모바일 환경을 위한 맞춤형 서비스 유비쿼터스 영상전송 시스템의 설계

이상욱* · 안용범** · 김응수***

Design and Implementation of High-Resolution Image Transmission Interface for Mobile Device

Sang-wook Lee* · Yong-beom Ahn** · Eung-soo Kim***

이 논문은 산학협동재단의 지원으로 수행되었음

요 약

영상전송에 있어서 주변 환경에 대한 영상인식은 대단히 중요하다. 최근 정보인프라의 대중화에 따라 모바일 환경에까지 유, 무선을 통한 대상의 상태를 영상으로 확인하려는 사용자 중심적 욕구가 증가하였다. 하지만 기존의 모바일용 솔루션은 저대역폭을 가지는 모바일 단말기 기반의 네트워크 특성상 소프트웨어 코덱만을 지원함으로써 소프트웨어적인 처리의 한계로 인한 고화질의 영상은 기대할 수 없었다. 이러한 단점을 해결하기 위해서 본 논문에서는 모바일 환경에서 적용할 수 있는 고화질 지원과 실시간적 요소를 고려하여 높은 압축률을 가지는 국제 비디오 압축 표준인 MPEG-4 기반 하드웨어 코덱 설계에 관하여 기술하였다. 이를 위한 시스템의 최적화된 설계를 위하여 임베디드용 전용 CPU를 채용하였으며 구현된 시스템은 모바일 환경에서도 고기능을 서비스하고 전송속도 및 해상도 등에 있어서 우수한 성능을 나타내었다.

ABSTRACT

An image recognition for surrounding conditions is very important in image transmission. In recently years, as the information infrastructure is more general, the user-centered demands in which they want to identify by object's states image using wire or wireless environment have increased. However, existing mobile solution could be hard to expect high quality image, because limitation of software processing according as network based on mobile terminal which has low band width supports software codec. To solve this weak point, this paper describes on hardware codec design based on MPEG-4 which is international video compression standard. Implemented system contains the embedded CPU for optimized design and it works high quality service as transmission speed and resolution in mobile circumstance.

키워드

고화질 영상 전송(High-resolution image transmission), MPEG-4, 실시간 모니터링(Real-time monitoring)

* 경상대학교 정보통신공학과 교수
** 대전대학교 대학원 전자공학과
*** 대전대학교 전자공학과 교수(교신저자)

접수일자 : 2008. 1. 17

Ⅰ. 서 론

정보통신분야의 급격한 발전과 더불어 인터넷 사용 환경 또한 빠른 속도로 발전하고 일반화되어가고 있으며 PC이외의 다양한 정보가전 기기들이 등장함에 따라 영상전송기술은 데스크톱뿐만이 아니라 DVR, CCTV, 네트워크 카메라 등 모바일 환경에까지 대상의 상태를 영상으로 확인할 수 있는 단말들이 제공되고 있다[1][2][3].

영상전송기술을 처리하기 위한 코덱 연구들을 살펴보면 크게 소프트웨어와 하드웨어적인 접근 방법이 진행되어오고 있다. 소프트웨어적인 방법으로는 알고리즘 개발 및 개선이 있으며 하드웨어적인 방법으로는 하드웨어 구조의 개발 및 개선이 있다[4][5].

기존의 모바일 솔루션은 CPU, 메모리 등의 자원이 상대적으로 부족했고 제한된 리소스 환경으로 인해 단말기 내에서 영상을 처리하기에는 많은 한계가 있었다. 그에 따라 소프트웨어적인 처리의 한계로 인해 최대 해상도 QCIF급(176*144, 15fps), QVGA급(320*240, 15fps)의 저화질 영상만을 서비스 했으며 또한 전송속도 및 실시간 응용에도 부족함이 있었다. 이에 따른 해결방안으로 본 논문에서는 모바일 환경에 적합한 저비트율로 고효율 고압축 부호화를 가지는 국제 비디오 압축 표준인 MPEG-4에 기반을 둔 하드웨어 코덱 설계에 관하여 기술하였다.

MPEG-4는 비디오의 복잡하고 융통성 있는 VLSI 기술을 이용한 전용 하드웨어 구현을 매우 어렵게 하며, 프로그램이 가능한 범용 DSP로 구현할 경우에도 높은 압축률을 제공하기 위해 요구되는 많은 계산량으로 인해 성능 면에서 큰 제약을 받는다. 이에 따라 본 논문에서는 많은 계산량을 요구하는 동시에 융통성 있는 구조를 갖는 MPEG-4 모바일 전용 하드웨어 코덱을 구현하고, 모바일용 영상전송단말의 안정된 동작, 정확한 영상인식 및 고속의 프로세서 기능 지원을 위해 임베디드용 전용 CPU를 채용하였다. 또한 D1급 이상의 고화질 영상 서비스를 지원하고 특정 영상관련 응용프로그램에 적합한 하드웨어 구조가 아닌 다양한 영상관련 응용프로그램을 수행할 수 있는 사용자 중심의 서비스를 위해 별도의 MPEG-4 하드웨어 코덱을 설계하였다. 또한 향후 기능의 추가 및 변경을 용이하게 하기 위해 제어시스템 부분과 MPEG-4 코덱 부분으로 나누어 모듈별로 설계하였다.

Ⅱ. 시스템 구성

2.1 구현 기능

영상입력으로는 화질이 우수한 CCD(Charge Coupled Device)방식의 카메라를 사용하며 SD 메모리의 채택으로 데이터의 저장 및 보관이 용이하며 용량에 따라 대용량화가 가능하다. 네트워킹으로는 유, 무선을 동시 지원하고 유선 10/100Mbps, 무선 IEEE 802.11b/g를 지원한다. 이를 통하여 사용자는 네트워크상에서 언제 어디서나 원격지의 영상을 모니터링 할 수 있다. 리코딩 코덱으로는 MPEG-4를 지원하고 최초 QVGA급(320*240, 15fps)부터 D1(720*480, 30fps)까지 서비스 하며 외부 이벤트에 따라 사용자 중심에 맞게 어떤 단말이 서비스를 요청하는지에 따라 해상도를 결정해 비트스트림을 출력하고 멀티채널 인코딩, 디코딩 및 멀티채널 디스플레이가 가능해 사용자는 보다 더 고기능 고품질의 서비스를 받을 수 있다.

2.2 하드웨어 시스템 구성

MPEG-4에 기반을 둔 시스템의 전체 하드웨어 블록도는 그림 1과 같다.

S3C2440과 FLASH, SDRAM, ETHERNET, SD MEMORY 부분을 제어 시스템 부분, VIDEO DECODER, AUDIO ADC, AUDIO DAC, AT2042 부분을 MPEG-4 하드웨어 코덱 부분이라 하고 차후에 하드웨어 및 소프트웨어의 추가, 수정이 용이하도록 모듈별로 설계하였다.

최초 CCD 카메라로 입력되는 영상신호를 VIDEO DECODER 부분을 통해 디지털 영상으로 컨버젼하고, MICROPHONE 으로 들어오는 음성신호는 AUDIO ADC 부분을 통해 디지털 음성으로 컨버젼 한다. 이후 MPEG-4 코덱 부분에서 서비스를 요청하는 단말의 종류에 따라 해상도를 결정하고 인코딩 과정을 거쳐 비트스트림이 출력되고 전송된다. S3C2440 제어 시스템 부분을 통해 단말들은 접속될 수 있고 유, 무선을 통해 직접 혹은 원격지의 영상을 모니터링 할 수 있다.

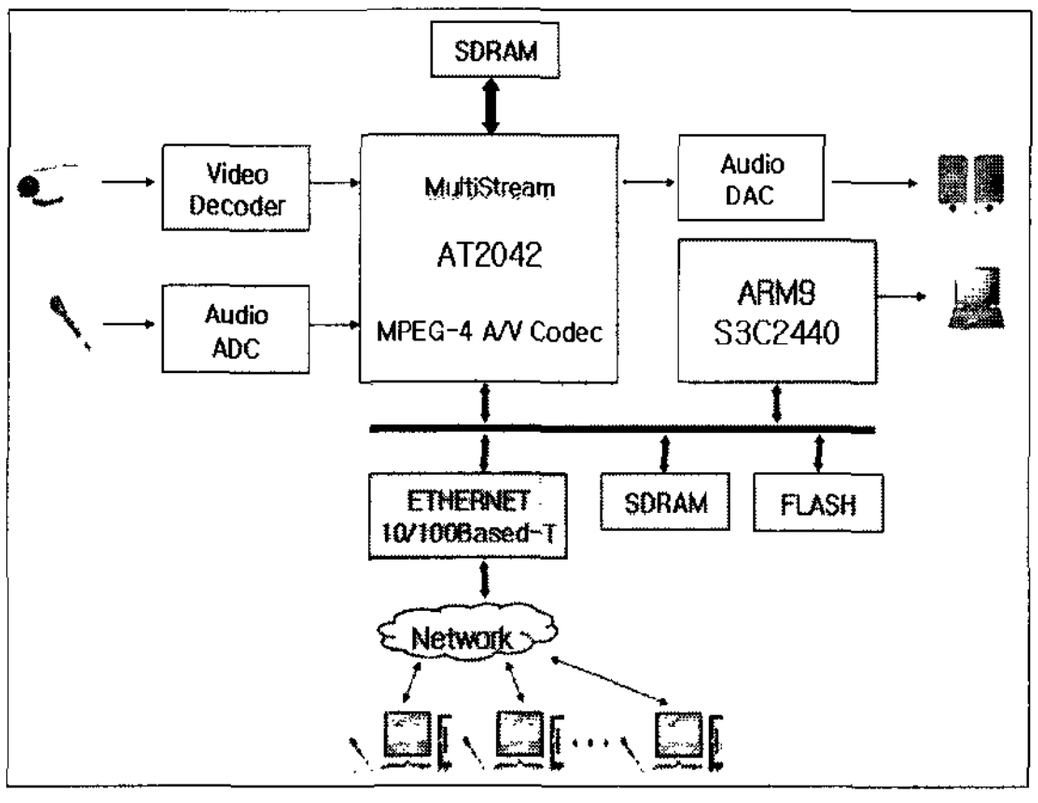

그림 1. 하드웨어 시스템 구성
Fig. 1 Hardware System Configuration

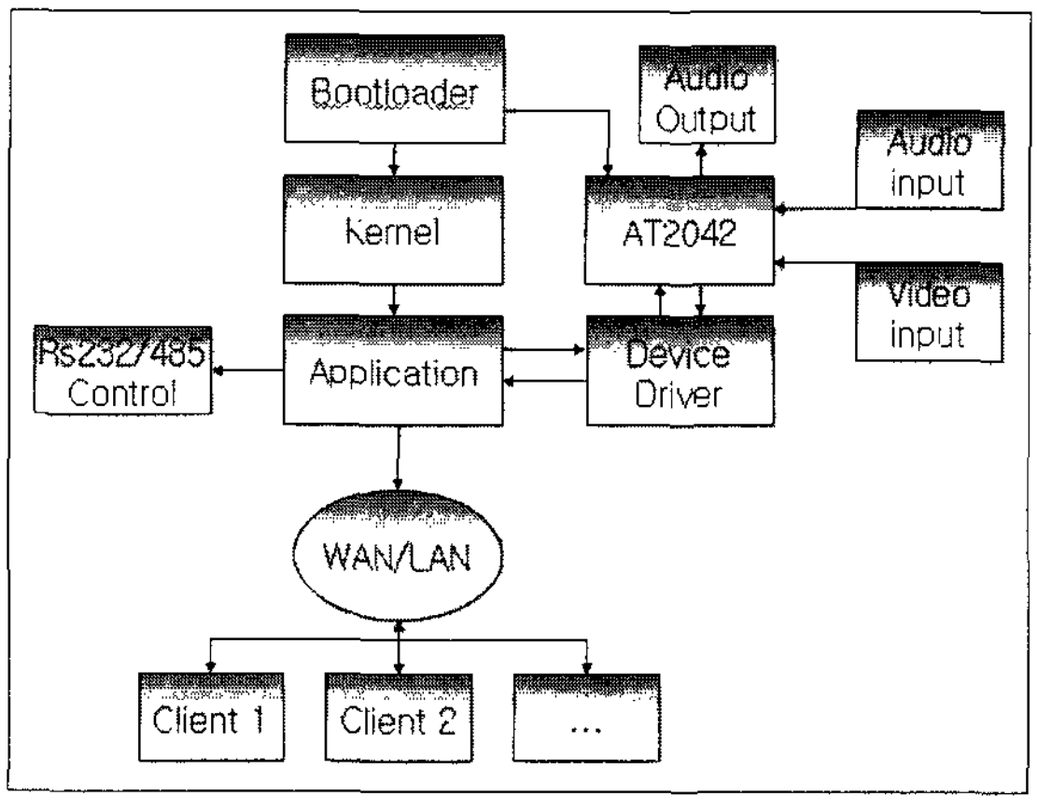

그림 2. 소프트웨어 블록도
Fig. 2 Software Block Diagram

2.3 소프트웨어 시스템 구성

그림 2.는 MPEG-4에 기반한 하드웨어 코덱의 소프트웨어 블록도를 나타낸다. OS로는 임베디드 리눅스를 사용하였고 커널 버전 2.4.20으로 구현하였다.

처음 전원을 인가하거나 하드웨어적으로 리셋을 시키면 부트로더는 시스템을 시작하기 위해 필요한 최소한의 하드웨어를 초기화하게 된다. 이후 메모리에 담겨있는 커널이미지와 파일시스템을 퓨징시켜 부팅이 된다. S3C2440 레퍼런스 보드에서 제공되는 부트로더를 사용하였으며 board 디렉토리에 SDRAM, NOR FLASH, NAND FLASH 사용을 위한 메모리 맵핑을 수정하였으며 commom 디렉토리에는 Video Encoder(SAA7121)칩의 I2C 통신 설정과 최초 부팅시에 하드웨어 초기화 내용 및 USB 사용을 위한 usb_init() 함수가 추가되고 arm920t 디렉토리에 S3C2440의 CPU 클럭과 스피드를 설정해주었다. Kernel은 시스템내의 자원을 User Application이 사용할 수 있도록 관리하는 프로그램으로서 프로세스, 파일 시스템, 메모리, 네트워크 등의 관리를 맡고 User Application이 임의의 기능을 Kernel에 요구하게 되면 System Library 또는 Interrupt를 이용해서 Trap을 걸어 Kernel에 접근하게 되며 이러한 모든 접근은 System Call Interface를 통해 이루어진다.

부팅 후 제어 시스템 부분에서 시스템을 관리하며 AT2042로 들어온 데이터는 인코딩 후 각종 어플리케이션 및 디바이스 드라이버를 경유하며 유, 무선 네트워크를 통해 보내지게 된다.

그림 3.은 디바이스 드라이버의 블록도를 나타낸다. 임베디드 리눅스 시스템은 어플리케이션 적용을 위한 디바이스 드라이버 층을 가진다. AT2042 디바이스 드라이버는 어플리케이션 프로그램의 요청을 위해 AT2042 FIFO 와 레지스터에 접근한다. Status register, Command register 등 2개의 register들과 Tx FIFO, Rx FIFO, Multiplex FIFO 그리고 De-multiplex FIFO 등 4개의 FIFO로 구성되어 있다. 디바이스 드라이버는 커널에 포함될 수 있지만 시스템 변경이나 수정 시에 커널을 재 컴파일 해야 하기 때문에 모듈형태로서 업로드 시킨다.

Status register는 Host Interface Unit 내부의 FIFO 상태를 확인하기위한 16bit read only register 구조로 되어있고 Command register는 내부 RISC에서 사용되는 신호를 제어하기 위해 사용하며 16bit write only register 구조로 되어있다. Rx FIFO는 Data ID와 데이터를 저장 및 전달하는 역활을 담당하고 Tx FIFO는 System status 등의 데이터를 사용자가 모니터링하기 위한 용도로 사용된다. Multiplex FIFO는 인코딩 비트스트림을 외부 Host Controller에서 read 하기 위해 사용되고 De-multiplex FIFO는 디코딩 할 비트스트림을 write하기 위해 사용된다.

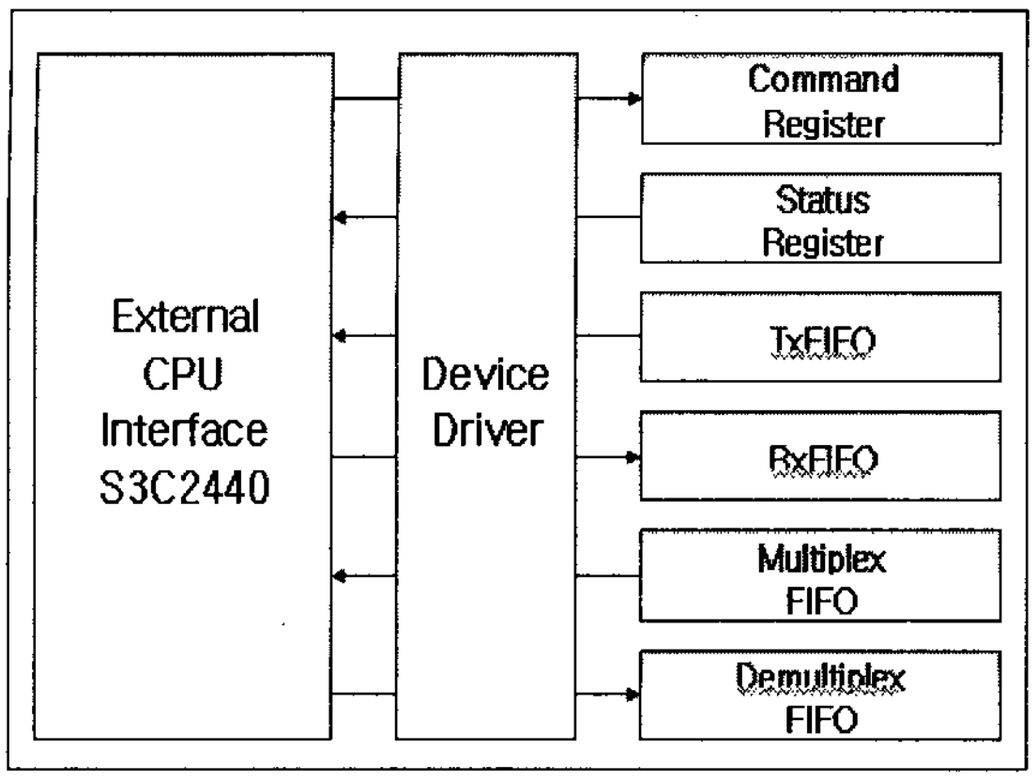

그림 3. 디바이스 드라이버 블록도
Fig. 3 Device Driver Block Diagram

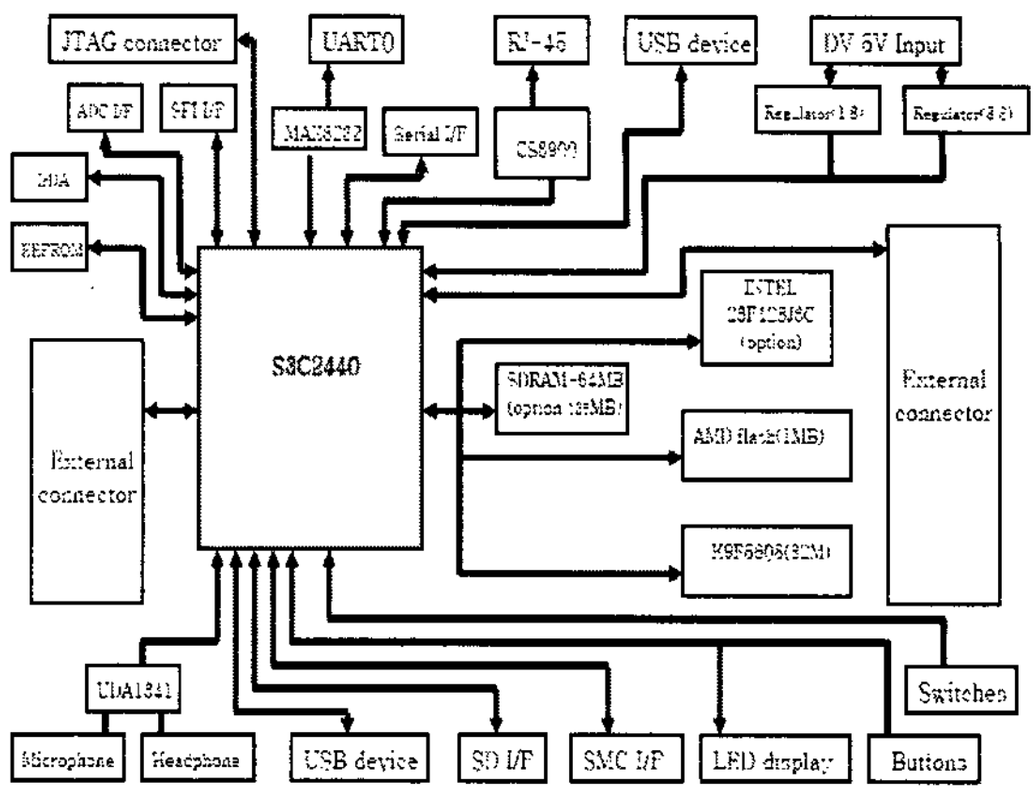

그림 4. 제어 시스템 구성
Fig. 4 System Configuration of Control

Ⅲ. MPEG-4 H/W CODEC 모듈의 설계

3.1 제어 시스템의 설계

기존까지의 모바일 환경에서는 가격대 성능비가 좋은 ARM920T를 내장한 S3C2410(ARM920T, 200MHz)을 많이 사용하는 추세였지만 낮은 동작스피드와 향후 단종의 우려가 있기에 제어시스템 부분의 전용 CPU는 32bit RISC 마이크로프로세서로서 저전력 소모와 높은 성능을 지원하는 S3C2440 (ARM920T, 400MHz)을 선택하였다.

그림 4.는 제어시스템 부분의 구성을 나타낸다. 레퍼런스 보드를 기반으로 하여 기능 테스트 및 불필요한 블록을 삭제해 제어시스템 부분을 설계 하였다.원격지에서의 모니터링 서비스를 제공하므로 자체 LCD부분을 삭제하고 카메라는 AT2042에 연결되기 때문에 카메라 인터페이스 부분을 삭제하였다. 이더넷은 기존 10M에서 빠른 전송속도를 위해 10/100M Based-T로 변경하고 무선랜은 IEEE 802.11 b/g를 지원하는 USB형의 무선랜 모듈을 사용한다. 플래시 메모리는 NOR FLASH(1M)로서 부트 코드가 들어가고, NAND FLASH(32M)는 데이터 저장용으로 사용되며 SDRAM(32M*2)를 사용하며 펌웨어, 비디오 및 오디오 데이터, 매크로 블록정보 데이터, 스트림 데이터 등의 데이터 저장용으로 사용된다.

3.2 MPEG-4 H/W CODEC 모듈의 설계

MPEG-4 비디오 표준은 비디오, 오디오 데이터의 이식성과 범용접근성, 고비율 압축, 비디오 데이터의 상호작용을 고려하여 멀티미디어 응용에서 비디오 코딩을 위한 표준으로 설계되었다. 이것은 비디오 회의나 비디오 폰 같은 많은 실시간 비디오 시스템에 적용되고 기본적인 목표는 고효율 부호화의 추구이므로 본 논문에서는 MPEG-4를 기반으로 하드웨어 코덱을 설계하였다.

그림 5.는 AT2042 시스템 블록도를 나타낸다. 영상 입력을 위한 Video Input Interface 부분과 압축을 위한 Motion Estimation, DCT, QIQ, IDCT, VLC, VLD 부분 및 Audio, Host, Video Output Interface으로 나눌 수 있다. MPEG-4 하드웨어 코덱 칩으로는 AT2042를 사용한다. 기존 모바일 시스템에서 사용되고 있는 칩들은 영상 압축 기능만 가능하므로 영상복원을 위해서는 별도의 DSP를 사용해야 하는 방식을 채택할 수밖에 없었으나 AT2042의 사용으로 임베디드 시스템에 절실히 요구되던 다채널 복원기능이 가능하게 되었다. 그리고 칩에 내장된 스케일러를 이용하여 비디오 입, 출력 시 다양한 해상도를 지원할 수 있게 하여 시스템에 별도의 추가 부품이나 로직 없이도 하드웨어 코덱 설계가 가능하였다.

플랫폼 선정이후 개발의 용이성을 위해 레퍼런스 보드를 사용하였다. 제어 시스템(S3C2440)모듈 기반에 MPEG-4 H/W CODEC(AT2042)이 올라가는 형태이므로 제어 시스템과의 Interrupt, Control, Status, Address, Data 신호를 중심적으로 하드웨어 변경 및 수정을 하였다. NTSC 신호의 영상입력을 위해 BNC커넥터가 추가되고,

입력되는 아날로그 영상을 AT2042에서 처리하기위해 Video Decoder부분이 추가되었다. 모바일용을 고려하여 저전력, 최소 사이즈를 위해 TVP5150A 칩을 선택하였다. 마이크로폰을 통해 들어오는 아날로그 음성을 처리하기 위해서 Audio ADC회로를 추가하면서 CS5340 칩을 사용하였고 101dB, 24bit로 컨버젼하고 200KHz의 오디오 샘플 레이트를 지원하도록 설계되었다. 출력되는 음성을 스피커를 통해 플레이하기 위해 Audio DAC회로가 추가되었으며 24bit, 96KHz 스테레오 Audio DAC인 CS4340 칩을 사용하였다. 그 외에 AT2042에 리셋을 위한 리셋회로, 전원회로, 클럭회로가 추가되었으며 완성된 MPEG-4 H/W CODEC 모듈은 외부 커넥터를 통해 제어 시스템과 연결된다. NTSC 방식의 경우 최대 720*480, 30fps의 해상도 지원이 가능하며 처리 가능한 프레임 수는 각각 D1 비디오 30fps, CIF 비디오 120fps, QCIF 비디오 480fps이다. AT2042는 트랜스코딩 방식 채택으로 디코딩 한 후 다시 인코딩 하는 re-encoding 방식을 사용하기 때문에 부호화 표준간의 변환, 해상도의 변환, 프레임율의 변환, 부호화 비트율의 변환을 모두 바꿀 수 있다. 예를 들어 720*480, 30fps MPEG-2 비트스트림을 352*240, 15fps의 MPEG-4 비트스트림으로 트랜스코딩이 가능하다. 그렇기 때문에 네트워크 트래픽 및 모니터링 장치에 따라 자동으로 해상도를 제어할 수 있다.

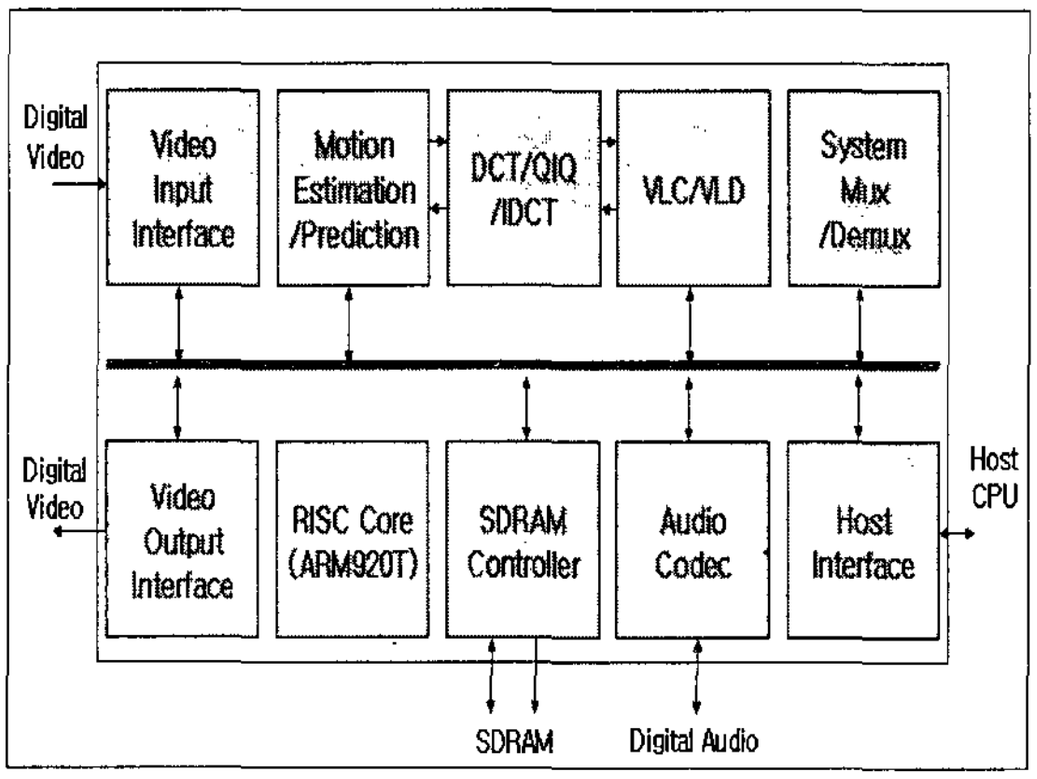

그림 5. AT2042 시스템 블록도
Fig. 5 System Block Diagram of AT2042

그림 6.은 AT2042의 Start-Up sequence를 나타낸다. Host는 AT2042 host interface unit의 command register에 "SDRAM scheduler start"명령을 write하고 AT2042의 DemuxFIFO를 통해 F/W data를 upload한다. F/W upload가 끝나면 Host는 AT2042 host interface unit의 command register에 "SDRAM access enable"명령을 write한다. 이후 AT2042 F/W가 동작을 시작한다. AT2042는 Host와 인터페이스할 준비가 되면 "chip ready" 메시지를 Host에 보내고 Host는 "chip ready message"를 받으면 "Ack for Tx" 파라메터를 AT2042에 보낸다.

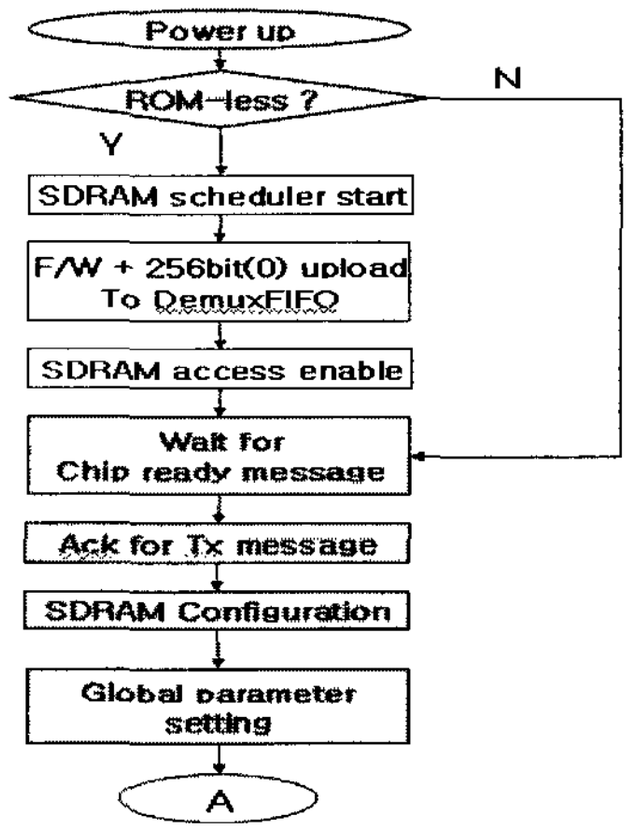

그림 6. Start-Up 순서
Fig. 6 Start-Up sequence

3.3 멀티 채널 디스플레이

그림 7.은 Multi-Channel Display를 보여준다. 최대 16채널까지 동시에 디스플레이 할 수 있으며 최대 16개의 video output window를 가지고 각각의 디코딩 된 채널은 서로 다른 video output window로 출력된다. 그림 7.에서와 같이 Display 함수를 0 또는 1로 설정함으로서 5개의 채널을 디코딩하여, 그 중 3개의 채널을 동시에 디스플레이 하고, 다른 2개의 채널은 디스플레이 하지 않을 수 있다. 이 경우 CH1, CH3, CH5의 display parameter [GID=0x9, PID=0x4]는 1로 하고, 나머지 CH0, CH6의 display parameter[GID=0x9, PID=0x4]는 0으로 한다. 그리고 CH1은 scale-down을 하지 않고, CH3과 CH5는 2:1 scale-down할 수 있다. [GID=0x9, PID=0x10]. Scale-down한 채널은 display offset parameter[GID=0x9, PID=0x11]를 이용해 디스플레이할 위치를 설정할 수 있다. 각각의 채널은 디스플레이 레벨 함수에 의해 bottom < middle < top의 서로 다른 우선순위를 가지며, 그 값에 따라 위에 위치하거나 아래에 위치할 수 있다. 우선순위는 display

priority parameter[GID = 0x9, PID = 0x5]를 이용해 변경할 수 있다. 그림 7.의 경우 CH3의 우선순위가 가장 높고, CH1의 우선순위가 가장 낮은 것을 볼 수 있다.

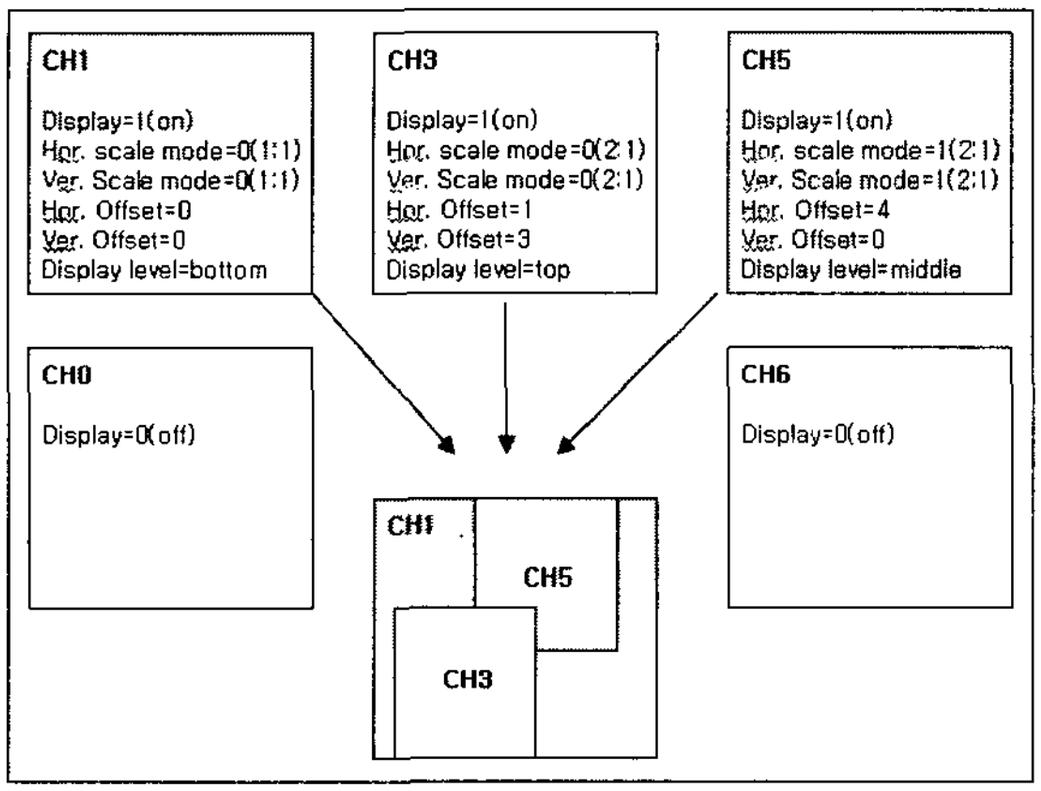

그림 7. 멀티 채널 디스플레이
Fig. 7 Multi-Channel Display

Ⅳ. 실험 및 결과

그림 8.은 Encoding Sequence를 보여준다. Host는 AT2042를 이용해 Encoding하는 경우 그림 8.과 같은 순서로 AT2042와 인터페이스 한다. Encoding을 시작하기 위해 "Encoding start" 명령을 AT2042에 보내고 AT2042로부터 "data ready" 메시지를 기다린다. 이후 "data ready" 메시지에 포함된 크기("output data size *256 bits) 만큼 AT2042의 MuxFIFO로부터 encoding된 데이터를 읽는다. "Ack for Tx" parameter를 AT2042에 보내고 계속 encoding하고자 하는 경우 "Acknowledge for data ready" 파라메터를 AT2042에 보낸다. 그렇지 않으면 더 이상 "data ready" 메시지가 발생하지 않는다. Encoding을 멈추고자 하는 경우 "encoder stop" 파라메터를 AT2042에 보낸다.

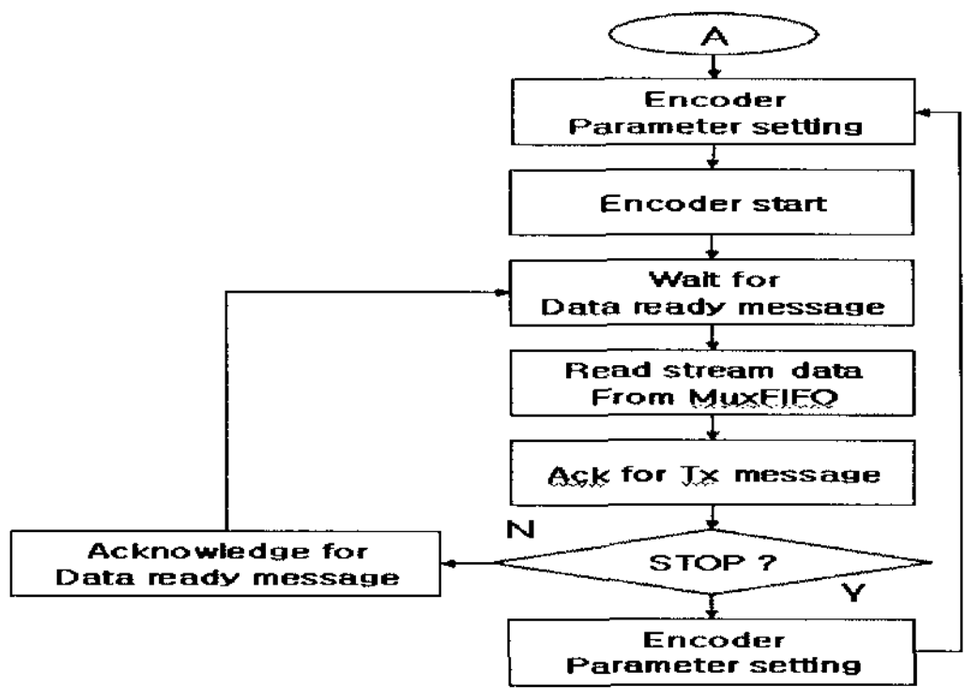

그림 8. 인코딩 시퀀스
Fig. 9 Encoding Sequence

그림 9.는 Decoding Sequence를 보여준다. Host는 AT2042를 이용해 디코딩하는 경우 그림 9.와 같은 순서로 AT2042와 인터페이스 한다. 디코딩을 시작하기 위해 "ecoder start"명령을 AT2042에 보내고 AT2042로부터 "data request" 메시지를 기다린다. 디코딩할 데이터를 AT2042의 Demux FIFO에 write하고 256bit 이상의 "0" 데이터를 DemuxFIFO에 write한다. "ack for Tx" 파라메터를 AT2042에 보내고 계속 디코딩하고자 하는 경우 "acknowledge for data request" 파라메터를 AT2042에 보낸다. 그렇지 않으면 더 이상 "data request message"가 발생하지 않는다. 디코딩을 멈추고자 하는 경우 "decoder stop" 파라메터를 AT2042에 보낸다.

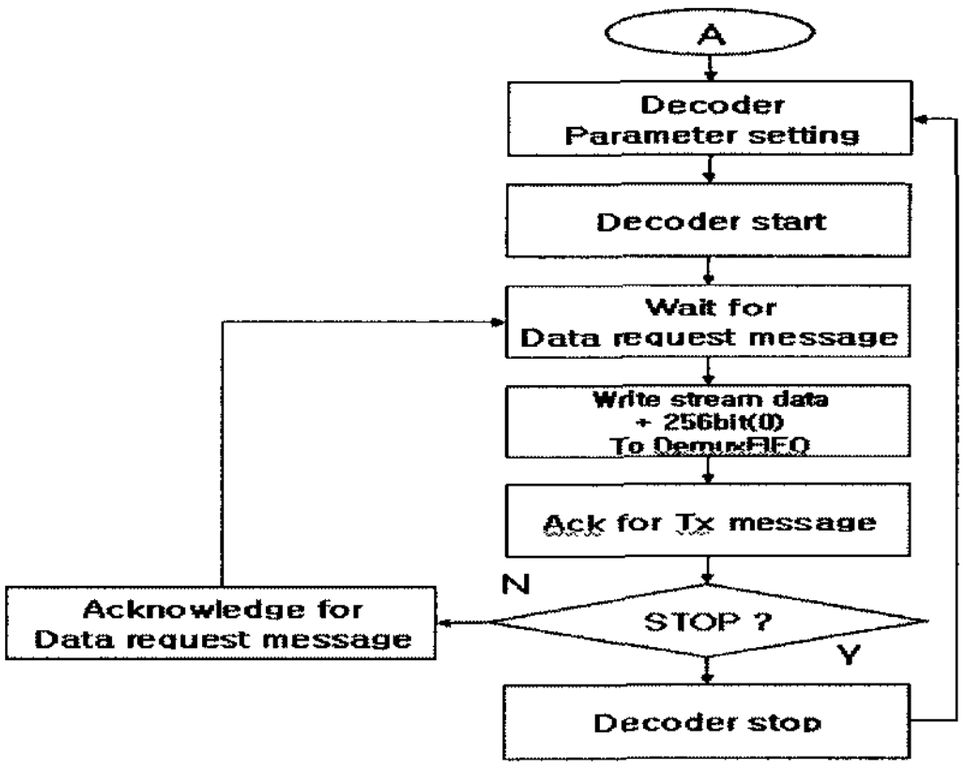

그림 9. 디코딩 시퀀스
Fig. 9 Decoding Sequence

그림 10.은 모듈별 설계에 따라 완성된 MPEG-4 하드웨어 코덱 시스템이다. 향후 기능의 추가, 변경, 수정을 용이하게 하기 위해 모듈별로 설계를 하였다. 아래쪽의 보드는 S3C2440 기반의 제어 시스템 부분이고 위쪽의 보드는 AT2042 기반의 MPEG-4 하드웨어 코덱 부분이다. 두 보드는 외부 커넥터를 통해 상호 연결된다.

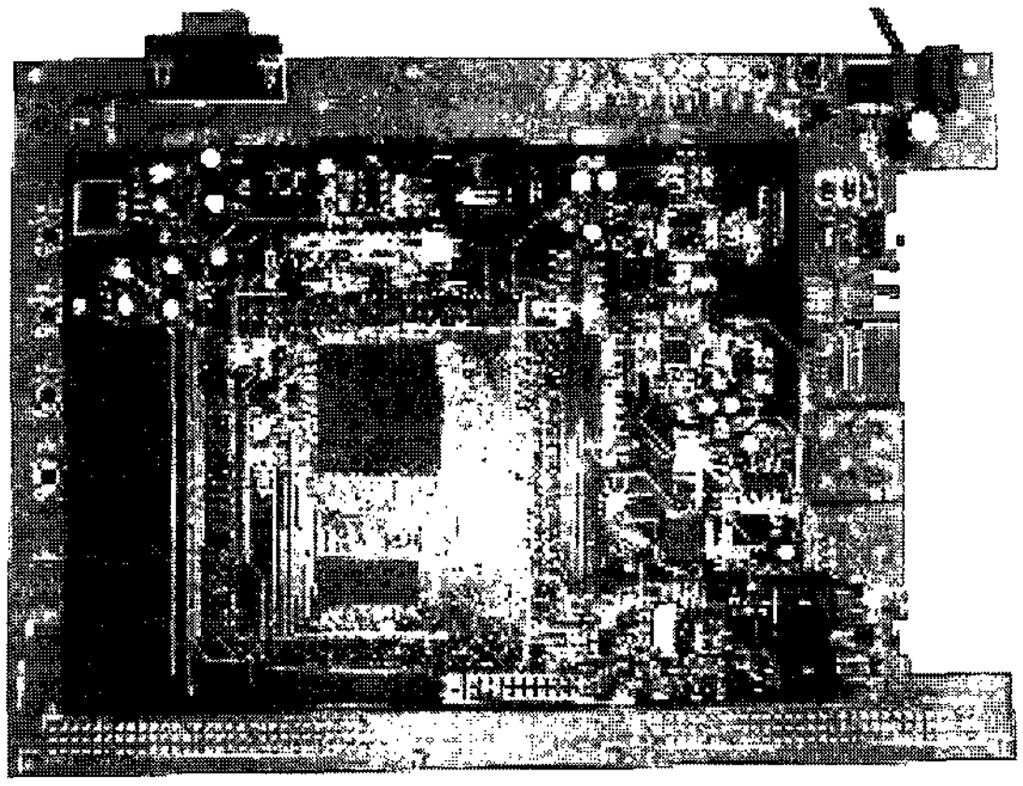

그림 10. 완성된 MPEG-4 H/W CODEC 시스템
Fig. 10 Completed MPEG-4 H/W CODEC System

표 1.은 비디오 해상도와 채널수에 따른 초당 평균 프레임 수를 보여준다. 입력 비디오 채널수가 증가함에 따라 평균 프레임수가 감소는 하지만 모바일용임에도 불구하고 최대 16채널까지 지원가능하다.

표 1. 최대 프레임 비율
Table. 1 Maximum Frame Rate

해상도	최대 프레임	채널 개수	평균 프레임
720x480 640x480	30	1	30
		2	15
		4	7.5
		16	1.875
720x240 640x240 352x480 352x480	60	2	30
		4	15
		8	7.5
		16	3.75
352x240 320x240	120	4	30
		8	15
		16	7.5
176x120 160x120	480	16	30

표 2.는 해상도 비교를 위해 지원되는 해상도에 따른 원본영상과 2배, 5배 확대 영상의 픽셀 수를 나타낸다. 기존 모바일 솔루션에서 최대로 지원하는 QVGA급(320*240) 해상도의 픽셀 수와 MPEG-4 하드웨어 코덱에서 지원하는 D1급(720*480)의 픽셀 수를 비교해 볼 때 확연한 차이를 볼 수 있으며 원본영상의 확대시 QVGA급(320*240)은 원본에 비해 잡음과 왜곡이 심하게 나타난다. QVGA급(320*240)에 비해 D1급(720*480)의 블록 픽셀수를 비교해 볼 때 약 4.5배 현저한 차이를 보이며 해상도의 향상을 볼 수 있다.

표 2. 블록 픽셀의 비교
Table. 2 Comparison of block pixel

	원본	2배	5배
320* 240	76,800개	153,600개	384,000개
640* 240	153,600개	307,200개	768,000개
640* 480	307,200개	614,400개	1536,000개
720* 480	345,600개	691,200개	1,728,000개

그림 11, 12는 같은 시간, 장소에서 촬영한 기존 최대 QVGA급 지원 모바일 솔루션과 MPEG-4 하드웨어 코덱 시스템의 영상을 비교한 사진이다. 그림 11.은 기존 솔루션의 QVGA급의 촬영 영상을 D1급 사이즈로 확대하여 캡처한 그림이고 그림 12.는 MPEG-4 하드웨어 코덱의 D1급 촬영 영상으로서 화질상의 뚜렷한 차이를 구분할 수 있다. 그림 11.에서와 같이 확대한 영상은 해상도의 향상은 존재하지 않는다. 더욱이 원본 영상에서 존재하지 않았던 잡음이 나타난다. MPEG-4 하드웨어 코덱 설계로 인해 모바일용임에도 불구하고 PC나 DVR급 장치에 못지않은 D1급 이상의 고화질 영상을 얻을 수 있었다. 또한 어떠한 단말이 모니터링 서비스를 요청하는지에 따라 그에 맞는 해상도로 스트림을 출력한다.

그림 11. 기존 모바일 카메라 솔루션 시스템 영상
Fig. 11 Image of existing mobile camera solution system

그림 12. MPEG-4 H/W CODEC 영상
Fig. 12 Image of MPEG-4 Hardware Codec

V. 결 론

본 논문에서는 MPEG-4에 기반한 비디오, 오디오 데이터를 모바일 단말상에서도 영상을 고화질로 처리하고 서비스 요청 단말에 따라 자동으로 그에 맞는 해상도로 비트스트림이 출력되고 멀티채널 인코딩, 디코딩 및 멀티채널 디스플레이가 가능한 사용자 중심적인 서비스를 제공할 수 있는 시스템을 제안하였으며 제안한 시스템의 설계 및 구현 결과를 보였다.

기존의 모바일용 솔루션은 제한된 리소스와 사용자 인터페이스가 작은 등의 제한점을 가지며 최대 QVGA급(320*240, 15fps)의 저화질 영상만을 서비스 하였고 지원 기능 또한 제한적이었다. 이에 본 논문에서는 소프트웨어 코덱이 아닌 별도의 MPEG-4 하드웨어 코덱 설계로 모바일용의 임베디드 시스템에서도 DVR에 못지않은 고기능을 지원하고 실시간으로 D1급(720*480, 30fps)의 고화질 영상을 얻을 수 있는 시스템을 구현하였다. 네트워크는 유, 무선 동시 지원으로 원격지에서도 실시간 모니터링이 가능하고 고속 및 저전력의 CPU 채용으로 안정성 및 속도 등에 있어 우수한 성능을 나타내었다.

참고문헌

[1] 김준기, 이호석 "실시간 고압축 MPEG-4 부호화를 위한 비디오 객체 분할과 프레임 전처리" 한국통신학회논문지, Vol.28 No.2C, 2003

[2] 강미연, 김도완, 김윤수, 정원호 "모바일 단말을 위한 다채널 미디어 데이터 처리기" 한국컴퓨터종합학술대회, Vol.32 No.1(A), 2005

[3] 김난영, 김상욱 "객체 기반 MPEG-4 재생 기술의 PDA 적응 기법" 제33권 제2호, 2006

[4] 김혁중, 최성혁, 김중배 "범용 영상보드를 위한 디바이스 드라이버 설계" 한국정보과학회 가을 학술발표논문집(III), p15-17, 1999

[5] 신선영, 박현상 "모바일 향 저전력 동영상 압축을 위한 고집적 MPEG4@SP 동영상 압축기" 한국방송공학회논문지. 제10권 제3호, 2005

[6] 조정민,이태훈 "실시간 디지털 원격감시시스템의 설계 및 구현에 관한 연구" 한국통신학회 학술대회 논문집,p1190-1194,1997

[7] 김지훈, 홍승욱 "리눅스 기반 DVR시스템 통신모듈 설계 및 성능 측정" 한국정보과학회 학술발표논문집, p28-30,2003

[8] SeHyun Park, EungSoo Kim, SeHoon Park "Remote Water Quality Warning System Using Water Fleas"J. of KIMICS, Vol. 4 No. 2, pp, 92-96, June, 2006

[9] 박세현, 박세훈, 김응수 "원격제어 모니터링 시스템을 위한 임베디드 리눅스 시스템의 다중 채널 직렬장치 구현" 한국해양정보통신학회 논문지, Vol. 9 No. 5, Aug, 2005

[10] Hatabu, A.; Miyazaki, T.; Kuroda, I. "QVGA/CIF resolution MPEG-4 video codec based on a low-power and general-purpose DSP" Signal Processing Systems,

2002. (SIPS '02). IEEE Workshop on, 2002
[11] Kuroda, K.; Nishitani, T."Design approaches for MPEG engines for broadband and mobile applications" Signal Processing Systems, 2002. (SIPS '02). IEEE Workshop on, 2002
[12] Park, J.H.; Kim, I.K.; Kim, S.M.; Park, S.M.; Koo, B.T.; Shin, K.S.; Seo, K.B.; Cha, J.J. "MPEG-4 video codec on an ARM core and AMBA" MPEG-4. 2001 Proceedings of Workshop and Exhibition on, 2002
[13] Austerry. "The Technology of Video & Audio Streaming" Focal Press, 2002

저자소개

이 상 욱(Sang-Wook Lee)

1977년 부산대학교 전자공학과
1988년 부경대학교 전자공학과 공학석사
2000년 부경대학교 대학원 전자공학과 공학박사
1995년~현재 경상대학교 정보통신공학과 교수
해양산업연구소 연구원
※관심분야 : 컴퓨터비전, 신호처리

안 용 범(Yong-beom Ahn)

2006년 대전대학교 전자공학과학사
2007~ 대전대학교 석사과정
※관심분야: : EMS Design, EMS System

김 응 수(Eung-Soo Kim)

1977년 부산대학교 전자공학과
1979년 부산대학교 전자공학과 공학석사
1993년 Tohoku University 전자공학과 공학박사
1982년~1993년 한국전자통신연구원 책임연구원
1993년~2000년 선문대학교 전자공학과 교수
2000년~현재 대전대학교 IT전자공학과 교수
※관심분야 : HCI, Embedded System