

임베디드 OS를 적용한 전력용 디지털 장치의 개발

오재훈*, 권효철, 오성민, 홍정기
(주)효성 중공업연구소

A Development of Intelligent Electronic Device(IED) for Electric Power System applied embedded OS

Jae-Hoon Oh, Hyo-Chul Kwon, Sung-Min Oh, Jung-Ki Hong
HYOSUNG Corporation, Power & Industrial Systems R&D Center

Abstract - As a micro computer or micro processor technology has been advanced, the essential part which consists of electric power system has been changed into intelligent electronic device(IED), that is one kind of embedded system. There are already many parts changed into digital from analog in control and maintenance devices of electric system. These trends make advanced and novel concepts of devices and systems, that we couldn't even imagine in the analog circumstances. Now, the rate's of change is comparatively slower than IT field that is mainly represented by mobile communication, but that is a big change in the history of electric industries. Some people in those field called it Power-IT, and there have been many kinds of researches. Accelerated systems to go digital, users always and continuously demand more convenient, practical, and new functions. So, a lot of electric system manufactures have had tasks realizing many kinds of functions in an electronic device. According to the high-end technologies in semi-conductor and micro-computer field, manufactures can fully satisfy almost all of customer's needs, but there are other kinds of problems, such as large amount of program code, increasing complexity in proportion with program amount, and the hardness of maintenance and revision of the program. So nowadays, the necessity of embedded OS was emerged as a solution, and many researchers and manufactures have concentrated on studies related to embedded OS and its application.

1. 서 론

반도체 및 마이크로 프로세서 관련 기술, 그리고 임베디드 소프트웨어 기술 발전에 힘입어, 우리 주변에는 많은 종류의 IT 기기들이 보급되어 왔고, 이로 인해 일상생활은 점점 더 편리하게 변화하고 있다. 이러한 기술의 진보는 우리가 직접 피부로 느끼지 못하는 분야에서도 활발히 적용되고 있다. 100년이 넘는 전기 산업의 역사에서 IT 관련 기술의 전력 시스템 적용은 혁명적이라 할만한 큰 변화를 일으키고 있다. 아날로그 방식의 제어 장치들, 설비를 유지하고 관리하기 위한 제반의 아날로그 설비들이 점차 디지털 장치로 탈바꿈하고 있다.

초기에 전력 시스템에 적용되는 디지털 장치들은 기존의 아날로그 장치들의 역할을 단순히 디지털 방식으로 대체하는 수준에서 제작되어져서, 그 기능이 매우 간단한 임베디드 장치로 분류될 수 있었다. 하지만, 디지털 기술의 무한한 잠재력은 기존의 기능들을 넘어서 새로운 기술들, 새로운 개념들을 창출해 냈고, 이로 인해 디지털 장치들이 수행해야 하는 역할도 점점 확대되어졌다. 또한, IT 기기에 익숙해져 있는 사람들의 다양한 요구사항들을 설비 제작사들이 가능한 수용하는 시장 상황에서, 디지털 장치의 다양한 기능은 하나의 제품 경쟁력으로 인식되어 지고 있는 현실이다.

단순한 기능의 임베디드 장치는 인터럽트를 포함하는 매우 간단한 소프트웨어 구조로 구현이 가능하다. 또한 하나의 장치를 한 사람의 개발자가 전담하여 수행하는 것이 개발 속도나 효율성 면에서 탁월하였다. 하지만, 임베디드 장치의 기능이 다양해지면서, 임베디드 장치를 동작시키는 프로그램에 대한 새로운 고찰이 필요하게 되었다. 기능이 복잡해지면 그에 따라 프로그램의 분량이 기하급수적으로 늘어나게 되고, 이것은 프로그램의 복잡성을 증가시키며, 이러한 복잡성의 증가는 프로그램의 유지 및 보수에 어려움을 배가시키고 있다. 또한, 기능이 많아지면서 한 사람이 아닌 다수의 개발자가 하나의 제품 개발에 참여하게 되는데, 이때 효율적인 작업 방법이 강구되지 못하면 제품 개발은 큰 어려움에 봉착하게 된다. 이러한 어려움을 해결해 줄 수 있는 방법으로 대부분의 IT 기기에서 사용하는 방법이 임베디드 OS의 사용이며, 이러한 기술은 점점 전력 시스템에 적용되는 임베디드 장치에도 보급되어지고 있다.

본 논문에서는 임베디드 OS를 사용한 디지털 계전기의 개발을 소개한다. 계전기는 90년대 후반부터 디지털화하기 시작하였으며, 그 후 지속적인 발전을 진행하면서 현재에는 이더넷 통신, 간단한 PLC 기능, 그리고 그래픽 LCD나 터치스크린 방식 등의 편리하고 화려한 디스플레이를 제공하는 등의 기능이 시장에서 지속적으로 요구되어 지고 있다. 이러한 요구를 충족하는 기능을 확보하기 위해, 본 개발에서는 임베디드 OS를 적용하였다.

2. 본 론

2.1 임베디드 OS

임베디드 장치란 미리 정해진 특정 기능을 수행하기 위해 컴퓨터의 하드웨어와 소프트웨어가 조합된 전자 제어 시스템을 말한다. 그리고, 이러한 장치에서 사용되는 OS(Operating System)을 임베디드 OS라 한다. 일반적으로 운영체제는 하드웨어와 기본 시스템의 동작을 제어하고 관리하는 시스템 소프트웨어이다. 즉, 운영체제는 하드웨어 자원을 효과적으로 관리하고, 사용자에게 시스템을 보다 편리하게 사용할 수 있도록 하고, 또한 응용 프로그램의 오류로부터 시스템을 보호한다. 임베디드 OS도 동일한 특성을 갖는다. 다만, 임베디드 OS는 매우 적은 하드웨어 자원에서도 사용이 가능하도록 만들어졌을 뿐이다.

임베디드 OS도 일종의 소프트웨어이므로, 설치와 작동을 위해서는 일정한 사양의 하드웨어가 필요하다. 만약, 8비트 CPU를 기반으로 하는 간단한 임베디드 시스템을 구성한다면, 하드웨어적인 자원인 CPU와 메모리의 일정 부분을 점유해야 하는 OS를 설치할 필요성은 없을 것이다. 굳이 하드웨어 자원을 따지기 전에, 간단한 임베디드 장치라면 그 소프트웨어도 간단하게 구성되기 때문에 OS를 사용함으로써 얻을 수 있는 장점도 크지 않다. 명확히 어떤 기능, 어떤 사양 이상일 경우 OS를 적용하는 것이 좋다는 기준은 없다, 모든 결정은 개발될 제품의 사양, 기능, 호환성, 개발 환경 등을 면밀히 따져본 후, 개발자가 결정해야 할 몫이다.

임베디드 OS 중, 실시간성을 보장하는 종류를 RTOS(Real-Time OS)라고 부른다. 여기에서 실시간성이란, 무조건 빠른 처리를 의미하는 것이 아니라, 주어진 시간내에 올바른 결과를 낼 수 있음을 의미한다.

2.2 임베디드 리눅스

리눅스는 원래 386기반의 개인용 컴퓨터에서 동작하는 운영체제로 시작했다. 출발당시부터 리눅스는 공개 소스 정책을 표방하였고, 이 덕분에 시간이 지나면서 점차 다양한 하드웨어 플랫폼으로 퍼져나가기 시작했다. 그 부산물중의 한 가지가 임베디드 장치에 리눅스라는 운영체제를 이식할 수 있는 길이 열렸다는 것이다.

현재 임베디드 리눅스 이외에도 많은 종류의 임베디드 OS들이 존재하고, 사용되어지고 있다. 시스템의 특성에 따라 적합한 OS가 다를 것이며, 또한 OS 마다의 장단점도 다를 것이다. 그럼에도 현재 임베디드 리눅스는 임베디드 장치 개발시에 우선적으로 고려되어지는 OS 중 하나이다. 그 장단점을 살펴보면 다음과 같다.

◎ 장점

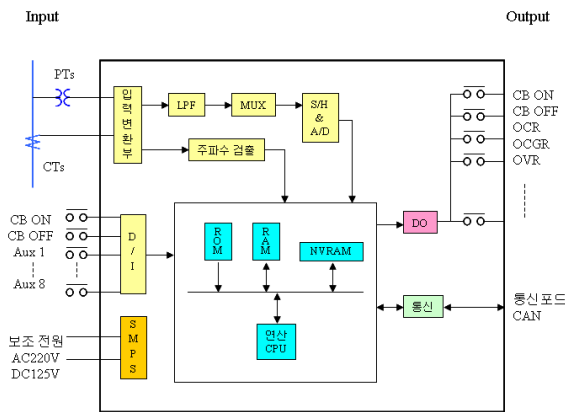
- 낮은 비용, 자유로운 라이선스, 공개 소스 운영체제
- 단일 운영체제 플랫폼과 다양한 주변 장치 지원
- 뛰어난 개발 환경
- 기존 임베디드 운영체제에 없는 첨단 기능
- 풍부한 개발자

◎ 단점

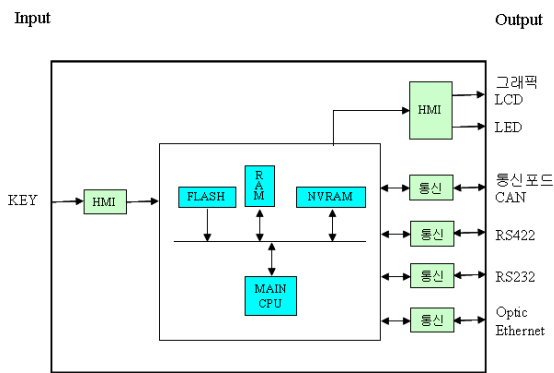
- 거대한 커널 크기
- 공개 라이선스 문제
- 보안과 안정성 문제
- 적합성과 시장 압력

2.3 하드웨어의 구성

본 논문에서 소개하는 계전기는 크게 연산부와 응용부로 구분할 수 있다. 연산부는 계전요소 및 각종 전기량의 계측과 관련된 연산, 그리고 DI(Digital Input)과 DO(Digital Output) 관련된 실시간 처리를 담당하는 부분으로 TI사의 TMS320F2812를 중심으로 구성되었다. 연산부는 실시간 연산처리가 주요한 임무로 OS는 사용하지 않고, firmware 레벨의 stand-alone 방식으로 동작한다. 응용부는 계전 및 계측 등의 연산을 제외한 부분의 역할을 담당하는데, 통신 및 HMI(Human Machine Interface)가 주요한 역할이 되겠다. 응용부는 XScale core를 사용하는 INTEL사의 IXP420 마이크로 프로세서를 중심으로 구성되었다. IXP420은 산업용 제어 장치 및 네트워크 장치 개발을 위해 특화된 마이크로 프로세서로, 응용부에서 수행해야 할 처리의 많은 부분이 통신 관련 처리이기 때문에 선정하였다. 응용부에는 OS로 임베디드 리눅스(kernel 2.6)가 적용되었다. 계전기라는 기본 기능에 해당하는 부분은 고성능 DSP 기반의 하드웨어에서 실시간으로 처리되어지고, 본 기능 이외의 부가적인 기능에 해당하는 부분은 고성능 임베디드 프로세서와 임베디드 OS를 기반으로 운영되도록 구성하였다. 응용부와 연산부 사이의 데이터 교환은 통신을 이용하여 수행하도록 하였다. 응용부와 연산부의 개략적인 하드웨어 구성은 아래의 그림1, 그림2와 같다.



〈그림 1〉 연산부 하드웨어의 구성



〈그림 2〉 응용부 하드웨어의 구성

2.4 소프트웨어의 구성

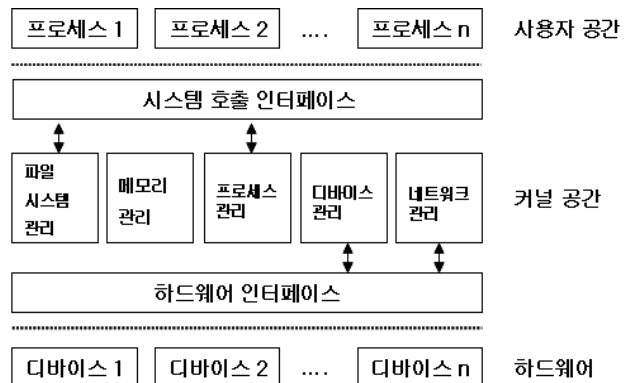
연산부의 소프트웨어는 OS를 사용하지 않고, 인터럽트를 포함하는 순차처리를 기본으로 하는 stand-alone 방식으로 작성되었다. 연산부는 실시간 처리가 매우 중요하기 때문에, 실시간 처리가 가능한 OS를 검토하였으나 OS를 사용하여 얻을 수 있는 장점이 크지 않다고 판단되었다. 하지만, 응용부의 소프트웨어는 임베디드 OS를 사용하는 것으로 결정하였다. 응용부의 경우 역할의 대부분이 통신 기능에 집중되어 있고, 그 중에서도 이더넷 통신이 가장 중요한 부분이 되겠다. 복잡한 이더넷 통신을 신뢰성 있게 구현하기 위해서는 OS의 도움을 받는 것이 필요하다는 결론에 도달하였으며, 임베디드 OS로는 임베디드 리눅스를 선정하였다. 응용 프로그램은 임베디드 리눅스 기반에서 실행이 되어지게 된다. 응용 프로그램은 그 기능별로 프로세스를 구분하여 임베디드 리눅스에서 제공하는 장점중 하나인 멀티 프로세스 기능을 충분히 살려서 전체적인 기능이 수행되도록 하였다. 사용하는 리눅스는 kernel 2.6을 기반으로 한다. 리눅스 커널은 논리적으로 시스템 호출 인터페이스와 디바이스 인터페이스 사이에 위치한다. 사용자 프로세스, 즉 사용자의 어플리케이션 프로그램은 시스템 호출 인터페이스를 통해 커널과 정보를 교환

하며, 커널은 디바이스 인터페이스를 통해 디바이스, 즉 하드웨어와 정보를 교환한다. 리눅스 커널은 프로세스 관리, 메모리 관리, 파일 시스템 관리, 디바이스 관리, 네트워크 관리와 같이 5가지 부분으로 나뉘며, 그 구체적인 기능은 아래의 표와 같다.

〈표 1〉 리눅스 커널의 기능

구분	기능
프로세스 관리	프로세스의 생성 및 소멸, 프로세스 사이의 통신, 프로세스 스케줄링 동기화
메모리 관리	가상 메모리 관리 기법 제공, 메모리 하드웨어의 효율적 관리
파일시스템 관리	가상 파일시스템에 의한 다양한 파일시스템 지원, 파일 및 디렉토리 관리
디바이스 관리	입출력 요청 작업의 검증 및 스케줄링, 주변 장치와 메모리 간의 데이터 전송
네트워크 관리	통신 프로토콜 구현, 네트워크 라우팅 및 주소 지정

기능 구현을 위한 프로그램은 다수개의 프로세스들로 분리하여 프로그램 작성한다. 이러한 구성에 의해 여러 명의 개발자가 동시에 개발 작업을 진행할 수 있으며, 각각의 개발자는 각자가 맡은 프로세스에 대해서 코딩과 디버깅을 수행하고, 전체적인 소프트웨어 개발 조율자가 각 프로세스 간의 데이터 흐름을 조정하여 원하는 기능을 최종적으로 구현할 수 있으며, 또한 이러한 방법에 의해서 전체적인 디버깅도 빠르게 진행해 나갈 수 있다. 아래의 그림 3은 전체적인 개발 프로그램의 구조를 도식화 하였다. 그림을 통해 이해할 수 있듯이 프로그램 개발자는 하드웨어를 사용하기 위해서 커널의 도움을 받는다. 이것은 OS를 사용하지 않을 경우 어플리케이션 프로그램에서 하드웨어도 직접 접근해야 하는 방법에 비해 보다 안정적인 프로그램 구동을 보장할 수 있고, 또한 프로그램 개발자의 부담을 덜어 그만큼 제어 알고리즘이나 기능 구현에 좀 더 집중할 수 있는 공간을 제공할 수 있다.



〈그림 3〉 임베디드 리눅스 기반의 프로그램 구조

3. 결 론

본 논문에서는 임베디드 OS를 적용한 전력용 디지털 장치로서, 디지털 계전기의 개발을 소개하였다. 임베디드 OS의 필요성과 개괄적인 특성을 고찰하였으며, 개발품의 하드웨어와 소프트웨어 구성에 대해 전반적으로 살펴보았다. 현재 개발품은 프로토타입으로 만들어져 지속적인 기능 시험을 실시하고 있다.

고기능의 디지털 장치 개발에 OS를 적용함으로써, 각 개발자의 협업을 용이하게 하여 개발 시간을 단축할 수 있었으며, 안정적인 OS의 사용으로 인해 고사양의 기능들의 안정적인 동작을 보장할 수 있게 되었다.

[참 고 문 헌]

- [1] 박재호, 임베디드 리눅스, 한빛미디어, p 86-107, 2003
- [2] P. 라가반 외, 임베디드 리눅스 시스템 설계와 개발, p 9-12, 2007
- [3] 우중정 외, 임베디드 리눅스 기초와 응용, p 100-112, 2007