

윈도우 시스템에서의 IEEE 1394 버스지원 방법 비교

이창환, 김영호
부산대학교 전자계산학과

A Comparison of IEEE 1394 Handling in the Windows System

ChangHwan Lee, YoungHo Kim
Dept. of Computer Science, Pusan National Univ.

요 약

IEEE 1394는 디지털 가전기기와 전문 영상장비등의 디지털 데이터 전송을 일차적 목표로 하여 제시된 버스 표준이다. 디지털 데이터를 고속으로 전송할 수 있으며 확장연결 가능한 구조와 통신 제충진 연결구성이 가능한 특징을 지닌다. IEEE1394표준은 컴퓨터 뿐만 아니라 하드 디스크, 디지털 오디오와 비디오 장비들에 폭 넓게 채택되고 있다. 동시성(isochronous)인 특징을 가지고 있기에 시간 엄수(time critical)성 멀티미디어자료 연결에 유용하다. 1394용 호스트 어댑터는 adaptec사의 AHA-8920, AHA-8940등과 TI의 TSBKPCI PCILynx와 TSBKPCITST PCILynx등이 현재 제품화되어 있다. 본 논문은 이들 호스트 어댑터에서 개발자들에게 제공하는 1394관련 API들과 그 구조, 그리고 윈도우 차기 버전들에서 제공하는 API의 구조들에 대한 비교를 제시하였다.

1. 서 론

IEEE 1394 버스는 1986년 개인용 컴퓨터의 실시간 고속 데이터를 전송하기 위한 메커니즘을 제공하고 상호 연결할 수 있는 네트워크를 단순화시키기 위해 애플에서 FireWire라는 이름으로 개발되었다. 유용함이 입증된 후 1995년 12월 IEEE(Institute of Electrical and Electronics Engineers)에서 표준으로 제정되었다. FireWire표준은 다른 회사들에게 라이선스되었고, 1994년 1394구조를 사용하여 손쉬운 연결을 할 수 있는 컴퓨터와 가전기기를 개발하기 위해 관련회사들의 1394TA(Trade Association)가 만들어졌다.

IEEE 1394표준은 디지털 인터페이스이기 때문에 데이터 전송시 아날로그로의 변환없이 전송하므로 자료 손실을 막을 수 있다. 가는 케이블의 얇은 직렬버스가기 때문에 병렬버스에 비해서 활동성이 좋으며, SCSI 방식과는 달리 터미네이터의 짐작, 디바이스ID 할당, 추가적인 셋업 등이 필요하지 않다. 핫 플러그(Hot pluggable) 기능으로 작동중인 버스에 사용자 요구에 적절히 장치들을 설치하거나 해제할 수 있으며 버스에 있는 장치들이 98.304, 196.608, 그리고 393.216 Mbps의 속도를 낼 수 있다. 유연적인 연결성으로 데이터체인을 지원하며, 제충진 통신으로 확장

가능하다[1].

이 1394직렬버스는 그간 상호연결에 문제가 많았던 DVC(Digital Video Consortium) 장치, DVD(Digital Versatile Disk), MPEG 시스템, MIDI장치, PCI 시스템, 저장매체(storage media), 프린터, 스캐너등의 다른 멀티미디어 장비 연결에도 적용가능하다.

2. IEEE 1394 표준

2.1 IEEE 1394의 구조

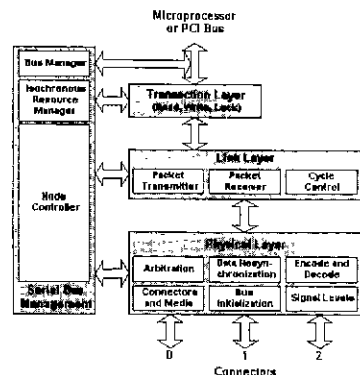


그림 1 1394 프로토콜 스택과 직렬버스 관리계층기

1394표준은 backplane과 cable 두가지 버스 범주로 사용가능하다. 16bit로 주소를 할당하기 때문에 한 시스템으로의 연결은 64K개 노드로 구성할 수 있으며 노드들간은 16 cable hop을 지원한다[2]. 총 16bit 중 노드ID는 6bit이기에 한 bridge에 63개 노드를 구성할 수 있고 버스ID는 10개의 비트로 구성되기 때문에 1,023개의 bridge들로 구성될 수 있다. 표준 구성상에는 디바이스당 27개의 포트를 제공할 수 있지만 일반적으로 3개의 포트들의 지원이 제품화되어 있다. 노드사이 cable 최대 길이는 4.5m이고 16개의 cable hop가 가능하므로 최대 72m까지 연결 가능하다. IEEE 1394는 그림 1에서와 같이 물리 층, 링크 층, transaction층의 3가지 층과 이 층들을 제어하는 버스 관리층이 있다.

2. 2 IEEE 1394용 호스트 어댑터

현재 제품화된 1394용 호스트 어댑터는 adaptec사의 3 포트용 AHA-8940, 1포트용 AHA-8920, SCSI 어댑터겸용인 AHA-8945와 TI(Texas Instrument)사의 TSBKPCI PCILynx와 TSBKPCITST PCILynx들이 있다. 그중 Adaptec사의 3포트용 AHA-8940 Developer's Kits를 비교 대상으로 한다.

3. IEEE 1394 API

3.1 개발자용 Adaptec의 API들

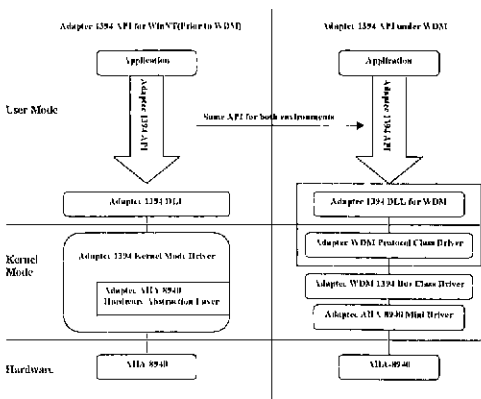


그림 2 윈도우에서 Adaptec사의 API구조

그림 2에서와 같이 Adaptec API구조는 AHA-8940위 커널 모드에 AHA-8940 하드웨어를 추상화하기 위한 하드웨어 추상화 계층(HAL: Hardware Abstraction Layer)이 존재하고 커널모드 드라이버가 상위층(upper layer)에서 adaptec이 제공하는 DLL을 통해 1394 API명령들을 받아들인다[3]. Windows95 OSR2 1버전부터 지원하는 WDM은 현재와 미래의 윈도우용 운영체제를 위해 I/O서비스의 공통세트와 2진 호환 장치 제어기를 제공한다. 이 WDM 1394버스 클래스드라이버에 WDM 프로토콜과 WDM용 DLL을 제공한다.

Adaptec에서 제공하는 API들은 개발자들에게 IEEE1394노드나 장치들을 직접 제어할 수 있는 응용프로그램을 용이하게 만들도록 한다. Wnpapi32.dll을 통하여 다음의 3가지 Win32 함수를 제공한다.

①DWORD GetPAPISupportInfo(BYTE *NumberOfAdaptors),

이 함수는 모든 API에 대해 처리를 해는 PAPI manager를

초기화하고 버스에 연결된 호스트 어댑터의 수에 대한 정보를 알려준다.

②DWORD SendPAPICommand(PPRB lpPRB);

이 함수는 IEEE 1394 I/O에 대한 요구를 처리한다. 이 함수는 하위 명령어로 나뉘어서 사용되는데 표 1에 나타나 있다

표 1 1394 입출력 API(Adaptec)

기능	구분	코드	설명
동시성명령		P_QUEUE_ISOC_출	동시성명령제정
		P_START_ISOC	시작
		P_STOP_ISOC	종료
비동기		P_EXEC_ASYNC_CMD	비동기명령
		P_GET_HA_INFO	호스트어댑터정보
호스트		P_GET_CAP_VERSIONS	호스트어댑터의 용량
		P_NOTIFY_ON_ACCESS	레보리의 접근이된
		P_ABORT_1394_CMD	명령중단
		P_GET_BUS_EVENT	버스의 이벤트확인
		P_GET_DEV_INFO	ROM구성정보

③DWORD BusConfig(PSBC lpSBC);

버스구성에 대한 요구를 처리하는 함수이다. 이 함수를 역시 하위명령어로 나뉘어서 처리되는데 표 2에 나타나 있다.

표 2. 1394 버스 구성API(Adaptec)

기능	구분	코드	설명
노드제어		P_GET_HANDLE	노드의 핸들신청
		P_RELEASE_HANDLE	노드의 핸들해제
대역폭제어		P_GET_NODEID	노드ID
		P_ALLOCATE_BANDWIDTH	대역폭할당
채널조각		P_DEALLOCATE_BANDWIDTH	대역폭해제
		P_ALLOCATE_CHANNEL	채널할당
부기기능		P_DEALLOCATE_CHANNEL	채널해제
		P_SET_ROOT	루트노드설정
		P_SET_GAPCOUNT	gap크기설정
		P_1394_BUS_RESET	버스재시작
		P_LINK_ON	링크초기화
공스피어명령		P_GET_SPEED_BTWN_NODES	노드간 속도
		P_LOCAL_HA_READ	읽기
DMA 제어		P_LOCAL_HA_WRITE	쓰기
		P_RESERVE_ISOC_RESOURCE	DMA 채널할당
메모리제어		P_RELEASE_ISOC_RESOURCE	DMA 채널해제
		P_MAP_1394_ADDRESS	메모리할당
		P_UNMAP_1394_ADDRESS	메모리 해제
		P_READ_MAP_CONTENTS	메모리읽기
		P_WRITE_MAP_CONTENTS	메모리쓰기
		P_UPDATE_MAP_CONTENTS	메모리갱신

3.2 윈도우에서의 1394지원구조

윈도우98과 윈도우NT 50에서 IEEE 1394 버스 드라이버가 지원된다. 그림 4에서와 같이 WDM stream class를 이용하여

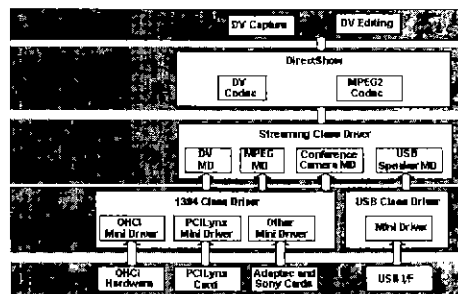


그림 3 과거 윈도우 시스템의 1394 구조 [4]

IEEE 1394 A/V장치들에 대한 저수준의 지원을 제공한다[5].

하드웨어 층에서 각각의 회사마다의 어댑터를 지원하고 버스 드라이버층에서는 OICTL_1394_CLASS에서 정의되어 있는 IoControlCode를 IRB(I/O Request Block)에 넣어 각각의 명령을 수행시킨다. 표 3에 해당 기능에 대한 함수들의 설명을 보았다.

표 3. 윈도우 1394 API

기능	구분	코드	설명
비동기 전송	메모리 할당	AllocateAddressRange	주소범위 할당
		FreeAddressRange	주소범위 해제
	명령어	AsyncRead	읽기
		AsyncWrite	쓰기
		AsyncLock	잠금
동시성 전송	대역폭	IsochAllocateBandwidth	대역폭 할당
		IsochFreeBandwidth	대역폭 해제
	채널	IsochAllocateChannel	채널 할당
		IsochFreeChannel	채널 해제
	자원	IsochAllocateResources	자원 할당
		IsochFreeResources	자원 해제
	버퍼	IsochAttachBuffers	버퍼 할당
		IsochDetachBuffers	버퍼 해제
	부가기능	IsochListen	주변 채널 확인
		IsochQueryCurrentCycleNumber	사이클 확인
		IsochQueryResources	대역폭과 채널 확인
		IsochSetChannelBandwidth	대역폭 확인
		IsochStop	전송 중지
부기명칭	AppendUnitDirectoryEntry	단위 디렉토리 엔트리 추가	
	DeleteUnitDirectoryEntry	단위 디렉토리 엔트리 삭제	
	GetLocalHostInformation	호스트 어댑터 정보	
	Get1394AddressFromDeviceObject	노드 주소 확인	
	Control	개발회사용 예약	
	GetMaxSpeedBetweenDevices	노드간의 속도	
	SetDeviceXmitProperties	속도와 우선순위	
	GetConfigurationInformation	구성 정보	
	BusReset	버스 초기화	
	GetGenerationCount	Generation 수 확인	
	SendPhyConfigurationPacket	phy 레지스터 전송	

3.3 API들간의 비교

기본적으로 제공되는 동시성(isochronous) 전송을 위한 메모리, 자원, 대역폭, 채널의 할당과 해제, 동시성 전송의 시작과 멈춤, 비동기 전송을 위한 읽기, 쓰기, 잠금, 그리고 호스트 어댑터에 대한 기브와 노드간의 속도, 디바이스에 대한 정보, 버스 재시작 등은 공통적으로 지원한다. 윈도우 API와는 달리 Adaptec의 API는 시스템에 다수의 호스트 어댑터를 지원하지 못한다. 표 4에서와 같이 윈도우 API는 노드간의 속도를 일대 다수간 속도, 대역폭확인, 명령어의 추가 지정, 버퍼를 관리, 대역폭 관리, 서비스의 투명성, 우선순위와 전송속도의 지정, 연결망에 대한 확인 작업등에 대한 지원을 제공한다. Adaptec의 API는 특정노드의 루트선택, 동시성전송에 필요한 gap의 크기, 버스에 생기는 이벤트확인, 동시성전송을 위한 버퍼를 지정하는 기능을 제공한다. Adaptec의 API들은 모든 API요구를 처리하는 응용인 PAPI Manager에 의해 수행된다. DLL로 제공되는 PAPI Manager는 커널드라이버를 통해 버스관리층, transaction층 그리고 연결층으로의 해당 명령을 수행하게 된다. 이 과정에 필요한 PRB(IEEE 1394 Request Block)와 SBC(Serial Bus Configuration Packet)자료 구조를 구조체로 정의하고 필요한 정보들의 매개변수들을 맵 버퍼번호로 정의해서 시스템 접근과 데이터 전송을 지원한다. 윈도우는 기본 전송모델인 WDM(Windows Driver Model)구조에 적용시키기 위해 IRB(I/O Request Block)을 이용하여

각 명령어를 구조체 내의 내부 공용체로 정의하여 MDL(Memory Descriptor List)에 대한 처리를 통해 전송에 필요한 버퍼들을 관리하고 시스템에서 제공하는 메모리에 대한 접근을 구체화시킨다.

표 4. API들의 비교

기능	종류	Windows API	Adaptec API
		비동기 전송	메모리 할당
비동기 전송	명령어	FreeAddressRange	P_UNMAP_1394_ADDRESS
		AsyncRead	P_EXEC_ASYNC_CMD
		AsyncWrite	P_EXEC_ASYNC_CMD
		AsyncLock	P_EXEC_ASYNC_CMD
		대역폭	IsochAllocateBandwidth
동시성 전송	대역폭	IsochFreeBandwidth	P_DEALLOCATE_BANDWIDTH
		채널	IsochAllocateChannel
	IsochFreeChannel	P_DEALLOCATE_CHANNEL	
	명령어	IsochStop	P_ISOC_STOP
		IsochTalk	P_ISOC_START
	자원	IsochAllocateResources	P_ALLOCATE_CHANNEL
		IsochFreeResources	P_DEALLOCATE_CHANNEL
부가기능	GetLocalHostInformation	P_GET_HA_INFO	
	GetConfigurationInformation	P_GET_HA_INFO	
부기명칭	동시성 전송	BusReset	P_1394_BUS_RESET
		IsochAttachBuffers	
	IsochDetachBuffers		
	IsochListen		
	IsochQueryResources		
	IsochSetChannelBandwidth		
	단위 디렉토리	AppendUnitDirectoryEntry	
	DeleteUnitDirectoryEntry		
	부가기능	Control	
		SetDeviceXmitProperties	

4. 결 론

본 논문에서는 다양한 기능들을 만족하기 위해서 IEEE 1394 표준의 응용프로그램 개발시 필요한 API들에 대해 비교하였다. 현재 1394호스트 어댑터의 개발사에서 제공하는 것과 차기 윈도우에서 지원하는 API들을 제공하는 기능과 구조적인 면에서 살펴보았다. 향후 기능적인 비교의 구체화를 위해 비교된 각 API들의 처리에 관련된 속도와 응용프로그램의 특정기능에 대한 API 호출절차 등을 비교 및 분석할 것이다.

참고문헌

- [1] Don Anderson, FireWire System Architecture, Mind share Inc, 1998
- [2] IEEE Std 1394-1995, IEEE Standard for a High Performance Serial Bus
- [3] AHA-8940 Developers Guide, Adaptec, 1997
- [4] IEEE 1394 Support in Windows98 and Windows NT 5.0, Microsoft, 1998
- [5] IEEE 1394 Bus Driver Design Notes and Reference, Microsoft, 1998