

내장형 제어기를 위한 컴포넌트 소프트웨어 아키텍처

Component Software Architecture for Embedded Controller

。 송 오 석*, 김 동 영*, 전 윤 호*, 이 윤 수*, 홍 선 호*, 신 성 훈*, 최 중 호*

* 서울대학교 전기공학부 (Tel : 82-02-875-9183; Fax : 82-02-888-4182;
E-mail: {osok, young, yunho, chchoi, yslee, sun, tamarix}@csl.snu.ac.kr)

Abstract : PICARD (Port-Interface Component Architecture for Real-time system Design) is a software architecture and environment, which is aimed to reduce development time and cost of real-time, control system. With PICARD, a control engineer can construct a control system software by assembling pre-built software components using interactive graphical development environment. PICARD consists of PVM(Picard Virtual Machine), a component library, and PICE(Picard Configuration Editor). PVM is a real-time engine of the PICARD system which runs control tasks on a real-time operating system. The component library is composed of components which are called task blocks. PICE is a visual editor which can configure control tasks by creating data-flow diagrams of task blocks or ladder diagrams for sequential logics. For the communication between PVM on a target system and PICE on a host computer, a simple protocol and tools for stub generation was developed because RPC or CORBA is difficult to be applied for the embedded system. New features such as a byte-code based run time system and a simple and easy MMI builder are also introduced.

Keywords : component software, real-time system, visual editor

1. 서론

공장 자동화, 로봇의 제어 등에 사용되는 제어기를 위한 소프트웨어는 일반적으로 C, C++, 어셈블리어 등의 언어를 이용하여 작성되어 왔다. 그러나 이를 위한 체계적인 방법론이 제공되지 않아 프로그램의 작성에 많은 시간과 노력이 소요될 뿐 아니라 프로그램의 유지보수가 힘들고 재사용이 거의 없는 단점을 지닌다. 이를 극복하기 위하여 소프트웨어 컴포넌트들을 바탕으로 각 컴포넌트의 입력과 출력을 연결하는 방식으로 제어시스템을 손쉽게 개발할 수 있는 방법론과 이를 위한 소프트웨어 아키텍처인 PICARD(Port Interfaced Control Architecture for Real-time system Design)을 제안하였다[1]. Picard는 재사용성을 극대화 하는 컴포넌트 구조를 가지고 있는 태스크블록 라이브러리, 여러 실시간 운영체제에 이식되어 효율적이고 예측이 가능한 실행성능을 보장하는 실시간 가상기계(Picard Virtual Machine, PVM), 태스크블록을 조합하여 실제 응용프로그램을 손쉽게 만들 수 있도록 해주는 PICE(Picard Configuration Editor), JavaBeans로 구성이 되는 GUI 등으로 이루어져 있다.

최근 PC의 성능 및 안정성, 가격경쟁력이 높아짐에 따라 자동화 환경에서 사용되는 여러 제어기를 구현하는데 있어서 전용의 하드웨어 및 소프트웨어를 사용하는 것에서 벗어나 PC를 이용한 제어기를 적용하는 예가 많아지고 있다. PICARD 또한 실시간 운영체제를 탑재한 PC환경을 바탕으로 하여 PC를 이용한 제어기를 만드는 데 주안점을 두고 개발되어 왔다. 그러나 근래 내장형 제어기에서도 사용되는 프로세서의 성능이 급격히 향상되고, 수행해야 할 일이 복잡해짐에 따라 과거의 간단한 형태의 순차적 논리 제어에서 벗어나, 실시간 운영체제를 채용하는 경우가 많아지고 있어, PICARD를 내장형 제어시스템에까지 적용할 수 있게 되었다. 본 논

문에서는 전윤호 등[1] 이후에 PICARD에 구현된 여러 가지 새로운 기능과 이를 위한 구현 방법을 설명하고 이것이 어떻게 내장형 제어기에 적용이 되는지를 보인다.

2. PICARD의 소개

PICARD는 제어 시스템의 RAD(Rapid Application Development)를 위해 Task block이라 불리는 소프트웨어 컴포넌트를 사용한다. Task block은 C++ class header 파일을 변형한 형태의 taskblock.p 파일에서 task block의 입출력을 정의하고, task block.cpp 파일에서 그 행동을 함수로 기술해 주면 pgen이라는 프로그램에 의해 task block에 대한 여러 가지 meta 정보들을 담고 있는 파일들이 자동으로 생성되어 task block library에 추가된다.

여러 가지 기능을 가지는 Task block들은 PICARD 실시간 가상기계(PVM)에 적재되고, PVM에 연결된 editor인 PICE에서 task를 만드는데 사용된다. Task는 task block을 data-flow 형태로 연결하여 매 주기마다 실행될 기능을 표현한다.

Task들 사이의 data 교환은 Chimera[2,3]에서와 같이 local variable table과 global variable table사이의 data를 복사하는 방식으로 일어난다. 이를 통해 각 task들이 사용하는 data의 일관성을 지킬 수 있고, 우선순위 역전 등의 문제를 해결할 수 있다.

PICE에서는 data-flow 형태의 task block diagram뿐만 아니라 PLC에서 많이 사용되는 ladder diagram의 형태로 task를 만드는 것을 지원한다. Ladder diagram의 입력과 출력 변수들은 task block diagram으로 구성된 다른 task들로부터 읽어들이거나, 또는 사용될 수 있다.