의 구문 FMotor(8) ms(1000)을 바로 실행한다.</p>

for문을 이용한 무한루프
for(; ;)
 이와 같이 for문 내에 어떠한식도 넣지 않으면 프로그램이 무한 반복되어 실행된다.

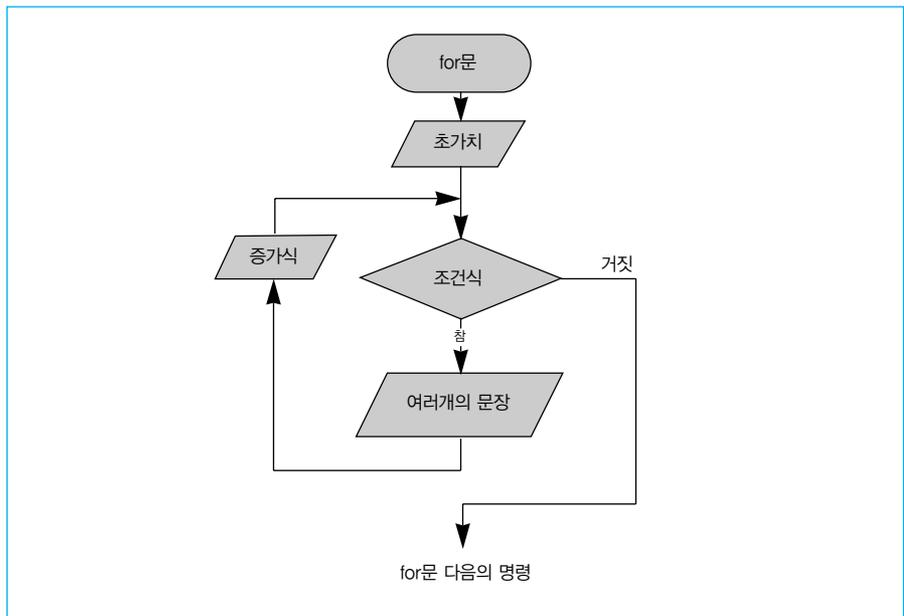
이처럼 else if 문은 조건식이 참이 될 때 나머지 구문을 실행하지 않으므로써 프로그램의 속도를 빠르게 해주는 효과가 있다. 또한 if else의 마지막 구문 else {}는 앞의 if else 구문의 조건식이 모두 만족하지 않을 때 실행이 된다.

다. for문

형식 :

for(초기식 ; 조건식 ; 종감식)
 실행문장 ;

for문은 기능이 강력하고 유연성이 있다. 세 개의 독립된 문은 여러 종류의 변형을 만들어 낸다.



〈그림VII-17〉 for문의 구조



for문의 동작 순서

- ① 초기식을 수행한다.
 - ② 조건식을 판별한다.
 - ③ 조건식이 참이면 실행 문장을 실행하고, 증감식을 실행한다.
- 위 ②~③단계 반복하며 조건식이 거짓이 될 때까지 한다.

●● 예제 프로그램 2 ●● 반복문 if문을 이용한 리모컨 제어

For 문을 이용해 로봇 반복 동작을 하는데 정수 변수 i는 0에서부터 2까지 증가하면서 조건식을 만족한다. i가 3으로 증가한 순간 조건식 $i < 3$ 에 거짓이 되므로 아래의 수행문을 더 이상 반복하지 않는다.

```
#include "main.h"

int main(void)
{
    int i = 0; // 정수형 변수

    Init_AVRSetting();// AVR 초기화

    Delay_ms(2000); / 약 2초 정도 대기

    for(i =0; i < 3; i ++)//3회 반복
        FRONT_S10 ;    //전진 동작
        Delay_ms(1000);
```



```

for(i=0; i<3 ; i++) // 3회 반복
    FRONT_LS30 ; // 후진 동작
    Delay_ms(1000);
    for(i =0; i < 3; i ++ ) //3회 반복
        BACK_S10; // 빠르게 전진
        Delay_ms(1000);

for(i=0; i<3 ; i++) // 3회 반복
    BACK_LS10;
    Delay_ms(1000);
}

```

결과

로봇을 초기화 한 후 약 2초 정도 로봇은 대기한다. 그리고 전진 동작 3번을 하고 다시 후진 동작 3번을 한 후에 빠른 전진 3번 후 빠른 후진 3번을 하고 로봇은 정지한다.

4) switch - case 문

형식 :

```

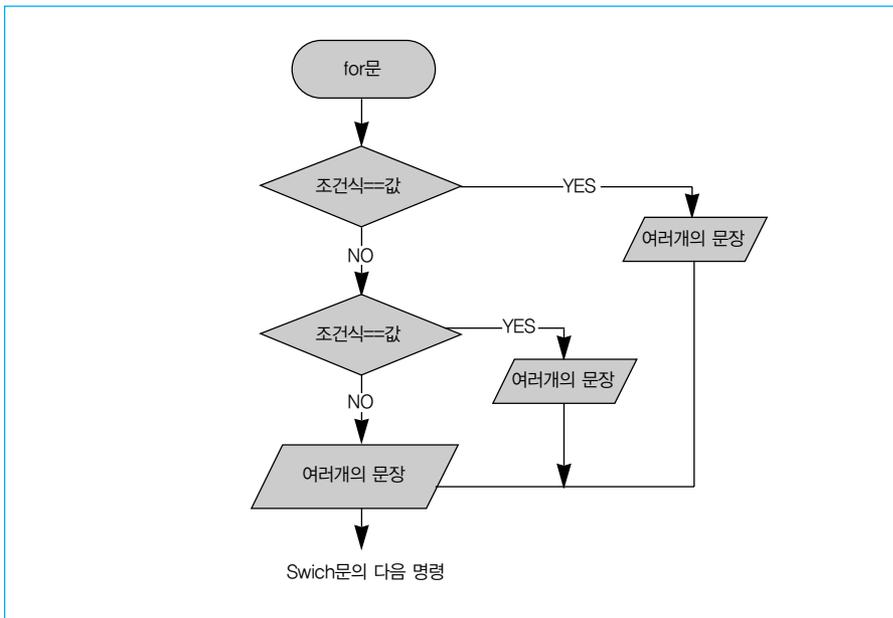
switch(수식){
case 값1 : 문장1;문장2;문장3;...문장n;break;
case 값2 : 문장1;문장2;문장3;...문장n;break;
case 값3 : 문장1;문장2;문장3;...문장n;break;
.
default : 문장1;문장2;문장3; ... 문장n; }

```



switch 문에서는 수식의 결과 값과 case 다음에 나오는 결과값이 일치하게 되면 그 행으로 점프하여 그 다음에 나오는 문장들을 수행한다. 명령문들 다음에는 break 문이 올 수 있는데 이는 문장들을 수행하고 switch 문을 종료시켜 주는 역할을 한다. break 문이 없으면 그 다음 문장들을 계속 수행한다.

default 다음에 나오는 문장은 수식의 값과 case 다음에 나오는 값들과 일치하는 값이 없을 경우 수행하는 문장이며, default 다음에는 break 문을 사용하지 않아도 자동적으로 switch 문이 종료된다.



〈그림VII-18〉 switch문의 구조

●● 예제 프로그램 3 ●● 조건식 switch - case을 이용한 LCD 제어하기

remote 변수는 리모컨의 수신 데이터 값을 가지고 있는 변수인데 조건문 switch-case문을 이용하여 각 버튼의 값을 비교 판단하고 Lcd_push1(), Lcd_push2() 함수는 LCD의 각 행에 문자열을 출력할 수 있게 해주는 함수이다.

```
#include "main.h"
int main(void)
{
    Init_AVRSetting(); //AVR 초기화
    while(1)
    {
        switch(remote) //remote 변수값 비교
        {
            case channel_up :
                Lcd_push1("Channel");
                // 1행에 문자열 출력
                Lcd_push2(" UP ");
                // 2행 문자열 출력
                break;

            case channel_down :
                Lcd_push1("Channel");
                Lcd_push2(" Down ");
                break;

            case volume_up :
                Lcd_push1("Volumel");
                Lcd_push2(" UP ");
                break;
```



```

case volume_down:
    Lcd_push1("Volume1");
    Lcd_push2(" Down ");
    break;
}
}
}

```

결과

리모컨의 수신 값과 remote 변수 값을 switch-case을 이용하여 구별하여 채널 업 버튼을 누르면 Lcd_push1("Channel"), Lcd_push2("UP")을 수행하여 로봇에 LCD에는 1행에 "Channel, UP"이 출력된다. 나머지 버튼을 누르면 해당되는 이름이 출력된다.

5) while 문

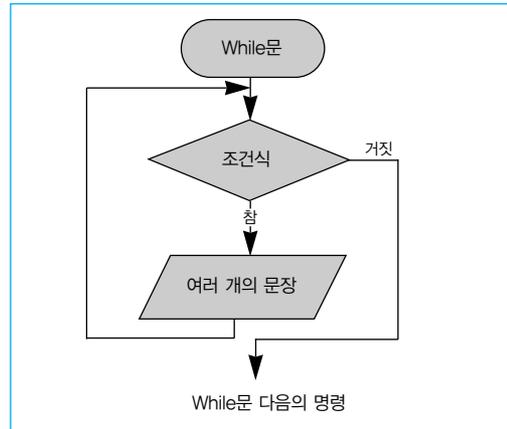
형식

```

while(조건식)
    실행문장;

```

조건식을 만족하는 동안 { } 안의 문장을 반복 수행하고 만족하지 않으면 while 다음 문장을 수행한다.



〈그림Ⅶ-19〉 while문의 구조

이전 예제 프로그램에 쓰인 `while(1)` 이라는 무한 반복을 생각해 보자. C언어에서 0이 아닌 모든 수는 참으로 인식한다. `while`의 조건식에 1을 쓰면 항상 참으로 인식하여 무한 반복을 하게 된다.

●● 예제 프로그램 4 ●● 반복문 `while`를 이용한 초음파 센서 동작

`usw()`는 초음파 센서를 이용하여 거리 측정 할 때 측정할 거리 설정하는 함수이다.

함수 인자에 15, 20, 25, 30, 35을 넣는데 아래와 같이 `usw(20)`을 설정하면 장애물이 20cm 안으로 들어면 `usw_1`이란 변수의 값이 1로 변경된다. `usw_1`변수는 초음파 센서에 장애물이 인식되면 `usw_1`의 값이 0에서 1로 변화 한다. 그러므로 `usw_1`의 값을 확인해서 장애물을 인식했는지에 대해서 알 수 있다.

```
=====
```

```
#include "main.h"
```

```
int main(void)
```



```

{
  int i=0;
  Init_AVRSetting();//AVR 초기화
  Dat0; // 센서 값 갱신
  while(1) // 무한 반복
  {
    usw(20);//장애물 감지거리 20cm로설정
    if(usw_1 ==1)// 장애물 감지 확인
    {
      while(i < 3)
      {
        BACK_LS10; // 후진 동작
        Delay_ms(500);
        i++;
      }
      i=0;
      usw_1=0;
    }
  }
}

```

결과

위 프로그램은 장애물이 20cm안으로 들어오면 while(i<3)에서 while 문이 참이 될 때까지 반복 하면서 후진 동작을 한다. while(1)은 무조건 참이므로 무한 반복을 위해서 사용한 것이다.

연산자 '=' 과 “==”의 비교

1) 연산자 '='는 대입의 의미이다.

예) int i = 0

⇒ 변수 i에 0을대입한다.

2) 연산자 “==”는 같다는 의미이다

예) i == 0

⇒ 변수 i의 값은 0과 같다.

단원 학습 정리



- 1** 뱀 로봇의 시스템 구조는 2개의 Controller 보드가 USART 통신을 통해서 데이터를 주고받는 것을 기반으로 한다. 상부 Controller 보드에 서는 12개의 RC-Servo Motor와 리모컨 센서를 제어하고 하부 Controller 보드는 장애물 감지를 위한 초음파 센서와 초음파 센서의 거리 값을 표시하는 LCD를 제어한다.
- 2** 메뉴는 에뮬레이터의 기능을 사용할 수 있으며 상태 표시줄 표시, 컴파일 다운로드, 조건문, 제어, 에뮬레이터 정보 보기의 과정을 수행할 수 있다. 각 메뉴는 단축키 사용이 가능하다.
- 3** 서보 모터 데이터 입력 테이블은 【메뉴】->【제어】->【모터제어】클릭 시 활성화 되며, 툴바에서 어에 추가적인 조건들을 제어하는 부분이다. 이는 마우스의 클릭으로 사용이 가능하다.
- 4** 로봇 제어 요소는 LCD의 글자 입력, 초음파 센서의 거리선택, 리모컨의 코드 선택을 할 수 있는 부분으로 로봇의 제어에 추가적인 조건들을 제어하는 부분이다. 이는 마우스의 클릭으로 사용이 가능하다.



5 초음파 센서는 【메뉴】→【조건문】→【if】→【초음파 센서】클릭 시 활성화 되며, 툴바에서  선택 시 활성화 된다. 그런 다음 정하고 싶은 거리를 정하면 된다. 단, 한번 거리를 선택하고 수정하면 안 된다.

6 리모컨은 【메뉴】→【조건문】→【Switch】→【리모트 제어】클릭 시 활성화 되며, 툴바에서  선택 시 활성화된다. 리모컨 이용한 제어 시 사용할 데이터 값이다. 총 13개의 리모컨 버튼을 사용할 수 있다. 해당 버튼을 클릭한 뒤 원하는 모터값 또는 데이터를 입력 후 【완료】 버튼을 클릭하면 된다.

7 C 코드 상태창은 컴파일 할 현재의 C 코드 상태를 보여주는 창으로서 버튼으로 클릭한 입력상태를 직접 눈으로 확인 할 수 있다. 컴파일 상태 창은 【메뉴】→【프로그램】→【컴파일 다운로드】클릭 시, 또는 툴바의  선택시 활성화 된다. 컴파일 에러 및 다운로드 완료 상태를 확인 할 수 있다.



○ 찾아보기

가반중량	42	범퍼회로	85
가이드센서	56	베어링프레임	50
관절형로봇	24	벨트폴리	49
광량센서	32	볼트스크루	50
그리퍼	78	블루투스	33
극좌표로봇	15	비상정지	85
근접센서	64	센서	31
납축전지	31	수평다관절로봇	43
다관절로봇	16	스카라로봇	17
대향성초음파센서	73	스테핑모터	13
데모프로그램	112	스토퍼	51
라인트레이서	24	슬라이더	41
로봇소프트웨어설계	34	시리얼통신	86
로봇의어원	11	아코디언식운동	48
로봇제어방법	78	액추에이터	63
리모트컨트롤	70	어칭멘던트	14
리밋	51	에뮬레이터	214
마이크로마우스	24	엔드이펙트	42
명령어사용법	185	연산자	269
모델링	85	온도센서	32
모터드라이버	160	원점복귀	104
모터배열	58	원통좌표계로봇	15
모터제어	245	유압구동모터	13
무선통신	33	이동로봇	44
반사형초음파센서	73	인코더	17
배터리과방전시은행	164	입력센서	32
베틀로봇	24	자동모드	201
뱀로봇	57	자동전원차단기능	85

자료형	255	플래시메모리	67
적외선거리감지센서	32	플로차트	35
적외선센서	47	필러	15
적외선	56	하모닉드라이브	64
전기구동모터	13	핸드모듈	42
전기배선도	79	회전모듈	62
전원공급기	122	휴머노이드	19
절대값인코더	51	AGV.....	45
절대모드	145	AVR	29
제어문	257	C언어	35
제어보드	156	CAM	26
제어부	63	CAN	121
제임스왈트	12	DC모터	29
조향제어방법	164	DSP보드	170
주석문	254	for문	262
주행선로	200	ISP	29
직각좌표계로봇	14	LM가이드	51
직교로봇	40	main()함수	254
창작로봇	57	MCU	28
초음파거리감지센서	31	PIC	29
캐릭터LCD디스플레이	75	RC서보모터	30
커넥터브라켓	63	RISC	29
커넥터	63	TP	163
커플링	50	USB단자	121
컴파일러	35		
태그인식센서.....	161		
팔레타이징	16		
팔레타이징	43		

▶ 참고문헌 ◀

- [1] K. Dowling, “Limbless locomotion: Learning to crawl with a Snake Robot”
- [2] B.C. Jayne, “Kinematics of terrestrial snake locomotion,” *Copeia*, vol. 4, pp. 915–927, 1986
- [3] H. Lissmann, “Rectilinear locomotion in a snake(*Boa occidentalis*),” *J. Exp. Biol.*, vol. 26, pp. 368–379, 1950
- [4] R. Worst and R. Linnemann, “Construction and operation of a snake-like robot,” in *Proc. IEEE Int. Joint Symp. Intelligent System*, Gaithersburg, MD, 1996, pp. 164–169
- [5] *The GMD-Snake Real-Time Scheduling of a Flexible Robot Application at Run-Time (1997)*
- [6] 박광식, 유영선, 주영훈, 두평수, 박현빈, “PDA를 이용한 스네이크 로봇의 동작 제어” 한국 퍼지 및 지능시스템학회 추계학술회의, Vol. 14, No. 1, pp. 418–421, 2004, 4
- [7] 김성주, 김종수, 전홍태, “지능형 뱀 로봇에 관한 연구” 한국퍼지및지능시스템학회논문지, 1598–7078, 제13권1호, pp.70–75, 2003
- [8] http://mail.fibo.kmutt.ac.th/FIBO_Eng/
- [8] <http://www.ucs.louisiana.edu/~brm2286/locomotn.htm>
- [9] 홍준희, 송화섭 “자동화를 위한 메카트로닉스” 시그마프레스
- [10] 삼성첨단기술연구소 “로보틱스 입문”
- [11] 이대길/최상민 공역 흥릉과학출판사 RSP(ROBOT SIMULATION PACKAGE) 사용자 매뉴얼
- [12] 川材貞夫 저 이종선 역성안당 “ESP 로봇제어 입문”

