



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2010년12월30일
(11) 등록번호 10-1005397
(24) 등록일자 2010년12월24일

(51) Int. Cl.

G06F 13/38 (2006.01)

(21) 출원번호 10-2005-7004937

(22) 출원일자(국제출원일자) 2003년05월22일

심사청구일자 2008년05월21일

(85) 번역문제출일자 2005년03월23일

(65) 공개번호 10-2005-0053683

(43) 공개일자 2005년06월08일

(86) 국제출원번호 PCT/IB2003/002236

(87) 국제공개번호 WO 2004/029817

국제공개일자 2004년04월08일

(30) 우선권주장

PCT/SG02/00239 2002년09월24일 싱가포르(SG)

(56) 선행기술조사문헌

JP12196648 A*

US05784581 A1*

*는 심사관에 의하여 인용된 문헌

(73) 특허권자

에스티 에릭슨 에스에이

스위스 제네바 1228 플란-레스-오우아테스 체민
두 캠프-데스-필레스 39

(72) 발명자

니그 치 와이

네덜란드 엔엘-5656 에이에이 아인드호펜 홀스트
란 6

창 예우 케이

네덜란드 엔엘-5656 에이에이 아인드호펜 홀스트
란 6

(뒷면에 계속)

(74) 대리인

김창세

전체 청구항 수 : 총 5 항

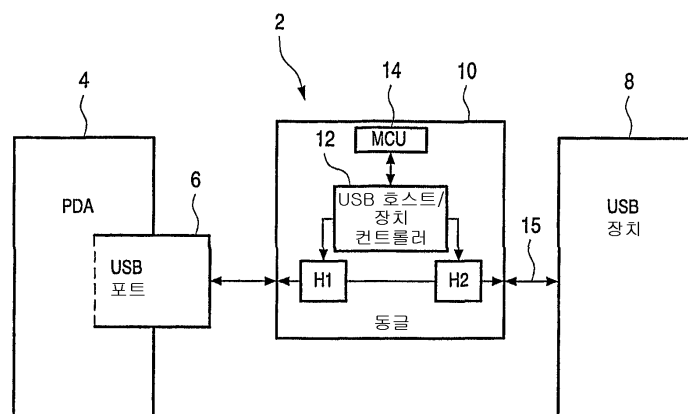
심사관 : 안지현

(54) 버스 스테이션 및 버스 통신 시스템

(57) 요약

하드웨어 동글의 형태인 버스 스테이션은 적절한 소프트웨어를 구동하는 USB 장치와 조합해서 동작한다. 버스 스테이션이 버스 호스트가 그 제 1 버스 통신 포트에 접속되었다고 판정하면, 이는 송수신기로 동작해서 이 버스 호스트와 제 2 버스 통신 포트에 접속된 버스 장치 사이의 기존의 버스 통신을 가능하게 한다. 버스 스테이션이 적절한 소프트웨어를 구동하는 버스 장치가 그 제 1 버스 통신 포트에 접속되어 있다고 판정하면, 이는 또 다른 호스트로서 동작해서 제 1 버스 통신 포트에 접속된 제 1 버스 장치와 제 2 버스 통신 포트에 접속된 버스 장치 사이의 버스 통신을 가능하게 한다.

대표도 - 도1



(72) 발명자

티지아 제롬

네덜란드 엔엘-5656 에이에이 아인드호펜 홀스트란
6

샤마 카우솔

네덜란드 엔엘-5656 에이에이 아인드호펜 홀스트란
6

특허청구의 범위

청구항 1

버스 통신 시스템에서 사용하는 버스 스테이션으로서,

제 1 통신 포트 및 제 2 통신 포트를 포함하고,

상기 버스 스테이션은 상기 제 2 통신 포트에 연결된 호스트 스테이션의 존재를 검출하게 되면 제 1 동작 모드로 동작하되,

상기 제 1 동작 모드는 상기 제 2 통신 포트에 연결된 상기 호스트 스테이션과 상기 제 1 통신 포트에 연결된 제 1 장치 스테이션 사이의 통신을 전송하며,

상기 버스 스테이션은 또한 상기 제 2 통신 포트에 연결된 호스트 스테이션이 존재하지 않음을 검출하게 되면 대체 호스트 스테이션으로서 제 2 동작 모드로 동작하되,

상기 버스 스테이션은 상기 제 2 동작 모드에서, 상기 제 2 통신 포트에 연결된 다른 장치 스테이션 대신에 상기 제 1 통신 포트에 연결된 상기 제 1 장치 스테이션과 통신 프로토콜에 따라 통신하고, 상기 버스 스테이션은 통신을 개시하며 상기 다른 장치 스테이션은 상기 제 1 장치 스테이션에 대한 호스트 스테이션으로서 나타나고 상기 제 1 장치 스테이션과의 통신을 포함하는 동작을 수행하는

버스 스테이션.

청구항 2

제 1 항에 있어서,

상기 버스 스테이션은 상기 제 1 동작 모드에서 USB 송수신기로서 동작하고, 상기 제 2 동작 모드에서 USB 호스트로서 동작하는

버스 스테이션.

청구항 3

제 1 항에 있어서,

상기 버스 스테이션은 상기 제 1 동작 모드에서, 상기 제 1 및 제 2 통신 포트에 연결되어, 상기 제 2 통신 포트에 연결된 상기 호스트 스테이션과 상기 제 1 장치 스테이션 사이의 통신을 전송하는 송수신기 회로를 더 포함하는

버스 스테이션.

청구항 4

버스 시스템에서 사용하는 버스 스테이션으로서,

통신 포트에 연결된 장치 컨트롤러를 포함하되,

상기 통신 포트는 장치 스테이션으로서 동작하고,

상기 버스 스테이션은 운영 체제, 및 호스트 컨트롤러와 통신하고 상기 운영 체제와 정보를 주고 받는 호스트 스테이션 드라이버 소프트웨어를 포함하는 시스템 소프트웨어의 제어 하에 동작하며,

상기 시스템 소프트웨어는 상기 호스트 스테이션 드라이버 소프트웨어에 대해서 호스트 컨트롤러의 존재를 애플레이트하고, 상기 장치 컨트롤러에 대해서 장치 스테이션 드라이버 소프트웨어의 존재를 애플레이트하며, 또한 상기 호스트 스테이션 드라이버 소프트웨어로부터 상기 장치 컨트롤러로 혹은 상기 장치 컨트롤러로부터 상기 호스트 스테이션 드라이버 소프트웨어로 통신을 변환하는 호스트 애플리케이션 소프트웨어를 더 포함하는

버스 스테이션.

청구항 5

제 1 버스 통신 포트를 포함하는 제 1 버스 스테이션 및 제 2 버스 스테이션을 포함하는 버스 통신 시스템으로서,

상기 제 2 버스 스테이션은 제 2 버스 통신 포트를 더 포함하고,

상기 제 2 버스 스테이션은 상기 제 2 버스 통신 포트에 연결된 호스트 스테이션의 존재를 검출하게 되면 제 1 동작 모드로 동작하고, 상기 제 2 버스 통신 포트에 연결된 호스트 스테이션이 존재하지 않음을 검출하게 되면 제 2 동작 모드로 동작하며,

상기 제 2 버스 통신 포트에 장치 컨트롤러가 연결되되,

상기 장치 컨트롤러는 운영 체제, 및 호스트 컨트롤러와 통신하고 상기 운영 체제와 정보를 주고 받는 호스트 스테이션 드라이버 소프트웨어를 포함하는 시스템 소프트웨어의 제어 하에 동작하고,

상기 시스템 소프트웨어는 상기 호스트 스테이션 드라이버 소프트웨어에 대해서 호스트 컨트롤러의 존재를 애플레이트하고, 상기 장치 컨트롤러에 대해서 장치 스테이션 드라이버 소프트웨어의 존재를 애플레이트하며, 또한 상기 호스트 스테이션 드라이버 소프트웨어로부터 상기 장치 컨트롤러로 혹은 상기 장치 컨트롤러로부터 상기 호스트 스테이션 드라이버 소프트웨어로 통신을 변환하는 호스트 애플리케이션 소프트웨어를 더 포함하는

버스 통신 시스템.

청구항 6

삭제

명세서

기술분야

[0001] 본 발명은 버스 접속 시스템에 관한 것이고, 특히 버스 통신 시스템 내의 전자 장치와 함께 사용되어 이 전자 장치가 시스템 내에서 호스트의 역할을 할 수 있게 하는 장치에 관한 것이다.

배경기술

[0002] USB(Universal Serial Bus) 통신 시스템이 널리 사용되고 있다.

[0003] USB 시스템에서, 개인용 컴퓨터, 스캐너, 무선 전화, 프린터 등과 같은 많은 전자 장치의 아이템들을 상호 접속시킬 수 있다. 어떤 시스템에서든, 하나의 장치의 아이템이 항상 USB 호스트로 지정되어 있어서, 다른 모든 아이템 또는 USB 장치와의 접속을 제어한다. 개인용 컴퓨터에는 통상 USB 호스트의 역할을 가능하게 하는데 필요한 하드웨어 및 소프트웨어가 제공되지만, 다른 아이템들에는 통상 필수 하드웨어 및 소프트웨어가 제공되지 않으므로, 유일하게 USB 장치의 역할을 할 수 있다.

[0004] 그러나, 장치를 크게 수정하지 않고도, USB 호스트의 역할을 할 수 있는 장치의 아이템을 사용해야 하는 상황이 있을 수 있다.

[0005] 본 발명의 일 측면에 따라서, 바람직하게는 하드웨어 동글(dongle)의 형태인 버스 스테이션이 제공되며, 이는 버스 통신 장치의 버스 통신 포트에 접속되어, 버스 통신 장치가 버스 호스트의 역할을 할 수 있게 한다. 본 발명이 다른 버스 통신 시스템에도 적용될 수 있지만, 본 발명의 바람직한 실시예에서 버스 스테이션은 USB 시스템에서 동작한다.

발명의 상세한 설명

[0006] 더 상세하게는, 본 발명의 일 측면은 버스 스테이션을 제공하되, 이 버스 스테이션은 버스 호스트가 그 제 1 버스 통신 포트에 접속되었다고 판정하면, 이 버스 호스트와 그 제 2 버스 통신 포트에 접속된 버스 장치 사이의 규정 버스 통신을 가능하게 하는 송수신기의 역할을 하고, 또한 적절한 소프트웨어를 구동하는 버스 장치가 제

1 버스 통신 포트에 접속되었다고 판정하면 제 1 버스 통신 포트에 접속된 제 1 버스 장치와 제 2 버스 통신 포트에 접속된 버스 장치 사이의 버스 통신을 가능하게 하는 다른 호스트의 역할을 한다.

실시예

- [0010] 도 1은 본 발명에 따른 버스 통신 시스템의 개략 블록도이다.
- [0011] 시스템(2)은 USB 포트(6)를 구비한 제 1 USB 장치(4), 제 2 USB 장치(8) 및 동글(10)을 포함하고 있다.
- [0012] 도시된 실시예에서, 제 1 USB 장치(4)는 PDA이지만, 본 발명이 모바일 통신 장치, 디지털 카메라 또는 전자 수첩과 같은 임의의 USB 장치에 적용될 수 있다는 것을 이해할 것이다.
- [0013] 제 2 USB 장치(8)는 프린터, 마우스, 하드 디스크 또는 모뎀을 포함한 임의의 USB 장치가 될 수 있다.
- [0014] 동글(10)은 제 1 호스트 포트(H1) 및 제 2 호스트 포트(H2)를 구비한 내장형 USB 호스트/장치 컨트롤러(12) 및 저전력 마이크로컨트롤러 장치(MCU:14)를 포함한다.
- [0015] 동글(10)의 제 1 호스트 포트(H1)는 도 1에 도시된 바와 같이 PDA(4)의 USB 포트(6)에 접속된다. 동글(10)의 제 1 호스트 포트(H1)가 PDA(4)의 USB 포트(6)에 접속되면, PDA는 USB 호스트의 역할을 효율적으로 수행할 수 있게 되며, 동글(10)의 제 2 호스트 포트(H2)는 PDA(4)의 호스트 포트로서의 역할을 효율적으로 수행한다. 따라서, PDA(4)는 USB 버스(15)를 사용해서 동글(10)의 제 2 호스트 포트(H2)에 접속된 임의의 USB 장치와의 통신을 제어할 수 있다.
- [0016] 제 1 호스트 포트(H1) 및 제 2 호스트 포트(H2)가 같은 USB 호스트/장치 컨트롤러(12)에 접속되는 것으로 도시되어 있지만, MCU(14)가 독립된 USB 호스트/장치 컨트롤러를 통해서 제 1 호스트 포트(H1) 및 제 2 호스트 포트(H2)와 통신할 수 있다는 것을 이해할 것이며, 여기서 제 1 호스트 포트는 PDA(4)와의 통신 전용이고, 제 2 호스트 포트는 접속된 USB 장치 또는 장치(8)와의 통신 전용이라는 것을 이해할 것이다.
- [0017] PDA(4)가 동글(10)에 접속되었을 때, USB 호스트의 역할을 할 수 있게 하기 위해서, PDA(4)는 드라이버 업데이트를 필요로 한다.
- [0018] 드라이버 업데이트는 사용되는 특정 USB 장치로 특정되는 것으로, PDA(4) 내의 현재의 USB 장치 하드웨어의 최상위에서 구동중인 가상 하드웨어 추상화 계층(Virtual HAL:Virtual Hardware Abstraction Layer) 소프트웨어 드라이버를 추가하는 역할을 한다.
- [0019] 도 2는 도 1의 시스템의 하드웨어 및 소프트웨어의 블록도이다.
- [0020] 종래와 같이 USB 장치(4)는 운영 체제(18), 호스트 스택(20) 및 장치 스택과 장치 하드웨어(22)를 구비하고 있다. USB 호스트로서 역할을 할 수 있기 위해서, PDA(4)는 가상 HAL 드라이버 소프트웨어(16)도 구동한다. USB 장치(4)가 가상 HAL 드라이버 소프트웨어(16)를 구동하고 있을 때, 이는 호스트 온 장치(HostOnDevice)라고 표현된다.
- [0021] 동글(10)은 호스트 하드웨어(12: 즉, USB 호스트 컨트롤러), MCU(14) 및 소프트웨어(24)를 포함한다.
- [0022] 이하 더 상세하게 설명될 소프트웨어 호스트 프로토콜 계층은 동글 커넥터(28) 상에서 장치(4)와 동글(10) 사이의 통신을 제어한다.
- [0023] 도 3은 MCU(14)의 제어하에서, 동글(10)의 동작을 나타내는 흐름도이다. 도 3의 단계(32)에서, 전력이 상승하면 MCU는 단계(34)에서 제 1 호스트 포트(H1)를 조사해서 여기에 어떤 접속이 있는지 판정한다. 아무 것도 접속되어 있지 않다면, 단계(36)로 가서 처리를 종료한다.
- [0024] 무언가 접속되어 있다면, 단계(38)에서 MCU는 USB 호스트가 제 1 호스트 포트(H1)에 접속되어 있는지 측정한다. USB 호스트가 제 1 호스트 포트(H1)에 접속되어 있다면, 처리는 단계(40)로 넘어가고 여기서 동글(10)은 USB 송수신기의 역할을 한다. 즉, 이는 제 1 호스트 포트(H1)와 제 2 호스트 포트(H2) 사이에서 직접 통신해서, 접속된 USB 호스트가 종래의 방식으로 제 2 호스트 포트(H2)에 접속된 임의의 USB 장치(또는 장치)와의 통신을 제어할 수 있게 한다.
- [0025] MCU가 단계(38)에서 USB 호스트가 아닌 USB 장치가 제 1 호스트 포트(H1)에 접속되어 있다고 판정하면, 처리는

단계(41)로 넘어가서 제 2 호스트 포트(H2)에 USB 호스트가 접속되어 있는지 판정한다. 접속되어 있다면, 단계(42)에서, USB 호스트/장치 컨트롤러(12) 내에서 USB 장치 코어가 동작해서, 제 1 호스트 포트(H1)에 접속된 USB 장치와 제 2 호스트 포트(H2)에 접속된 USB 호스트 사이에서 종래의 USB 통신을 가능하게 한다.

- [0026] 제 2 호스트 포트(H2)에 USB 장치가 접속되어 있다면, MCU(14)는 단계(43)에서 제 1 호스트 포트에 접속된 USB 장치의 수를 세고, 이것이 가상 HAL을 구동하는 장치인지 체크한다. MCU가 단계(43)에서 접속된 장치(4)가 구동 가상 HAL이 아니라고 판정하면, 단계(44)에서 장치(4)를 디스에이블시키고, 예컨대 플래싱 LED를 트리거해서 접속된 장치(4)가 가상 HAL을 지원하지 않는다는 신호를 보낸다.
- [0027] 단계(42)에서 MCU가 접속된 장치(4)가 구동 가상 HAL이라고 판정하면(즉, 가상 HAL 드라이버(16)를 갖고 있다고 판정하면), MCU(14)는 단계(46)에서 동작 모드로 들어가서 동글(10:장치(4)와 함께)이 다른 USB 호스트의 역할을 할 수 있게 할 것이다. 이 모드에서, 동글(10)은 제 2 호스트 포트(H2)에 접속된 임의의 USB 장치(또는 장치)와의 통신을 제어할 수 있으며, 이는 이하 더 상세하게 설명될 것이다.
- [0028] 개인용 컴퓨터가 USB 호스트인 종래의 시스템에서, 호스트 스택은 호스트 HAL을 통해서 하위의 USB 하드웨어에 액세스한다. 유사하게, 종래의 PDA USB 장치에서, 장치 스택은 장치 HAL을 통해서 하위의 USB 하드웨어에 액세스한다.
- [0029] 그러나, 본 발명에 따라서, 소프트웨어 호스트 시스템에서, 호스트 스택(또는 호스트 스테이션 드라이버 소프트웨어:20)가 호스트 하드웨어(12)에 액세스하려하면, 이는 가상 HAL 드라이버(16)에 액세스 동작의 세부 사항을 전송한다. 가상 HAL 드라이버(16)는 이들 액세스 동작의 세부 사항을 미리 정해진 소프트웨어 호스트 프로토콜 내로 둘러싼다. 소프트웨어 호스트 프로토콜 패킷은 소프트웨어 호스트 동글(10)이 자신을 아웃스탠드(outstand)하기 위해서 이를 조사할 때 현재의 USB 장치 하드웨어(22)를 통해서 전송된다.
- [0030] 따라서, 가상 HAL 드라이버 소프트웨어(16)는 호스트 스테이션 드라이버 소프트웨어에 대해서 호스트 컨트롤러의 존재를 에뮬레이트한다. 즉, 호스트 스택(20)의 관점에서 가상 HAL 드라이버(16)와의 통신은 종래의 시스템에서의 호스트 HAL과의 통신과 다르지 않다. 호스트 스택(20)은 가상 HAL 드라이버(16)를 통해서 실제 호스트 하드웨어를 볼 것이다.
- [0031] 반대로, 가상 HAL 드라이버 소프트웨어(16)는 장치 컨트롤러(또는 장치 스택:22)에 대해서 장치 컨트롤러의 존재를 에뮬레이트한다. 따라서 가상 HAL 드라이버 소프트웨어(16)는 호스트 스테이션 드라이버 소프트웨어(20)와 장치 컨트롤러 사이의 통신을 변환한다.
- [0032] 소프트웨어 호스트 프로토콜은 다음과 같은 액세스 기능을 제공한다.
- [0033] 동글 호스트 하드웨어(12)에서 레지스터를 판독하는 기능.
- [0034] 동글 호스트 하드웨어(12)에 레지스터를 기록하는 기능.
- [0035] 동글 호스트 하드웨어(12) 내의 버퍼 메모리를 판독하는 기능.
- [0036] 동글 호스트 하드웨어(12) 내의 버퍼 메모리에 기록하는 기능.
- [0037] 예컨대 어떤 값과 AND/OR시키거나, 수정된 값을 레지스터로 재기록시키는 것과 같은 더 개선된 기능이 추가되어 시스템 성능을 개선할 수 있다.
- [0038] 소프트웨어 호스트 프로토콜은 가상 HAL 상에서 수행 중인 호스트 스택(20)이 장치 컨트롤러 하드웨어를 사용하는 호스트 컨트롤러(12)의 하드웨어를 액세스할 수 있는 방법을 정의한다. 소프트웨어 호스트 프로토콜이 이하 상세하게 설명된다. 이 설명에서, 용어 "호스트 동글"은 동글(10)을 나타내는 데 사용되고, 여기서 용어 "호스트 온 장치"는 장치(4), 즉 가상 HAL(16) 상에서 호스트 스택(20)을 구현하는 USB 장치 하드웨어(22)를 구비한 내장형 시스템을 나타내는 데 사용된다.
- [0039] 소프트웨어 호스트 프로토콜은 도 3의 끝에서 시작하고, 여기서 호스트 동글(10)은 접속된 장치(4)의 수를 세고, 접속된 장치(4)는 호스트 온 장치임이 확인된다.
- [0040] 동작 모드에서, MCU(14)는 인터럽트 파이프(an interrupt pipe)를 설정하고, 가상 HAL 드라이버(16)에 대해서 데이터를 수 밀리 초마다 조사한다. 장치(4)와 동글(10) 사이에 전송된 데이터는 소프트웨어 호스트 프로토콜을 통해서 소프트웨어 호스트 패킷의 형태로 전송된다. PDA(4) 상의 호스트 스택(20)이 가상 HAL 드라이버(16)를 통해서 하드웨어 액세스 요청을 전송했다면, 동글(10)의 제 1 호스트 포트(H1)가 인터럽트 파이프를 통해서 이를 조사

할 때, 가상 HAL 드라이버(16)는 요청을 소프트웨어 호스트 패킷으로 전송할 것이다.

- [0041] MCU(14)는 내장된 USB 호스트 컨트롤러(12)의 버퍼 메모리로부터 이 소프트웨어 호스트 패킷을 검색할 것이며, 따라서 하드웨어 액세스를 수행할 것이다. 리턴되는 데이터가 있다면(판독 동작), MCU(14)는 대응하는 데이터를 호스트(1)를 통해서 전송할 것이다.
- [0042] 트래픽
- [0043] 호스트 동글(10) 및 호스트 온 장치(4)는 전용 양방향 벌크 파이프를 사용해서 통신한다. 4가지의 타입의 페이로드가 있다.
- [0044] HRU (호스트 동글 요청 유닛)
- [0045] 호스트 동글(10)에 의해 전송됨
- [0046] 8바이트 페이로드의 벌크 패킷
- [0047] 호스트 온 장치(4)를 조사하는 데 사용됨
- [0048] 인터럽트 정보(HRU_IRQ)를 포함할 수 있음
- [0049] 호스트 동글(10)은 HRU를 전송한 이후에 항상 64바이트의 벌크 인(bulk-in)을 전송함
- [0050] 호스트 온 장치(4)는 이 벌크 인을 통해서 NOB 또는 CRP를 가지고 응답할 것임
- [0051] NOB (아웃스탠딩 비즈니스(Outstanding Business)가 아님)
- [0052] 호스트 온 장치(4)에 의해 전송됨
- [0053] 8바이트 페이로드의 벌크 패킷
- [0054] 아웃스탠딩 트랜잭션이 없을 때 전송됨
- [0055] CRP (공통 요청 패킷)
- [0056] 호스트 온 장치(4)에 의해 전송됨
- [0057] 16-64 바이트의 벌크 패킷
- [0058] 이전에 수신된 CRP 커맨드 및 최적의 데이터 세트의 결과를 포함
- [0059] APR (요청되는 바에 따라서)
- [0060] 호스트 동글(10)에 의해 전송됨
- [0061] 16-64 바이트의 벌크 패킷
- [0062] 플로우
- [0063] 모든 USB 시스템에서와 같이 호스트에 의한 액션으로 전송이 시작한다. 소프트웨어 호스트 프로토콜의 경우에, 소프트웨어 호스트 동글(10)이 항상 호스트이다. 소프트웨어 호스트 전송 사이클은 위에 설명된 바와 같이 HRU를 통해 시작한다. 현재의 전송 사이클은 호스트 동글(10)이 다음 전송 사이클을 개시하기 전에 완료되어야 한다.
- [0064] 폴 낫씽(Poll-nothing) 사이클 : HRU-NOB
- [0065] 호스트 동글(10)은 아웃스탠딩 커맨드 세트에 관해 호스트 온 장치(4)를 조사한다. 아웃스탠딩 커맨드 세트가 없으면, 호스트 온 장치(4)는 NOB로 응답한다.

- [0066] 트랜잭션
- [0067] 호스트 동글(10)이 BULK-OUT를 전송한다.
- [0068] 호스트 동글(10)이 DATA(HRU)를 전송한다.
- [0069] 호스트 온 장치(4)가 ACK를 전송한다.

- [0070] 호스트 동글(10)은 BULK-IN을 전송한다.
- [0071] 호스트 온 장치(4)는 DATA(NOBS)를 전송한다.
- [0072] 호스트 동글(10)은 ACK를 전송한다.

- [0073] 폴 섹션(Poll-something) 사이클: HRU-CRP-APR
- [0074] 호스트 동글(10)은 호스트 온 장치(4)에 대해서 아웃스탠딩 커맨드 세트를 조사한다.
- [0075] 호스트 온 장치(4)는 CRP에 의해 아웃스탠딩 커맨드 세트를 전송한다. 호스트 동글(10)은 커맨드를 수행하고, APR에 의해 결과를 리턴한다.

- [0076] 트랜잭션
- [0077] 호스트 동글(10)은 BULK-OUT을 전송한다.
- [0078] 호스트 동글(10)은 DATA(HRU)를 전송한다.
- [0079] 호스트 온 장치(4)는 ACK를 전송한다.

- [0080] 호스트 동글(10)은 BULK-IN을 전송한다.
- [0081] 호스트 온 장치(4)는 DATA(CRP)를 전송한다.
- [0082] 호스트 동글(10)은 ACK를 전송한다.

- [0083] 호스트 동글(10)은 BULK-OUT을 전송한다.
- [0084] 호스트 동글(10)은 DATA(ARP)를 전송한다.
- [0085] 호스트 온 장치(4)는 ACK를 전송한다.

- [0086] 인터럽트 사이클 : HRU_IRQ-CRP-APR
- [0087] 호스트 동글(10)은 아웃스탠딩 하드웨어 인터럽트시에 호스트 온 장치(4)에 경고한다.
- [0088] 호스트 온 장치(4)는 적절한 커맨드 세트를 결정해서 이를 CRP에 의해 전송한다.
- [0089] 호스트 동글(10)은 커맨드를 수행해서, APR에 의해 결과를 리턴한다.
- [0090] 호스트 온 장치(4)는 아웃스탠딩 인터럽트를 제거하거나 HRU_IRQ의 생성을 불가능하게 해야 하며, 혹은 호스트 동글(10)이 계속해서 HRU_IRQ를 계속해서 전송해야 할 것이다.

- [0091] 트랜잭션

- [0092] 호스트 동글(10)은 BULK-OUT을 전송한다.
- [0093] 호스트 동글(10)은 DATA(HRU_IRQ)를 전송한다.
- [0094] 호스트 온 장치(4)는 ACK를 전송한다.
- [0095] 호스트 동글(10)은 BULK-IN을 전송한다.
- [0096] 호스트 온 장치(4)는 DATA(CRP)를 전송한다.
- [0097] 호스트 동글(10)은 ACK를 전송한다.
- [0098] 호스트 동글(10)은 BULK-OUT을 전송한다.
- [0099] 호스트 동글(10)은 DATA(ARP)를 전송한다.
- [0100] 호스트 온 장치(4)는 ACK를 전송한다.
- [0101] 패킷 포맷
- [0102] HRU 포맷
- [0103] HRU는 다음 정보를 포함한다.
- [0104] 현재 프레임 번호
- [0105] 호스트 동글(10) 내의 호스트 컨트롤러(12)의 HcInterruptStatus
- [0106] 장치 컨트롤러(22)의 인터럽트 상태
- [0107] NOB 포맷
- [0108] 특별히 요구되는 정보 없음
- [0109] CRP 및 APR 포맷
- [0110] 헤더 내의 액티브 비트는 CRP의 경우에 1이고, APR의 경우에 0이다.
- [0111] CRP은 16-64바이트의 크기가 될 수 있다. 전체 크기는
- [0112] 커맨드 세트의 수(각각 8바이트)
- [0113] 선택적인 데이터 세트
- [0114] CRP 내의 최대 커맨드 세트의 수는 8이다.
- [0115] 데이터 세트의 최대 크기는 $64-(8 \times \text{커맨드 세트의 수})$ 이다.
- [0116] 단일 커맨드 요청 패킷 내의 다수의 커맨드 세트는 일련의 하드웨어의 액세스가 단일 패킷 내에서 통신될 수 있게 하고, 따라서 지연 전송을 감소시킨다.
- [0117] 커맨드 세트 포맷
- [0118] 커맨드 세트는 8바이트 데이터 구조이다. 이는 다음 정보를 포함한다.
- [0119] 커맨드 세트 헤더(1바이트)
- [0120] 커맨드 세트 인덱스(2바이트)

[0121] 커맨드 세트 데이터(4바이트)

[0122] 커맨드 세트 보조(1바이트)

[0123]

비트	7	6	5	4	3	2	1	0
그룹	액티브	나머지 세트			OpCode			
속성	불	0-7.0은 최종 세트를 의미함			0-15			

OpCode	MCU에 의한 동작
0	[보조]바이트를 [데이터]로부터 [인덱스] 레지스터에 기록
1	[보조] 바이트를 [인덱스] 레지스터로부터 [데이터]로 판독
2	[데이터] 바이트를 데이터세트로부터 어드레스 위치 [인덱스]에 기록
3	[데이터] 바이트를 어드레스 위치 [인덱스]로부터 데이터세트로 판독
4	[인덱스]레지스터로부터 [보조]바이트를 판독하고 [데이터]와 OR을 수행해서 [인덱스] 레지스터로 재기록함
5	[인덱스]레지스터로부터 [보조] 바이트를 판독하고 [데이터]와 AND를 수행해서 [인덱스] 레지스터로 재기록함
6	폴링 레이트를 [인덱스]로 설정
7	HRU_IRQ 온/오프 설정
8-15	유지됨. MCU에 의한 액션 없음

[0124]

[0125] 따라서, 가상 하드웨어 추상화 계층(가상 HAL)은 USB 장치 하드웨어를 사용해서 접속된 동글 상의 타겟 하드웨어로의 완전한 액세스를 제공한다. 즉, 현재의 USB 장치 하드웨어는 비동기식 마이크로프로세서 인터페이스 버스로서 사용되어 USB 호스트 드라이버가 타겟 하드웨어에 액세스할 수 있게 한다.

[0126] 가상 HAL을 사용함으로써 하드웨어 동글이 USB 소프트웨어를 처리할 필요가 없게 되고, 이로써 동글의 비용이 낮아지고, 호스트 소프트웨어가 USB 장치 상의 내장된 시스템에 의해 처리될 수 있다는 이점을 제공한다.

[0127] 따라서, 본 발명의 바람직한 실시예에서, USB 장치로 하여금 현재의 하드웨어를 변경하는 일 없이 USB 호스트의 성능을 획득할 수 있게 하는 하드웨어 동글이 제공된다. 이를 달성하기 위해서, USB 장치는 USB 장치 상의 내장된 시스템에 의해 처리될 수 있는 애플리케이션 소프트웨어를 구동한다. 이는 하드웨어 동글이 USB 소프트웨어를 처리할 필요가 없어서, 동글이 저렴해진다고 하는 이점을 제공한다.

[0128] 본 발명은 가상 HAL 드라이버 소프트웨어가 USB 장치로 하여금 동글(10)과 관련해서 USB 호스트로서 기능하게 하는 시스템에 대해서 설명되었다. 그러나, 유사한 구조의 가상 HAL 드라이버 소프트웨어가 USB 장치 성능을 사용해서 시스템에 다수의 인터페이스/기능을 추가하는데 사용될 수도 있다. 예컨대, 드라이버 소프트웨어는 USB 장치가 블루투스, IrDa, USB-OTG 또는 다른 통신 프로토콜로 통신할 수 있게 한다.

도면의 간단한 설명

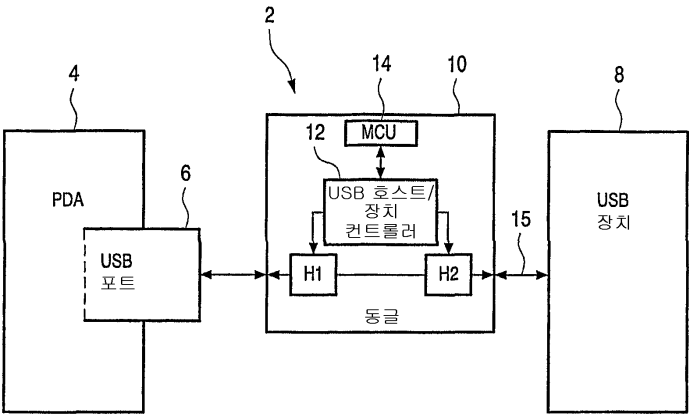
[0007] 도 1은 버스 통신 시스템의 개략 블록도,

[0008] 도 2는 도 1의 시스템의 하드웨어 및 소프트웨어를 도시하는 블록도,

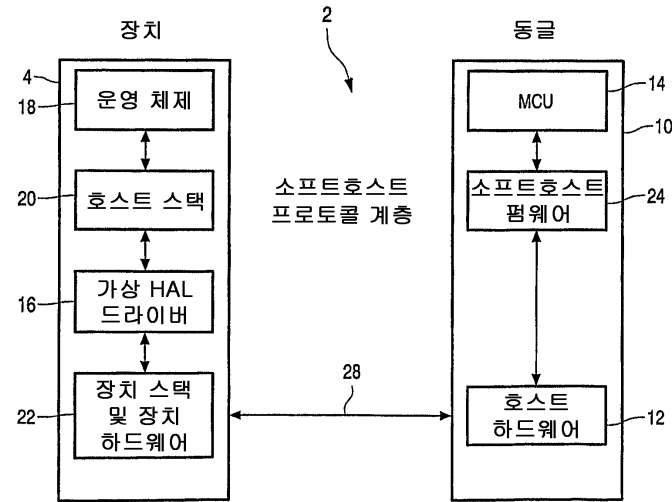
[0009] 도 3은 본 발명의 일 측면에 따른 방법을 도시하는 흐름도.

도면

도면1



도면2



도면3

