마이크로 컨트롤러를 위한 iRTOS 포팅 연구

A Study on iRTOS Porting for Microcontroller

이 상 길*. 고 재 용*. 조 경 연*. 이 철 훈* 충남대학교 컴퓨터공학과*

Lee Sang-gil. Ko Jae-yong. Jo Kyung-yeon. Lee Cheol-Hoon Chungnam Natl. Univ.

최근 다양한 분야에서 IoT(Internet of Things)를 위한 마이크로 컨트롤러 디바이스가 사용되고 있다. BLE 통신을 통한 전력소모 감소로 인해 작은 크기에서 다양한 기능을 수행할 수 있게 되었으며, 이를 위해 마이크로 컨트롤러에서 RTOS가 사용되는 분야 가 늘어나고 있다. 본 연구에서는 국내 기술로 개발된 iRTOS를 IoT 디바이스에 포팅하는 연구를 수행한다. Cortex-M 시리즈를 위해 내부 컨텍스트 관리 및 RTOS 타이머 관련 내용을 수정하여 포팅하였다.

I. 서론

현재 다양한 IoT 기반의 임베디드 장치가 개발되어 출 시되고 있으며, 글로벌 시장 조사 기관은 Gartner은 2020 년에는 무려 200억 개에 달하는 임베디드 장치들이 사용 될 것으로 예상하고 있다. IoT 기기의 보급은 더욱 가속 화되고 있으며, 이것이 스마트 생산, 물류, 서비스 핵심 요소로 자리 잡을수록 이와 관련된 소프트웨어 기술 개 발에 대한 수요가 증가할 것이다.

임베디드 장치에서는 필요한 최소한의 기능만을 수행 하기 위해 RTOS(Real-Time Operating System)을 사용한 다. RTOS는 범용 운영체제로 알려진 윈도우나 리눅스와 같은 다양한 플랫폼에서 많은 프로그램을 지원하는 운영 체제가 아닌, 한정된 플랫폼에서 필요한 기능을 간소화 시켜서 수행할 수 있도록 하는 경량화 된 운영체제라고 볼 수 있으며, 기능 수행에 있어서 시간 결정성의 보장으 로 인해 군사기기, 의료장비, 웨어러블 컴퓨팅 등 정확한 시간 성능을 요구하는 시스템에서 주로 사용되고 있는 운영체제이다.

국내 기술로는 iRTOS가 개발되어 있지만, 이는 기존 General ARM Archtecture 기반으로 이루어져 있기 때문 에 최근 다양하게 사용되고 있는 마이크로 컨트롤러 기 반의 Cortex-M4에서 직접 사용하기에 어려움이 있다. 이 에 따라 본 연구에서는 마이크로 컨트롤러를 위한 ARM Cortex-M4에 적용할 수 있도록 iRTOS의 내부 구조를 개 선한다. 기존 다양한 레지스터 및 5개의 시스템 모드를 지원할 수 있는 환경을 축소하여 2개의 모드 및 15개의 레지스터만을 사용할 수 있도록, 컨텍스트 관리 구조를 개선하고, 시스템 관련 함수를 개선한다. 또한 RTOS에서 중요한 Heat-Beat의 역할을 수행하는 시스템 타이머를 ARM Cortex-M4에 맞게 수정하여 실시간성이 보장될 수 있도록 하였다.

본 논문에서는 2장에서 관련 연구로 iRTOS에 대해서 소개하며, 3장에서는 iRTOS를 수정한 내역을 서술하고, 4장에서 정상적으로 동작하는 것을 확인하며, 마지막으 로 5장에서는 결론을 맺는다.

Ⅱ. 관련연구

1 iRTOS

iRTOS는 ARM9 기반의 플랫폼에서 동작하는 저전력 실시간 운영체제로서 멀티태스킹을 지원하며, 우선순위 기반의 선점형 실시간 운영체제로 필요에 따라 64, 256, 512의 우선순위를 제공한다. 또한 태스크간의 통신 및 동기화를 위해 세마포어, 메시지큐, 메시지 메일박스, 시 그널 및 태스크 포트를 지원하고 있으며, 정적 메모리 관 리기법 뿐만 아니라 동적 메모리 관리기법도 제공되고 있다. 그리고 저전력 기법인 DPM(Dynamic Power Management)를 적용하고 있다[1].

Ⅲ. 마이크로 컨트롤러를 위한 iRTOS의 구조 개선

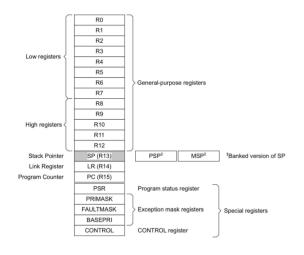
마이크로 컨트롤러는 기존에 사용되던 임베디드 시스 템보다 더욱 저전력으로 동작하며 최소한의 메모리 공간 을 가진다. 이를 위해 전통적인 ARM의 구조를 사용하지 않고 있으며, 이를 위해서 본 연구에서는 컨텍스트 관리 기법을 수정하고, 인터럽트 및 태스크 상태를 관리할 수 있는 USER/SYSTEM 모드의 구조를 Cortex-M에 맞게 변 경하는 과정을 수행하였다. 이를 통해 iRTOS를 마이크로 컨트롤러인 Cortex-M4에 적용할 수 있도록 하였다.

1. 콘텍스트 및 익셉션 관리 기법

IoT를 위한 마이크로 컨트롤러는 기존의 전통적인 ARM의 구조와 다른 구조를 가지고 있다. 다음 [그림 1] 과 [그림 2]은 기존 ARM의 구조와 Cortex-M으로 표현할 수 있는 마이크로 컨트롤러를 위한 ARM 시스템의 구조 의 차이를 나타낸다.

전통적인 ARM 시리즈에서는 특권 레벨에 따른 User/System 모드 및 FIQ, IRQ 등 각 모드에 따른 분리 된 SPSR mode 및 각각의 스택 포인터 레지스터를 보유 하고 있어서, 시스템의 목적에 따라 각각의 모드를 초기 화 하고 관리를 수행할 수 있었다. 이를 뱅크 레지스터라 고 불렀다.

Cortex-M 시리즈에서는 뱅크 레지스터를 삭제함으로 써 기존의 38개의 레지스터를 18개로 축소하였고, 이는 회로를 줄일 수 있기 때문에 시스템의 물리적인 크기가 작아질 수 있도록 하였다. 이를 관리하기 위해 더 이상 Mode에 대한 정보를 저장하지 않으며, 필요한 몇몇의 레지스터의 정보만을 컨택스트로 관리한다.



▶▶ 그림 3. Cortex-M4 시리즈의 레지스터

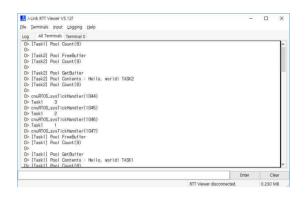
2. 인터럽트 및 익셉션 관리 기법

기존의 ARM 프로세서는 인터럽트나 예외가 발생했을 때 뱅크 레지스터를 사용하여 모드 및 컨택스트를 관리 하고 있으나, 회로가 축소된 마이크로 컨트롤러는 Tread Mode 및 Handler Mode로 구분하고 있으며, 이에 대한 모드 스위칭 및 컨텍스트 저장을 하드웨어 레벨에서 제 공하고 있다.

이를 활용하여 기존에 사용되던 ISR 루틴을 축소하였 고, 각 모드에 대한 레지스터를 저장하던 부분을 제외할 수 있었다. 또한, 통합된 인터럽트 및 익셉션 관리를 위 한 NVIC(Nested Vector interrupt Controller)가 도입되 어. 이를 활용할 통합 ISR를 사용하게 되었다.

Ⅳ. 실험 결과

다음은 개선된 iRTOS를 Cortex-M4가 적용된 개발보드 인 BMD-300에 적용한 화면이다. 여러 개의 태스크가 잘 실행되는 것을 확인할 수 있었다.



▶▶ 그림 4. iRTOS 동작 화면

V. 결론 및 향후 연구

본 논문에서는 마이크로 컨트롤러에 iRTOS를 적용하 는 연구를 수행하였다. 이는 기존 ARM 레지스터에서 사 용되던 코드를 수정하여 마이크로 컨트롤러에 맞게 축소 하였으며, 이를 통해 iRTOS가 잘 동작하는 것을 볼 수 있었다. 향후 연구로 BLE를 iRTOS에서 동작할 수 있도 록 하는 연구가 필요하다.

■ 참 고 문 헌 ■

- [1] 조문행, 이철훈 "다차워 기법을 이용한 실시간 태스크 스케줄링 알고리즘", 한국콘텐츠학회, pp 810-815, 2009
- [2] 임종수, "임종수의 Cortex-M3/M4완벽가이드(기초편)", 높이깊이(2014).
- [3] 오승택, 고영관, 이철훈. "cnu RTOS를 위한 저전력 기능 의 구현." 한국컴퓨터정보학회 학술발표논문집, 21.1 (2013, 1): 13-16.