

RTOS 기반 무선랜 장치가 연결된 영상기록저장장치의 Progressive Download 방식 영상전송 기술 개발

Development of Progressive Download Video Transmission EDR based RTOS on Wireless LAN

남 의 석*
(Eui-Seok Nahm)

Abstract - Event Data Recorder(Car Black-Box) with WiFi dongle have been released, and the platform of the majority is the Linux platform. This is because the platform development is possible in little investment cost by reducing the source licensing costs by taking advantage of the open source. But utilizing Linux platform has the limitations of boot-up time and consuming processing power due to the limitation of battery capacity, to be cost-competitive to minimize the use of memory. In this paper, the real-time operating system(RTOS) is utilized to optimize these portions. MP4 encoder and Muxer are developed to be about ten seconds boot up and minimized memory. It has the advantages of operating at lower power consumption than the Linux utilizing WiFi dongle. Utilizing a WiFi dongle is to provide a progressive download feature on smart phones to lower product prices. But RTOS has the weakness in WiFi. Porting TCP /IP, Web and DHCP server and combination with the USB OTG Host interface by implementing the protocol stack are developed for WiFi. And also SPI NOR flash memory is utilized for faster boot time and cost reductions, low processing power to be consume. As the results, the developed proved the 10 seconds booting time, 24 frame rate/sec. and 10% lower power consumption.

Key Words : RTOS, Event data recorder, Wireless LAN, Mobile application processor

1. 서 론

차량의 속도, 위치 등 운행정보를 기록하는 장치인 차량용 영상기록저장장치는 사고 방지 효과와 저장된 자료를 활용하는 분야가 넓어지면서 자동차 부품시장에서 필요성이 증가하고 있다. 최근에는 영상기록저장장치에 WiFi 장치를 연결하여 영상정보를 스마트폰으로 전송하는 다양한 방법들이 제시되고 있으며, 또한, 출시되는 제품은 WiFi를 기본으로 지원하고 있다. 이들 제품들은 대부분 Linux 플랫폼을 사용하는데 이는 소스 라이선스 비용을 절감을 위해 모바일 AP(Application Processor)를 공급하는 칩회사가 Linux 플랫폼만을 지원하기 때문이다[1-3].

차량용 영상저장장치의 경우 1) 부팅시간을 최소화하여 녹화가 빨리 시작되는 것이 필요하고, 2) 자동차의 배터리 용량의 한계 때문에 최소한의 프로세싱 파워를 소모해야 하며, 3) 제품의 가격 경쟁력을 위해 메모리의 사용을 최소화해야 하는 요구사항 등이 있다[2]. 그러나, 기존의 Linux 플랫폼에서는 OS의 특성으로 부팅시간, 전력소모 및 메모리 사용 최소화에 한계가 있다[4, 5].

본 논문에서는 Application Processor에 RTOS를 적용하여 GPS/G Sensor 알고리즘과 차량용 블랙박스의 시나리오(UI)를 RTOS로 구현함으로써, 최대 10초대의 부팅속도와, 최소한의 메모리 사용이 가능하도록 하였다. 또한, RTOS 시스템의 약점인 Wifi 지원에 있어서도, RTOS에 TCP IP/Web server를 포팅하고 이를 USB host 인터페이스와 결합하여 프로토콜 스택을 구현함으로써 RTOS 상에서도 WiFi를 사용할 수 있도록 하였다. 최종적으로 개발된 차량용 영상기록장치는 부팅시간 10초 이내 및 프레임 전송속도 24 fps 이상, 소모전류 및 발열 성능 10% 이상 개선됨을 보여줌으로써 개발 제품의 성능을 증명하였다.

2. 차량용 영상저장장치(블랙박스)

차량용 영상저장장치의 구현에 필요한 핵심적인 기술요소는 사고 감지 기술, 정보 저장기술, 정보 운용 기술, 정보 분석 기술로 구분할 수 있다[6, 7].

사고 감지 기술은 전방, 측면 충돌센서 등을 이용하여 감지할 수 있으며, 가속도 센서를 통해 얻을 수 있는 충돌감지에 적합한 물리량을 설정한 후 기준인 임계값을 두어 임계값 이상의 수치가 감지된 경우 사고로 인식한다[8, 9].

정보 저장 기술은 다양한 데이터를 저장하고, 분석해야하기 때문에 사고 발생 후 데이터 분석을 위해 저장한 데이터를 백업할 수 있어야 한다. 사고가 난 후에는 전원이 계속적으로 공급될 수

* Corresponding Author : Dept. of Aviation and IT Convergence,
Far East University, Korea.

E-mail: nahmes@kdu.ac.kr

Received : November 8, 2017; Accepted : November 21, 2017

없기 때문에 별도로 보조 배터리를 장착하여 차량전원이 중단되었을 때, 전원공급을 계속적으로 할 수 있게 하기도 하며 데이터를 보존할 수 있도록 플래시 메모리를 사용한다[10].

정보 운용 기술은 GPS 통해 위치정보와 사고정보를 실시간으로 전송하여 신속한 응급전달 및 후송을 가능하게 하여 교통 혼잡 완화의 역할을 수행할 수 있다. 운전자 정보를 실시간으로 전송하여 운전자 및 차량 관리에도 활용이 가능하다. 저장된 정보를 바탕으로 사고 재현 시뮬레이션 구현, 차량 주행 궤적, 충돌 속도 변화, 운전자 조작 및 차량 거동 등을 재현하는 것이 가능하다[11].

정보 분석 기술은 차량의 거동이나 운전자의 조작 상태를 감지하고, 차량 충돌 시 운전자의 조작 상태, 충돌 펄스 및 여러 가지 신호들을 저장하는 EDR(Event Data Recorder)의 도입하여 자동차 사고에 대한 획기적 발전을 가져왔다. 전통적인 사고 재현 방식은 현장에 남겨진 타이어의 스키드 마크, 사고 차량의 파손 정도, 차량의 최종 정지 위치, 목격자 진술 등을 바탕으로 이루어져 현장의 주변 상황을 신속하고 면밀하게 파악하여야만 했다. 그러나 이러한 사후 정보들에 의한 분석은 사고 현장 훼손 가능성 등 정보의 불확실성으로 인해 사후 처리에 많은 어려움이 존재하였다. 차량용 블랙박스는 사고 당시의 상황을 차량 내 센서와 기록 장치를 통해 실시간으로 수집하여 사고 해석 또한 차량의 종횡 가속도, 차량 속도, 위치 정보, 주행 방향, 사고 직후의 차량 거동 등의 차량 동적 거동에 대한 데이터와 조향휠 조작(조향각도), 전장비 구동 여부(방향등, 전조등), 제동 여부, 가속 여부 등의 운전자 조작 관련 데이터를 제공하여 사고 당시의 차량 거동과 운전자의 각종 조작 상태를 표시하여 차량의 거동을 재현한다. 따라서 보다 객관적인 법적 분쟁의 증거 자료 및 도로설계, 차량 설계, 운전자의 운전성향 정보 구축 등의 자료로 활용이 가능하며 다양한 분야에 적용 가능한 DB로 구축이 가능하다. 차량의 데이터를 분석하기 위해서는 PC를 기반으로 하는 분석 시스템이 요구되며 별도의 분석 소프트웨어가 제공되어야 한다[12, 13].

3. 차량용 영상저장장치 설계

기존 차량용 영상저장장치의 부팅시간, 프로세싱 파워 소모 및 제품 가격 경쟁력을 위한 메모리 사용 최소화 문제점 개선을 위하여 모바일 AP에 RTOS를 적용하고 차량용 MP4 인코더/디코더/Muxer를 개발하고, GPS 및 G-Sensor 알고리즘과 차량용 영상 저장장치의 시나리오(UI)를 RTOS 기반하에서 구현함으로써, 최대 10초대의 부팅속도와 최소한의 메모리 그리고 Linux에 비해 낮은 성능으로 동작, 범용의 WiFi 동글을 활용하여 낮은 제품단가로 스마트폰 등에 스트리밍 기능을 제공하는 RTOS 플랫폼 영상기록장치용(블랙박스) 개발하고자 한다.

RTOS 시스템의 약점인 WiFi 지원하기 위해 TCP/IP와 Web 및 DHCP 서버를 포팅하고 이를 USB 2.0 Host 인터페이스와 결합하여 프로토콜 스택을 구현함으로써 RTOS 상에서도 WiFi를 사용할 수 있도록 한다[6]. 제품의 경쟁력은 RTOS 기반의 플랫

폼을 사용함으로써, 빠른 부팅시간과 최소한의 메모리 소모, 그리고 Linux에 비해 낮은 프로세싱 파워를 소모하는 것이 그 장점이다. 기술 개발의 주요 내용은 아래 표 1과 같다.

표 1 기술개발 주요 내용

Table 1 List of Development

List of Development	Basic Technology
RTOS device driver for WiFi USB dongle	USB OTG, WiFi, RTOS
Embedded TCP/IP stack porting and Web, DHCP Server functions in RTOS	TCP/IP, HTTP, DHCP
Encoder and Muxer for streaming services by interworking web-server and camera video	YUV Format(1420/NV12), H.264 Encoder, MP4
Real-time preview of camera video in smart phone browser	HTML, AJAX, HTTP Streaming

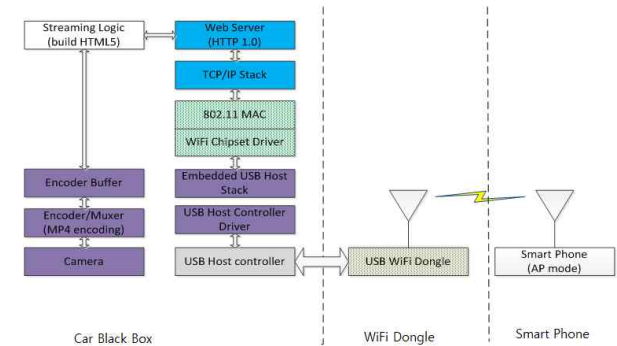


그림 1 HTTP server를 이용한 Web방식 스트리밍

Fig. 1 Web-based Streaming using HTTP server

3.1 RTOS상에서 WiFi 동영상 스트리밍 서비스가 되도록 프로토콜 설계 및 구현

기술의 핵심 개발내용은 RTOS상에서 동영상 스트리밍이 되도록 프로토콜의 구현 및 성능 개선에 있다. 안드로이드 iOS 등 스마트폰의 종류와 관계없이 스트리밍을 구현하기 위해 HTTP server를 이용한 Web방식으로 스트리밍을 구현하는 것으로 그림 1과 같다.

스마트폰의 브라우저를 통해 WiFi가 연결된 차량용 블랙박스로 HTML request 형태로 스트리밍 request를 요구한다. 개발시에는 스마트폰의 자체 브라우저를 활용하여 개발하고 상용화시 브라우저를 호출하는 형태의 어플리케이션을 개발하여 호환성과 경제성을 확보한다. USB WiFi 동글은 상용화된 USB 동글중 소프트웨어 AP기능을 제공하는 USB 동글을 대상으로 개발한다. USB Host Controller Driver/Embedded USB Host Stack은 기존 RTOS용 USB Host 드라이버를 WiFi 동글이 요구하는 기능을 지원하도록 변경한다. WiFi chipset Driver는 USB 동글 제작사

가 배포하는 리눅스용 device driver를 RTOS에 맞추어 포팅한다. TCP IP/Web Browser porting은 HTTP server를 지원하기 위해 RTOS용 오픈 소스를 이용하여 개발한다. 임베디드상에 오픈 소스작업은 성능 개선이 관건이다. 또한 Web server의 경우는, 스마트폰의 브라우저에 올라오는 request를 분석하여 차량용 블랙박스의 기능을 활성화할 수 있도록 하는 명령어 해석/수행부를 추가 개발 포팅한다. Streaming Logic은 HTTP server로부터 요청받은 내용을 muxer를 활용하여 stream을 만들고, 이를 담은 HTML 문서를 생성하여 웹 서버에 전달하는 기능을 하는 모듈로 자체 개발한다.

3.2 블랙박스 기능구현 및 FULL HD 30fps + HD 30 fps의 녹화기능

차량용 블랙박스의 기본기능도 충실하게 만족시켜야 하므로 그림 2와 같은 구성으로 시스템을 구성한다.

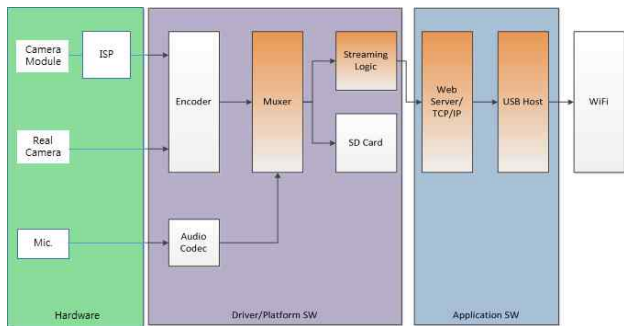


그림 2 시스템 구성
Fig. 2 Block Diagram of System

- Camera module / 후방 카메라 / Mic.
영상을 촬영하여 Application processor로 전달하는 장치로, 전방의 Full HD의 경우 ISP (Image Sensor Processor)를 통해 Image를 enhance시킨 입력 값을 전달한다. Mic는 블랙박스 동작시의 음성 등을 녹음하는 장치로 audio codec의 입력으로 들어간다.

- Encoder/Muxer
입력받은 이미지와 음성자료를 인코더 및 오디오 codec을 통해 압축하여 muxer로 전달하며, muxer는 MP4 container에 넣어 SD 카드로 저장한다. 이때 인코더의 성능은 전/후방 camera 자료를 30fps 이상으로 인코딩할 수 있어야 하며, SD의 녹화와 WiFi로의 데이터 전송 2가지를 충분히 할 수 있도록 버퍼설계 및 자료 구조의 설계가 이루어져야 한다.

- Streaming Logic
Muxer로부터의 스트리밍 자료를 웹 브라우저가 이해할 수 있는 형태의 포맷으로 변형하여 Web server/TCP IP stack을 통해 전달하는 역할을 한다. 따라서 HTML template의 생성 및 Web server request해석 Logic 등을 개발 한다.

- Web server/TCP IP
스마트로부터 전달 받은 request를 해석하여 streaming logic에 request하는 웹서버와 웹서버가 원활히 동작하도록 지원해 주는 TCP IP stack의 포팅이 관건이며, TCP IP의 performance 튜닝을 해야 한다.
- WiFi chipset Driver/ USB Host
USB host 드라이버 및 관련 모듈이 WiFi chipset 드라이버가 요구하는 기능을 만족하도록 변형하는 작업이 필요하며, WiFi chipset 드라이버의 포팅 작업이 필요하다.

4. 시스템 구성

4.1 시스템 구성

- 프로토콜
프로토콜 구조는 그림 3과 같다. Hostapd는 리눅스 커널의 mac802.11 서브시스템을 사용하는 무선 랜카드에 대해 무선공유기(Access Point) 및 인증서버 역할을 담당하는 유저 공간의 데몬 프로그램이다. 해당 프로그램은 IEEE 802.11 AP 관리 기능과 IEEE 802.1x/WPA/WPA2/EAP 그리고 RADIUS 인증서버와 클라이언트 기능을 구현했다. 802.11n으로 암호화하여 hostapd로 전달하고, Wi-Fi 동글 SoftAP에서 스마트폰으로 Request/Response에 대하여 Authentication을 수행한다. 그리고, Web Server와 DHC Server 에서도 Request/Response를 처리한다.

그림 3에서 사용한 USB 인터페이스 방식의 Wi-Fi는 Realtek 8192cu 제품으로 Wi-Fi AP(Access Point)는 일정주기 간격으로 비콘(Beacon)이라 불리는 프레임을 무선망에 브로드캐스팅 한다. 해당 프레임에는 timestamp 및 SSID(Service Set Identifier) 등 무선 네트워크에 접속에 필요한 정보를 포함하고 있다. 또한 단말은 주변 AP를 스캔하기 위해 Probe Request 프레임을 송신하는데, 이 때 AP는 응답 프레임에 해당 BSSID(Basic SSID)의 capability 정보 및 국가, 지원되는 data rate에 대한 내용 등을 포함한다. 이 후 Authentication Request 프레임을 통해 인증절

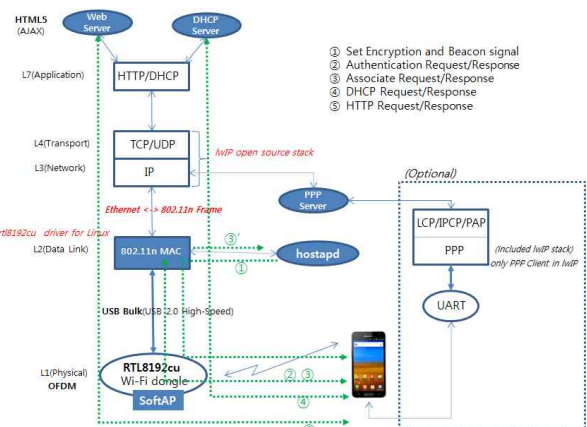


그림 3 프로토콜 구조
Fig. 3 Block Diagram of Protocol

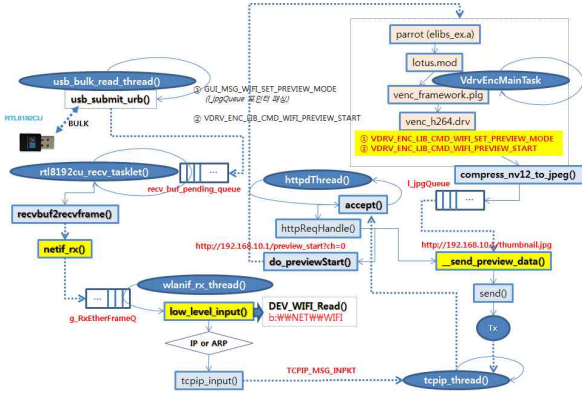


그림 4 Thread별 인터페이스 구조도
Fig. 4 Flow of Interface for each Thread

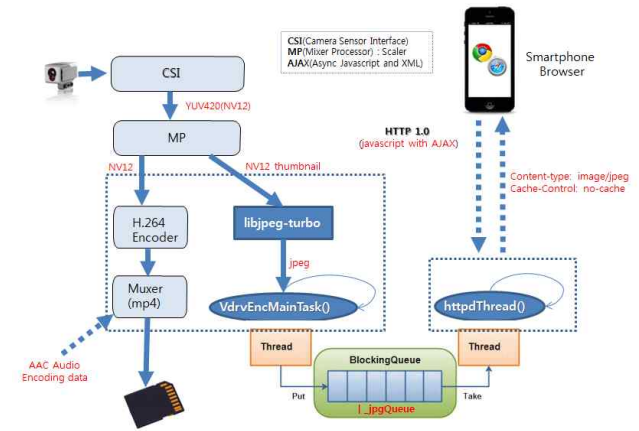


그림 6 웹 서버와의 인터페이스 구조도
Fig. 6 Flow of Web-Server Interface

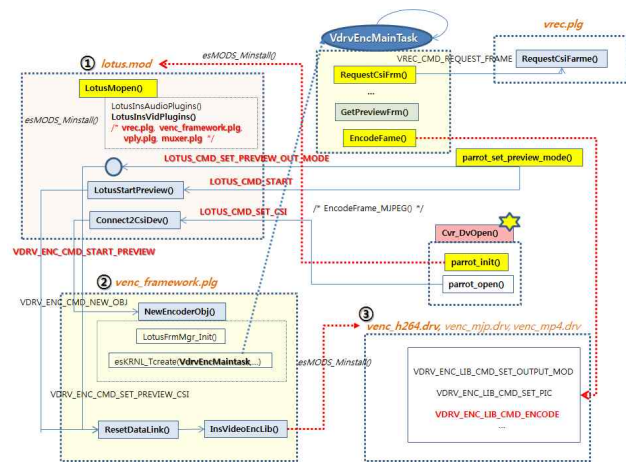


그림 5 비디오 인코더 초기화 과정
Fig. 5 Flow of Initialization Video Encoding

차를 거친 후 Association 프레임을 주고 받으면서 실제 AP에 접속하는 절차를 마무리 한다.

- Thread별 인터페이스 구조도
Thread별 인터페이스 구조도는 그림 4와 같다.
- 비디오 인코더 초기화 과정
비디오 인코더 초기화 과정은 그림 5와 같다. 보내진 비디오 신호는 lotus.mod에서 초기화 되어, venc_framework.plg에서 전달되고 venc_h264.driv을 통하여 최종 초기화 된다.
- 웹 서버와의 인터페이스 구조도
웹 서버와의 인터페이스 구조도는 그림 6과 같다. 카메라의 영상은 H.264로 인코딩되어 Muxer 거쳐 메모리 카드에 저장된다. 또한, libjpeg-turbo를 통해 VdrvEncTask() 과정을 거쳐 Thread 형태로 httpThread()를 거쳐 스마트폰으로 전달된다.
libjpeg-turbo의 해당 라이브러리는 YUV 포맷의 영상 이미지를 JPEG 이미지로 인코딩해주는 오픈소스 라이브러리인 libjpeg

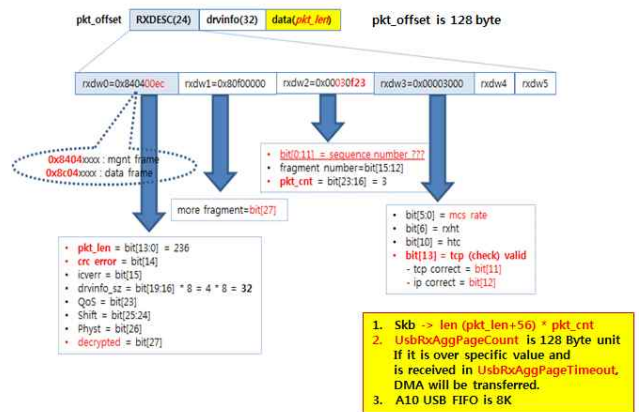


그림 7 RTL8192cu USB 패킷 구조
Fig. 7 Structure of RTL8192cu USB Packet

과 동일한 기능을 지원한다. 그러나 ARM의 NEON이라 불리우는 SIMD(Single Instruction Multiple Data) 명령어를 사용하여 libjpeg보다 2~4배 정도의 속도 향상을 지원하는 오픈소스 라이브러리이다.

- USB 2.0 OTG Host 인터페이스
USB 2.0 OTG Host 인터페이스를 위한 RTL8192cu USB 패킷 구조는 그림 7과 같다. pkt_offset은 128바이트 단위로 처리하고, 이상의 데이터에 대해서는 일정 시간(UsbRxAggPageTimeout) 이내에 수신되는 경우만 DMA로 전송한다. A10 USB FIFOsms 8K이다.

USB Bulk 통신 함수 흐름도는 그림 8과 같다. Input 설정하고 송수신 데이터 버퍼 및 완료 함수 등록하고 큐에 저장한다. Short Packet은 최대 Packet 512바이트 보다 작은 것을 의미한다. HCD(Host Controller Driver) 초기화시 USB OTG는 인터럽

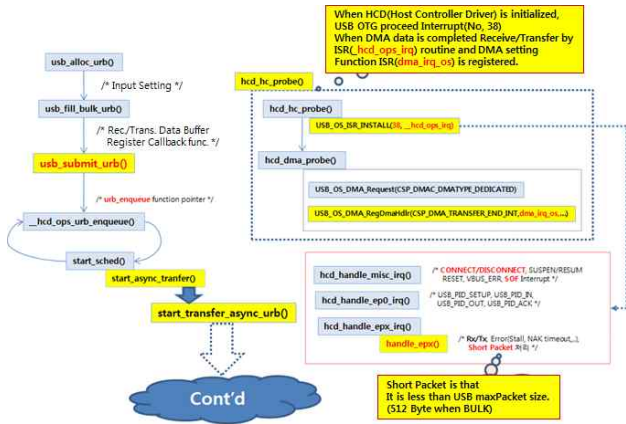


그림 8 USB Bulk 통신 함수 흐름도
Fig. 8 Flow of USB Bulk Communication Functions

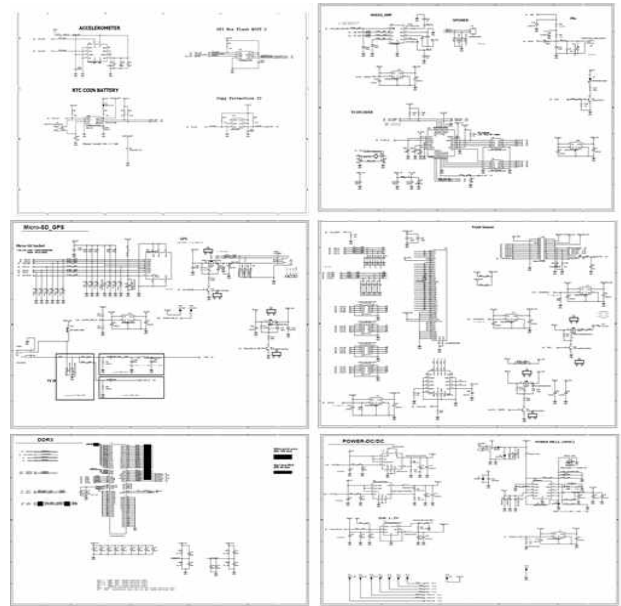


그림 11 하드웨어 설계도(일부분)
Fig. 11 Design of Hardware(Partial)

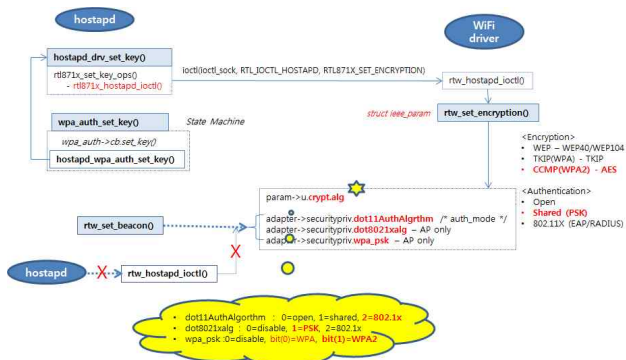


그림 9 Authentication
Fig. 9 Authentication



그림 12 개발된 하드웨어
Fig. 12 Developed Hardware

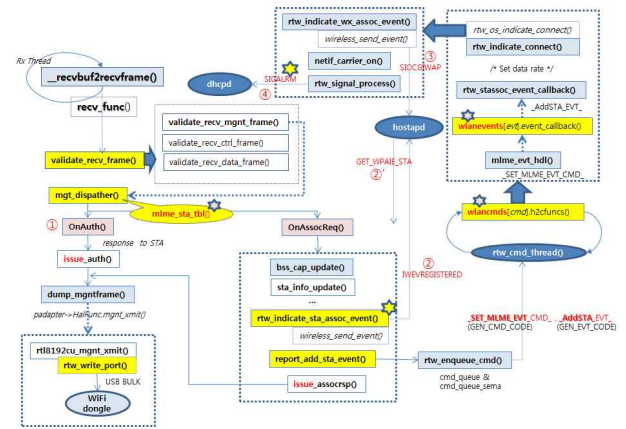


그림 10 Association
Fig. 10 Association

트 38번으로 처리한다.
8192cu Wi-Fi 관리 프레임 처리를 위한 Authentication 및

Association 과정은 각각 그림 9, 그림 10과 같다.

4.2 하드웨어

사용된 AP 칩은 AllWinner사의 A10이며 기본 특징은 아래와 같다.

- CPU : Cortex-A8 1GHz 1 Core (VFPv3 and NEON SIMD)
- GPU : Mali400
- VPU : CedaX (H.264 1080p@60fps encoding)
- RAM : DDR2/DDR3 up to 2G
- Flash : NAND up to 64GB and NOR(using SPI)
- LCD/LVDS/HDMI Interface
- SPI/I2C/USB2.0/SD Card Interface

개발된 하드웨어 보드의 기본 설계도와 실물은 각각 그림 11, 그림 12와 같다.

5. 시험 및 결과

개발 결과물에 대한 시험 환경은 그림 13과 같다. 웹 서버와의 인터페이스 구조도에서 보듯이, 전후방 카메라에서 입력되는 영상은 H.264 비디오 인코더와 AAC 오디오 인코더에서 인코딩된 후 muxer를 통해 mp4 파일 포맷으로 SD 카드에 저장된다.

스마트폰은 먼저 무선 AP(Access Point) 모드로 동작하는 WIFI USB Dongle를 통해 개발보드에 WiFi로 접속한다. 이 후 스마트폰 브라우저에서 http://192.168.10.1로 접속해서 HTTP 스트리밍 영상 요청을 할 경우, H.264 비디오 인코더에서 함께 출력되는 YUV 이미지를 libjpeg-turbo 라이브러리를 이용해 JPEG 이미지로 변환 후 스마트폰 브라우저로 송신한다. 아래 시험환경 그림에서 LCD에 출력되는 영상과 스마트폰 브라우저에 출력되는 영상이 같다는 것을 통해 실시간 스트리밍이 진행되고 있음을 확인할 수

있다.

아울러 SD 카드에 저장되는 영상은 파일이름 형태로 저장되며 이때 맨 앞의 첫 글자 'F'는 전방 영상을 'R'은 후방 영상을 의미하며, 파일당 녹화시간은 기본 1분이다.

개발 결과물의 시험 데이터 측정을 위해 스마트폰 브라우저에서 표시되는 정보는 그림 14와 같으며 “avg fps”를 통해 개발보드에서 초당 수신되는 평균 프레임수를 표시한다.

위에서 설명한 시험환경 하에서 전후방 2채널 녹화와 동시에 WIFI를 통해 스마트폰 기본 브라우저에서 측정된 평균 초당 수신 영상 프레임수(fps)는 표 2와 같다. 표3에는 평균 부팅시간(10회 측정 평균값), 메모리 사용 측정값 및 코드 사이즈 측정값을 표시하였다. 부팅시간은 10초로 일정하게 측정되었으며 Working Ram은 128MB, 최종 코드 사이즈는 8MB NOR 플래시중 6.1MB Code로 측정되었다.

표 2 평균 초당 수신 프레임수(fps)

Table 2 No. of Average Frames per Second

Measuring Time(Min.)	No. of Frames/Sec.
1	25.7
3	25.1
5	24.9
15	24.1
20	23.7

표 3 부팅시간, 메모리, 코드 사이즈

Table 3 No. of Average Frames per Second

구분	Results
Booting Time.	10 Sec.
Working RAM	128 MB
Code Size	6.1MB Code + User Data Space in 8MB NOR Flash

6. 결 론

본 논문에서는 기존 Linux 플랫폼의 차량용 영상저장장치의 문제점인 부팅시간, 전력소모 및 메모리 사용 최소화를 해결하기 위하여, Application Processor에 RTOS를 적용하여 GPS/G Sensor 알고리즘과 차량용 블랙박스의 시나리오(UI)를 RTOS로 구현함으로써, 최대 10초대의 부팅속도와, 최소한의 메모리 사용이 가능하도록 하였다. 또한, RTOS 시스템의 약점인 Wifi 지원에 있어서도, RTOS에 TCP IP/Web server를 포팅하고 이를 USB host 인터페이스와 결합하여 프로토콜 스택을 구현함으로써 RTOS 상에서도 WiFi를 사용할 수 있도록 하였다. 최종적으로 개발된 차량용 영상기록장치는 부팅시간 10초 이내 및 프레임 전송속도 24 fps 이상, 소모전류 및 발열 성능 10% 이상 개선됨을 보여줌으로써 개발 제품의 성능을 증명하였다.

본 논문은 RTOS를 사용하여 MP4 encoder 및 muxer 부분을 개발하여, 최대 10초대의 부팅속도와 최소한의 메모리 그리고 리눅스에 비해 낮은 파워 소모로 동작하는 장점을 가지며 범용의 WiFi Dongle를 활용하여 낮은 제품단가로 스마트폰 등에서 스트리밍 기능을 제공하는데 그 성과가 있다고 할 수 있다. 또한 RTOS 시스템의 약점인 WiFi 지원에 있어서도 RTOS에 TCP/IP와 Web 및 DHCP 서버를 포팅하고 이를 USB OTG Host 인터페이스와 결합하여 프로토콜 스택을 구현함으로써 RTOS 상에서도 WiFi를 사용할 수 있도록 하였다.

본 논문을 통해 향후 제품화될 제품의 경쟁력은 Linux를 사용할 경우 리눅스 자체 커널 크기와 탑재되는 프로그램들로 인해 전체 코드 사이즈가 증가하여 NAND 플래시를 사용할 수밖에 없고 그로 인해 제품단가가 비싸지는 문제점이 있다. 그에 비해 저용량의 SPI NOR 플래시를 사용하면서도 WiFi가 지원되는 RTOS 기반의 플랫폼을 사용함으로써 빠른 부팅시간과 부품 단가 절감 그리고 리눅스 비해 낮은 프로세싱 파워를 소모하는 것이 그 장점이자 핵심기술이다.

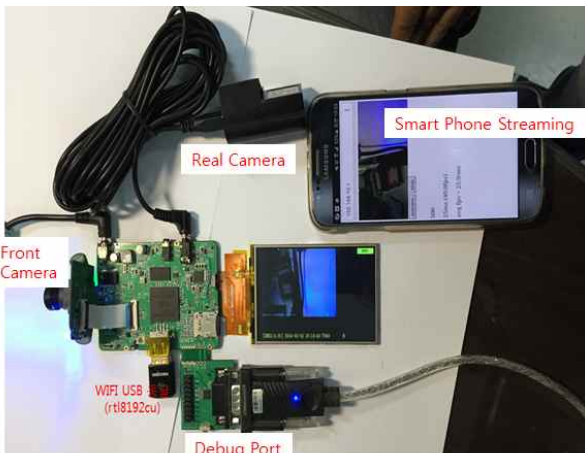


그림 13 시험 환경

Fig. 13 Test Environment

감사의 글

이 연구는 2017년도 극동대학교 교내연구비 지원에 의하여 수행된 것임(FEU2017S04)

References

[1] Yungyu Kim, Bum Han Kim, Dong Hoon Lee, Real-time Integrity for Vehicle Black Box System. Journal of The Korea Institute of Information Security & Cryptology, vol. 19, no. 6, pp. 49-61, Dec. 2009.

[2] Jaeduck Choi, Kangsuk Chae, Souhwan Jung, Video Data Collection Scheme From Vehicle Black Box Using Time and Location Information for Public Safety. Journal of The Korea Institute of Information Security & Cryptology, vol. 22, no. 4, pp. 771-783, Aug. 2012.

[3] Seong-Deok Han, An Implementation of Car Black Box on Embeded Linux System. M.S Theses, Ajou University, 2008. 2.

[4] Dea-Woo Park, Jeong-Man Seo, A Study of Using the Car's Black Box to generate real-time Forensic Data. Journal of the Korea society of computer and information, vol. 13, no. 1, pp. 253-260, 2008.

[5] Chung Se Myoung, A study on the implementation of Car Black Box system based on WinCE O/S and SoC platform. M.S Theses, Chonbuk National University, 2009. 2.

[6] Jang-Ju Kim, Implement of blackbox with in vehicle network data and the external sensor data. M.S Theses, Dongeui University, 2011.

[7] Jang-Hyeok Yun, Jin-Il Kim, Implementation of A Car Video Blackbox System using Smartphone. The Journal of Korean Institute of Information Technology, vol. 8, no. 10, pp. 135-142, Oct. 2010.

[8] Hyung-Chul Won, Design and implementation of android based car blackbox system. M.S Theses, Pusan National University, 2011.

[9] Yoshitaka Deguchi, Kouichi Kuroda, Makoto Shouji, Taketoshi Kawabe, HEV Charge/Discharge Control System Based on Navigation Information. JSAE Proceeding, Vol.29, No.3, pp. 1-4, 2003.

[10] Arun Rajagopalan, Gregory Washington, Intelligent Control of Hybrid Electric Vehicles Using GPS Information. SAE, 2002.

[11] Jeom-Hun Lee, In-One Joo, Seong-Kyun Jeong, Sang-Uk, Lee, GPS Jamming Monitoring System Prototype. 2012 Joint Conference on Satellite Communications,

2012.

[12] Seongkyun Jeong, Taehee Kim, Sanguk Lee, and Jaehoon Kim, Test Environment Establishment for GNSS Spoofing Interference Detection. Asia Navigation Conference, Busan, 2013.

[13] Inone Joo, Jeom-Hun Lee, Sanguk Lee, and Jaehoon Kim, Data Acquisition Prototype for Analyzing GPS Jamming. Asia Navigation Conference, Busan, pp. 83-89, 2013.

저 자 소 개



남 의 석 (Eui-Seok Nahm)

1991년 2월 : 연세대학교 전기공학과(공학사).
1993년 2월 : 연세대학교 전기공학과(공학석사).
1998년 2월 : 연세대학교 전기공학과(공학박사).
1998년~2002년 : LS산전 책임연구원.
2003년 2월~현재 : 극동대학교 항공IT융합학과 교수