

## 인공위성 시스템소프트웨어 모니터링 도구의 설계

조병태<sup>○</sup> 김선욱 박형식 김형신  
충남대학교 컴퓨터공학과

최종욱 채동석 이종인  
한국 항공우주연구원

{btcho<sup>○</sup>, sunwook, hs\_park, hyungshin}@cnu.ac.kr, {jwchoi, dschae, jilee}@kari.re.kr

### Design of a monitoring tool for spacecraft system software

Byungtae Cho<sup>○</sup>, Sunwook Kim, Hyungsic Park, Hyungshin Kim  
Department of Computer Science and Engineering  
Chungnam National University

Jongwook Choi, Dongsock Chae, Jongin Lee.  
Korea Aerospace Research Institute

#### 요 약

인공위성의 탑재 컴퓨터는 일종의 내장형 시스템으로서, 인공위성 전체의 동작을 제어하는 핵심 부품이다. 이러한 인공위성 탑재 컴퓨터에 이용되는 시스템 소프트웨어는 실시간 고신뢰성을 요구하고 있으며, 따라서, 위성 컴퓨터에 탑재되는 시스템소프트웨어 및 응용소프트웨어의 동작성능 분석은 매우 중요한 문제로 인식되고 있다. 본 연구에서는 차세대 저궤도 인공위성의 ERC32 기반 탑재 컴퓨터용 시스템 소프트웨어의 동작상태를 분석할 수 있는 모니터링 도구를 설계하였다. 이를 위하여 인공위성 시스템 소프트웨어 모니터링 도구의 요구조건을 분석하였다. 제안한 도구는 상위수준 및 하위수준에서의 모니터링이 동시에 가능하며, 시스템 소프트웨어의 실시간성 및 신뢰성을 선행적(proactive) 방법으로 감시할 수 있는 장점이 있다.

#### 1. 서 론

인공위성의 탑재 컴퓨터는 위성 전체의 제어를 담당하는 핵심 서브시스템이다. 이러한 인공위성의 제어를 위한 컴퓨터는 초기에는 하드웨어를 이용한 고정된 임무를 수행하는 순차로직 기반의 제어회로가 이용되었으나, 최근에는 다양한 마이크로프로세서를 이용하는 추세로 발전해 가면서, 보다 복잡한 업무를 수행할 수 있게 되었다. 최근의 위성 컴퓨터는 위성의 관리 및 제어, 그리고 탑재물(payload)들의 운용에 이르기 까지 다양한 업무를 자율적으로 수행하고 있으며, 탑재 컴퓨터의 시스템 소프트웨어가 그 핵심역할을 담당하고 있다.

위성 탑재 컴퓨터 시스템 소프트웨어는 일종의 내장형 소프트웨어로 실시간성과 고 신뢰성을 그 특징으로 갖는다. 본 연구에서는 차세대 저궤도 지구관측위성 탑재 컴퓨터의 시스템소프트웨어 개발에 이용될 수 있는 시스템 소프트웨어 모니터링 도구의 설계하였다. 차세대 저궤도 위성 탑재 컴퓨터의 시스템 소프트웨어는 부트로더, 운영체제, 통신프로토콜, 디바이스 드라이버 등으로 구성되어 있다. 본 연구에서는 이 중에서 특히 실시간 운영체제인 RTEMS(Real-Time Operating System for Multiprocessor Systems)[2] 기반 시스템소프트웨어들의 동작을 동적으로 감시할 수 있는 모니터링 도구를 설

계하였다.

인공위성 탑재 컴퓨터의 내장형 운영체제로는 VxWorks 나 VRTX등의 실시간 커널 등이 이용되어 왔으나, 최근 유럽우주기구(European Space Agency)와 미 항공우주국(NASA)에서는 공개소프트웨어 기반 운영체제를 차세대 인공위성 탑재 컴퓨터의 운영체제로 활용하는 방안을 모색하고 있다[3]. 특히 유럽에서는 RISC(Reduced Instruction Set Computer) 구조의 Sun SPARC[1] 프로세서 코어를 채택한 우주용 프로세서들을 이용하여 탑재 컴퓨터를 개발하고, 이를 위한 운영체제로 RTEMS의 탑재를 지원하고 있다.

차세대 저궤도 위성의 탑재 컴퓨터는 이러한 세계적인 추세를 대비하여 Sun SPARC 프로세서 코어 기반의 ERC32 프로세서를 이용하고 있으며, 운영체제로 RTEMS를 이용할 계획이다. 그러나, RTEMS 및 추가적인 시스템 소프트웨어의 개발 시 요구되는 디버깅 도구는 기본적인 GNU GDB 기반 저 차원 디버깅 도구와 일반적인 태스크 수준 API 만을 제공하고 있어서, 지원되는 기능이 매우 제한적이라는 문제점이 있다. 또한, 실시간성과 고 신뢰성을 요구하는 탑재 컴퓨터 시스템 소프트웨어의 특성을 고려할 때, 이 둘 특성을 검증할 수 있는 도구는 없는 실정이다.

따라서, 본 연구에서는 이러한 공개소프트웨어인 RTEMS 기반 시스템 소프트웨어 개발 시에 사용할 수 있는 ERC32 프로세서용 디버깅 및 모니터링 도구를 설계하였다. 설계한 도구는 탑재컴퓨터의 저차원 하드웨어 상태 및 상위수준의 실시간 동작성, 자원이용 현황 등의 정보를 개발자에게 제공할 수 있도록 하였다.

본 논문의 구조는 다음과 같다. 2절에서는 개발환경인 차세대 저궤도 위성 탑재컴퓨터에 대해 소개하고, 3절에서는 설계한 모니터링 도구의 요구조건을 기술하였다. 4절에서는 모니터 설계결과를 설명하였으며, 5절에서 구현현황을 소개하고, 6절에서 결론을 맺는다.

2. 차세대 저궤도 위성 탑재 컴퓨터

차세대 저궤도 위성은 과학 탐사, 지구 원격탐사, 이동통신을 목적으로 발사되는 위성으로, 다양한 임무 요구사항을 만족하기 위하여 높은 처리 능력과 저전력 소모 및 경량성과 안정성을 필요로 한다.

2.1 하드웨어 구조

위성 운용 환경 처리용 컴퓨터 모듈은 32비트 SPARC 임베디드 프로세서인 TSC695F CPU(ERC32)를 기반으로 메모리, 스페이스와이어 제어기와 같은 각종 외부 장치 인터페이스를 제공한다. 그림 1은 위성 운용환경 처리용 컴퓨터 모듈의 블록 다이어그램이다.

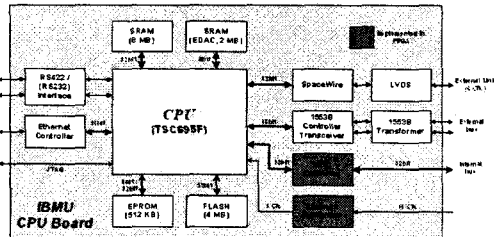


그림 1. 위성 운용환경 처리용 컴퓨터 모듈

프로세서는 고성능과 안정성, 신뢰성이 중요한 임베디드 항공 우주 응용프로그램 환경에 적합하도록 ESA(European Space Agency)사에서 제공되는 32비트 SPARC v7 아키텍처 기반의 TSC695F CPU를 채택하였다. 이는 다양한 장치들을 하나의 보드에 집적하여 공간 및 비용절감을 가능하게 하고, 적은 전력 소모와 경량성을 만족시킬 수 있다[4].

부트코드와 EDAC, 시스템 소프트웨어 및 응용프로그램을 저장하기 위한 플래시 메모리와 프로그램과 데이터를 위한 SRAM이 포함되어 있으며, 각종 디버깅과, 시스템 버스를 위한 외부 인터페이스들이 존재한다.

스페이스와이어 인터페이스는 인공위성의 하드웨어 모듈 간의 쉬운 연결과 시험이 가능하도록 하여 개발 비용과 시간을 단축 시킴으로써 인공위성의 신뢰성과 개발의 편의성을 향상시킬 수 있으며, 직렬 통신 포트를 추가하는 것만으로도 전송 속도를 향상시킬 수 있는 가변확장성도 함께 제공함으로써 인공위성의 전체적인 성능을 향상시킬 수 있다[5].

2.2 시스템 소프트웨어

현재 인공위성 시스템에서 사용되는 실시간 운영체제로서는 대표적으로 Windriver사의 VxWorks와 OAR사의 RTEMS가 있다[2].

RTEMS는 OAR에서 미 육군의 미사일 개발의 일환으로 개발된 실시간 운영체제로서 높은 신뢰성을 강점으로 유럽 ESA에서 인공위성용 실시간 운영체제로 채택하여 사용하고 있으며, NASA에서도 이와 유사한 연구가 진행 중이다. RTEMS는 GNU Tool chain을 기본으로 개발환경을 구축할 수 있으며 리눅스, 유닉스 및 윈도우와 같은 다양한 플랫폼에서 개발이 가능하다[6]. 또한, 오픈 소스 정책을 취하면서도 기능면에서 기존의 상용 운영체제에 뒤지지 않는 높은 신뢰성과 고성능을 제공함으로써 저비용, 고성능의 시스템을 구축할 수 있다. 또한, 위성 운용 환경 처리용 컴퓨터 모듈에 장착되어 있는 특정 장치들을 손쉽게 지원할 수 있는 장점을 지닌다.

RTEMS의 시스템 구조는 그림 2와 같이 타겟 하드웨어를 지원하기 위한 디바이스 드라이버와 마이크로 커널, 개발 환경을 지원하는 기본 응용프로그램 요소와 응용 소프트웨어로 구성되어 있다. 대부분의 상용 실시간 운영체제의 구성요소와 동일한 구조를 제공함으로써 특정 시스템 모듈에 맞는 BSP를 개발하는데 편의성을 제공하며, RTEMS의 다양한 서비스들은 높은 신뢰성과 정확한 실시간 처리를 요구하는 위성 운용 처리에 있어서 매우 중요한 요소가 된다[6].

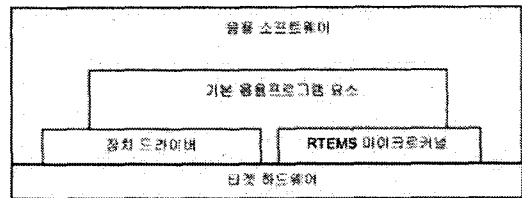


그림 2. RTEMS Application Architecture

3. 시스템 소프트웨어 모니터링 도구 설계 요구사항

본 연구에서 사용되는 ERC32SC 타겟 시스템은 인공위성용이라는 사용 환경 특성으로 인해, 다른 시스템에 비해 높은 자원 제약성을 가지고 있으며 따라서 고 신뢰성과 저전력소비, 경량성, 실시간성 등의 특성을 지니고 있다. 따라서 실시간 임베디드 운영체제의 관점에서 앞서 기술한 인공위성 시스템의 특성을 감안하여 모니터링 도구를 개발 시에는 다음을 고려해야 한다[7][8].

- 이벤트 발생에 따른 디버그 및 모니터링 태스크의 동적 로딩(Dynamic loading)
- SpaceWire 디바이스 인터페이스를 이용한 확장성(Extensibility)
- 하드웨어 의존성을 명확히 구분하여 소프트웨어의 높은 이식성(Portability) 보장

- 요청에 대해 빠른 인터럽트 지연(Interrupt Latency) 처리로 주요 성능의 실시간성(Real-time) 보장
- 타겟 하드웨어에 동작에 영향을 주지 않으며 실시간 성능분석 수행의 독립성(Independency) 보장
- 안전성(Safety)과 신뢰성(Reliability) 보장

이런 특성을 고려하여 연구중인 모니터링 시스템은 ERC32SC 타겟시스템의 외부 버스로 SpaceWire 디바이스 인터페이스를 이용하며, 외부로부터의 디버깅이나 모니터링 요청이 있기 전까지는 블록(Blocked) 상태로 스케줄링을 받지 않게 하여 인공위성의 자원부하를 줄이며, 요청 발생시에는 빠른 인터럽트 지연(Interrupt Latency) 처리를 통해 실행(Executing) 상태로 전이하여, 신속히 요청을 수행해야 한다. 이 때에도 디버깅 또는 모니터링 정보에 대한 처리로 인해 전체 시스템의 성능에 영향을 주지 않아야 한다. 또한 오류 발생시 시스템 전체 실패가 되지 않도록 보장해야 한다.

또한 디버깅 및 모니터링 도구는 실시간 시스템에서 이용되는 다른 시스템 모니터링 도구에서 제공되는 다음의 정보를 포함하여 제공하도록 설계되어야 한다.

- 레지스터 정보
- 프로세스, 쓰레드 모니터링(태스크 상태, 우선순위, 스택 메모리, 엔트리 포인트 등)
- 시스템 자원 사용 현황(CPU 사용량, 메모리, 메시지 큐 등)
- 디바이스 드라이버 정보
- GDB 수준의 디버깅 기능 확장
- 디버깅 및 모니터링 Agent 태스크
- 시스템 이벤트 정보(치명적인 에러 발생시 이를 확인하는 기능 등)

디버깅 및 모니터링 도구가 이와 같은 정보를 제공하기 위해 RTEMS 시스템 내부 정보 테이블을 접근하는 에이전트 태스크가 필요하며, 이를 이용하여 프로세스, 쓰레드 모니터링과 자원 사용 상황 등에 대한 정보를 제공하도록 구성하여야 한다.

#### 4. RTEMS 기반 인공위성 시스템 소프트웨어 모니터링 도구 설계

##### 4.1 모니터링 시스템 전체 구조

RTEMS 기반의 인공위성 시스템 소프트웨어의 모니터링 도구를 개발하기 위해 제안하는 시스템은 그림 3과 같다. 본 연구에서 사용된 ERC32SC 타겟시스템 환경에서의 전체 모니터링 시스템은 SpaceWire 드라이버 인터페이스, 원격 디버깅 지원을 위한 RTEMS 스타브, 시스템 모니터링을 위한 성능 및 태스크 모니터링 오브젝트, SpaceWire와 어플리케이션 사이의 데이터 전송을 위한 전송계층 드라이버 인터페이스로 구성되어 있다.

SpaceWire 드라이버 인터페이스로 전달된 메시지는 수신 필터 모듈에 등록된 필터에 맞는 오브젝트로 전달된

다. 하위수준의 디버깅 및 모니터 메시지는 원격 디버깅 지원을 위한 RTEMS 스타브로 전달된다. RTEMS 스타브는 ERC32SC 타겟시스템상에서 데몬 태스크 형태로 동작하며 지상 호스트 시스템의 GDB 서버로부터 전달된 디버깅 명령과 하위수준 모니터 명령을 처리하여 타겟시스템에서 디버깅과 모니터링을 수행한다. 상위수준의 모니터 메시지는 요청처리 오브젝트로 전달되어 성능 및 태스크 모니터링 오브젝트를 통해 시스템 내부의 데몬 태스크 형태로 동작중인 내부 모니터링 태스크로부터 모니터링 정보를 수집하고, 수집된 결과는 전송계층 드라이버 인터페이스를 통해 외부버신인 SpaceWire를 거쳐 지상으로 재전송된다.

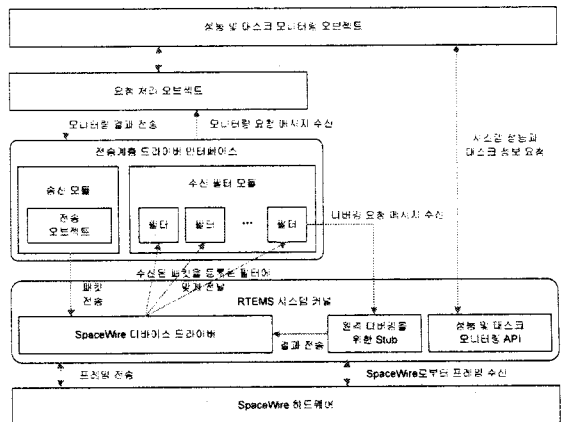


그림 3. 시스템 소프트웨어 모니터링 도구 전체 구조

##### 4.2 상위수준 모니터링 시스템

상위수준 모니터링 시스템은 RTEMS 기반 인공위성 시스템 소프트웨어의 실시간성과 자원사용량 등의 정보를 확인하여 자원제약적이며 접근제약적인 인공위성 시스템의 원격 모니터링을 위해 반드시 요구되는 기능이며, 이를 지원하기 위한 상위수준 모니터 시스템을 그림 4와 같이 제안하고 있다.

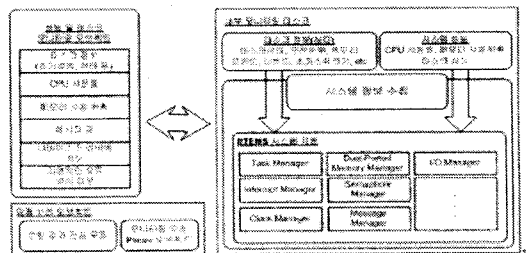


그림 4. RTEMS 상위수준 모니터 시스템

상위수준 모니터 시스템은 지상에 위치한 호스트 시스템에서 요청한 모니터 요청 메시지의 종류를 분석하거나, 요청 받은 결과를 전송계층 드라이버 인터페이스로 전달하는 요청처리 오브젝트와 요청 메시지의 종류에 따라 실제 정보를 요청하는 성능 및 태스크 모니터링 오브젝트, 그리고 실제로 RTEMS 시스템 내부의 태스크 관리 테이블 등에서 정보를 얻어오는 내부 모니터링 태스크로 구성되어 있다.

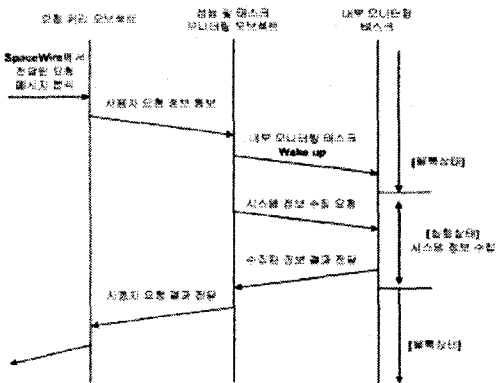


그림 5. RTEMS 상위수준 모니터링 시스템의 처리

상위수준 모니터 시스템의 처리과정은 그림 5와 같다. 그림 3에 나타난 수신필터모듈을 통해 SpaceWire 인터페이스로 전달된 사용자의 요청 메시지는 요청처리 오브젝트에서 요청 명령의 종류를 결정하게 되며, 해당 명령에 맞는 정보를 성능 및 태스크 모니터링 오브젝트로 요청한다. 요청을 받은 성능 및 태스크 모니터링 오브젝트는 요청 이전까지 블록 상태인 내부 모니터링 태스크를 깨운 후, 해당 정보를 수집하도록 요청하며, 블록 상태에서 실행 상태로 전이된 내부 모니터링 태스크는 전체 태스크 정보를 관리하는 \_Object\_Information\_table 객체 등에서 요청한 정보에 해당하는 정보를 성능 및 태스크 모니터링 오브젝트로 전달한다. 사용자가 요청한 시스템 정보를 수집이 완료되면 요청처리 오브젝트의 수행결과 전송 모듈을 통해 SpaceWire를 통해 시스템 정보를 요청한 사용자에게 전달하게 된다.

#### 4.3 하위수준 모니터

하위수준 모니터는 인공위성의 안정성과 신뢰성 향상을 위해 스택, 메모리, 레지스터 등의 상태를 모니터링 함으로써 오류를 감지하고 그 원인을 분석하여 해결하기 위해 반드시 요구되는 도구이다[9]. 본 연구는 공개소스 기반의 안정화된 원격 디버깅 기능과 하위수준의 모니터링 기능을 제공하는 GDB를 기반으로 한 하위수준 모니터링 도구를 구현하고자 한다.

본 연구에서 사용되는 RTEMS는 타겟 시스템의 GDB 기반의 원격디버깅을 위한 GDB스터브를 제공한다. RTEMS의 GDB스터브는 타겟 시스템 상에서 데몬 태스크 형태로 존재하며 호스트 시스템의 GDB서버로부터 전달

된 GDB 명령을 수행함으로써 타겟 시스템의 디버깅과 모니터링 가능하게 한다[10].

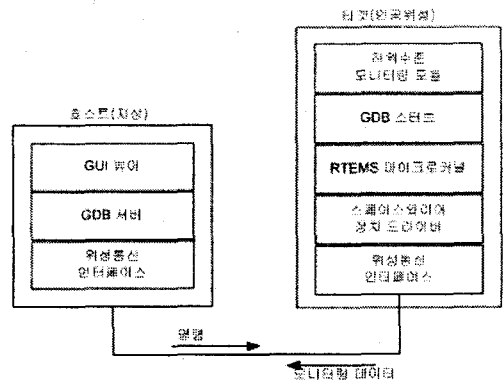


그림 6. 하위수준 모니터의 설계

현재 OAR사에서 제공되는 RTEMS-4.6.6버전은 m68k, PowerPC, i386 플랫폼을 위한 TCP/IP 프로토콜 기반의 Ethernet과 Serial인터페이스 기반의 GDB스터브를 제공함으로써 원격 타겟의 디버깅 기능을 제공한다[11][12]. 우리는 이러한 RTEMS의 GDB스터브를 우리의 타겟환경 수정/보완하여 인공위성 시스템에 적합한 ERC32프로세서 기반의 스페이스와이어 인터페이스를 통한 원격 타겟 모니터링 기능을 구현하고자 한다.

본 연구의 하위수준 모니터링 시스템은 그림 7에 표현한 바와 같이 지상에 위치한 호스트 시스템의 GDB서버를 통해 내려진 명령은 위성 통신 인터페이스를 통하여 인공위성으로 전달되게 되어 위성으로 전송된 명령은 스페이스와이어 인터페이스를 통해 RTEMS의 GDB스터브로 전달되어 해당 명령을 수행한 후 그 결과를 다시 통신장치를 통해 지상의 GDB 서버로 전달하는 방식으로 운용되게 된다.

원격에 존재하는 타겟의 하위수준 모니터링 과정은 준비, 수행, 종료의 3단계로 구분할 수 있다. 첫 번째 단계인 준비 단계는 연결과 프로세스접합 절차로 구성된다. 연결단계는 지상 호스트 내의 GDB서버와 위성에 존재하는 GDB스터브 간 연결을 성립시키는 절차로 연결이 완료된 후 GDB스터브는 프로세스접합 절차에 진입하게 된다. 프로세스접합 절차는 멀티 프로세스를 지원하지 않는 RTEMS의 특징에 따라 실행중인 프로세스의 종속 쓰레드 형태로 포함시킴으로써, GDB스터브와 기존의 태스크를 멀티태스크 방식으로 동시에 실행되도록 하는 절차로 프로세스접합 과정이 종료된 후 타겟의 시스템은 멀티쓰레드의 단일 프로세스 형태로 수행단계에 진입함으로써 위성의 GDB스터브는 지상의 GDB서버와의 교신이 가능하게 된다.

프로세스접합 과정이 완료된 후 지상의 GDB서버의 명령에 따라 태스크의 정지 이벤트를 발생시킴으로써 하위수준 모니터링 모듈을 이용한 실시간 하위수준의 자원의 모니터링을 가능하게 한다.

5. 상위수준 모니터링 시스템 실행

현재 개발된 상위수준 모니터 시스템은 ERC32SC 타겟 시스템에서 모니터링 태스크 형태로 동작하며, 사용자의 요청에 따라 정보를 수집하여 결과를 전달한다. 그림 7, 8, 9에서는 실제 타겟 시스템에서 동작중인 태스크들에 대한 정보를 표현하고 있다.

```
rtems $ cpusage
Total Units = 11084
CPU Usage by thread
ID      NAME      TICKS
0x09010001  IDLE      11538
0x0a010002  TA1       00027
0x0a010003  TA2       00012
0x0a010004  TA3       00009
```

그림 7. CPU 사용률

그림 7의 CPU 사용률은 시스템 시작 이후 RTEMS 동작 동안의 Ticks와 각 태스크 별 Ticks를 나타내고 있다.

```
rtems $ task
ID      NAME      PRIO  STAT  MODES  EVENTS
-----
0a010002  TA1       1     DELAY P:T:nA NONE
0a010003  TA2       1     DELAY P:T:nA NONE
0a010003  TA3       1     DELAY P:T:nA NONE
0a010004  RMON      1     READY P:T:nA NONE
```

그림 8. 태스크 정보

그림 8은 현재 수행(Executing 상태)중인 혹은 대기(Blocked 상태)중인 태스크들의 이름, ID, 상태, 우선순위, 선점가능성, 이벤트 등에 대한 정보를 나타내고 있다.

```
rtems $ driver
Major   Entry points
-----
0       init: [0x2001868]: control: [0x2001980]
        open: [0x20018d8]: close: [0x2001944]
        read: [0x2001958]: write: [0x200196c]
1       init: [0x2001794]: control: [0x20017bc]
        open: [0x0]: close: [0x0]
        read: [0x0]: write: [0x0]
```

그림 9. 드라이버 엔트리 정보

그림 9은 RTEMS 시스템에서 동작중인 디바이스 드라이버 엔트리에 대한 정보를 나타내고 있다.

6. 결론

본 연구에서는 차세대 저궤도 위성 탑재컴퓨터의 시스템소프트웨어 개발에 이용될 수 있는 RTEMS기반 소프트웨어 모니터링 도구의 설계결과를 소개하였다. 이를 위하여 시스템소프트웨어의 동작상태를 하드웨어기반 저 차원과 응용태스크들의 상태를 상위수준에서 동시에 모니터링 할 수 있도록 도구를 설계하였다. 제안하는 도구를 이용하여 공개소프트웨어인 RTEMS 실시간 운영체제 개

발도구에서 제공하지 못하는 원격디버깅 기능이 제공되며, 운영체제의 자원이용상황, 스케줄러 및 인터럽트 등의 동작특성을 분석하게 되어 향후 탑재컴퓨터 시스템 소프트웨어 개발 시 실시간성 및 신뢰성 검증에 매우 유용하게 이용될 수 있을 것으로 기대된다.

7. 참고문헌

- [1] 김대영 외, " 최근의 위성용 탑재컴퓨터 개발 현황" 항공우주산업기술동향 2권 1호, 2004. pp76~85
- [2] OAR Corporation. RTEMS(Real-Time Operating System for Multiprocessor Systems)
- [3] 채동석 외, "인공위성 탑재소프트웨어 개발동향", 항공우주산업기술동향 3권 1호, 2005. pp.43~49
- [4] ATMEL Corporation. TSC695F SPARC 32-bit Space Processor User Manual. 2003
- [5] Rastetter P. et.al., Simple Interprocessor Communication Protocol Specification, DIPSAPII-DAS-31-01, Issue 3, 08.10.96
- [6] OAR Corporation. RTEMS User Manual. 2006
- [7] 김대영 외, " 차세대 위성용 탑재 컴퓨터 개발 동향에 관한 연구" . 한국항공우주학회 30권 2호, 2002. pp144~151
- [8] Y.Eustache, J-Ph.Diguet, M.El Khodary, " RTOS extensions for dynamic hardware / software monitoring and configuration management" , Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International, pp1-4, 2006
- [9] Tankut Akgul, T., Kuacharoen, P., Mooney, V., and Madisetti, V., " A debugger RTOS for embedded systems" , in Proceedings of the 27th EUROMICRO Conference, pp. 264-269, Sep. 2001
- [10] Till Straumann, " GDB Agent for RTEMS" , 2006.
- [11] OAR Corporation, " RTEMS Remote Debugger Server Specifications" , 2003
- [12] OAR corporation, " RTEMS/GDB User' Guide" , 2003