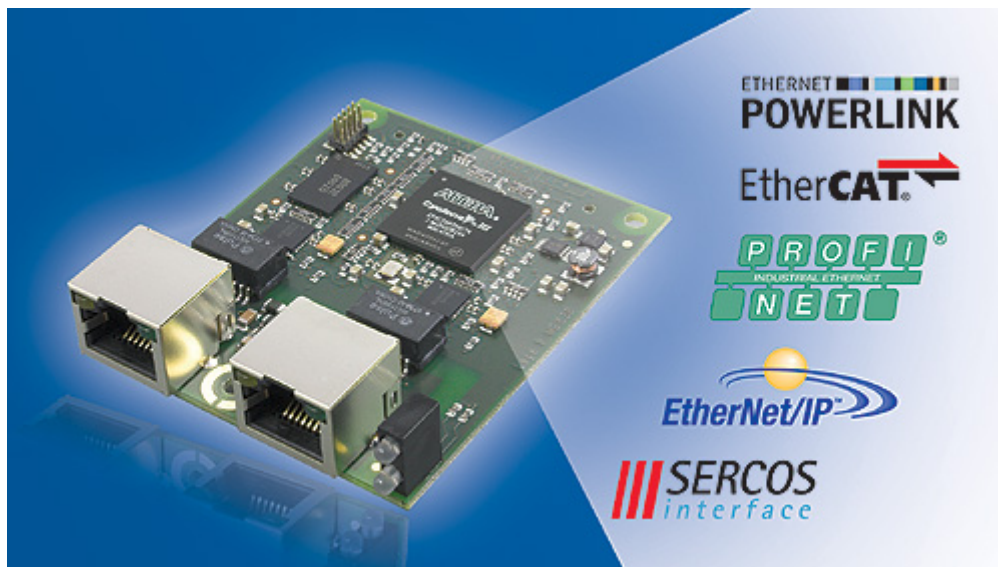


산업용 실시간 이더넷 인터페이스 제품 소개



오늘날, 이더넷에 기반한 통신과 관련된 산업용 자동화 시장은 그 수요가 증가하고 있습니다. 전통적인 필드 버스들과 함께, 여러 다양한 프로토콜 표준들이 산업용 실시간 이더넷에 사용될 수 있으나, 이것들은 디바이스 제조사들에 의해 지원되어야 합니다. 이러한 서로 다른 이더넷 표준들은 그들의 하드웨어와 소프트웨어 요건에 따라 매우 다양해지기 때문에, 디바이스내에서 이들의 통합을 위한 특수한 프로토콜 솔루션이 요구됩니다.

다양한 시스템 방법들

이더넷을 통한 산업용 통신의 대중적인 방법을 찾으려는 IAONA 의 노력에도 불구하고, 수많은 기존의 필드 버스 공급자들의 후원으로 본질적으로 다른 프로토콜들이 생겨나게 되었습니다. 산업용 통신에 적절한 이더넷을 만들기 위한 11 개의 개념들이 현재 IEC 61158 내에 표준화되어 있습니다. 이른바 80 년대의 "필드 버스 전쟁"과 유사하게, 결국에는 11 개의 산업용 실시간 이더넷 중 몇 개만이 시장에서 살아 남을 것입니다.

Profinet 과 EtherNet/IP 에 관한 솔루션 외에, 자동화 기술의 시장 선도자 Siemens/PNO 와 Rockwell/ODVA 로 부터, 유럽에서 EtherCAT (Beckhoff/ETG), POWERLINK (B&R/EPG) 그리고 Sercos III (Bosch-Rexroth/ITG) 솔루션들이 다른 것들을 압도할 것으로 예상되고 있습니다.

분명한 것은, 하나의 공통적인 솔루션을 찾기 위한 노력에도 불구하고 디바이스 제조업자들이 그들의 기존 필드 버스 솔루션과 호환되는 산업용 실시간 이더넷 솔루션을 찾기 위한 고집 때문이었습니다. 이러한 제조업자들을 위해, 산업용 이더넷에 의해 소개되는 어떤 기능들은 무난하게 이동할 수 있도록 기존의 필드 버스 솔루션의 디바이스 프로파일과 애플리케이션

프로토콜들과 호환되는 것이 매우 중요합니다.

CAN 버스에서 작용하는 DeviceNet 상위 계층 프로토콜을 채용하는 소위 Common Industrial Protocol (CIP)을 살펴보면, EtherNet/IP 솔루션도 CIP 애플리케이션과 디바이스 프로파일들을 사용합니다. 비슷한 방식이 CANopen 메커니즘의 애플리케이션 서비스와 디바이스 프로파일들이 실시간 이더넷으로 변환된 POWERLINK 에 사용되고 있습니다. 따라서 POWERLINK 는 애플리케이션 관점에서 CAN 을 산업용 이더넷으로 손쉽게 이동시킬 수 있는 CANopen 과 비슷합니다.

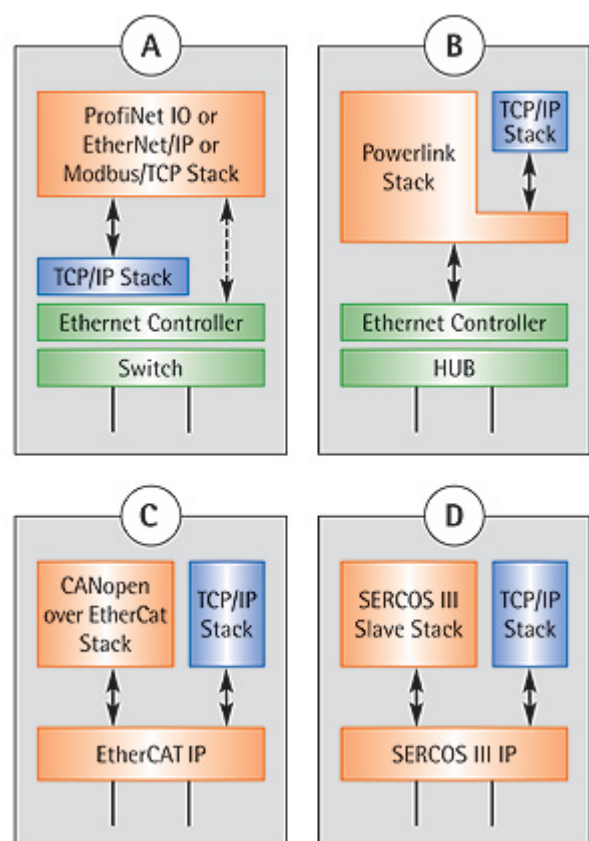
산업용 실시간 이더넷 프로토콜 구현

다양한 이더넷 기반 프로토콜들의 여러 접근법과 요건들로 인하여, 프로토콜들 중의 일부는 표준 이더넷 컨트롤러를 사용하여 구현될 수 있으며, 다른 것들은 IP 코어 형태나 ASICS 을 지원하는 특정 하드웨어를 필요로 하기도 합니다.

또한, 산업용 시장에서 중요한, line topology 가 산업용 이더넷 네트워크의 설계시 고려되어야 합니다. 이더넷은 허브와 스위치 같은 active components 를 통한 연결이 필요하므로, 이더넷 인터페이스는 원하는 line topology 를 지원하기 위한 필요 요소들을 실행해야만 합니다. 사용되는 프로토콜에 따라, 스위치, 허브 또는 특정 인터페이스가 필요합니다.

오른쪽 그림은 특정 프로토콜을 사용하여 슬레이브를 구현하는데 필요한 여러 가지 구조들을 대략적으로 보여주고 있습니다. Profinet RT IO Device, Ethernet/IP Adapter 와 Modbus/TCP Client/Server 의 구현을 위해, 일반적인 이더넷 컨트롤러가 사용될 수 있으며, 이것은 이미 많은 CPU 들에서 실행되고 있습니다. Line topology 지원을 위해서는, 이더넷 인터페이스에서 스위치 통합이 유용합니다. 이를 위해, 통합된 PHY 와 함께 조립식 스위치 모듈을 이용할 수 있습니다.

노드 또는 제어된 노드를 관리하는 POWERLINK 또한 일반 이더넷 컨트롤러를 사용하여 구현될 수 있습니다. 그러나, 매우 짧은 주기 시간이 지원되어야 한다면, 수신된 메시지에 빠른 회신 전송이 가능한 수정된 이더넷 컨트롤러가 필요합니다. 더 나아가, 이더넷 인터페이스의 허브 구현을 위해서 line topology 를 지원하는 것이 유용합니다. 스위치 기술이 허브 기술을



대신하게 되었으므로, 허브 모듈들은 더 이상 이용할 수 없습니다. 이로 인해, 이더넷 컨트롤러와 PHY 사이에는 프로그램이 가능한 칩이 필요합니다.

EtherCAT 과 SERCOS III 슬레이브는 특별한 하드웨어 인터페이스를 필요로 합니다. EtherCAT 의 경우는 Beckhoff 사가 ASIC 또는 IP 코어 형태로 제공하고 있으며, SERCOS III 의 경우는, Sercos International 에 의하여 IP 코어가 제공되고 있습니다. 이 하드웨어 인터페이스는 이더넷 컨트롤러를 대체하며 line topology 지원에 필요한 기본 조직을 제공합니다. 또한 사용되는 프로토콜 스택의 실행을 위한 CPU 도 필요로 합니다.

Profinet IRT 의 구현을 위해서는, 현재는 오직 전용 CPU 만 이용할 수 있습니다 (ERTEC 200/400 or NetX). 이러한 CPU 들은 특수 Profinet IRT 스위치를 포함하고 있으며 Profinet IO Device 프로토콜 스택을 실행합니다.

과거의 경험에 의하면 디바이스 제조업체들이 모든 관련 프로토콜들을 지원하지 않을 수 없을 것입니다. 이더넷 인터페이스를 위한 대략적인 여러 구조에서 보여지듯, 디바이스 제조업체들은 모든 관련 프로토콜들을 실행하려면 이더넷용 하드웨어 인터페이스를 제공해야만 합니다.

모든 관련 프로토콜들을 지원 하는 것이 가능하게 하는 하드웨어 인터페이스는 첫째로, 각 프로토콜을 위한 특정 하드웨어와 소프트웨어 방법을 실행 하거나 아니면 둘째로, 각각의 프로토콜을 지원하는 피기백 모듈을 사용하는 것으로 이루어질 수 있습니다.

분명 값비싼 방법인 위의 첫번째 방법과 비교했을때, 피기백 모듈을 사용하는 가장 큰 장점은 통신의 독립적인 처리와 관리입니다. 독립적인 이더넷 모듈은 자동으로 통신을 실행할 수 있으며 반면 디바이스 CPU 는 성능의 손실없이 자신의 애플리케이션을 처리할 수 있습니다.

디바이스 애플리케이션과 서로 다른 통신 프로토콜들 간의 공통적인 인터페이스 모듈은 유지가 훨씬 쉽고 추가적인 프로토콜 통합도 더 용이합니다.

실시간이 가능한 이더넷 프로토콜을 위한 프로토콜 스택은 통신에 의해 요구되는 짧은 주기 시간에 의해 야기되는 CPU 성능의 막대한 부분을 소비합니다. 애플리케이션과 프로토콜 스택이 하나의 CPU 에서 실행되고 필요한 성능을 얻을 수 없다면, 강력한 32 비트 CPU 들이 사용되어야 할 것입니다. 그러나 서로 다른 CPU 에 있는 애플리케이션과 통신의 분리는 보다 강력한 애플리케이션 CPU 를 필요로 하지 않습니다. 프로토콜 스택이 이더넷 인터페이스의 한 개 CPU 에서 실행된다면, 디바이스 CPU 는 애플리케이션을 위해 모두 사용될 수 있고 애플리케이션의 조건에 따라 선택 가능하며 또는 이미 기존의 CPU 라도 사용할 수 있습니다.

그러나 실시간 이더넷 인터페이스로서의 피기백 모듈 사용은 서로 다른 모듈들이 다양한 구조들을 지원해야 하기 때문에, 최선의 방법은 아닙니다.

Field Programmable Gate Array (FPGA)에 근거한 범용적 방법

모든 산업용 실시간 이더넷 프로토콜의 물리적 계층이 IEEE802.3 에 기초하고 있으므로, 물리적 연결 관점에서는 모든 프로토콜들을 지원하는 하나의 하드웨어 설계가 가능할 수 있습니다. 그럼에도 불구하고, 이것은 몇몇 프로토콜들이 특정한 처리 체계를 필요로 하는 특수한 실시간 기능을 갖고 있기 때문에 문제가 됩니다. 현재 이용 가능한 CPU 들 중에, 하나를 제외하고, 서로 다른 프로토콜들의 모든 요건을 충족하는 CPU 는 없습니다. 이 문제를 해결하기 위해, 자유롭게 프로그램이 가능한 로직 칩이 사용될 수 있습니다. Altera 의 Cyclone® 계열 같은, FPGA (Field Programmable Gate Array)는, 빠른 프로그램이 가능한 로직 외에, NIOS® II 로 불리는, 32 비트 RISC 구조를 이용한 역동적인 loadable and scalable 프로세서 코어들을 위한 옵션을 제공합니다. 이 조합으로, 서로 다른 프로토콜들의 모든 요건들이 실제로 충족될 수 있습니다.

EtherCAT 과 SERCOS III 같은, 특별한 사양의 프로토콜들을 위해, 특수한 하드웨어 인터페이스들이 FPGA IP 코어의 형태로 제공되는데, 이는 FPGA 에 로드가 가능하며 NIOS® 프로세서에 연결될 수 있습니다. FPGA IP 코어는 이더넷 패키지의 빠르고 특수한 처리를 책임지며, NIOS® II 프로세서는 버스 특정 프로토콜 스택을 실행합니다. 특정 하드웨어 인터페이스가 필요하지 않다면, 스위치와 허브 기능 같은 일반적인 이더넷 컨트롤러는 FPGA IP 코어로 이용할 수 있습니다.

FPGA 기반 이더넷 인터페이스의 개발을 위해서는, FPGA 는 외부 요소에 연결되어야 하는데, 이것은 CPU 와 같이 이루어지기 때문입니다. 필요 조건들로는 oscillator 또는 quartz, 동적 RAM 그리고 이더넷 인터페이스를 위한 두 개의 PHY 들과 시리얼 플래시 칩이 포함됩니다. FPGA configuration 파일과 프로그램 모두, FPGA 내의 CPU 에서 실행되는, 시리얼 플래시에 저장됩니다. FPGA 기반 하드웨어 솔루션은 CPU 기반 솔루션과 같은 가격대에서 비교했을 때 추가적인 요소를 필요로 하지 않으나 매우 높은 유연성을 제공합니다.

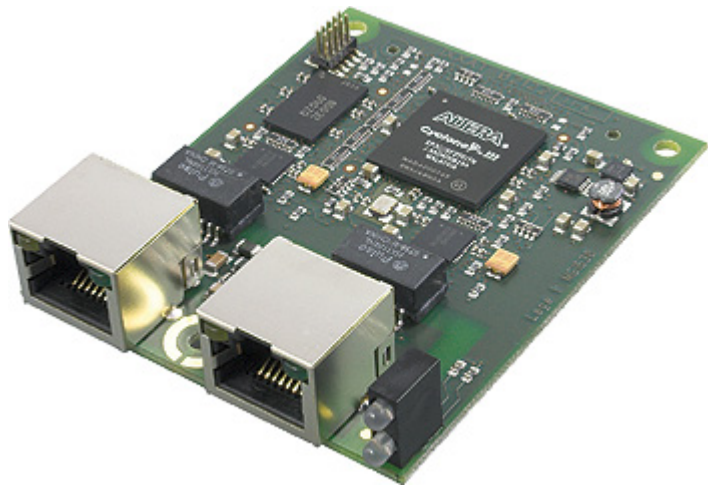
FPGA configuration file 의 생성은 Quartus™ II 라 불리는, Altera 의 composition tool 로 이루어집니다. 이 툴은 다양하게 제공되는 IP 코어의 리스트 cores (예. processor, memory interface, PCI 같은 external interfaces, shared memory, SPI, 기타)에서 필요한 기능들을 선택하고 연결하여 이것을 VHDL 또는 Verilog 에 기록된 missing programming 메커니즘으로 보완하여 FPGA configurations (FPGA design)의 생성을 가능하게 합니다. 또한, 이 툴은 사용자가 설계한 것을 시뮬레이트하고 실제 전류 소비와 더불어 시간을 분석할 수 있게 합니다. 끝으로 이 툴은 FPGA 용 binary file 을 생성하는데, 이것은 JTAG 인터페이스 같은 것을 통해 플래시에 업로드됩니다.

NIOS II 소프트웨어 개발을 위해, Eclipse 와 GNU 에 기반한 자유로운 개발 환경을 이용할 수 있습니다. 생성된 바이너리 코드는 시리얼 플래시 메모리에 FPGA configuration file 과 함께 저장됩니다. 소프트웨어 디버깅은 FPGA 의 JTAG 인터페이스를 통해 실행될 수 있습니다.

FPGA 기반 이더넷 연결의 장점은 하나의 하드웨어 플랫폼이 모든 산업용 실시간 이더넷 프로토콜들을 지원할 수 있다는 것입니다. 단 필요한 FPGA configuration 과 소프트웨어가 선택된 프로토콜을 위해 업로드 되어야만 합니다. 이것은 생산 단계 동안 혹은 심지어 디바이스 CPU 를 사용하여 나중에 이루어질 수도 있습니다. 그 결과 향후 이더넷 기반 프로토콜 표준들은 하드웨어 변경 없이 언제라도 지원될 수 있습니다. 물론, 특정 업체의 특수 칩을 사용하기 위한 별도의 소스는 거의 없지만, 그러나 적당한 노력으로 제조업체는 또 다른 FPGA 공급자로 전환할 수 있는데, VHDL 과 Verilog 가 FPGA 프로그래밍 표준 언어이기 때문입니다.

범용 실시간 이더넷 인터페이스 모듈

FPGA 기술을 바탕으로, IXXAT 는 산업용 이더넷 모듈을 만들었는데 이것은 이더넷 인터페이스를 어떤 종류의 디바이스에서도 빠르게 활용할 수 있도록 해줍니다. 이 모듈은 모든 필요한 요소들과 인터페이스들이 포함되어 있으며 다음의 프로토콜들을 지원합니다: Profinet RT IO, Ethernet/IP, POWERLINK, EtherCAT, SERCOS III 와 Modbus/TCP. 추가적인 프로토콜들이 언제라도 구현 가능합니다.

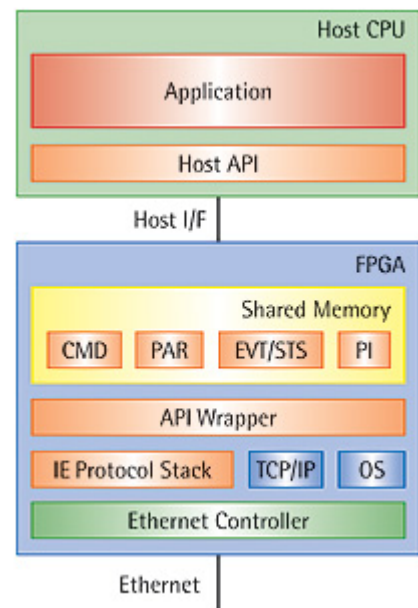


모듈을 사용자의 하드웨어에 통합하기 위해서는, 플러그가 사용되어 모듈의 전원 공급이 제공되어야 하며 시리얼 인터페이스 또는 parallel address/data 인터페이스를 통해 디바이스 CPU 와 모듈 간의 통신이 수행됩니다.

디바이스 소프트웨어에 대한 연결은 디바이스 소프트웨어에 일반적인 방식으로 서로 다른 산업용 이더넷 프로토콜들의 기능을 제공하는, 포함되어 있는 프로그래밍 인터페이스를 사용하여 이루어집니다. 따라서 디바이스 소프트웨어가 사용되는 산업용 실시간 이더넷 프로토콜과 독립적으로 개발될 수 있습니다. 또한 향후 프로토콜 확장도 디바이스 소프트웨어의 주요 수정 없이 이루어질 수 있습니다. 프로그래밍 인터페이스는 C 코드로 전달되며 디바이스 CPU 에 쉽게 적용될 수 있습니다.

다음 그림은 프로그래밍 인터페이스를 4 개의 영역으로 분할하는 것을 보여주고 있습니다: command interface (CMD), event and status interface (EVT/STS), parameter interface (PAR), process data interface (PI).

인터페이스 모듈은 명령 인터페이스를 통해 디바이스 소프트웨어로 제어됩니다. 따라서 Init, Reset, Connect 와 같은 명령어들을 이용할 수 있습니다. 디바이스 소프트웨어는 네트워크와 모듈 상태에 대한 정보와 더불어 이벤트와 상태 인터페이스를 통해 발생한 에러들을 수신합니다. 파라미터 인터페이스는 디바이스 파라미터에서 읽기와 쓰기 액세스들을 처리하는데 사용됩니다. 때로는 디바이스들이 1000 개 이상의 파라미터들을 갖기도 하므로 (예. 드라이브), 이 데이터는 인터페이스 모듈에 저장되는 것이 아니라 이 데이터에 대한 액세스가 실행을 위해 디바이스 소프트웨어로 직접 건네집니다.



파라미터들은 사용되는 프로토콜에 따라 서로 다르게 할당됩니다. 이것 때문에, 파라미터 관리자가 사용되어 액세스를 디바이스 소프트웨어로 보내기 전에 서로 다른 종류들의 할당을 하나의 공통적인 그리고 프로토콜과는 독립적인 단계로 가져옵니다. 최종적으로, 전송을 위한 처리 데이터는 처리 데이터 인터페이스 내에서 인풋과 아웃풋 데이터로 분리됩니다. 실시간 가능 프로토콜을 위한 디바이스 소프트웨어에 의해 동기식 처리 데이터의 처리가 가능하도록, 인터페이스 모듈은 각 통신 주기의 초기 단계에서 디바이스 CPU 를 위한 인터럽트를 생성할 수 있습니다.

산업용 이더넷 모듈은 FPGA 기술을 바탕으로 하고 있기 때문에, 사용자의 특정 인터페이스들은 약간의 작업으로 모듈 내에 통합될 수 있습니다. 그러한 인터페이스의 예로 모듈을 디바이스에 연결하기 위한 CAN 인터페이스가 될 수 있습니다. 더 나아가, 프로그래밍 인터페이스는 디바이스에서 기존에 이용하던 사용자의 특정 프로그래밍 인터페이스로 대체될 수도 있습니다.

산업용 이더넷 모듈을 사용하여 디바이스 제조업체들이 얻을 수 있는 이점으로는, 신속한 제품 출시와 저렴한 개발 비용입니다. 또한 이미 검증되고 공인된 모듈을 사용하므로 개발 위험을 최소화할 수 있습니다. 서로 다른 프로토콜들의 유지보수는 일반적인 제품 유지보수의 범위 내에서 IXXAT 사에 의해 지원되며, 따라서 고객들은 프로토콜들의 버전과 업데이트에 관해 염려할 필요가 없습니다.

요약

FPGA 기술은 하드웨어 개발에 매우 적합한 기술로, 서로 다른 real-time 가능 산업용 이더넷 프로토콜들과 non-real-time 산업용 이더넷 프로토콜들을 지원합니다. 낮은 전력 소비 외에도, 이것은 확장된 온도 범위에서 하드웨어 개발을 할 수 있습니다. 비용을 살펴보다라도, 이러한 솔루션은 CPU 기반 솔루션보다 더 비싸지 않습니다. 산업용 이더넷 모듈과 함께, IXXAT 사는 모든 산업용 실시간 이더넷 프로토콜들을 지원하는 이더넷 인터페이스의 빠르고, 안전하며, 유연한 통합을 위해 FPGA 기반 솔루션을 제공하고 있습니다.