

논문 2010-47IE-1-7

특정목적 수행을 위한 임베디드 시스템 플랫폼의 최적 선택

(An Optimal Selection of Embedded Platform for Specific Applications)

문 호 선*, 김 용 득**

(Ho Sun Moon and Yong Deak Kim)

요 약

본 논문은 특정목적의 수행을 위해 설계될 임베디드 플랫폼의 효율적인 선택에 관하여 다룬다. 특히, 영상 및 음향 신호 처리를 통해 실시간 상황인지를 목적으로 하는 차량 시스템을 예로 들어 설명한다. 실험을 위하여 검증된 성능평가 프로그램들 중에서 응용목적에 맞는 프로그램을 선택한 후 시뮬레이터를 사용하여 ARM7, ARM9, ARM10 프로세서 코어를 기준으로 코드 및 데이터 메모리 크기, 명령어 수, 실행 사이클을 측정하였다. 이 연구에서 제안된 방법으로 특정목적 수행을 위한 하드웨어 최소사양이 측정 가능하였으며, 이 결과는 설계 및 구현에 많은 시간이 소요되는 임베디드 시스템의 설계단계에서 시스템의 목적 수행에 필요한 플랫폼 사양을 도출하여 효율적인 플랫폼선택의 기준을 제시하는 것이 가능함을 보였다.

Abstract

The goal of this paper is to determine optimal hardware platform for specific applications. In order to develop an understanding of how select the optimal platform, we focus upon the real-time embedded vehicle system for processing forward image and sound. In this paper we propose to measure parameters such as instructions, execution cycle, required memory size for program and data by using ARMulator. We have measured three types of processor cores: ARM7, ARM9 and ARM10. The results of the study indicated that the proposed methods could measure the minimal requirements of hardware platform for specific applications. By defining lower limit of hardware specifications in embedded systems, we can minimize expenses with suitable system performance without implementing the system.

Keywords : Embedded, Real-Time, Platform, Simulation, ARMulator

I. 서 론

최근 프로세서, 집적회로, 신호처리 기술의 급격한 발전에 힘입어서 다양한 기존제품들의 디지털화가 광범위하게 이루어지고 있고, 이와 같은 이유로 특정한 목적을 수행하기 위하여 전용하드웨어와 소프트웨어로 구성된 임베디드 시스템에 대한 개발 또한 산업 전 분야에 확대되어지고 있다.^[1]

이렇듯 급변하는 주위환경과 수많은 임베디드 시스템의 출현에도 불구하고 일부 숙련된 개발자만이 신규

제품을 독자적으로 설계 할 수 있는 능력을 보유하고 있으며, 많은 개발자들은 임베디드 프로세서 제조업체에서 제공하는 임베디드 플랫폼의 일부를 수정하여 사용하고 있고, 이는 개발된 임베디드 시스템을 통하여 구현하고자 하는 목적에 최적화된 플랫폼이 아닐 가능성이 높다.

하지만, 하루가 다르게 변하는 시장의 요구사항에 대응하기 위해서는 플랫폼에 대한 최적화를 통하여 얻을 수 있는 비용절감을 포기하더라도 원하는 성능을 보유한 제품의 개발이 필수적이므로 실제 필요한 사양보다 높은 사양의 플랫폼을 적용하는 경우가 빈번하다.

본 연구에서는 이와 같이 발생될 수 있는 자원의 낭비를 줄이기 위한 방법으로 하드웨어 플랫폼의 설계 이전에 시뮬레이션 실험을 통하여 시스템에 필요한 하드

* 평생회원, 향로표지기술협회 연구소
(Korea Association of Aids to Navigation)

** 평생회원, 아주대학교 전자공학부
(Dept. of Electronics Engineering Ajou University)
접수일자: 2010년1월8일, 수정완료일: 2010년3월10일

웨어의 최소사양을 도출할 수 있는 방안을 제안한다. 먼저, 제품의 목적을 위해 구현될 여러 가지 알고리즘들은 임베디드 시스템의 성능측정에 사용되는 다양한 벤치마크 소프트웨어 및 이미 검증된 소프트웨어 모듈에서 선택하여 준비하고, ARMulator^[2]를 사용하여 실제 사용할 프로세서 코어들을 대상으로 구현에 필요한 코드 및 데이터 메모리의 크기, 명령어 개수 및 실행 사이클 등을 측정 후 그 결과를 분석하여 주어진 응용 프로그램에 부합되는 플랫폼을 선택할 수 있도록 한다.^[3]

본 연구에서는 주행 중인 차량에서 주변 환경을 인식하기 위한 임베디드 시스템의 예를 들어 플랫폼의 최적 선택 방법에 대한 내용을 기술하려 한다. 차량용 임베디드시스템을 대상으로 선정한 이유는 관심분야이며, 영상, 음향 등 다양한 입력을 처리할 수 있고, 실시간성이 필요한 응용분야이며, 많은 연구가 진행되어지고 있기 때문이다.

이후의 구성은 다음과 같다. II장에서 특정한 목적을 수행하기 위한 임베디드 하드웨어 플랫폼에 대해 차량용 시스템의 예를 들어 설명하며, III장에서는 성능평가를 위해 사용되어지는 벤치마크 소프트웨어의 종류 및 특징에 대해 기술하고, IV장에서는 성능평가 환경 및 실험결과를 보인다. 마지막으로 V장에서는 결론을 내린다.

II. 특정목적 수행을 위한 임베디드 플랫폼

임베디드 시스템의 하드웨어 플랫폼은 프로세서, 메모리, 입출력장치로 구성되어 있으며, 입력장치를 통해 획득된 데이터를 정해진 프로그램에 따라 처리하여 그 결과를 출력하며, 본 논문에서는 전방영상 및 음향에

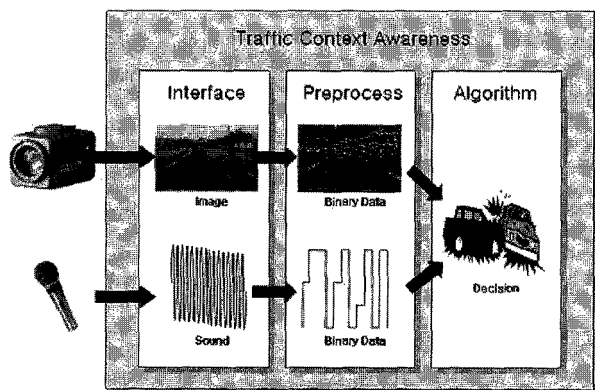


그림 1. 영상 및 음향을 이용한 상황인지 시스템
Fig. 1. Context Aware System using Image and Sound.

대한 처리를 통해 차량 주변의 실시간 상황인지를 수행하는 시스템의 예로 설명하였다.

영상 및 음향을 이용한 상황인지 시스템은 <그림 1>과 같이 획득과정과 전처리과정 그리고 인지과정으로 이루어지며, 획득과정을 제외한 전처리 및 인지과정은 시스템의 메모리크기, CPU의 연산성능에 따라 달라지므로 이러한 하드웨어 플랫폼에 대한 효율적인 선택과정이 필수적인 것으로 판단된다.

본 절에서는 구현하려는 영상처리 및 음향처리를 위한 응용 소프트웨어를 예로 그 결과를 설명한다.

1. 영상처리

영상을 통한 상황인지에는 매우 다양한 방법이 존재하지만 본 연구에서는 차량 전방의 차선정보, 전방차량 및 장애물정보등의 교통파라미터를 분석하기위한 영상처리 기법들을 적용하였으며, 본 연구에서는 실제 차량 시스템에 적용하기 위한 세부적인 영상처리 알고리즘들을 윈도우 운영체제 기반의 개인용 컴퓨터에 이식하여 단계별 처리결과를 보였으며, <그림 2>와 같다.

본 연구에서 사용하는 영상은 차량 앞부분에 장착한 캠코더에서 수집된 영상을 320×240 8비트 흑백영상으로서 도로와 차량 및 차선의 명암 차를 찾아내기 위해 Sobel Edge Detection^[4] 방법을 사용하여 <그림 3 (b)>와 같은 명암차가 생기는 윤곽을 추출하였고, 도로를 제외한 부분에서 검출된 윤곽을 이진영상으로 변환하기

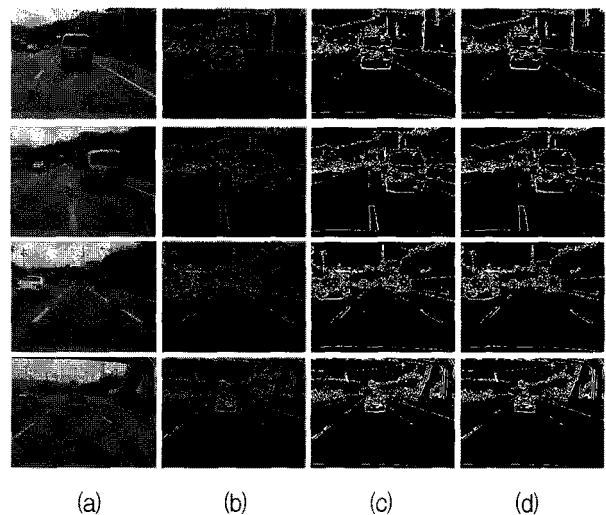


그림 2. 영상처리 결과 (a) 원본영상 (b) 윤곽검출 (c) 이진영상 (d) 라벨링을 통한 영상
Fig. 2. Results of Processing.
(a) Original Image (b) Edge Detect (c) Binary Image (d) Labeling Image

위해서 배경인 도로와 전방 차량, 차선 등의 물체집단으로 나누어 두 집단의 통계량이 다르도록 임계값을 결정하는 방법인 판별분석법^[5]을 적용하여 처리한 결과가 <그림 3 (c)>와 같다.

이진영상은 본 논문에서 상황인지를 위한 대상으로 하는 차선, 전방차량, 장애물 등을 원본영상처럼 잘 나타낼 수 있을 뿐 아니라 원본영상의 한 픽셀에 해당하는 8비트 데이터를 1비트로 줄일 수 있게 되어 데이터 처리속도를 높일 수 있게 되지만, 윤곽 추출 후 이진영상으로의 변환 과정을 거치게 되면 원하지 않던 잡음이 생기게 되고, 또 이들. 잡음은 실리는 위치가 불규칙적이며, 크기가 불규칙한 특징을 가지게 되므로^[5] 이러한 잡음을 제거하기 위해 독립 픽셀의 제거와 라벨링을 통한 잡음 성분 제거 방법을 사용하였다. 이진화된 영상에서 독립픽셀의 주위 3×3 픽셀을 비교하여 독립 픽셀 및 연결된 픽셀수가 임계값 이하로 구성된 라벨을 이진영상에서 제거하는 방법이며, 라벨링은 연결되어 있는 모든 픽셀에 번호를 붙이고, 다른 연결성분에는 다른 번호를 붙이는 처리로서, 그 원리는 <그림 3>과 같다.^[5]

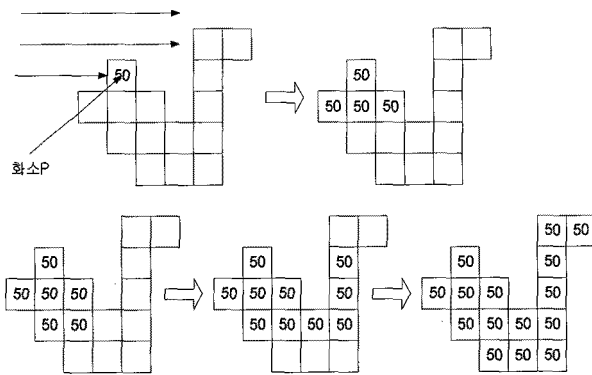


그림 3. 라벨링의 과정
Fig. 3. The Process of labeling.

2. 음향처리

차량의 실시간 상황인지를 위해서는 영상데이터 뿐 아니라 사고음향을 포함한 차량 주변의 여러 종류의 음향을 다른 소음과 구별하여 인식하는 과정을 거쳐야 한다. 모든 음향에는 각기 다른 주파수 특징이 있고, 사고음향도 다른 음향들과 구별되는 특징이 있다.^[6]

주행 중인 차량시스템은 다양한 소음을 획득하게 되며, 여러 가지 소음 중에서 자동차의 경적이나 긴급차량의 사이렌 등과 같이 주파수 성분이 사고음향 등 다

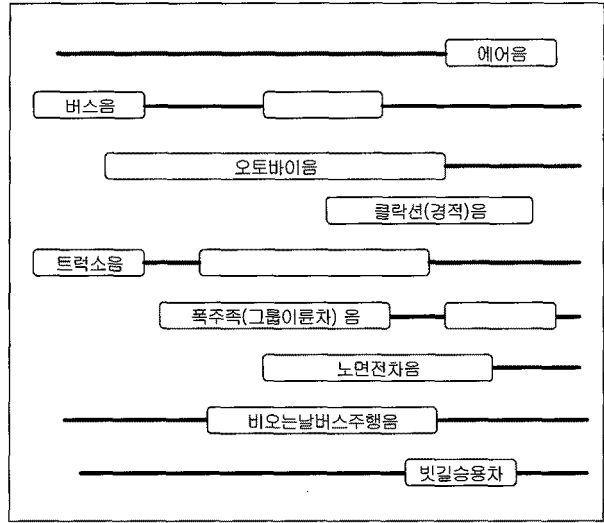




그림 4. 교통음의 주파수 분석결과의 모형도
Fig. 4. Frequency analysis of various traffic sounds.

른 음향과 명확히 구분되기도 하지만, 대부분 교통음은 사고음과 주파수 성분이 유사하기 때문에 주파수 성분만에 의해서 사고음을 식별하는 것은 매우 어려움이 따른다.^[7] 이러한 다양한 교통음의 일부를 주파수로 해석한 결과는 <그림 4>과 같으며,^[7] 각각의 음원에 대하여  로 표시한 주파수대는 에너지가 높고,  로 표시한 주파수는 에너지를 낮음을 표시하였다.

급브레이크 음은 저주파 성분으로 복수의 정점을 유지하는 특징이 있으며, 낮은 레벨의 음은 비교적 넓은 주파수 성질을 가지고 있다. 오토바이 통과음은 저주파 성분과 고주파성분이 대부분 2가지로 나누어져 시간경과에 의해 쇠퇴하지 않는 특징을 나타내며, 경적음은 0~13KHz까지 비교적 평범한 특징을 나타내고 있다.^[7] 교통사고 음은 승용차대 승용차의 충돌의 경우, 승용차와 오토바이와의 충돌의 경우에도 비교적 유사한 특징을 나타내고 있고, 충돌 개시 전에 넓은 주파수에 걸쳐 대부분 균일한 레벨의 분포를 나타내고 있다. 그러나 이 특징적인 성분은 단시간에 쇠퇴하고 그 후에는 비교적 낮은 주파수 성분이 지속되는 특징이 나타난다.^[7]

획득된 Sound는 PCM(Pulse Code Modulation)을 통한 A/D변환에 의해 디지털 신호로 변환하고, 이 시간영역에서의 신호를 주파수영역의 신호로 바꾸기 위해 고속 푸리에 변환을 수행하고, 여기에서 얻어진 주파수 스펙트럼 특징을 검출하여 음향을 통한 상황인지를 수행한다.

$$x(n) = \frac{1}{n} \sum_{k=1}^N a(k) \cos\left(\frac{2\pi(k-1)(n-1)}{N}\right) + b(k) \sin\left(\frac{2\pi(k-1)(n-1)}{N}\right) \quad (1)$$

where $a(k) = \text{real}[X(k)], b(k) = \text{imag}[X(k)], 1 \leq n \leq N$

Fourier Transform은 연속 시간인 아날로그 신호를 주파수 영역으로 바꾸어 신호에서 추출하는 특정파라미터를 찾기 쉽게 바꾸어주는 알고리즘으로서, 본 연구에서는 디지털 처리에서 가장 많이 사용되는 FFT 알고리

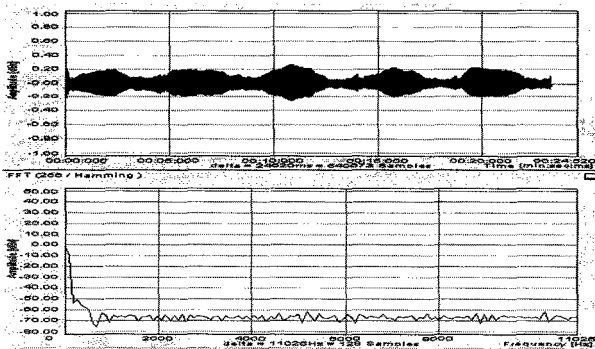


그림 5. 구급차 사이렌음
Fig. 5. ambulance siren.

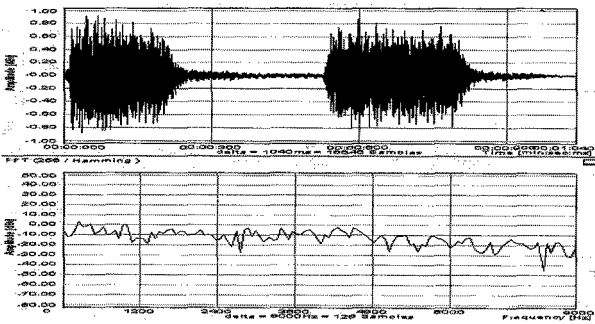


그림 6. 경적음
Fig. 6. horn sound.

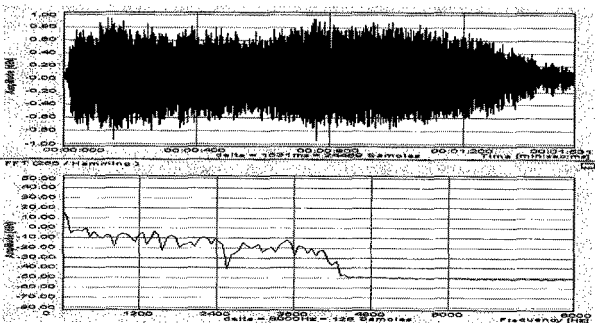


그림 7. 스키드음
Fig. 7. Skid sound.

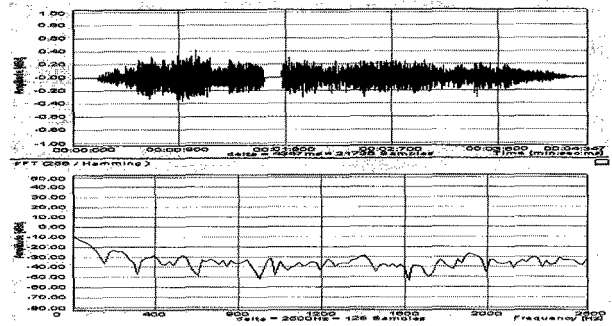


그림 8. 자동차 충돌음
Fig. 8. Car crash sound.

즘으로 음향신호를 처리하였으며 식(1)과 같다.^[6]

<그림 5>에서 <그림 8>는 차량 사고에 관련된 음향 데이터를 PCM변환하여 표시하였고, 변환된 데이터는 고속푸리에 변환하여 그 결과를 나타내었다.

위의 변환결과에서 보이듯이 차량운행 중 발생 가능한 음향데이터를 고속푸리에 변환을 수행하면, 특정 주파수 패턴을 가지는 것을 볼 수 있으며, 이를 통해 음향 데이터를 통한 특정 상황 인지가 가능함을 알 수 있다.

III. 성능측정을 위한 벤치마크 소프트웨어

시스템의 하드웨어 및 소프트웨어를 설계하는 단계에서 구현하려는 시스템의 필요자원 및 성능을 미리 알 수 있는 방안은 경험적인 예측에 의한 방법 밖에 없으며, 본 연구에서는 검증된 벤치마크 소프트웨어를 조합하여 최대한 구현목표와 근접하게 설정하고 그에 해당하는 여러 코드를 목표 프로세서코어 상에서 각각 성능측정 한 후, 그 결과에 따라 필요한 하드웨어 플랫폼의 사양을 도출하는 방법을 사용한다.

응용프로그램의 성능을 측정하기 위해서 멀티미디어 프로그램 및 임베디드 시스템에서 사용되는 응용 프로그램의 벤치마크를 위한 공개 코드 모음인 미디어벤치(MediaBench)^[8-9]와 미벤치(MiBench)^[10-11] 및 ARM社에서 제공하는 ARM 응용프로그램(ARM application libs)의 세 종류의 벤치마크 수트(suite)를 사용하였으며 그 내역은 <표 1>과 같다.

미디어벤치 1.0의 경우 영상처리, 통신 및 DSP(digital signal processing)관련된 JPEG, MPEG, GSM, G.721 음성압축, PGP, PEGWIT, Ghostscript, RASTA, EPIC, ADPCM의 19가지의 응용프로그램으로 구성되어 있으며, 미벤치의 경우 임베디드 시스템을 위한 벤치마

표 1. 벤치마크 프로그램
Table 1. Benchmark Program.

Suite	Field	Program	
Media bench	Image	jpeg, mpeg	
	Comm.	gsm, g.721, pgp	
	DSP	pegwit, ghostscript, mesa pasta, epic, adpcm	
Mibench	Auto	basicmath, bitcount, qsort susan(edge, corners, smoothing)	
		Consumer	jpeg, lame, mad, tiff2bw, typeset tiffrgba, tiffdither, tiffmedian
	Office	ghostscript, ispell, rsynth sphinx, stringsearch	
		Network	dijkstra, patricia, crc32, sha blowfish
	Security		blowfish(enc, dec), pgp sign, sha pgp verify, rijndael(enc, dec)
		Telecomm.	crc32, fft, ifft, adpcm(enc, dec) gsm(enc, dec)
	ARM application library		Signal
		Processing	g.711, huff, is54, math, mtrmf

크 수트로서 자동차제어(automotive and industrial control), 가전장치(consumer devices), 사무자동화(office automation), 통신(telecommunications), 네트워크, 보안(security)에 관련된 35종류의 벤치마크 프로그램을 제공하며, ARM 응용 라이브러리는 ADS 패키지에서 제공하는 다양한 신호처리를 위한 소프트웨어로서 음성 및 영상처리를 위한 adpcm, DCT, G.711 등 10종류의 프로그램을 제공한다.^[8-11]

IV. 실험 및 결과

1. 실험환경

실험을 위해서는 측정의 대상이 되는 벤치마크 프로그램과 이를 컴파일하여 실행 가능한 파일로 변환해주는 개발환경인 ADS1.2(ARM Developer Suite) 그리고 ARM GUI 디버거인 AxD가 제공해주는 가상실행환경

에서 ARMulator를 사용하여 시뮬레이션 실험을 실시하였다.^[2, 12]

ARMulator는 ADS에 포함된 명령어 세트 시뮬레이터(Instruction Set Simulator)로서 ARM사에서 제작한 여러 프로세서 코어로 시뮬레이션 실험이 가능하며, 본 실험에서는 ARM7, ARM9, ARM10 프로세서 코어를 기준으로 하여 명령어 개수, 실행 사이클 수를 측정하였고, 시뮬레이션 과정을 진행한 호스트 시스템은 Pentium-IV 3.0GHz의 중앙처리 장치와 SDRAM 2GB의 주기억 장치, MS사의 windows XP 운영체제 기반의 개인용 컴퓨터를 사용하였다.

2. 실험방법 및 결과

본 실험에서는 개발하려고 하는 특정 응용을 대상으로 검증된 소프트웨어를 적용하여 그 결과를 측정하는 것을 주제로 하고 있으며, 본 연구에서 예를 들어 설명한 차량 시스템을 대상으로 하여 주변 환경을 인지하기 위한 응용 소프트웨어를 적용하여 <그림 9>와 같은 절차로 실험하였다.

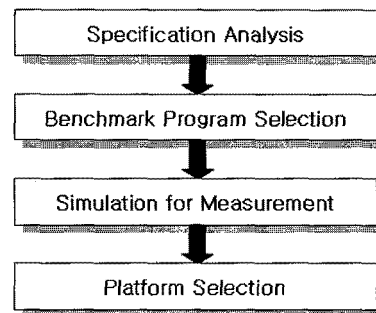


그림 9. 실험절차
Fig. 9. Process of Experiment.

먼저 구현하고자 하는 시스템의 주요 기능을 정리한 후 각각의 기능의 특성에 따라서 시뮬레이션 실험이 필요한 그룹과 필요하지 않은 그룹으로 분류하며, 시뮬레이션이 필요한 그룹은 기준은 다음과 같다.

- 1) 주기적으로 실행되는 기능
- 2) 시스템자원을 많이 사용하는 기능
- 3) 실시간성이 필요한 기능

다만, 위의 조건을 만족하더라도 단순 센서 입출력 또는 짧은 시간 내에 처리할 수 있는 기능은 시뮬레이션 대상에 포함하지 않았다.

시뮬레이션이 필요한 기능들은 그 유사성에 따라서 각각의 검증된 벤치마크 프로그램들로 대체하며, 이들 프로그램의 코드크기 및 실행속도를 측정한다. 여러 조건에 따라 시뮬레이션을 수행하였다면 그 결과를 분석하여 해당 시스템에 효율적으로 적용될 수 있는 하드웨어 플랫폼을 선택한다.

<표 2>에서는 일반적으로 실시간 상황인지를 위한 차량용 임베디드 시스템에 필요한 기능을 정리하였으며^[3], 이들을 시뮬레이션 필요 여부에 따라 분류하였고, 구현을 제외할 기능 및 동작 측정이 필요 없는 단순 동작을 시뮬레이션 대상에서 제외하였다.

표 2. 차량시스템의 필요기능
Table 2. Required functions for vehicle.

Parameter	Function	Simulation
영상처리	전방차량	O
	차선정보	O
	장애물정보	O
	신호정보	X
	기상상태	X
음향처리	차량이상음	O
	스키드음	O
	경적음	O
	충돌음	O
GPS데이터	차량속도	X
	차량위치	X
	시간정보	X
입출력센서	충돌센서	X
	가속도센서	X

표 3. 벤치마크 프로그램의 선택
Table 3. Benchmark Program Selection.

Parameter	Function	Benchmark Program
영상처리	전방차량	susan(edge, corners, smoothing)
	차선정보	
	장애물정보	
음향처리	차량이상음	g.711, adpcm, fft
	스키드음	
	경적음	
	충돌음	

<표 3>에서는 시뮬레이션이 필요한 기능과 각각의 벤치마크 프로그램을 그 유사성에 따라 나타내었고, 시뮬레이션에 적용한 벤치마크 프로그램을 실행 가능한 이미지로 변환하여 실제 코드 및 읽기전용 데이터(RO-Data)에 필요한 메모리 용량과 데이터 메모리에 필요한 메모리용량을 <표 4>에 나타내었다.

<표 5>에 ARM920T 프로세서 코어에서 벤치마크 프로그램들을 실행한 결과를 보였으며, 각각의 동작에 필요한 입력 파라미터, 명령어의 개수, 실행에 필요한 사이클 수 및 200MHz의 클럭을 기준으로 측정한 msec 단위의 실행시간을 나타내었다. 실행시간은 실제 프로세서에 입력되는 기준 클럭의 속도와 실행 사이클의 수에 따라 결정된다. 예를 들어, 100MHz의 기준 클럭을 가진 프로세서 코어가 1,000,000사이클을 실행하기 위해서는 10msec의 시간이 소요된다.

<그림 10>에는 ARM7, ARM9, ARM10 프로세서 코어를 대상으로 측정한 필요 명령어, 실행 사이클 수 및 각각의 프로세서 코어에 따른 일반적인 기준 클럭으로 산출한 실행시간을 나타내었으며, ARM7은 평균적으로 50MHz, ARM9의 경우는 평균 200MHz, ARM10 프로

표 4. 코드 및 데이터 메모리의 크기
Table 4. Code and data memory size.

Program	Code+RO data (Bytes)	RW+ZI data (Bytes)
g.711	41,890	384
adpcm	55,930	300
susan	43,174	656
fft	31,636	528
Total	172,630	1,868

표 5. ARM9TDMI 코어에서의 실험결과
Table 5. Result of experiment on ARM9TDMI.

	Parameter	Instruction	Cycles	Time (ms)
g.711	to pcm	1,291,495	2,029,959	10.1
adpcm	adpcm	4,475,997	6,907,561	34.5
susan	corner	1,088,475	1,767,519	8.8
	edge	2,303,550	4,051,081	20.3
	smothing	19,138,855	36,341,315	181.7
fft	2 1024	14,100,957	20,882,248	104.4
Total		42,399,329	71,979,683	359.9

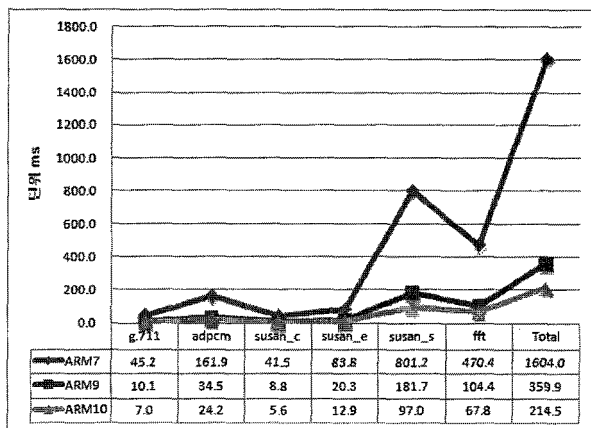


그림 10. 프로세서 코어별 동작 사이클

Fig. 10. Total cycles of ARM cores.

세서 코어는 평균 300MHz의 클럭속도를 가진다.^[14]

위의 결과에서 알 수 있는 것처럼 영상 및 음향데이터를 처리하여 외부상황을 인지하는 차량용 임베디드 시스템에서는 운용 및 사용자 인터페이스를 고려하여 1MBytes 이상의 코드 메모리와 200KBytes이상의 데이터 메모리가 필요할 것으로 판단되며, 차량 시스템 외부의 상황인지를 위하여 매 초당 2회 이상의 음성 및 영상 처리를 수행하는 시스템을 구현하려면, 200MHz의 기준 클럭을 갖는 ARM9 프로세서 코어 이상을 적용해야 함을 알 수 있다. 같은 방법으로 구현하려는 목적에 따라서 적용 가능한 알고리즘을 선정하고, 검증된 벤치마크 프로그램을 사용하여 필요한 명령어 수, 실행 사이클 수를 산출한 후, 해당 알고리즘이 적용되어질 빈도수를 적용하여 임베디드 시스템의 코드 메모리 및 데이터 메모리와 기준 클럭 주파수 및 프로세서 코어의 최소사양을 도출 할 수 있을 것으로 판단된다.

V. 결 론

본 논문은 특정 목적을 수행을 위한 임베디드 시스템을 개발하고자 할 때, 그 목적과 유사한 검증된 벤치마크 프로그램을 조합하여 성능측정을 수행한 후 실제 구현에 필요한 하드웨어 플랫폼을 제시하기 위한 연구이며, 실제로 전방영상과 음향데이터의 처리를 위한 차량용 임베디드 시스템의 예를 들어 그 결과를 보였다.

기존의 차량용 임베디드 시스템의 개발은 실제 영상 및 음향 처리를 위한 시스템 자원을 예측하는 것은 쉬운 작업이 아니었고, 따라서 특정 응용에 필요한 시스템 자원 보다 월등한 플랫폼을 적용하는 경우가 대부분

일 것으로 판단된다. 본 연구에서는 영상 및 음향 처리에 필요한 각각의 프로그램코드, 프로세서 속도 등 시스템 주요 자원에 대한 고찰을 통하여 실시간 차량용 시스템에 적용 가능한 플랫폼 사양을 도출하는 것을 보였으며, 이를 통해 검증된 소프트웨어 세트를 활용하여 특정한 응용을 목적으로 하는 임베디드 시스템 플랫폼을 효율적으로 선정하는 방법을 제시하였다.

본 연구에서는 3종류의 임베디드 벤치마크 수트를 사용 하였으나, 이를 확대하여 상용 소프트웨어인 EEMBC^[15], 이전버전의 제품 및 기존에 보유하고 있던 소프트웨어 모듈을 포함하여 성능측정을 수행한다면 더욱 정확한 예측이 가능 하리라 판단되며, 향후 더 많은 프로세서, 프로그램 및 데이터 메모리 등의 하드웨어 요소 뿐 아니라 개발환경, 기간 및 비용적인 측면까지 정량적인 요소로 분석할 수 있는 임베디드 시스템 플랫폼에 대한 최적화 연구도 필요할 것이다.

참 고 문 헌

- [1] Qing Li, Caroline Yao, "Real-Time Concepts for Embedded Systems", CMP Books, pp.48-51, 2003.
- [2] Arm Limited, "Application Note 32 The ARMuator", september 2003.
- [3] ARM Limited., "ARM DUI 0066D: ARM Developer Suite AXD and armsd Debuggers Guide", November 2001.
- [4] Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing 2nd Edition", Prentice Hall, pp.57-70, pp.149-166, pp.569-587, 2002.
- [5] NHK 방송기술연구소 화상연구부, "C언어에 의한 화상처리실무", 국제테크노정보연구소, 154-155쪽, 2000.
- [6] 박문수, 김재이 "교차로 교통사고 자동감지를 위한 사고음의 음향특성 분석", 한국산학기술학회논문지 7권 6호, 1143-1147쪽, 2006년
- [7] 이철주, "영상을 이용한 실시간 교통사고 감지 및 교통사고영상 자동저장 시스템", 석사학위논문, 아주대학교 ITS대학원, 10-14쪽, Feb, 2007.
- [8] C. Lee, mediabench webpage, "http://euler.slu.edu/~fritts/mediabench/mb1/."
- [9] C. Lee, M. Potkonjak, W. H. Mangione-Smith, "MediaBench: A Tool for Evaluating and Synthesizing Multimedia and Communication systems", In Proceedings of the 30th Annual International Symposium on Microarchitecture, pp.330-335, 1997.

[10] A. Arbor, mibench webpage,
 "http://www.eecs.umich.edu/mibench/"

[11] M. R. Guthaus, J. S. Ringenberg, D. Ernst, T. M. Austin, T. Mudge, R. B. Brown, "Mibench: A free, commercially representative embedded benchmark suite.", In IEEE 4th Annual Workshop on Workload Characterization, 2001.

[12] Arm Limited, "ARM DUI 0056D: ARM Developer Suite Developer Guide", November 2001.

[13] 박창규, 최요한, 한성덕, 이준희, 문호선, 김용득, 이정원, "자동차 사고재현을 위한 영상블랙박스 모델링", 한국정보과학회 가을 학술발표 논문집 제34 권 2호, 308-312쪽, 2007년

[14] 안효복, "ARM으로 배우는 임베디드 시스템", 한빛미디어, 2006년

[15] EEMBC, "http://www.ebenchmarks.com/"

저 자 소 개



문 호 선(평생회원)
 1998년 아주대학교 산업공학과
 학사 졸업.
 2008년 아주대학교 전자공학과
 박사 졸업.
 2008년~2010년 1월 부천대학
 컴퓨터제어과 전임강사
 2010년 1월~현재 향로표지기술협회 연구소 과장
 <주관심분야 : 임베디드시스템, RTOS, 통신>



김 용 득(평생회원)
 1971년 연세대학교 전자공학과
 학사 졸업.
 1973년 연세대학교 전자공학과
 석사 졸업.
 1978년 연세대학교 전자공학과
 박사 졸업.
 1979년~현재 아주대학교 전자공학부 교수
 1973년~1974년 불란서 E.S.E 전자공학 연구실
 1973년~1974년 미국 Stanford대학교 연구교수
 1981년~1982년 한국전자통신연구소 위촉연구위원
 1994년~1998년 ITS 연구기획단연구위원
 전자부문 총괄
 <주관심분야 : 통신, 컴퓨터, ITS>