(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2011년06월03일
(11) 등록번호 10-1038619
(24) 등록일자 2011년05월26일

(51) Int. Cl.
H04L 12/12 (2006.01) *B25J 9/16* (2006.01)
H04L 29/06 (2006.01)
(21) 출원번호 10-2010-0043446
(22) 출원일자 2010년05월10일
심사청구일자 2010년05월10일
(56) 선행기술조사문헌
논문(2007.12)
KR200363404 Y1

(73) 특허권자
인하대학교 산학협력단
인천 남구 용현동 253 인하대학교
주식회사 서보산전
경기도 부천시 원미구 약대동 193번지 부천테크노파크 401동 802호
(72) 발명자
김덕민
인천광역시 서구 가좌2동 81-10 현대아파트 2동 604호
이진영
경기도 부천시 원미구 상3동 528-3 라일락마을 2328동 404호
(뒷면에 계속)
(74) 대리인
배용철

전체 청구항 수 : 총 5 항 심사관 : 김창범

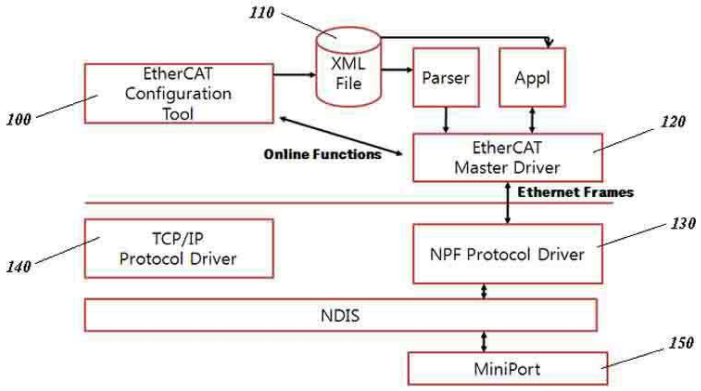
(54) 로봇용 다자유도 스마트 액추에이터를 위한 네트워크 기반 제어 모듈

(57) 요 약

본 발명은 로봇용 다자유도 스마트 액추에이터를 위한 네트워크 기반 제어 모듈에 관한 것으로, EtherCAT Configuration Tool을 이용하여 EtherCAT 마스터 디바이스 환경 설정에 필요한 내용과 초기화 명령, 실행 동작에 대해 정의하여 XML 파일을 생성하고, 생성된 XML 파일은 각 구문분석 작업을 통해서 EtherCAT Master Driver에서 동작하며, 사용자 인터페이스(UI)는 Application을 통해 동작하고, EtherCAT 마스터 드라이버는 내부적으로 NPF Protocol Driver와 TCP/IP Protocol Driver를 통해 MiniPort에서 입/출력된 데이터를 분석 및 생성하도록 구성된 EtherCAT 마스터와; EtherCAT 컨트롤러(ET1100)와 통합 보드의 메인 컨트롤러인 TMS320F20335에 버스 인터페이스 방식으로 구현되고, 16비트의 데이터 버스와 16비트의 어드레스 버스로 인터페이스되며, 메인 컨트롤러의 0x100000-0x10FFFF 영역에 메모리를 할당하도록 설계되고, RJ45 포트를 통해 이더넷 연결되도록 구성된 EtherCAT 슬레이브와; RJ45 포트를 통해 EtherCAT 마스터나 혹은 EtherCAT 슬레이브와 통신할 수 있고, PDI Contro, PDI Configuration, Pulse Length SyncSignals, Extended PDI Configuration, Configured Station Alias의 ET1100의 초기화에 필요한 다양한 정보를 가진 EEPROM을 통해 통신방식이 결정되는 EtherCAT 통신 모듈;을 포함하여 구성되는 것을 특징으로 하는 로봇용 다자유도 스마트 액추에이터를 위한 네트워크 기반 제어 모듈을 제공한다.

본 발명에 따르면, 인간 친화적인 로봇 구현의 일환으로 보다 정교한 모션 구현과 안정성을 확보할 수 있고, 여러 제어기를 통합 제어 가능하며, 실시간 고성능 정밀 제어가 가능한 효과를 얻을 수 있다.

대 표 도 - 도1



(72) 발명자

최상방

인천광역시 남구 용현3동 인하대학교 전자공학과 하이테크관 904호

정진하

강원도 춘천시 퇴계동 주공1차아파트 105동 602호

윤완오

인천광역시 남동구 만수동 벽산아파트 116동 401호

장문석

경기도 부천시 원미구 상동 반달마을 삼익아파트 1831동 901호

김지원

인천광역시 남구 용현3동 인하대학교 하이테크 913호

김진혁

인천광역시 남구 용현3동 인하대학교 하이테크 913호

윤진원

인천광역시 남구 용현3동 인하대학교 913호

송인성

인천광역시 남구 용현3동 인하대학교 913호

임현수

인천광역시 남구 용현3동 인하대학교 913호

김병조

인천광역시 남구 용현3동 인하대학교 913호

유병철

인천광역시 남구 용현3동 인하대학교 913호

특허청구의 범위

청구항 1

EtherCAT Configuration Tool을 이용하여 EtherCAT 마스터 디바이스 환경 설정에 필요한 내용과 초기화 명령, 실행 동작에 대해 정의하여 XML 파일을 생성하고, 생성된 XML 파일은 각 구문분석 작업을 통해서 EtherCAT Master Driver에서 동작하며, 사용자 인터페이스(UI)는 Application을 통해 동작하고, EtherCAT 마스터 드라이버는 내부적으로 NPF Protocol Driver와 TCP/IP Protocol Driver를 통해 MiniPort에서 입/출력된 데이터를 분석 및 생성하도록 구성된 EtherCAT 마스터와;

EtherCAT 컨트롤러(ET1100)와 통합 보드의 메인 컨트롤러인 TMS320F20335에 버스 인터페이스 방식으로 구현되고, 16비트의 데이터 버스와 16비트의 어드레스 버스로 인터페이스되며, 메인 컨트롤러의 0x100000-0x10FFFF 영역에 메모리를 할당하도록 설계되고, RJ45 포트를 통해 이더넷 연결되도록 구성된 EtherCAT 슬레이브와;

RJ45 포트를 통해 EtherCAT 마스터나 혹은 EtherCAT 슬레이브와 통신할 수 있고, PDI Contro, PDI Configuration, Pulse Length SyncSignals, Extended PDI Configuration, Configured Station Alias의 ET1100의 초기화에 필요한 다양한 정보를 가진 EEPROM을 통해 통신방식이 결정되는 EtherCAT 통신 모듈;을 포함하여 구성되는 것을 특징으로 하는 로봇용 다자유도 스마트 액추에이터를 위한 네트워크 기반 제어 모듈.

청구항 2

청구항 1에 있어서;

상기 EtherCAT 마스터의 XML 파일에는, 마스터 디바이스와 이 마스터 디바이스에 연결된 모든 슬레이브 디바이스들에 대한 일반적인 정보 및 EtherCAT 스테이트 머신에 따라 수행되어야 할 EtherCAT 초기화 명령이 담겨지고, 설정 툴을 통해 오프라인으로 생성되거나 혹은 EtherCAT 마스터 디바이스를 통해 온라인으로 로드될 수 있는 것을 특징으로 하는 로봇용 다자유도 스마트 액추에이터를 위한 네트워크 기반 제어 모듈.

청구항 3

청구항 1에 있어서;

상기 EtherCAT 마스터에서 EtherCAT 슬레이브를 동작시키기 위해 전송하는 데이터 패킷은 바이너리 코드로 구성되되, 상기 바이너리 코드는 Ethernet Header, EtherCAT Frame Header, EtherCAT Datagrams, Data, 1st Working Count, EtherCAT Datagrams, Data, 2st Working Count, Pad bytes로 이루어진 것을 특징으로 하는 로봇용 다자유도 스마트 액추에이터를 위한 네트워크 기반 제어 모듈.

청구항 4

청구항 1에 있어서;

상기 EtherCAT 마스터에서 EtherCAT 슬레이브로 데이터를 전송하기 위한 마스터와 슬레이브간 통신 프로토콜은, 마스터에서 슬레이브에게 데이터를 쓰는 과정과, 마스터에서 슬레이브의 정보를 읽어오는 과정으로 이루어지되,

상기 데이터를 쓰는 과정은 마스터가 슬레이브에 연결된 모터의 동작에 대한 명령과 모터의 기능을 설정하는 명령으로 나누어지며, 마스터가 EtherCAT 데이터 패킷을 전송하면, 슬레이브 컨트롤러(ET1100) SRAM의 0x1000-0x100F 영역에 데이터를 저장하고, 슬레이브 메인 컨트롤러(TMS320F28335)는 이 영역을 읽어 명령에 맞는 동작을 수행하며;

상기 데이터를 읽는 과정은 마스터가 슬레이브에 연결된 모터의 상태를 확인하기 위한 명령으로 이루어지며, 슬레이브 메인 컨트롤러는 현재의 모터 상태를 슬레이브 컨트롤러 SRAM의 0x2000-0x200F 영역에 데이터를 저장하면, 마스터의 데이터 읽기 명령을 요청하였을 때 슬레이브 컨트롤러는 EtherCAT 데이터 패킷을 생성하여 마스터에게 전송하도록 구성된 것을 특징으로 하는 로봇용 다자유도 스마트 액추에이터를 위한 네트워크 기반 제어 모듈.

청구항 5

청구항 4에 있어서;

상기 상기 EtherCAT 마스터에서 EtherCAT 슬레이브로 데이터를 전송하기 위한 마스터와 슬레이브간 통신 프로토콜은, 데이터를 읽거나 쓰기 위한 메모리 영역이 명령, 파라메터0, 파라메터1, 파라메터2로 이루어진 기본 4바이트 단위로 구성된 것을 특징으로 하는 로봇용 다자유도 스마트 액추에이터를 위한 네트워크 기반 제어 모듈.

명 세 서

기 술 분 야

[0001] 본 발명은 다자유도 스마트 액추에이터의 모터 제어가 실시간 보장되면서 많은 데이터를 처리할 수 있도록 개방형 표준 이더넷 호환성을 확보한 산업용 이더넷 프로토콜인 이더캣(EtherCAT)을 기반으로 하여 슬레이브 제어 모듈을 구현하고, 이를 통해 정교하고 안정된 액추에어터 제어 모듈을 구현할 수 있도록 한 로봇용 다자유도 스마트 액추에이터를 위한 네트워크 기반 제어모듈에 관한 것이다.

배 경 기 술

[0002] 로봇산업은 인간의 지령에 의해서 수동적, 반복적 단순작업을 수행하던 기존의 산업용 로봇에서 이제는 지능형 로봇 시대로 돌입하고 있다.

[0003] 2족 보행 로봇인 아시모와 휴보, 애완견 로봇 아이보 등과 같이 인간 친화적인 로봇이 실생활에 밀접하게 다가오고 있으며, 향후에 보다 정교한 모션 구현과 안정성을 확보하기 위해서는 다자유도 모션이 가능한 액추에이터의 개발 및 이를 제어하기 위힌 모듈이 필수적으로 요구된다.

[0004] 하지만, 아직까지 로봇의 네트워크에서 제어의 실시간이 보장되면서 많은 데이터를 처리할 수 있는 개방형 표준 이더넷 호환성을 확보한 산업용 이더넷 프로토콜인 이더캣(EtherCAT)을 기반으로 하여 지능형 로봇에 적용할 수 있도록 로봇의 팔이나 주행용 휠에 모듈 단위로 실장할 수 있는 제어 모듈이 개발된 바 없다.

[0005] EtherCAT은 필드버스와 비슷하면서도 토폴로지에 대한 유연성과 환경설정 및 취급방법이 간단한 측성이 있다.

[0006] 특히, EtherCAT은 컴퓨터의 이더넷 포트를 그대로 사용할 수 있어 비용 때문에 필드버스 네트워크를 선택할 수 없었던 기존의 어플리케이션에서도 EtherCAT을 필드버스로 적용할 수 있는 장점이 있다.

[0007] 이러한 EtherCAT은 모든 디바이스에서 개별적인 데이터를 해석, 처리하고 복사할 수 있으며, 이더넷 패킷을 받지 않아 기존의 이더넷 솔루션이 가지고 있었던 한계를 극복하고 있다.

[0008] 또한, EtherCAT 슬레이브 디바이스는 프레임이 노드를 통과하는 동안 그곳에 전달된 데이터를 읽는다. 그러므로 프레임이 통과하는 동안에 입력한 데이터가 삽입되며, 각 노드에서의 프레임은 겨우 몇 ns 동안만 지연된다.

[0009] 이로 인해, 이더넷 프레임이 수신방향과 발신방향 모두에서 많은 디바이스의 데이터를 포함하면서 사용가능한 데이터 속도는 90% 이상으로 증가한다. 즉, 100 Base TX의 full duplex 특성을 충분히 활용하여 100Mbps 보다 효과적인 데이터 속도를 얻을 수 있다.

[0010] 따라서, 이와 같은 기술을 가지는 EtherCAT을 이용하여 다자유도 스마트 액추에이터의 제어 모듈을 구현할 수 있을 것으로 기대된다.

발명의 내용

해결하려는 과제

[0011] 본 발명은 상술한 바와 같은 개방형 표준 이더넷 호환성을 확보한 산업용 이더넷 프로토콜인 EtherCAT을 기반으로 하여 로봇 관절의 다자유도 구현을 위한 스마트 액추에이터를 구동할 수 있도록 한 네트워크 기반의 제어 모듈을 제공하기 위한 것으로, 고속의 실시간 모션 제어 네트워크인 EtherCAT을 다자유도 스마트 액추에이터의 제어에 적용하여 여러 제어기를 제어할 수 있는 통합 제어 프로그램을 구현하고, 이를 통해 별도의 마스터 제어보

드 없이도 이더넷을 이용하여 제어할 수 있도록 하며, 지능형 로봇에 적용할 수 있도록 로봇의 팔이나 주행용 휠에 모듈 단위로 실장할 수 있도록 한 로봇용 다자유도 스마트 액추에이터를 위한 네트워크 기반 제어 모듈을 제공함에 그 주된 목적이 있다.

과제의 해결 수단

[0012] 본 발명은 상기한 목적을 달성하기 위한 수단으로, EtherCAT Configuration Tool을 이용하여 EtherCAT 마스터 디바이스 환경 설정에 필요한 내용과 초기화 명령, 실행 동작에 대해 정의하여 XML 파일을 생성하고, 생성된 XML 파일은 각 구문분석 작업을 통해서 EtherCAT Master Driver에서 동작하며, 사용자 인터페이스(UI)는 Application을 통해 동작하고, EtherCAT 마스터 드라이버는 내부적으로 NPF Protocol Driver와 TCP/IP Protocol Driver를 통해 MiniPort에서 입/출력된 데이터를 분석 및 생성하도록 구성된 EtherCAT 마스터와; EtherCAT 컨트롤러(ET1100)와 통합 보드의 메인 컨트롤러인 TMS320F20335에 버스 인터페이스 방식으로 구현되고, 16비트의 데이터 버스와 16비트의 어드레스 버스로 인터페이스되며, 메인 컨트롤러의 0x100000-0x10FFFF 영역에 메모리를 할당하도록 설계되고, RJ45 포트를 통해 이더넷 연결되도록 구성된 EtherCAT 슬레이브와; RJ45 포트를 통해 EtherCAT 마스터나 혹은 EtherCAT 슬레이브와 통신할 수 있고, PDI Contro, PDI Configuration, Pulse Length SyncSignals, Extended PDI Configuration, Configured Station Alias의 ET1100의 초기화에 필요한 다양한 정보를 가진 EEPROM을 통해 통신방식이 결정되는 EtherCAT 통신 모듈;을 포함하여 구성되는 것을 특징으로 하는 로봇용 다자유도 스마트 액추에이터를 위한 네트워크 기반 제어 모듈을 제공한다.

[0013] 이때, 상기 EtherCAT 마스터의 XML 파일에는, 마스터 디바이스와 이 마스터 디바이스에 연결된 모든 슬레이브 디바이스들에 대한 일반적인 정보 및 EtherCAT 스테이트 머신에 따라 수행되어야 할 EtherCAT 초기화 명령이 담겨지고, 설정 툴을 통해 오프라인으로 생성되거나 혹은 EtherCAT 마스터 디바이스를 통해 온라인으로 로드될 수 있는 것에도 그 특징이 있다.

[0014] 또한, 상기 EtherCAT 마스터에서 EtherCAT 슬레이브를 동작시키기 위해 전송하는 데이터 패킷은 바이너리 코드로 구성되되, 상기 바이너리 코드는 Ethernet Header, EtherCAT Frame Header, EtherCAT Datagrams, Data, 1st Working Count, EtherCAT Datagrams, Data, 2st Working Count, Pad bytes로 이루어진 것에도 그 특징이 있다.

[0015] 뿐만 아니라, 상기 EtherCAT 마스터에서 EtherCAT 슬레이브로 데이터를 전송하기 위한 마스터와 슬레이브간 통신 프로토콜은, 마스터에서 슬레이브에게 데이터를 쓰는 과정과, 마스터에서 슬레이브의 정보를 읽어오는 과정으로 이루어지되, 상기 데이터를 쓰는 과정은 마스터가 슬레이브에 연결된 모터의 동작에 대한 명령과 모터의 기능을 설정하는 명령으로 나누어지며, 마스터가 EtherCAT 데이터 패킷을 전송하면, 슬레이브 컨트롤러(ET1100) SRAM의 0x1000-0x100F 영역에 데이터를 저장하고, 슬레이브 메인 컨트롤러(TMS320F28335)는 이 영역을 읽어 명령에 맞는 동작을 수행하며; 상기 데이터를 읽는 과정은 마스터가 슬레이브에 연결된 모터의 상태를 확인하기 위한 명령으로 이루어지며, 슬레이브 메인 컨트롤러는 현재의 모터 상태를 슬레이브 컨트롤러 SRAM의 0x2000-0x200F 영역에 데이터를 저장하면, 마스터의 데이터 읽기 명령을 요청하였을 때 슬레이브 컨트롤러는 EtherCAT 데이터 패킷을 생성하여 마스터에게 전송하도록 구성된 것에도 그 특징이 있다.

[0016] 아울러, 상기 상기 EtherCAT 마스터에서 EtherCAT 슬레이브로 데이터를 전송하기 위한 마스터와 슬레이브간 통신 프로토콜은, 데이터를 읽거나 쓰기 위한 메모리 영역이 명령, 파라메터0, 파라메터1, 파라메터2로 이루어진 기본 4바이트 단위로 구성된 것에도 그 특징이 있다.

발명의 효과

[0017] 본 발명에 따르면, 인간 친화적인 로봇 구현의 일환으로 보다 정교한 모션 구현과 안정성을 확보할 수 있고, 여러 제어기를 통합 제어 가능하며, 실시간 고성능 정밀 제어가 가능한 효과를 얻을 수 있다.

도면의 간단한 설명

[0018] 도 1은 본 발명에 따른 EtherCAT 마스터 프로그램의 구조를 보인 구성블럭도이다.

도 2는 본 발명에 따른 다자유도 스마트 액추에이트 사용자 프로그램을 보인 예시 화면이다.

도 3은 본 발명에 따른 EtherCAT 마스터 데이터 패킷의 예시적인 캡쳐 화면이다.

도 4a 내지 도 4c는 본 발명에 따른 통합보드에 적용하기 위한 EtherCAT 슬레이브의 회로 설계예를 보인 것으로, 도 4a는 EtherCAT 슬레이브 컨트롤러 인터페이스를 보인 예시적인 회로도이며, 도 4b는 PHY Port 0을 보인 회로도이고, 도 4c는 PHY Port 1을 보인 회로도이다.

도 5는 본 발명에 따른 통합 보드에서의 슬레이브를 보인 샘플 사진이다.

도 6 내지 도 7은 본 발명에 따른 SPI 인터페이스를 이용한 EtherCAT 슬레이브의 펌웨어 개발예를 보인 개념도이다.

도 8은 본 발명에 따른 EEPROM의 정보와 EtherCAT을 적용하기 위한 예시적인 구성블럭도이다.

발명을 실시하기 위한 구체적인 내용

[0019] 이하에서는, 첨부도면을 참고하여 본 발명에 따른 바람직한 실시예를 보다 상세하게 설명하기로 한다.

[0020] 본 발명은 다자유도 스마트 액추에이터 모터 제어를 위한 EtherCAT 마스터와 슬레이브 구현을 하기 위해 EtherCAT의 SPEC과 프로토콜을 명확하게 분석하였다.

[0021] 그리고, 마스터 구현을 위해서 XML 파일의 기본 구조를 분석, 적용하였으며, 사용자 영역의 통신 프로토콜을 설계 구현하였다.

[0022] 사용자 영역의 통신 프로토콜을 다자유도 모터를 제어하기 위한 명령, 모터의 상태를 확인하기 위한 명령, 모터를 설정하기 위한 명령으로 분류하여 구현하여 통신 패킷을 만들어 슬레이브에게 전달하도록 구현하였다.

[0023] 또한, 마스터 사용자 인터페이스 프로그램을 개발하여 구현하였으며, 개발된 마스터 UI 프로그램에서 전송되는 EtherCAT 패킷을 분석하였다.

[0024] 한편, 슬레이브 구현을 위해서 하드웨어적으로 통합 보드에 EtherCAT 슬레이브를 적용하기 위해 회로를 설계, 제작하였고, 소프트웨어적으로는 Evaluation Board의 연결된 PIC 18F452를 슬레이브 메인 컨트롤러로 사용하여 슬레이브 컨트롤러와의 SPI 버스를 이용하여 슬레이브 보드의 동작 확인을 검증을 한 후 슬레이브의 동작을 통합 보드에 적용하기 위해 현재 TMS320F28335를 메인 컨트롤러로 사용하여 슬레이브 컨트롤러와 버스 인터페이스 방식으로 구현하였다.

[0025] EtherCAT은 산업용 이더넷으로 표준 이더넷 프레임 사용하고, 이더넷의 모든 기술을 지원하며, 실시간성의 품질을 떨어뜨리지 않는 HTTP, FTP, TCP/IP 지원하고, 수직적 통합의 단순화하였다.

[0026] 그리고, 뛰어난 동기화 특성과 함께 사용할 수 있는 가장 빠른 시스템으로 토폴로지에 거의 의존하지 않는 성능을 나타내고, 기초가 되는 서브시스템이 필요 없으며, 현재와 미래의 요구사항을 만족시킨다.

[0027] 사용 측면에서의 EtherCAT은 주소의 수동설정이 없으며, 수위치 환경설정이 없고, 자동 환경설정 기능이 있으며, 정확한 로컬라이제이션 진단이 가능하다. 그리고 네트워크 토폴로지 구성에 있으서 라인, 트리, 스타 그리고 링과 같은 모든 토폴로지가 지원이 된다.

[0028] 이러한 토폴로지 구성에 있어서 노드, 스위치 또는 허브의 종속접속 문제가 없으며, 세그먼트마다 65535개까지의 노드가 있고, Redundancy, Hot Connect, Hot Swap과 같은 옵션을 설정할 수 있다.

[0029] 이더넷에서 EtherCAT의 안전성 측면에서는 하나의 네트워크를 사용하는 Safety 어플리케이션 및 표준 자동화를 구성할 수 있고, 국제안전표준 IEC 61508에 따라 개발한 프로토콜이고, Safety I/O 및 Safety 구동에 적합하고, Gateway 및 필드버스 시스템을 경유하여 경로설정이 가능하다.

[0030] EtherCAT의 개방에서 EtherCAT은 완전히 공개된 개방기술이고, 세계 최대의 이더넷 관련조직인 ETG의 지원을 받고 있다. 그리고 IEC 사양은 IEC/PS 62407이고, 잘 정립된 디바이스 프로파일을 지원한다.

[0031] 그리고, EtherCAT은 ETG를 통해 널리 지원되고, 여러 회사로부터 제공되는 컨트롤러 칩을 제공하며, 국제적 표준화 시대에 뒤떨어진 기술을 사용하지 않는다.

[0032] 따라서, EtheCAT은 다양하게 응용할 수 있는데 이것은 마스터와 슬레이브, 슬레이브 간 및 마스터 간 통신 지원, 프로세스 데이터와 파라미터 데이터를 위한 주기적/ 비주기적 서비스, 중앙집중 또는 분산 제어구조에 적합하고, 기계 제어, 로봇공학, 내장형 시스템, 빌딩 자동화, 운송 시스템 등에 응용할 수 있다.

[0033] 비용적인 측면에서의 EtherCAT은 필드버스의 가격수준 대비 훨씬 낮고, 특별한 마스터 카드가 필요하지 않으며, 온 보드 MAC 또는 저가 표준 NIC 사용하고, 낮은 인터페이스 비용으로 고도로 통합된 슬레이브 컨트롤러 기반구조(인프라) 구성요소가 필요 없다.

[0034] 그리고, EtherCAT은 2003년 이래 어플리케이션 시리즈로 계속해서 발달해 오고 있기 때문에 이미 수천개의 노드를 출하하여 다양한 컨트롤러와 운영체제상에서 구현되므로 제품 선택 범위가 넓은 장점을 가지고 있다.

[0035] 이와 같이, EtherCAT은 공지된 것이므로 그 자체적인 특징이나 프로토콜, FMMU(Fieldbus Memory Management Units) 팹핑, 싱크매니저의 통신, 모니터링, 클럭동기화, 오프셋, 드리프트 등에 대한 구체적인 설명은 생략하기로 한다.

[0036] 본 발명은 도 1에 도시된 바와 같은 형태로 EtherCAT의 마스터가 구현될 수 있다.

[0037] EtherCAT 마스터는 도시된 바와 같이, EtherCAT Configuration Tool(100)을 이용하여 EtherCAT 마스터 디바이스의 환경 설정에 필요한 내용과 초기화 명령, 실행 동작 등에 대한 정의를 통해 XML 파일(110)을 생성한다.

[0038] 그리고, 생성된 XML 파일(110)은 각 구문분석 작업을 거쳐 EtherCAT 마스터 드라이버(120)에서 동작하고, 유저 인터페이스를 통해 동작하게 된다.

[0039] 또한, EtherCAT 마스터 드라이버(120)는 내부적으로 NPF 프로토콜 드라이버(130)와 TCP/IP 프로토콜 드라이버(140)를 통해 MiniPort(150)에서 입/출력된 데이터를 분석 및 생성하는 과정을 거치게 된다.

[0040] 이때, EtherCat XML 설정 파일에는 EtherCAT 마스터 디바이스 환경 설정에 필요한 내용들이 담겨있다. 이 파일에는 마스터 디바이스와 그 마스터 디바이스에 연결된 모든 슬레이브 디바이스들에 대한 일반적인 정보가 담겨있다.

[0041] 또한, 파일에는 특정 과정에서 EtherCAT 스테이트 머신에 따라 수행되어야 할 EtherCAT 초기화 명령어들이 담겨있다. 이 파일은 설정 툴(오프라인)을 이용해 생성하거나 EtherCAT 마스터 디바이스(온라인)를 사용하여 로드할 수 있다.

[0042] 예컨대, 상기 XML 파일의 기본 구조는 다음과 같을 수 있다.

표 1

태그	사건 발생 (기본 1)	설명
<EtherCATConfig>		루트
<Config>		
<Master>...</Master>		EtherCAT 마스터 디바이스에 관한 일반적인 정보. 예를 들어, 디바이스의 MAC 주소, 모든 슬레이브들이 사용하게 되는 시작 명령어, 등.
<Slave>...</Slave>	1 혹은 이상	각 EtherCAT 슬레이브 디바이스에 관한 정보. 예를 들어, 디바이스의 이름, 물리 주소, 초기화 명령어 등.
<Cyclic>...</Cyclic>	0 or 1 (옵션)	이 부분에는 슬레이브로 주기적으로 전송되어야 하는 명령어들이 있다. 데이터 통신을 위한 명령어도 이 부분에 포함된다.
<ProcessImage> ...	0 or 1 (옵션)	프로세스 입려과 출력의 레이아웃을 정의한다.
</Config>		
</EtherCATConfig>		루트

[0043]

[0044] 이때, 상기 표 1에 예시된 각 태그에 대하여 더 구체적으로 설명할 필요는 없으나, Master 태그는 EtherCAT 마스터 디바이스와 모든 슬레이브 디바이스에 적용되는 설정이 포함되어 있고, Slave 태그에는 각 EtherCAT 슬레이브 디바이스 설정에 관한 내용이 포함되어 있으며, Cyclic 태그에는 마스터가 주기적으로 보내야 하는 EtherCAT 명령어, 프로세스 데이터 통신을 위한 명령어 등이 포함된다.

[0045] 또한, ProcessImage 태그는 입력, 출력 프로세스 이미지의 레이아웃에 관한 내용을 포함하고 있다.

[0046] 한편, 마스터에서 슬레이브에게 데이터를 전송하기 위해서는 마스터와 슬레이브 간의 통신 프로토콜을 정해야 한다.

[0047] 본 발명에 따른 통신 프로토콜은 마스터에서 슬레이브에게 데이터를 쓰는 과정과 마스터에서 슬레이브의 정보를 읽어 오는 과정으로 나누어진다.

[0048] 먼저, 데이터를 쓰는 과정은 마스터가 슬레이브에 연결된 모터의 동작에 대한 명령과 모터의 기능을 설정하는 명령으로 나누어지며, 마스터가 EtherCAT 데이터 패킷을 전송하면, 슬레이브 컨트롤러(ET1100) SRAM의 0x1000-0x100F 영역에 데이터를 저장하고, 슬레이브 메인 컨트롤러(TMS320F28335)는 이 영역을 읽어 명령에 맞는 동작을 수행한다.

[0049] 그리고, 데이터를 읽는 과정은 마스터가 슬레이브에 연결된 모터의 상태를 확인하기 위한 명령으로 이루어지며, 슬레이브 메인 컨트롤러는 현재의 모터 상태를 슬레이브 컨트롤러 SRAM의 0x2000-0x200F 영역에 데이터를 저장하면, 마스터의 데이터 읽기 명령을 요청하였을 때 슬레이브 컨트롤러는 EtherCAT 데이터 패킷을 생성하여 마스터에게 전송하게 된다.

[0050] 이와 같이 데이터를 읽거나 쓰기 위한 메모리 영역의 구성은 기본 4바이트 단위로 구성이되며, 명령, 파라메터 0, 파라메터1, 파라메터 2로 구성된다.

[0051] 또한, 각 명령에 대한 의미와 파라메터 값에 대한 정의는 표 2, 3, 4에 나타내었다.

[0052] 예컨대, 표 2는 마스터에서 슬레이브의 모터를 제어하기 위한 명령을 나타내며, 표 3은 슬레이브 모터의 상태를 나타내는 명령을 나타내고, 표 4는 슬레이브 모터를 설정 또는 설정 변경을 나타내는 명령이다.

표 2

Address	0x1000	0x1004	0x1008	0x100C
Data / 명령	CMD	PARA0	PARA1	PARA2
원점수행	0x100	0	0	0
비상정지	0x101	0	0	0
Servo On	0x102	0	0	0
Servo Off	0x103	0	0	0
Jog 속도 변경	0x104	0-100	0-100	0-100
운전속도 변경	0x105	0-100	0-100	0-100
절대위치이동	0x290	0-100	0-100	0-100
상대위치이동	0x291	0-100	0-100	0-100
절대위치이동 저장	0x120	0-100	0-100	0-100
상대위치이동 저장	0x121	0-100	0-100	0-100
저장프로그램 수행	0x130	0	0	0
프로그램 수행 정지	0x131	0	0	0
1축+JOG 이동	0x141	0	0	0
1축-JOG 이동	0x151	0	0	0
2축 +JOG 이동	0x142	0	0	0
2축 -JOG 이동	0x152	0	0	0
3축 +JOG 이동	0x143	0	0	0
3축 -JOG 이동	0x153	0	0	0
JOG 정지	0x150	0	0	0

[0053]

표 3

Address	0x2000	0x2004	0x2008	0x200C
Data / 명령	CMD	PARA0	PARA1	PARA2
상태(현재위치)	0x1004	현재위치0	현재위치1	현재위치2
상태(현재전류)	0x1005	현재전류0	현재전류1	현재전류2
상태(위치에러)	0x1007	위치에러0	위치에러1	위치에러2
상태	0x1006	SERVOON, ORIGIN, RUN, ALARM		

[0054]

표 4

Address	0x1000	0x1004	0x1008	0x100C
Data / 명령	CMD	PARA0	PARA1	PARA2
가감속도 시간 설정	0x112	0-1	0-1	0-1
감속비 설정	0x113	1-100	1-100	1-100
모터 최고 RPM 설정	0x114	1-100	1-100	1-100
모터 회전방향 설정	0x115	+/- 1	+/- 1	+/- 1
엔코더 분해능 설정	0x116	0-100	0-100	0-100
모터 위치 비례게인 변경	0x200	0-100	0-100	0-100
모터 위치 적분게인 변경	0x201	0-100	0-100	0-100
모터 속도 비례게인 변경	0x202	0-100	0-100	0-100
모터 속도 적분게인 변경	0x203	0-100	0-100	0-100
모터 전류 비례게인 변경	0x204	0-100	0-100	0-100
모터 전류 적분게인 변경	0x205	0-100	0-100	0-100

[0055]

[0056] 한편, 본 발명에 따른 EtherCAT 마스터 UI 프로그램은 도 2와 같이 예시할 수 있다.

[0057] 도 2에 따르면, 본 발명에 따른 EtherCAT 마스터 UI 프로그램은 MicroSoft Visual Studio 2005으로 개발되었으며, EtherCAT 마스터를 제어하기 위한 부분과 모터를 제어하기 위한 부분으로 나뉜다.

[0058] 예컨대, 도 2의 (a)는 Main 화면으로 왼쪽 부분은 표 2에서 사용한 모터 제어 명령에 대해 나타내며, 오른쪽 상단은 EtherCAT 마스터로 동작하기 위한 메뉴로 구성되고, 오른쪽 하단은 표 3에서 사용한 슬레이브 모터 상태를 나타낸다.

[0059] 또한, 도 2의 (b)는 Main 화면에서 SUB MENU를 선택하였을 때 실행되는 화면으로, 표 4에서 사용한 슬레이브 모터를 설정 또는 설정 변경하는 명령을 실행한다.

[0060] 아울러, EtherCAT의 마스터에서 전송되는 데이터 패킷을 슬레이브에 전달하여 올바른 동작을 하기 위해서는 전송되는 데이터 패킷을 분석하여야 한다.

[0061] 도 3은 WireShark 프로그램을 이용하여 마스터에서 슬레이브로 전송하는 데이터 패킷을 캡쳐한 것이다.

[0062] 이때, 도 3의 상단은 데이터 패킷을 EtherCAT 데이터 그램의 각 Field 의미를 나타내고, 하단은 데이터 패킷을 바이너리 코드로 표현을 한 것이다.

[0063] 바이너리 코드의 구성은 Little Endian으로 표현되어 있다.

[0064] 이것은 각 필드의 데이터가 0x1234이면 34 12 형식으로 표현되며, 0x12345678이면 78 56 34 12 형식으로 표현된

다.

[0065] 그리고, 도 3의 패킷을 EtherCAT 데이터 그램의 각 Field와 패킷의 바이너리 데이터를 정리하면 표 5와 같이 나타낼 수 있다.

표 5

데이터 패킷 Field		바이너리 코드	크기
Ethenet Header	Destination	ff ff ff ff ff ff	48 bit
	Source	02 1e 37 ce d0 66	48 bit
	EtherType	88 a4	16 bit
EtherCAT Frame Header	Length+Rev +Type	20 10	11 + 1 + 4 bit
EtherCAT Datagrams	Command	0c	8 bit
	Index	01	8 bit
	Logical Addrs	00 00 01 00	32 bit
	Length+R+C+M	06 c0	11+3+1+1 bit
	IRQ	04 00	16 bit
Data		00 11 00 00 00 00	Length
1st Working Count	WKC	00 00	16bit
EtherCAT Datagrams	Command	07	8 bit
	Index	01	8 bit
	Slave Addrs	01 00	32 bit
	offset Addrs	30 01	11+3+1+1 bit
	Length+R+C+M	02 40	16 bit
	IRQ	04 00	Length
Data		01 00	16bit
2st Working Count	WKC	01 00	
Pad bytes		00 00 00 00 00 00 00 00 00 00 00 00	Padding Bytes

[0066]

[0067] 그리고, 본 발명에 따른 EtherCAT 슬레이브는 EtherCAT 슬레이브 보드를 하나의 통합 보드로 설계하기 위해 회로를 설계하고, 통합 보드를 구현하였다.

[0068] 통합 보드는 EtherCAT 컨트롤러(ET1100)와 통합 보드의 메인 컨트롤러인 TMS320F20335에 버스 인터페이스 방식을 사용하여 구현하였으며, 향후 이러한 통합 보드 환경에서 사용할 EtherCAT 슬레이브 펌웨어의 개발이 요구된다.

[0069] 통합 보드에 적용하기 위한 EtherCAT 슬레이브 회로 설계는 슬레이브 보드의 회로를 메인 컨트롤러에 버스 방식으로 인터페이스 하기 위한 회로 설계와 통신 포트의 크기를 줄이고, 단순화하기 위해 PHY0, PHY1의 포트를 통합하여 설계하였다.

[0070] 도 4a 내지 도 4c는 이러한 기능을 가지는 슬레이브 회로를 나타낸다.

[0071] 이때, 도 4a는 메인 컨트롤러와 슬레이브 컨트롤러와의 버스 인터페이스를 나타내고 있다.

[0072] 이것은 16비트의 데이터 버스와 16비트의 어드레스 버스로 인터페이스가 되며, 메인 컨트롤러의 0x100000-0x10FFFF 영역에 메모리를 할당하도록 설계가 되었다.

[0073] 그리고, 도 4b,c는 슬레이브와 마스터 간의 이더넷 연결을 위한 PHY 부분을 나타내고 있다.

[0074] 이더넷 연결을 위해서는 RJ45 포트를 사용하여야 하나, 본 발명에서는 RJ45 포트에서 사용하는 신호인 TX+, TX-, RX+, RX- 신호만 추출하여 1.25인치 크기의 컨넥터로 인터페이스 할 수 있게 설계하였다.

[0075] 아울러, 도 5는 이러한 회로 설계를 적용하여 통합 보드에 슬레이브를 적용시킨 것이다.

[0076] 여기에서, 통합 보드를 개발하기 이전에 EtherCAT의 슬레이브의 동작 및 BackHoff 사에서 제공하는 샘플 소스의 분석을 위해 본 발명에서는 Evaluation Board에서 제공하는 SPI 버스 방식의 슬레이브 동작을 구현하였다.

[0077] SPI 버스를 이용한 EtherCAT 슬레이브의 구현은 도 6에서와 같이 메인 컨트롤러를 MicroChip사의 PIC18F452를 사용하여 슬레이브 컨트롤러와 SPI 버스로 인터페이스를 하여 동작하도록 되어있다.

[0078] 뿐만 아니라, EtherCAT 슬레이브 동작 확인은 도 7에서와 같이 Evaluation Board에 연결되어 있는 LED를 출력 신호로 사용하고, DIP Switch를 입력 신호로 사용하여 마스터 테스트 프로그램에서 이를 확인 하는 과정을 거쳤다.

[0079] PIC18F452의 컴파일러를 사용하여 샘플 소스를 컴파일하기 위해 주의해야 할 점은 PIC 컴파일러의 버전 별로 호환이 되지 않기 때문에 샘플 소스를 컴파일하기 위해서는 반드시 HI-TECH사의 PICC-18 컴파일러의 8.x 버전으로 컴파일해야만 프로그램이 동작된다는 것이다.

[0080] SPI 버스를 이용한 슬레이브 동작 확인은 도 7 (a)의 마스터 테스트 프로그램에서 패킷의 바이너리 데이터를 비교 분석하여 확인하고, 도 7(b)에 연결되어 있는 LED와 DIP Switch 값의 변경으로 데이터 변화 유무를 확인하였다.

[0081] 마지막으로, 슬레이브 컨트롤러에 버스 인터페이스 방식으로 사용하기 위해서는 도 6과 같은 구조로 슬레이브를 구성해야 한다.

[0082] 특히, EEPROM에 저장되는 정보를 버스 인터페이스 정보에 맞게 슬레이브의 환경 정보를 변경하여 저장하여야 한다.

[0083] EEPROM에 저장되는 환경 정보는 PDI Control, PDI Configuration, Pulse Length SyncSignals, Extended PDI Configuration, Configured Station Alias 등이다.

[0084] 이것은 BackHoff 사에서 제공해주는 ET1100_Configuration_and_Pinout_V4.0 파일을 참고하여 슬레이브에 맞는 환경만 설정해주면 자동으로 생성할 수 있다.

[0085] 그리고, 도 8은 EtherCAT 슬레이브 펌웨어 개발 환경을 나타내고 있다.

[0086] 본 발명에서 슬레이브의 펌웨어 개발은 개발 환경을 단순화 하기 위해 통합 보드를 사용하지 않고, 메인 컨트롤러 모듈과 슬레이브 보드와의 메모리 인터페이스를 이용하여 개발될 수 있으며, 소프트웨어는 통합 환경인 Code Composer Studio Platinum V3.3을 사용하고 있다.

[0087] 즉, 슬레이브 펌웨어 개발은 슬레이브 컨트롤러의 제어에 있어 레지스터 세팅 및 전체 프로그램 작성에 디버깅 단계로서, 향후 지속적인 개발이 요구되는 사항이므로 여기서는 생략하기로 한다.

[0088] 이상에서와 같이, 본 발명은 EtherCAT 마스터와 슬레이브 구현을 하기 위해 EtherCAT의 SPEC과 프로토콜을 명확하게 분석함으로써 다자유도 스마트 액추에이터 모터 제어를 고정밀도로 정확하게 제어할 수 있음을 확인하였다.

[0089] 특히, 마스터 구현을 위해서 XML 파일의 기본 구조를 분석, 적용하였으며, 사용자 영역의 통신 프로토콜을 설계 구현할 수 있었다.

[0090] 그리고, 사용자 영역의 통신 프로토콜을 다자유도 모터를 제어하기 위한 명령, 모터의 상태를 확인하기 위한 명령, 모터를 설정하기 위한 명령으로 분류하여 구현하여 통신 패킷을 만들어 슬레이브에게 전달하도록 구현할 수 있었다.

[0091] 아울러, 슬레이브 구현을 위해서 하드웨어적으로 통합 보드에 EtherCAT 슬레이브를 적용하기 위해 회로를 설계, 제작하였고, 소프트웨어적으로는 Evaluation Board의 연결된 PIC 18F452를 슬레이브 메인 컨트롤러로 사용하여 슬레이브 컨트롤러와의 SPI 버스를 이용하여 슬레이브 보드의 동작 확인을 검증할 수 있었다.

부호의 설명

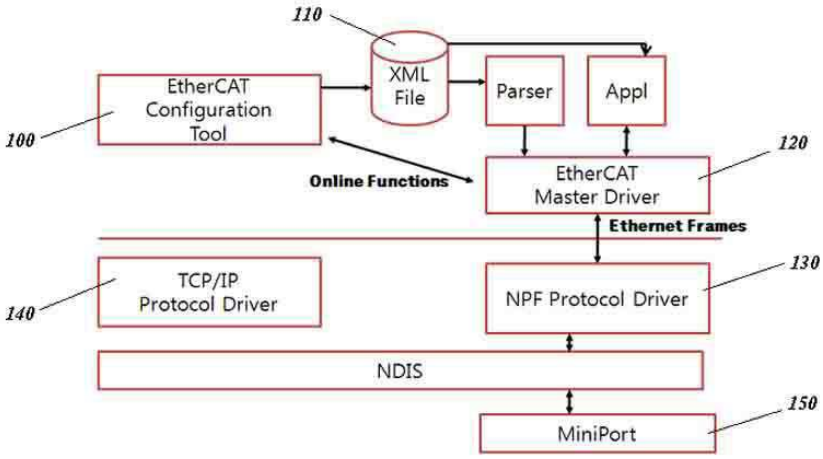
[0092] 100 : EtherCAT Configuration Tool　　110 : XML File

120 : EtherCAT Master Driver　　130 : NPF Protocol Driver

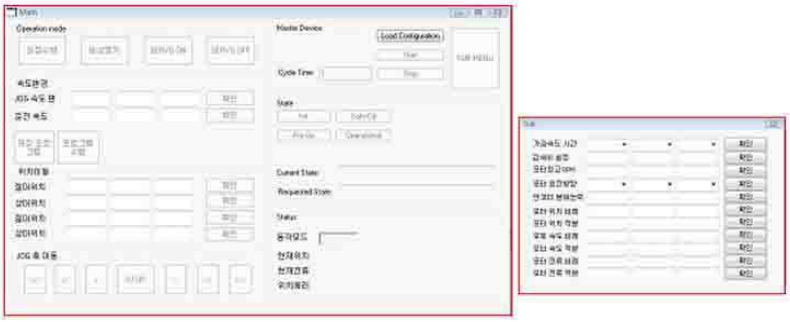
140 : TCP/IP Protocol　　150 : MiniPort

도면

도면1



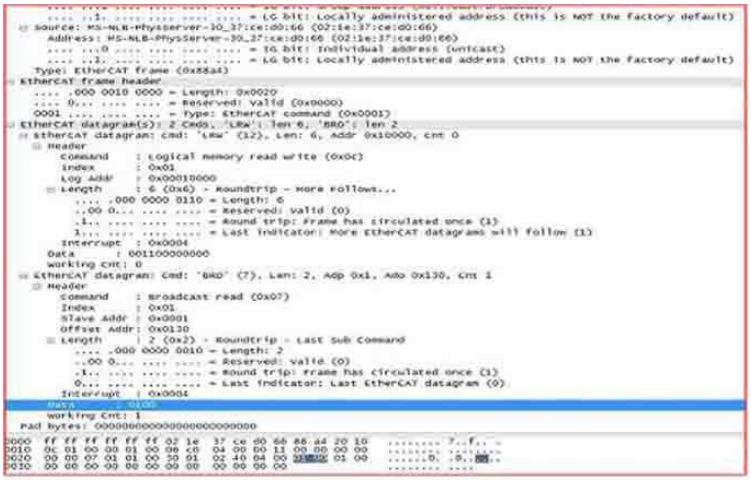
도면2



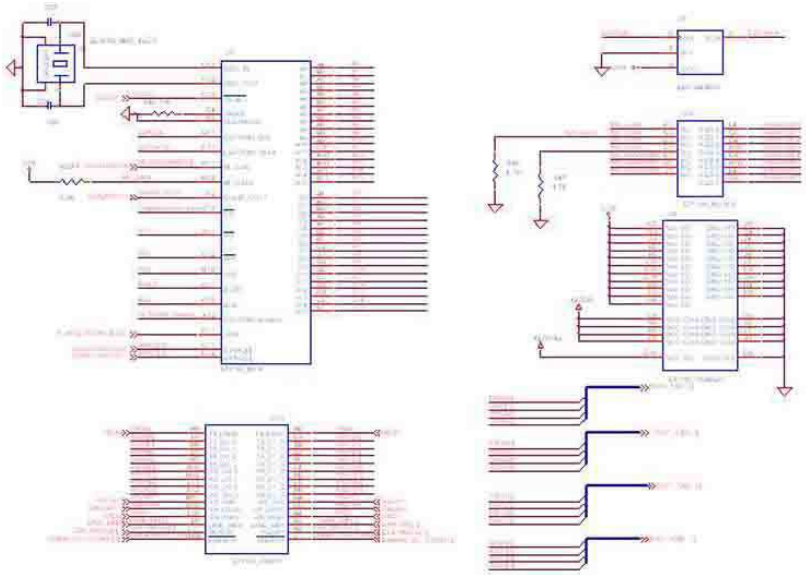
Main Program

SUB MENU

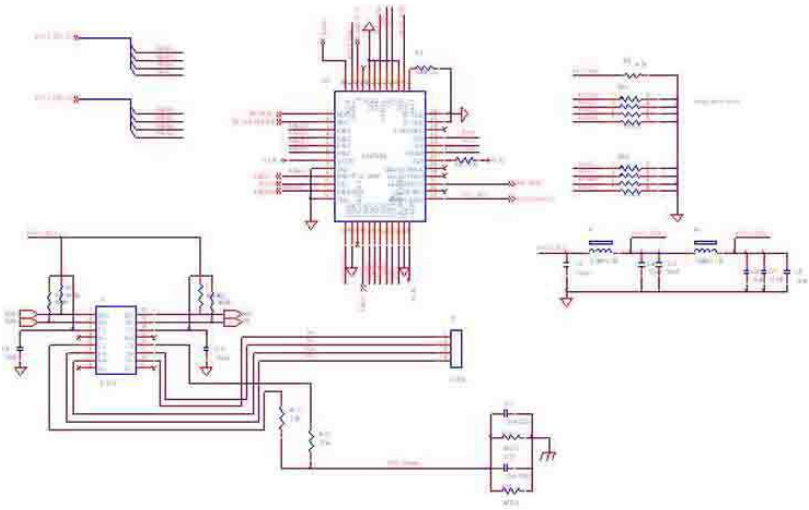
도면3



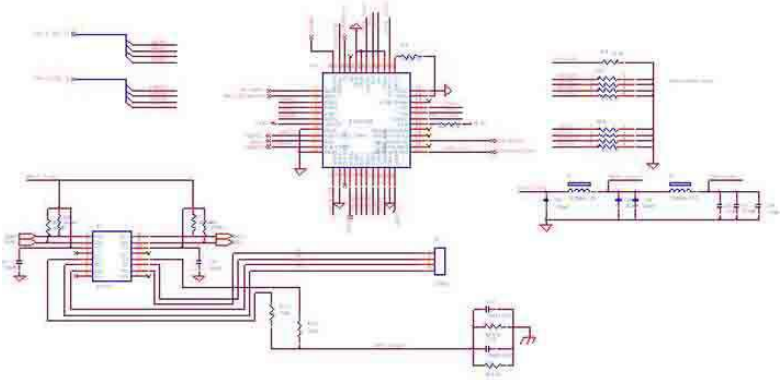
도면4a



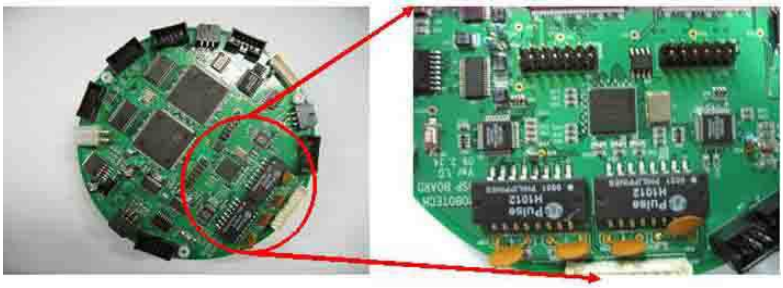
도면4b



도면4c



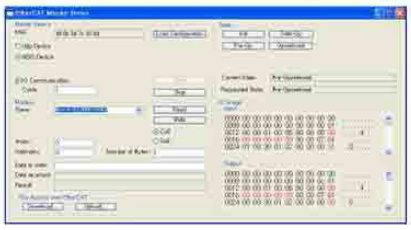
도면5



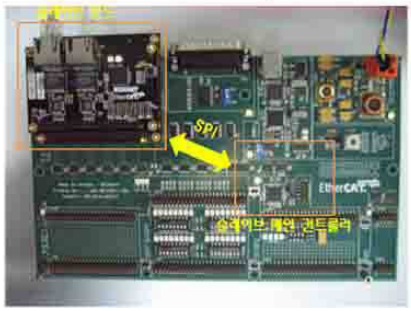
도면6



도면7



(a) Master Test Program



(b) SPI 인터페이스 실험환경



(c) SPI 인터페이스 펌웨어 컴파일

도면8


ESI EEPROM settings
Byte view (low byte to high byte)
Word view (low word to high word)
XML EEPROM ConfigData
Word
Value
Register description
Word Address
0
0x060A
PDI Control
0x0140
1
0xEE42
PDI Configuration
0x0150
2
0x000A
Pulse Length SyncSignals
0x0982
3
0x0B30
Extended PDI Configuration
0x0152
4
0x0000
Configured Station Alias
0x0010
5
0x0000
reserved
6
0x0000
reserved
7
0x0001
CRC
EEPROM
Slave FirmWare 개발 환경(통합 보드 이후)
BUS
TMS320F28335
ET1100
RJ45 Port