
차 례

학습모듈의 개요	1
학습 1. 로봇 모니터링 툴 개발하기	
1-1. 로봇 모니터링 프로그램 개발	3
• 교수·학습 방법	13
• 평가	14
학습 2. 외부 툴과의 동기화 소프트웨어 개발하기	
2-1. 로봇 원격 조정 툴 개발	17
• 교수·학습 방법	29
• 평가	30
학습 3. 로봇 응용 소프트웨어 개발하기	
3-1. 로봇 원격 모니터링 앱 개발	33
• 교수·학습 방법	51
• 평가	52
학습 4. 로봇 훈련 소프트웨어 개발하기	
4-1. 로봇 학습 알고리즘 개발	55
• 교수·학습 방법	65
• 평가	66
학습 5. 다수 로봇 제어 소프트웨어 개발하기	
5-1. 다수 로봇 제어 프로그램 개발	69
• 교수·학습 방법	79
• 평가	80
참고 자료	83

로봇 운영 및 통합 소프트웨어 개발 학습모듈의 개요

학습모듈의 목표

로봇 하드웨어 상태를 모니터링하고 조정하는 프로그램을 통해 효율적으로 활용하기 위한 응용 프로그램을 설계, 작성, 수정하며 훈련하는 프로그램을 개발할 수 있으며, 여러 대의 로봇 및 서버가 연동하여 동작할 때 이를 일괄적으로 모니터링하고 원활하게 동작할 수 있도록 하는 프로그램을 개발할 수 있다.

선수학습

로봇 공학, 메카트로닉스, MCU 프로그래밍, GUI 프로그래밍, 스마트폰 앱 프로그래밍, 데이터 통신, 시리얼 통신, 블루투스 통신, 머신 러닝, 머신 비전, 산업용 네트워크

학습모듈의 내용 체계

학습	학습 내용	NCS 능력단위 요소	
		코드번호	요소 명칭
1. 로봇 모니터링 툴 개발하기	1-1. 로봇 모니터링 프로그램 개발	1903080311_14v1.1	모니터링 툴 개발하기
		1903080312_14v1.3	통합 모니터링 소프트웨어 개발하기
2. 외부 톨과의 동기화 소프트웨어 개발하기	2-1. 로봇 원격 조정 툴 개발	1903080311_14v1.2	외부 톨과의 동기화 소프트웨어 개발하기
3. 로봇 응용 소프트웨어 개발하기	3-1. 로봇 원격 모니터링 앱 개발	1903080311_14v1.3	로봇 응용 소프트웨어 개발하기
4. 로봇 훈련 소프트웨어 개발하기	4-1. 로봇 학습 알고리즘 개발	1903080311_14v1.4	로봇 훈련 소프트웨어 개발하기
5. 다수 로봇 제어 소프트웨어 개발하기	5-1. 다수 로봇 제어 프로그램 개발	1903080312_14v1.1	다수 로봇 제어 소프트웨어 개발하기
		1903080312_14v1.2	서버 연동 소프트웨어 개발하기

핵심 용어

로봇 모니터링 프로그램, 티치 펜던트, 스마트폰 앱, 머신 러닝, 머신 비전, EtherCAT

학습 1 모니터링 툴 개발하기

학습 2	외부 툴과의 동기화 소프트웨어 개발하기
학습 3	로봇 응용 소프트웨어 개발하기
학습 4	로봇 훈련 소프트웨어 개발하기
학습 5	다수 로봇 제어 소프트웨어 개발하기

1-1. 로봇 모니터링 프로그램 개발

학습 목표

- 여러 가지 종류의 로봇 컨트롤러 특성을 이해하고 제어할 수 있다.
- 여러 대의 로봇이 동시에 연동하여 움직일 때 발생할 수 있는 문제점을 예측하여 제어하는 프로그램을 구현할 수 있다.
- 로봇의 상태를 쉽게 파악할 수 있는 모니터링 프로그램을 작성할 수 있다.
- 모니터링 프로그램을 설치하고 배포할 수 있다.
- 로봇의 동작 상태 및 주변 환경을 파악하는 프로그램을 설계할 수 있다.
- 각종 센서의 정보를 취득하고 분석하는 프로그램을 설계할 수 있다.
- 설계된 모니터링 프로그램을 코딩할 수 있다.
- 웹, 앱을 통해 로봇을 제어하고 모니터링 할 수 있는 프로그램을 작성할 수 있다.
- 모니터링 프로그램을 설치, 배포할 수 있다.

필요 지식 /

① 마이크로컨트롤러의 통신 방식

RS232C는 PC와 마이크로컨트롤러(MCU(micro-controller unit)라고도 한다)가 데이터를 주고 받기 위한 통신 방식이다. 또 RS232C 통신 방식을 이용하여 직렬 통신(serial communication)하기 위한 마이크로컨트롤러의 기능을 USART(universal synchronous and asynchronous serial receiver and transmitter)라 한다.

1. 직렬 통신

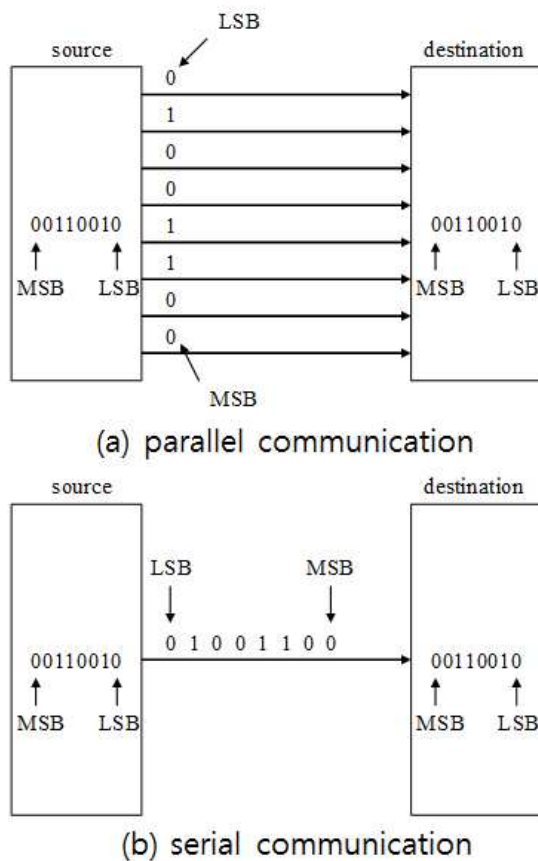
데이터를 전송하는 방식으로는 여러 개의 전송 매체를 이용하여 동시에 전송하는 병렬 통신(parallel communication)과 1개의 전송 매체를 이용하여 한 번에 한 비트씩 전송하는 직렬 통신(serial communication)으로 나눌 수 있다.

병렬 통신은 여러 개의 데이터 전송 라인을 이용하여 다수의 비트가 한 번에 전송하는 방

식으로서, 전송 속도가 직렬 통신에 비해 빠르다. 주로 2개의 마이크로컨트롤러 간이나 마이크로컨트롤러와 IC 간의 데이터 전송에 사용된다.

직렬 통신은 1개의 데이터 전송용 라인을 이용하여 한 번에 한 비트씩 전송하는 방식으로, 병렬 통신에 비하여 구현하기 쉽고 가격이 싸며 거리 제한이 적은 반면 병렬 통신에 비하여 전송 속도가 느리다는 단점이 있다. 직렬 통신은 주로 마이크로컨트롤러와 PC 간의 데이터 전송에 사용된다.

[그림 1-1](a)는 8개의 데이터 버스(data bus, 마이크로컨트롤러에서는 전송 매체를 데이터 버스라고 표현하기도 한다)를 이용하여 2개의 디바이스가 데이터를 교환하는 병렬 통신 방식이며, [그림 1-1](b)는 1개의 전송 매체를 이용하여 데이터를 교환하는 직렬 통신 방식이다.



[그림 1-12] 직렬 통신과 병렬 통신

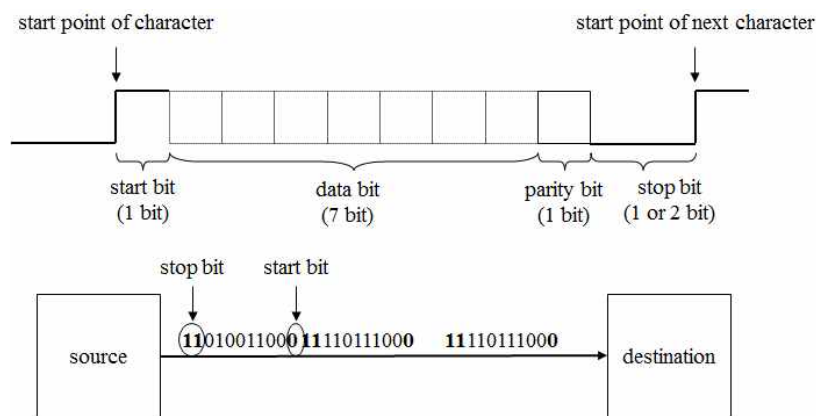
2. 비동기 전송

마이크로컨트롤러와 PC가 직렬 통신을 할 때에는 주로 비동기 전송 방식을 사용한다. 비동기 전송은 하이 레벨과 로우 레벨을 가지고 0과 1을 구분하는 NRZ-L과 같은 클러킹 (clocking) 기능이 내장되지 못한 신호를 이용하여 전송하는 경우에 주로 사용하는 전송 방식이다. 비동기 전송에는 비트 타이밍(bit timing) 문제를 해결하기 위하여 긴 비트열을 보내지 않고, 한 번에 한 문자(character) 단위로 전송하는 방식이다. 이때 각 문자의 길이는

7비트 또는 8비트를 많이 사용한다. [그림 1-2]는 전형적인 비동기 전송의 프레임을 나타내고 있다. 프레임에는 7비트의 데이터와 1비트의 패리티 비트(parity bit), 그리고 1비트의 시작 비트(start bit), 1 또는 2비트의 정지 비트(stop bit)가 포함되어 총 10~11비트의 문자 단위로 전송된다.

비동기 전송에서 전송 매체에는 항상 1의 신호, 즉 0V의 신호가 흐른다. 만약, 송신기가 프레임을 보내려고 할 때에는 시작 비트를 0으로 변화시키고 데이터를 보내게 된다. 반면, 수신기는 전송 매체를 관찰하다가 1에서 0으로 변화될 때 프레임의 전송이 시작되고 있음을 감지하고 데이터를 수신하게 된다. 이러한 방법에 의하여 송신기와 수신기의 동기는 시작 비트에 의하여 재동기화(resynchronization)가 이루어진다.

비동기 전송은 동기 전송에 비하여 매우 간단하고 가격이 저렴하다. 이러한 이유로 비동기 전송은 RS232C, RS485 등 여러 프로토콜에서 채택되고 있다. 그러나 8비트의 데이터 전송에 3비트의 오버헤드가 포함된다는 단점이 있다. 따라서 비동기 전송은 매우 낮은 전송 속도에서만 사용 가능하다.



[그림 1-2] 비동기 전송

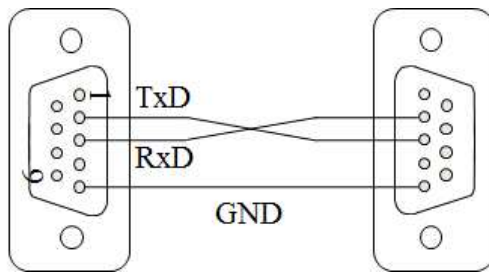
3. RS232C 통신

RS232는 EIA(electronic Industries Association)에서 제정한 시리얼 통신(serial transmission)을 위한 인터페이스 표준이다. RS232에는 A, B, C 세 가지 버전이 있는데, 각 버전들은 1과 0의 전압 차의 범위에 따라 달라진다. 그 중에서 가장 많이 사용되는 것이 C 버전, 즉 RS232C이다.

RS232C는 25핀 커넥터와 9핀 커넥터를 정의하고 있는데, 일반적으로는 [그림 1-3]과 같이 간략화된 9핀 커넥터를 더 많이 사용한다. 9핀 커넥터의 2번은 RxD, 3번은 TxD, 5번은 GND로 PC와 마이크로컨트롤러를 연결할 때 3핀만 연결하면 된다.

RS232C의 전기적 규격에는 전압 레벨이 정의되어 있다. RS232C는 양극성 NRZ-L 인코딩 방식을 사용하고 있다. 또 공통 접지(0V)를 기준으로 -3V와 -12V 사이면 1로 인식하고, +3V와 +12V 사이면 0으로 인식한다. RS232C의 데이터 전송률은 보통 20 Kbps보다 작고,

전송 거리는 15 m 이내이다.



[그림 1-14] RS232C의 케이블 연결

수행 내용 / 로봇 모니터링 프로그램 개발하기

재료 · 자료

- 센서 데이터시트
- 작업에 대한 사용자 요구 사양서
- 로봇의 하드웨어 사양서 및 사용 설명서
- 프로그램 목적에 맞는 주변 환경 데이터
- 프로그램 저장용 저장장치

기기(장비 · 공구)

- 컴퓨터, 프린터
- 프로그램 대상 로봇
- 프로그램 개발용 소프트웨어
- 문서 작성용 소프트웨어
- 프로그램 설치용 장치 및 케이블

안전 · 유의 사항

- 모니터링 툴 개발에서는 사용자 요구 사항을 효과적으로 구현하기 위하여 필요한 정보를 모니터링 범위로 하여야 한다.
- 학습의 이해도 향상을 위하여 특정 모델의 로봇과 모니터링 툴을 사용하지만, 실제 업무에서는 이와 다르다는 것을 미리 이해해야 한다.

수행 순서

① PC용 모니터링 프로그램과 연결하기 위하여 로봇 컨트롤러에 모니터링 프로그램에서 출력할 파라미터 값들을 보내는 프로그램을 구현한다.

1. PC용 모니터링 프로그램에서 출력할 파라미터들을 선정한다.

로봇 컨트롤러에는 다양한 모터와 센서 등이 연결되어 있다. 이러한 모터와 센서 값들 중에서 어떠한 사항들을 모니터링 프로그램에 출력할 것인지를 결정해야 한다.

<표 1-1>은 라인트레이서를 위한 파라미터의 예를 보여주고 있다. 라인트레이서는 두 바퀴로 구동하고, 로봇의 전방에 있는 4개의 초음파센서가 바닥에 있는 라인을 감지하여 방향을 제어한다. 라인트레이서의 경우 다음과 같이 모터의 RPM과 초음파센서의 센서 on/off 값을 모니터링하는 것이 좋다.

<표 1-2> 라인트레이서에서 모니터링해야 할 파라미터

장치	의미	값	비고
모터	회전 속도(RMP)	0 - 360RMP	정수
초음파센서	라인 감지 여부	0 또는 1	이진수

2. 로봇 컨트롤러에서 파라미터 값을 주기적으로 측정한다.

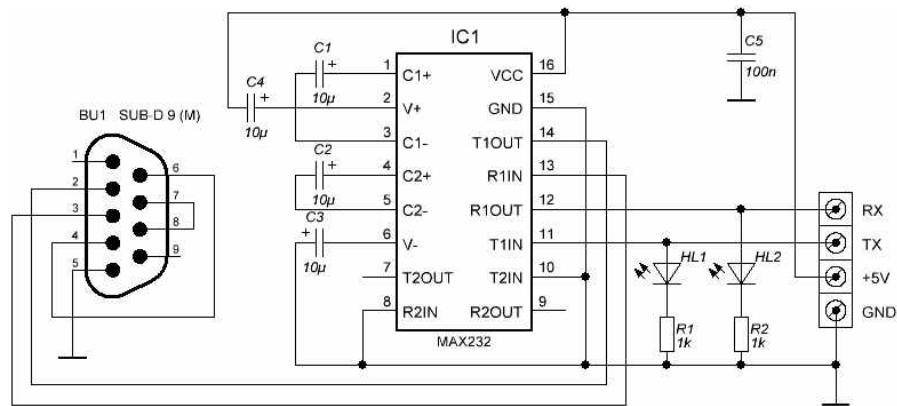
로봇 컨트롤러에 모터와 센서의 값을 주기적으로 측정하는 기능을 구현해야 한다. 모터의 경우에는 엔코더와 카운터를 이용하여 회전 속도를 주기적으로 측정해야 하며, 초음파센서의 경우에는 센서 회로와 ADC를 이용하여 초음파센서의 감지 여부를 주기적으로 측정하여야 한다.

3. 마이크로컨트롤러의 USART 기능을 이용하여 주기적으로 측정한 파라미터 값을 PC용 모니터링 프로그램에 전송할 수 있는 RS232 회로와 시리얼 통신 프로그램을 구현한다.

로봇 컨트롤러와 PC용 모니터링 프로그램을 연결하는 가장 단순한 방법은 시리얼 통신(serial communication)을 이용하는 방법이다. PC용 모니터링 프로그램과 연결하기 위하여 로봇 컨트롤러의 MCU(micro controller unit)에서 동작하는 시리얼 통신 프로그램을 구현하여야 한다.

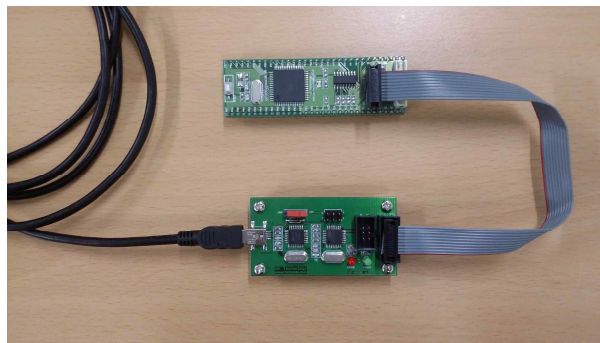
(1) 로봇 컨트롤러에 RS232 회로를 구현한다.

먼저 RS232 회로를 구성하여야 한다. 일반적으로 MCU는 0~5V(저전력의 경우 0~3.3V)의 디지털 값을 많이 사용한다. 그러나 PC에서는 -12V - +12V 디지털 값을 사용하고 있다. 이를 연동하기 위해서는 MCU의 출력레벨과 PC의 출력레벨을 상호 변환해야 한다. 이를 위하여 MAX232와 같은 IC를 사용하여 [그림 1-4]와 같은 RS232 회로를 구성하여야 한다. 특히 최근에 판매되는 PC나 노트북에 시리얼 포트가 없는 경우가 많이 있다. 이런 경우에는 USB 포트에 연결하여 USB 시리얼 포트를 사용하게 하는 [그림 1-5]와 같은 드라이버를 이용하여야 한다.



출처: FAQs System(<http://www.faqssys.info/tag/max232/>). 2016. 8. 8.

[그림 1-15] RS232 회로



[그림 1-5] USB 시리얼 포트를 위한 컨버터

(2) 로봇 컨트롤러에 시리얼 통신 프로그램을 구현한다.

다음으로 로봇 컨트롤러에 시리얼 통신 프로그램을 구현하여야 한다. 시리얼 통신 프로그램은 통신 초기화 모듈, 송신 함수, 수신 함수로 구성된다. 통신 초기화 모듈에서는 통신 속도 등을 설정하여야 한다. 다음은 매우 간단한 시리얼 통신 프로그램을 나타내고 있다.

로봇 컨트롤러의 값을 주기적으로 보내려면 타이머 등을 이용하여 주기적으로 동작하도록 설정하고, 보내야 하는 파라미터 값을 보내는 프로그램을 작성하면 된다.

```

void serial_Init(void);
unsigned char Uart_Getch(void);
void Uart_Putch(unsigned char PutData);
unsigned char Receive_data;

void main(void) {
{
    serial_Init();                // 시리얼 통신을 초기화한다.
    while(1) {
        Uart_Putch( "a" );        // a값을 송신한다.
    }
}

void serial_Init(void) {
{
    SREG = 0x00;                  // 모든 인터럽트 비활성화
    UBRR0H = 0; UBRR0L = 16;      // 57600bps (16MHz)
    UCSRA = (0<<RXC0) | (1<<UDRE0); // 수신·송신 상태비트 초기화
    UCSRB = (1<<RXEN0) | (1<<TXEN0); // 수신·송신 기능 활성화
    UCSRC = (3<<UCSZ00);          // START 1비트/DATA 8비트/STOP 1비트
    DDRE = 0b00000010            // RXD0 핀 입력, TXD0 핀 출력으로 설정
    SREG = 0x80;                  // 모든 인터럽트 활성화
}

void Uart_Getch(void) {          // 시리얼로부터 1바이트 값을 받는 함수
    while(!( UCSRA & (1<<RXC0)) ); // 데이터가 올 때까지 기다린다.
    Receive_data = UDR0;         // 데이터를 받는다.
}

void Uart_Putch(unsigned char Send_Data) { // 시리얼로부터 1바이트 값을 보내는 함수
    while(!( UCSRA & (1<<UDRE0)) ); // 데이터가 빌 때까지 기다린다.
    UDR0 = Send_Data;           // 데이터를 전송한다.
}

```

② 로봇 컨트롤러로부터 송신된 파라미터 값들을 출력할 수 있는 모니터링 프로그램을 개발한다.

1. PC에서 시리얼 포트(COM 포트)를 설정할 수 있는 기능을 구현한다.

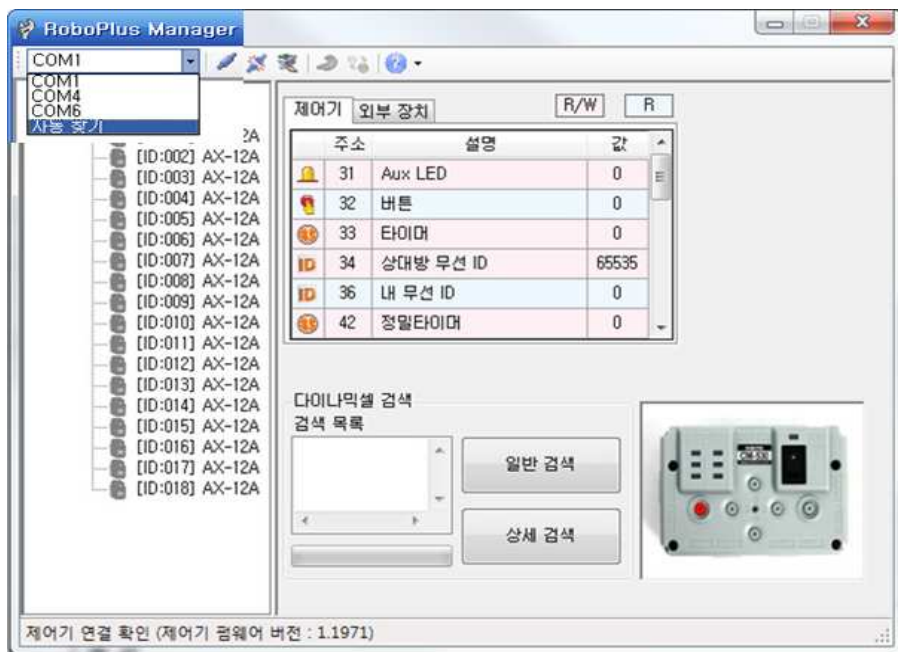
PC에서 시리얼 통신을 하기 위해서는 로봇 컨트롤러와 PC의 시리얼 통신 속도가 일치해야 한다. 또 PC는 시리얼 통신을 지원하는 모듈이 연결되는 경우 자동적으로 특정 COM 포트를 할당하기 때문에 COM 포트를 설정하여야 한다. 따라서 모니터링 프로그램을 개발할 때에는 시리얼 포트를 찾는 프로그램 코드를 만들어 넣어야 한다.

[그림 1-6]은 00사의 로봇 제어기와 PC를 연결하는 장면을 보여주고 있으며, [그림 1-7]은

RoboPlus Manager라는 모니터링 프로그램을 이용하여 COM 포트를 설정하는 방법을 보여준다.



[그림 1-6] 시리얼 포트를 이용한 로봇 제어기와 PC의 연결



[그림 1-7] PC용 모니터링 프로그램에서 시리얼 포트 선택

2. PC에서 로봇 컨트롤러의 파라미터를 출력할 수 있는 프로그램을 구현한다.

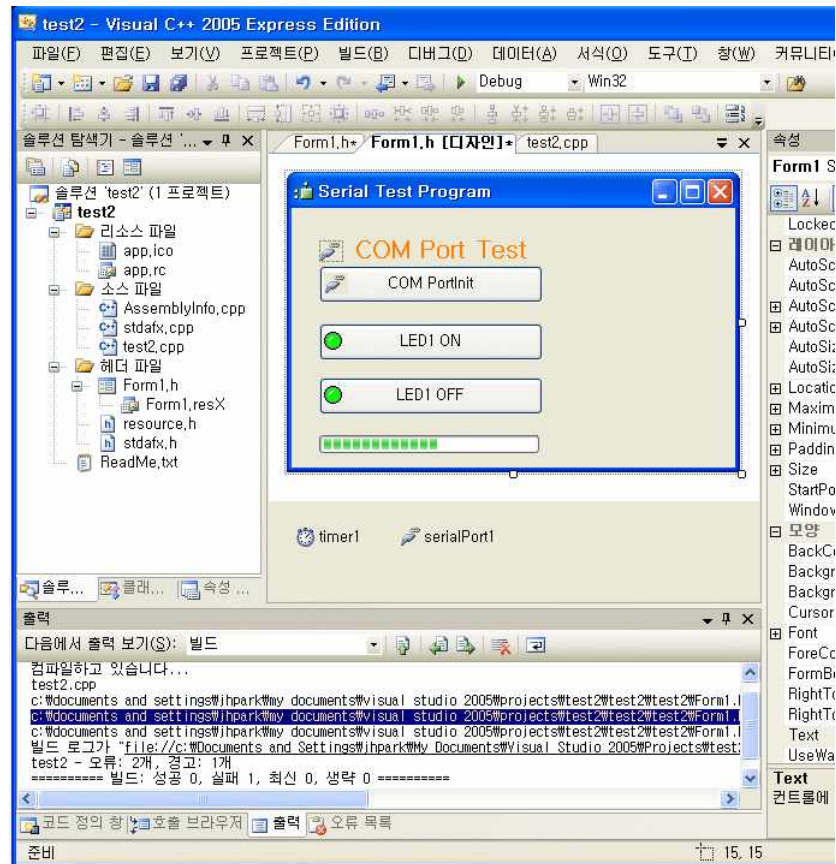
COM 포트를 설정하고 나면, 로봇 컨트롤러와 모니터링 프로그램이 연결된다. 로봇 컨트롤러와 연결되고 나면 다음과 같은 모니터링 프로그램을 구현한다.

첫째, 로봇 컨트롤러로부터 파라미터들을 송신할 수 있는 프로그램 코드를 구현한다.

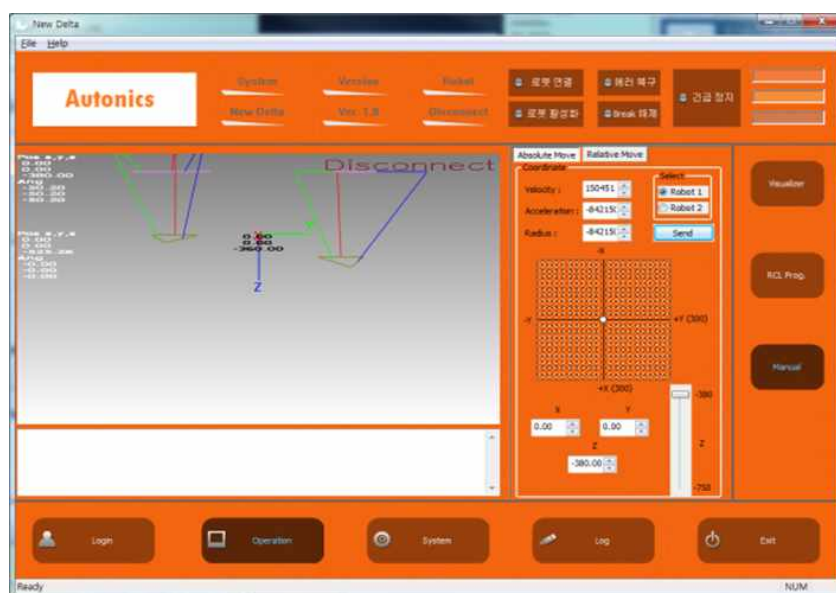
둘째, 수신된 파라미터들을 GUI 화면에 출력하는 프로그램 코드를 구현한다.

일반적으로 Visual C++이나 Visual Basic을 사용하는 경우 시리얼 통신의 초기화, COM 포트의 설정, 데이터 송신 및 수신 등에 관한 API와 GUI를 위한 API를 제공하고 있다. 이러

한 API를 이용하는 경우 [그림 1-8]과 같은 직관적인 모니터링 프로그램을 만들 수 있다. [그림 1-9]는 00사의 PC 기반 상용 로봇 제어기를 나타낸다. 그림에서 로봇과의 상태를 알 수 있으며, 각종 파라미터를 입력하여 로봇의 위치를 제어할 수 있다.



출처: My Story(<http://nexp.tistory.com/82>). 2016. 8. 8. 스크린샷.
[그림 1-8] Visual C++을 이용한 PC용 모니터링 프로그램의 제작 과정 예



[그림 1-9] 상용 PC 기반 로봇 제어기의 예

수행 tip

- 프로그램 개발 도구를 이용하여 프로그램을 개발하기 위해서는 프로그램 언어와 통합 개발 환경에 대한 사용법을 미리 익혀 두어야 한다.
- 대다수의 로봇 완성 업체의 경우 모니터링 프로그램을 제공하기 때문에 직접 모니터링 프로그램을 구현할 필요는 없다.
- 로봇 완성 업체에서 제공하는 모니터링 프로그램을 사용하기 위하여 프로그램 설치 방법에 대해 이해하여야 한다.

학습 1	모니터링 툴 개발하기
학습 2	외부 툴과의 동기화 소프트웨어 개발하기
학습 3	로봇 응용 소프트웨어 개발하기
학습 4	로봇 훈련 소프트웨어 개발하기
학습 5	다수 로봇 제어 소프트웨어 개발하기

2-1. 로봇 원격 조정 툴 개발

학습 목표

- 로봇을 조정하는 데 사용되는 툴의 종류 및 조정 방법에 맞게 프로그램을 설계할 수 있다.
- 로봇과 조정 툴이 동시에 동일한 정보 버전으로 동작할 수 있도록 동기화 프로그램을 할 수 있다.
- 근거리 또는 원격지 유·무선 조정에서 통신 및 전파의 특성을 고려하여 동기화 프로그램을 할 수 있다.
- 조정 툴이 정상적으로 운영되는지 시험 및 테스트할 수 있다.

필요 지식 /

① 로봇 프로그래밍

유용한 작업을 하기 위하여 로봇은 동작 사이클을 수행하도록 프로그래밍되어야만 한다. 로봇 프로그램은 작업 사이클을 지원하는 주변 행위들과 결합된 로봇 팔에 의해 따라가는 공간상의 경로로 정의될 수 있다. 주변 행위들의 예는 그리퍼의 개폐, 논리적 결정 과정의 수행, 로봇 셀의 다른 장치들과의 통신 등을 들 수 있다. 로봇은 명령을 제어기 메모리에 입력함으로써 프로그래밍될 수 있고 서로 다른 로봇은 서로 다른 명령 입력 방법을 사용한다. 오늘날 거의 모든 산업용 로봇이 메모리를 저장장치로 갖는 디지털 컴퓨터를 제어기로 사용하고 있다. 이들 로봇은 세 가지 구별되는 프로그램 방식을 가지고 있다.

1. 교시형 프로그래밍

교시형 프로그래밍의 역사는 컴퓨터 제어가 널리 쓰이기 전인 1960년 초로 거슬러 올라간다. 그 때 사용된 것과 같은 기본 방법들이 오늘날의 많은 컴퓨터 제어 로봇에도 사용된

다. 교시형 프로그래밍에서는 요구되는 동작 사이클을 따라 로봇 팔을 움직이고, 나중의 재연을 위하여 제어기 메모리에 프로그램을 저장함으로써 로봇에게 작업이 가르쳐진다.

(1) 동력 이용 교시와 수동 교시

프로그래밍 교시 과정을 수행하는 데는 동력 이용 교시와 수동 교시의 두 가지 방법이 있다. 이 두 방법의 차이점은 프로그래밍 중에 로봇 팔이 동작 사이클을 따라 움직이게 하는 방법에 있다. 동력 이용 교시는 점간 제어를 하는 재연 로봇의 프로그래밍 방법으로 흔히 사용된다. 이것은 로봇 팔 관절의 움직임을 제어하기 위한 토크 스위치나 접촉 버튼이 있는 티치 펜던트(teach pendant, 손바닥 크기의 제어 상자)를 사용하는 것이다.

[그림 2-1]과 [그림 2-2]는 티치 펜던트의 예를 나타내고 있다. 토크 스위치나 버튼을 이용하여 프로그래머는 로봇 팔이 원하는 위치로 구동되도록 순서대로 조작한다. 그리고 그 위치를 메모리에 기록한다. 차후의 재연 시 로봇은 자체의 동력으로 위치를 연속적으로 통과하면서 움직인다.



[그림 2-1] 티치 펜던트



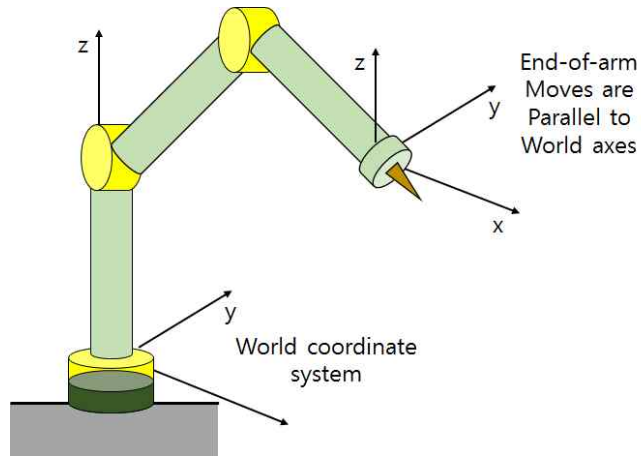
출처: ○○중공업(2016). Hi 제어기 보수 설명서.
[그림 2-2] 티치 펜던트의 설치 예

수동 교시는 스프레이 페인팅과 같은 불규칙 경로 동작 패턴일 때 연속경로 제어를 하는 재연 로봇의 프로그래밍에 편리하다. 이 프로그래밍 방법에서는 조작자가 팔 끝이나 팔에 달려 있는 공구를 붙잡고서 손으로 동작 순서에 따라 움직이면서 그 경로를 메모리에 기록해야 한다. 로봇 팔 자체는 상당한 질량을 가지고 있어 움직이기 어렵기 때문에 특별한 프로그래밍 장치가 교시 과정 동안 실제 로봇을 대신하기도 한다. 이 프로그래밍 장치는 로봇과 같은 관절 배치를 가지고 있고 트리거 핸들(또는 다른 제어 스위치)이 장착되어 있는데, 이 장치는 작업자가 동작을 메모리에 기억시키고자 할 때 사용된다. 동작은 촘촘하게 위치한 점의 연속으로 기억된다. 경로를 재연할 때 앞에서 입력한 것과 동일한 점의 연속을 따라 실제 로봇 팔을 제어함으로써 경로가 다시 생성된다.

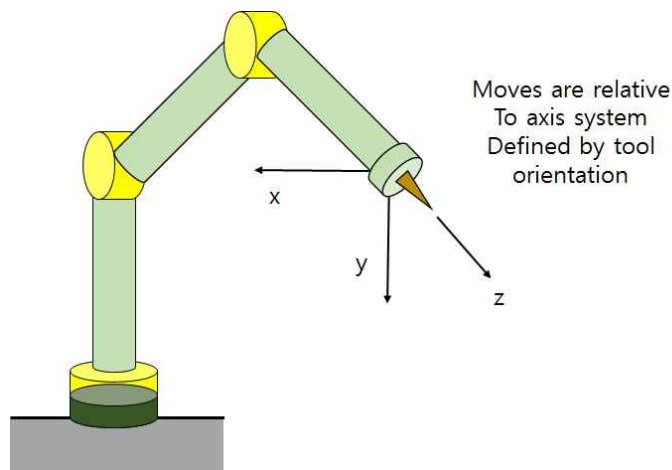
(2) 동작 프로그래밍

교시 펜던트로 각 관절을 조종하는 것은 로봇에게 동작 명령을 입력하는 불편한 방법이 될 수 있다. 예를 들면 다관절 로봇의 각 관절을 조정하여 손끝을 직선 운동시키는 것은 어려운 일이다. 그러므로 동력 이용 교시를 사용하는 많은 로봇이 개별 관절 제어 외에도 두 가지의 방법을 추가로 제공하고 있다. 이들 방법을 이용하여 프로그래머는 로봇의 손목 끝을 직선 경로로 움직이도록 제어할 수 있다.

이 두 가지 방법은 월드좌표계와 공구좌표계이다. 두 좌표계는 모두 직교좌표계를 사용한다. [그림 2-3]과 같은 월드좌표계에서는 기준 원점과 좌표축은 어떤 고정된 위치와 로봇 베이스에 대한 상대적인 위치로 정의된다. [그림 2-4]에서 보여준 공구좌표계에서 좌표계는 손목 면판(faceplate, 엔드이펙터가 붙는 곳)의 자세에 대해 상대적으로 정의된다.



[그림 2-3] 월드좌표계에서의 로봇의 이동



[그림 2-4] 공구좌표계에서의 로봇의 이동

이런 방식에서 프로그래머는 공구를 원하는 방향으로 향하게 할 수 있고, 로봇을 공구에 평행 또는 직각 방향으로 직선운동하도록 제어할 수 있다. 월드좌표계와 공구좌표계는 로봇이 손목 끝을 좌표계 축 중의 하나에 평행한 직선운동을 할 수 있을 때만 유용한 것이다. 직선운동은 직교 로봇에게는 아주 자연스럽지만, 회전관절의 조합을 갖는 로봇에게는 몹시 부자연스럽다. 이런 형식의 관절을 갖는 로봇으로 직선운동을 얻으려면 로봇 제어기에 의해 수행되는 직선 보간 과정이 필요하다. 직선 보간에서 두 점 사이의 직선을 얻기 위해서 제어 컴퓨터는 로봇 손목 끝이 통과해야만 하는 공간상의 점을 계산해야 한다.

로봇이 사용할 수 있는 다른 종류의 보간들이 있다. 직선 보간보다 더 흔한 것이 관절 보간이다. 로봇의 손목 끝이 두 점 사이를 관절 보간을 이용하여 이동하도록 명령받았을 때 로봇은 각 관절을 각자의 등속도로 동시에 움직이게 하는데, 이때 각 관절의 속도는 각 관절의 시작과 종료를 같은 시간에 할 수 있도록 하는 속도이다. 관절 보간이 직선 보간보다 더 좋은 점은 동작하기 위해 필요한 전체 동작 에너지가 더 적게 필요하다는 것이다. 이것은 같은 동작을 더 빨리 할 수 있다는 의미일 수 있다. 직교 로봇

의 경우에는 관절 보간과 직선 보간이 동일한 동작 경로 경과를 준다.

수동 교시 프로그래밍에서는 전혀 다른 보간 방법이 사용된다. 이 경우에 로봇은 수동 교시 프로그램 과정에서 정의된 밀접하게 배치된 점의 연속을 따라야 한다. 사실 이것은 불규칙적인 동작으로 구성된 경로의 보간 과정이다.

로봇의 속도는 티치 펜던트나 제어 패널에 위치한 다이얼이나 다른 입력 도구를 사용하여 제어된다. 작업 사이클의 어떤 동작들은 고속으로 수행되어야 한다(예: 작업 셀에서 상당히 떨어진 곳으로 물건을 옮기는 작업). 반면에 어떤 동작들은 낮은 속도의 작업을 요구한다(예: 정밀도가 높게 부품을 놓는 작업). 속도 제어는 주어진 프로그램이 일단 안전한 느린 속도로 시험된 후에, 생산에 사용되기 위한 더 높은 속도로 운전할 수 있게 해 준다.

(3) 장점과 단점

교시 방법의 장점은 작업자가 쉽게 로봇을 학습시킬 수 있다는 점이다. 로봇 팔을 요구되는 동작 경로를 따라서 움직이게 하여 로봇 프로그램을 작성하는 것은 사람에게 작업 사이클을 가르치는 논리적 방법과 유사하다.

교시 방법 고유의 단점이 몇 가지 있다. 첫째, 정규 생산이 교시형 프로그래밍 과정 동안에는 중지되어야만 한다. 즉, 교시형 프로그래밍은 로봇 셀이나 생산 라인의 작업 중지를 초래한다. 이것의 경제적인 의미는 교시 방법이 상대적으로 긴 생산 기간을 갖는 대상에 사용되어야만 하고 작은 배치 크기에는 부적절하다는 것이다.

둘째, 동력 이용 교시에 사용되는 티치 펜던트와 수동 교시에 사용되는 프로그래밍 장치는 프로그램에 들어가야 할 의사 결정 논리의 사용이 제한받는다. 교시 방법보다 컴퓨터와 같은 로봇 언어를 사용하여 논리적 지침을 만드는 것이 훨씬 더 쉽다.

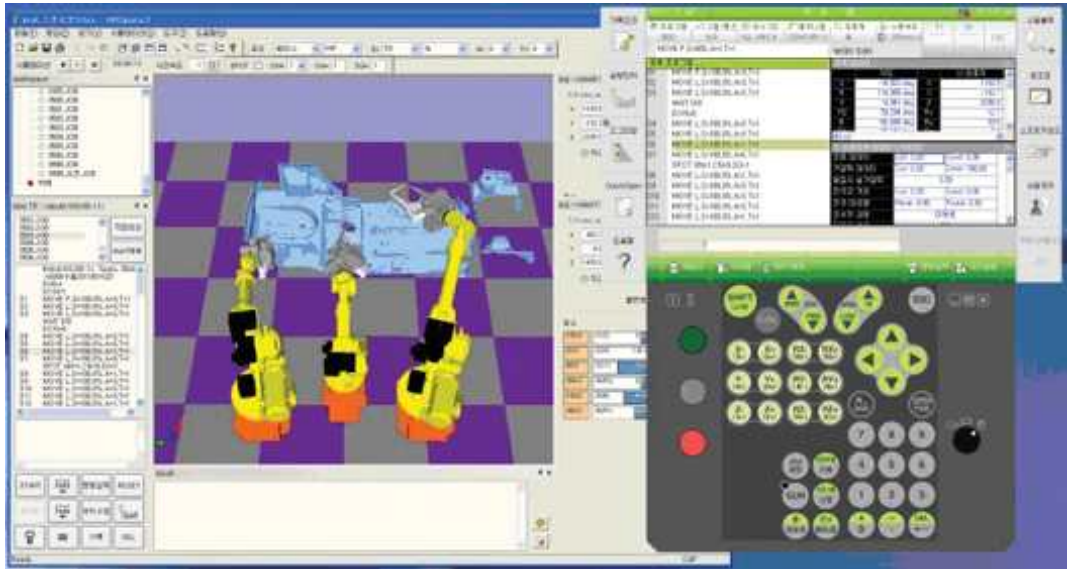
셋째, 교시 방법은 컴퓨터 제어가 로봇에 사용되기 전에 개발되었기 때문에, 이들 방법은 CAD/CAM, 생산 데이터베이스, 지역 통신망과 같은 최신 컴퓨터 기술과 쉽게 호환되지 않는다. 공장의 여러 컴퓨터로 자동화된 하부 시스템에 쉽게 인터페이스되는 능력이 컴퓨터 통합 생산을 이루기 위하여 필요하다.

2. 시뮬레이션과 오프라인 프로그래밍

교시 방법의 문제점은 로봇이 프로그램을 위하여 얼마간의 시간 동안 생산 작업으로부터 제외되어야 한다는 것이다.

[그림 2-5]와 같은 오프라인 프로그램은 로봇 프로그램이 원격지 컴퓨터 터미널에서 준비된 후 실행되기 위하여 로봇 제어기로 다운로드 되는 것이 기능하게 해 준다. 진정한 오프라인 프로그래밍에서는 현재의 프로그래밍 언어에서처럼 실제적인 작업장에서의 위치를 로봇에게 지정해 줄 필요가 없다. 그래픽을 이용한 컴퓨터 시뮬레이션이 오프라인 상에서 작성된 프로그램을 검증하기 위하여 필요하다. 이는 NC 파트 프로그래밍에서 사용되는 오프라인 과정과 비슷하다. 진정한 오프라인 프로그래밍의 장점은 생산을

중단하지 않은 채 새로운 프로그램이 준비되고 로봇에게 다운로드 될 수 있다는 것이다.



출처: 정대상(2012.8.8). “글로벌 로봇 메이커의 시뮬레이션 소프트웨어 길라잡이 - ○○중공업편”, 산업뉴스.
[그림 2-5] ○○중공업의 3D OLP(off-line programming) 시뮬레이터

현재 개발되고 있고 상업적으로 제공되는 오프라인 프로그래밍 과정은 검증과 프로그램 작성을 위한 삼차원 모델을 만들기 위해서 그래픽 시뮬레이션을 사용한다. 로봇 셀은 로봇, 공작 기계, 컨베이어, 그리고 다른 기계로 구성될 수 있다. 시뮬레이션은 이런 셀 구성 요소들이 모니터상에 표시되고, 동영상상을 통해 로봇이 작업 사이클을 수행할 수 있도록 해 준다. 시뮬레이션 과정을 이용하여 프로그램이 개발된 뒤에 셀에서 사용하는 특정 로봇에 맞는 문자 언어로 전환된다. 이것은 NC 파트 프로그래밍의 후처리 과정과 유사한 단계이다.

현재의 상업적 오프라인 프로그래밍 패키지에서는 컴퓨터 시스템의 3차원 모델과 실제 물리적 셀 사이의 형상 차이를 보정하기 위한 작업이 수행될 수 있다. 예를 들면 실제 레이아웃에서의 공작기계 위치는 오프라인 프로그램에서 사용되는 모델과는 약간의 차이가 있을 수 있다. 로봇이 신뢰성 있게 그 기계에 장착과 탈착을 하기 위해서는 로봇이 제어 메모리에 기억될 장착/탈착점의 정확한 위치를 가지고 있어야 한다. 이 보정 모듈은 컴퓨터 모델에서 얻어진 근사값을 실제 셀에서 얻어진 위치 데이터로 대체함으로써 3D 컴퓨터 모델을 보정하는 데 사용된다. 하지만 이 과정을 수행하는 데 시간이 소비된다는 단점이 있다.

수행 내용 / 로봇 원격 조정 툴 사용하기

재료 · 자료

- 센서 데이터시트
- 작업에 대한 사용자 요구 사양서
- 로봇의 하드웨어 사양서 및 사용 설명서
- 프로그램 목적에 맞는 주변 환경 데이터
- 프로그램 저장용 저장장치

기기(장비 · 공구)

- 컴퓨터, 프린터
- 프로그램 대상 로봇
- 프로그램 개발용 소프트웨어
- 문서 작성용 소프트웨어
- 프로그램 설치용 장치 및 케이블

안전 · 유의 사항

- 학습의 이해도 향상을 위하여 특정 모델의 로봇과 제어 툴을 사용하지만, 제어 툴에 따라 사용법이 다르다는 것을 미리 이해해야 한다.

수행 순서

① 티치 펜던트를 이용하여 로봇을 조정한다.

1. 티치 펜던트의 주요 기능들을 이해한다.

[그림 2-6]은 00사의 티치 펜던트의 외형을 나타낸다. 티치 펜던트에는 시작, 정지, 긴급 정지, 모터 ON, 원격/자동/수동 모드 선택 버튼 등이 있으며, 로봇을 이동시킬 수 있는 X, Y, Z축을 위한 축 조작 키, 좌표계를 선택할 수 있는 좌표계 버튼, 유닛과 메커니즘을 선택할 수 있는 버튼, 프로그램을 입력하기 위한 버튼 등이 있다.



출처: ○○중공업(2008). Hi 제어기 보수 설명서.
[그림 2-6] 티치 펜던트의 기능

[그림 2-7]은 티치 펜던트의 화면을 나타낸다. 티치 펜던트 화면은 컬러 터치스크린으로 10개의 화면 창으로 구성되어 있다. 티치 펜던트를 사용하기에 앞서 화면에 있는 각 기능들을 이해해야 한다.



출처: ○○중공업(2008). Hi 제어기 보수 설명서.
[그림 2-7] 티치 펜던트의 초기 화면

(1) 상태 표시줄에 표시된 기능에 대해 이해한다.

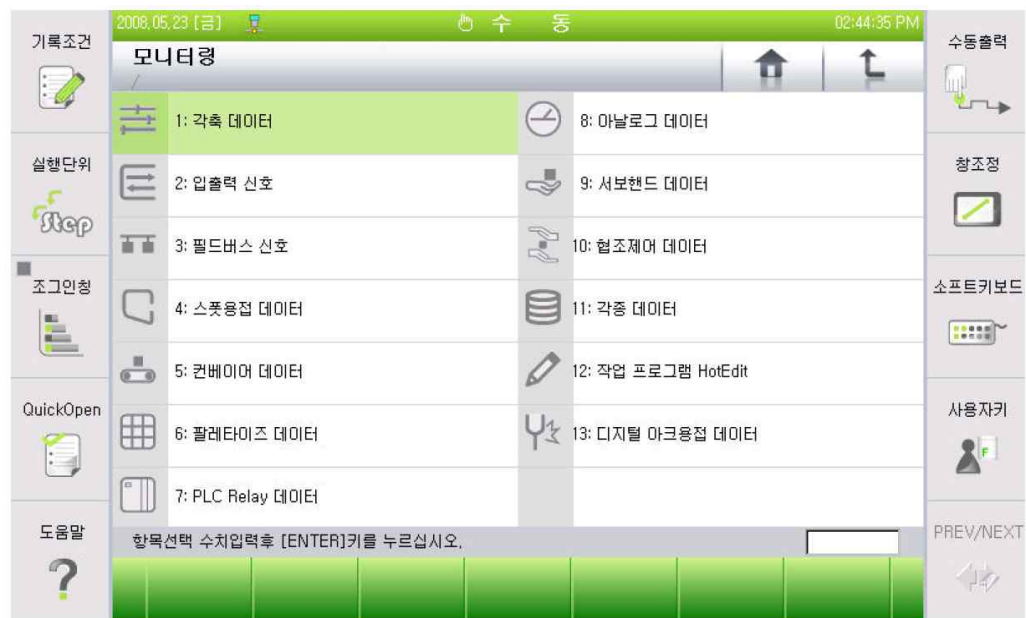
티치 펜던트의 상태 표시줄에는 로봇 운전을 위한 각종 상태 값이 표시된다. 프로그램은 현재 선택되어 있는 작업 프로그램 번호를 표시하며, 프로그램이 존재할 경우에 화면 편집 창에 기록되어 있는 프로그램이 열리게 된다. 유닛은 사용자가 구성한 메커니즘의 조합 상태를 표시하며, 스텝/평선은 현재 선택된 프로그램 내의 스텝이나 평선의 번호를 나타낸다. 메커니즘은 로봇 타입이나 선택한 부가 축 번호를 표시한다. 좌표계는 수동 운전을 위해 선택된 로봇의 좌표계의 상태를 표시하는데, 축, 직교, 툴 좌표계의 순으로 상태 표시가 변경된다.

(2) 모니터링 창에 표시되는 데이터에 대해 이해한다.

티치 펜던트의 모니터링 창에는 각 축별 위치 데이터, I/O 데이터, 각 응용별 상태 데이터가 실시간으로 표시된다.

(3) 내용 선택을 클릭한 경우의 화면에 대해 이해한다.

내용 선택 버튼을 클릭하면, [그림 2-8]과 같은 현재 선택한 창에 표시할 모니터링할 정보를 지정할 수 있는 화면이 나타난다.



출처: ○○중공업(2008). Hi 제어기 보수 설명서.

[그림 2-8] 티치 펜던트의 모니터링 창

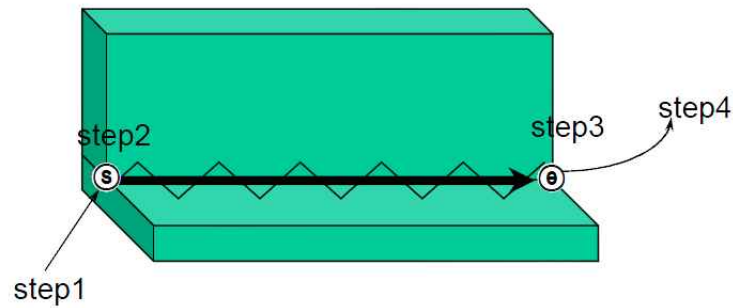
2. 아크 용접 작업을 위한 티치 펜던트 프로그램을 작성해 본다.

(1) 티칭을 하기 위한 작업 대상을 결정한다.

[그림 2-9]는 티치 펜던트를 이용하여 아크 용접을 티칭하기 위한 용접 작업의 예를 나타내고 있다.

㉓ :시작점

㉔ :종료점

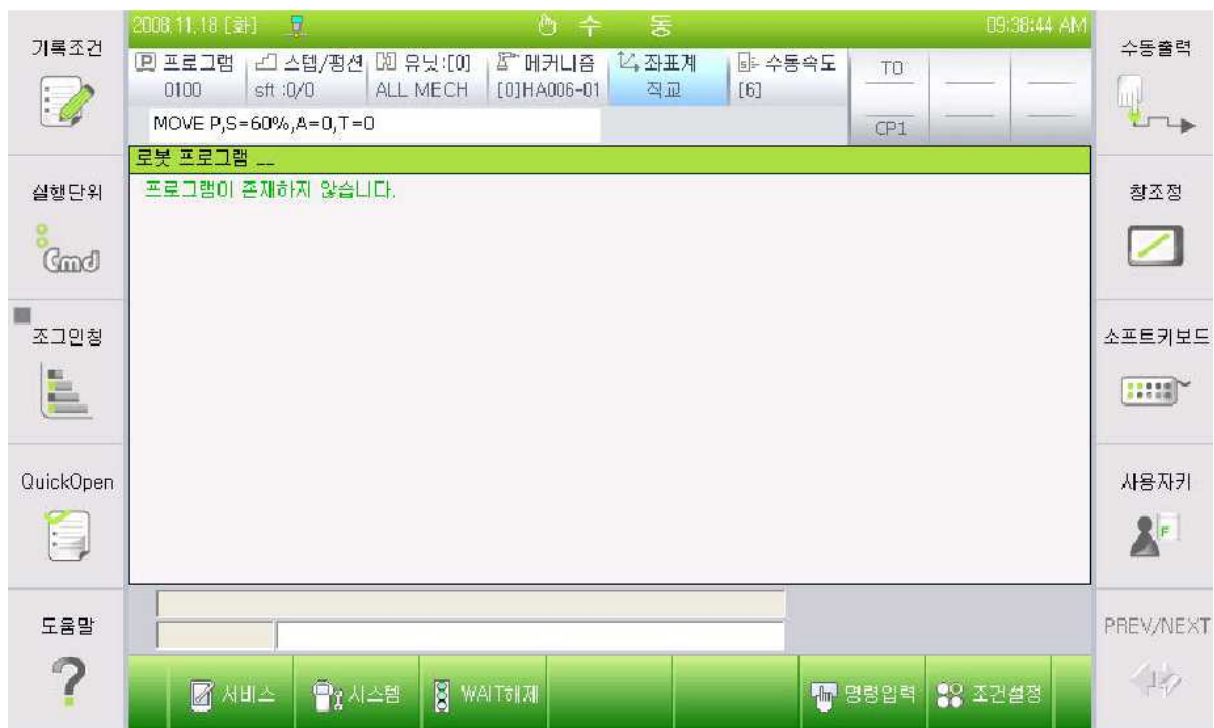


출처: ○○중공업(2009). Hi 제어기 기능 설명서-아크 용접
[그림 2-9] 티치 펜던트를 이용한 아크 용접의 예

아크 용접을 위한 티치 펜던트 조작 절차는 다음과 같다.

(가) 티치 펜던트 전면부의 전원 스위치를 켜고, 모드 스위치를 수동 모드로 선택한다. 그리고 나서 프로그램 버튼을 누른 후 프로그램 번호를 지정한다. 예로 키 패드를 이용하여 0100과 같이 지정한다.

(나) 모터 ON 버튼을 누르면 [그림 2-10]과 같은 화면이 나타난다.

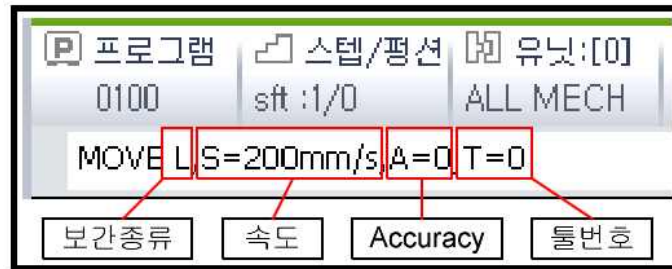


출처: ○○중공업(2009). Hi 제어기 기능 설명서-아크 용접
[그림 2-10] 티치 펜던트의 시작 화면

(다) 축 조작 버튼을 조작하여 로봇의 끝점을 step 1 위치로 이동시킨다.

(라) 티치 펜던트 화면의 기록조건 버튼을 누른 후 [그림 2-11]과 같이 원하는 보간

종류, 속도, 정밀도, 톨 번호를 지정한다. 항목 간의 이동은 방향키를 이용하여 설정하고, 톨 번호는 톨 버튼을 누른 후 원하는 톨 번호를 입력할 수 있다.



출처: ○○중공업(2009). Hi 제어기 기능 설명서-아크 용접
[그림 2-11] 기록 조건 표시 내용

(마) (라)단계에서 로봇의 명령어를 설정한 후 기록 버튼을 누르면, [그림 2-12]와 같이 프로그램 창에 step 1을 위한 로봇의 기능(로봇을 step 1의 위치로 움직이라는 명령어)이 기록된다.



출처: ○○중공업(2009). Hi 제어기 기능 설명서-아크 용접
[그림 2-12] 첫 번째 명령어의 입력

(바) (다), (라), (마)단계를 반복하여 step 1에서 step 4까지 로봇을 이동시키기 위한 로봇 프로그램을 [그림 2-13]과 같이 작성한다.



출처: ○○중공업(2009). Hi 제어기 기능 설명서-아크 용접
[그림 2-13] 용접 지점으로의 이동 작업을 위한 터치 펜던트 프로그램

(사) 용접 작업을 프로그래밍하기 위하여 step 2로 커서를 이동시킨다. 사용자키 버

튼을 눌러 화면 하단에 등록된 메뉴들이 나타나게 한 후, WEAVON 메뉴를 선택하여 패턴 번호를 입력한다. 다음으로 사용자 버튼을 누른 후 화면 하단에 있는 ARCON 메뉴를 선택하여 패턴 번호를 입력한다.

- (아) 아크가 종료되는 지점인 step 3으로 커서를 이동시킨 후, 사용자 키를 눌러 ARCOF 메뉴를 선택하여 명령을 입력한다. 또 WEAVOF 메뉴를 선택하여 WEAVOF 명령문을 입력한다. 마지막으로 FLOW 제어 버튼을 이용하여 END 명령을 입력한다. [그림 2-14]는 step 1로 로봇의 이동 작업과 용접 작업에 관한 명령어를 포함한 로봇 프로그램을 나타낸다.

The screenshot displays a robot programming software interface. The main window shows a sequence of steps (S1, S2, S3, S4) with their corresponding commands. A table on the right shows the current coordinates (X, Y, Z) and joint angles (R1, R2, R3) for each step.

각축	각도	좌표치
S	15.086 deg	X 928.6
H	71.711 deg	Y 242.1
V	-17.338 deg	Z 536.6
R2	-5.760 deg	Rx 177.7
B	-52.104 deg	Ry 4.6
R1	92.324 deg	Rz 106.1

출처: ○○중공업(2009). Hi 제어기 기능 설명서-아크 용접
[그림 2-14] 아크 용접 작업을 포함한 티치 펜던트 프로그램

수행 tip

- 티치 펜던트 프로그램을 하는 경우, 티치 펜던트 제조사에 따라서 프로그래밍 방법이 매우 차이가 난다. 따라서 프로그램을 하기 전에 매뉴얼을 충분히 학습하여야 한다.

학습 1	모니터링 툴 개발하기
학습 2	외부 툴과의 동기화 소프트웨어 개발하기
학습 3	로봇 응용 소프트웨어 개발하기
학습 4	로봇 훈련 소프트웨어 개발하기
학습 5	다수 로봇 제어 소프트웨어 개발하기

3-1. 로봇 원격 모니터링 앱 개발

학습 목표

- 로봇을 이용한 다양한 기능을 구현할 수 있는 프로그램을 설계할 수 있다.
- 로봇이 웹, 앱 등의 프로그램과 연계하여 운영될 수 있도록 프로그램 할 수 있다.
- 로봇과 함께 활용되는 주변장치의 성능을 활용하는 프로그램을 구현할 수 있다.
- 프로그램을 설치하고 배포할 수 있다.

필요 지식 /

① 블루투스 통신

블루투스(영어: Bluetooth)는 1994년에 에릭슨이 최초로 개발한 개인 근거리 무선 통신을 위한 산업 표준이다. 1999년 5월 20일에 공식적으로 발표되었다. 블루투스라는 이름은 10세기 덴마크와 노르웨이의 국왕 하랄드 블라톤의 이름을 영어식으로 바꾼 것이다. 제안을 한 사람은 JIM KARDACH인데, 계기는 소설가 프란스 G. 벵 트손의 ‘바이킹’이라는 소설과 하랄드 블라톤의 관한 역사 소설 ‘The Long Ships’를 읽고 있어서 제안했다. 하랄드 블라톤이 스칸디나비아를 통일한 것처럼 무선통신도 블루투스로 통일하자는 의미인 것이다. 1998년에는 Bluetooth Special Interest Group이라는 블루투스에 관한 동맹이 결성되면서 에릭슨을 비롯한 인텔, 레노버, 마이크로소프트, 애플, 노키아, 도시바, IBM이 참여하였다. 블루투스는 다양한 기기들이 안전하고 저렴한 비용으로 전 세계적으로 이용할 수 있는 무선 주파수를 이용해 서로 통신할 수 있게 한다. 또 IEEE 802.15.1 규격을 사용하는 블루투스는 PANs의 산업 표준 중 하나이다. 블루투스는 ISM 대역인 2.45 GHz를 사용한다. 버전 1.1과 1.2의 경우 속도가 723.1 kbps에 달하며 버전 2.0의 경우 EDR을 특징으로 하는데, 이를 통해 2.1 Mbps의 속도를 낼 수 있다. 블루투스는 유선 USB를 대체하는 개념이며, 와이

파이는 이더넷을 대체하는 개념이다. 암호화에는 SAFER을 사용한다. 장치끼리 믿음직한 연결을 성립하려면 키워드를 이용한 페어링이 이루어지는데, 이 과정이 없는 경우도 있다.

1. 사양과 특징

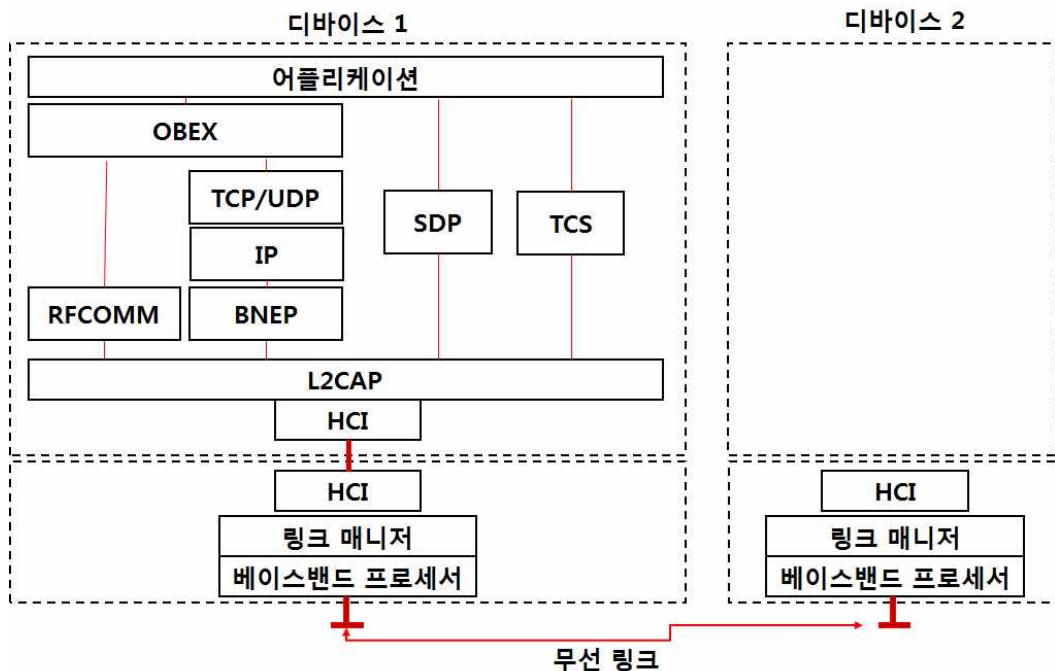
(1) 1.X

가장 초기에 나온 1.0부터 2002년에 등장한 1.1을 거쳐 1.2까지 개선되었다. 다만 최대 전송 속도가 723 Kbps라서 대용량 데이터를 전송하기에는 조금 부적절했었다.

(2) 2.X + EDR (enhanced data rate)

2004년 10월에 [그림 3-1]과 같은 2.0이 표준화되었으며, 기존의 최대 전송 속도였던 721 Kbps를 3 Mbps로 끌어올렸다. 다만 실제 최대 전송 속도는 2.1Mbps 정도 된다. 2007년 7월 26일에 2.1이 발표되었으며, 2.0과의 가장 눈에 띄는 차이는 페어링이 더 손쉽게 가능하도록 SSP(secure simple pairing) 기능이 추가된 것이다. 그 외 커넥션 시 필터링이 쉽도록 EIR(extended inquiry response)이 강화되고, Low Power 모드에서 소비전류를 줄이는 기능이 추가되었다.

안드로이드를 사용한 대다수의 초기 스마트폰과 태블릿 컴퓨터, 게임기와 노트북 등에 탑재되어 있는 버전이다.



[그림 3-1] 블루투스 BR/EDR (버전 1.0, 2.0) 프로토콜 스택

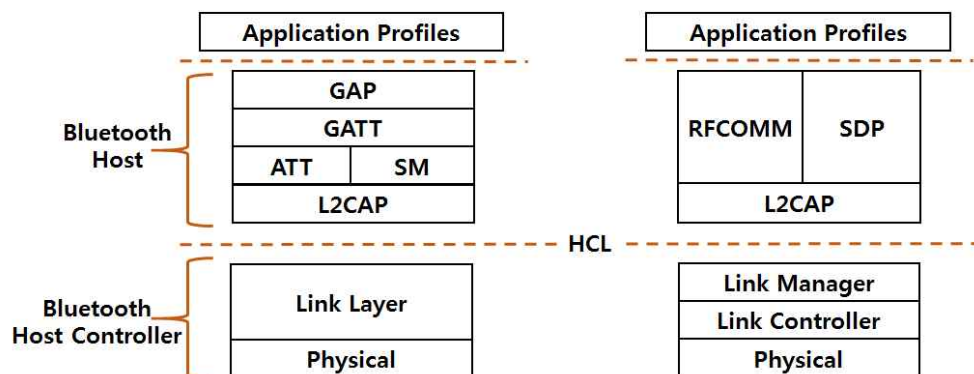
(3) 3.0 + HS(high speed)

2009년 4월 21일에 발표되었으며 이론적으로 24 Mbps이라는 전작에 비해 매우 빠른

속도를 제공한다. 다만 3.0 + HS(high speed)만 24 Mbps를 지원하며, 지원하지 않으면 2.X와 똑같은 속도를 낸다. Bluetooth Link는 접속에만 관여하고, 실제 고속 데이터 통신은 802.11 Wi-Fi 쪽에 추가된 PAL(protocol adaptation layer)를 이용하기 때문이다.

(4) 4.0

[그림 3-2]와 같은 4.0은 2010년 6월 30일에 채택되었으며, Bluetooth Low Energy(BLE)라는 새로운 프로토콜이 추가되어 기존 버전들보다 전력을 적게 소모하는 것이 장점이다. 기술에 따라 3가지로 나뉘지는데, 클래식 블루투스(Classic Bluetooth)는 1.0부터 2.1로 이어져온 기존 블루투스 기술이고, 고속 블루투스(Bluetooth High Speed)는 3.0에서 더해진 Wi-Fi를 활용한 HS 고속 전송 기술의 연장이고, 저전력 블루투스(Bluetooth Low Energy)는 전력 소모를 최소화하고 배터리 수명을 연장하는데 중점을 둔 새로운 표준이다. 그 중 저전력 블루투스(Bluetooth Low Energy)는 전지로 몇 년을 지속할 수 있는 주변 기기가 주요 타겟이기 때문에, 속도는 다른 무선 전송 규격보다 상대적으로 느린 편이다. BLE만 지원되는 칩은 Single Mode라고 부르고, 단방향 전송만 지원되고 칩이 탑재된 제품을 Bluetooth SMART라고 부르며, 앞서 설명한 클래식 블루투스(Classic Bluetooth)와 함께 들어있는 칩은 Dual Mode라 부르며, 양방향 전송이 지원되며 탑재된 제품을 Bluetooth SMART READY라고 부른다. 심장 박동 검사기 같은 주변 기기는 Single Mode 솔루션이 탑재되고, 스마트폰 등에는 Dual Mode 솔루션이 탑재된다. 각각 1Mbps와 3Mbps를 지원한다. HS를 지원한다면 24Mbps도 지원한다.



[그림 3-2] 블루투스 BLE 프로토콜 스택과 BR/EDR 프로토콜 스택의 비교

(5) 4.1

2013년 12월 4일에 블루투스 4.1이 발표되었으며, 주요 특징은 다음과 같다.

(가) 공존성(coexistence) 향상

블루투스와 LTE 무선이 서로 통신 상태를 조정해 가까운 대역폭으로 인한 간섭 현상을 줄여준다.

(나) 더 나은 연결(better connections)

블루투스 연결 장치끼리의 거리가 멀어져 잠시 연결이 끊어지게 되면, 블루투스 4.1 장치는 거리 내로 되돌아올 시 자동으로 재연결된다.

(다) 데이터 전송 개선(improved data transfer)

블루투스를 사용하는 액세서리 장치(헬스 기구) 등과의 통신 전송 상태를 보다 효율적으로 개선하였다.

(라) 개발자에게 더 많은 유연성 제공(more flexibility to developers)

앞으로 있을 웨어러블 기기 붐에 대비한 업데이트로, 블루투스 연결을 통해 웨어러블 기기가 스마트폰의 주변장치이자 동시에 다른 장치와의 허브 역할도 할 수 있게 해 준다. 또 장래를 위해 사물 인터넷(the internet of things)을 위한 새로운 IPv6 사용 표준도 들어가 있다. 또 128비트 AES 암호화가 추가되어 보안성이 증가되었다.

(6) 4.2

2014년 12월 4일에 블루투스 4.2가 발표되었다. 블루투스 4.2의 핵심 업데이트 내용은 크게 세 가지로 요약된다.

(가) 더욱 빨라진 전송 속도

블루투스 SIG의 보도 자료에 따르면, 블루투스 4.2 버전은 기존 4.0 규격 대비 전송 속도가 2.5배 증가했다. 특히, 한 번에 보낼 수 있는 패킷 용량이 10배로 늘어나 전송 오류와 배터리 소비를 동시에 잡았다. 그래프만 보면 250배나 빨라진 것으로 보일 수 있으나 데이터를 주고받는 빈도를 조금 줄이는 대신 한 번에 더 많은 정보를 전송해 전반적으로 2.5배의 전송 속도 향상과 효율적인 데이터 전송을 한다.

(나) 사물 인터넷을 위해 연결성 강화

IPv6나 6LoWPAN을 통해 인터넷에 직접 접속할 수 있다. 블루투스 4.2와 함께 올해 말 승인 예정인 'IPSP(Internet Protocol Support Profile)' 기술이 채택되었기 때문인데, 기존 IP 인프라를 갈아엎을 필요 없이 사물인터넷 기기나 스마트 디바이스가 인터넷에 직접 접속할 수 있어 이전보다 좀 더 유연하게 스마트홈을 구현할 수 있게 될 것으로 예상된다.

(다) 개인 정보 강화

사용자의 허락 없이 블루투스 기기 위치를 추적할 수 없게 했는데, 예를 들어 애플의 아이비콘(iBeacon) 기술을 도입한 매장을 방문하더라도 사용자가 허용하지 않는 한 매장이 사용자의 위치를 마음대로 추적할 수 없게 된다. 미 IT매체 아스텍니카가 취재한 바에 따르면 2.5배의 데이터 전송속도 향상은 새로운 하드웨어가 필요하지만, 개인정보보호는 기존 블루투스 기기에 대한 펌웨어 업데이트만으로도 대응할 수 있게 될 예정이다.

(7) 5.0

2016년 6월 17일에 공개되었으며, 2016년 말에서 2017년 초에 정식으로 출시될 예정이다. 기존과 비교해 전송 범위는 4배, 속도 2배, 비연결 데이터 브로드캐스트 용량은 8배가 향상되었다.

② 스마트폰 앱 프로그래밍

스마트폰 애플리케이션(이후 앱으로 지칭)은 네이티브 앱, 웹 앱, 하이브리드 앱으로 나눌 수 있다.

1. 네이티브 앱

네이티브 앱은 모바일 기기에 직접 다운로드하여 로컬(기기)에 저장되는 바이너리 실행 파일이다. 사용자가 직접 설치하거나 경우에 따라 기업의 관리자가 설치할 수 있다. 네이티브 앱은 애플사의 앱스토어(App Store), 안드로이드의 마켓플레이스(Marketplace), 블랙베리사의 앱 월드(App World) 등과 같은 공용 앱스토어에서 다운로드하면 되나, 모바일 공급업체에서 직접 제공하는 경우도 있다.

모바일 기기에 앱을 설치한 후에는 기기에서 제공하는 다른 서비스처럼 사용자가 앱을 실행한다. 초기화할 경우 네이티브 앱 인터페이스는 중개 요소나 컨테이너 없이 모바일 운영체제와 직접 통신한다. 네이티브 앱은 모바일 OS에서 제공하는 모든 API에 액세스할 수 있으며, 대부분 해당 모바일 OS의 고유 기능을 활용한다.

네이티브 앱을 만들려면 개발자가 사람이 읽을 수 있는 형식으로 소스 코드를 작성하고, 이미지, 오디오, 다양한 OS별 선언 파일 등의 리소스를 만들어야 한다. 이러한 리소스를 개발한 후 모바일 OS에서 제공하는 툴을 이용해 소스 코드를 컴파일하고 배포할 수 있는 바이너리 형태의 실행 파일을 만든다.

이러한 툴 및 기타 유틸리티/파일을 일반적으로 모바일 OS의 소프트웨어 개발 키트(SDK)라고 한다. 네이티브 앱 개발 과정은 모바일 운영체제 특성을 타지 않고 모두 비슷하지만, SDK는 OS별로 다르며, 모바일 OS별로 고유 툴을 제공한다. <표 3-1>은 주요 모바일 OS에서 제공하는 툴, 언어, 패키징 형식, 앱 배포 방법을 보여준다.

모바일 OS에서 제공하는 언어, 툴 등이 다르다는 점이 네이티브 앱 개발 방식의 가장 큰 단점이다. 예를 들어 애플 iOS용으로 작성된 코드를 안드로이드에서는 사용할 수가 없다. 그러므로 멀티 플랫폼을 지원하는 네이티브 앱을 개발하려면 개발 기간도 많이 걸리고 비용도 많이 든다.

이런 단점에도 불구하고 기업들이 네이티브 방식을 선호하는 이유는 애플리케이션 프로그래밍 인터페이스(이하 API) 때문이다. 모바일 기기에 네이티브 앱을 설치하고 실행하면, 앱은 모바일 운영 체제에서 제공하는 전용 API를 호출해 모바일 운영체제와 통신한다. 모바일 운영 체제에서 제공하는 API는 로우 레벨(low level) API와 하이 레벨(high level) API로 구분된다.

(1) 로우 레벨 API

앱은 로우 레벨 API를 호출해 터치스크린이나 키보드와 직접 입출력을 하고, 그래픽을 렌더링하며, 네트워크를 연결한다. 또 마이크를 통해 수신한 오디오를 처리하고, 스피커나 헤드폰을 통해 사운드를 재생하며, 카메라로부터 이미지나 동영상 등을 수신해 저장한다. GPS(global positioning system) 액세스, 방향 정보 수신, 반도체 디스크 또는 상용 하드웨어의 파일 읽기 및 쓰기 등도 로우 레벨 API 호출로 이뤄진다.

(2) 하이 레벨 API

모바일 운영 체제는 로우 레벨 하드웨어 액세스 서비스 이외에, 개개인의 사용자 경험에 영향을 미치는 하이 레벨 서비스도 제공한다. 하이 레벨 서비스는 웹 검색, 일정 및 연락처 관리, 사진 앨범 관리 및 전화 또는 문자 메시지 송수신 기능 등을 일컫는다. 대부분의 모바일 OS는 이러한 서비스를 수행할 수 있는 일련의 애플리케이션을 내장하고 있지만, 네이티브 앱에서는 퍼블릭하게 공개된 하이 레벨 API를 불러 사용할 수도 있다. 다운로드가 가능한 앱은 다른 API를 통해 OS 업체가 제공하는 푸시 알림이나 인앱(In App) 구매와 같은 다양한 클라우드 기반의 서비스에 액세스할 수 있다.

(3) 그래픽 사용자 인터페이스(GUI) 킷

OS 업체에서 제공하는 또 다른 중요한 API 세트는 GUI 킷이다. 주요 모바일 OS는 버튼, 입력 필드, 슬라이더, 메뉴, 탭 바, 대화 상자와 같은 OS 고유의 사용자 인터페이스 구성 요소를 제공한다. 이러한 구성 요소들은 해당 모바일 OS의 기능이 그대로 전이되므로 사용자가 앱을 쉽게 사용하고 즐길 수 있다.

각각의 모바일 플랫폼은 고유의 사용자 인터페이스(UI) 구성 요소를 가지고 있다. 따라서 여러 운영 체제에서 작동하는 앱을 설계하려면 설계자가 각 OS의 서로 다른 UI 구성 요소에 대해 잘 알고 있어야 한다.

API가 OS별로 다르므로 여러 개의 네이티브 앱을 개발하려면 개발시간도 많이 걸리고 비용도 많이 들지만, 모바일 기기가 제공하는 모든 기능을 제대로 활용하는 모바일 앱을 개발하기 위해서는 API 활용은 반드시 필요하다.

<표 3-1> 주요 모바일 OS에서 제공하는 툴, 언어, 패키징 형식, 앱 배포 방법

	애플 IOS	안드로이드	윈도우 폰
언어	Objective-C, C++	Java(일부 C, C++)	C#, VB.NET 등
툴	Xcode	Android SDK	Visual Studio
패키징 형식	.app	.apk	.xap
앱스토어	Apple App Stor	Google Play	Windows Phone Market

출처: IBM 소프트웨어(2014). 모바일 앱 개발 방식 비교: 네이티브, 웹, 하이브리드.

2. 모바일 웹 앱

최신 모바일 기기는 HTML5의 최신 기능, Cascading Style Sheets 3(CSS3) 및 고급 JavaScript를 지원하는 브라우저로 구성되어 있다. 이러한 최신 기능과 함께 HTML5는 페이지 정의 언어에서 브라우저 기반의 풍부한 기능을 가진 애플리케이션을 개발하는 표준으로 자리잡고 있다.

HTML5는 유려한 UI 구성 요소, 다양한 미디어 형식 액세스, 위치 정보 서비스, 네트워크가 끊긴 상태에서도 오프라인으로 사용할 수 있는 다양한 기능을 지원한다. 이러한 HTML5 기술과 현재 개발 중인 많은 기능을 사용하여 개발자들은 웹 기술만으로 고품질의 애플리케이션을 만들 수 있다.

모바일 웹 앱 개발을 위하여 dojox.mobile, Sencha Touch, jQuery Mobile 등의 JavaScript 툴킷이 개발되었다. 이러한 툴킷은 네이티브 앱과 같은 사용자 인터페이스를 생성한다. 이들 모두 모바일 디바이스의 브라우저에서 실행되고 모바일 브라우저에서 지원하는 JavaScript, CSS 및 HTML5 기능을 활용한다.

3. 하이브리드 앱

하이브리드 방식은 네이티브 개발과 웹 기술을 혼합한 것이다. 이 방식을 사용하면 개발자가 모바일 플랫폼과 상관없이 웹 기술로 애플리케이션을 개발할 수 있으며, 필요 시 네이티브 API에 직접 액세스할 수 있다.

애플리케이션의 네이티브 부분은 운영체제 API를 사용하여 브라우저와 디바이스 API 간에 연결고리 역할을 하는 임베디드 HTML 렌더링 엔진을 만든다. 이 연결고리를 통해 하이브리드 앱은 최신 디바이스가 제공하는 모든 기능을 활용할 수 있다.

앱 개발자는 이 기능을 직접 코딩하거나 PhoneGap과 같은 기존 솔루션을 활용할 수 있다. PhoneGap은 모든 모바일 운영체제에 일관된 단일 JavaScript 인터페이스를 제공하는 오픈 소스 라이브러리이다.

앱의 네이티브 부분은 독립적으로 개발할 수 있지만, 일부 솔루션들은 네이티브 컨테이너를 제품의 구성 요소로 제공하기도 한다. 따라서 개발자는 웹 언어만을 사용하여 모든 디바이스 기능을 활용하는 고급 애플리케이션을 개발할 수 있다. 경우에 따라 개발자가 기업의 고유한 요구 사항에 따라 네이티브 컨테이너를 직접 지정할 수 있도록 허용하는 솔루션도 있다.

4. 세 가지 개발 방식 비교

<표 3-2>는 세 가지 개발 방식을 비교 요약한 것이다.

네이티브 방식은 성능과 디바이스의 액세스 가용성을 높이지만 높은 비용과 업데이트 문제가 단점이다. 웹 방식은 상대적으로 간단하고 비용이 적게 들며 업데이트가 쉽지만, 기능상 제한이 따르므로 네이티브 API 호출을 사용하여 제공할 수 있는 우수한 사용자 환경을 제공하기 어렵다. 하이브리드 방식은 이 두 방식의 절충안으로, 여러 기업들이 선택하

고 있으며 특히, 여러 운영체제를 한 번에 지원해야 할 경우 선호되는 방식이다.

<표 3-2> 주요 모바일 OS에서 제공하는 툴, 언어, 패키징 형식, 앱 배포 방법

특성	네이티브 앱	하이브리드 앱	웹 앱
개발 언어	네이티브 전용	네이티브 및 웹 또는 웹 전용	웹 전용
코드 이식성 및 최적화	없음	높음	높음
디바이스 고유의 기능 액세스	높음	중간	낮음
기존 지식 활용도	낮음	높음	높음
유리한 UI	높음	중간	중간
업그레이드 유연성	낮음(항상 앱스토어 이용)	중간(일반적으로 앱스토어 이용)	높음
설치 경험 수준	높음(앱스토어에서)	높음(앱스토어에서)	중간(모바일 브라우저 이용)

출처: IBM 소프트웨어(2014). 모바일 앱 개발 방식 비교: 네이티브, 웹, 하이브리드.

수행 내용 / 로봇 원격 모니터링 앱 개발하기

재료 · 자료

- 센서 데이터시트
- 작업에 대한 사용자 요구 사양서
- 로봇의 하드웨어 사양서 및 사용 설명서
- 프로그램 목적에 맞는 주변 환경 데이터
- 프로그램 저장용 저장장치

기기(장비 · 공구)

- 컴퓨터, 프린터
- 프로그램 대상 로봇
- 프로그램 개발용 소프트웨어
- 문서 작성용 소프트웨어
- 프로그램 설치용 장치 및 케이블

안전 · 유의 사항

- 원격 제어 툴 개발에서는 사용자 요구 사항을 효과적으로 구현하기 위하여 필요한 정보를 모니터링 범위로 하여야 한다.
- 학습의 이해도 향상을 위하여 특정 모델의 로봇과 모니터링 툴을 사용하지만, 실제 업무에서는 이와 다르다는 것을 미리 이해해야 한다.

수행 순서

① 스마트폰을 이용하여 로봇을 원격 제어할 수 있는 툴을 개발한다.

1. 안드로이드 개발 환경을 구성한다.

로봇용 안드로이드 앱 프로그램을 개발하기 위하여 안드로이드 개발 환경을 구성해야 한다. 안드로이드 개발 환경의 구성을 위하여 [그림 3-3]과 같은 JDK, [그림 3-4]와 같은 자바 개발 도구인 이클립스, [그림 3-5]와 같은 안드로이드 개발을 위한 소스들이 들어있는 개발 킷인 안드로이드 SDK를 포함하는 안드로이드 스튜디오를 다운로드하여 설치한다.

Oracle Technology Network > Java > Java SE > Downloads

Java SE
Java EE
Java ME
Java SE Support
Java SE Advanced & Suite
Java Embedded
Java DB
Web Tier
Java Card
Java TV
New to Java
Community
Java Magazine

Overview Downloads Documentation Community Technologies Training

Java SE Downloads



DOWNLOAD

Java Platform (JDK) 8u101 / 8u102



DOWNLOAD

NetBeans with JDK 8

Java Platform, Standard Edition

Java SE 8u101 / 8u102
Java SE 8u101 includes important security fixes. Oracle strongly recommends that all Java SE 8 users upgrade to this release. Java SE 8u102 is a patch-set update, including all of 8u101 plus additional features (described in the release notes).
[Learn more](#)

출처: Oracle(<http://www.oracle.com>). 2016. 9. 30. 스크린샷.
[그림 3-3] JDK 다운로드 및 설치

eclipse

GETTING STARTED MEMBERS PROJECTS MORE

HOME / DOWNLOADS / ECLIPSE DOWNLOADS - SELECT A MIRROR

All downloads are provided under the terms and conditions of the **Eclipse Foundation Software User Agreement** unless otherwise specified.

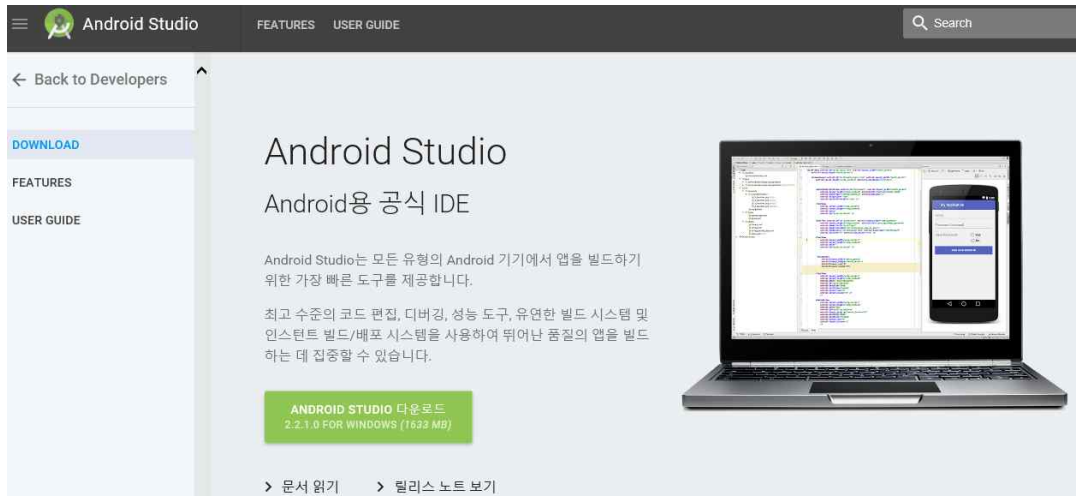
DOWNLOAD

Download from: Taiwan - Computer Center, Shu-Te University (<http>)

File: **eclipse-inst-win64.exe** SHA-512

>> Select Another Mirror

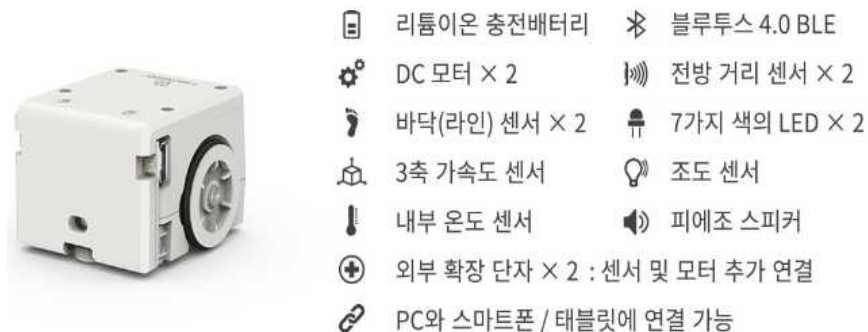
출처: Eclipse(<http://www.eclipse.org>). 2016. 9. 30. 스크린샷.
[그림 3-4] 이클립스 다운로드 및 설치



출처: Android Studio(<https://developer.android.com>). 2016. 9. 30. 스크린샷.
[그림 3-5] 안드로이드 스튜디오 다운로드 및 설치

2. 원격 제어를 위해 사용할 로봇을 선택한다.

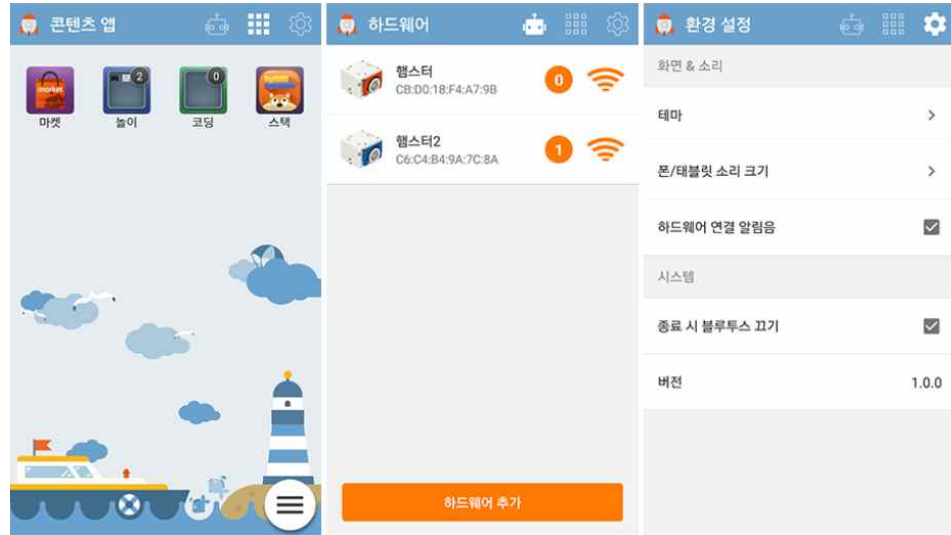
원격 제어를 위해 사용할 로봇을 제작하거나 완성품을 선택한다. 교육용 로봇으로 다양한 형태의 로봇들이 제공되고 있으나, 본 학습모듈에서는 OO사에서 제공하는 햄스터 로봇을 이용한다. [그림 3-6]은 햄스터 로봇의 기능을 나타내고 있다.



출처: OO메이션(<http://robomation-shop.co.kr>). 2016. 9. 30. 스크린샷.
[그림 3-6] 햄스터 로봇의 기능

3. 로보이드 론처를 설치한다.

로보이드 론처는 작고 감쪽한 교육용 로봇을 위한 앱들을 실행시켜 주는 론처 형태의 애플리케이션이다. 로보이드 론처를 사용하기 위해서는 햄스터 로봇 등의 실재 로봇이 있어야 한다. 로보이드 론처는 구글 플레이에서 로보이드 론처를 검색하여 설치할 수 있다. 로보이드 론처는 블루투스 4.0 BLE 통신으로 하드웨어 로봇과 통신하기 때문에 스마트폰이 블루투스 4.0 BLE를 지원하고, 안드로이드 OS 버전이 4.3 젤리빈 이상이어야 한다. [그림 3-7]은 로보이드 론처의 화면을 나타낸다.

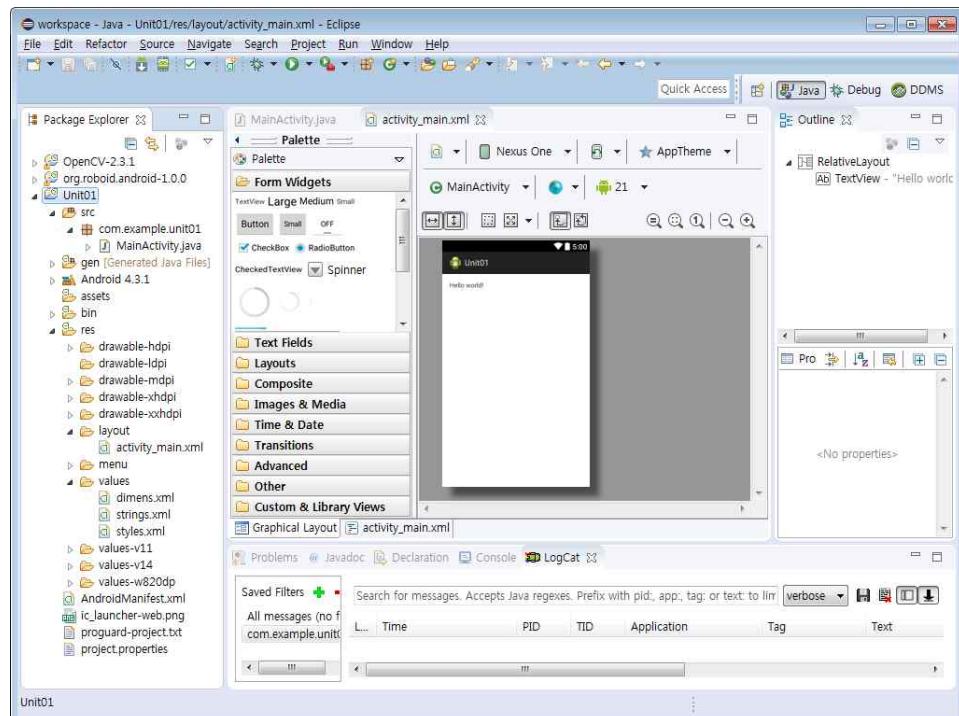


출처: 햄스터 스쿨(<http://hamster.school>). 2016. 9. 30. 스크린샷.
[그림 3-7] 로보이드 론처의 화면

4. 이클립스를 이용하여 앱 프로그램을 작성한다.

(1) 새 프로젝트를 생성한다.

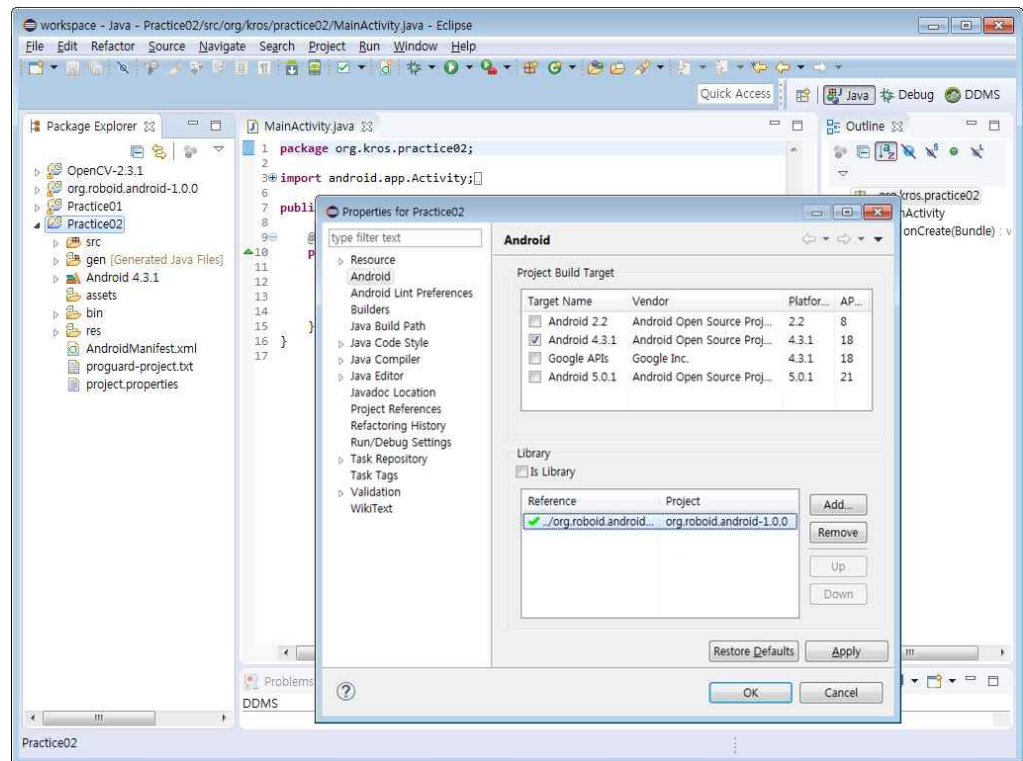
이클립스의 File > New > Android Application Project 메뉴를 선택하여 새로운 프로젝트를 생성한다. [그림 3-8]은 이클립스의 실행 화면을 나타낸다.



출처: 햄스터 스쿨(<http://hamster.school>). 2016. 9. 30. 스크린샷.
[그림 3-8] 이클립스에서 새 프로젝트 만들기

(2) 이클립스에 안드로이드 SDK를 연결시킨다.

안드로이드 스튜디오에는 안드로이드 앱 개발을 위한 소스들이 들어 있는 개발 킷인 안드로이드 SDK를 가지고 있다. 개발 툴들에 대한 설치가 끝나면 이클립스에 플러그인되어 있는 안드로이드 ADT가 안드로이드 SDK를 인식할 수 있도록 연결 작업을 해주어야 한다. 연결 작업은 project 메뉴의 하위 메뉴인 properties 메뉴를 통해서 [그림 3-9]와 같이 진행된다.



출처: 햄스터 스쿨(2016). 스마트폰 활용 영상 기반 로봇 프로그래밍. <http://hamster.school>에서 2016. 9. 30. 검색.
[그림 3-9] 이클립스와 안드로이드 SDK의 연결

(3) 로봇을 조정하기 위하여 액티비티 파일을 수정한다.

(가) MainActivity.java 파일을 열고 Activity를 RobotActivity로 수정한 후, Ctrl + Shift + o 키를 동시에 눌러 import 추가한다.

```
package org.kros.practice02;

import org.roboid.android.RobotActivity;

import android.os.Bundle;
import android.view.WindowManager;

public class MainActivity extends RobotActivity {
```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    getWindow().addFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON);
    setContentView(R.layout.activity_main);
}
}

```

(나) onInitialized 메소드, onExecute 메소드 추가하고, Ctrl + Shift + o 키를 동시에 눌러 import 추가한다.

```

public class MainActivity extends RobotActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        getWindow().addFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON);
        setContentView(R.layout.activity_main);
    }

    @Override
    public void onInitialized(World world) {
        super.onInitialized(world);
    }

    @Override
    public void onExecute() {
    }
}

```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    ...
    View view = findViewById(R.id.message);
    ...
}

```

```

@Override
public void onInitialized(World world) {
    super.onInitialized(world);
    ...
    Device device = world.findDeviceById(Hamster.LEFT_WHEEL);
    ...
    device.read(...);
    device.write(...);
}

```

(다) 바퀴를 전진시키기 위하여 RobotActivity에 다음과 같은 코드들을 추가한다.

왼쪽 바퀴(mLeftWheelDevice)와 오른쪽 바퀴(mRightWheelDevice)를 Device 클래스로 선언한 후, World 클래스의 findDeviceById 메소드를 이용하여 양쪽 바퀴의 ID를 얻을 수 있다. 그리고 나서, onExecute()에 mLeftWheelDevice.write(30);을 기재하게 되면, 로봇의 왼쪽 바퀴가 30의 값으로 전진을 하게 된다.

```

public class MainActivity extends RobotActivity {

    private Device mLeftWheelDevice;
    private Device mRightWheelDevice;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        ...
    }

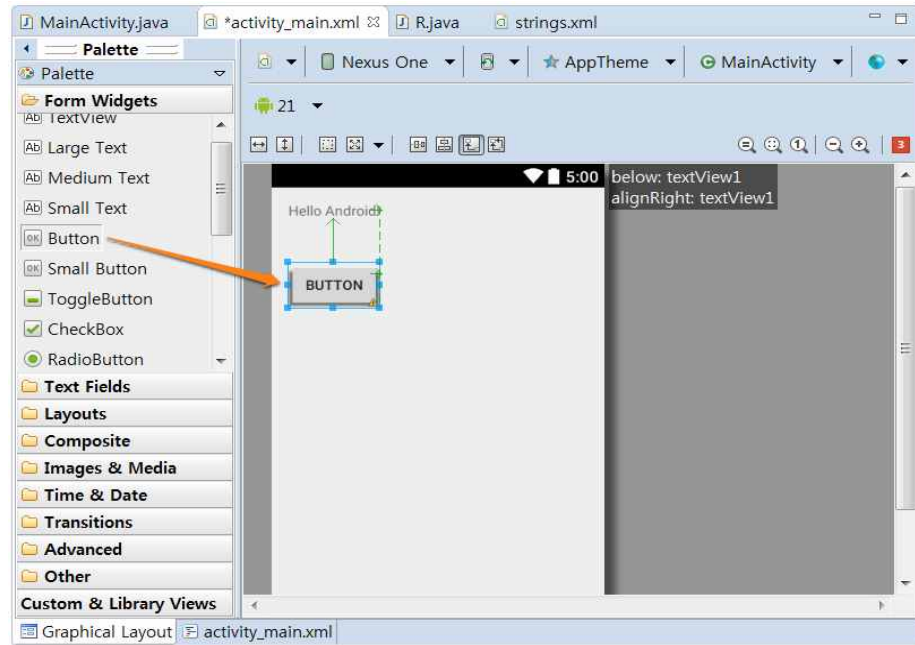
    @Override
    public void onInitialized(World world) {
        super.onInitialized(world);
        mLeftWheelDevice = world.findDeviceById(Hamster.LEFT_WHEEL);
        mRightWheelDevice = world.findDeviceById(Hamster.RIGHT_WHEEL);
    }

    @Override
    public void onExecute() {
        mLeftWheelDevice.write(30);
        mRightWheelDevice.write(30);
    }
}

```


(4) 스마트폰 화면에 표시될 안드로이드 버튼을 생성한다.

이클립스 왼쪽의 package explorer에서 activity_main.xml 파일을 클릭한 후, 오른쪽 화면에 있는 graphical layout 탭을 클릭한다. 마우스로 Form Widgets에 있는 Button이나 RadioButton을 클릭해서 오른쪽 화면으로 드래그하면, [그림 3-10]과 같이 오른쪽 화면 상에 Button이나 RadioButton이 생성된다.



출처: 슈만의 워드프레스와 미래공작소(<http://shuman.tistory.com/126>). 2016. 9. 30. 스크린샷.
[그림 3-10] 이클립스에서 버튼 만들기

activity_main.xml 탭을 클릭하면 해당 소스가 나타난다. 소스에는 [그림 3-11]과 같이 <Button> 태그로 시작하는 소스가 자동으로 생성되어 있음을 알 수 있다. 여기에서 안드로이드 앱 시스템은 ID 사용을 위하여 @+id란 형식을 사용한다. 즉, 그림에서 android:id의 값인 button1은 MainActivity.java 파일에서 호출될 것이다.



출처: 슈만의 워드프레스와 미래공작소(<http://shuman.tistory.com/126>). 2016. 9. 30. 스크린샷.
[그림 3-11] 이클립스에서 버튼 만들기

(5) [그림 3-12]와 같이 버튼을 눌렀을 때 로봇이 전진하는 프로그램을 만든다.

```

package com.example.hellojava;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button startBtn = (Button)findViewById(R.id.startBtn);
        startBtn.setOnClickListener( new OnClickListener(){
            public void onClick(View v){
                Toast.makeText(getApplicationContext(), "시작 버튼이 눌렸어요.",
                    Toast.LENGTH_LONG).show();
            }
        });
    }

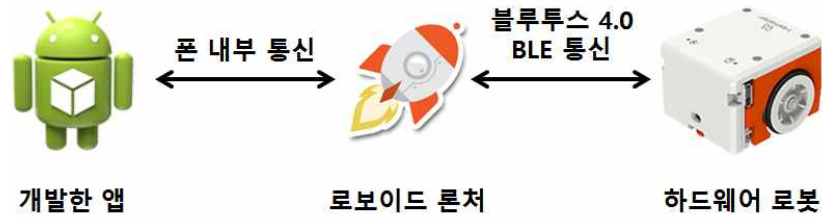
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.activity_main, menu);
        return true;
    }
}

```

출처: 슈만의 워드프레스와 미래공작소(<http://shuman.tistory.com/126>). 2016. 9. 30. 스크린샷.
[그림 3-12] 로봇이 전진하는 프로그램

(6) 스마트폰 앱을 실행시켜 로봇을 조정해 본다.

- [그림 3-13]과 같이 로보이드 론처를 실행시킨다.
- 로봇에 연결되면 론처를 종료하지 않은 상태에서 개발한 앱을 실행한다.
- 개발한 앱을 종료하면 로봇은 정지한다.



출처: 햄스터 스쿨(2016). 스마트폰 활용 영상 기반 로봇 프로그래밍. <http://hamster.school>에서 2016. 9. 30. 검색.
[그림 3-13] 로보이드 론처를 이용한 로봇 제어

학습 1	모니터링 툴 개발하기
학습 2	외부 툴과의 동기화 소프트웨어 개발하기
학습 3	로봇 응용 소프트웨어 개발하기
학습 4	로봇 훈련 소프트웨어 개발하기
학습 5	다수 로봇 제어 소프트웨어 개발하기

4-1. 로봇 학습 알고리즘 개발

학습 목표

- 로봇의 학습 절차를 설정하고 학습 프로그램을 설계할 수 있다.
- 인공지능 프로그램의 구조를 파악하고 프로그램을 할 수 있다.
- 로봇 훈련에 필요한 데이터를 수집, 활용하는 절차를 이해하고 프로그램을 할 수 있다.
- 데이터베이스를 구축하고 활용하는 프로그램을 할 수 있다.

필요 지식 /

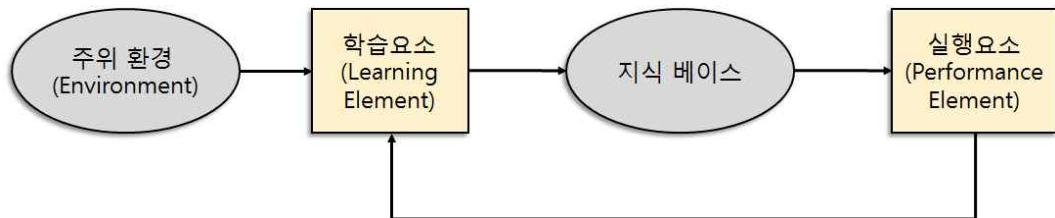
① 로봇 학습

1. 기계 학습(machine learning)

머신 러닝 즉, 기계 학습은 인공 지능의 한 분야로 컴퓨터가 학습할 수 있도록 하는 알고리즘과 기술을 개발하는 분야를 말한다. 가령 기계 학습을 통해서 학습한 수신한 이메일이 스팸인지 아닌지를 구분할 수 있도록 훈련할 수 있다.

[그림 4-1]은 단순 기계 학습 시스템의 모델을 나타낸다. 그림에서 원으로 표시된 부분은 정보의 선언적인 부분(즉, 사실 표현)을 나타내며 사각형은 절차들을 나타낸다. 화살표는 학습 시스템 내에서의 자료 흐름의 방향을 나타낸다. 주위 환경은 학습 요소(learning element)에 특정 정보를 제공하고, 학습 요소는 이 정보를 이용하여 지식베이스를 개선한다. 실행 요소는 지식베이스를 이용하여 특정 작업을 수행하고, 작업을 수행하는 동안에 얻은 정보는 다시 학습 요소에 반영시킨다. 이 모델은 가장 기본적인 것으로서 많은 중요한 기능들이 생략되어 있지만, 이들 네 가지 구성 요소들을 어떻게 구현하는가에 따라서

학습 시스템을 분류하는 데 많은 도움을 줄 수 있다. 특정의 응용 시스템이 있다면, 주위 환경, 지식베이스, 실행 작업들이 주어진 학습 문제의 본질을 결정지을 수 있으며, 이를 통해서 학습 요소가 수행해야 하는 특정 기능들을 결정할 수 있다.



[그림 4-1] 단순 기계 학습 시스템의 모델

기계 학습은 주어진 데이터를 훈련시켜(training), 훈련된 지식을 기반으로 새로운 입력(test input)에 대해 적절한 답(test output)을 찾고자하는 일련의 과정이다. 이때 훈련시키는 데이터가 질문(training input)과 정답(training output)이 모두 주어진 경우가 있고, 질문만 주어진 경우도 있다. 전자의 경우를 라벨링(labeling)이 되어 있다고 말한다. 기계 학습의 문제는 이 라벨링의 유무에 따라 지도 학습(supervised learning), 비지도 학습(unsupervised learning), 강화 학습(reinforcement learning)으로 나눌 수 있다.

지도 학습은 훈련 데이터에 라벨링이 되어있는 경우, 즉 각 질문(input)에 대해 무엇이 정답(output)인지 훈련 데이터가 알고 있는 경우이다. 대표적인 방법으로 예측(regression)은 (x, y)들의 데이터를 가지고 모함수 $y=f(x)$ 의 관계를 예측하는 방법이며, 분류(classification)는 각 그룹별 특징을 학습하여 새로운 데이터가 어느 그룹에 속해야 하는지 판단하는 방법이다.

비지도 학습은 데이터만 주어져 있는 경우이다. 즉, 문제는 있는데 답은 없는 경우이다. 대표적인 방법으로 클러스터링(clustering)은 어떠한 데이터끼리 군집으로 묶을 수 있는지 판단하는 방법이며, 차원 축소(dimension reduction)는 독립적인 특징(축)들을 추출해 이를 중심으로 데이터를 압축하는 저차원으로 압축하는 방법이다.

강화 학습은 어떤 환경 안에서 정의된 에이전트가 현재의 상태를 인식하여, 선택 가능한 행동들 중 보상을 최대화하는 행동 혹은 행동 순서를 선택하는 방법이다. 강화 학습은 로봇 제어나 엘리베이터 스케줄링 등에 성공적으로 적용되어 왔다.

② 머신 비전

로봇 시스템에서 로봇이 작업물을 쉽게 잡을 수 있도록 고정구나 지그 등을 이용하여 작업물의 방향이 일정하게 유지된다면 로봇은 엔드이펙트를 이용하여 쉽게 작업물을 잡을 수 있다. 그러나 작업물의 종류가 많거나 작업물의 위치가 움직일 때에는 로봇이 작업물을 잡기 위해서 머신 비전(machine vision) 기술로 작업물의 형상을 파악해야 한다.

1. 디지털 영상 처리

디지털 영상 처리(digital image processing)는 컴퓨터를 이용하여 영상을 생성하고, 처리하고 영상을 해석 및 인식하는, 즉 영상과 관련된 모든 분야를 의미한다. 이 기술을 이용하여 우리는 흐린 영상을 보다 선명하게 볼 수 있다거나, 영상이 훼손된 경우 다시 원 영상으로 복원한다든지 영상에서 필요한 정보만을 추출하여 얻을 수 있는 등 다방면으로 활용할 수 있다.

영상 처리는 컴퓨터 그래픽(computer graphics), 머신 비전(computer vision)과 밀접한 관계가 있다. 컴퓨터 그래픽은 컴퓨터를 이용하여 영상을 생성 시키는 쪽에 주력하는 분야이고, 머신 비전은 영상 처리 중에서도 특히 영상의 인식, 이해 등을 중점적으로 주로 연구하는 분야이다. 반면에 영상 처리는 보통 여러 장치들을 통하여 이미 생성된 영상을 입력하여 영상을 변화시키는 것이며 영상을 재가공하거나 영상에서 정보를 추출하는 과정이라 할 수 있다.

산업용 로봇에서는 머신 비전이 많이 사용된다. 산업용 로봇에서 사용되는 머신 비전은 2D와 3D로 분류된다. 2차원 시스템은 장면(scene)을 2D 이미지로 나타낸다. 산업 응용에서의 많은 상황이 2D 장면을 포함하기 때문에 대부분의 경우 2차원 시스템이 적당하다. 예로서는 로봇이 작업물을 잡기 전에 작업물의 치수를 측정하는 일이 포함된다.

최근에는 빈 피킹(bin picking)과 같은 보다 복잡한 형태의 로봇 작업이 필요한 경우가 있다. 작업물을 일정한 형태로 놓기 힘든 부품들(예를 들면 스마트폰 케이스에 들어가는 이어폰 등)을 로봇이 잡기 위해서는 3D 영상을 인식해야 한다.

머신 비전은 이미지 획득과 디지털화, 이미지 처리와 분석, 해석의 세 단계로 구분된다.

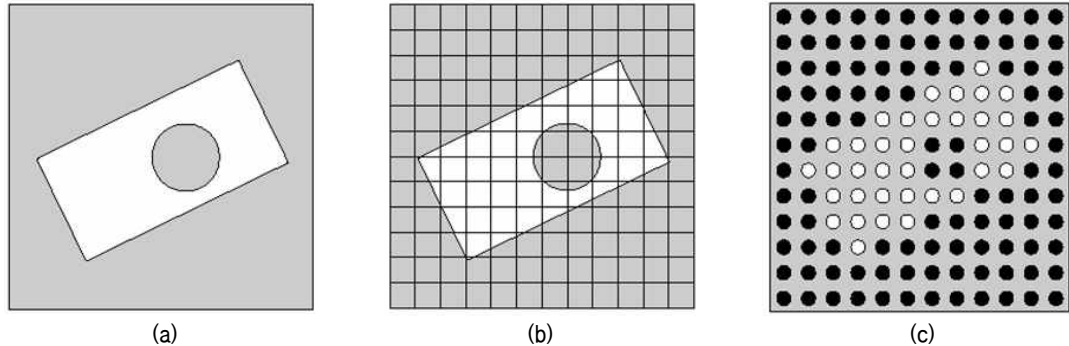
2. 이미지 획득과 디지털화

이미지 획득과 디지털화는 다음 단계의 분석을 위한 이미지 데이터를 저장하기 위하여 카메라와 디지털화 시스템을 사용하여 수행된다. 카메라는 측정하고자 하는 물체에 초점을 맞춘다. 이미지는 시각 영역(viewing area)을 이산적인 화소(픽셀이라 부름)의 행렬로 나뉘어진다. 이로써 얻어지는데, 이때 각 화소는 장면 각 부분의 빛의 세기에 비례하는 값을 갖는다. 각 화소의 빛의 세기는 ADC에 해당되는 디지털 값으로 바뀐다.

[그림 4-2]는 2진 비전 시스템이라고 불리는 가장 간단한 형식의 비전 시스템으로부터 얻어질 수 있는 이미지를 보여준다. 2진 비전(binary vision)에서는 각 화소의 빛의 세기가 주어진 임계값(threshold)을 넘었는지의 여부에 따라서 흰색과 검은색의 두 가지 극단 값으로 수렴된다. 더욱 정교한 비전 시스템은 회색의 서로 다른 농도를 구분하고 저장할 수 있다. 이것을 그레이 스케일(gray scale) 시스템이라 부른다. 이 형식의 시스템은 물체의 윤곽과 면적 특징뿐만 아니라 질감이나 색깔과 같은 표면 특성도 알아낼 수 있다.

그레이 스케일 비전 시스템은 보통 8비트 메모리를 사용하는데, 256개의 농도 레벨을 구분할 수 있다.

카메라의 경우, 최근에는 반도체 카메라가 많이 사용된다. 반도체 카메라는 CCD(charge-coupled device), CID(charge-injected device), CPD(charge-printing device)로 구분된다.

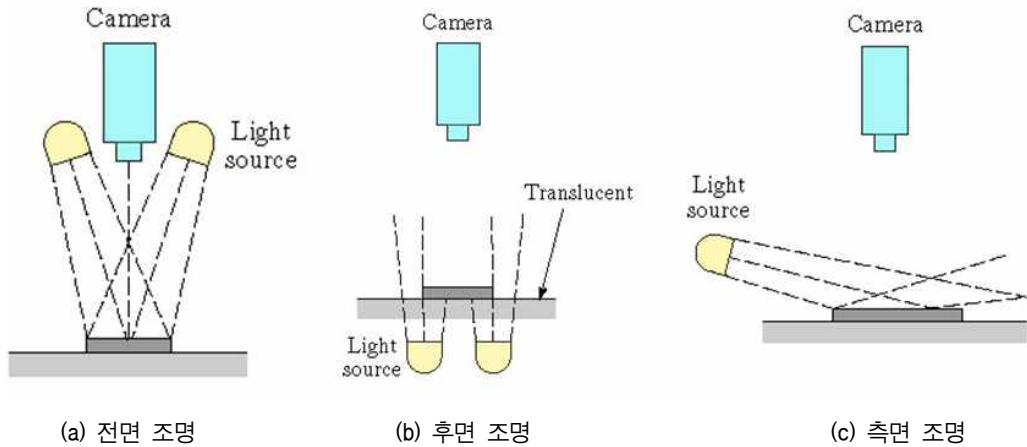


출처: 한영근(2009). 현대 생산자동화와 CIM. 시그마프레스.
[그림 4-2] 2진 비전 시스템의 이미지

화소 배열은 640(수평)×480(수직), 1024×768, 1040×1392 화소 등이 사용된다. 비전 시스템의 해상도(resolution)는 이미지의 미세한 부분과 특징들을 감지해내는 능력을 의미한다. 해상도는 사용된 화소(픽셀)의 개수에 달려 있다. 즉, 비전시스템에 설계된 화소가 많을수록 해상도는 높아지고, 화소의 수가 늘어날수록 카메라의 가격은 높아진다. 더욱 중요한 점은 화소의 수가 늘어날수록 화소를 순차적으로 읽고 데이터를 처리하는 데 걸리는 시간이 증가한다.

머신 비전에서 가장 중요한 것 중의 하나는 조명이다. 비전 카메라에 잡힐 장면은 반드시 잘 조명되어야 하며, 조명은 시간에 대하여 일정해야 한다. 설비의 주변광에 의존하지 말고 머신 비전 응용을 위한 특별한 조명이 설치되어야 한다.

[그림 4-3]은 조명의 종류를 나타내고 있다. 조명에는 전면조명, 후면조명, 측면조명 등이 있다. 전면조명의 경우, 광원은 물체에 대해 카메라와 같은 쪽에 위치한다. 이것은 상표의 프린트 같은 표면 특징과 용접선과 같은 표면 패턴을 검사할 수 있는 물체가 반사하는 반사광을 만들어낸다. 로봇 응용에서 카메라는 로봇의 암 끝단에 설치를 많이 하며, 조명은 로봇이 작업해야 하는 공간의 천정에 설치하는 경우가 많다. 후면조명의 경우, 광원은 카메라가 바라보는 물체의 뒤에 놓인다. 이것은 밝은 배경과 뚜렷한 명암 대조를 이루는 물체의 어두움 윤곽을 만들어 낸다. 이 형태의 조명은 2진 비전시스템이 물체의 치수를 검사하고 서로 다른 물체의 윤곽을 구별하는 데 사용된다. 측면조명의 경우, 평탄한 표면에 불규칙성이 있을 때 그 부분이 비전 시스템으로 감지할 수 있는 그림자를 만들도록 만든다. 이것은 물체의 표면의 결함이나 흠의 검사에 사용된다.



출처: 한영근(2009). 현대 생산자동화와 CIM. 시그마프레스.
[그림 4-3] 조명의 종류

3. 이미지 처리와 분석

머신 비전의 두 번째 작업은 이미지 처리와 분석이다. 로봇이 실시간으로 작업하기 위해서는 매우 빠른 속도로 이미지 처리가 이루어져야 한다(보통 초당 10FPS(frame per second)에서 30FPS). 이를 위하여 다양한 이미지 처리 기술들이 개발되었다.

분할(segmentation)은 이미지 안에 있는 관심 영역을 정의하고 분리해 내는 기술이며, 이진화(binartization or thresholding)는 영상의 픽셀 값을 0 또는 255로 만드는 방법이다. 윤곽선 검출(edge detection)은 이미지에서 물체와 배경 사이의 경계의 위치를 찾는 방법이다.

이진화는 각 픽셀의 농도를 흑이나 백을 나타내는 2진 레벨로 변환시키는 것인데, 농도 값을 정의된 임계값(threshold)보다 밝은 픽셀들은 모두 흰색으로, 그렇지 않은 픽셀들은 모두 검은색으로 바꾸는 것을 말한다.

윤곽선 검출은 이미지에서 물체와 배경 사이의 경계의 위치를 찾는 것이다. 윤곽선 검출은 미분 연산자를 이용하여 이미지의 밝기 값의 변화를 이용하여 찾아내는 것이다. 소벨 마스크(Sobel mask)나 프리윗(Prewitt) 마스크 등과 같은 마스크 연산 방법이 사용된다.

분할 작업에 뒤따른 이미지 처리와 분석의 다른 기술은 특징 형상 추출(feature extraction)이다. 대부분의 머신 비전 시스템은 물체의 특징 형상을 이용하여 이미지 안의 물체를 특징짓는다. 물체의 특징 형상으로는 물체의 면적, 길이, 폭, 지름, 둘레, 무게 중심, 폭/높이 비율 등이 있다. 특징 형상 추출 방법은 이진화나 모서리 추출 등을 이용하여 얻어진 물체의 면적과 경계를 기준으로 이들 특징 형상을 결정하도록 설계된다.

4. 해석

어떤 주어진 응용에서 이미지는 추출된 특징 형상을 기초로 해석되어야 한다. 해석 기능은 보통 물체를 인식하는 것과 관련되는데, 물체 인식 또는 패턴 인식(pattern recognition)이라는 용어로 불리는 작업이다. 이 작업의 목적은 미리 정의된 모델이나 표준값과 이미

지 안의 물체를 비교하여 그 물체를 인식하는 것이다. 두 가지 흔히 사용되는 해석 기술은 템플리트 매칭(template matching)과 특징 가중(feature weighting)이다.

템플리트 매칭은 이미지의 하나 또는 여러 특징 형상을 컴퓨터 메모리에 저장된 모델이나 템플리트의 해당 특징과 비교하는 방법이다. 각종 기초적인 템플리트 매칭 기술은 이미지의 각각의 픽셀을 해당 컴퓨터 모델의 픽셀과 비교하는 방법이다. 템플리트 매칭의 어려움 중의 하나는 복잡한 이미지 처리 없이 비교가 쉽게 이루어지게 하기 위하여 카메라의 앞에 물체를 같은 위치, 같은 방향으로 정렬하는 것이다.

특징 가중은 물체를 인식하는 데 있어 상대적인 중요도에 따라 각 특징(예로 면적, 길이, 둘레와 같은)에 가중치를 할당함으로써 여러 특징을 하나의 측정으로 결합하는 방법이다. 이미지 상의 물체의 점수는 적절히 인식하기 위하여 컴퓨터 메모리에 저장되어 있는 이상적인 물체의 점수와 비교된다.

수행 내용 / 로봇 학습 알고리즘 개발하기

재료 · 자료

- 센서 데이터시트
- 작업에 대한 사용자 요구 사양서
- 로봇의 하드웨어 사양서 및 사용 설명서
- 프로그램 목적에 맞는 주변 환경 데이터
- 프로그램 저장용 저장장치
- 훈련 목적에 맞는 훈련 대상물(예: 사람 얼굴 인식 훈련 - 인물 사진)

기기(장비 · 공구)

- 컴퓨터, 프린터
- 프로그램 대상 로봇
- 프로그램 개발용 소프트웨어
- 문서 작성용 소프트웨어
- 프로그램 설치용 장치 및 케이블

안전 · 유의 사항

- 해당사항 없음

수행 순서

① 머신 비전 프로그램을 개발한다.

1. 적용 대상을 선정하고, 이에 적합한 카메라 시스템을 설계한다.

인식해야 될 대상의 크기와 정밀도를 고려하여 비전 시스템을 위한 설계 요구 조건을 결정해야 한다. 예로, 인식 대상물의 크기가 100mm, 비전 정밀도가 $\pm 0.3\text{mm}$ 이하 등으로 설계한다.

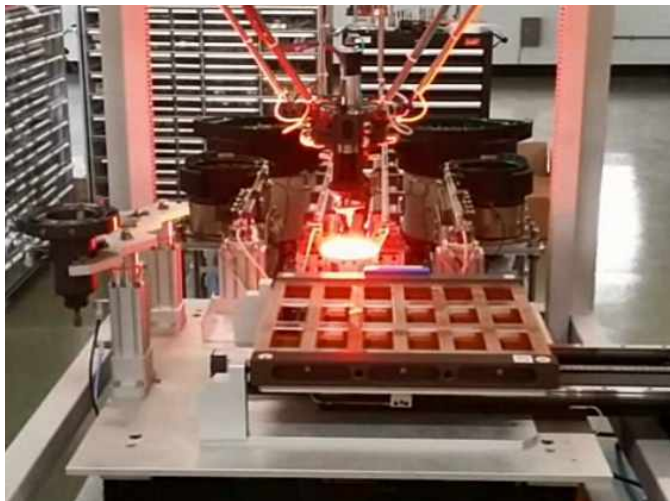
설계 목표가 정해지고 나면, 카메라 시스템을 선정한다. 카메라 시스템에는 카메라의 종류, 해상도, 단위픽셀의 크기, 프레임 속도, FOV, 비전 분해능 등이 있다. <표 4-1>은 카메라 시스템에서 결정되어야 할 사양들을 나타낸다.

<표 4-1> 카메라 시스템의 사양

	사양 1	사양 2
카메라	5Mega CCD	12Mega CMOS
해상도	2448 × 2058	4096 × 3072
단위픽셀(um)	3.45 um	5.5 um
프레임(fps)	15 fps	11 fps
렌즈 배율	0.22x	0.055x
FOV	x-38	x-409
비전 분해능(um/pixel)	15.68 um/pixel	100 um/pixel

2. 조명 방식을 선정한다.

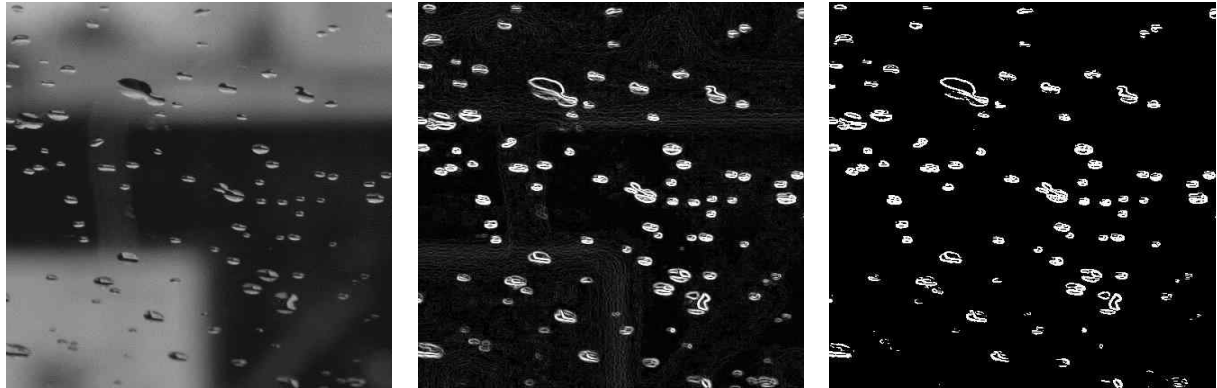
조명 방식은 전면 조명, 후면 조명, 측면 조명 등이 있다. 적용되어야 할 응용 특성에 맞게 적합한 조명 방식을 선정한다. [그림 4-4]는 로봇 암에 조명을 부착한 응용 방법이다. 최근에는 로봇 암에 조명을 달고 부품을 잡을 때 조명을 비추는 방법도 많이 사용되고 있다.



[그림 4-4] 비전 시스템을 이용한 물체 인식 응용의 예

3. 이미지 처리 프로그램을 개발한다.

인식해야 할 부품의 형상, 크기, 색상 등을 인식 및 분류할 수 있는 이미지 처리 프로그램을 개발한다. 이미지 처리 프로그램에는 이진화와 윤곽선 검출 알고리즘 등이 들어간다. 이미지 처리 알고리즘의 경우 소스 코드를 직접 개발하는 방법과 상용 라이브러리를 사용하는 방법이 있다. 최근에는 Halcon, Cognex, MIL, Euresys 등과 같은 상용 라이브러리들이 많이 사용되고 있다.



(a) 원 영상 (b) Sobel 마스크 처리 결과 이미지 (c) 이진화 적용 결과 이미지

[그림 4-5] 이미지 처리 결과의 예

4. 템플릿 매칭을 할 수 있는 분류기를 개발한다.

비전 시스템에서 여러 개의 물체를 인식하여 각 물체를 구분하거나, 제품의 합격과 불합격을 구분하기 위해서는 지식베이스에 있는 특징 형상과 비교를 하여야 한다. 이러한 분류기는 다양한 형태의 알고리즘을 이용하여 만들 수 있는데, 기계 학습 알고리즘, 특히 강화 학습이나 비지도 학습 등을 이용하여 개발한다.

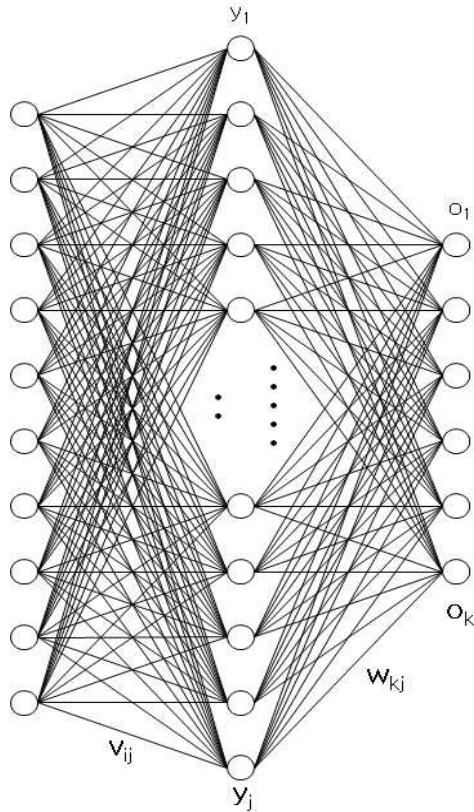
② 로봇 시스템에서 사용할 수 있는 기계 학습 응용 프로그램을 개발한다.

1. 기계 학습을 위한 데이터를 수집한다.

머신 비전에서 사용되는 분류기와 같은 머신 러닝 알고리즘을 개발하기 위하여 관련 데이터를 수집해야 한다. 인식해야 될 대상에 대한 정확한 형상, 크기, 색상 등에 대한 정보를 수집해야 한다.

2. 적합한 알고리즘을 선택하고, 알고리즘에 적합한 입력 데이터를 만든다.

기계 학습 알고리즘에는 매우 다양한 알고리즘들이 존재한다. 따라서 개발하고자 하는 응용에 가장 적합한 형태의 알고리즘을 선택해야 한다. 최근에 널리 알려진 딥러닝(deep learning)은 매우 강력한 학습 능력을 가지고 있다고 알려져 있지만 매우 많은 데이터와 매우 많은 컴퓨팅 파워가 필요하다. 복잡한 형상을 인식해야 하는 머신 비전에서는 딥러닝이 좋은 대안이 될 수 있겠으나, 간단한 부품의 형상을 인식하는 머신 비전 응용에서는 [그림 4-6]과 같은 단순한 뉴럴 네트워크가 좋은 대안이 될 수 있다.



[그림 4-6] 기계 학습 알고리즘 중 하나인 뉴럴 네트워크의 예

알고리즘이 선택되었다면, 알고리즘에 적합한 형태의 입력 데이터를 만들어야 한다. 어떤 알고리즘은 부품에 대한 특별한 형식의 속성을 요구하기도 하고, 문자열처럼 목적 변수와 속성을 대응시킬 수 있고, 또 어떤 경우는 속성이 정수가 되도록 해야 될 수도 있다.

3. 머신 러닝 프로그램을 개발한 후 알고리즘을 훈련시킨다.

알고리즘이 선택되었다면, 다양한 형태의 프로그래밍 언어를 이용해서 머신 러닝 프로그램을 개발한다. 최근에는 머신 러닝 프로그램의 유료 또는 무료 라이브러리 형태로 제공되고 있다. 이러한 라이브러리를 이용하는 경우 손쉽게 머신 러닝 프로그램을 만들 수 있다. 앞 단계에서 준비된 다수의 훈련 데이터를 이용하여 머신 러닝 프로그램을 충분히 훈련시킨다.

4. 새로운 입력을 이용하여 머신 러닝 프로그램을 테스트한다.

알고리즘이 충분히 훈련되었다면, 훈련이 얼마나 잘 되었는지를 테스트해 보아야 한다. 지도 학습의 경우에는 알려진 입력 값과 출력 값을 이용하여 알고리즘이 제대로 동작하는지를 평가해 본다. 비지도 학습의 경우에는 인정 가능한 알고리즘을 평가하는 데 사용할 수 있는 몇 가지 알려진 값을 이용할 수 있다. 비지도 학습의 경우에는 성공 여부를 평가하기 위해 다른 통계를 사용할 수 있다. 두 경우 중 어떤 것도 만족스럽지 못하다면 전 단계로 돌아가 또 다른 훈련 세트를 가지고 알고리즘을 충분히 훈련시킨 후 다시 테스트를 시도한다.

학습 1	모니터링 툴 개발하기
학습 2	외부 툴과의 동기화 소프트웨어 개발하기
학습 3	로봇 응용 소프트웨어 개발하기
학습 4	로봇 훈련 소프트웨어 개발하기

학습 5 다수 로봇 제어 소프트웨어 개발하기

5-1. 다수 로봇 제어 프로그램 개발

학습 목표

- 다양한 사양의 로봇 특성을 파악하여 프로토콜을 설계하고 작성할 수 있다.
- 여러 대의 로봇이 동시에 연동하여 움직일 때 발생할 수 있는 문제점을 예상하고 경로계획 및 충돌회피 등의 대응 방안을 마련할 수 있다.
- 로봇끼리 정보를 교환할 수 있는 통신 방법을 이해하고 프로그램 할 수 있다.
- 프로그램을 로봇에 설치하고 배포할 수 있다.
- 여러 대의 로봇을 통제하는 서버의 특성을 파악하고 프로그램을 설계할 수 있다.
- 서버와 로봇, 로봇과 로봇을 동기화 하여 연동하는 프로그램을 할 수 있다.
- 로봇에 고유코드를 부여하고 구별하여 운용할 수 있다.
- 로봇을 서버에 등록하고 정보를 수정 및 삭제 할 수 있는 프로그램을 작성할 수 있다.
- 로봇의 로그인 정보를 관리하고 배포할 수 있는 프로그램을 작성할 수 있다.

필요 지식 /

① EtherCAT

1. EtherCAT 개요

EtherCAT 프로토콜은 2002년에 독일 BeckHoff사에서 개발된 이후 2003년 11월에 EtherCAT technology group을 결성해서 기술을 공개한 개방형 산업용 Ethernet 기술이다. 또 IEC 규격(IEC/PAS 62407)과 ISO 규격(ISO15745-4)으로 인증된 국제표준 프로토콜로 뛰어난 동기화 특성과 함께 제한된 토폴로지에 의존하지 않는 성능을 가진다. 특히, EtherCAT 프로토콜은 뛰어난 Ethernet 호환성, 간단한 디바이스에서도 구현이 가능한 인터넷 기술, Ethernet에서 제공하는 대역폭을 최대한 활용할 수 있는 특성과 낮은 비용으로

뛰어난 실시간 특성을 구현할 수 있다는 장점을 가지고 있다. EtherCAT 프로토콜은 100Mbps 이상의 통신 속도를 제공하며 256개의 분산 디지털 I/O에 약 11 μ s의 업데이트 속도를 지원한다.

EtherCAT 프로토콜의 전송 방법은 브로드캐스트(broadcast) 방식으로 마스터에서 데이터 프레임을 전송한다. 마스터와 연결된 각각의 슬레이브는 데이터 프레임을 수신 받아 해석 및 처리하고 다음 슬레이브로 전송을 한다. 즉, EtherCAT 프로토콜에서 각각의 슬레이브는 데이터 프레임이 슬레이브 모듈을 통과하는 동안 해당 모듈에 전달된 데이터를 읽어 데이터를 수신하고 전송할 데이터가 있으면 해당되는 텔레그램이 통과하는 동안 입력할 데이터를 해당되는 텔레그램에 삽입하여 전송한다. 여기서 해당되는 슬레이브 모듈에서 지연되는 시간은 겨우 몇 ns만이 소요된다.

EtherCAT 마스터는 일반적으로 PC에서 사용하는 NIC를 사용하여 구성할 수 있으며, NDIS(network driver interface specification) 인터페이스를 이용하여 하부계층과 통신할 수 있다. NDIS 구조는 물리 계층과 데이터링크 계층, 그리고 응용 계층만을 가지고 있으며, 응용 계층에서 생성된 메시지가 2계층에 해당하는 데이터링크 계층으로 직접 접근이 가능하다. 또 packet 단위의 데이터 송수신을 직접적으로 제어하는 것도 가능하다. EtherCAT 슬레이브 모듈은 BeckHoff사에서 제공하는 EtherCAT 컨트롤러(controller)와 이를 제어하는 MCU로 구성된다. EtherCAT 컨트롤러를 이용하면 SPI 또는 bus 방식을 통해 다양한 MCU와 직접 연결하여 EtherCAT 슬레이브 모듈을 구현할 수 있다.

EtherCAT 프로토콜의 데이터 프레임은 스위칭 허브를 통하여 일반 Ethernet과 혼용하여 사용할 수 있는 UDP(user datagram protocol) 타입과 실시간 특성을 보장하기 위한 EtherCAT 전용 프레임 타입이 있다. [그림 5-1]은 EtherCAT 프로토콜의 전용 데이터 프레임의 구조를 나타내고 있다. 그림에서 데이터 프레임은 주소 정보를 포함하는 Ethernet 헤더(header), EtherCAT 정보를 포함하는 ECAT, EtherCAT 데이터를 포함하는 EtherCAT 텔레그램(telegram)으로 구성된다. Ethernet 헤더는 프레임의 시작을 의미하는 Pre(preamble), 목적지 주소를 의미하는 DA(destination address), 출발지 주소를 의미하는 SA(source address)와 프레임의 타입을 결정하는 type으로 구성된다. ECAT의 frame HDR은 EtherCAT 프레임의 길이와 프로토콜 타입을 의미한다. 슬레이브 모듈이 데이터를 기입하는 EtherCAT 텔레그램은 EtherCAT 명령에 따라 필요한 지침과 주소를 기입하는 EtherCAT HDR, 필요한 데이터를 기입하는 data와 슬레이브 모듈의 데이터 작성 유무를 표시하는 WKC로 구성된다. 마지막으로 FCS는 EtherCAT 프레임의 에러를 확인하는 기능을 수행한다.

EtherCAT 프로토콜은 라인, 스타, 트리 토폴로지를 지원하며 브로드캐스트 방식으로 스위치나 허브의 개수에 제한이 없이 어떤 토폴로지에도 적용이 가능하다. 특히, 라인과 트리 토폴로지의 조합을 시스템의 배선에 활용하면 각각의 디바이스들은 배선 선택의 유연성이 증가된다. 일반적으로 EtherCAT 프로토콜은 고속 Ethernet(100Base Tx 내장형 네트워크

서버)으로 각각의 디바이스 간의 배선의 길이는 최대 100m까지 가능하다.

EtherCAT Header				EtherCAT	EtherCAT PDU			EtherCAT PDU				Enet
Pre	DA	SA	Ether Type	Frame HDR	EtherCAT HDR	DATA	W K C	EtherCAT HDR	DATA	W K C		FCS
(8)	(6)	(6)	(2)	(2)	(10)	(1..1486)	(2)	(10)	(1..1486)	(2)		(4)

[그림 5-1] EtherCAT 프레임 구조

2. EtherCAT 네트워크 인터페이스 구현을 위한 계층

(1) 물리 계층

EtherCAT의 물리 계층은 데이터 링크 레이어로부터 데이터를 수신하여, 전송 매체에 연결된 다른 노드에서 전송할 수 있는 신호 레벨로 인코딩하는 역할을 수행한다. 또 다른 노드로부터 수신된 신호를 디코딩하여 수신 측 데이터 링크 레이어로 전달하는 역할을 수행한다.

(2) 데이터링크 계층

데이터링크 레이어는 EtherCAT을 통해 연결된 디바이스의 데이터 통신을 위한 핵심적인 실시간 기능을 담당한다. 실시간(time-critical)은 어플리케이션 계층에서 사용하는 시간에서 특정한 액션을 수행하기 위해 요구되는 시간으로 정의한다.

이러한 실시간성에 문제가 생기면 사용하고 있는 어플리케이션 계층의 응용프로그램의 오류와, 더 나아가 디바이스 자체의 기기 결함으로 이어질 수 있다.

데이터링크 레이어는 Ethernet frame을 통해서 수신된 데이터의 계산과 비교, FCS의 생성 등을 담당하고 있으며, 기 정의된 메모리에 데이터를 저장하는 역할도 동시에 수행한다.

(3) 어플리케이션 계층

어플리케이션 계층은 자동화 환경에서 요구되는 실시간 응용프로그램의 동작 및 작성의 편리성을 제공하는 역할을 함. EtherCAT은 여러 가지 선택적인 서비스와 프로토콜을 제공하고 있으며, TCP/UDP/IP-based Protocol을 내장하고 있다.

대표적으로 EtherCAT에서 제공하고 있는 응용프로그램의 프로파일로는 파라미터의 관리를 위한 SDOs, PDOs와 Object Dictionary Structure를 지원하는 CANopen이 있다.

수행 내용 / 네트워크 기반 로봇 제어 프로그램 개발하기

재료 · 자료

- 각 로봇의 하드웨어 사양서 및 사용 설명서
- 각 로봇의 소프트웨어 개발 사양서
- 통합에 서버가 사용된 경우 서버의 하드웨어 사양서 및 소프트웨어 개발 사양서
- 정보 교환에 사용되는 프로토콜 사양서
- 프로그램 저장용 저장장치

기기(장비 · 공구)

- 서버, 네트워크 환경
- 컴퓨터, 프린터
- 프로그램 대상 로봇
- 프로그램 개발용 소프트웨어
- 문서 편집용 소프트웨어
- 프로그램 설치용 장치 및 케이블

안전 · 유의 사항

- 원격 제어 툴 개발에서는 사용자 요구 사항을 효과적으로 구현하기 위하여 필요한 정보를 모니터링 범위로 하여야 한다.
- 학습의 이해도 향상을 위하여 특정 모델의 로봇과 모니터링 툴을 사용하지만, 실제 업무에서는 이와 다르다는 것을 미리 이해해야 한다.

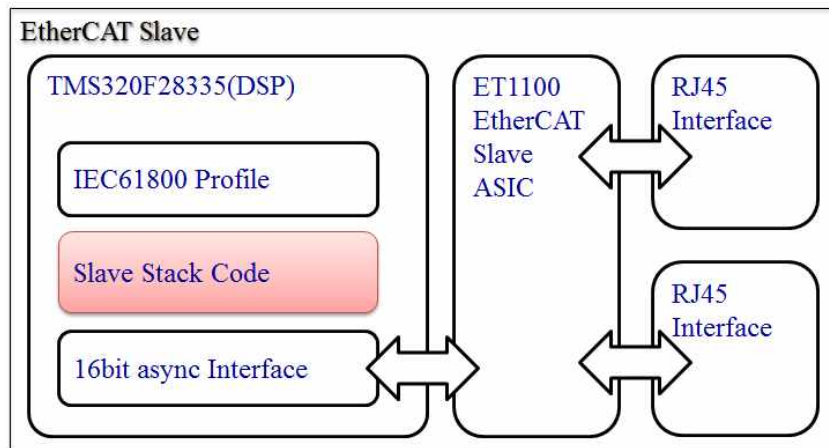
수행 순서

① EtherCAT 통신 소프트웨어를 개발한다.

1. EtherCAT 슬레이브 스택을 설계한다.

EtherCAT의 MAC에서는 특정 어플리케이션 동작을 처리하기 위한 산업용 어플리케이션과 Ethernet 기반 디바이스 프로파일을 지원한 위한 표준 TCP/IP 및 UDP/IP 스택이 필요하다.

따라서 [그림 5-2]와 같은 구조를 가지는 EtherCAT 슬레이브 어플리케이션 스택을 구현하여야 한다.



[그림 5-2] EtherCAT 슬레이브 어플리케이션의 구조

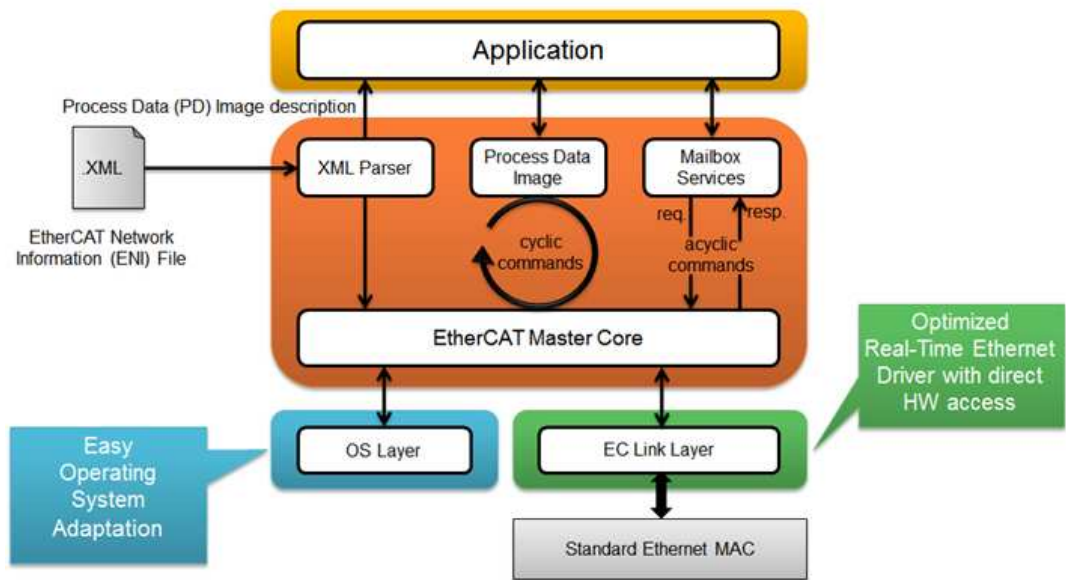
EtherCAT 슬레이브 스택은 <표 5-1>와 같은 파일 구조를 가지고 있다.

<표 5-1> EtherCAT 슬레이브 스택에 구성되어 있는 파일 종류와 기능

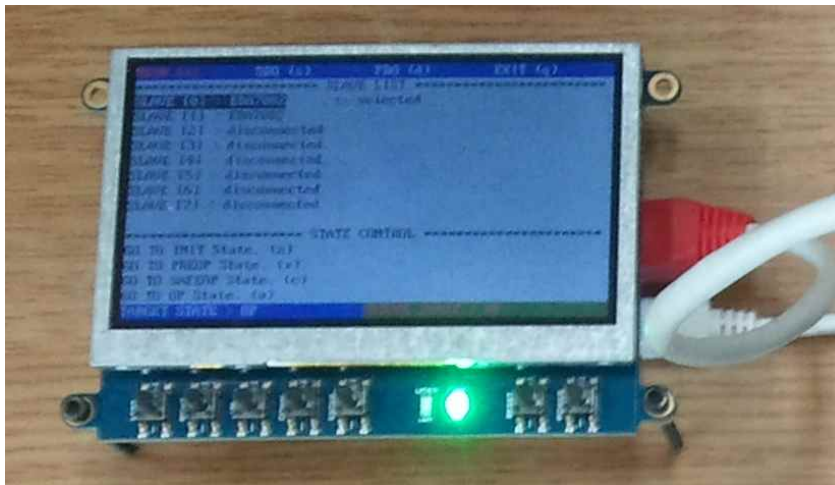
File List	Detailed Description
coeappl.c	CANopen over EtherCAT application interface
ecatappl.c	EtherCAT state machine and process data interface
ecatcoe.c	CANopen over EtherCAT functions
ecatslv.c	EtherCAT state machine
emcy.c	Emergency interface
mailbox.c	EtherCAT mailbox handler
objdef.c	CANopen over EtherCAT object dictionary
sdoserv.c	SDO server
spihw.c	SPI interface

2. EtherCAT 마스터 스택을 구현한다.

일반적으로 EtherCAT 마스터의 경우 상용 EtherCAT 마스터 스택을 이용하여 구현한다. EtherCAT 마스터 스택을 구현하기 위해서는 우선적으로 [그림 5-3]과 같은 00사의 상용 EtherCAT 마스터 스택을 구매한다. 다음으로 [그림 5-4]와 같은 하드웨어 모듈에 RTOS(real-time operating system)을 설치한 후 마스터 스택을 포팅한다. 마스터 스택이 정상적으로 포팅이 된 경우 [그림 5-5]와 같은 내용의 화면을 볼 수 있다.



출처: ○○○(<http://www.acontis.com>). 2016. 9. 30. 스크린샷.
 [그림 5-3] EtherCAT 마스터 동작 구조



[그림 5-4] EtherCAT 마스터 스택 포팅

```

MAIN (a)      SDO (s)      PDO (d)      EXIT (q)
***** SLAVE LIST *****
SLAVE [0] : EDA7002      <--selected
SLAVE [1] : EDA7002
SLAVE [2] : disconnected
SLAVE [3] : disconnected
SLAVE [4] : disconnected
SLAVE [5] : disconnected
SLAVE [6] : disconnected
SLAVE [7] : disconnected

***** STATE CONTROL *****
GO TO INIT State. (z)
GO TO PREOP State. (x)
GO TO SAFEOP State. (c)
GO TO OP State. (v)
TARGET STATE : OP      ACTUAL STATE : OP

```

[그림 5-5] EtherCAT 마스터 스택 포팅 결과

EtherCAT 마스터 스택은 <표 5-2>와 같은 파일 구조를 가지고 있다.

<표 5-2> EtherCAT 마스터 스택에 구성되어 있는 파일 종류와 기능

File List	Detailed Description
ATEMDemoMain.cpp	EtherCat command line을 해석하고 현재 시스템에 적합하게 설정하는 기능
ATEMDemo.cpp	EtherCat 통신을 초기화 하고 Master 동작 기능
ATEMDemoConfig.h	EtherCat 기본 구성에 필요한 변수들을 포함하고 있다.
selectLinkLayer.cpp	EtherCat Command line을 분석하여 Link Layer Parameter로 변환
ecatNotification.cpp	Slave 모니터링 및 에러 검출 기능
ecatDemoCommon.cpp	EtherCat bus 상의 Slave 연결을 찾는 기능 등을 구성.
SlaveInfo.h	사용 가능한 Slave들의 정보들이 저장되어 있다.
Logging.cpp	메시지 처리 기능

3. CiA402 표준 기반의 xml을 설계한다.

IEC 61800 표준은 고속 전기 파워 드라이버(electrical power driver) 시스템의 공통적인 규격을 정의 하고 있는 표준이며 특히, IEC61800-7 표준은 제어 시스템과 파워 드라이버 시스템 사이의 일반적인 표준 인터페이스를 정의하고 있다.

CiA402 표준 기반 xml 설계를 위해 <표 5-3>과 같이 일반 함수와 개체(object)에 적합한 드라이버 프로파일과 다른 통신 인터페이스와 적절하게 매칭되는 것을 목표로 모션 제어의 일반적인 사항을 구현한다.

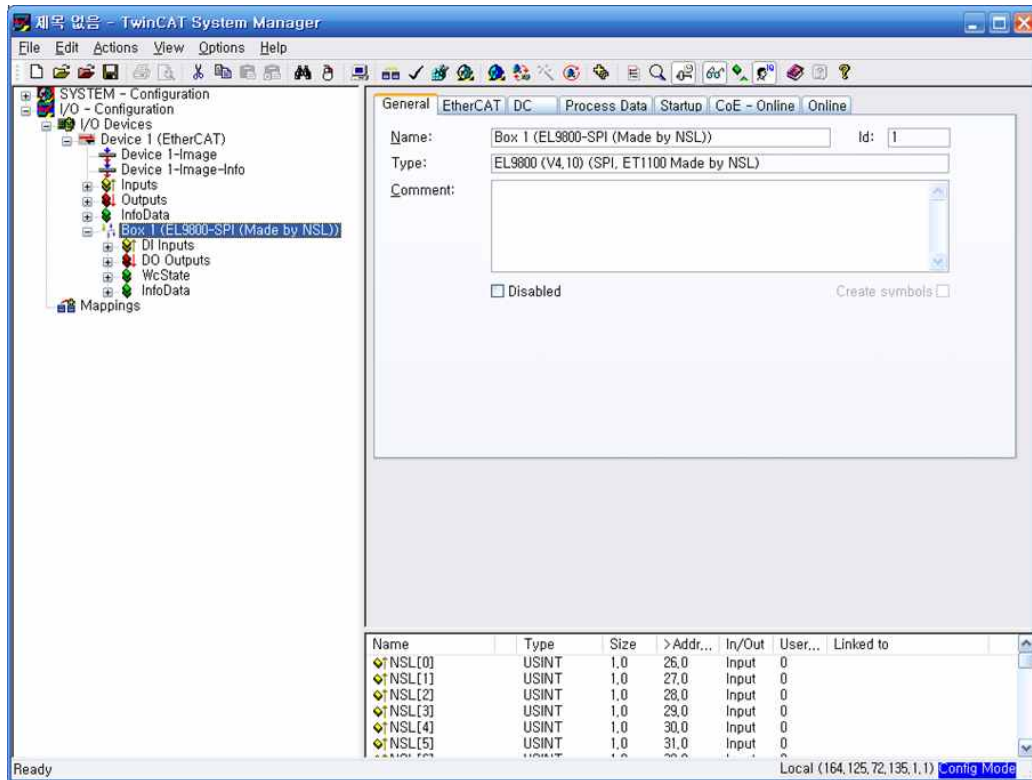
<표 5-3> CiA402 표준 기반 xml의 프레임 구조

Header															
Command Group								Command Header							
r/w	Group Command							Main Command				Sub Command			
msb							lsb	msb				lsb	msb		lsb

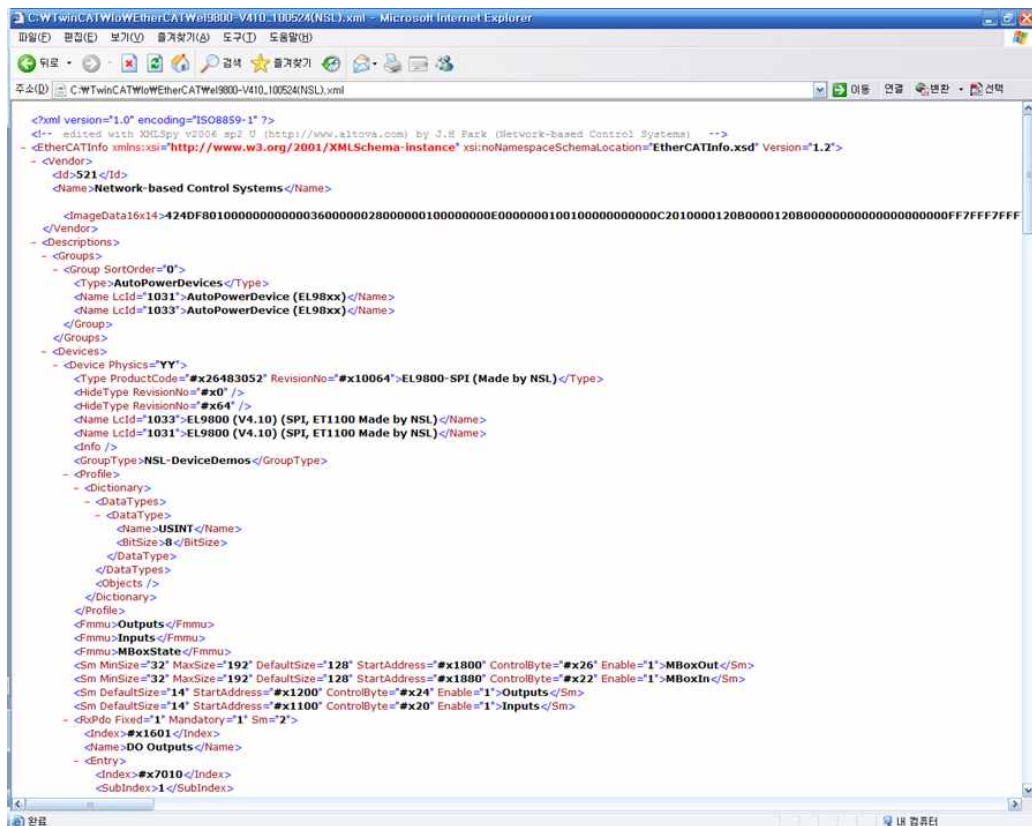
Key Value(K)	Meaning (Main Command)
0	Not Use
1	Request Invalidity (NOP)
2	Request Connect Confirm (CONNECT)
3	Request Disconnect Confirm (DISCONNECT)
4	Request Synchronization Confirm (SYNC_SET)
5	Control Mode Setting
6	Read Parameter (RPM_RD)
7	Write Parameter (RPM_WR)
8	Request Device Set-up
9	Read Alarm, Warning
10	Write Alarm, Warning
11	Read Non-Volatile Parameter
12	Write Non-Volatile Parameter
13	Device Setup Mode
14	Request Sensor OFF
15	Request Coordination Set-up
16	Request Brake Operation ON (Braking)
17	Request Brake Operation OFF
18	Request Motion Stop
19-255	Reserved

4. 상용 EtherCAT master 프로그램인 TwinCAT을 통한 설계된 xml 파일을 이식한다.

[그림 5-6]은 상용 EtehrCAT master 프로그램인 TwinCAT을 나타내며, [그림 5-7]는 xml로 표현된 CiA402 프로파일 규격 및 네트워크 표준 분석을 통한 전체 네트워크의 데이터 Index 정의 및 메시지를 나타낸다.



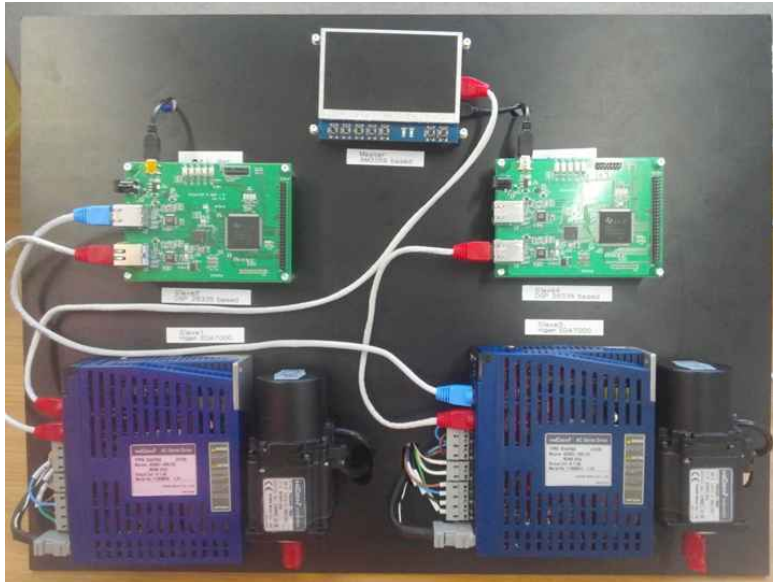
[그림 5-6] TwinCAT을 이용한 EtherCAT 마스터 설정



[그림 5-7] 전체 네트워크의 데이터 Index 정의 및 메시지의 xml 표현

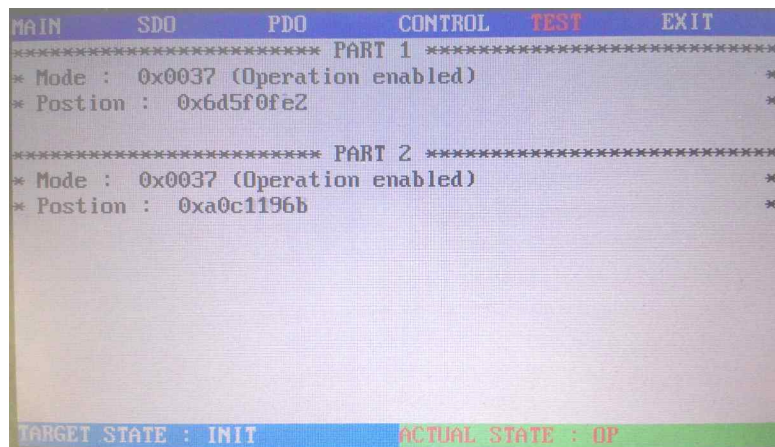
5. 네트워크 기반 로봇 제어를 시험해 보기 위하여 실험 모델을 구성한다.

네트워크 기반 제어 시스템을 테스트해 보기 위하여 두 대의 서보 모터와 EtherCAT 마스터 모듈 및 EtherCAT 슬레이브 모듈로 구성된 [그림 5-8]과 같은 실험 모델을 구성한다. 이러한 형태의 실험 모델은 상용으로 구하기가 쉬우며, 각 부품들을 구매하여 직접 실험 모델을 구성해 볼 수도 있다.



[그림 5-8] 네트워크 기반 로봇 제어 시스템을 위한 실험 모델

[그림 5-9]는 실험 모델을 이용하여 두 대의 모터를 동작시켰을 때의 EtherCAT 마스터에 나타는 화면을 보여준다. 실험 모델에서 모터를 회전시키게 되면, Position의 값이 피드백 되어 변하는 모습을 확인할 수 있다.



[그림 5-9] 모터 제어 실험 결과