우 종 정 교수지도
석사학위 청구논문

WiFi를 이용한 안드로이드 플랫폼 기반 로봇 원격 제어 시스템 설계 및 구현

2011

성신여자대학교 일반대학원
컴퓨터학과
노 성 동

WiFi를 이용한 안드로이드 플랫폼 기반 로봇 원격 제어 시스템 설계 및 구현

우 종 정 교수지도

이 논문을 석사학위논문으로 제출함

2011년 5월

성신여자대학교 일반대학원

컴퓨터학과

노 성 동

인 준 서

노 성 동의 석사학위 논문을 인준함.

심사위원 (인)

심사위원 (인)

심사위원 (인)

성신여자대학교 일반대학원

논 문 개 요

최근에 로봇을 대중화 시키려는 노력들이 이곳저곳에서 볼 수가 있다. 가정 내에서의 청소로봇을 사용한다든지 각 관공서나 전시회에서 정보 이용 로봇으로 많이 활용하고 있다. 하지만 아직도 인간이 제어하거나 활용하는 것은 먼 미래의 일이다. 그만큼 인간이 로봇을 제어하는 데는 여러 불편한 사항이 많이 있는 것이다.

본 연구에서는 이러한 로봇을 좀 더 사용자와 친숙하게 할 수 있는 방법을 찾으려고 하였다. 세계적으로 대중화 되고 있고 되어 있는 스마트폰에 적용을 해보고자 하였다. 스마트폰 중에서도 안드로이드를 이용하는 스마트폰에서 로봇을 자유자재로 이용할 수 있다면 불편한 로봇이용이 그만큼 인간에게 친숙할 수 있을 것이다. 안드로이드는 사용자와 많이 친숙한 운영체제로써 스마트폰뿐만 아니라 휴대용 단말기에서 부각하고 있는 운영체제 중에 하나이다. 구글에서 개발하여 개발 툴(안드로이드 SDK)을 무료 배포를 하고 있어서 다른 비용이 들지 않는 장점도 있다. 여기에 WiFi를 통한 휴머노이드 로봇 제어 애플리케이션 개발(로봇 제어 및 모니터링 할 수 있는 안드로이드 기반 어플)을 중심으로 로봇에 탑재된 센서를 실시간으로 모니터링 할 수 있고 로봇이 가지고 있는 카메라 영상을 모니터링 할 수 있는 애플리케이션 개발을 주요로 하고 있다.

따라서 안드로이드용 로봇 원격 제어 애플리케이션이 개발 완료가 됨으로써 많은 사용자들에게 로봇 제어의 흥미 유발을 할 수 있을 뿐만 아니라 로봇의 친숙함을 줄 수 있는 시스템이다.

목 차

표 목 차

그 림 목 차

I. 서 론

로봇이 만들어져 사용되기 시작한 지 50여 년이 지났다. 최초의 로봇을 포함하여 기존의 로봇은 주로 산업용 분야에서 주로 사용 되었으며 그 일은 사람을 대신해서 무거운 물건을 옮기거나 조립하는 단순 반복 작업을 되풀이하는 것이 전부였다[1]. 즉, 자동차 조립 라인에서 프레임을 옮기고 용접하거나, 전자 제품 생산에서 부품을 납땜하는 주로 정해진 프로그램을 따라서 작업을 하는 경우가 대부분이었다. 따라서 기본에 로봇이라는 말은 이러한 로봇을 뜻하는 것으로 인식되었다.

최근에는 경제 및 기술 발전에 힘입어 앞서 단순 로봇을 뛰어넘어 다양한 형태와 다양한 기능을 가지는 로봇들이 소개 되고 있다. 1999년 선보인 소니사의 강아지 로봇 아이보[2]를 시작으로 2000년 혼다사의 휴머노이드형 로봇 아시모[3]가 소개 되면서 로봇이 사람들의 많은 관심을 받게 되었다. 그리고 최근에는 일부 로봇의 보급이 이루어져 여러 회사에서 다양한 기능이 있는 청소로봇을 판매하고 있는 실정이다. 이렇듯 인간 생활의 편리함을 위해 여러 분야에서 로봇이 요구되고 있으며, 단순한 편리성을 제외하고 교육, 오락적인 여러 측면이 부각되면서 그러한 기능을 가진 로봇에 대한 관심도 높아지고 있다.

이에 따라 요즈음 로봇이라고 하면 청소나 스케줄 관리와 같은 복합 기능을 할 수 있는 서비스 로봇을 생각 하게 되고 영화나 애니메이션에서만 볼 수 있던 생각을 하고 여러 가지 일을 할 수 있는 로봇을 기대하기에 이

르렀다. 그러나 아직은 이러한 로봇들이 사람들이 원하는 수준에 크게 못 미친다[4].

본 연구는 이러한 사용자와의 친숙함이 없는 것에 착안하여 사용자와 로봇이 친숙하면 어떨까에 중점을 두고 개발을 하였다. 먼저 로봇을 관리 및 동작을 안드로이드 기반 애플리케이션을 개발하여 폭발적인 인기를 끌고 있는 안드로이드 폰에서 구현하였다.

이러한 안드로이드 플랫폼을 실제 스마트 폰이 아닌 교육용 S5PV210 기반 교육용 모바일 기기에 탑재 뿐 아니라 이를 이용 Metal Fighter(MF)라는 로봇을 컨트롤하는 시스템 설계 및 구현에 대한 것을 연구하였다.

II. 배경연구

1. URC

URC(Ubiquitous Robot Companion)는 언제 어디서나, 나와 함께 하며, 나에게 필요한 서비스를 이음새 없이 제공하는 지능형 로봇 서비스를 총칭한다. 즉 로봇에 네트워크를 활용하여 로봇이 제공할 수 있는 응용 서비스를 확장하고, 로봇이 모든 기능을 자체적으로 가짐으로써 안게 되는 기술적인 제약성이나 비용 상의 문제를 네트워크를 통해 기능을 분담하는 것이다[5]. 이것이 구현되기 위해서는 기존의 로봇 자체 기술인 메카트로닉스 기술 이외에도 추가적인 정보통신기술을 요구한다.

이러한 추가적인 정보통신기술을 안드로이드 폰에서의 로봇의 제어를 함으로써 로봇과의 친숙함을 가질 수 있다.

2. 기기별 로봇 제어

로봇을 제어하기 위한 방법은 많은 방법이 있다. 그 중 PC와 휴대폰으로 제어 할 수 있는 방법이 있다. PC는 PC에 별도의 소프트웨어를 설치하여 단순히 인터넷 홈페이지 접속만으로 원격지에 있는 로봇을 자신이 직접 제어하면서 로봇에 달린 카메라의 영상을 볼 수 있는 인터넷용 원격지 이동 로봇이 개발되었다. 이 인터넷 원격지 영상전송 이동 로봇은 내외 웹 카메라 시장이 후끈 달아오를 때 출시되어 단순히 보기만 하던 인터넷 실시간

카메라의 기능에서 벗어나 직접 홈페이지에 접속해서 로봇에게 전진, 후진 등의 명령을 내리면서 로봇의 눈을 통해 영상을 보게 된다는 획기적인 발상 아래 개발하게 되었다. 이 로봇을 이용하면 로봇제어를 통해 미국에 있는 해변을 걸어 다니면서 주변을 구경한다든지 공장이나 주택, 거리를 돌아다니면서 구경도 할 수 있는 그야말로 인터넷 가상공간이 아닌 인터넷 실시간 공간으로의 탈바꿈의 한 예라고 볼 수 있을 것이다[6]. 로봇의 영상이나 제어는 모든 게 무선으로 제어된다. 하지만 이것도 모바일에는 한계를 들어낼 수밖에 없었다.

2009년 4월 KTF가 세계 최초로 휴대폰으로 원격 조정할 수 있는 로봇청소기를 선보였다. 마이크로로봇과 협력해 영상통화를 통해 원격조정을 할 수 있는 영상통화 로봇청소기를 출시한다고 밝혔다. 로봇청소기에 영상통화 기능을 갖춰 쇼(SHOW) 휴대폰만 있으면 언제 어디서든지 원격으로 조정할 수 있다. 영상통화가 되는 휴대폰이면 이용이 가능하며 로봇청소기로 영상전화를 건 후 화면을 보며 휴대폰 버튼을 이용하여 청소기를 전후좌우로 조정할 수 있다. 동작 정지나 충전, 전체자동청소 등도 가능하다[7].

이처럼 휴대폰으로 로봇을 제어하여 방문객 영상 확인은 물론 세대 간 화상통화, 휴대폰 및 외부 인터넷 원격 제어, 조명, 가스밸브 상태, 에너지 사용량의 조회까지 가능하게 만들었다. 이 모든 것이 홈 네트워크 시스템이라 할 수 있다[8]. 하지만 이것도 한계가 있었다. 휴대폰 단말기의 성능의 한계로 인한 기대효과에 미치지 못하였다.

위의 PC 및 휴대폰으로 로봇의 제어는 한계를 나타내고 있다. 그래서 선택한 것이 성능의 고급화와 모바일의 대세인 스마트폰으로 로봇을 제어하고 스마트폰에는 iOS 또는 안드로이드 등이 있는 데 그 중 안드로이드 플

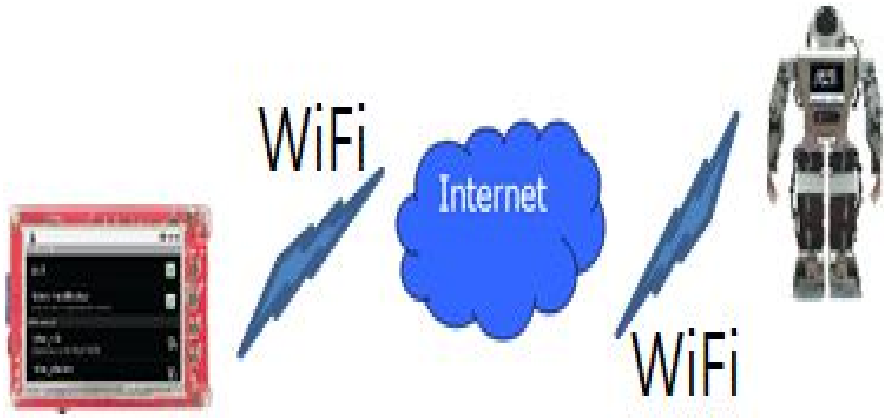
랫폼이 탑재된 안드로이드 폰으로 로봇을 제어하면 유저와 친숙하고 성능 또한 월등할 것이라 예상하였다.

구글에서 안드로이드 플랫폼을 발표하면서 많은 개발자들의 관심이 안드로이드에 쏠렸다. 구글에서 정의한 안드로이드는 운영체제, 미들웨어, 응용 프로그램을 포함하고 모바일 디바이스를 위한 소프트웨어 스택이라고 정의한다[9]. 모바일 플랫폼이란 모바일 기기에 이식 할 수 있는 S/W로서, 웹 상에서 큰 영향력을 가진 구글의 첫 모바일 플랫폼이면서 기존에 존재하던 모바일 플랫폼과는 차별적인 모습으로 사용자와 개발자 모두에게 큰 관심을 받고 있다.

III. 로봇 원격 제어 시스템

1. 시스템 구성

시스템 구성은 아래의 그림과 같다. 안드로이드 플랫폼이 탑재된 H-AndroSV210 이 무선 네트워크를 통하여 로봇(Metal Fighter)을 제어하는 것이다. 즉 안드로이드 플랫폼 기반의 응용 프로그램을 제작하여 안드로이드 플랫폼이 탑재된 스마트폰 대용 기기에 탑재 및 실행하여 WiFi를 이용하여 언제어디서나 로봇을 제어 할 수 있게 구현하였다.



[그림1] 시스템 구성

1-1. 로봇(Metal Fighter)의 구성

MF 로봇의 구성은 [표1]과 같다.

품 명	내 용
Embedded Module	Processor(Marvell PXA320-P(Core : 806MHz))
메모리	266MHz DDR SDRAM 128MByte, NAND Flash 128MByte
무선랜	Wireless LAN Module
블루투스	Bluetooth Module
3축가속	3축 가속 센서
카메라	Camera 1.3M Pixel
오디오/USB	Audio(IN,OUT)/Host 1port/Slave 1port
시리얼	RS232 2port/TTL 1port
AD/JTAG	AD port 4EA/JTAG 1port
LED/스위치	LED 3EA/스위치 3EA
관절	17관절 2족 보행 로봇
제어모터	MRS-D2009SP x 17EA
LCD	Color LCD

[표1] Metal Fighter Specifications

1-2. 안드로이드 플랫폼 기반 실험용 보드(H-AndroSV210)의 구성

H-AndroSV210 의 구성은 [표2]와 같다.

품 명	내 용
Case	고급플라스틱케이스 / 아크릴오픈프레임
크기	140 x 90 x 23 (mm)
CPU	Samsung S5PV210 (Cortex-A8)
Memory	DDR2 SDRAM 512MByte / Nand Flash 256MBytes / SDcard지원(2GB제공)
DisPlay	4.8인치 TFT LCD(800 x 480) / HDMI OUTPUT
운영체제	안드로이드 2.2(Froyo)
무선통신	Wi-Fi, Bluetooth
입력방식	Touch Screen / 저항막 방식
Sensor	3axis Acceleration Sensor, Magnetic Sensor, GPS(SiRF)
전원	리튬-이온 베터리(1900mAh) 사용 / USB충전 / 5V아답터
Audio	ALC5622(I2S) - mic / speaker and ear jack
Camera	CMOS Camera(150만 화소)
USB	USB 2.0Host 1 port, USB 2.0 OTG 1port
LED	Charging:1EA, App:3EA
KEY	7EA

[표2] H-AndroSV210 Specifications

1-3. 제어 시나리오

서론에서 기술했던 것처럼 로봇과 사용자와의 친숙함이 없는 로봇들이 산업용으로써 많이 사용이 되고 친숙하다고 할 수 있는 로봇들조차 사용자가 제어하는 것이 그 다지 용이 하지 않기에 안드로이드 플랫폼이 탑재된 스마트 폰에서 로봇을 제어한다면 로봇이 사용자와 친숙할 수 있을 것이란 생각에 이 시스템과 사용자 애플리케이션을 만들게 되었다. 각 시스템의 스펙들은 위의 시스템 구성에 나열 한 것과 동일한 시스템을 이용하였다. 그리고 개발 내용은 Wireless LAN을 통한 휴머노이드 로봇 제어 애플리케이션 개발(메탈파이터를 제어 및 모니터링 할 수 있는 안드로이드 기반 어플)을 중심으로 메탈파이터의 센서를 실시간으로 모니터링 할 수 있고 메탈파이터의 카메라 영상을 모니터링 할 수 있는 애플리케이션 개발을 주요로 하고 있다.

개발에서 사용되어 지는 타겟보드는 H-ArdroSV210를 이용하고 로봇은 H-MF-17AI(이하 메탈파이터)이다. 메탈파이터에 기본으로 탑재되어 있는 Wireless LAN과 H-ArdroSV210의 1Wireless LAN을 이용하여 무선으로 동작을 제어할 수 있다. 이 응용 프로그램은 기존의 단순히 장치에서만 구동되는 정보를 제공하거나 장치내의 센서 및 GPS등의 H/W를 제어하는 것과 달리 외부 장치와 연결하여 제어함으로써 외부 장치의 제어 방식을 이해할 수 있도록 구성되어 있다.

로봇 제어 응용 프로그램의 특징은 다음과 같다.

• 메탈 파이터의 동작 제어는 물론 Camera영상, 센서값 등을 프로그램 상에서 확인할 수 있다.

• 메탈파이터에 사용되는 프로그램은 C기반의 Socket Server 프로그램으

로 H-AndroSV210의 Java기반의 Client프로그램과 통신한다.

• UDP 기반의 Socket 프로그램으로 구성되어 무선 환경에서 사용할 수 있다.

• 무선 환경에서의 안정성을 고려하여, 연결이 끊어지더라도 자동으로 재접속을 하여 안정적인 사용 환경을 제공한다.

2. 안드로이드 개발

안드로이드의 개발은 크게 게임, 플레이어 등을 개발하는 애플리케이션을 작성하는 부분과 커널, 파일 시스템, 디바이스 드라이버 등을 타겟에 포팅하는 부분으로 나눌 수 있다.

3. 애플리케이션 개발 환경 구축

안드로이드 개발환경 구축에 있어서는 이클립스라는 툴을 이용해 안드로이드 관련 어플리케이션을 작성하고 에뮬레이터를 통해 동작을 확인 할 수 있도록 구성되어있다. 기본적으로 eclipse와 Android emulator 그리고 Java SE Development Kit(JDK 5.0이상)의 버전이 필요하다.

4. Real Target 개발 환경 구축

Android 용 Real Target을 포팅하기 위해서 준비해야 하는 Arm 컴파일용 툴체인 설치와 Busybox컴파일을 통하여 파일 시스템을 구성하기 위

한 준비 작업을 하고 H-Andro210(S5PV210 기반 교육용 모바일 타겟 보드) 타겟보드에 적용될 커널을 컴파일 하여 본다.

IV. 로봇 원격 제어 시스템 구현 및 결과

1. 원격 로봇 제어 시스템의 개발 환경

로봇 제어 응용 프로그램을 개발 하기 위해서는 다음과 같은 개발 환경이 필요하다. 먼저 로봇 시스템은 위하여 리눅스 기반의 메탈파이터의 프로그램 개발을 해야 하며 다음과 같은 사용 환경 구축이 되어 있어야 한다.

- 리눅스가 설치되어 있는 PC(VMware사용 가능)
  - USB 1Port
  - Serial 1Port(USB to Serial 사용가능)
- arm-lunux-gcc 크로스 컴파일러
- RoboBASIC 프로그램(메탈파이터)

위의 환경이 정상적으로 구축이 되어 있어야 메탈 파이터에서 구동되는 프로그램의 개발이 가능하다.

또한 로봇을 제어하기 위한 스마트폰용 컨트롤러(Android 기반 플랫폼 : H-AndroSV210)가 필요하다. 안드로이드 기반의 플랫폼에 MF_Android 프로그램을 개발하기 위해서는 다음과 같은 사용 환경이 구축되어 있어야 한다.

- Android 기반의 플랫폼(Android 1.6이상)
- Android 개발이 가능한 PC
- USB 1Port

- Serial 1Port(USB to Serial 사용가능)

• Android SDK

• Java SDK(JDK)

• Eclipse

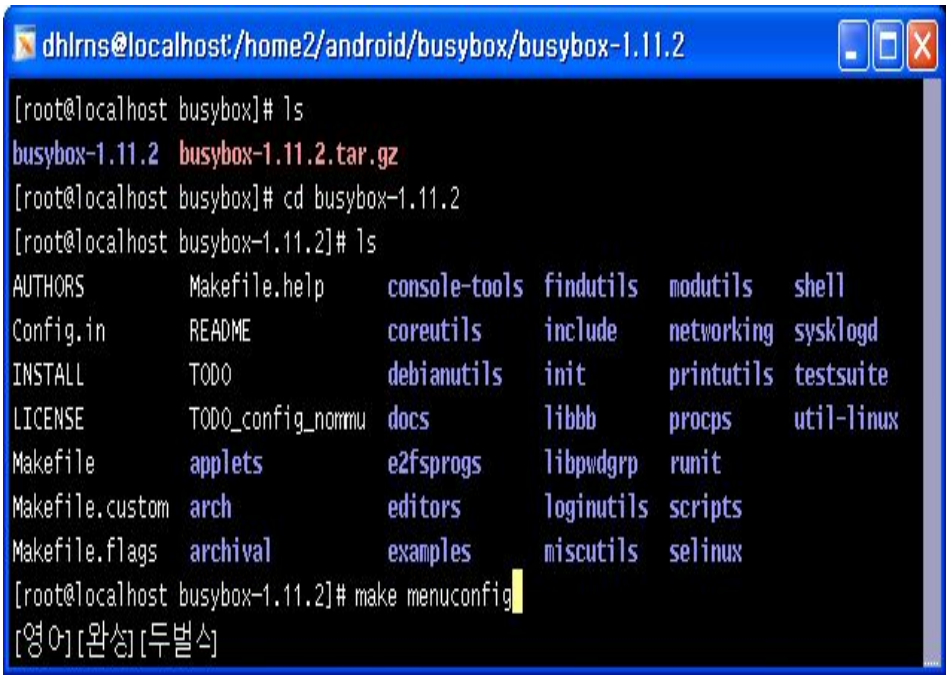
안드로이드 기반의 프로그램을 개발하기 위해서는 위와 같은 개발 환경이 구축되어 있어야 한다.

2. 안드로이드 플랫폼

구현에 사용된 시스템은 삼성 계열의 S5PV210 MCU, 128Mb NAND flash, 128Mb DDR2, Touch가 가능한 7인치 TFT LCD 등으로 구성되어 있다. 우선 애플리케이션 제작을 하기 위한 환경 구축 한 이후에 안드로이드 SDK 와 개발 툴인 이클립스를 플러그 인 하여 하나의 애플리케이션을 만든 후에 시뮬레이터를 실행하여 리얼 타겟에 탑재할 파일 시스템의 이미지를 작성하였다.

Real Target에서 사용할 커널 및 파일 시스템을 빌드하기 위한 크로스 컴파일러인 toolchain 을 다운로드하여 설치한다. 그리고 Busybox를 다운로드하여 압축을 푼 후 컴파일하여 사용한다. Busybox는 Android Emulator에서 파일시스템에 관련한 디렉토리를 이미지 형태로 추출하기 위하여 사용된다. 일반적으로 Busybox는 방대한 표준 리눅스 커맨드 라인 유틸리티를 대체할 수 있는 작은 효율적인 툴이다. 또한 제한된 자원을 가진 임베디드 플랫폼에서는 기반 툴로 사용되기도 한다. Busybox는 대부분의 데스크탑 리눅스나 임베디드 리눅스 배포 버전에서 볼 수 있는 작은 크기의

프로그램이다. 그리고 안드로이드용 커널을 다운로드 하여 컴파일 하여 사용한다. 안드로이드용 커널은 구글에서 배포를 하고 있다. 마지막으로 이들을 컴파일하여 나온 결과물들을 타겟보드에 포팅하여서 사용한다. 자바로 이루어진 안드로이드 애플리케이션을 타겟보드에 탑재하여 사용하기도 한다.



[그림2] Busybox 설치

[그림3] 안드로이드를 타겟 보드에 탑재한 모습

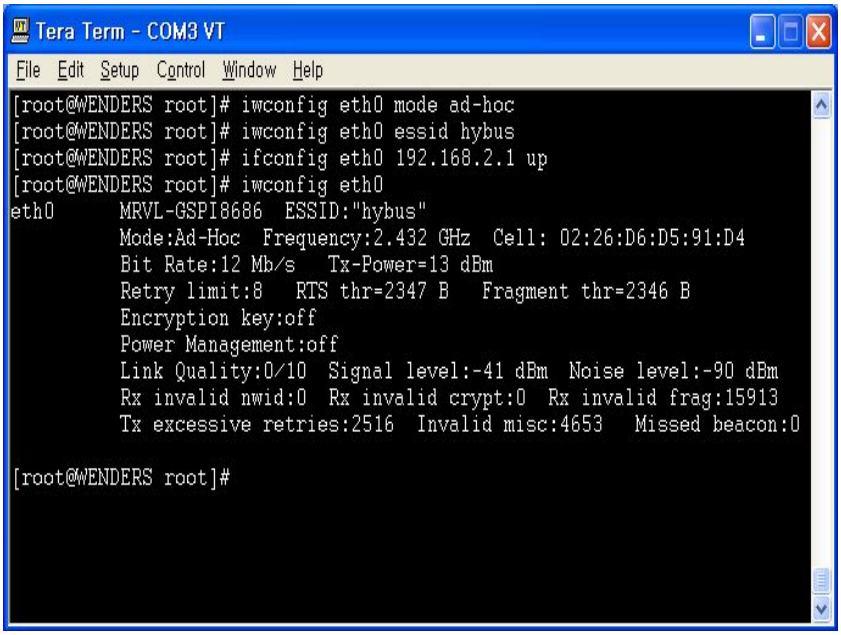
3. 무선랜 환경 설정

메탈 파이터 H-AndroSV210은 무선랜을 사용하여 통신을 하기 때문에 프로그램 구동에 앞서 무선랜에 관한 설정이 되어 있어야 한다. 여기서는 무선랜에 대한 설정 방법에 대해서 알아보도록 한다.

메탈 파이터를 디버깅 보드를 통해 PC의 Serial포트에 연결한다. 연결을 완료하였으면 전원을 켜고 부팅이 완료되면 다음과 같이 입력한다.

```
☞iwconfig eth0 mode ad-hoc
☞iwconfig eth0 essid hybus
☞ifconfing eth0 192.168.2.1 up
☞ifconfig eth0
```

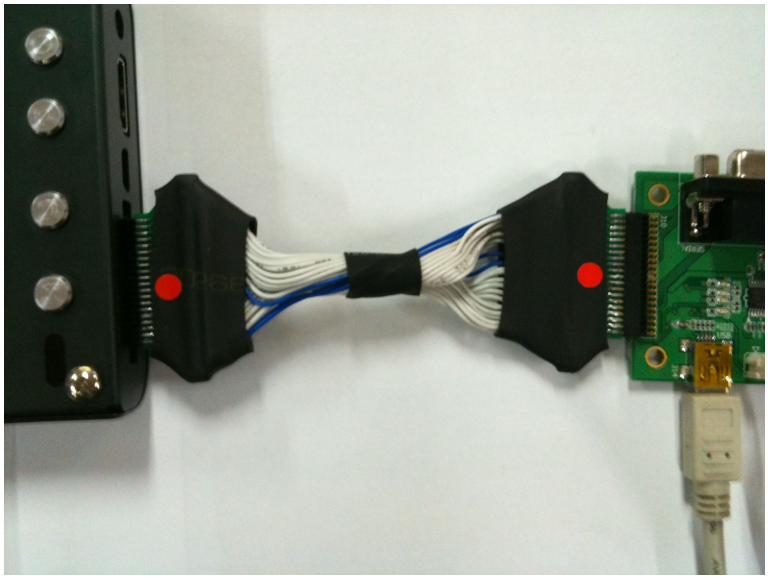
[그림4] ip 설정 명령문



[그림5]ip 설정 화면

위의 설정은 메탈파이터의 무선랜 설정을 한 내용이다. iwconfig 명령을 이용하여 무선랜장치(eth0)의 mode와 essid를 설정하고 IP를 192.168.2.1로 설정한다.

시리얼 콘솔 메시지를 확인하기 위해서 H-AndroSV210 과 디버깅 보드를 다음과 같이 연결한다.



[그림6] 디버깅 보드 연결

H-AndroSV210의 전원을 켜고 완전히 부팅이 되면 다음과 같이 입력한다.

```
☞iwconfig eth0 mode ad-hoc
☞iwconfig eth0 essid hybus
☞iwconfig eth0 channel 5
☞ifconfing eth0 192.168.2.2 up
☞ifconfig eth0
```

[그림7] essid 설정 명령문

위의 메탈파이터의 설정과 마찬가지로 H-AndroSV210의 무선랜장치(eht0)의 mode와 essid를 설정하고 사용할 channel 을 설정 후에 IP를 192.168.2.2로 설정한 내용이다.

위의 설정으로 메탈파이터와 H-AndroSV210이 정상적으로 연결되었는지 확인하기 위해서 ping테스트를 진행한다. ping 테스트를 진행하여 ping을

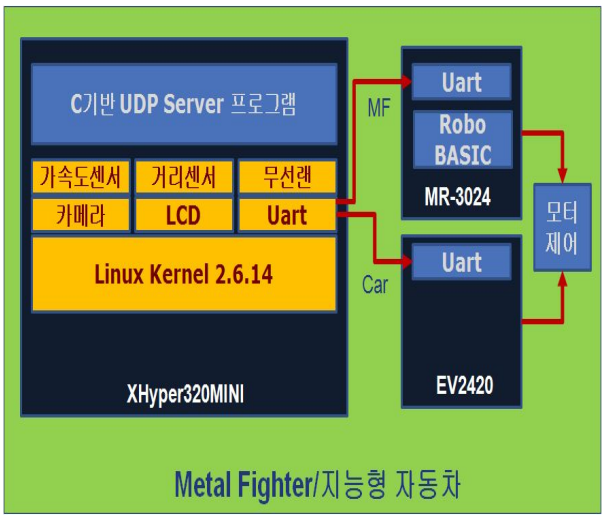
주고 받는 것을 확인했다면 메탈파이터와 H-AndroSV210의 연결이 정상적으로 된 것이다.

4. 메탈파이터 Server 프로그램

MF_Android 프로그램에서 메탈파이터 프로그램인 Sever프로그램을 컴파일하고 다운로드 하여 실행하는 방법에 대해서 알아보도록 한다.

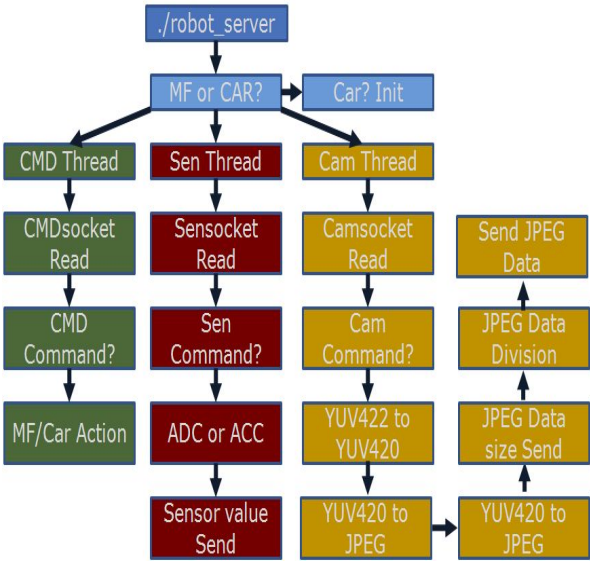
4.1. 서버 시스템의 전반적 구성

[그림8]은 메탈파이터에서 프로그램이 구동되는 내용을 블록도로 나타낸 것이다.



[그림8] 프로그램 블록도

메탈파이터의 시스템은 위와 같이 구성되어 있으며 X-Hyper320MINI보드의 FFUART와 MR-3024으로 구동하게 된다.
다음은 메탈 파이터 에서 구동되는 프로그램의 플로어차트이다.



[그림9] 프로그램 플로어 차트

메탈파이터의 프로그램은 크게 3개의 Thread로 구성되어 동작하며 3개의 Thread는 각각 Socket통신을 하기 위한 내용으로 구성되어 있다. 프로그램이 실행되면 현재 프로그램이 실행되는 장치가 메탈파이터인지 판단하여 프로그램이 실행되고 있는 장치에 맞도록 프로그램이 구성되어 구동된다.
[그림9]에서 CMD Thread는 메탈파이터를 구동시키기 위한 커맨더 메시지를 받기 위한 내용으로 구성되어 있으며 Client에서 커맨드 메시지가 전

송될 경우 메탈파이터를 구동시킨다. Sen Thread는 메탈파이터에 있는 거리센서와 가속도 센서의 값을 전송하기 위한 내용이다. Client에서 센서값을 요구하는 메시지가 전송될 경우 거리센서인 ADC 또는 가속도센서인 ACC값을 전송하게 된다. Cam Thread는 메탈파이터의 Camera 데이터를 전송하기 위한 내용이다. Client에서 Camera 데이터 요구 메시지가 전송되어지면 Camera영상을 캡처하여 캡처된 YUV422영상 포맷을 YUV420으로, 다시 YUV420영상을 JPEG로 변환하여 데이터를 전송하게 된다. 이는 Camera 데이터의 사이즈를 줄여서 전송되는 시간을 절약하기 위함이다.

4.2. 프로그램 구성

메탈파이터에 사용되는 프로그램의 main은 다음과 같다.

```
int main(int argc, char *argv[])
{
    struct sockaddr_in cmd_address;
    struct sockaddr_in cam_address;
    struct sockaddr_in sen_address;
    int status;
    /*---------- user select -----------
    if(argc == 1){
        printf("MetalFighter : %s mf₩n", argv[0]);
        printf("    AI CAR     : %s car₩n", argv[0]);
        exit(1);
    }
                    .
                    .
                    .

    ///* Auto Select MF/CAR
    check_target();
    //--------------------*/
    uart_fd = open_serial();
    init_camera();
    if(car_ctrl == 1)
                car_init();

    /* CMD socket address setting */
    cmd_sockfd = socket(PF_INET, SOCK_DGRAM, IPPROTO_UDP);
    cmd_address.sin_family = AF_INET;
    cmd_address.sin_addr.s_addr = htonl(INADDR_ANY);
    cmd_address.sin_port = htons(CMD_PORT);

    /* CAM socket address setting */
    cam_sockfd = socket(PF_INET, SOCK_DGRAM, IPPROTO_UDP);
    cam_address.sin_family = AF_INET;
    cam_address.sin_addr.s_addr = htonl(INADDR_ANY);
    cam_address.sin_port = htons(CAM_PORT);

                    .
                    .
                    .
    pthread_join(cmd_ThreadID, (void **)&status);
    pthread_join(cam_ThreadID, (void **)&status);
    pthread_join(sen_ThreadID, (void **)&status);

    return 0;
}
```

[그림10] 메탈파이터 Server 프로그램

프로그램이 실행되는 구조는 크게 다음과 같다.

- check_target() : 실행되는 장치가 메탈파이터인지 판단한다.
- open_serial() : Serial포트를 Open한다.
- init_camera() : Camera를 초기화한다.
- cmd_socket, cam_socket, sen_socket을 생성한다.
- cmd, sen, cam Thread가 시작된다.

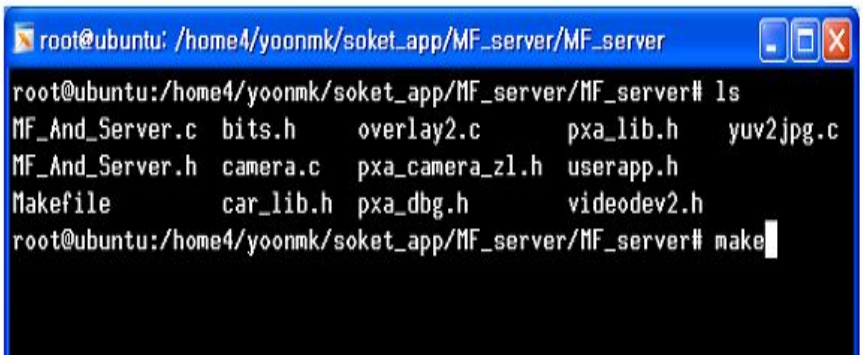
프로그램은 위와 같은 구조로 되어 있으며 각 각에 대한 세부적인 내용은 해당 소스를 참고하기 바란다.

4.3. 프로그램 컴파일 및 실행

MF_Android의 Server프로그램인 메탈파이터의 프로그램은 그림과 같이 구성되어 있으며 프로그램의 main을 포함하고 있는 MF_And_Server.c와 기타 헤더파일, 라이브러리 파일을 포함하고 있다.

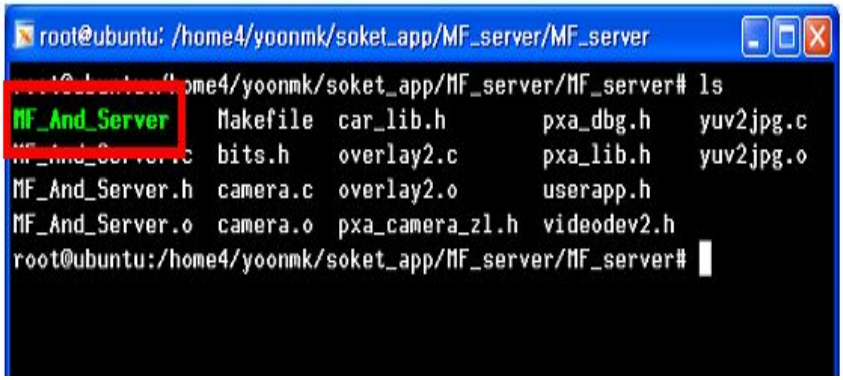
“make"명령을 이용해서 프로그램을 컴파일 한다.

[그림11] MF_server 프로그램 파일



[그림12] 컴파일

컴파일이 완료되며 다음과 같이 MF_And_Server 라는 파일이 생성된 것을 볼 수 있다.



[그림13] 컴파일 후 파일 생성

위에서 컴파일된 MF_And_Server 바이너리 파일을 tftp 또는 zmodem을 통하여 메탈파이터에 다운로드 한다. 프로그램을 실행하기에 앞서 메탈파이터에서 실행할 경우에는 메탈파이터의 모드변환 스위치가 AI제어 모드로 설정이 되어 있어야 정상적인 동작이 가능하다.

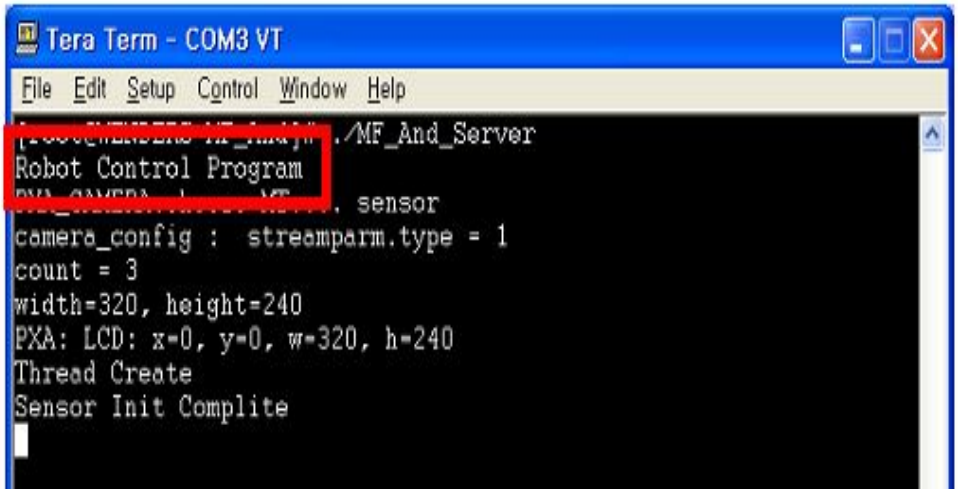
[그림14] 통신 포트 설정

위의 과정이 완료되면 MF_And_Server 프로그램을 실행한다.

```
☞./MF_And_Server
```

[그림15] MF 안드로이드 Server 실행 명령문

프로그램이 실행되면 메탈파이터에서 Camera의 정지 영상이 LCD에 출력되며 시리얼 출력화면은 다음과 같다.



[그림16] 시리얼 출력 화면

LCD의 정지영상이 출력되는 것은 H-AndroSV210이 연결되지 않았을 때의 현상으로 연결되어지면 정상적으로 현재 영상이 LCD에 출력되어 진다.

5. H-AndroSV210 Client 프로그램

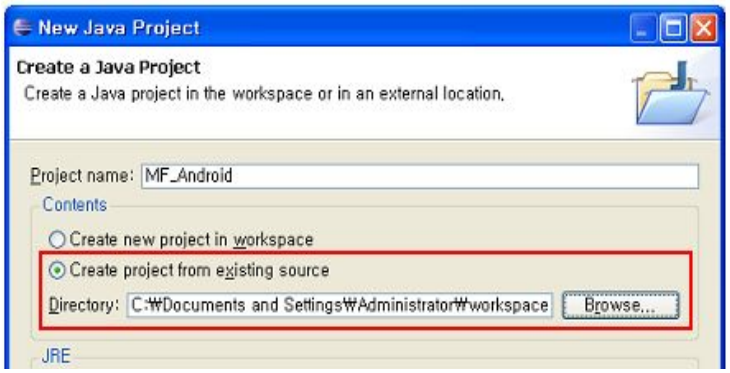
MF_Android 프로그램에서 안드로이드 플랫폼Client 프로그램을 컴파일하고 다운로드 하여 실행하는 방법에 대해서 알아보도록 한다. 프로그램을 컴파일 하고 다운로드 하기 위해서는 위에서 설명한 Client 프로그램에 대한 환경 구축이 되어 있어야 한다.

5.1 Client 프로그램 다운로드 및 실행

(1) 이미 작성되어 있는 MF_Android 프로젝트를 Eclipse에 추가한다. 이클립스를 실행하여 File → New → Java Project를 선택한다.

[그림17] 프로젝트를 선택 화면

(2) 다음으로 나타나는 New Java Project 창에서 “Contents"항목의 "Create project from existing source"를 체크하고 MF_Android 프로젝트가 있는 경로를 지정한다.



[그림18] 프로젝트 경로 지정 화면

(3) "Finish"버튼을 클릭하면 다음과 같이 Navigator창에 위에서 추가한 MF_Android의 프로젝트가 추가된 것을 볼 수 있다.

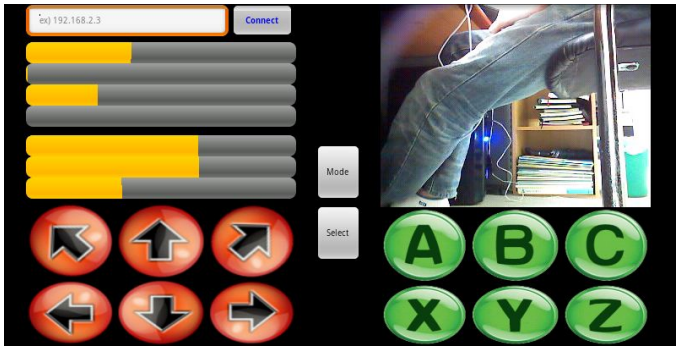
(4) Navigator 창에서 MF_Android를 선택한 후 "Run → Run"을 선택하여 프로젝트에 대한 빌드를 진행한다.

(5) MF_Android 프로그램을 안드로이드 플랫폼 장비로 설치하기 위해서 "Run → Run Configurations..."를 선택한다.

(6) Debug Configurations 창에서 MF_Android를
선택한 후에 Target항목에서 Manual을 선택한 후 Run 버튼을 클릭한다.

(7) 현재 Android 프랫폼과 PC가 연결되어 있다면 다음의 화면과 같이 연결된 장비의 이름이 나타난다. 여기서는 H-AndroSV210과 연결된 모습이다. 해당 장치를 선택하고 "OK"를 클릭한다.

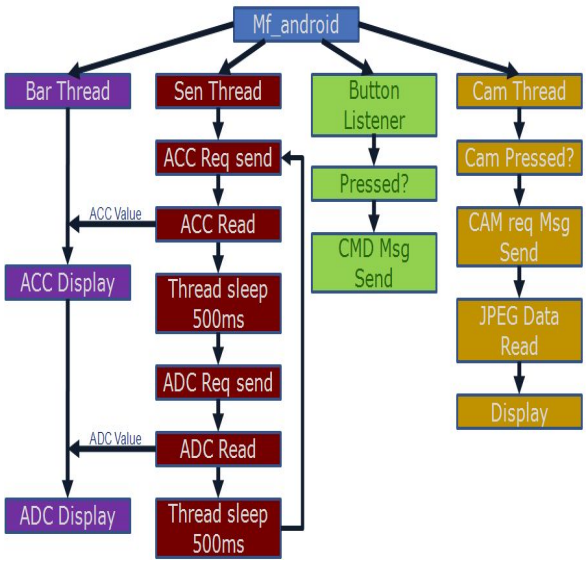
(8) MF_Android 프로그램이 장비에 다운로드 됨과 동시에 H-AndroSV210에서 프로그램이 실행되는 것을 볼 수 있다.



[그림19] 프로젝트를 실행 화면

5.2 Client 프로그램 구성

[그림20]은 H-AndroSV210에서 구동되는 프로그램의 블록도 이다.



[그림20] 프로그램 플로어 차트

H-AndroSV210에서 구동되는 프로그램은 크게 4개의 Thread로 구성되어 동작하며 Bar Thread를 제외한 3개의 Thread는 각각 Socket통신을 하기 위한 내용으로 구성되어 있다.

Bar Thread는 센서값을 표시하기 위한 ProgressBar의 상태 표시와 연결 확인을 위한 Connect버튼 디스플레이, 그리고 Camera데이터 출력을 위한 Camera 디스플레이를 하기 위한 내용으로 구성되어 있다. Sen Thread는 일정주기(500ms)마다 한번씩 Server에 센서 데이터 출력을 위한 센서 데이터 요구 메시지를 전송한 후 Server에서 전송되는 센서값에 대한 데이터를 받는다. 센서값은 ADC와 ACC의 데이터를 일정주기마다 교대로 요청하게 된다. Button Listener는 프로그램에서 버튼이 눌렸을 때의 이벤트를 감지하여 해당 동작을 수행하도록 구성되어 있다. 버튼이 눌려졌을 경우 해당 버튼에 대한 데이터를 Server로 전송하게 된다. Cam Thread는 연속해서 Cam Socket을 통하여 어떠한 데이터를 전송한다. 이 데이터가 전송됐을 경우 Server인 메탈파이터에서 Client와의 연결을 감지하고 Camera 데이터가 메탈파이터의 LCD에 연속해서 출력되어진다. 만약 Camera 영상 전송을 요구하는 버튼이 눌려졌을 때에는 Server로 Camera 데이터 전송 요구 메시지를 전송하여 Camera데이터를 Server로부터 전송받을 수 있다.

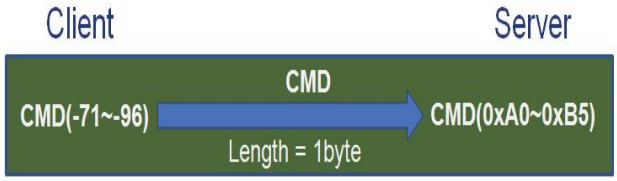
5.3. MF <-> H-AndroSV210 데이터 전송 형식

여기서는 메탈파이터와 H-AndroSV210에서 데이터를 주고받는 내용에

대해서 알아보도록 한다. 일단 위에서 언급한 것처럼 메탈파이터의 Server 프로그램과 H-AndroSV210의 Client 프로그램은 크게 3가지의 데이터를 주고받는다. Command, Sensor, Camera의 3가지 데이터이다. Server와 Client의 데이터 송수신에 대한 내용은 다음과 같다.

5.3.1. Command

Command 데이터는 일방적으로 Client에서 Server로 데이터를 전송하는 형태로 구성되어 있다. Command 데이터를 1byte의 데이터를 가지며 Client에서 버튼이 클릭되었을 때 Server로 Command 데이터를 전송하여 이 Command 데이터를 받은 Server에서는 Command에 맞는 동작을 하도록 구성되어 있다. 다음은 위에서 설명한 내용을 표시 한 것이다.

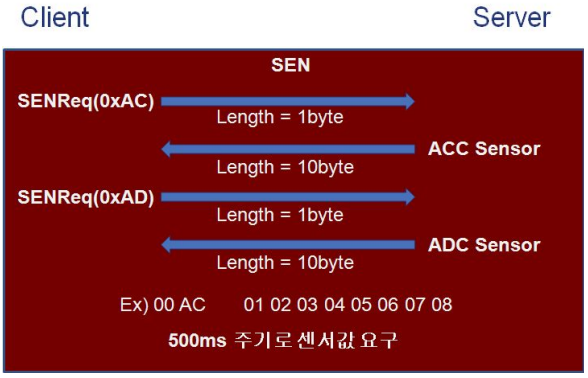


[그림21] Command 전송 형식

5.3.2. Sensor

Client에서 Sensor 데이터 전송 요구 메시지가 Server에 전송될 경우 Server에서는 센서의 데이터를 Client로 전송하게 된다. 센서 데이터 요구 메시지는 1byte로 구성되어 있으며 1byte의 센서 데이터 요구 메시지를 받은 Server에서는 10byte의 Sensor 데이터를 Client로 전송한다.

다음은 Sensor 데이터 전송에 대한 내용을 표시 한 것이다.

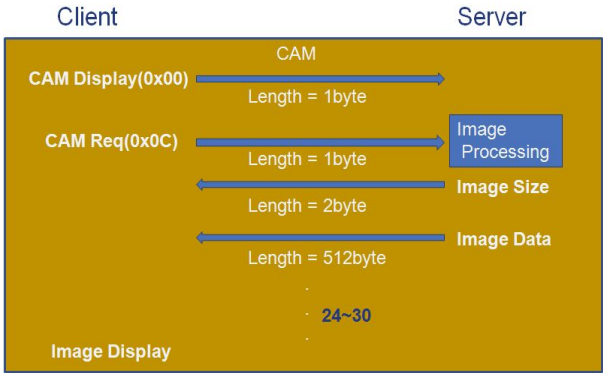


[그림22] Sensor 데이터 전송 형식

5.3.3. Camera

Camera 데이터도 Sensor 데이터와 마찬가지로 Client에서 Camera 데이터의 요구 메시지를 Server로 전송할 경우 Server에서는 이 메시지를 받고 현재 Camera 영상을 캡처하여 이미지를 변환한 후에 해당 데이터를 Client로 전송하게 된다. Client에서 Server로 전송되는 Camera데이터 요구 메시지는 1byte로 구성되어 있으며, 이 메시지를 받은 서버에서는 YUV422포맷의 영상데이터를 JPEG로 변환하여 변환된 JPEG데이터의 크기를 1차적으로 보내고 후에 JPEG로 변환된 데이터를 전송하게 된다. JPEG의 데이터를 전송할 때엔 UDP의 특성상 512byte로 JPEG의 데이터를 쪼개서 보내며 Client에서 512byte의 데이터를 받았다는 확인 메시지가 Server로 전송되면 다음의 512byte데이터를 보내게 된다. JPEG의 경우 영상에 따라 압축률이 달라지기 때문에 위에서 언급한 내용을 적게는 24~30회 정도 반복하여 JPEG데이터를 전송한다.

[그림23]은 Camera영상 데이터의 전송에 대한 내용을 표시 한 것이다.

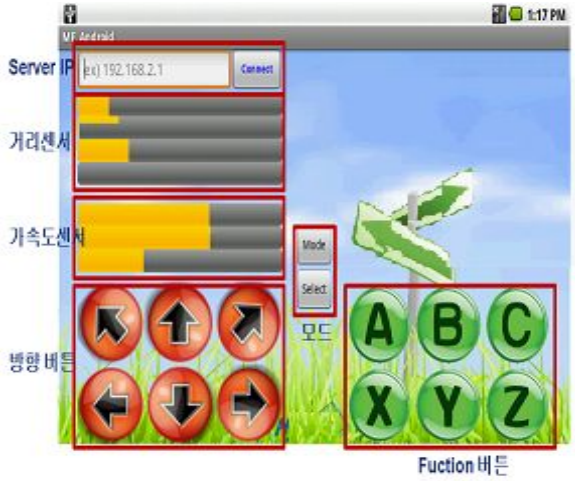


[그림23] Camera 데이터 전송 형식

5.4. 프로그램 사용

5.4.1. 프로그램화면 구성

[그림24]는 MF_Android의 프로그램 구성 모습이다.



[그림24] 실제 프로그램 실행 모습

(1) Server IP

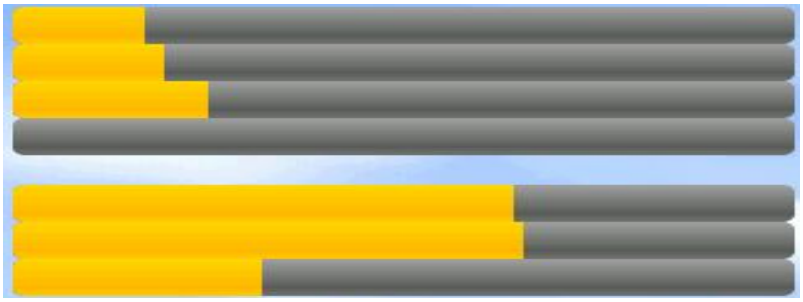
[그림25] 서버 IP 설정 화면

Server IP는 접속할 메탈파이터의 IP주소를 입력하여 해당 장치로 접속할 수 있다. 기본 값으로 192.168.2.1 이 세팅되어 있으며 IP주소를 변경 시에는 "Connect“버튼을 클릭하여야 해당 IP주소로 접속할 수 있다. Connect버튼의 텍스트는 Server와 Client가 연결되었을 때 파란색 글씨로 바뀌며 연결이 정상적으로 이루어 지지 않았을 경우 붉은색 글씨로 변경되어 연결 상태를 확인 할 수 있다.

[그림26] 네트워크 연결 상태 화면

(2) 거리센서/가속도센서

거리센서와 가속도센서는 ProgressBar로 출력이 된다. 메탈파이터에서 사용할 수 있는 최대 ADC는 4개로 ADC를 사용하는 거리센서의 출력개수는 4개, 3축 가속도 센서는 3개로 구성되어 있다. 메탈 파이터의 경우 3개의 거리센서를 사용하기 때문에 3개의 거리 센서 값이 출력되며 자동차의 경우 4개 모두 출력이 된다. 거리센서는 위에서부터 순서대로 CH1~CH4의 값을 출력하며 가속도 센서의 경우 Z, Y, X 축의 순서로 출력되어 진다.



[그림27] 거리/가속도 센서 출력 화면

(3) 방향버튼

[그림28] 방향 버튼

방향버튼은 메탈파이터의 이동버튼이다. 메탈파이터의 경우 방향버튼을 클릭했을 경우 해당 방향으로 이동 또는 턴을 하게 되어 있다.

(4) 모드

[그림29] 메탈파이터 기능 버튼

모드는 메탈파이터 기능 버튼이다. 메탈파이터의 경우 Mode스위치를 클

릭하게 되면 Mode를 변경할 수 있는 상태가 되며 F1~F6중에서 원하는 모드의 스위치를 클릭한 후 Select 버튼을 클릭하면 해당 모드로 변경된다.

(5) Camera 버튼/ 영상 출력

[그림30]카메라 영상 출력 화면

Camera버튼은 버튼의 기능과 Camera영상을 출력하는 두 가지 기능을 담당한다. Camera 버튼을 클릭할 경우 Server에서 전송된 이미지 데이터가 출력되어 진다.

(6) Function 버튼

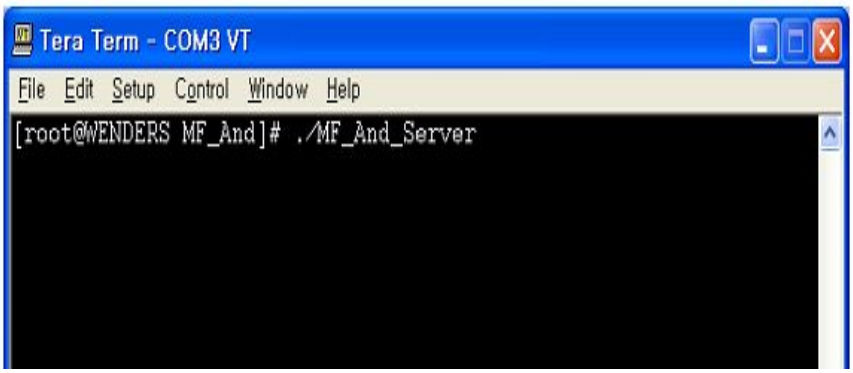
[그림31] Function 버튼

Function 버튼은 메탈파이터의 기능적인 동작을 하는 버튼이다. 메탈 파이터의 경우 RoboBASIC에서 프로그래밍 되어 있는 1~6까지의 동작을 수행한다.

5.4.2. 프로그램 구동

Android프로그램을 이용하여 메탈 파이터를 제어하기 위해서 다음과 같은 순서로 진행한다.

(1) 메탈파이터에서 "./MF_And_Server"를 입력하여 메탈파이터 Server 프로그램을 실행시킨다.

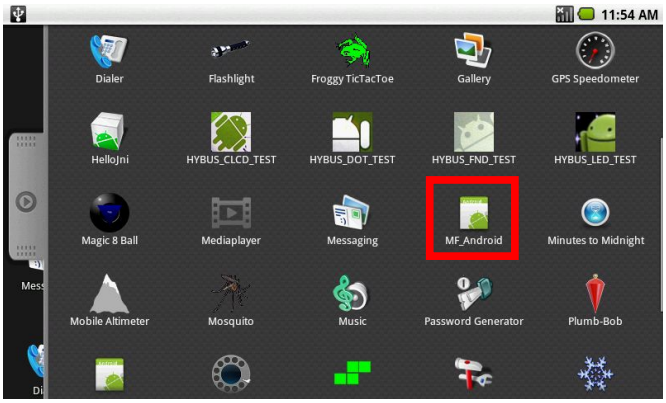


[그림32] 서버 프로그램 동작 명령어

프로그램이 실행되면 메탈파이터의 L 카메라 정지 영상이 출력되어 진다.

(2) Android 타겟에서 위에서 다운로드 받아 놓은 MF_Android 프로그램

을 실행시킨다.



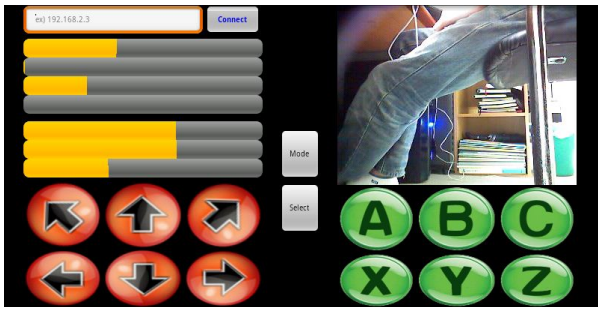
[그림33]메탈파이터 안드로이드 앱 아이콘

(3) Android 프로그램이 실행되면 Server IP 텍스트 박스를 클릭하여 접속할 Server의 IP를 입력한다. 메탈파이터의 IP를 192.168.2.1로 설정했을 경우 프로그램이 시작됨과 동시에 접속되어 프로그램을 사용할 수 있다. IP를 입력하고 Done 버튼을 클릭한다.

[그림34]아이피 설정 화면

(4) 위에서 설정한 IP로 접속하기 위해서 Connect 버튼을 클릭한다.

(5) Server와 정상적으로 연결이 완료되면 Connect글씨가 파란색으로 바뀌며 센서 값이 출력되는 모습과 방향 버튼, 모드버튼, Function버튼을 클릭하며 메탈파이터의 동작을 확인 할 수 있다. 또한 Camera 버튼을 클릭할 경우 Camera영상이 출력되는 것을 볼 수 있다.



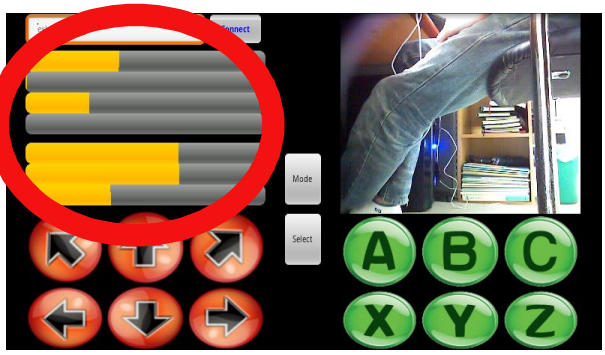
[그림35] MF 동작, 센서 출력 및 카메라 영상 화면

5.4.3. 동작 결과

(1) 안드로이드 플랫폼이 탑재된 기기에 실행 된 애플리케이션의 방향 버튼과 모드버튼 Function버튼을 클릭하면 메탈파이터의 동작을 아래와 같이 확인 할 수 있다.

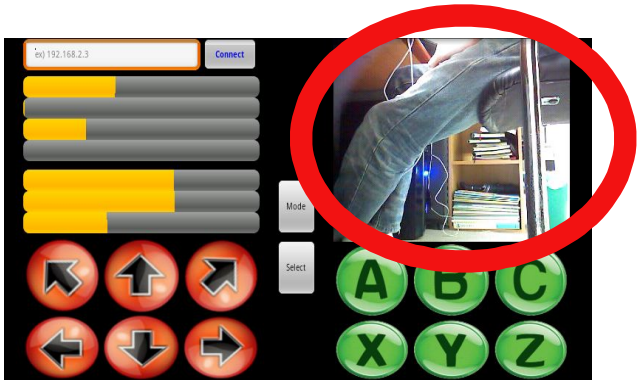
[그림36] 로봇의 동작 장면

(2) Server와 정상적으로 연결이 완료되면 Connect글씨가 파란색으로 바뀌며 센서 값이 출력되는 모습을 볼 수 있다.



[그림37]센서 값이 출력되는 화면

(3) Camera 버튼을 클릭할 경우 Camera영상이 출력되는 것을 볼 수 있다.



[그림38]카메라 출력 영상 화면

V. 결론 및 향후 연구 방향

안드로이드를 모바일 플랫폼에 탑재를 함으로서 여러 문제가 발생하기도 했다. 이러한 문제를 해결하기 위해서는 다양한 지식을 가지고 있어야 하며 이를 위해서는 최우선적으로 안드로이드 구조에 대해서 충분히 선행이 되어야 한다. 본 연구에서는 다루지 않았지만 NDK라는 과제가 남았다. 안드로이드 어플리케이션은 Dalvic 이라는 JAVA 가상머신 위에서 실행되는데, NDK(Native Development Kit)는 안드로이드 애플리케이션의 일부를 C, C++ 등의 네이티브 언어로 작성할 수 있도록 해준다. 즉, java 에서는 jni 라는 것을 이용 하여서, native method 를 사용 하여 애플리케이션을 제작 및 구현을 해보는 방법이 남았다.

참 고 문 헌

[1] 김 현, 조 영 조, 오 상 록, “URC(Ubiquitous Robotic Companion): 네트워크 기반 서비스 로봇”, 한국전자통신연구원, 2006.

[2] 문 용 선, 이 광 석, 서 동 진, 이 성 호, 배 영 철, “모듈로봇 구현을 위한 네트워크기반 모터제어 드라이버 개발”, 순천대학교 공과대학 정보통신 공학부, 2007.

[3] 장 재 식, 김 은 이, 장 상 수, 김 항 준, “비전 기반의 모바일 로봇 제어 시스템”, 한국 컴퓨터 종합 학술 대회 논문, 2005.

[4] 조 영 조, 오 상 록, “지능형 서비스 로봇과 URC(Ubiquitous Robotic Companion)”, 누리미디어, 2004.

[5] 김 현, 조 영 조, 오 상 록, “URC(Ubiquitous Robotic Companion): 네트워크 기반 서비스 로봇”, 정보과학회지 제24권 제3호, 2006.

[6] 안 호 석, 사 인 규, 백 영 민, 안 윤 석, 최 진 영, “핸드폰을 이용한 서비스 로봇 제어 시스템”, 정보 및 제어 학술 대회 논문, 2007.

[7] 김 택 수, 박 상 조, “무선 로봇을 이용한 네트워크 영상 제어 시스템의 설계”, 서원대학교 정보 통신 대학원 논문

[8] http://wizzie.tistory.com/556/ 안드로이드 개발

[9] http://www.androidside.com/ 안드로이드 개발

[10] http://developer.android.com/ 안드로이드 SDK 무료 다운로드 사이트

[11] http://www.aesop-embedded.org/ 임베디드 시스템 개발

[12] http://www.kandroid.org/ 안드로이드 개발 한국사이트

[13] http://www.androidpub.com/ 안드로이드 개발

ABSTRACT

Development of Application for Controling Robot Based on Android Platform using WiFi

No, Sung Dong
Major in Science
Graduate School
SungShin Women's University

Android Platform is ported to certain mobile Kit as H-AndroSV210. For this reason, it is downloaded Eclipse, Java JDK, Android SDK from internet for developing file system image. In addition, kernel for android platform, cross compiler, and busybox are downloaded to develop images uploading to real target. The real target based on the S5PV210 from SamSung has TFT-LCD included touch screen for applications. The Android Platform is adopted as an operating system, and application programs are implemented by using Java. It is developed an application which controls a Robot that called Metal Fighter(MF). This paper is explained to develop an android application to control MF Robot coding by Java.