

디지털 티브이-레디를 위한 태스크 기반의 내장형 소프트웨어

전승훈, 이종인
삼성전자 디지털미디어연구소

Task-Based Embedded Software for DTV-Ready

Seung-Hun Jeon, Jong In Lee
Digital Media R&D Center
Samsung Electronics Co., LTD

E-mail: shjeon01@samsung.com, chadlee@samsung.com

Abstract

본 논문에서는 기존의 DTV-Ready 에서 사용되고 있는 슈퍼 루프 형태의 내장형 소프트웨어 구조를 개선하여 태스크 기반의 내장형 소프트웨어 구조를 제안하였다. Gomaa 의 DARTS(Design Approach for Real-Time Systems) 방법을 사용하여 DTV-Ready 시스템을 분석하고 태스크 기반의 시스템으로 디자인하였다. 태스크간의 의존성을 최소화하여 내장형 소프트웨어의 재사용성을 높였으며, 실시간성을 고려한 디자인과 실시간 운영체제를 적용하여 내장형 시스템의 실시간 성능 향상을 제시하였다. 제안된 디자인은 직접 시스템에 적용하여 구현하고 테스트 결과를 통하여 기존 시스템의 성능과 비교 분석하였다.

I. 서론

내장형 시스템은 텔레비전, 셋탑박스(set-top-box), 디지털 카메라 등과 같은 가전제품에서 다양하게 적용되고 있다. 최근 CE(Consumer Electronics) 업계는 내장형 시스템에 대해 크게 세가지 문제점에 대해 관심이 크게 증가되고 있다. 첫째, 제품 각각의 내장형 소프트웨어가 갈수록 복잡해지고 있다. 20 세기 후반 32bit 마이크로프로세서의 가격이 5 달러 threshold 를 깨뜨려 내장형 시스템 시장이 폭발적으로 증대하였기 때문이다. 둘째, 다

양한 종류의 제품에 대한 시장 요구가 급증하고 있다. 내장형 시스템 시장에서는 대략 5 만여 개의 다른 디자인을 가진 제품들이 매년 나오고 있는 것으로 추정하고 있으며, 한 모델에 파생되는 제품들을 고려하면 그 종류는 셀 수 없이 다양하다. 마지막으로 복잡성은 증가하는데 반해 갈수록 내장형 시스템의 성능 향상 요구는 증가하고 있다.

이미 수년 전부터 이러한 문제 발생을 예상하고 이를 해결하기 위해 많은 연구들이 있었다. 대표적인 것으로 객체 지향의 패러다임이 바로 그것이다[1]. 내장형 시스템을 디자인할 때, 전통적인 방법은 슈퍼 루프(super loop) 시스템이다. 이러한 구조의 시스템은 디자인하는 과정이 비교적 용이하나, 실시간 특성을 이해하기가 매우 어렵게 되며, 입출력 폴링(polling)을 하는데 CPU(Central Processing Unit)의 자원을 낭비하게 된다. 또한 기능을 추가하거나 소프트웨어를 수정할 때 복잡성이 증가하게 된다[2].

Stepner 등은 RTOS(Real-Time Operating System)을 사용하여 내장형 시스템을 디자인 할 때 발생하는 issue 에 관해 정리를 하였다[3]. 그러나 그들은 시간적 제약 사항, 안전성, 디바이스 드라이버, 인터럽트 등 일반적인 RTOS 의 사용시 고려해야 될 점에 대해서만 나열하였다. Gomaa 는 그의 저서에서 실시간 시스템에 적용하기 적합한 구조적 방법론으로 DARTS 방법을 제시하였다[4]. 적용 예로 차량의 cruise control 시스템을 들었다.

본 논문에서는 CE 제품의 대표제품인 DTV-ready 시스템에 DARTS 방법을 적용한 내장형 소프트웨어의 디자인과 구현을 제시한다. 기존의 슈퍼 루프 형태의 제어 소프트웨어를 가진 DTV의 기능들을 태스크로 구분하고 RTOS를 사용할 수 있도록 재구성한다. 또한 내/외부 이벤트와 태스크간의 통신을 메시지로 처리하여 태스크간의 독립성을 최대화하여 시스템의 모듈화, 이해성, 재사용성, 그리고 확장성을 개선하고자 한다. 또한 태스크 기반의 시스템이 실시간 특성이 향상됨을 실험을 통해서 보여준다.

2 장에서는 기존 시스템에 대한 설명과 문제점을 정의하고, 3 장에서는 제안된 태스크 형태의 내장형 시스템에 대해 설명한다. 실험 환경 및 결과에 대해서 4 장에서 설명을 하고, 마지막으로 5 장에서 본 논문의 결론을 기술한다.

II. Super-Loop 제어 루틴

일반적으로 실시간 특성 개선의 척도로 신뢰성 (computational reliability)을 사용한다[5]. 여기서 신뢰성 $R^*(s, t, T)$ 이란 시스템을 시간 t 에서 태스크 T 를 기동하여 deadline 안에 성공적으로 상태 s 로 완료 가능할 확률을 의미한다. 보통 실패할 수 있는 확률은 포아송 (Poisson) 분포 함수로 모델링한다. 실시간 시스템의 최악의 반응 시간에 제약이 있으므로 실시간 시스템의 포아송 상수는 실시간 시스템이 아닌 경우보다는 값이 작다. 그래서 실시간 시스템의 신뢰성이 정의에 의해 실시간 시스템이 아닌 경우보다 좋다고 간주 할 수 있다.

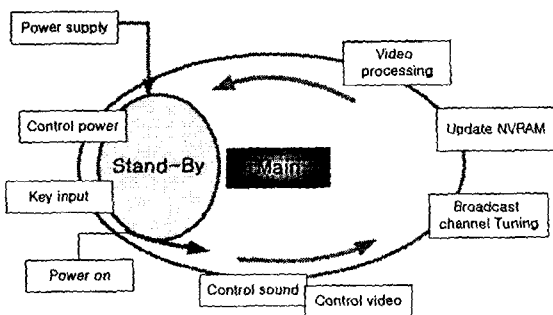


그림 1 슈퍼 루프 형태의 내장형 소프트웨어

그림 1은 기존 DTV-ready 시스템의 내장형 소프트웨어를 도식화하였다. 크게 스탠바이 모드와 메인 모드

로 나뉜다. 시스템에 전원이 연결된 경우 스탠바이 모드에 머물러 있으면서 각 디바이스를 초기화하고 전원을 제어하며 입력이 있는지 루프를 반복적으로 수행하면서 체크한다. 이때 리모컨이나 키패드(keypad) 입력으로 전원이 켜지게 되면 스탠바이 모드에서 빠져나오며 메인 루프에서 동작하게 된다. 보통 이와 같은 외부 입력은 인터럽트(external interrupt)로 처리하게 되어 있으며, 최근의 대부분의 내장형 시스템은 이러한 인터럽트를 기본적으로 지원하고 있다.

위와 같은 구조를 가진 내장형 소프트웨어는 개발 과정이 용이하고 간단하여 널리 사용되고 있다. 그러나 다양한 파생 모델이 있어 기능의 추가 및 삭제가 빈번히 일어나는 경우 실시간 동작 특성을 보장하기 위해 많은 노력이 필요로 한다. 또한 시스템 내부에서는 입출력을 폴링하기 위해 자원을 계속 낭비해야 하는 등 여러 가지 문제점이 존재한다. 여기서의 DTV-ready 시스템을 위한 내장형 소프트웨어의 문제점을 다음과 같이 정의한다.

Problem: 내장형 소프트웨어의 재사용성을 향상시키고 내장형 시스템의 실시간 성능을 향상시킨다.

가정은 다음과 같다. 첫째, 기존 시스템에서 이를 해결하기 위해서 슈퍼 루프 - 또는 다른 용어로 무한 폴링 루프(infinite polling loop) - 시스템 형태로 제어를 하고 있다. 둘째, 시스템의 자원에 제약 조건이 있어 UML(Unified Modeling Language)과 같은 복잡한 객체 지향 기술을 적용할 수 없다. 위와 같은 가정하에 기존의 DARTS 방법을 다음과 같이 7 단계로 적용할 수 있다. 1) 환경도(context diagram), 2) 데이터 흐름도(data flow diagram), 3) 태스크 구조도(task architecture diagram), 4) 소프트웨어 구조도(software architecture diagram), 5) 구조도(structure chart), 6) 태스크 경계도(task interface diagram), 7) 타이밍 분석(timing chart)으로 단계적으로 적용하여 시스템 소프트웨어를 디자인한다. 각 단계에서 디자인된 다이어그램이 서로 일관성이 유지되는지 확인함으로써 디자인에 오류가 있는지 검증할 수 있다.

III. 태스크 기반의 내장형 소프트웨어

본 절에서는 기존의 슈퍼 루프 형태의 내장형 소프트웨어를 태스크 기반으로 디자인하기 위한 단계를 기술한다. 먼저 환경도에 대해서 기술하고, 데이터 흐름도를 차례대로 설명한다. 또한 내장형 시스템의 성능

(performance)를 측정할 수 있는 객관적인 방법을 제시한다.

3.1 대상 시스템과 외부 인터페이스

그림 2는 DTV-ready 시스템을 위한 환경도이다. 대상 시스템과 외부 인터페이스를 자세히 보여주고 있어 대상 시스템의 기능을 추측할 수 있다. 본 시스템은 외부 데이터 입력으로 CVBS(Composite Video Band Signal), Y/C(Luminance/Chrominance), YPbPr(), Digital RGB 가 있으며 외부 제어 신호 입력으로는 리모컨과 키패드로부터의 키코드가 있다. 패널로 나가는 외부 입력과 디버깅을 위한 RS232, JTAG 포트(port)가 있다.

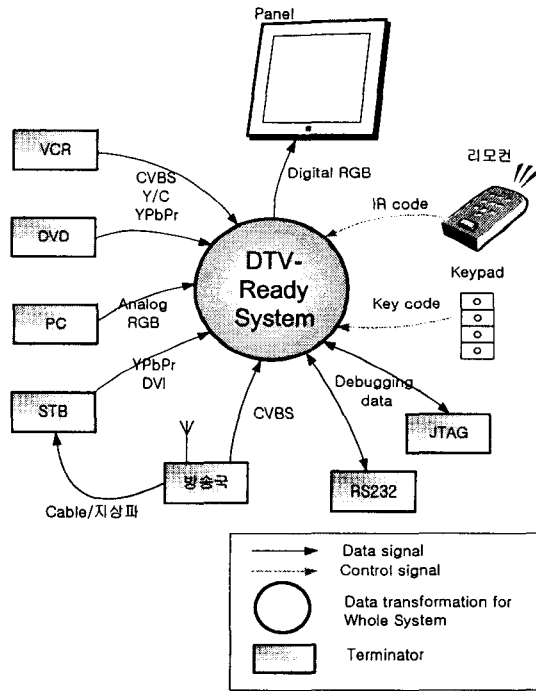


그림 2 DTV-ready 시스템의 환경도

그림 3은 데이터 흐름도를 나타낸다. 외부 제어 입력은 I/O 객체에서 처리를 한다. 일반적으로 외부 I/O는 가장 높은 우선 순위(priority)를 갖는다. I/O 객체에서 키코드를 분해한 뒤 이벤트에 의해 사용자 인터페이스 객체를 깨운다. 이벤트는 RTOS에서 제공하는 시스템 콜로써 태스크 전환을 플링 방식에 비해 빠르게 한다. 입력된 키코드는 대부분 입력 데이터 처리에 대한 제어 명령을 가지고 있어, 사용자 인터페이스 객체에서는 즉

시 필요한 기능을 수행하라는 명령을 source 객체에 이벤트와 함께 전달한다. 모든 입력 데이터는 source 객체에서 처리한다. 태스크를 디자인할 때 기능과 연관이 있도록 분할하며 그러므로 source 객체는 여러 개의 태스크로 나뉜다. 디바이스 드라이버와 RTOS 객체는 상위 객체에 공통으로 사용된다.

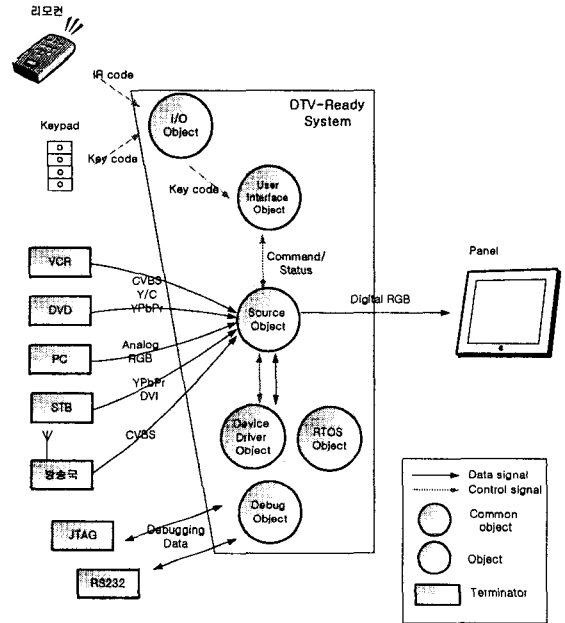


그림 3 DTV-ready 시스템의 데이터 흐름도

Source 객체는 다섯 개의 태스크로 세분화되며 각 태스크는 실행단위가 된다. 각 태스크들은 주기적으로 발생하는 태스크(periodic task)와 비주기적으로 발생하는 태스크(aperiodic task)로 크게 나뉘며 주기는 서로 다른 multi-rate 시스템이다.

3.2 Performance Measurement

N 개의 독립된 태스크가 하나의 프로세서에서 동작하고 있는 시스템을 고려하고 있다. i 번째 task T_i 의 비주기적 태스크의 parameter는 deadline d_i , 실행시간 c_i 가 있다. $T_{i,j}$ 는 T_i 의 j 번째 request 이고, $T_{i,j}$ 의 실행시간은 $c_{i,j}$ 이다. 그림 4와 같이 $c_{i,j}$ 는 상수가 아닌 확률 변수로 가정한다. 본 시스템의 성능 측정의 척도(index)는 $E[c_{i,j}]$ 로 확률 변수 $c_{i,j}$ 의 기대치(expectation)이다. 평균 점유도(utilization)은 $E[c_{i,j}]$ 를 주기로 나누는 것이나 본 시스템은 비주기성의 태스크들이 중요하므로 점유도 보다는

timing chart 가 더 중요하다. 실험 단계에서는 편의상 $E\left[\sum_i^M c_{i,j}\right]$ 를 측정하여 성능 비교의 척도로 사용하였다. 여기서 M 은 DTV-ready 의 한 기능을 수행하기 위해 수행되는 태스크의 개수이다.

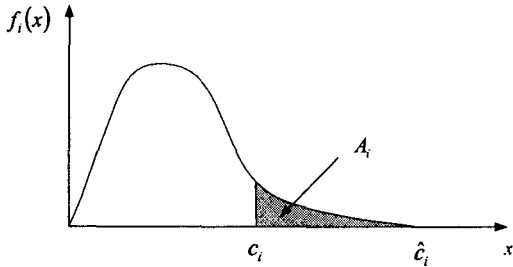


그림 4 태스크 T_i 의 확률 분포 함수.

IV. 실험 결과 및 성능 향상

본 논문에서 사용한 실험 환경은 다음과 같다. 대상 시스템의 CPU 는 16bit RISC(Reduced Instruction Set Computer) 마이크로 프로세서인 C167 을 사용하였으며, 상용 RTOS 를 적용하였다. 스케줄링 (scheduling) 방법은 고정 우선 순위(fixed priority)를 사용하였고, 같은 우선 순위를 가진 태스크는 round robin 과 FIFO(First In First Out)을 적용하였다. 개발 환경은 Pentium IV 1.5Ghz IBM PC 를 사용하였다. 대부분 C 언어로 프로그래밍하고, JTAG 에뮬레이터로 바이너리 코드를 RAM 에 다운로드하여 디버깅하였다. 그림 5은 실험 결과의 일부로써 입력 전환 제어 신호에 대해 DTV-ready 시스템의 반응 평균 시간을 의미한다.

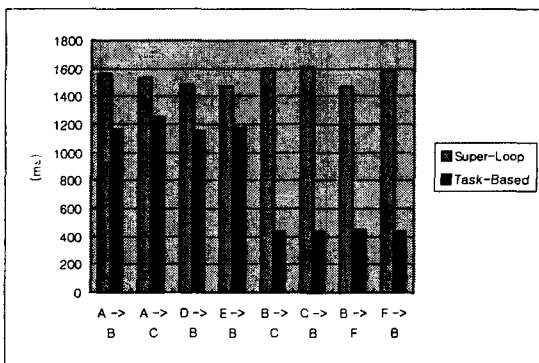


그림 5 입력 source 전환의 평균 실행 시간. A: Analog RGB, B: Radio frequency, C: DVD, D: DTV, E:

DVI, F: CVBS.

그림 5은 입력 비디오 source 를 전환시키고자 하는 event 가 발생했을 때 수행되는 태스크들의 $E\left[\sum_i^M c_{i,j}\right]$ 를 측정된 결과이다. 기존의 슈퍼 루프 시스템보다 태스크 기반의 시스템의 성능이 30% 이상 좋아지는 것을 볼 수 있다. 특히 슈퍼 루프 시스템에서 불필요하게 반복적으로 수행하는 부분을 태스크 기반에서는 분리해서 수행하지 않도록 할 수 있는 부분이 있었다.

V. 결론

본 논문에서는 DTV-ready 시스템에 적합한 태스크 기반의 내장형 소프트웨어를 제시하였다. 첫째, DARTS 방법론을 DTV-ready 시스템에 적용하였으며, 둘째, 적합한 성능 척도를 제시하였다. 실험 결과 제안된 소프트웨어를 적용함으로써 기존 시스템에 비해 성능 향상을 입증하였다. 또한 명확한 디자인 방법으로 소프트웨어의 재사용 측면에서도 적합하다는 것을 예상할 수 있다. 추후 이를 검증하기 위해 재사용성의 수치화가 필요하다. 또한 더욱 복잡한 기능을 가진 시스템에 대해서 동적 스케줄링을 적용하는 것이 구현되어야 할 것이다.

참고문헌

- [1] P. Pesonen, V. Seppanen, "Object-based design of embedded software using real-time operating systems", Sixth Euromicro Workshop on Real-Time Systems, pp. 194-198, 1994.
- [2] J. H. Labrosse, "Designing with real-time kernels", Conf. on Embedded Systems, no. 204, 2001.
- [3] D. Stepner, N. Rajan, D. Hui, "Embedded application design using a real-time OS", Conf. on Design Automation, pp. 151-156, 1999.
- [4] Hassan Gomaa, Software design methods for concurrent and real-time systems, Addison-Wesley, 1993.
- [5] C. M. Krishna, K. G. Shin, Real-Time Systems, McGraw-Hill, 1997