

(19)대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) 。 Int. Cl. ⁷ G06F 3/023	(45) 공고일자 (11) 등록번호 (24) 등록일자	2005년08월25일 10-0510048 2005년08월17일
--	-------------------------------------	--

(21) 출원번호	10-2002-7005138	(65) 공개번호	10-2002-0043232
(22) 출원일자	2002년04월22일	(43) 공개일자	2002년06월08일
번역문 제출일자	2002년04월22일		
(86) 국제출원번호	PCT/GB2000/004090	(87) 국제공개번호	WO 2001/31428
국제출원일자	2000년10월23일	국제공개일자	2001년05월03일

(81) 지정국

국내특허 : 아랍에미리트, 안티구와바부다, 알바니아, 아르메니아, 오스트리아, 오스트레일리아, 아제르바이잔, 보스니아 헤르체고비나, 바르바도스, 불가리아, 브라질, 벨라루스, 벨리제, 캐나다, 스위스, 중국, 코스타리카, 쿠바, 체코, 독일, 덴마크, 도미니카, 알제리, 에스토니아, 스페인, 핀란드, 영국, 그라나다, 그루지야, 가나, 감비아, 크로아티아, 헝가리, 인도네시아, 이스라엘, 인도, 아이슬란드, 일본, 케냐, 키르키즈스탄, 북한, 대한민국, 카자흐스탄, 세인트루시아, 스리랑카, 리베이라, 레소토, 리투아니아, 룩셈부르크, 라트비아, 모로코, 몰도바, 마다가스카르, 마케도니아공화국, 몽고, 말라위, 멕시코, 모잠비크, 노르웨이, 뉴질랜드, 폴란드, 포르투갈, 루마니아, 러시아, 수단, 스웨덴, 싱가포르, 슬로베니아, 슬로바키아, 시에라리온, 타지키스탄, 투르크멘, 터키, 트리니다드토바고, 탄자니아, 우크라이나, 우간다, 우즈베키스탄, 베트남, 세르비아 앤 몬테네그로, 남아프리카, 짐바브웨,

AP ARIPO특허 : 가나, 감비아, 케냐, 레소토, 말라위, 모잠비크, 수단, 시에라리온, 스와질랜드, 탄자니아, 우간다, 짐바브웨,

EA 유라시아특허 : 아르메니아, 아제르바이잔, 벨라루스, 키르키즈스탄, 카자흐스탄, 몰도바, 러시아, 타지키스탄, 투르크멘,

EP 유럽특허 : 오스트리아, 벨기에, 스위스, 사이프러스, 독일, 덴마크, 스페인, 핀란드, 프랑스, 영국, 그리스, 아일랜드, 이탈리아, 룩셈부르크, 모나코, 네덜란드, 포르투갈, 스웨덴,

OA OAPI특허 : 부르키나파소, 베닌, 중앙아프리카, 콩고, 코트디부아르, 카메룬, 가봉, 기니, 기니 비사우, 말리, 모리타니, 니제르, 세네갈, 차드, 토고,

(30) 우선권주장 09/426,855 1999년10월26일 미국(US)

(73) 특허권자 인터내셔널 비지네스 머신즈 코퍼레이션
미국 10504 뉴욕주 아몽크 뉴오차드 로드

(72) 발명자 케토로버트
미국27609노스캐롤라이나주랄레이그렌빌드라이브3040

 두푼
미국27513노스캐롤라이나주캐리바르텔드라이브301

 코리쿠마
미국27511노스캐롤라이나주캐리메라트코트105

본 발명은 전반적으로 입력 디바이스에 관한 것이며, 특히, 다수의 모드로 동작할 수 있는 USB(Universal Serial Bus) 디바이스로서 동작하는 키보드 및 다른 디바이스에 관한 것이다.

배경기술

POS(Point-of-Sale) 터미널과 같은 수 많은 애플리케이션은 스캐너, 현금 인출기, 비디오 모니터, 프린터, 키보드와 같은 다양한 디바이스를 지원하기 위한 기초로서 개인용 컴퓨터를 사용한다. POS 터미널에서, 키보드는 통상적으로 개인용 컴퓨터에서 사용되는 112 키보드(즉, "통상적인 키 스트로크 데이터(conventional keystroke data)"를 입력하는데 사용되는 "표준형 키보드"임)의 키보다 많은 키를 포함한다. POS 터미널의 추가된 키는 현금 등록 명령, 및 POS 터미널과 특정하게 관련된 다른 명령과 같은 "비통상적인 키 스트로크 데이터"의 입력을 가능하게 한다. 수 많은 이들 터미널은 도스(DOS), 윈도우 3.11, 윈도우 9x 또는 윈도우 NT와 같은 표준 운영 체제를 사용한다. 이들 터미널은 상기 운영 체제와 POS 애플리케이션 프로그램 사이에서 층으로 존재하는 소프트웨어 드라이버 및 프로그램의 "중간 레벨(mid-level)" 시스템 또는 특수한 소프트웨어 시스템(specialised software system)을 사용한다. 이 드라이버 및 프로그램의 시스템은 본 기술의 당업자에게 POSS(Point of Sale Subsystem)으로 알려진다. 이들은 운영 체제가 로드된 후, 그러나 애플리케이션 프로그램이 로드되기 전에 컴퓨터 메모리로 로드되기 때문에, "중간 레벨"로 간주된다. POSS는 체크 스캐너와 현금 등록 인출기와 같은, POS 터미널 환경에만 있는 하드웨어 및 장치의 제어와 동작을 위해 필요한 추가적 드라이버와 기능성을 포함한다.

애플리케이션 프로그램은 "하이 레벨(high level)"로 간주되는데, 그 이유는 통상적으로 애플리케이션 프로그램이 운영 체제 및 POSS가 로드된 후에(즉, 운영 체제의 상부 상에) 로드되기 때문이다. 애플리케이션 프로그램은 두 개의 타입의 태스크를 수행하는데, 한 가지는 판매 거래와 관련된 입력 텍스트 또는 숫자를 입수하는 것과 같은 수행되고 있는 작업에 관한 것이며, 다른 한 가지는 파일을 관리하고 정보를 화면 상으로 디스플레이하는 것과 같은 유지 작업(maintenance chores)에 관한 것이다. 유지 작업은 실제적으로 애플리케이션 프로그램으로부터의 요청에 응답하여 운영 체제 또는 POSS에 의해 수행된다. 유지 작업은 본 기술의 당업자에게 하위 레벨 기능으로 간주된다. 표준형 애플리케이션 프로그램은 가령 마이크로소프트 워드 또는 엑셀을 포함하는 통상적으로 개인용 컴퓨터 상에서 동작하는 프로그램이다. 말하자면, 이들 표준형 애플리케이션 프로그램은 POS 환경에서 특정하게 동작하도록 설계된 애플리케이션 이외의 것이다. 대조적으로, 특수한 애플리케이션 프로그램은 POS 환경에서 특정하게 동작하도록 설계된 애플리케이션 프로그램이다.

새롭게 도입된 USB가 수많은 주변 장치를 PC로의 통합을 제공하지만, 비표준형 장치에 문제가 존재한다. 기존 USB 아키텍처는 POS 키보드의 추가된 키 뿐만 아니라 통상적인 키 스트로크 데이터 이외의 키보드 데이터도 지원하지 않는다. 그러므로, USB 키보드는, POS 애플리케이션 프로그램이 특정 키보드로 동작하도록 설계되지 않는한, POS 터미널 상에서는 사용될 수 없다. 달리 말하면, 각 애플리케이션 프로그램은 다양한 POS 키보드와 인터페이스하도록 특정하게 설계되어야 하는데, 이는 소프트웨어 개발, 유지 및 지원 비용을 높인다. 또한, 애플리케이션 프로그램이 특정 키보드로 동작하도록 설계하는 것은 유연성에 있어서 상당한 손실을 초래하는데, 그 이유는 새로운 또는 상이한 POS 키보드가 지원될 필요가 있을 때마다 애플리케이션 프로그램이 개정되어야 하기 때문이다.

또한, POS 키보드가, 표준형 키보드와 같은 제 1 모드 및 POS 키보드 모드와 같은 하나 또는 그 이상의 제 2 및 제 3 모드의 다중 모드로 동작할 수 있는 것이 바람직하다. 제 1 모드는 바이오스 시스템(BIOS system), 운영 체제, 진단 시스템(diagnostics)과 같은 하위 레벨 소프트웨어를 갱신 및 로드하는데 바람직하다. 키보드가 표준형 키보드로 기능할 수 없는 한, 본 실시예가 아니라면, 다른 키보드가 하위 레벨 소프트웨어를 갱신할 때 사용되어야만 할 것이다. 또한, 그러한 키보드는 동작 모드 간에 쉽게 전환될 수 있어야 한다.

미국 특허 5,504,483은 개인용 컴퓨터 키보드에 부착된 마그네틱 카드 판독기를 개시한다. 키보드의 추가 기능은 개인용 컴퓨터 키보드에 의해서는 사용되지 않는 제어 코드를 사용함으로써 활성화된다.

그러므로, 필요한 것은 표준형 USB 포트를 통해 POS 터미널에 부착되어 표준형 USB 키보드를 에뮬레이션하는, 제 2 또는 비표준형 타입 USB 키보드이다. 다른 모드에서, 키보드는 추가된 POS 키의 인식을 가능하게 한다(POS 모드). 다른 모드에서, 키보드는 USB를 통해 통신하는 다른 프로토콜의 사용을 허용한다.

따라서, 본 발명은 컴퓨터 시스템을 제공하는데, 상기 컴퓨터 시스템은 컴퓨터 유닛과, 표준형 애플리케이션 프로그램을 동작시킬 수 있는 상기 컴퓨터 유닛 상에서 동작하는 운영 체제와, 특수한 애플리케이션 소프트웨어를 동작시킬 수 있는 상기 운영 체제와 동시에 동작하는 특수한 소프트웨어 시스템과, 상기 컴퓨터 유닛에 접속된 데이터 버스와, 통상적인 키를 구비한 키보드와, 제 1 모드에서 상기 통상적인 키가 상기 표준형 애플리케이션 프로그램과 통신하도록 하게 하는 회로를 포함한다. 본 발명은, 상기 키보드가, 상기 데이터 버스에 접속된 비통상적인 키와, 제 2 모드에서, 상기 통상적인 키가 상기 표준형 애플리케이션 프로그램 및 상기 특수한 애플리케이션 소프트웨어와 통신하는 것을 가능하게 하며, 상기 비통상적인 키가 상기 특수한 애플리케이션 소프트웨어와 통신하는 것을 가능하게 하는 회로를 더 포함한다.

본 발명은 데이터를 호스트 컴퓨터에 입력하는 USB 호환가능한 디바이스를 제공하는데, 상기 디바이스는 USB 노드, 이 USB 노드에 접속된 제어 시스템—상기 제어 시스템은 표준형 UBS 모드 및 비표준형 USB 모드를 가짐—을 갖는 온 보드 디바이스 제어기(on-board device controller)를 포함한다.

본 발명은 USB 디바이스와 호스트 컴퓨터 상에서 동작하는 소프트웨어 애플리케이션 프로그램 간의 통신—상기 통신은 표준형 USB 모드 및 적어도 하나의 비표준형 USB 모드로 발생함—을 하는 방법을 제공하며, 상기 방법은 호스트 컴퓨터 상에서 동작하는 운영 체제 내의 표준형 데이터 스택을 사용하는 제 1 데이터 스택을 생성하는 단계와, 상기 호스트 컴퓨터 상에서 동작하는 운영 체제 내의 상기 표준형 데이터 스택의 일부를 바이패스하는 제 2 데이터 스택을 생성하는 단계와, 통신이 상기 표준형 USB 모드로 발생할 때 상기 표준형 데이터 스택을 사용함으로써 하드웨어 드라이버 및 애플리케이션 프로그램으로/으로부터 데이터를 전송하는 단계와, 상기 USB 디바이스가 상기 비표준형 USB 모드로 기능하고 있을 때 상기 제 2 데이터 스택을 사용함으로써 하드웨어 드라이버 및 애플리케이션 프로그램으로/으로부터 데이터를 전송하는 단계를 포함한다.

바람직하게는, 실시예는 세 개의 기능 모드, 바람직하게는 USB 모드로 동작하는 키보드를 사용하는 컴퓨터 시스템을 제공한다. 제 1 또는 표준형 동작 모드는 오직 표준형 컴퓨터 키보드 및 발광 다이오드(LED) 만을 지원하는 표준형 USB 키보드 모드이다. 이 모드에서, POS 기능키는, USB 표준(Universal Serial Bus Specifications, revision 1.1, Setempber 23, 1998) 내에 포함되지만 F13, F14, 뮤트(mute) 등과 같은 표준형 PC 키보드에는 존재하지 않는 비범용 키(unusual keys)로 맵핑된다. 이 모드의 키보드에서 발견되는 그러한 표준형 키는 POS 기능키에 의존하지 않는 임의의 소프트웨어 애플리케이션에 의해서는 사용될 수 있다. 이로써, 사용자는 키보드가 사용될 수 있기 전에 비교적 높은 레벨의 드라이버의 로드와 의존하지 않고도 BIOS를 갱신하고 DOS를 사용하거나 다른 기능을 수행할 수 있다.

삭제

제 2 모드는 표준형 컴퓨터 키보드 키 뿐만 아니라 POS 기능키도 지원한다. 이 모드는 사용자에 의해 현재의 활성 윈도우가 되도록 선택되어 있는 임의의 애플리케이션에 의해 사용될 수 있다. 오퍼레이터의 관점에서 현재 활성인 윈도우가 되는 것은 이 윈도우가 다른 윈도우의 상부 상에 존재한다는 것을 의미한다. 그러므로, 오퍼레이터는 현재 활성 윈도우와 직접적으로 인터페이싱한다. 키보드로부터 키 스토르크는 새로운 소프트웨어 드라이버를 통해 애플리케이션 프로그램과 통신되는데, 상기 새로운 소프트웨어 드라이버는 표준형 운영 체제 USB 소프트웨어 스택의 일부를 바이패스한 다음 다시 표준형 운영 체제 USB 스택으로 접속된다. 이어서, USB 스택은 운영 체제 메시지 큐로 접속된다. 메시지 큐는 전송을 대기하는 메시지들의 정렬된 FIFO 라인이다.

제 3 모드는, USB를 통해 통신하는 다른 프로토콜을 이용하여, 표준형 컴퓨터 키보드 및 POS 기능키 및 LED를 지원한다. 이 모드는 오직 POS 애플리케이션에 의해서만 사용될 수 있다. 모든 키보드 데이터는 POSS 드라이버와 직접적으로 통신하는 소프트웨어 드라이버를 통해 통신된다. 이로써, 통신은 표준형 운영 체제 USB 소프트웨어 스택 및 표준형 운영 체제 메시지 큐 모두를 바이패스한다. 이 모드에서, 오퍼레이터는 터미널의 비POS(non-POS) 애플리케이션으로 액세스할 수 없다.

키보드는 이들 모드로 동작할 수 있는데, 그 이유는 (1) 키보드의 펌웨어에 추가된 소프트웨어 루틴으로 인해 자신을 다른 USB 디바이스와 식별할 수 있게 하며, (2) 표준형 USB 드라이버와 공존하는 호스트 내의 추가된 소프트웨어 드라이버(상기 드라이버는 운영 체제를 동반함) 때문이다. 이들 추가된 드라이버는 표준형 USB 데이터 스택과 독립적으로 동작하는, 호스트 메모리 내의 추가된 데이터 스택을 생성한다. 또한, 키보드가 모드 간에서 스위칭될 수 있도록 하기 위해, USB 호스트로 하여금 키보드를 새로운 디바이스로서 리스트하거나 식별하게 하는 하드웨어 수단이 제공된다.

본 발명의 실시예는 첨부 도면을 참조하여 실시예에 의해 기술될 것이다.

도면의 간단한 설명

도 1은 실시예에 따르는 USB 인터페이스 및 USB 인에이블 키보드(USB enabled keyboard)를 갖는 호스트 데이터 프로세싱 시스템의 블록도,

도 2는 실시예에 따르는 호스트 컴퓨터의 메모리 내에 존재하는 데이터 구조 및 스택의 도면,

도 3은 실시예에 따르는 USB 인에이블 키보드의 블록도,

도 4는 실시예에 따르는 USB 인에이블 키보드의 모드 스위칭 절차를 도시한 흐름도,

도 5는 실시예에 따르는 키보드의 메모리로 로드된 펌웨어의 모드 스위칭 절차의 흐름도.

발명의 상세한 설명

도 1에서, 호스트 컴퓨터 유닛 또는 호스트 중앙 처리 장치(CPU)(101)를 포함하는 컴퓨터가 도시된다. CPU(101)는 프로그램, 인스트럭션 및 데이터를 RAM(106)으로 판독한다. CPU(101)는 USB 호스트 제어기(102)를 이용하여 데이터 버스를 따라 데이터를 교환한다. USB 호스트 제어기(102)는 대응하는 USB 네트워크 또는 버스, 즉 데이터 통신 수단에 접속된 USB 디바이스와 인터페이스하며 이를 제어한다. 이 실시예에서, USB 호스트 제어기(102)는 USB 인에이블 키보드(105)를 제어한다. 모니터(104), 하드 드라이브, 병렬 프린터, 직렬 모뎀 등과 같은 다른 비USB(non-USB) 디바이스는 입출력 서브시스템(103)을 통해 CPU(101)와 통신한다. I/O 서브시스템(103)은 다수의 컴퓨터 주변장치와 통신하기 위한 다수의 상이한 포트를 포함한다.

적당한 드라이버 및 데이터 구조가 RAM(106)으로 로드될 때, USB 호스트 제어기(102)는 CPU(101)로 하여금 USB 디바이스에 명령케하여, 그 USB 디바이스를 USB 호스트 제어기(102)로부터 논리적으로 분리하고 완전히 상이한 디바이스로서 재접속하도록 할 수 있다. 원래의 디바이스가 분리되고 다른 USB 디바이스가 그 자리에서 접속되는 것으로 호스트 컴퓨터에게 보이지만, 하드웨어 인터페이스는 동일하게 유지된다. 새로운 디바이스는 상이한 USB 소프트웨어 인터페이스를 통해 RAM(106) 내의 USB 데이터 스택의 상이한 부분과 통신한다.

USB 프로토콜은 시작(start-up) 시에 디바이스를 구성(configure)하거나 이들 디바이스가 실행 중에 플러그 인될 때 디바이스를 구성할 수 있다. 이들 디바이스는 다양한 디바이스 클래스로 분할된다. 각 디바이스 클래스는 유사한 기능을 하는 디바이스에 대한 공통의 동작 및 프로토콜을 규정한다. HID(human interface device) 클래스는 컴퓨터 시스템의 동작을 제어하기 위해 인간에 의해 사용되는 디바이스로 주로 구성된다. 전형적인 실례는 키보드 및 포인팅 디바이스이다.

도 2는 호스트 컴퓨터의 RAM(106)에서 생성된 데이터 스택의 기능 개략도이다. 애플리케이션 프로그램, POSS, 운영 체제, 마이크로프로세서 간의 통신은 데이터 스택 또는 통신 관로(communication conduit)를 통해 발생한다. 데이터 구조는 특정 동작이 다양한 루틴 및 프로그램에 의해 데이터 상에서 수행될 수 있게 해주기 위한, 데이터에 적용된 구성 체계(organisational schemes)이다.

USB/HID 클래스 디바이스는 모든 데이터를 수신하고 라우팅(route)하기 위해 대응하는 HID 클래스 드라이버(202)를 사용한다. USB 디바이스로부터의 USB 데이터는 도 1에 도시된 USB 호스트 제어기(102)와 같은 USB 허브 또는 USB 호스트 제어기 또는 서브시스템으로 보내진 다음, HID 클래스 드라이버(202)와 같은 하위 레벨 프로그램 또는 드라이버를 통해 호스트 컴퓨터 메모리, RAM(106)으로 보내진다. 윈도우 NT와 같은 통상적인 운영 체제에서, 통상적인 PC 키보드는 "부트" 모드로 동작된다. "부트" 모드 키보드 데이터는 데이터 경로(203)를 통해 USB 키보드 드라이버(204)로 전달되며, 상기 USB 키보드 드라이버(204)는 키보드와 인터페이스하도록 설계된 표준형 드라이버이다. USB 키보드 드라이버(204)

는 키보드로부터의 키보드 데이터를 데이터 경로(205)를 통해 USB 키보드 클래스 드라이버(206)로 보낸다. USB 키보드 드라이버(204)는 통상적으로 운영 체제의 일부가 되는 "부트 모드" 키보드 드라이버이다. USB 키보드 드라이버(204)는 POS 특정 키 스트로크를 인식하지 않는다. 키보드 클래스 드라이버(206)는 키보드 클래스에 대한 드라이버이다. 이 드라이버는 모든 키보드 관련 기능을 다루도록 설계된다. 이 드라이버는 운영 체제 특정 드라이버이지만, 통상적으로는 키보드 버스(가령, PS/2 또는 USB)에 대해서는 독립적이다. 이로써, USB 키보드 클래스 드라이버(206)는 USB 및 시스템 부착된 키보드를 지원하는 일반적 키보드 드라이버이다. 한편, USB 키보드 드라이버(204)는 오직 부트 모드 USB 키보드와 함께 사용될 수 있다. USB 키보드 클래스 드라이버(206)는 키보드가 임의의 특수한 소프트웨어 드라이버 없이도 동작하게 하는 디폴트 드라이버이다.

USB 키보드 클래스 드라이버(206)로부터, 데이터는 데이터 경로(207)를 통해 운영 체제 메시지 큐(208)에 보내진다. 운영 체제는 메시지 큐 내에 저장된 메시지를 통해 애플리케이션 프로그램(210 또는 212)과 통신한다. 메시지 큐(208)는 운영 체제를 위한 주 입력 큐이다. 그것은 키 스트로크 데이터 및, 마우스 이동 또는 윈도우 크기 조절 또는 이동 사건(event)과 같은 다른 입력 사건을 위해 설계된다. 운영 체제 메시지 큐(208)로부터, 데이터는 데이터 경로(209)를 통해 POS 애플리케이션(210)으로 전송되거나 운영 체제 환경 내에서 동작할 수 있는 통상적인 (비POS) 애플리케이션(212)으로 전송된다. 운영 체제 메시지 큐(208) 및 통상적인 (비POS) 애플리케이션(212) 간의 데이터 경로는 데이터 경로(213)이다. 데이터 경로(203,205,207,213)를 따라 HID 클래스 드라이버(202)로부터 비POS 애플리케이션(212)으로의 데이터의 흐름은 데이터 스택으로 지칭된다.

데이터 경로(203,205,207,213)에 의해 규정된 이러한 데이터 스택은 표준형 "부트 모드" 운영 체제 USB 스택으로 알려져 있다. 단지 표준형 컴퓨터 키 스트로크만이 이러한 데이터 스택을 통해 애플리케이션 프로그램(203)으로 보내질 것이다. 이러한 데이터 스택은 표준형 운영 체제 드라이버를 이용한다. 본 발명의 실시예가 이러한 스택을 이용할 때, 통상적인 컴퓨터 키보드와 같이 동작한다. 따라서, 컴퓨터는 추가된 POS 기능키를 인식하지 못하는데, 그 이유는 표준형 운영 체제 드라이버가 추가된 POS 기능키를 인식하지 못하기 때문이다.

표준형 키 스트로크 데이터가 표준형 USB 스택의 일부로 라우팅되고, 한편 마그네틱 스트립 판독기(magnetic stripe reader)(MSR)와 같은 POS 데이터 및 POS 기능 데이터(LED, 키 로크(key lock), 디스플레이 등등)가 새로운 데이터 스택을 통해 라우팅될 때, 제 2 "비부트 모드(non-boot mode)"가 규정된다. 이 모드에서, 모든 데이터는 경로(215)를 따라 HID 클래스 드라이버(202)로부터 USB POS 키보드 드라이버(214)로 보내진다. USB POS 키보드 드라이버(214)가 바람직할데, 그 이유는 표준형 키 스트로크 데이터 및 POS 키 스트로크 데이터를 동기할 필요가 있기 때문이다. 달리 말하면, USB POS 키보드 드라이버(214)는 USB 키보드 드라이버(204)에 대한 다른 드라이버이다. USB POS 키보드 드라이버(214)는 타입에 따라 세 개의 루트(route)로 데이터를 분할한다. 제 1 루트에서, 키 스트로크 데이터는 USB 키보드 드라이버(204)를 바이패스하며 키보드 클래스 드라이버(206)의 바닥(bottom)에서 마이크로소프트 USB 스택(Microsoft USB stack)에 들어간다. 이로써, 비부트 모드 키 스트로크 데이터에 대한 데이터 스택은 데이터 경로(215,217,207,209)에 의해 규정될 수 있다. 제 2 데이터 타입, MSR 데이터는 경로(219)를 따라 POSS(216) 내의 MSR 루틴으로 보내진다. 제 3 데이터 타입, POS 기능 데이터는 경로(221)를 통해 POSS(216)의 키보드 루틴으로 보내진다. POSS(216)는 데이터 경로(225)를 통해 요구될 때 MSR 및 POS 기능 데이터를 애플리케이션 프로그램에 전송한다. 그러므로, USB POS 키보드 드라이버(214)는 비부트 모드(표준형 PC 키와 POS 키가 더해짐)로 키보드를 서비스한다. 비부트 모드는 키보드의 POS 기능 뿐만 아니라 표준형 컴퓨터 키 및 POS 키의 지원을 가능하게 한다. 그러므로, 오퍼레이터는 POS 애플리케이션 프로그램 뿐만 아니라 통상적인 애플리케이션 프로그램을 동작시킬 수 있다.

제 3 모드("레거시 모드")("Legacy Mode")는, 모든 키보드 데이터가 표준형 드라이버 및 표준형 USB 스택을 바이패스할 때, 규정된다. 이 모드에서, HID 클래스 드라이버(202)는 데이터를 데이터 경로(218)를 통해 USB POS 키보드 드라이버(214)로 보낸다. USB POS 키보드 드라이버(214)는 운영 체제 USB 스택 주위의 모든 키보드의 데이터 트래픽(즉, USB 키보드 드라이버(204), 키보드 클래스 드라이버(206), 운영 체제 메시지 큐(208))을 라우팅한다. USB POS 키보드 드라이버(214)는 MSR 데이터를 데이터 경로(219)를 통해 POSS(216) 내의 MSR 루틴에 라우팅한다. 키 로크 데이터(key lock data) 및 POS LED 데이터는 데이터 경로(221)를 통해 POSS(216) 내의 키보드 루틴으로 라우팅된다. 레거시 모드의 키 스트로크 데이터는 경로(223)를 통해 POSS(216)의 키보드 루틴으로 라우팅된다. 이 모드는 POS 키 및 키보드의 임의의 다른 POS 기능 뿐만 아니라 표준형 컴퓨터 키의 지원을 가능하게 한다. 이 모드에서, 키보드는 비POS 윈도우 프로그램과 직접적으로 상호작용하지 않는다. POSS(216)는 단지 이 POSS(216)과 함께 등록된 POS 애플리케이션 프로그램과 상호작용할 뿐이다. 유리하게는, 이 동작 모드는 오퍼레이터가 호스트 컴퓨터의 비POS 애플리케이션 프로그램을 액세스하지 못하도록 한다.

도 3은, 키보드 제어 시스템 및 하드웨어의 다양한 내부 특징부를 도시하는 블록도이다. 키보드(105)는 USB 케이블(도시되지 않음)을 통해 호스트 CPU(101)와 인터페이스한다. 키보드(105)는 마이크로제어기/USB 인터페이스(304), 키보드

스위치 매트릭스 및 키보드 인터페이스 회로(310), 디지털 메모리(306 및/또는 308), 전압 조정기(302)와 같은 다수의 통상적인 키보드 구성 요소를 포함한다. 디지털 메모리는 플래시 메모리, ROM(306), RAM(308)이 될 수 있다. 키보드의 동작 및 제어는 ROM(306)으로부터 판독된 인스트럭션 또는 오퍼레이팅 로직에 응답하여 마이크로제어기/USB 인터페이스(304)에 의해 주로 수행된다. 이러한 저장된 로직은 "펌웨어"로 알려져 있다. RAM(308)는 컴퓨터 오퍼레이터에 의해 입력된 정보 및 동작 중 마이크로제어기/USB 인터페이스(304)에 의해 사용된 프로그램 변수를 저장한다.

마이크로제어기/USB 인터페이스(304)는 USB 애플리케이션을 위한 주변 마이크로제어기(peripheral microcontroller)이다. 이러한 마이크로제어기는 온 칩 USB 모듈(on-chip USB module)이다. USB 모듈은 마이크로제어기와 함께 동작하여, 알려진 방식으로 USB 디바이스의 능력을 규정 및 제공한다. 마이크로제어기/USB 인터페이스(304)는 와치독 타이머(watchdog timer)(307)를 포함한다. 와치독 타이머(307)는 동작이 완료되지 못할 경우 마이크로제어기/USB 인터페이스(304)를 자동적으로 리셋하는 회로이다. ROM(306)에 저장된 펌웨어에 의해 인에이블될 때, 와치독 타이머(307)는 동작하기 시작하며, 펌웨어가 개입하지 않는다면, 와치독 타이머(307)는 오버플로우되고 마이크로제어기/USB 인터페이스(304)의 칩 리셋을 시작한다. 이로써, 마이크로제어기/USB 인터페이스(304)가 리셋된다. 통상적인 동작에서, 펌웨어는 와치독 타이머(307)를 주기적으로 클리어하여 리셋을 방지한다.

키보드 인터페이스 회로(310)는 PC 타입 또는 POS 컴퓨터 시스템의 키보드를 위한 통상적인 프로토콜에 따라 동작한다. 키보드 인터페이스 회로(310)는 키보드 스위치 매트릭스를 모니터링하며, 마이크로제어기/USB 인터페이스(304)에 의해 스캐닝될(scanned) 때에는 키보드 상에서 눌러진 임의의 키에 대응하는 키 스트로크 신호를 생성한다.

도 4는 본 발명의 디바이스의 실시예의 동작을 도시하는 흐름도이다. 도 4에 도시된 프로세스 단계는, 이 실시예에서, USB POS 키보드 드라이버(214)에 착수된 단계를 나타낸다. USB POS 키보드 드라이버(214)가 이 특정 루틴을 호출할 때 실행은 단계(402)에서 시작된다. 단계(404)에서의 예비 탐색은 USB에 접속된 임의의 (부트) 키보드가 존재하지의 여부를 판정한다. 단계(404)가 USB에 접속된 임의의 (부트) 키보드가 존재한다고 판정하면, 단계(406)는 USB 인터페이스를 통해 키보드에 의해 보내진 디바이스 기술자(device descriptor)를 검사함으로써 이들 임의의 키보드가 다중모드 키보드인지를 판정한다. 만약 단계(404)의 판정 또는 단계(406)의 판정이 부정적인 판정이라면, 다중모드 키보드가 존재하지 않으며 루틴의 잔여 부분은 스킵된다. 호스트 컴퓨터는 제 1 모드, 즉 부트 모드로 동작할 수 있으며 표준형 드라이버, 데이터 스택 및 구조를 사용한다. 이와 달리, 단계(406)의 판정이 다중모드 키보드가 존재함을 보이면, CPU(101)는 단계(408)에서 모드 변경 명령을 다중모드 키보드에 보낸다. 모드 변경 명령의 수신에 응답하여, 다중모드 키보드는 USB 네트워크로부터 논리적으로 분리된 다음에 상기 네트워크에 논리적으로 재접속된다. 이러한 재접속 후에, 키보드는 그 자신을 다른 USB 디바이스, 즉 POS 키보드로 식별한다. 단계(410)에서, POS 키보드는 운영 체제의 하위 플러그 앤 플레이 인프라스트럭처(underlying Plug and Play infrastructure)에 의해 검출된다. 단계(412)에서, HID 클래스 드라이버(202)는 키보드가 사용될 준비가 되었음(즉, 디바이스가 온라인임)을 애플리케이션에 보고한다. 단계(414)에서, 키보드 모드 스위치 절차는 완료되며, 통상적인 절차 및 애플리케이션 프로그램의 실행은 계속된다.

호스트 컴퓨터의 모드 변경에 대한 키보드의 응답이 도 5의 흐름도에서 설명된다. 이 절차의 실행은 단계(502)에서 시작된다. CPU(101)에 의해 단계(408)에서 전송된 모드 변경 명령은 단계(504)에서 마이크로제어기/USB 인터페이스(304)에 의해 수신된다. 단계(506)에서, 마이크로제어기/USB 인터페이스(304)는 (CPU(101)에 세트된 모드 및 데이터 구조를 정합하기 위해) 다음 동작 모드를 식별하도록 RAM(308) 내의 비트 패턴을 세트한다. 단계(508)에서, 마이크로제어기/USB 인터페이스(304)는 개입 유지 신호(intervening maintenance signal)를 와치독 타이머(307)에 보내는 것을 중지한다. 이로써, 와치독 타이머(307)는 오버플로우되고 단계(510)에서 칩 리셋을 시작한다. 리셋 후에, 마이크로제어기/USB 인터페이스(304)는 단계(512)에서 이전에 저장된 비트 패턴을 RAM(308)로부터 판독하여 의도된 동작 모드를 판정한다. 단계(512)의 판정을 기초로 하여, 마이크로제어기/USB 인터페이스(304)는 단계(513)에서 새로운 모드로 스위칭되며 단계(514)에서 새로운 디바이스 기술자 또는 식별 코드를 호스트 컴퓨터에 보낸다. 각 동작 모드는 상이한 디바이스 기술자를 가지며, 그래서 CPU(101)는 키보드를 상이한 디바이스로 간주한다. 이로써, 통신은 호스트 컴퓨터에 있어서 도 4의 단계(410)에서 도시된 새로운 디바이스로서 재확립되며 키보드 모드 변경 루틴은 단계(516)에서 종료된다.

다중모드 USB 디바이스는 종래 기술에 대해 몇가지 실질적인 장점을 갖는다. 이 디바이스는 하드웨어 드라이버 레벨에서는 인식되지만 애플리케이션 레벨에서는 인식되지 않거나 양 레벨 모두에서 인식되는 비표준형 키의 인식을 가능하게 하는 다중 모드로 동작할 수 있다. 이전에는 USB POS 키보드의 사용은 대응 애플리케이션 프로그램의 수정을 요구했다. 특정 키보드에 따라 동작하는 각 애플리케이션 프로그램을 설계하는 것은 보다 높은 소프트웨어 개발, 유지 및 지원 비용을 초래하였다. 또한, 특정 키보드에 따라 동작하는 애플리케이션 프로그램의 설계는 유연성에 있어서 상당한 손실을 초래하는데, 그 이유는 대응하는 애플리케이션 프로그램이 새로운 POS 키보드가 지원될 필요가 있을 때마다 개정되어야 하기 때문이다.

다중모드 동작은 또한 POS 키보드가 표준형 키보드로 인식되는 것이 가능하게 한다. 표준형 키보드 모드는 BIOS 및 운영 체제와 같은 하위 레벨을 갱신 및 로드하는데 적합하다. 만약 POS 키보드가 표준형 키보드로 기능할 수 없다면, 다른 (표준형) 키보드가 하위 레벨 소프트웨어를 갱신할 때 사용되어야 한다.

상기 실시예들이 입력 디바이스에 모드 변경 명령을 보내는 컴퓨터를 참조하여 기술되었지만, 본 발명은 여기에만 한정되는 것이 아니다. 모드 선택이 가령 입력 디바이스와 관련된 하드웨어 또는 소프트웨어 스위치와 같은 몇몇 다른 수단을 사용하여 성취되는 실시예가 실현될 수 있다. 이 스위치의 동작에 응답하여, 입력 디바이스는 가용한 동작 모드 중의 하나에 따라 동작하도록, 그 자신을 구성하며 이에 따라 컴퓨터 시스템에게 통지한다.

(57) 청구의 범위

청구항 1.

컴퓨터 시스템에 있어서,

컴퓨터 유닛과,

상기 컴퓨터 유닛에 접속된 데이터 버스와,

상기 데이터 버스에 접속되어 상기 컴퓨터 유닛과 통신을 수행하는 컴퓨터 디바이스와,

제 1 기능 세트에 의해 상기 컴퓨터 디바이스가 상기 데이터 버스를 거쳐 상기 컴퓨터 유닛과 통신을 가능케 하도록 하는 제 1 모드 수단과,

제 2 기능 세트에 의해 상기 컴퓨터 디바이스가 상기 데이터 버스를 거쳐 상기 컴퓨터 유닛과 통신을 가능케 하도록 하는 제 2 모드 수단과,

상기 컴퓨터 유닛 상에서 실행되어, 표준형 애플리케이션 프로그램을 실행시킬 수 있는 운영 체제와,

상기 컴퓨터 유닛 상에서 상기 운영 체제와 동시에 실행되어, 상기 컴퓨터 유닛 상에서 특수한 애플리케이션 프로그램을 실행시킬 수 있는 특수한 소프트웨어 시스템을 포함하되,

상기 컴퓨터 디바이스는 통상적인 키스트로크 데이터 및 비 통상적인 키스트로크 데이터를 입력하는 수단을 갖는 컴퓨터 키보드이며,

상기 제 2 기능 세트는 통상적인 키스트로크 데이터를 상기 표준형 애플리케이션 프로그램에 전송하고 비 통상적인 키스트로크 데이터를 특수한 애플리케이션 프로그램에 전송하기 위한 수단을 포함하는

컴퓨터 시스템.

청구항 2.

제 1 항에 있어서,

제 3 기능 세트에 의해 상기 컴퓨터 디바이스가 상기 데이터 버스를 거쳐 상기 컴퓨터 유닛과 통신을 가능케 하도록 하는 제 3 모드 수단을 더 포함하는

컴퓨터 시스템.

청구항 3.

제 1 항에 있어서,

상기 제 1 기능 세트는 통상적인 키스트로크 데이터를 상기 데이터 버스를 거쳐 상기 표준형 애플리케이션 프로그램에 전송하는 수단을 포함하는

컴퓨터 시스템.

청구항 4.

제 2 항에 있어서,

상기 제 3 기능 세트는 통상적인 키스트로크 데이터 및 비 통상적인 키스트로크 데이터를 특수한 애플리케이션 프로그램에 전송하기 위한 수단을 포함하는

컴퓨터 시스템.

청구항 5.

제 1 항에 있어서,

상기 데이터 버스는 USB인

컴퓨터 시스템.

청구항 6.

데이터를 호스트 컴퓨터에 입력하는 USB 호환가능 디바이스(USB compatible device)에 있어서,

USB 모드와,

상기 USB 모드에 접속된 제어 시스템을 갖는 온 보드 디바이스 제어기(on-board device controller)-상기 제어 시스템은 표준형 USB 모드와 비 표준형 USB 모드를 가짐-와,

상기 호스트 컴퓨터가 제 1 모드, 제 2 모드 및 제 3 모드에서 동작중일 때 상기 USB 모드와 통신하는 회로와,

상기 모드들 간을 스위칭하기 위한 회로와,

마이크로제어기와,

메모리 저장장치를 포함하되,

상기 모드들 간을 스위칭하는 상기 회로는 상기 메모리 저장장치 내의 비트 패턴을 새로운 모드(new mode)로 세트하기 위한 회로와, 상기 마이크로제어기를 리셋하는 회로와, 상기 메모리 저장장치로부터 상기 비트 패턴을 판독하는 회로와, 상기 비트 패턴에 기반하여 새로운 식별 코드를 상기 호스트 컴퓨터에 전송하는 회로를 포함하는

USB 호환가능 디바이스.

청구항 7.

삭제

청구항 8.

삭제

청구항 9.

삭제

청구항 10.

삭제

청구항 11.

삭제

청구항 12.

삭제

청구항 13.

삭제

청구항 14.

삭제

청구항 15.

삭제

청구항 16.

삭제

청구항 17.

삭제

청구항 18.

삭제

청구항 19.

삭제

청구항 20.

삭제

청구항 21.

삭제

청구항 22.

컴퓨터 시스템에 있어서,

컴퓨터 유닛과,

상기 컴퓨터 유닛에 접속된 데이터 버스와,

상기 데이터 버스에 접속되어 상기 컴퓨터 유닛과 통신을 수행하는 컴퓨터 디바이스와,

제 1 기능 세트에 의해 상기 컴퓨터 디바이스가 상기 데이터 버스를 거쳐 상기 컴퓨터 유닛과 통신을 가능케 하도록 하는 제 1 모드 수단과,

제 2 기능 세트에 의해 상기 컴퓨터 디바이스가 상기 데이터 버스를 거쳐 상기 컴퓨터 유닛과 통신을 가능케 하도록 하는 제 2 모드 수단과,

상기 컴퓨터 유닛 상에서 실행되어, 표준형 애플리케이션 프로그램을 실행시킬 수 있는 운영 체제와,

상기 컴퓨터 유닛 상에서 상기 운영 체제와 동시에 실행되어, 상기 컴퓨터 유닛 상에서 특수한 애플리케이션 프로그램을 실행시킬 수 있는 특수한 소프트웨어 시스템을 포함하되,

상기 컴퓨터 디바이스는 통상적인 키스트로크 데이터 및 비 통상적인 키스트로크 데이터를 입력하는 수단을 갖는 컴퓨터 키보드이며,

상기 제 2 기능 세트는 통상적인 키스트로크 데이터를 상기 표준형 애플리케이션 프로그램에 전송하고 비 통상적인 키스트로크 데이터를 특수한 애플리케이션 프로그램에 전송하기 위한 수단을 포함하며,

상기 컴퓨터 키보드는, 메모리 저장장치와, 상기 메모리 저장장치로부터 동작 로직을 판독하는 마이크로제어기와, 상기 마이크로제어기에 접속된 데이터 버스 인터페이스와, 상기 마이크로제어기에 접속된 키보드 인터페이스 회로를 포함하며,

상기 메모리 저장장치는 상기 호스트 컴퓨터로 송신될 정보를 저장하기 위한 랜덤 액세스 메모리와, 동작 로직을 포함하는 디지털 메모리를 더 포함하며,

상기 동작 로직은 상기 컴퓨터 디바이스가 상기 제 1 모드 수단, 상기 제 2 모드 수단 및 상기 제 3 모드 수단에서 동작중일 때 상기 데이터 버스와 통신하는 수단과, 상기 모드들 간을 스위칭하는 수단을 포함하며,

상기 스위칭 수단은 상기 키보드의 상기 마이크로제어기 내에서 상기 키보드의 랜덤 액세스 메모리의 비트 패턴을 세트하여 새로운 모드로 세트하는 수단과, 상기 마이크로제어기를 리셋하는 수단과, 상기 랜덤 액세스 메모리로부터 상기 비트 패턴을 판독하는 수단과, 상기 비트 패턴에 기반하여 상기 호스트 컴퓨터에 새로운 식별 코드를 전송하는 수단을 포함하는

컴퓨터 시스템.

청구항 23.

컴퓨터 시스템에 있어서,

컴퓨터 유닛과,

상기 컴퓨터 유닛 상에서 실행되어 표준형 애플리케이션 프로그램을 실행할 수 있는 운영 체제와,

상기 운영 체제와 동시에 실행되어 특수한 애플리케이션 소프트웨어를 실행할 수 있는 특수한 소프트웨어 시스템과,

상기 컴퓨터 유닛에 접속된 데이터 버스와,

상기 데이터 버스에 접속된 통상적인 키와 비 통상적인 키를 갖는 키보드와,

상기 키보드의 통상적인 키가 상기 데이터 버스를 통해 제 1 모드에서 상기 컴퓨터 유닛과 통신을 가능케하는 회로-상기 통상적인 키의 코드들은 제 1 모드에서 상기 표준형 애플리케이션 프로그램에 제공됨-와,

상기 키보드의 통상적인 키가 상기 데이터 버스를 통해 제 2 모드에서 상기 컴퓨터 유닛과 통신을 가능케하는 회로-상기 통상적인 키의 코드들은 상기 애플리케이션 프로그램과 상기 특수한 애플리케이션 소프트웨어 중의 하나에 제공되며, 상기 비 통상적인 키를 가능케하는 코드는 제 2 모드에서 상기 특수한 애플리케이션 소프트웨어에 제공됨-와,

상기 키보드가 제 3 모드에서 상기 컴퓨터 유닛과 통신을 가능케하는 회로-통신하기 위한 상기 통상적인 키와 상기 비 통상적인 키는 제 3 모드에서 상기 특수한 애플리케이션 소프트웨어와 통신하기 위해 제공됨-를 포함하는

컴퓨터 시스템.

청구항 24.

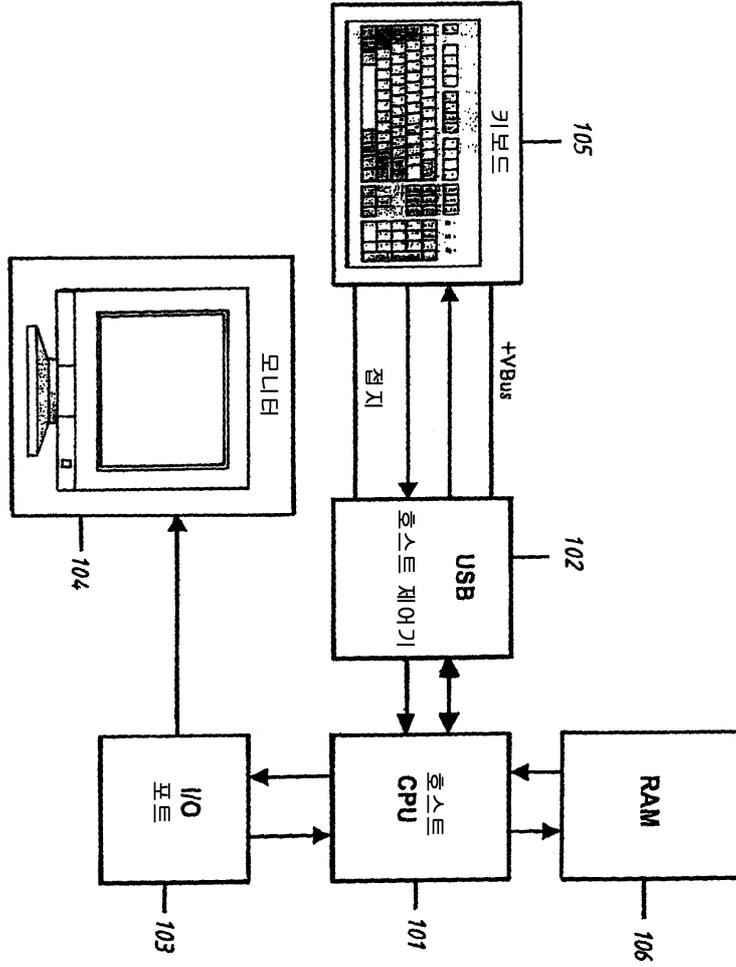
세계의 기능적인 USB 모드, 즉

- (1) 단지 표준형 컴퓨터 키보드 키스트로크만을 지원하는 표준형 시스템 USB 부트 키보드 모드와,
- (2) 표준형 컴퓨터 키보드 키스트로크 및 POS 키스트로크를 지원하는 시스템 년 부트(system non-boot) USB 키보드 모드와,
- (3) 표준형 컴퓨터 키보드 키스트로크 및 POS 키스트로크를 지원하는 년 시스템 년 부트(non-system non-boot) USB 키보드 모드 중의 하나에서 동작하도록 구성되는

정보 처리 시스템.

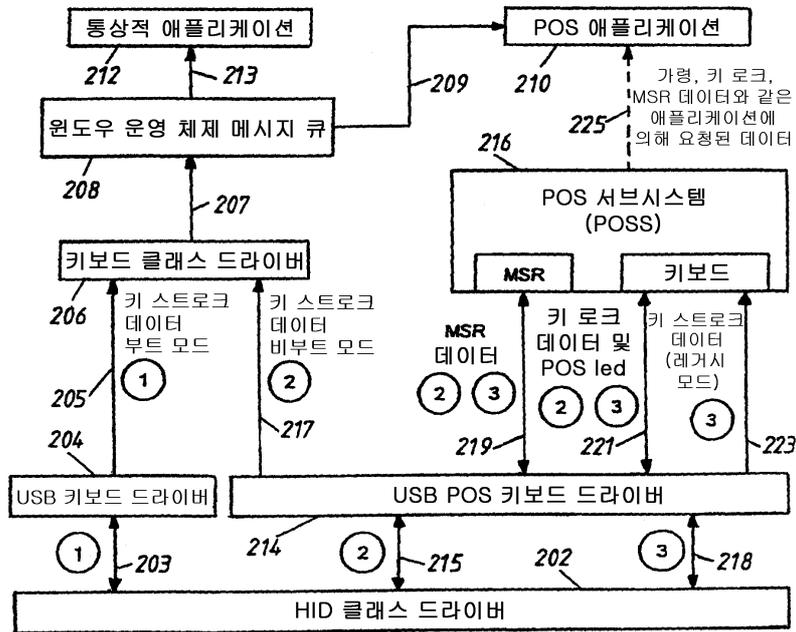
도면

도면1

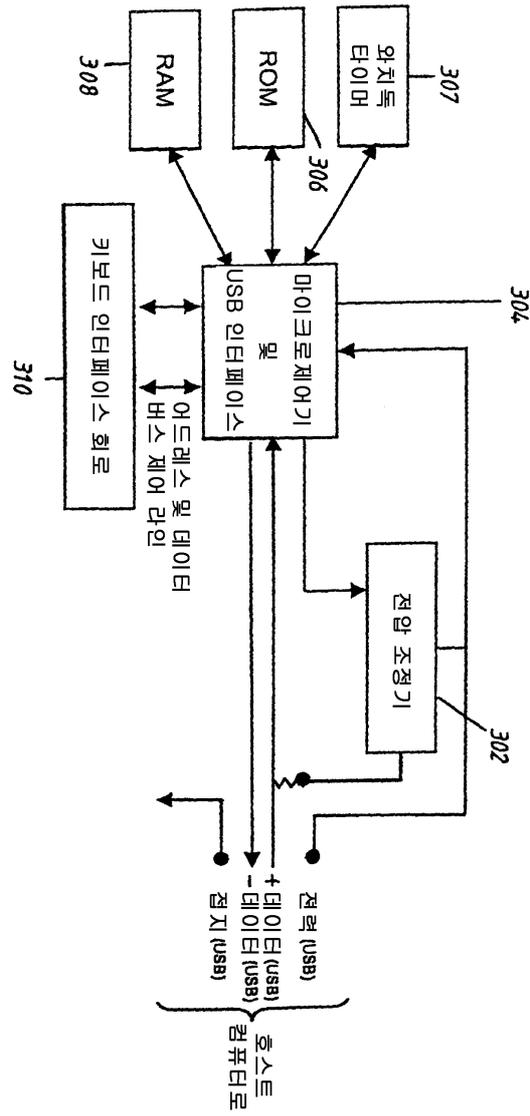


도면2

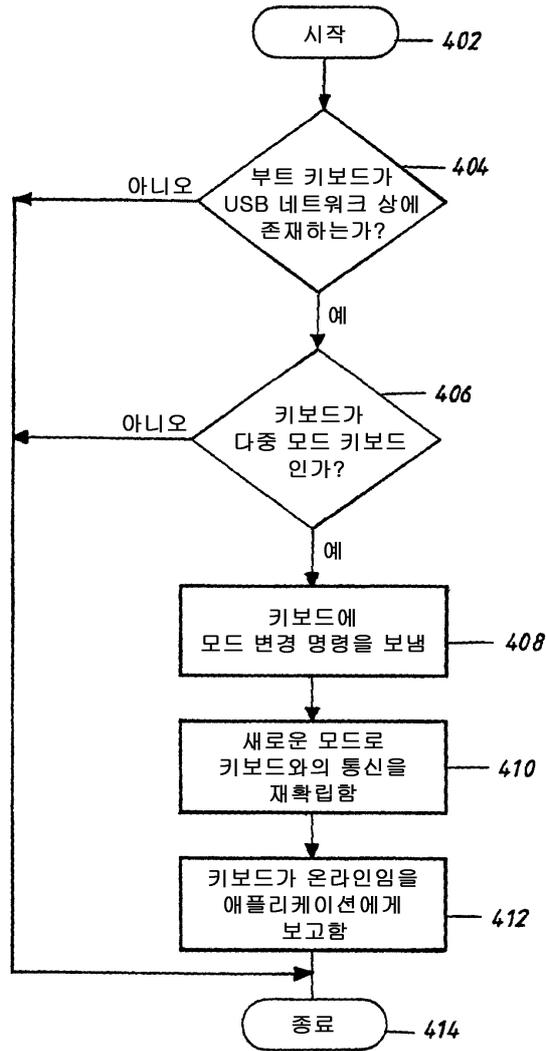
USB 부착된 NANPOS 키보드에 대한 데이터 흐름
(윈도우 NT 환경)



도면3



도면4



도면5

