
MIPS Core에 기반한 인터넷 라디오 플랫폼에 관한 연구

김종덕* · 김영길**

A Study of Internet Radio Platform based on MIPS Core

Jong-duk Kim* · Young-kil Kim**

요 약

최근 우리는 인터넷 상에서 여러 뮤직 콘텐츠를 쉽게 접할 수 있으며, 이를 음원 저작권의 허가 없이 개인의 목적에 따라 쉽게 사용하고 있다. 이것으로 인해 뮤직 콘텐츠 시장의 피해가 심각해지고 붕괴의 우려 또한 가지고 있다. 이에 따라 지적 재산권 보호법이 개정되고 지적 재산권 소유자의 소유권 행사가 시작되고 있다. 이는 그들의 지적 재산권을 보호하기 위한 방법으로서, 본 논문에서는 음원에 대한 지적 재산 소유권자와 이를 이용하려는 사용자 간의 문제를 해결하고 뮤직 콘텐츠의 새로운 시장을 형성하고자 본 연구를 진행한다.

ABSTRACT

Recently years There are numerous music contents on the internet and it can easily be used without any authority. because of this, casualties of music contents has been increased and music market has collapsed. Therefore Record Copyright Association protection law has revised and using music contents are being managed by copyright holders group from this year. It is one of ways to protect their contents copyright. So on this paper Internet Radio solution is provided to solve the problem between copyright holders and someone wants to use their contents and to make a part of new market on the music contents.

키워드

Network Streaming 네트워크 스트리밍, Internet Radio, 인터넷 라디오, Network NAT 네트워크 NAT MIPS Core, Shotcast

I. 서론

현재 우리는 인터넷을 통하여 다양한 뮤직 콘텐츠를 손쉽게 접할 수 있으며 또한 이를 개인의 임의의 목적성에 따라 무분별하게 이용하고 있다. 예를 들어 개인의 홈페이지에 배경음악으로 사용하기도 하며, P2P 공유를 통하여 인터넷 사용자간 파일 공유를 하기도 한다. 이로 인해 뮤직 콘텐츠 시장의 흐름이 흐트러지고 있고, 사업자들 즉, 뮤직 콘텐츠 판매자들 및 음원에 대한 저작권자

들의 사업 피해 심각성은 그 정도를 지나치고 있다. 따라서 2006년 12월 저작권법 변경에 따른 저작재산권의 제한과 음반제작자의 권리의 새로운 시행개정에 따라 음원의 무분별한 개인 사용에 대한 음원 소유자의 고소에 따라 법적 제지를 받게 되고, 올해를 기점으로 음원단체들이 이러한 사용에 대해 적발을 시작하며 공공장소에서의 개인이 구입한 음원이라 해도 전면 금지한다고 한다. 이는 즉 개인이 직접 구입한 CD로부터 MP3 파일을 추출해서 개인에 한하여 사용하는 것은 문제가 없으나

* 아주대학교 전자공학과 박사과정

** 아주대학교 전자공학과 정교수

파일 전송 및 공공장소에서의 사용 및 인터넷 블로그 등에서의 사용은 금지된다는 것이다. 현재도 대형 마트 및 여러 체인점에서는 유효기간을 가지는 CD를 발행하여 사용하기도 하고, 이러한 불편함을 해소하고자 PC기반의 스트리밍 서비스를 받기도 하고 있다. 그러나 이 PC기반의 스트리밍은 언제나 PC를 켜고 있어야 하는 단점이 있고 또한 콘텐츠 사업자들이 신규 사업의 일환으로 Internet Radio라는 새로운 사업 분야를 개척하고 있다. 이러한 시점에서 Internet Radio의 기본 모델을 제시하고 이를 구현하는 것을 본 연구의 당위성으로 부합시키며, 본 논문을 토대로 새로운 모델로의 확장을 가져가는데 용이하도록 하는 것이다.

II. Internet Radio System의 이해

2.1 Internet Radio System

인터넷 라디오는 임베디드 보드에서 LAN Port로부터 들어오는 데이터를 넘겨 받아 MP3 Encoding Chip을 통해 데이터를 인코딩하여 플레이 하는 것이다. 아래 그림 1은 일반적인 인터넷 라디오 시스템을 보여준다.

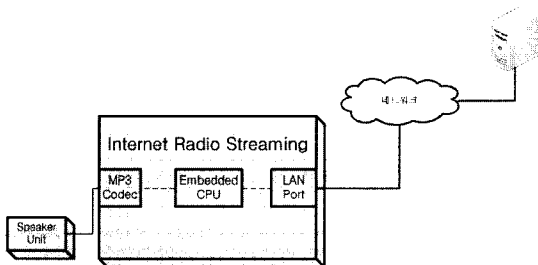


그림 1. 일반적인 인터넷 라디오 시스템
Fig 1. General Internet Radio System

2.2 HTTP Protocol

HTTP Protocol은 지속적인 연결을 하며 데이터를 주고 받는 것이 아니라 필요한 데이터가 있을 때마다 소켓을 연결하여 데이터를 받고 끊는다. 아래 그림 2와 3은 파일 전송을 기본으로 한 HTTP Protocol을 보여준다.

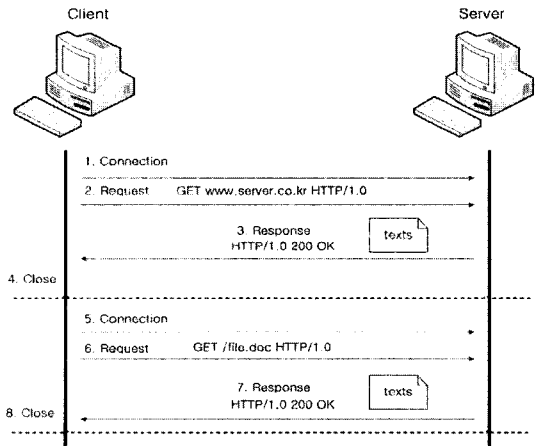


그림 2. HTTP Protocol 1.0
Fig 2. HTTP Protocol 1.0

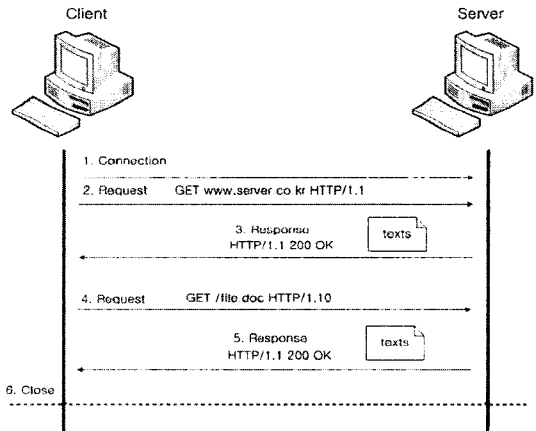


그림 3. HTTP Protocol 1.1
Fig 3. HTTP Protocol 1.1

HTTP Protocol 1.0 과 1.1의 차이점은 HTTP 1.1은 Connection 이후 하나의 문서 문서내에 다른 Objects가 존재한다면 Re-connection 없이 기존의 Connection 통해 데이터를 받을 수 있다.

2.3 Streaming Format

트리밍 프로토콜은 두 가지로 볼 수 있다. MMS Protocol과 Shoutcast Protocol이 있으나 본 논문에서는 Shoutcast를 사용하였고 그 Stream Format은 그림 4와 같다.

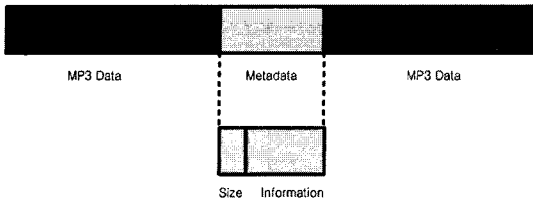


그림 4. Shoutcast Stream Format
Fig 4. Shoutcast Stream Format

위의 그림 4는 Shoutcast의 Stream Format을 보여주고 있는데, Metadata를 보면 Metadata의 첫 블록은 Size 정보를 담고 있다. 다시 말하면 Metadata는 MP3 Data의 중간 중간마다 존재하는데, 이때 첫 바이트가 0이면 Metadata의 Information 정보가 없고, 첫 바이트가 0이외의 값이라면 Information이 정보가 들어있는 것이다. Metadata의 첫 바이트가 0 이외의 값인 경우는 두 가지 경우가 존재한다. 첫째는, 네트워크 스트리밍 서버로 접속할 때 이며, 두 번째는 네트워크 스티리밍이 새로이 업데이트 될 경우이다. 즉, 곡이 바뀌는 경우를 말한다.

이 Data Format은 2.3에서 설명할 Streaming Protocol을 통해 얻을 수 있다.

2.2 Shoutcast Streaming Protocol

Shoutcast Streaming Protocol은 HTTP 1.x에 기반한 것이며, 표 1과 표2는 각각 Client에서 Server로 Request 하는 것과, Server에서 Client로 Reply 하는 것을 보여준다.

표 1. Client의 Request Protocol
Table 1. Request Protocol of Client

GET_HTTP	"GET /마운트명 HTTP/1.0WrWn"
HOST	"Host: 접속할 Server명WrWn"
USER_AGENT	"User-Agent: Client 명WrWn"
ACCEPT	"Accept: */*WrWn"
ICY_METADATA	"Icy-MetaData: 1WrWn"
CONNECTION_CLOSE	"Connection: closeWrWn"
DUMMY	"WrWn"

위 표 1에서 마운트명은 서버의 구성에 따라 그 유무가 달라진다. 서버가 자동으로 Client에게 포트를 할당해주는 경우는 마운트 포인트가 없으나, Streaming Server가 마운트 포인트를 정해놓고 서비스를 한다면 해당 부분에 마운트 포인트가 필요하다.

다음 표 2는 Client Request로부터 Server의 Reply를 보여준다.

표 2. Server의 Reply Protocol
Table 2. Reply Protocol of Server

ICY {Value}	"200 OKWrWn, 404 Not FoundWrWn"
ICY-NOTICE{N}	"Notice ContentsWrWn"
ICY-NAME	"BroadCast 명WrWn"
ICY-GENRE	"Genre명WrWn"
CONTENT-TYPE	"Content type명WrWn"
ICY-BR	"Stream BitrateWrWn"
ICY-PUB	"Publish ValueWrWn"
ICY-METAINT	"MetaData Interval ValueWrWn"
DUMMY	"WrWn"

위의 표 2는 Server의 Reply로서 ICY-METAINT Field는 Client에서 절대적으로 필요한 Field로서 이를 토대로 Client에서 Streaming Data를 Control을 하도록 한다.

2.3 Remote Controller Communication Format

Remote Controller는 2.4 Ghz 대역을 사용하며, Internet Radio의 Channel, Volume, Play, Stop에 관한 사용을 제어한다. 아래 표3은 통신 포맷을 보여준다.

표 3. Remote Controller 통신 포맷
Table 3. Remote Controller Communication Format

Remote Controller Communication Format	
StartCMD	STX
Length	2Bytes(CMD+DATA)
CMD	Play, Stop, Channel up & down or Volume up & down
DATA	Channel info or Title info
EndCMD	ETX
LRC	Normally zero

II. MIPS Core에 기반한 Internet Radio Platform

3.1 MiPSCore기반의 TwinPass-VE Platform

본 Platform은 MIPS Core에 기반한 Infineon의 TwinPass-VE를 사용하였다. 아래 그림 5는 TwinPass-VE의 Architecture를 보여준다.

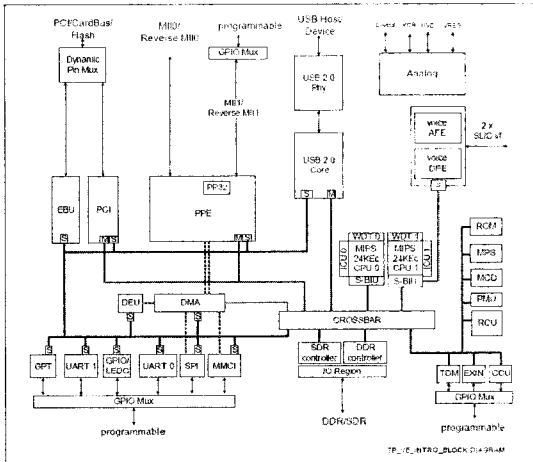


그림 5. TwinPass-VE Architecture
Fig 5. TwinPass-VE Architecture

해당 Platform은 Network Solution이므로 본 논문의 주제와 같은 Platform을 연구하는데 적합하다고 보겠다. Twinpass-VE는 Dual-Core이며 DDR RAM Interface를 제공하여 System의 Performance 향상에도 크게 도움이 된다.

3.2 Internet Radio Platform의 블록도

본 Platform은 Network Solution을 User의 Network 사용에 편의를 주기 위해 2Ports LAN을 구성하였고, Serial Interface를 사용하여 Remote Controller를 연결하여 User의 Platform 사용에 대하여 편의성을 부여하였다. 아래 그림 6은 구현한 Platform의 블록도이다.

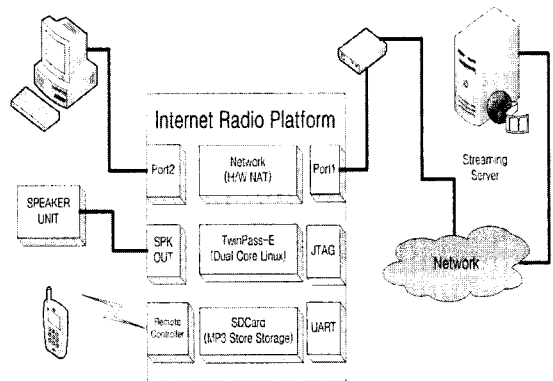


그림 6. 구현한 Platform의 블록도
Fig 6. Implemental Platform Block Diagram

3.3 Internet Radio Software Sequence

Internet Radio의 Software는 Linux의 Thread 기반으로 구성하였으며, Thread는 Streaming Thread, Network Play Thread, Remote Control Thread와 Local Play Thread로 구성하였다. Local Play Thread는 Network가 끊겼을 시에 User들의 시간적 공간적 공백을 없애기 위하여 구성하였다.

아래 그림7에서 보는 바와 같이 Software의 수행 절차는 Streaming Thread에서 초기 Channel Lists를 loading 하고 Channel Lists에 따라 Connection을 시도 하고, Connection 후에는 Network Streaming을 받아 1MBytes의 Buffer에 계속적으로 Buffering을 한다. Network Play Thread에서는 Buffering을 Check하여 MP3 Codec Chip으로 SPI Interface를 통하여 MP3 Data를 보낸다. Remote Control Thread는 Channel 변경, Volume Control, Play & Stop에 대해 제어하고, Remote Controller로 Channel 정보와 Title 정보를 보낸다. 또한, Local Play Thread는 Network의 연결과 종료료를 Check하여 Network가 끊겼을 경우 USB Memory에 있는 MP3를 재생한다.

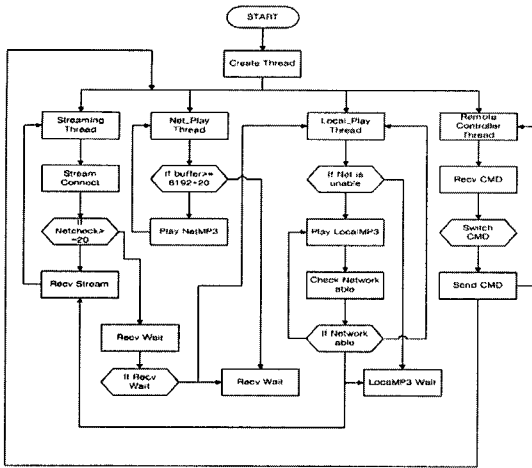


그림 7. Software 수행 절차
Fig 7. Software Execution Sequence

IV. 실험 결과 및 분석

4.1 Internet Radio Prototype

현재 개발된 Internet Radio Platform으로서 전체적인 테스트를 진행하였다. 테스트는 Network Share와 Application 테스트를 진행하였다. Application 테스트에는 RF Remote controller, SPI MP3 Codec chip 테스트를 동반하였으며 하기 그림8은 개발된 Platform을 보여준다.

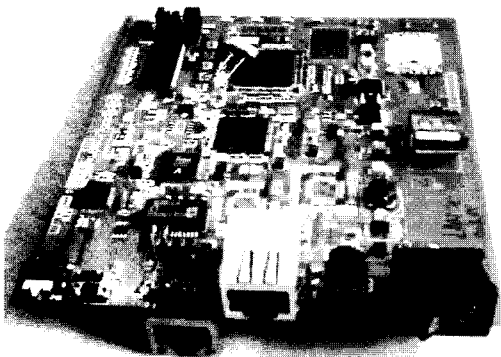


그림 8. 개발된 인터넷 라디오 플랫폼 프로토타입
Fig 8. developed Internet Radio Platform Prototype

하기 표 4는 개발된 플랫폼의 하드웨어 사양을 보여준다.

표 4. 하드웨어 세부사항
Table 4. Hardware Feature

구 성	세부사항
CPU	TwinPass-VE (32-bit MIPS 333MHz DualCore)
Memory	NOR FLASH 8Mbytes, RAM 32Mbytes, USB 1Gbytes
RAM	DDR 32Mbytes(Winbonds)
Interface	SPI, LAN, USB, Serial
Remote Controller	2.4GHz 무선모듈
AUDIO	VS1011E(VLSI)

4.2 Network Share Test

하기 그림 9와 10은 각각 Network Share에 대한 테스트를 보여준다. 그림 9는 Platform에서의 IP를 보여주고 있다. Platform의 IP의 eth1을 보면 static IP로 Wan Port로 IP를 할당받은 것이고, eth0은 br0로 바인딩하여 br0 IP가 게이트 웨이 역할을 한다.

```
* ifconfig -a
br0      Link encap:Ethernet Hwaddr 08:01:11:11:11:11
        inet addr:192.168.100.250 bcast:192.168.100.255 Mask:255.255.255.0
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
        RX packets:294521 errors:0 dropped:0 overruns:0 frame:0
        TX packets:716213 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueue:len:0
        RX bytes:31667039 (30.1 MiB) TX bytes:999607828 (953.2 MiB)

eth0     Link encap:Ethernet Hwaddr 08:01:11:11:11:11
        UP BROADCAST RUNNING PROMISC MULTICAST MTU:1500 Metric:1
        RX packets:294629 errors:0 dropped:142 overruns:0 frame:0
        TX packets:716213 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueue:len:1000
        RX bytes:35842243 (34.1 MiB) TX bytes:999732356 (953.4 MiB)

eth1     Link encap:Ethernet Hwaddr 08:01:11:11:11:12
        inet addr:202.30.11.120 bcast:202.30.11.255 Mask:255.255.255.0
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
        RX packets:123432 errors:0 dropped:168 overruns:0 frame:0
        TX packets:254281 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueue:len:1000
        RX bytes:1049793114 (1001.1 MiB) TX bytes:28834730 (27.4 MiB)

lo       Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0
        UP LOOPBACK RUNNING MTU:16436 Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueue:len:0
        RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

*
*
```

그림 9. Network Share Test(WAN)
Fig 9. Network Share Test(WAN)

하기 그림 10은 LAN Test를 진행한 것으로서 PC에서 DHCP를 통해 IP를 받아온 것을 보여준다.

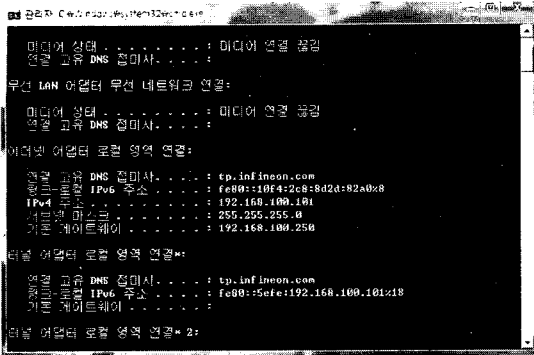


그림 10. Network Share Test(LAN)
Fig 10. Network Share Test(LAN)

4.3 Application Test

Application은 서버를 통해서 들어오는 Stream을 받아 일정량 buffering을 하고 이를 SPI를 통해 MP3 Codec으로 내보낸다. 그림 11은 Application의 전체 Test를 보여준다.

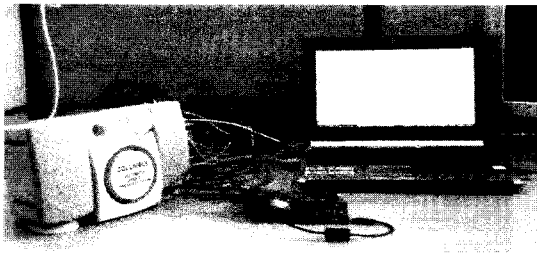


그림 11. 어플리케이션 테스트
Fig 11. Application Test

그림 12는 Remote Controller의 결과를 보여준다. Remote Controller는 해당 서비스에서의 Title과 Channel정보를 보여준다.

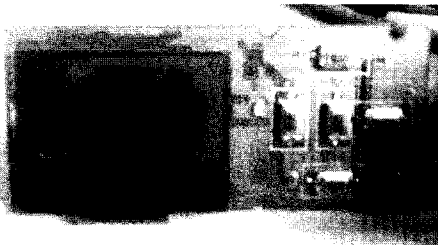


그림 12. 리모트 컨트롤러 테스트
Fig 12. Remote Controller Test

그림 13은 Software 테스트 결과로서 Server를 통해 들어온 Message를 출력하였고, 해당 Message에는 상기 2.2에서 설명한 Protocol 데로 Reply한다. 또한, Remote Controller를 통해 Volume Control한 결과를 보여준다.

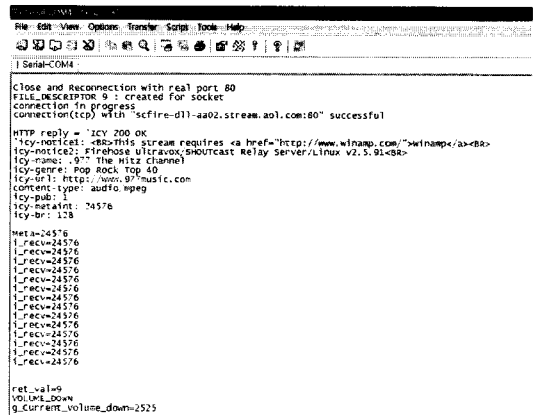


그림 13. 소프트웨어 테스트
Fig 13. Software Test

VI. 결론

본 논문에서는 2006년 12월 저작권법 변경에 따른 저작권의 제한과 음반제작자의 권리의 새로운 시행개정에 따라 음원의 무분별한 개인 사용에 대한 제재방침이 강화되고 개인이 직접 구입한 CD로부터 MP3 파일을 추출해서 개인에 한하여 사용하는 것은 문제가 없으나 파일 전송 및 공공장소에서의 사용은 금지되는 것에 이를 해결하기 위한 Solution의 일환으로 인터넷 라디오 플랫폼에 대해 연구하였고, 그에 따른 기본 플랫폼의 모델을 제시 하였다. 이에 따라 다음과 같은 결론을 얻을 수 있었다.

- (1) Streaming Thread에서 Metadata Control을 하지 않으면, Play중에 지속적인 음의 끊김 현상을 보였다.
- (2) Thread 수가 많아지고 각각의 Thread내에 Condition들이 많아지면, System의 Performance가 떨어짐은 물론 Network Play Thread와 Streaming Thread간에 공유 Buffer를 사용하는데 Mutex를 사용하지 않으면 Thread 전체가 불안한 동작을 보였다.
- (3) 또한 Server의 Network이 임시적 끊김 현상 빈도수가

찾음을 보였고, 이 상황에 Connection을 끊는다던지 혹은 끊김 전까지 받음만큼의 Data만 Control할 경우 다음 들어오는 Metadata의 위치를 알 수 없었기 때문에 Network가 임시적으로 끊겼다던지 혹은 Error를 Return할 경우에 하드웨어적인 Error나 초기 Connection에서의 Error가 아니라면 강제로라도 Connection을 유지하도록 해야 문제가 발생하지 않았다.

참고문헌

- [1] TwinPass Chip Manual-TP_VE_V1.1_UM_HD_Rev 1.0.pdf
- [2] TwinPass Software Manual- TwinPass_VE_V1.1_UM _BSP_SD_Rev1.0.pdf
- [3] VS1011 Chip Manual-VS1011.pdf
- [4] VS1011 Application Manual-VS10XX[an].pdf
- [5] HTTP Protocol Document.pdf

저자소개



김종덕 (Jong-duk Kim)

배재대학교 컴퓨터공학과 학사
아주대학교 전자공학과 석사
현 아주대학교 전자공학과 박사과정

※ 관심분야 : Embedded Platform, RFID Platform, Embedded Medical System



김영길 (Young-kil Kim)

고려대 전자공학과 학사
한국과학기술원 석사
ENST(프랑스) 박사
아주대 전자공학과 교수(현재)

※ 관심분야 : RFID Platform, Embedded system, 초음파 의료기기, Mobile 의료정보 시스템