



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2013년03월04일  
(11) 등록번호 10-1238099  
(24) 등록일자 2013년02월21일

- (51) 국제특허분류(Int. Cl.)  
*G06F 9/06* (2006.01) *G06F 13/10* (2006.01)
- (21) 출원번호 10-2010-0014029  
(22) 출원일자 2010년02월17일  
심사청구일자 2011년02월17일  
(65) 공개번호 10-2010-0094393  
(43) 공개일자 2010년08월26일  
(30) 우선권주장  
JP-P-2009-035818 2009년02월18일 일본(JP)

## (56) 선행기술조사문헌

EP01248179 A1\*

KR1020070009288 A\*

KR1020000018744 A

KR1020050101495 A

\*는 심사관에 의하여 인용된 문헌

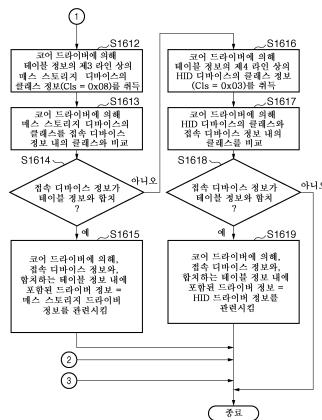
전체 청구항 수 : 총 5 항

심사관 : 복진요

## (54) 발명의 명칭 정보 처리 장치 및 제어 방법

**(57) 요약**

USB 접속에 호환성을 갖는 디바이스와 USB 접속가능한 정보 처리 장치이며, 상기 장치는, 상기 디바이스가 USB 접속되었을 경우에 디바이스 드라이버를 유효화할지 여부를 나타내는 드라이버 정보를 관리하도록 구성되는 관리 유닛을 가진다. 상기 정보 처리 장치는, 상기 디바이스의 디바이스 정보를 취득하도록 구성되는 디바이스 정보 취득 유닛 또한 가진다. 상기 정보 처리 장치는, 상기 드라이버 정보와 상기 디바이스 정보가 합치할 경우, 상기 드라이버 정보에 대응하는 디바이스 드라이버를 사용해서 상기 디바이스를 제어하도록, USB 코어 드라이버의 제어하에 상기 드라이버를 설정하도록 구성되는 설정 유닛 또한 가진다.

**대 표 도** - 도15b

## 특허청구의 범위

### 청구항 1

USB(Universal Serial Bus) 접속에 호환성을 갖는 디바이스와 USB 접속가능한 정보 처리 장치이며,

상기 디바이스의 USB 접속을 유효화하기 위한 유효화 버튼 및 상기 디바이스의 USB 접속을 무효화하기 위한 무효화 버튼을 표시하도록 구성되는 디스플레이 유닛;

등록 유닛

을 포함하며,

상기 등록 유닛은,

상기 유효화 버튼을 누른 경우에, 외부적으로 USB 접속된 디바이스와 내부적으로 USB 접속된 디바이스 둘 다를 사용하기 위해, USB 코어 드라이버 내에, 상기 외부적으로 USB 접속된 디바이스에 대응하는 디바이스 드라이버와 상기 내부적으로 USB 접속된 디바이스에 대응하는 디바이스 드라이버를 등록하고,

상기 무효화 버튼을 누른 경우에, 외부적으로 USB 접속된 디바이스를 사용하지 않고 내부적으로 USB 접속된 디바이스를 사용하기 위해, 상기 USB 코어 드라이버 내에 상기 외부적으로 USB 접속된 디바이스에 대응하는 디바이스 드라이버를 등록하지 않고, 상기 USB 코어 드라이버 내에 상기 내부적으로 USB 접속된 디바이스에 대응하는 디바이스 드라이버를 등록하도록 구성되고,

상기 USB 코어 드라이버는 상기 등록 유닛에 의해 등록된 디바이스 드라이버들 중에 상기 접속된 디바이스에 대응하는 디바이스 드라이버를 특정하고 상기 특정된 디바이스 드라이버를 유효화하는, 정보 처리 장치.

### 청구항 2

제1항에 있어서,

팩스 디바이스

를 더 포함하고,

상기 팩스 디바이스는 내부적으로 USB 접속되며,

상기 내부적으로 USB 접속된 디바이스에 대응하는 디바이스 드라이버는 팩스 드라이버인, 정보 처리 장치.

### 청구항 3

제1항에 있어서,

디바이스의 USB 접속이 유효화되는 경우에 사용되는 유효화 드라이버 리스트 및 디바이스의 USB 접속이 무효화되는 경우에 사용되는 무효화 드라이버 리스트를 유지하도록 구성되는 유지 유닛

을 더 포함하고,

상기 유효화 드라이버 리스트는 외부적으로 USB 접속된 디바이스에 대응하는 디바이스 드라이버와 내부적으로 USB 접속된 디바이스에 대응하는 디바이스 드라이버를 정의하고,

상기 무효화 드라이버 리스트는 외부적으로 USB 접속된 디바이스에 대응하는 디바이스 드라이버를 정의하지 않고 내부적으로 USB 접속된 디바이스에 대응하는 디바이스 드라이버를 정의하며,

상기 유효화 버튼을 누른 경우에, 상기 등록 유닛은 상기 유효화 드라이버 리스트에 기초하여 디바이스 드라이버를 등록하고, 상기 무효화 버튼을 누른 경우에, 상기 등록 유닛은 상기 무효화 드라이버 리스트에 기초하여 디바이스 드라이버를 등록하는, 정보 처리 장치.

### 청구항 4

USB(Universal Serial Bus) 접속에 호환성을 갖는 디바이스와 USB 접속가능한 정보 처리 장치의 제어 방법이며,

상기 정보 처리 장치의 디스플레이 유닛으로 하여금 상기 디바이스의 USB 접속을 유효화하기 위한 유효화 버튼과 상기 디바이스의 USB 접속을 무효화하기 위한 무효화 버튼을 표시하도록 하는 단계;

상기 정보 처리 장치의 등록 유닛으로 하여금,

상기 유효화 버튼을 누른 경우에, 외부적으로 USB 접속된 디바이스와 내부적으로 USB 접속된 디바이스 둘 다를 사용하기 위해, USB 코어 드라이버 내에, 상기 외부적으로 USB 접속된 디바이스에 대응하는 디바이스 드라이버와 상기 내부적으로 USB 접속된 디바이스에 대응하는 디바이스 드라이버를 등록하고,

상기 무효화 버튼을 누른 경우에, 외부적으로 USB 접속된 디바이스를 사용하지 않고 내부적으로 USB 접속된 디바이스를 사용하기 위해, 상기 USB 코어 드라이버 내에 상기 외부적으로 USB 접속된 디바이스에 대응하는 디바이스 드라이버를 등록하지 않고, 상기 USB 코어 드라이버 내에 상기 내부적으로 USB 접속된 디바이스에 대응하는 디바이스 드라이버를 등록하도록 하는 단계

를 포함하고,

상기 USB 코어 드라이버는 상기 등록 유닛에 의해 등록된 디바이스 드라이버들 중에 상기 접속된 디바이스에 대응하는 디바이스 드라이버를 특정하고 상기 특정된 디바이스 드라이버를 유효화하는, 제어 방법.

## 청구항 5

컴퓨터로 하여금,

디바이스의 USB 접속을 유효화하기 위한 유효화 버튼 및 상기 디바이스의 USB 접속을 무효화하기 위한 무효화 버튼을 표시하도록 구성되는 디스플레이 유닛과,

등록 유닛으로서 기능하도록 하는 프로그램을 저장하며,

상기 등록 유닛은,

상기 유효화 버튼을 누른 경우에, 외부적으로 USB 접속된 디바이스와 내부적으로 USB 접속된 디바이스 둘 다를 사용하기 위해, USB 코어 드라이버 내에, 상기 외부적으로 USB 접속된 디바이스에 대응하는 디바이스 드라이버와 상기 내부적으로 USB 접속된 디바이스에 대응하는 디바이스 드라이버를 등록하고,

상기 무효화 버튼을 누른 경우에, 외부적으로 USB 접속된 디바이스를 사용하지 않고 내부적으로 USB 접속된 디바이스를 사용하기 위해, 상기 USB 코어 드라이버 내에 상기 외부적으로 USB 접속된 디바이스에 대응하는 디바이스 드라이버를 등록하지 않고, 상기 USB 코어 드라이버 내에 상기 내부적으로 USB 접속된 디바이스에 대응하는 디바이스 드라이버를 등록하도록 구성되고,

상기 USB 코어 드라이버는 상기 등록 유닛에 의해 등록된 디바이스 드라이버들 중에 상기 접속된 디바이스에 대응하는 디바이스 드라이버를 특정하고 상기 특정된 디바이스 드라이버를 유효화하는, 컴퓨터 판독가능 저장 매체.

## 명세서

### 기술 분야

[0001] 본 발명은 USB 호스트 접속에 의해 하드웨어 확장이 가능한 정보 처리 장치 및 제어 방법에 관한 것이다.

### 배경 기술

[0002] 본 기술 분야에서, USB 호스트 접속은 USB 메모리, 키보드, 마우스 디바이스, 통신 dongle 및 프린터와 같은 다양한 디바이스에 대하여 이용된다. 디지털 복합기도 PC와 마찬가지로 USB 메모리, 키보드 및 카드 리더와 같은 디바이스의 보다 많은 접속을 가능하게 한다. USB 호스트 접속이 보급되고 있는 한가지 이유는, USB 호스트 접속은 범용 물리 I/F(인터페이스)이며, 손쉽게 이용 및 개발될 수 있기 때문이다. 그러나, 범용 물리 I/F이기 때문에, USB 호스트 접속은, 예를 들어 USB 메모리를 통한 정보 누설에 대한 보안 대책을 필요로 한다. 또한, USB 디바이스의 부주의한 접속은 바이러스와 같은 호스트 디바이스에의 악영향을 초래할 수도 있을 것이다. 그러한 위험성을 갖는 디바이스의 접속을 배제하는 것이 중요하다.

[0003] 이러한 문제에 대한 종래의 대책으로서, 유저가 USB 호스트를 제어하는 호스트 컨트롤러 칩을 무효화한다. 예

를 들어, 일반적인 OS(operating system)인 Windows® 및 Linux®도, 모든 USB 호스트 접속을 제어하는 호스트 컨트롤러 칩을 무효화하기 위해서, 호스트 컨트롤러 드라이버의 무효화 또는 삭제를 허용한다. 다른 방법으로서, 유저가 모든 종류의 USB 디바이스를 제어하기 위한 디바이스 드라이버를 무효화/삭제한다. 일본특허공개 제2006-65859호에 개시된 종래 기술은, USB 프로토콜에 존재하는 예약 비트를 사용한다. USB 호스트 디바이스가 접속되었을 때에, 호스트는 예약 비트의 정보에 기초하여 접속 포트를 무효화한다.

### 발명의 내용

#### 해결하려는 과제

- [0004] 유저에 의해 전체 호스트 컨트롤러 칩을 일괄해서 무효화하는 방법을 사용하는 경우에는, 내부 USB 호스트 접속 마저도 원치않게 무효화되는 것과 같은 단점이 존재할 수 있을 것이다.
- [0005] 유저에 의해 각각의 디바이스 드라이버를 무효화하는 방법은 손이 많이 간다. 또한, 보안의 이유로 몇몇 디바이스 드라이버를 일괄적으로 무효화하는 것이 종종 요구된다. 따라서, 이 방법은 불편하다.
- [0006] 일본특허공개 제2006-65859호에서, 예약 비트에 정보가 부가될 필요가 있어서, 접속될 USB 디바이스 또한 이에 대응할 필요가 있다. 이 방법은 의도하지 않은 디바이스의 접속을 배제할 수 없어, 이 역시 적절하지 않다.

#### 과제의 해결 수단

- [0007] 본 발명은 USB 접속에 호환성을 갖는 디바이스와 USB 접속가능한 정보 처리 장치를 제공하며, 이 장치는, 디바이스가 USB 접속되었을 때에 디바이스 드라이버를 유효화할지 여부를 나타내는 드라이버 정보를 관리하도록 구성되는 관리 유닛과, 디바이스의 디바이스 정보를 취득하도록 구성되는 디바이스 정보 취득 유닛과, 드라이버 정보가 디바이스 정보와 합치할 때에, 드라이버 정보에 대응하는 디바이스 드라이버를 사용해서 디바이스를 제어하도록, USB 코어 드라이버의 제어하에 드라이버를 설정하도록 구성되는 설정 유닛을 포함한다.

### 발명의 효과

- [0008] 본 발명의 장점은, 보안에 대한 조치가 미리 취해질 수 있으며, 시스템 내부의 필요한 접속을 무효화하지 않고서 USB 호스트 접속의 편리한 제어가 실행될 수 있다는 것이다. 이러한 메커니즘에 추가하여, 각각의 디바이스 드라이버에 대하여 유효화/무효화가 제어될 수 있어, 보다 편리한 USB 호스트 접속을 달성할 수 있다. 또한, 이러한 메커니즘에 추가하여, 정보 처리 장치의 각각의 물리 I/F에 대하여 유효화/무효화가 제어될 수 있어, 보다 편리한 USB 호스트 접속을 제공할 수 있다.
- [0009] 본 발명은 디바이스의 인터페이스 접속의 유효화/무효화를 적절하게 관리할 수 있다.
- [0010] 본 발명의 다른 장점은, 본 시스템이 몇몇의 USB 호스트 접속을 일괄적으로 무효화할 수 있다는 점이다. 그러나, 유효화된 상태로 남을 필요가 있는 디바이스(예를 들어, 내부적으로 접속될 필요가 있는 디바이스)는 예외적으로 사용될 수 있다.
- [0011] 또한, USB 디바이스측의 호환성에 관계없이, 의도하지 않은 디바이스의 접속이 배제될 수 있다.
- [0012] 본 발명의 추가적인 특징은 첨부된 도면을 참조하는 예시적인 실시예에 대한 아래의 설명으로부터 명확해질 것이다.

### 도면의 간단한 설명

- [0013] 도 1은 본 발명의 실시 형태에서의 시스템 구성의 일례를 설명하는 블록도.
- 도 2는 본 실시 형태에서의 소프트웨어 및 하드웨어를 포함하는 계층 구조를 도시하는 블록도.
- 도 3은 본 실시 형태에서의 코어 드라이버, 디바이스 드라이버 및 USB 디바이스의 관계를 도시하는 블록도.
- 도 4는 본 실시 형태에서의 코어 드라이버, 디바이스 드라이버 및 USB 디바이스의 관계를 도시하는 블록도.
- 도 5는 본 실시 형태에서의 코어 드라이버, 디바이스 드라이버 및 USB 디바이스의 관계를 도시하는 블록도.
- 도 6은 본 실시 형태에서의 코어 드라이버, 디바이스 드라이버 및 USB 디바이스의 관계를 도시하는 블록도.

도 7은 본 실시 형태에서의 USB 프로토콜 정보를 설명하는 도면.

도 8은 본 실시 형태에서의 조작 유닛을 예시하는 도면.

도 9는 본 실시 형태에서의 조작 유닛을 예시하는 도면.

도 10은 본 실시 형태에서의 조작 유닛을 예시하는 도면.

도 11은 본 실시 형태에서 내부 테이블에 유지되는 정보를 도시하는 도면.

도 12는 본 실시 형태에서의 처리 시퀀스의 일례를 설명하는 흐름도.

도 13은 본 실시 형태에서의 처리 시퀀스의 일례를 설명하는 흐름도.

도 14는 본 실시 형태에서의 처리 시퀀스의 일례를 설명하는 흐름도.

도 15a 및 도 15b는 본 발명의 제1 실시 형태에서의 처리 시퀀스의 일례를 설명하는 흐름도.

도 16은 본 발명의 제2 실시 형태에서의 조작 유닛을 예시하는 도면.

도 17은 제2 실시 형태에서의 조작 유닛을 예시하는 도면.

도 18a 및 도 18b는 제2 실시 형태에서의 처리 시퀀스의 일례를 설명하는 흐름도.

도 19는 본 발명의 제3 실시 형태에서의 조작 유닛을 예시하는 도면.

도 20은 제3 실시 형태에서의 처리 시퀀스의 일례를 설명하는 흐름도.

### 발명을 실시하기 위한 구체적인 내용

[0014]

<제1 실시 형태>

[0015]

USB 호스트 접속이 무효화되었을 경우에도 필요한 디바이스가 접속될 수 있는 제1 실시 형태가 아래에 기술될 것이다.

[0016]

<하드웨어 구성>

[0017]

본 실시 형태에 따른 시스템 구성이 도 1을 참조하여 설명될 것이다. 화상 형성 장치(100)는 정보 처리 장치의 일례이다. 리더 유닛(51)은, (예를 들어, 용지상에 제공되는)원고 화상을 광학적으로 판독하고, 이것을 화상 데이터로 변환하는 기능을 갖는다. 리더 유닛(51)은, 원고를 판독하는 기능을 갖는 스캐너 유닛(53)과, 원고 용지를 반송하는 기능을 갖는 원고 급지 유닛(52)을 포함한다. 디바이스 구성에 따라, 원고 급지 유닛(52)이 필요하지 않을 수 있을 것이다[이러한 경우에, 플래튼 글래스(platen glass) 상에 원고가 놓여 지고 센서에 의해 판독될 수 있어, 원고 급지 유닛을 필요로 하지 않는다]. 프린터 유닛(55)은, 기록지를 반송하고, 그 위에 인쇄 화상 데이터를 가시 화상으로서 인자하고, 그 기록지를 장치 외부에 배지하는 기능을 갖는다. 프린터 유닛(55)은 복수의 종류의 기록지 카세트를 갖는 급지 유닛(58)과, 화상 데이터를 기록지에 전사 및 정착시키는 마킹 유닛(56)과, 기록지를 소트(sort), 스테이플(staple)해서 이것을 장치 외부에 출력하는 배지 유닛(57)을 포함한다. 제어 디바이스(60)는 리더 유닛(51) 및 프린터 유닛(55)에 전기적으로 접속된다. 제어 디바이스(60)는, 리더 유닛(51)을 제어하여 원고의 화상 데이터를 판독하고, 프린터 유닛(55)을 제어하여 화상 데이터를 기록지에 출력하는 카피(copy) 기능을 제공한다.

[0018]

조작 유닛(70)은 액정 표시부와, 액정 표시부 상에 접착된 터치 패널 입력 디바이스와, 복수의 하드 키를 포함한다. 조작 유닛(70)은 유저가 다양한 조작을 행할 수 있도록 하는 유저 I/F를 제공한다. 터치 패널 또는 하드 키로부터 입력된 신호는 제어 디바이스(60)에 전달된다. 액정 표시부는 제어 디바이스(60)로부터 송신된 화상 데이터를 표시한다. 리더 유닛(51)에 의해 판독된 원고를 문서로서 HDD(65)에 저장하는 박스(Box) 기능(문서 저장 기능) 또한 제공된다. 이 박스 기능은 문서를 저장하는 기능과, 저장된 문서를 인쇄하는 기능을 포함한다. 제어 디바이스는 휘발성 메모리(67) 및 불휘발성 메모리(66)에도 접속된다. 휘발성 메모리(67)는 일시적인 데이터를 저장하고, 불휘발성 메모리(66)는 영구 데이터를 저장한다. 불휘발성 메모리(66)는 프로그램과, 통상적으로 시스템을 재기동할 때에도 유지될 필요가 있는 데이터의 설정값을 저장한다. 제어 디바이스(60)는 USB 호스트 컨트롤러(80)라고 불리는 칩을 포함한다. 이 USB 호스트 컨트롤러(80)는 USB 호스트 접속파의 모든 통신을 관리한다. 예를 들면, 시스템 구축자는 USB 호스트 허브(89)를 USB 호스트 컨트롤러(80)에 접속한다. USB 호스트 허브(89)의 접속은 외부 USB 블리 인터페이스로서의 역할을 하는 호스트 I/F(83, 84, 85)를 외부적으

로 제공한다. 유저가 이러한 물리 I/F에 키보드(93), 멀티미디어 카드 리더(94) 및 USB 메모리(95)와 같은 디바이스를 접속할 수 있어, 하드웨어를 확장한다. 이것은, IC 카드 리더 및 통신 동글과 같은 다른 USB 디바이스에도 적용된다. 어떤 경우에는, USB 호스트 허브(89)가 내부 USB 디바이스에 접속된다. 예를 들어, 시스템 구축자는, 외부 팩스 회선(300)에 접속하는 USB-팩스 보드(82)와, 화상 해석 보드(81)를 USB 호스트 컨트롤러(80)에 접속한다. 그러면, 시스템은 팩스 기능 및 화상 해석 기능을 실행할 수 있다.

#### [0019] <내부 구성>

도 2는, 제어 디바이스(60) 및 확장 하드웨어 모듈의 구조를 도시하는 블록도이다. 레이어(501, 502)는 소프트웨어 층이며, 레이어(503)는 하드웨어 층이다. 레이어(501)는 일반적으로 유저 스페이스(user space) 또는 유저랜드(userland)로 불리며, 시스템에 대한 특별한 권리를 가지고 있지 않다. 이 레이어에 속하는 모듈은 외부 디바이스를 직접 제어할 수 없다. 레이어(502)는 일반적으로 수퍼바이저 스페이스(supervisor space), 커널 모드(kernel mode) 또는 특권 모드(privileged mode)로 불리며, 시스템에 대한 특별한 권리를 갖는다. 이 레이어에 속하는 모듈은 외부 디바이스를 직접 제어할 수 있다.

확장 하드웨어 모듈은 키보드(93), 멀티미디어 카드 리더(94), USB 메모리(95), USB-팩스 보드(82), 화상 해석 보드(81) 등이다. 예를 들면, 유저는 이러한 모듈[예를 들면, 키보드(93), 멀티미디어 카드 리더(94) 및 USB 메모리(95)]을 도 1에 도시된 물리적인 USB 호스트 I/F(83, 84, 85)에 접속한다. 이와 달리, 시스템 구축자는 이러한 모듈[예를 들면, USB-팩스 보드(82) 및 화상 해석 보드(81)]을 내부적으로 접속한다. USB 호스트 컨트롤러(80)는 모든 USB 호스트 I/F를 제어한다. USB 호스트 컨트롤러(80)는, USB 호스트에 접속된 디바이스(주변 디바이스)와 화상 형성 장치(100)의 본체 사이의 USB 통신을 수행하기 위한 모든 통신을 관리한다. USB 호스트 컨트롤러(80)는 USB 표준 1.x 및 USB 표준 2.0과 같은 각각의 표준에 준거한 통신을 수행할 수 있다. USB 표준 1.x는 UHCI(Universal Host Controller Interface) 및 OHCI(Open Host Controller Interface)를 포함한다. USB 표준 2.0에 대해서는, EHCI(Enhanced Host Controller Interface)가 통일 표준으로서 간주된다. 대부분의 정보 디바이스는 일반적으로 UHCI 통신 또는 OHCI 통신 중 하나와, EHCI 통신을 지원한다.

[0022] 소프트웨어 층에 존재하는 이하의 모듈은, 불휘발성 메모리(66)에 저장되고, CPU(도시되지 않음)에 의해 관리되어 동작한다. 수퍼바이저 스페이스(502)는 USB 호스트 컨트롤러 드라이버(201), USB 호스트 코어 드라이버(202), HID 드라이버(203), 매스 스토리지(Mass Storage) 드라이버(204), 화상 해석 보드 드라이버(211) 및 팩스 드라이버(212)를 포함한다. 유저 스페이스(501)는 드라이버 제어 태스크(250) 및 애플리케이션(220)을 포함한다.

[0023] USB 호스트 컨트롤러 드라이버(201)는 USB 호스트 컨트롤러(80)를 제어하고, USB 프로토콜 레벨에서의 통신을 수행한다. USB 호스트 코어 드라이버(이후에는 코어 드라이버로 칭함)(202)는 USB 호스트 컨트롤러 드라이버(201) 위의 층에 배치된다. 디바이스 드라이버는 코어 드라이버(202) 위의 층에 배치된다. 이 디바이스 드라이버는, 상위층의 유저 스페이스에 존재하는 애플리케이션(220)이 USB 호스트 디바이스와 통신하기 위해 필요한 소프트웨어이다. 이 디바이스 드라이브는 다음과 같다. HID 드라이버(203)는 키보드 또는 마우스와 같은 범용 HID(Human Interface Device)를 제어한다. 통상적으로, HID 드라이버(203)는 키 스트라이크(key strike) 및 마우스 제스처(mouse gesture)를 이벤트로서 상위층의 애플리케이션에 발행한다. 매스 스토리지 드라이버(204)는 USB 메모리 또는 카드 디바이스와 같은 범용 저장 디바이스를 제어한다. 이 매스 스토리지 드라이버(204)는 통상적으로 USB 메모리와 같은 저장 디바이스의 기능을 매스 스토리지 드라이버(204) 상위에 구축된 파일 시스템(도시되지 않음)을 통해 애플리케이션에 제공한다. 팩스 드라이버(212)는 벤더 고유의(vendor-specific) USB-팩스 보드(82)를 제어한다. 팩스 드라이버(212)는, 예를 들어 팩스 회선(300)을 통한 외부와의 팩스 통신을 수행하기 위해 팩스 표준에 기초한 소프트웨어 처리를 수행한다. 화상 해석 보드 드라이버(211)는 벤더 고유의 화상 해석 보드를 제어한다. 화상 해석을 통해, 화상 해석 보드 드라이버(211)는, 예를 들어 주사된 데이터를 해석하기 위한 지시 또는 관리를 포함하는 소프트웨어 처리를 수행한다.

[0024] 특히, 코어 드라이버(202)는 본 발명에 관련되며, 상세하게 설명될 것이다. 통상적으로, 코어 드라이버(202)는 디바이스 드라이버 정보를 유지하고, 접속된 디바이스를 처리할 디바이스 드라이버를 결정한다.

[0025] 보다 구체적으로, 코어 드라이버(202)는 이하의 기능을 가진다.

[0026] 1) 코어 드라이버(202)는 상위층 모듈로부터 디바이스 드라이버의 등록을 수취하고, 디바이스 드라이버 정보를 테이블에 유지한다. 디바이스 드라이버는 언제나 등록될 수 있으며, 필요한 때에 테이블에 추가된다. 통상적으로 휘발성 메모리(67)가 테이블을 유지한다.

- [0027] 2) 코어 드라이버(202)는 USB 디바이스의 접속을 검지하고, 접속된 USB 디바이스의 정보를 취득한다. 통상적으로 휘발성 메모리(67)가 취득된 정보를 유지한다.
- [0028] 3) USB 호스트 디바이스가 접속되거나, 디바이스 드라이버가 등록된 경우에, 코어 드라이버(202)는 테이블과 접속된 USB 디바이스의 정보를 비교한다. 그 후에, 코어 드라이버(202)는, 어느 디바이스 드라이버가 접속된 USB 디바이스를 처리할 것인지를 결정한다.
- [0029] 4) 코어 드라이버(202)는 디바이스 드라이버 삭제 요청을 수취하고, 대응하는 디바이스 드라이버 정보를 테이블로부터 삭제한다.
- [0030] 5) 코어 드라이버(202)는 USB 디바이스의 제거(disconnection)를 검지한다. 마지막으로, 코어 드라이버(202)는 제거된 USB 디바이스의 처리를 중단한다. 본 발명에서의 코어 드라이버(202)는 특히 제1, 제2 및 제3 기능에 관련된다.
- [0031] 드라이버 제어 태스크(250)는 디바이스 드라이버를 코어 드라이버(202) 내에 등록한다. 화상 형성 장치(100)의 시스템이 기동할 때에, 드라이버 제어 태스크(250)는 CPU(도시되지 않음)에 의해 불휘발성 메모리(66)로부터 판독되어 동작한다.
- [0032] <디바이스 드라이버 등록의 시퀀스>
- [0033] 도 3은 드라이버 제어 태스크(250)에 의한 디바이스 드라이버의 코어 드라이버(202)에의 등록을 설명하는 도면이다. 도 3은 USB 호스트 접속이 유효한 경우를 도시한다. 아래의 모듈은 불휘발성 메모리(66)에 저장되고, CPU(도시되지 않음)에 의해 판독되어 동작한다. 이러한 모듈은 USB 호스트 컨트롤러 드라이버(201), 코어 드라이버(202), HID 드라이버(203), 매스 스토리지 드라이버(204), 화상 해석 보드 드라이버(211), 팩스 드라이버(212) 및 드라이버 제어 태스크(250)이다.
- [0034] 드라이버 제어 태스크(250)는 디바이스 드라이버의 리스트를 불휘발성 메모리(66)에 유지한다. 이 리스트는 USB 호스트의 유효화 시의 드라이버 리스트이며, 모든 디바이스 드라이버를 나타낸다. 이 리스트는, 리스트(601)에 의해 나타낸 바와 같이, HID 드라이버(203), 매스 스토리지 드라이버(204), 팩스 드라이버(212) 및 화상 해석 보드 드라이버(211)의 정보를 유지한다. 이때, 코어 드라이버(202)는 등록 순서대로 테이블을 구축한다. 많은 경우에, 상위층 모듈[본 발명에서는 드라이버 제어 태스크(250)]은 벤더 고유의 드라이버를 먼저 등록하고, 그 후에 범용 드라이버를 등록한다. USB 디바이스를 접속할 때의 코어 드라이버(202)의 동작으로서, 코어 드라이버(202)는 등록 순서대로 테이블을 주사한다. 따라서, 먼저 적용될 드라이버가 먼저 등록될 필요가 있다. 이것은, 범용 드라이버가 먼저 등록되면 이것이 먼저 사용될 것이고, 이것이 특수 드라이버 대신 사용될 수 있으므로, 벤더 고유의 드라이버와 같은 특수 드라이버가 사용되지 않을 수 있기 때문이다. 설명된 실시예에서는, 드라이버 제어 태스크(250)는 리스트(601)에 도시된 순서대로 벤더 고유의 드라이버인 팩스 드라이버 및 화상 해석 보드 드라이버를 먼저 유지하고, 그 후에 범용 드라이버인 매스 스토리지 드라이버 및 HID 드라이버를 유지한다. 리스트(602)는 직접 값을 예시하며, 드라이버명을 유지한다. 특히, 벤더 고유의 드라이버에 대하여, 처리될 디바이스를 보다 미세하게 특정하기 위해서 벤더 ID 및 프로덕트 ID가 종종 등록된다. 이러한 벤더 ID 및 프로덕트 ID에 대한 정보를 등록하기 위해서, 드라이버 제어 태스크(250)는 이들을 리스트(602)에 유지한다. 불휘발성 메모리(66)는 시스템의 재기동 후에도 이를 정보를 영속적으로 유지한다. 드라이버 제어 태스크(250)는 이 리스트를 판독하고, 그것을 코어 드라이버(202)에 등록한다. 등록 후에, 테이블이 코어 드라이버(202) 내에 구축되는데, 이것은 테이블(603)의 형태를 떨 수 있을 것이다. 이 테이블(603)은 아래의 순서의 드라이버, 즉, 등록된 팩스 드라이버(212), 화상 해석 보드 드라이버(211), 매스 스토리지 드라이버(204) 및 HID 드라이버(203)의 순서대로 구축된다. 테이블(604)은 직접 값을 예시하며, 리스트(602)로부터의 정보가 테이블(604)과 같이 코어 드라이버(202)에 그대로 등록된다.
- [0035] 도 4는 USB 디바이스의 접속시에 사용될 디바이스 드라이버를 설명하는 도면이다. 도 4는 USB 호스트 접속이 유효인 경우를 도시한다. 이하의 모듈은 불휘발성 메모리(66)에 저장되고, CPU(도시되지 않음)에 의해 판독되어 동작한다. 이들 모듈은 USB 호스트 컨트롤러 드라이버(201), 화상 해석 보드 드라이버(211), 팩스 드라이버(212), 코어 드라이버(202), HID 드라이버(203) 및 매스 스토리지 드라이버(204)이다. 화상 해석 보드(81), USB-팩스 보드(82), 키보드(93) 및 USB 메모리(95)는 접속될 USB 디바이스이며, USB 호스트 컨트롤러(80)는 USB 통신을 제어하는 칩이다.
- [0036] USB 디바이스의 접속시에, 코어 드라이버(202)는 등록 정보와 USB 디바이스의 정보를 코어 드라이버(202) 내에 유지된 테이블의 등록 순서대로 비교한다. 코어 드라이버(202)는 테이블 내의 항목을 순서대로 주사하고, 접속

된 디바이스의 정보와 합치하는 정보를 테이블 내에서 검지하는 때에, 드라이버와 디바이스를 관련시킨다. 테이블 내에 합치하는 정보가 없을 경우에는, 그 USB 디바이스는 처리될 수 없다. 이 경우, 모든 디바이스를 처리할 수 있는 드라이버가 준비되고 테이블의 최후에 등록될 수 있을 것이다. 이와 달리, 유저로부터 드라이버를 요청하는 메시지가 표시될 수 있을 것이다.

[0037] 본 실시 형태는 도 3에 도시된 테이블을 사용한다. 예를 들어, 팩스 드라이버(212), 화상 해석 보드 드라이버(211), 매스 스토리지 드라이버(204) 및 HID 드라이버(203)가 테이블(603) 내에 그 순서대로 등록된다. 테이블(604)은 직접 값을 예시한다. USB 디바이스가 접속된 때에, 코어 드라이버(202)는 테이블에 등록된 정보를 순서대로 주사한다. 예를 들어, 코어 드라이버(202)가 테이블 정보의 제1 라인 상의 USB-팩스 보드(82)의 클래스, 벤더 ID 및 프로덕트 ID 값(예를 들면, Cls=0xff, Vendor=0000, ProdID=1111)을 취득한다. 코어 드라이버(202)는 이 정보를 접속된 USB 디바이스의 클래스, 벤더 ID 및 프로덕트 ID의 정보와 비교한다. 비교된 정보가 서로 합치하는 경우에는 팩스 드라이버(212)가 이 디바이스를 처리한다[USB-팩스 보드(82)가 접속된 때에는 팩스 드라이버(212)와 관련되고, 동작이 종료한다]. 비교된 정보가 서로 합치하지 않는 경우에는 코어 드라이버(202)는 테이블 정보의 제2 라인 상의 화상 해석 보드(81)의 클래스, 벤더 ID 및 프로덕트 ID 값(예를 들면, Cls=0xff, Vendor=0000, ProdID=2222)을 취득한다.

[0038] 코어 드라이버(202)는 이 정보를 접속된 USB 디바이스의 클래스, 벤더 ID 및 프로덕트 ID의 정보와 비교한다. 비교된 정보가 서로 합치하는 경우에는 화상 해석 보드 드라이버(211)가 이 디바이스를 처리한다[화상 해석 보드(81)가 접속된 때에는 화상 해석 보드 드라이버(211)와 관련되고, 동작이 종료한다]. 비교된 정보가 서로 합치하지 않는 경우에는 코어 드라이버(202)는 테이블 정보의 제3 라인 상의 매스 스토리지 디바이스[USB 메모리(95)]의 클래스 값(예를 들면, Cls=0x08)을 취득한다. 코어 드라이버(202)는 이 정보를 접속된 USB 디바이스의 클래스의 정보와 비교한다. 비교된 정보가 서로 합치하는 경우에는 매스 스토리지 드라이버(204)가 이 디바이스를 처리한다[USB 메모리(95)가 접속된 때에는 매스 스토리지 드라이버(204)와 관련되고, 동작이 종료한다]. 비교된 정보가 서로 합치하지 않는 경우에는 코어 드라이버(202)는 테이블 정보의 제4 라인 상의 HID 디바이스 [키보드(93)]의 클래스의 값(예를 들면, Cls=0x03)을 취득한다. 코어 드라이버(202)는 이 정보와 접속된 USB 디바이스의 정보를 비교한다. 비교된 정보가 서로 합치하는 경우에는 HID 드라이버(203)가 이 디바이스를 처리한다[키보드(93)가 접속된 때에는 HID 드라이버(203)와 관련되고, 동작은 종료한다]. 비교된 정보가 서로 합치하지 않는 경우에는 접속된 USB 디바이스를 어떠한 드라이버와도 관련시키지 않고서 동작이 종료한다. 동작을 종료하는 대신에, 모든 디바이스와 관련될 수 있는 드라이버가 준비되고, 테이블의 최후에 등록될 수 있을 것이다. 이 경우에는 디바이스는 항상 드라이버와 관련된다.

[0039] USB 접속에 호환성을 갖는 디바이스에 USB 접속된 정보 처리 장치의 일례로서 화상 형성 장치가 개시되었다. 코어 드라이버(202)는 USB 접속시에 드라이버를 유효화할지 여부를 나타내는 드라이버의 정보를 관리한다. 코어 드라이버(202)는 디바이스의 USB 디바이스 정보를 취득하는 디바이스 정보 취득 모듈로서 기능한다.

[0040] 드라이버 정보와 디바이스 정보가 서로 합치할 때에, 코어 드라이버(202)는 그 드라이버 정보에 대응하는 디바이스 드라이버를 사용해서 디바이스를 제어하기 위해 다음 설정을 행한다. 보다 구체적으로, 코어 드라이버(202)는 USB 코어 드라이버의 제어하에 디바이스 드라이버를 설정하는 설정 모듈의 일례로서 기능한다.

[0041] 도 5는 드라이버 제어 태스크(250)에 의한 디바이스 드라이버의 코어 드라이버(202)에의 등록을 설명하는 도면이다. 도 5는 USB 호스트 접속이 무효인 경우를 도시한다. 다음 모듈은 불휘발성 메모리(66)에 저장되고, CPU(도시되지 않음)에 의해 관독되어 동작한다. 이들 모듈은 USB 호스트 컨트롤러 드라이버(201), 코어 드라이버(202), 화상 해석 보드 드라이버(211), 팩스 드라이버(212), HID 드라이버(203), 매스 스토리지 드라이버(204) 및 드라이버 제어 태스크(250)이다. HID 드라이버(203) 및 매스 스토리지 드라이버(204)는 무효화되어 있는 것으로 간주된다(도 5의 통신 화살표 위에 X 표시로 도시됨).

[0042] 드라이버 제어 태스크(250)는 디바이스 드라이버의 리스트를 불휘발성 메모리(66)에 유지한다. 이 리스트는 USB 호스트의 무효화시의 드라이버 리스트이며, USB 호스트의 무효화시에도 사용될 디바이스에 대한 디바이스 드라이버를 나타낸다. 이 리스트는 USB-팩스 보드(82) 및 화상 해석 보드(81)를 나타내는데, 이들이 내부적으로 USB 호스트를 사용하기 때문이다(이들은 USB 호스트의 무효화의 경우에도 이용가능하다). 따라서, 이 리스트는, 리스트(611)에 도시된 바와 같이, 팩스 드라이버(212) 및 화상 해석 보드 드라이버(211)의 정보를 유지한다(HID 드라이버(203) 및 매스 스토리지 드라이버(204)의 정보는 유지하지 않는다). 본 실시 형태에서, 드라이버 제어 태스크(250)는 팩스 드라이버 및 화상 해석 보드 드라이버를 그 순서대로 리스트에 유지한다. 리스트(612)는 직접 값을 예시하며, 드라이버명을 유지한다. 벤더 ID 및 프로덕트 ID의 정보를 등록하기 위해서, 드

라이버 제어 테스크(250)는 이들을 리스트 내에 유지한다. 불휘발성 메모리(66)는 시스템 재기동 후에도 이들을 영속적으로 유지하는 방식으로 이들 정보를 유지한다. 드라이버 제어 테스크(250)는 이 리스트를 관리하고, 이것을 코어 드라이버(202)에 등록한다. 등록 후에, 테이블(613)에 도시된 바와 같이, 테이블이 코어 드라이버(202) 내에 구축된다. 이때, 테이블은 아래의 순서, 즉 등록된 팩스 드라이버, 그 후에 화상 해석 보드 드라이버의 순서대로 구축된다. 테이블(614)은 직접 값을 예시하며, 리스트(612) 내의 정보가 그대로 등록된다.

[0043] 도 6은 USB 디바이스의 접속시에 사용될 디바이스 드라이버를 설명하는 도면이다. 도 6은 USB 호스트 접속이 무효인 경우를 도시한다. 다음 모듈은 불휘발성 메모리(66)에 저장되고, CPU(도시되지 않음)에 의해 관리되어 동작한다. 이들 모듈은 USB 호스트 컨트롤러 드라이버(201), 화상 해석 보드 드라이버(211), 팩스 드라이버(212), 코어 드라이버(202), HID 드라이버(203) 및 매스 스토리지 드라이버(204)이다. HID 드라이버(203) 및 매스 스토리지 드라이버(204)는 무효화되어 있는 것으로 간주하여 도 6에 도시되지 않는다. 화상 해석 보드(81), USB-팩스 보드(82), 키보드(93) 및 USB 메모리(95)는 접속될 USB 디바이스이며, USB 호스트 컨트롤러(80)는 USB 통신을 제어하는 칩이다.

[0044] USB 호스트 접속시의 동작으로서, 코어 드라이버(202)는 등록 정보와 USB 디바이스의 정보를 코어 드라이버(202) 내에 유지된 테이블의 등록 순서대로 비교한다. 코어 드라이버(202)는 테이블을 순서대로 주사하고, 테이블 내의 USB 디바이스와 합치하는 정보를 검지하는 때에 드라이버와 디바이스를 관련시킨다. 테이블 내에 합치하는 정보가 없을 경우에는 그 USB 디바이스는 처리될 수 없다.

[0045] 본 실시 형태는 도 5에서 등록된 테이블을 사용한다. 예를 들어, 팩스 드라이버(212) 및 화상 해석 보드 드라이버(211)는 그 순서대로 테이블(613)에 등록된다. 테이블(614)은 직접 값을 예시한다. USB 디바이스가 접속된 때에, 코어 드라이버(202)는 이 테이블에 등록된 정보를 순서대로 주사한다. 예를 들어, 코어 드라이버(202)는 테이블 정보의 제1 라인 상의 USB-팩스 보드(82)의 클래스, 벤더 ID 및 프로덕트 ID 값(예를 들어, C1s=0xffff, Vendor=0000, ProdID=1111)을 취득한다. 코어 드라이버(202)는 이 정보와 접속된 USB 디바이스의 클래스, 벤더 ID 및 프로덕트 ID의 정보를 비교한다. 비교된 정보가 서로 합치하는 경우에는 팩스 드라이버(212)와 관련되고, 동작이 종료한다. 비교된 정보가 서로 합치하지 않는 경우에는 코어 드라이버(202)는 테이블 정보의 제2 라인 상의 화상 해석 보드(81)의 클래스, 벤더 ID 및 프로덕트 ID 값(예를 들어, C1s=0xffff, Vendor=0000, ProdID=2222)을 취득한다. 코어 드라이버(202)는 이 정보와 접속된 USB 디바이스의 클래스, 벤더 ID 및 프로덕트 ID의 정보를 비교한다. 비교된 정보가 서로 합치하는 경우에는 화상 해석 보드 드라이버(211)가 이 디바이스를 처리한다[화상 해석 보드(81)가 접속되었을 때에는 화상 해석 보드 드라이버(211)와 관련되고, 동작을 종료한다]. 비교된 정보가 서로 합치하지 않는 경우에는 접속된 USB 디바이스를 어떠한 드라이버에도 관련시키지 않고서 동작이 종료한다. 이 경우에, 예를 들어, USB 메모리(95) 등의 사용은 보안의 이유로 제한될 수 있을 것이다. USB 호스트 접속이 무효화되어, 드라이버를 등록하지 않은 USB 메모리의 사용을 금지한다.

[0046] 도 7은 USB 디바이스의 접속시에 코어 드라이버(202)에 의해 USB 디바이스로부터 취득되는 USB 프로토콜 정보(630)를 도시한다. 도 7에 도시된 정보는, 화상 해석 보드(81), USB-팩스 보드(82), 키보드(93) 또는 USB 메모리(95)와 같은 USB 디바이스가 화상 형성 장치(100)에 접속되었을 때에, 코어 드라이버(202)에 의해 휘발성 메모리(67)에 유지되는 정보이다. 구체적으로, 도 7은 USB 메모리(95)가 접속된 때에 취득되는 정보를 도시한다.

[0047] 프로토콜 정보(630)는 다양한 종류의 정보를 포함한다. 본 발명에 관련된 항목으로서, USB 디바이스의 클래스는 항목(631)과 마찬가지로 C1s=08로 나타난다. 벤더 ID 및 프로덕트 ID는 항목(632)과 마찬가지로 Vendor=04bb 및 ProdID=0c2a로 나타난다. 프로토콜 정보(630)는 항목(650)과 같은 USB 디바이스에 물리적으로 접속된 포트와, 항목(651)과 같은 제품명에 대한 정보를 더 포함할 수 있을 것이다.

[0048] 항목(640)은 드라이버명을 표시한다. USB 디바이스가 접속된 때에는 항목(640)에는 정보가 존재하지 않는다. 도 4 또는 도 6에 도시된 바와 같이 디바이스 드라이버와 USB 디바이스가 관련된 후에, 이 드라이버 정보가 드라이버 정보에 포함된다[예를 들어, Driver=usb-storage(640)].

[0049] <조작 유닛>

[0050] 도 8, 도 9 및 도 10은 화상 형성 장치(100)의 조작 유닛을 도시한다. 기본적으로, 유저로부터 윈도우 조작을 수신한 때에, 조작 유닛의 소프트웨어(도시되지 않음)는 처리를 행하고, 그 후에 결과를 표시한다. 조작 유닛의 소프트웨어는 불휘발성 메모리(66)에 유지되고, CPU(도시되지 않음)에 의해 관리되어 동작한다. 예를 들어, 도 11에 도시된 바와 같이, 조작 유닛의 소프트웨어는 설정값의 재기입을 실행한다.

[0051]

도 8은 조작 유닛의 전형적인 표시를 도시한다. 텁 메뉴 키(401)는 텁 메뉴 윈도우 표시(로그인 윈도우 디스플레이는 도시되지 않음)로 복귀하기 위해 준비된다. 유저가 "마이 포탈(my portal)" 키(402)와 같은 키를 누르는 때에, 특정 로그인 유저와 관련된 정보만을 가지는 윈도우[예를 들어, 유저에 의해서 입력된 잡(job)의 정보]가 나타난다. 설정 키(403)는 환경 정보, 예를 들어 디바이스에 유지된 설정값과, 각각의 애플리케이션 및 각각의 유저에 대한 설정값을 설정하는 윈도우를 표시하는 데에 사용된다. 설정 키(403)를 누름으로써, 유저는 도 9를 참조하여 기술될 바와 같이 USB 호스트 접속 유효화 설정 윈도우를 열 수도 있다. 도 8의 윈도우는 기능키(404 내지 408)(각각 카피, 주사&송신, 주사&저장, 웹브라우저 및 팩스를 참조함) 및 쇼트컷 키(411 내지 414)(각각 카피, 팩스, 박스 및 송신을 참조함)를 갖는다. 등록 키(415)는 맞춤형 윈도우(personalised window)를 편집하는 데에 사용된다. 로그아웃 키(417)는 유저가 유저 로그인 상태를 삭제하고자 할 때에 사용된다. 유저가 로그아웃 키(417)를 누르면 텁 메뉴(도시되지 않음)가 나타난다. 상태 라인(418)은, 진행 중인 잡의 정보의 표시 및 소모품(예를 들면, 토너)의 경고 정보를 위한 영역이다. 시스템 모니터 키(419)는, 유저가 실행 중인 잡 리스트 및 잡 로그 리스트의 정보를 확인할 수 있도록 하는 윈도우를 표시하는 데에 사용된다. 이 윈도우는 표시의 일례일 뿐이며, 본 발명이 이것에 한정되는 것은 아니다.

[0052]

도 9는 도 8의 설정 키(403)를 누를 때에 표시되는 키를 도시한다. 유저가 설정 키를 누를 때에, 다양한 환경 설정이 가능해지지만, 그 설명이 여기서는 생략될 것이다. 도 9는 USB 설정 윈도우를 도시한다. 환경 설정 레이어는 계층 "환경 설정 > 외부 I/F > USB 설정" 내에 존재한다. 윈도우 표시 키(701)는 USB 호스트의 유효화를 설정하는 데에 사용된다. 유저가 이 키를 누르면, 도 10에 도시된 바와 같은 윈도우가 나타난다. 유저가 단기 키(702)를 누르면, 윈도우가 사라지고, 도 8의 윈도우 표시로 복귀한다. USB 호스트 접속 제어 지시가 도 9의 윈도우를 통해 유저로부터 접수된다. 이 윈도우는 단지 표시의 일례일 뿐이며, 본 발명이 이 형식에 한정되는 것은 아니다.

[0053]

도 10은 도 9의 USB 호스트의 유효화를 설정하는 키를 눌렀을 때에 표시되는 윈도우를 도시하며, 도 11은 관련된 내부 데이터를 도시한다. 영역(711)은 설정 항목을 표시한다. 유저는 ON(유효화) 키(712) 또는 OFF(무효화) 키(713)를 누름으로써 유효화/무효화를 설정할 수 있다. OK 키(714)는 유효화/무효화를 확정하는 데에 사용된다. 취소 키(715)는 현재의 입력 설정을 취소하는 데에 사용된다(즉, 이때에 이루어진 변경은 확정되지 않는다). 유저가 OK 키(714) 또는 취소 키(715)를 누르면, 윈도우는 도 9의 윈도우 표시로 복귀한다. 유저가 OK 키(714)를 누르면, 도 11에서 도시된 내부 테이블의 정보 또한 갱신된다. 내부 데이터로서, 테이블(720)에 의해서 표시된 바와 같이 윈도우 계층과 동일한 테이블이 유지된다. 계층 정보로서, 테이블은 정보 "환경 설정 > 외부 I/F > USB 설정"을 가지며, 정보 "USB 호스트의 유효화=ON"이 항목(721) 내에 유지된다. 이 항목(721)의 정보는 유효화 플래그로서 불휘발성 메모리(66)에 유지된다. 실제 설정은 시스템이 재기동한 후에 시스템에서 유효하게 된다. 도 10의 윈도우는 단지 표시의 일례일 뿐이며, 본 발명이 이 형식에 한정되는 것은 아니다.

[0054]

<USB 호스트 유효화 시퀀스>

[0055]

도 12는 USB 호스트가 사용가능한지 여부를 제어하는 제어 정보의 역할을 하는 USB 호스트 접속 유효화 플래그의 ON/OFF 상태를 설정하는 때의 처리를 도시하는 흐름도이다. S401에서, 도 10의 USB 호스트 유효화 윈도우가 유저 지시에 따라 표시된다. S402에서, 유저가 도 10의 ON(유효화) 키(712)를 눌렀는지 여부가 판정된다. 유저가 ON(유효화) 키(712)를 누른 경우에는 프로세스는 S403으로 진행하고, 이 제어 정보를 비휘발성 메모리(67)에 유지한다. S402에서 유저가 ON(유효화) 키(712)를 누르지 않은 것으로 판정된 경우에는 프로세스는 S404로 진행하고, 유저가 OFF(무효화) 키(713)를 눌렀는지 여부를 판정한다. 유저가 OFF(무효화) 키(713)를 누른 경우에는 프로세스는 S405로 진행하고, 마찬가지로 휘발성 메모리(67)에 이 제어 정보를 유지한다. S404에서 유저가 OFF(무효화) 키(713) 또한 누르지 않은 것으로 판정된 경우에는 프로세스는 S401로 복귀한다. S406에서, ON(유효화) 키(712) 또는 OFF(무효화) 키(713)의 제어 정보를 확정할 것인지, 또는 취소할 것인지가 판정된다. S406에서 OK 키(714)를 눌러서 제어 정보를 확정하는 것으로 판정된 경우에는, S407에서 유효화 플래그가 확정되어 불휘발성 메모리(66)[도 11의 항목(721)]를 갱신한다. S406에서 유저가 취소 키(715)를 누른 것으로 판정된 경우에는 프로세스는 S401로 복귀한다. 이러한 시퀀스에 따라서, USB 호스트 접속 제어 지시가 유저로부터 접수되어 설정을 수행한다.

[0056]

<시스템 기동시 처리 시퀀스>

[0057]

도 13은 디바이스 드라이브 등록의 시스템이 기동하는 때의 처리의 흐름도이다. S501에서 시스템이 기동한다. S502에서, 드라이버 제어 태스크(250)는 USB 호스트 접속 유효화 플래그를 참조한다. 유효화 플래그는 도 10

및 도 11을 참조하여 설명된 유효화의 ON/OFF 설정을 나타낸다. S503에서 유효화 플래그가 ON인 것으로 판정된 경우에는, S504에서 모든 디바이스 드라이버가 코어 드라이버(202) 내에 등록된다. 예를 들어, 본 실시 형태에서 이들 디바이스 드라이버는 팩스 드라이버(212), 화상 해석 보드 드라이버(211), 매스 스토리지 드라이버(204) 및 HID 드라이버(203)이다. S503에서 유효화 플래그가 OFF인 것으로 판정되면, S505에서 미리 등록된 디바이스 드라이버만이 코어 드라이버(202)에 등록된다. 본 실시 형태에서, 팩스 드라이버(212) 및 화상 해석 보드 드라이버(211)만이 등록된다. USB-팩스 보드(82) 및 화상 해석 보드(81)는 범용 드라이버가 아니라 벤더 고유의 드라이버를 사용한다. 따라서, 도 3에 도시된 바와 같이, 클래스, 벤더 ID 및 프로덕트 ID의 정보와, 드라이버 정보가 세트로 등록된다. 예를 들면, Cls=0xff, Vendor=0000, ProdID=1111, Driver=usb-fax가 팩스 디바이스에 대해 등록된다. 이와 대조적으로, 매스 스토리지 디바이스 및 HID 디바이스는 범용 드라이버를 사용할 수 있기 때문에, 클래스 및 드라이버 정보만이 등록되는 것으로 충분하다(예를 들면, 매스 스토리지 디바이스에 대해 Cls=0x08, Driver=usb-storage). 이 드라이버의 정보는 단지 일례일 뿐이다.

#### [0058] <USB 디바이스 접속이 검지된 때의 처리 시퀀스>

도 14는 USB 디바이스(예를 들어, 키보드, USB 메모리 또는 팩스)가 접속되는 때의 처리의 흐름도이다. S601에서, 유저가 USB 디바이스를 접속하는 때에, 또는 USB 디바이스가 접속된 동안 프로세스가 기동하는 때에, 또는 USB 디바이스가 시스템에 의해 승인되는 때에 시스템은 기동한다. S602에서, 코어 드라이버(202)는 디바이스 접속을 검지한다. S603에서 코어 드라이버(202)는 접속된 USB 디바이스의 디바이스 정보를 취득한다. 여기서 취득할 수 있는 정보는 도 7에 도시된 프로토콜 정보에 포함된다. S604에서, 코어 드라이버(202)는 내부 테이블을 주사하고, 취득된 USB 디바이스 정보와 테이블 내의 정보를 비교한다. S605에서, 코어 드라이버(202)는 취득한 디바이스 정보가 테이블 내의 정보와 합치하는지 여부를 판정한다. 이를 정보가 서로 합치하는 경우에는, S606에서 코어 드라이버(202)는 접속된 USB 디바이스 정보와, 합치하는 테이블 정보 내의 드라이버 정보를 관련시켜, 그 디바이스가 사용될 수 있다. S605에서 접속된 USB 디바이스 정보가 어떠한 정보와도 합치하지 않는 경우에는, 접속된 USB 디바이스 정보를 드라이버 정보와 관련시키지 않고서 프로세스는 종료한다. 이 USB 디바이스 정보는 단지 디바이스 정보의 일례일 뿐이다.

#### [0060] <USB 디바이스 접속이 검지된 때의 처리 시퀀스(상세)>

도 15a 및 도 15b는 도 14의 흐름도의 처리를 보다 상세하게 도시한다. S1601에서, 유저가 USB 디바이스를 접속한 때, 또는 USB 디바이스가 접속된 동안에 시스템이 기동한다. S1602에서, 코어 드라이버(202)는 디바이스 접속을 검지한다. S1603에서, 코어 드라이버(202)는 접속된 USB 디바이스의 디바이스 정보를 취득한다. 여기서 취득할 수 있는 정보는 도 7에 도시된 프로토콜 정보에 포함된다. S1604 및 후속 단계는 코어 드라이버(202)에 의해 코어 드라이버(202) 내의 테이블을 주사하고, 취득된 USB 디바이스 정보와 테이블 내의 정보를 비교하고, 이것을 합치하는 정보와 관련시키는 처리를 설명할 것이다. 본 실시 형태에서, 팩스 디바이스, 화상 해석 보드 등이 예시될 것이다. 그러나, 디바이스는 이것에 한정되지 않으며, 본 발명은 USB 인터페이스를 가지는 어떠한 디바이스에도 적용가능하다.

S1604에서, 코어 드라이버(202)는 테이블 정보의 제1 라인 상의 USB-팩스 보드(82)의 클래스, 벤더 ID 및 프로덕트 ID 값(예를 들어, Cls=0xff, Vendor=0000, ProdID=1111)을 취득한다. S1605에서, 코어 드라이버(202)는 이 정보와 도 7의 프로토콜 정보에 의해 나타난 바와 같은 USB 디바이스의 클래스, 벤더 ID 및 프로덕트 ID의 정보를 비교한다. S1606에서, 코어 드라이버(202)는 이를 정보가 서로 합치하는지 여부를 판정한다. 이를 정보가 서로 합치하는 경우에는 프로세스는 S1607로 진행한다. 코어 드라이버(202)는 접속된 USB 디바이스 정보와, 합치하는 테이블 정보에 포함된 드라이버 정보[이 경우에는 팩스 드라이버(212)의 정보]를 관련시킨다. 이러한 관련은, 이 팩스 드라이버(212)가 접속된 USB 디바이스를 처리할 수 있도록 한다. 그 후에, 프로세스는 종료한다.

S1606에서, 접속된 USB 디바이스 정보가 등록된 USB-팩스 보드(82)의 정보와 합치하지 않는 것으로 코어 드라이버(202)가 판정한 경우에는 프로세스는 S1608으로 진행하고, 접속된 USB 디바이스 정보를 다음 테이블 정보와 비교한다. S1608에서, 코어 드라이버(202)는 테이블 정보의 제2 라인 상의 화상 해석 보드(81)의 클래스, 벤더 ID 및 프로덕트 ID 값(예를 들어, Cls=0xff, Vendor=0000, ProdID=2222)을 취득한다. S1609에서, 코어 드라이버(202)는 이 정보와, 도 7의 프로토콜 정보에 의해 나타난 바와 같은 USB 디바이스의 클래스, 벤더 ID 및 프로덕트 ID의 정보를 비교한다. S1610에서, 코어 드라이버(202)는 이를 정보가 서로 합치하는지 여부를 판정한다. 이를 정보가 서로 합치하는 경우에는 프로세스는 S1611로 진행한다. 코어 드라이버(202)는 접속된 USB 디바이스 정보를 합치하는 테이블 정보에 포함되는 드라이버 정보[이 경우에는 화상 해석 보드 드라이버(211)의 정

보]와 관련시킨다. 이러한 관련은, 화상 해석 보드 드라이버(211)가 접속된 USB 디바이스를 처리할 수 있도록 한다. 그 후에, 프로세스는 종료한다. S1610에서, 접속된 USB 디바이스 정보가 등록된 화상 해석 보드(81)의 정보와 합치하지 않는 것으로 코어 드라이버(202)가 판정한 경우에는 프로세스는 S1612로 진행하고, 접속된 USB 디바이스 정보를 다음 테이블 정보와 비교한다. S1612에서, 코어 드라이버(202)는 테이블 정보의 제3 라인 상의 매스 스토리지 디바이스[멀티미디어 카드 리더(94)/USB 메모리(95)]의 클래스 값(예를 들어, C1s=0x08)을 취득한다. S1613에서, 코어 드라이버(202)는 이 정보와, 도 7의 프로토콜 정보에 의해 나타난 바와 같은 USB 디바이스의 클래스의 정보를 비교한다. S1614에서, 코어 드라이버(202)는 이들 정보가 서로 합치하는지 여부를 판정한다. 이들 정보가 서로 합치하는 경우에는 프로세스는 S1615로 진행한다. 코어 드라이버(202)는 접속된 USB 디바이스 정보와, 합치하는 테이블 정보에 포함된 드라이버 정보[이 경우에는 매스 스토리지 드라이버(204)의 정보]를 관련시킨다. 이러한 관련은, 매스 스토리지 드라이버(204)가 접속된 USB 디바이스를 처리할 수 있도록 한다. 그 후에, 프로세스는 종료한다.

[0064] S1614에서, 접속된 USB 디바이스 정보가 등록된 매스 스토리지 디바이스[멀티미디어 카드 리더(94)/USB 메모리(95)]의 정보와 합치하지 않는 것으로 코어 드라이버(202)가 판정한 경우에는 프로세스는 S1616으로 진행하고, 접속된 USB 디바이스 정보와 다음 테이블 정보를 비교한다. S1616에서, 코어 드라이버(202)는 테이블 정보의 제4 라인 상의 HID 디바이스[키보드(93)]의 클래스 값(예를 들어, C1s=0x03)을 취득한다. S1617에서, 코어 드라이버(202)는 이 정보와, 도 7의 프로토콜 정보에 의해 나타난 바와 같은 USB 디바이스의 클래스의 정보를 비교한다. S1618에서, 코어 드라이버(202)는 이들 정보가 서로 합치하는지 여부를 판정한다. 이들 정보가 서로 합치하는 경우에는 프로세스는 S1619로 진행한다. 코어 드라이버(202)는 접속된 USB 디바이스 정보와 합치하는 테이블 정보에 포함된 드라이버 정보[이 경우에는 HID 드라이버(203)의 정보]를 관련시킨다. 이러한 관련은, HID 드라이버(203)가 접속된 USB 디바이스를 처리할 수 있도록 한다. 그 후에, 프로세스는 종료한다. S1618에서, 접속된 USB 디바이스 정보가 등록된 HID 디바이스[키보드(93)]의 정보와 합치하지 않는 것으로 코어 드라이버(202)가 판정한 경우에는, 프로세스는 종료한다. 이 경우에, 접속된 USB 디바이스를 처리할 수 있는 드라이버는 주사된 테이블에 등록되지 않는다. 따라서, 디바이스는 드라이버와 관련될 수 없고, 접속된 USB 디바이스는 사용될 수 없다.

[0065] 이러한 처리에 의해서, 시스템은 USB 호스트 접속을 일괄적으로 무효화할 수 있다. 추가적으로, 의도하지 않는 디바이스의 접속은 USB 디바이스 측의 호환성에 관계없이 배제될 수 있다.

[0066] <제2 실시 형태>

[0067] USB 호스트 접속을 무효화한 후에도 사용가능한 디바이스의 접속을 허용하면서, USB 호스트 접속이 유효화된 때에 각각의 디바이스에 대하여 유효화/무효화가 설정될 수 있는 제2 실시 형태가 설명될 것이다. 제2 실시 형태의 대부분의 설명은 제1 실시 형태의 설명과 중복되므로, 차이점만이 설명될 것이다. 도 16 및 도 17의 조작 유닛 상의 표시에 대한 프로세스는 제1 실시 형태의 프로세스와 동일하다.

[0068] <조작 유닛>

[0069] 도 16은 제1 실시 형태에서 설명된 도 8의 설정 키(403)를 눌렀을 때에 표시되는 윈도우 및 키를 도시한다. 이러한 윈도우는, 특히 관리자에 의해서 조작가능한 모든 USB 디바이스에 대한 설정 윈도우이다. 관리자가 설정 키를 누르면, 다양한 환경 설정이 가능해진다. 도 16은 USB 관리자 설정 윈도우를 예시한다. 환경 설정 레이어는, 계층 "환경 설정 > 외부 I/F > USB 설정 > USB 관리자 설정" 내에 존재한다. 윈도우 표시 키(801)는 USB 호스트의 유효화를 설정하는 데에 사용된다. 윈도우 표시 키(802)는 USB-팩스 보드(82)의 유효화를 설정하는 데에 사용될 수 있다. 윈도우 표시 키(803)는 화상 해석 보드(81)의 유효화를 설정하는 데에 사용될 수 있다. 윈도우 표시 키(804)는 USB 매스 스토리지 디바이스[예를 들어, 멀티미디어 카드 리더(94)/USB 메모리(95)]의 유효화를 설정하는 데에 사용될 수 있다. 윈도우 표시 키(805)는 USB HID 디바이스[예를 들어, 키보드(93)]의 유효화를 설정하는 데에 사용될 수 있다. 관리자가 이들 키 중 하나를 누르면, 관리자가 유효화/무효화를 설정 할 수 있도록 하기 위해, 제1 실시 형태에서 설명된 도 10에 도시된 바와 같은 윈도우가 나타난다. 관리자가 단기 키(806)를 누르면, 이 윈도우는 사라지고, 도 8의 윈도우 표시로 복귀한다. 도 16의 윈도우는 단지 표시의 일례일 뿐이며, 본 발명은 이러한 형식에 한정되지 않는다.

[0070] 도 17은 제1 실시 형태에서 설명된 도 8의 설정 키(403)를 눌렀을 때에 표시되는 윈도우 및 키를 설명하는 도면이다. 이 윈도우는 일반 유저에 의해 조작가능한 특정 USB 디바이스의 설정 윈도우이다. 일반 유저는 관리자와는 달리 모든 디바이스를 조작할 수는 없으며, 단지 몇몇 기능만을 설정할 수 있다. 일반 유저가 설정 키를 누르는 때에, 다양한 환경 설정이 가능해지지만, 본 실시 형태에서는 그 설명은 생략될 것이다. 도 17은 USB

일반 유저 설정 윈도우를 예시한다. 환경 설정 레이어는 계층 "환경 설정 > 외부 I/F > USB 설정 > USB 일반 유저 설정"에 존재한다. 윈도우 표시 키(814)는 USB 매스 스토리지 디바이스[예를 들어, 멀티미디어 카드 리더(94)/USB 메모리(95)]의 유효화를 설정하는 데에 사용될 수 있다. 윈도우 표시 키(815)는 USB HID 디바이스[예를 들어, 키보드(93)]의 유효화를 설정하는 데에 사용될 수 있다. 일반 유저가 이들 키 중 하나를 누르면, 제1 실시 형태에서 설명된 도 10에 도시된 바와 같은 윈도우가 나타나서, 일반 유저가 유효화/무효화를 설정할 수 있도록 한다. 일반 유저가 단기 키(816)를 누르면, 이 윈도우가 사라지고, 도 8의 윈도우 표시로 복귀한다. 도 17의 윈도우는 단지 표시의 일례일 뿐이며, 본 발명은 이 형식에 한정되지 않는다.

[0071] <시스템 기동시의 처리 시퀀스>

도 18a 및 도 18b는 디바이스 드라이버를 등록하기 위한 시스템이 기동할 때의 처리의 흐름도이다. S701에서, 시스템은 기동한다. S702에서, 드라이버 제어 태스크는 USB 접속 유효화 플래그를 참조한다. 이 유효화 플래그는 제1 실시 형태에서 도 10 및 도 11을 참조하여 설명된 유효화의 ON/OFF 설정을 나타낸다. S703에서 유효화 플래그가 OFF인 것으로 판정되면, S704에서 드라이버 제어 태스크는 미리 등록된 디바이스 드라이버만을 코어 드라이버(202)에 등록한다. 예를 들어, 제2 실시 형태에서, 드라이버 제어 태스크는 팩스 드라이버(212) 및 화상 해석 보드 드라이버(211)만을 등록한다. S703에서 유효화 플래그가 ON인 것으로 판정된 경우에는 프로세스는 S705로 진행하고, 각각의 디바이스의 플래그를 판정하는 처리로 이동한다. S705에서 팩스 디바이스의 유효화 플래그가 OFF인 것으로 판정된 경우에, 프로세스는 S706으로 진행하고, 팩스 드라이버(212)를 등록하지 않도록 드라이버 제어 태스크 내에 정보를 유지한다. 그 후에, 프로세스는 S708로 진행한다. S705에서 팩스 디바이스의 유효화 플래그가 ON인 것으로 판정된 경우에, 프로세스는 S707로 진행하고, 팩스 드라이버(212)를 등록하도록 드라이버 제어 태스크 내에 정보를 유지한다. 그 후에, 프로세스는 S708로 진행한다.

[0073] S708에서 화상 해석 보드 디바이스의 유효화 플래그가 OFF인 것으로 판정된 경우에 프로세스는 S709로 진행하고, 화상 해석 보드 드라이버를 등록하지 않도록 드라이버 제어 태스크 내에 정보를 유지한다. 그 후에, 프로세스는 S711로 진행한다. S708에서 화상 해석 보드 디바이스의 유효화 플래그가 ON인 것으로 판정된 경우에 프로세스는 S710으로 진행하고, 화상 해석 보드 드라이버(211)를 등록하도록 드라이버 제어 태스크 내에 정보를 유지한다. 그 후에 프로세스는 S711로 진행한다. S711에서 매스 스토리지 디바이스의 유효화 플래그가 OFF인 것으로 판정된 경우에 프로세스는 S712로 진행하고, 매스 스토리지 드라이버(204)를 등록하지 않도록 드라이버 제어 태스크 내에 정보를 유지한다. 그 후에 프로세스는 S714로 진행한다. S711에서 매스 스토리지 디바이스의 유효화 플래그가 ON인 것으로 판정된 경우에 프로세스는 S713으로 진행하고, 매스 스토리지 드라이버(204)를 등록하도록 드라이버 제어 태스크 내에 정보를 유지한다. 그 후에 프로세스는 S714로 진행한다. S714에서 HID 디바이스의 유효화 플래그가 OFF인 것으로 판정된 경우에 프로세스는 S715로 진행하고, HID 드라이버(203)를 등록하지 않도록 드라이버 제어 태스크 내에 정보를 유지한다. S714에서 HID 디바이스의 유효화 플래그가 ON인 것으로 판정된 경우에 프로세스는 S716으로 진행하고, HID 드라이버를 등록하도록 드라이버 제어 태스크 내에 정보를 유지한다. S717에서, 등록을 위하여 코어 드라이버(202) 내에 유지된 정보에 기초하여, 드라이버 제어 태스크는 코어 드라이버(202)에 디바이스 드라이버의 정보를 등록한다. 그 후에, 프로세스는 종료한다.

[0074] 이러한 처리에 의해, 시스템은 하나 이상의 USB 호스트 접속을 일괄하여 무효화할 수 있다. 또한, 각각의 USB 디바이스에 대하여 유효화/무효화가 설정될 수 있다. USB 디바이스 측의 호환성에 관계없이, 의도하지 않는 디바이스의 접속이 배제될 수 있다.

[0075] <제3 실시 형태>

[0076] USB 호스트 접속을 무효화한 후에도 사용할 디바이스가 접속될 수 있도록, 각각의 물리 USB 호스트 I/F에 대하여 유효화/무효화가 설정될 수 있는 제3 실시 형태가 이하에 설명될 것이다. 제3 실시 형태의 대부분의 설명은 제1 실시 형태의 설명과 중복되므로, 차이점만 설명될 것이다. 도 19의 조작 유닛에의 표시를 위한 프로세스는 제1 실시 형태의 프로세스와 동일하다.

[0077] 도 19는 제1 실시 형태에서 설명된 도 8의 설정 키(403)를 눌렀을 때에 표시되는 윈도우 및 키를 예시한다. 이 윈도우는 관리자에 의해서 조작 가능한 모든 USB 호스트 I/F에 대한 설정 윈도우이다. 관리자가 설정 키를 누르면, 다양한 환경 설정이 가능해지지만, 본 실시 형태에서 그 설명은 생략될 것이다. 도 19는 USB 관리자 설정 윈도우를 도시한다. 환경 설정 레이어는 계층 "환경 설정 > 외부 I/F > USB 설정 > USB 관리자 설정"에 존재한다. 윈도우 표시 키(821)는 USB 호스트의 유효화를 설정하기 위한 항목을 나타낸다. 윈도우 표시 키(822, 823, 824, 825)는 정보 처리 장치의 물리적 USB 포트 1, 2, 3, 4의 유효화를 설정하는 데에 사용될 수 있다.

이 포트명은, "본체 옆 포트"와 같은 유저에게 친근한 표현일 수 있을 것이다. 관리자가 이들 키 중 하나를 누를 때에, 윈도우는 각각의 포트에 대응하는 설정 윈도우로 이동한다. 관리자가 닫기 키(826)를 누를 때에, 윈도우는 사라지고 도 8의 윈도우 표시로 복귀한다. 도 19의 설정 윈도우는 단지 표시의 일례일 뿐이며, 전술한 기능이 구현될 수만 있다면 그 형식은 상관없다. 제3 실시 형태는 제1 또는 제2 실시 형태와 조합되어 전술한 계층 및 윈도우를 변경할 수 있을 것이다.

#### [0078] <디바이스 접속시의 처리 시퀀스>

도 20은 USB 디바이스(예를 들어, 키보드, USB 메모리 또는 팩스)가 접속된 때의 처리의 흐름도이다. S801에서, 유저가 USB 디바이스를 접속하는 때, 또는 USB 디바이스가 접속된 동안에 시스템이 기동한다. S802에서, 코어 드라이버(202)는 이 USB 디바이스 접속을 검지한다. S803에서, 드라이버 제어 태스크는 각각의 포트에 대해 사용되는 유효화 플래그를 참조한다. S804에서, 코어 드라이버(202)는 도 7에서 설명된 프로토콜 정보에 포함된 USB 디바이스에 접속된 포트의 정보를 취득한다. S805에서, 코어 드라이버(202)는 S803에서 취득된 유효화 플래그와 포트 정보를 비교하고, S806에서 접속된 포트가 사용가능한지 여부를 판정한다. 접속된 포트가 사용가능하지 않을 경우에는(유효화 플래그가 OFF) 프로세스는 종료한다. 접속된 포트가 사용가능한 경우에는(유효화 플래그가 ON) 프로세스는 S808로 진행한다. S808에서, 코어 드라이버(202)는 내부 테이블을 주사하고, 취득된 디바이스 정보와 테이블 내의 정보를 비교한다. S809에서, 코어 드라이버(202)는 테이블 정보가 디바이스 정보와 합치하는지 여부를 판정한다. 이를 정보가 서로 합치하는 경우에는, 프로세스는 S810로 진행한다. 코어 드라이버(202)는 접속된 USB 디바이스의 디바이스 정보와 테이블 내의 드라이버 정보를 관련시켜, 디바이스가 사용될 수 있다. 그 후에, 프로세스는 종료한다. 코어 드라이버(202)가, S809에서 접속된 USB 디바이스가 어떠한 드라이버 정보와도 합치하지 않는 것으로 판정하거나, S806에서 접속된 포트가 이용가능하지 않은 것으로 판정한 경우에는, 프로세스는 종료한다.

[0080] 이러한 처리에 의해, 정보 처리 장치의 각각의 USB 물리 I/F에 대해 유효화/무효화가 제어될 수 있다. 유저에 의해 의도된 보다 편리한 USB 호스트 접속이 이루어질 수 있다.

#### [0081] 기타 실시예

[0082] 본 발명의 양태는, 전술한 실시예(들)의 기능을 수행하기 위하여 메모리 디바이스에 기록된 프로그램을 관독하고 실행하는 시스템 또는 장치의 컴퓨터(또는 CPU 또는 MPU와 같은 디바이스)에 의해, 그리고, 예를 들어 전술한 실시예(들)의 기능을 수행하기 위하여 메모리 디바이스에 기록된 프로그램을 관독하고 실행함으로써 시스템 또는 장치의 컴퓨터에 의해 단계들이 수행되는 방법에 의해 구현될 수도 있다. 이를 위하여, 예를 들어 네트워크를 통해, 또는 메모리 디바이스로서 기능하는 다양한 형태의 기록 매체(예를 들어, 컴퓨터 관독가능 매체)로부터 프로그램이 컴퓨터에 제공된다.

[0083] 본 발명이 예시적인 실시예들을 참조하여 기술되었지만, 본 발명이 전술한 예시적인 실시예들에 한정되는 것은 아님을 이해하여야 할 것이다. 첨부된 특허청구범위의 범위는 모든 변경과, 등가 구조 및 기능을 포함하도록 가능 넓은 해석과 일치하여야 한다.

### 부호의 설명

[0084] 81: 화상 해석 보드

82: USB-팩스 보드

93: 키보드

94: 멀티미디어 카드 리더

95: USB 메모리

100: 화상 형성 장치

203: HID 드라이버

204: 매스 스토리지 드라이버

211: 화상 해석 보드 드라이버

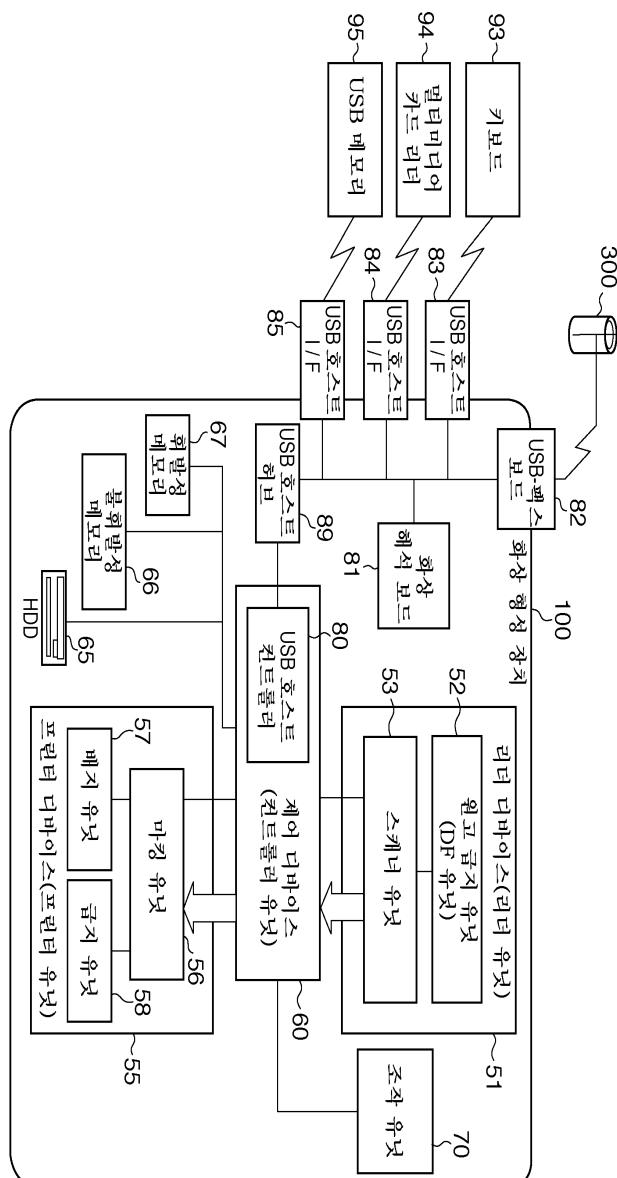
212: 팩스 드라이버

220: 애플리케이션

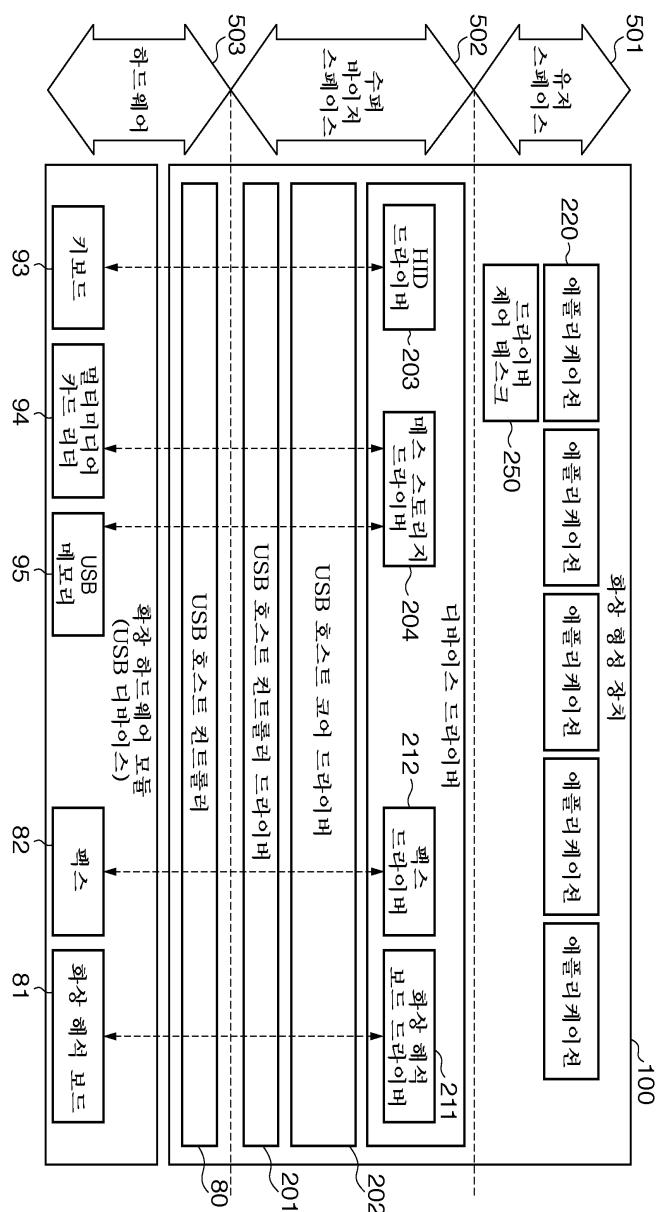
## 250: 드라이버 제어 태스크

도면

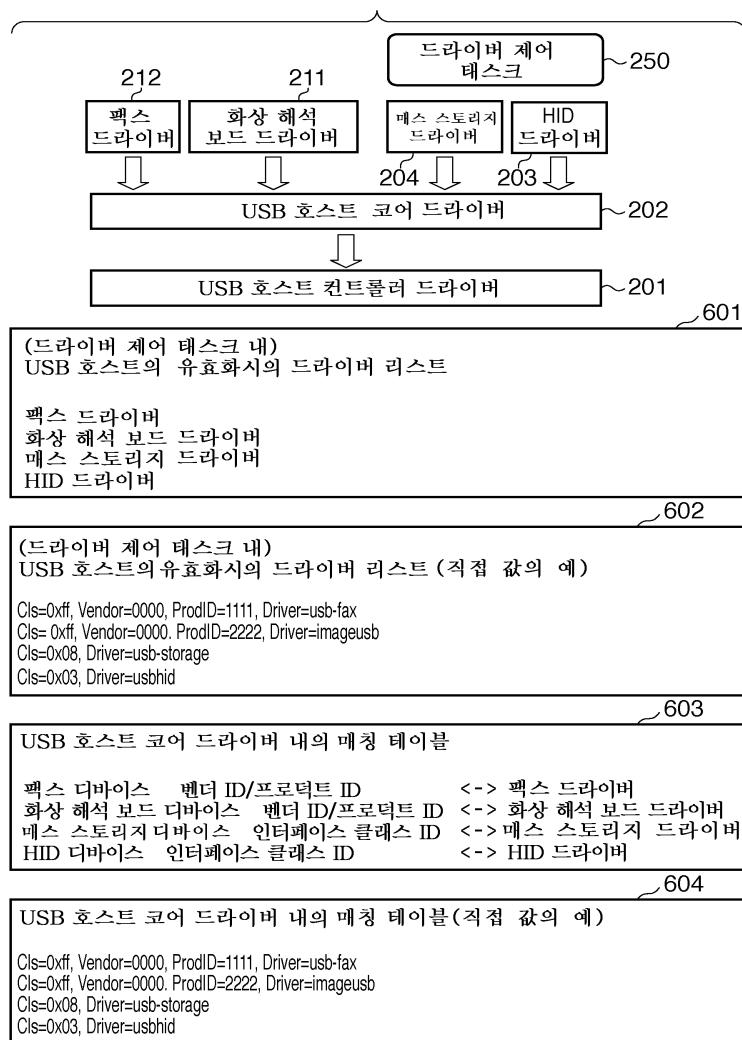
도면1



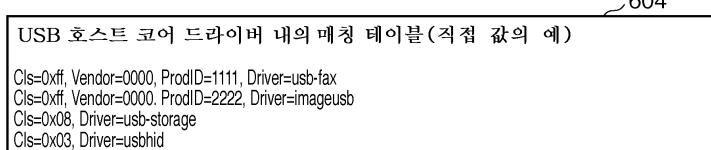
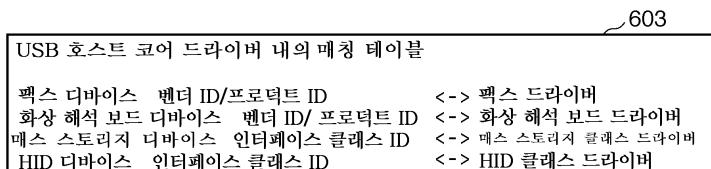
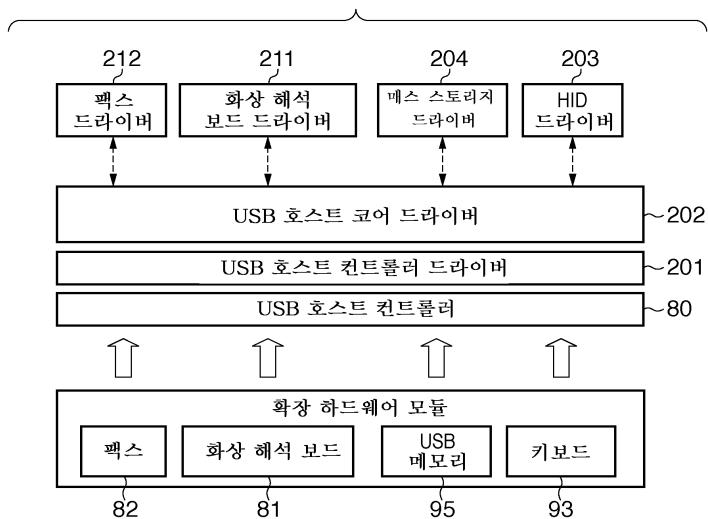
도면2



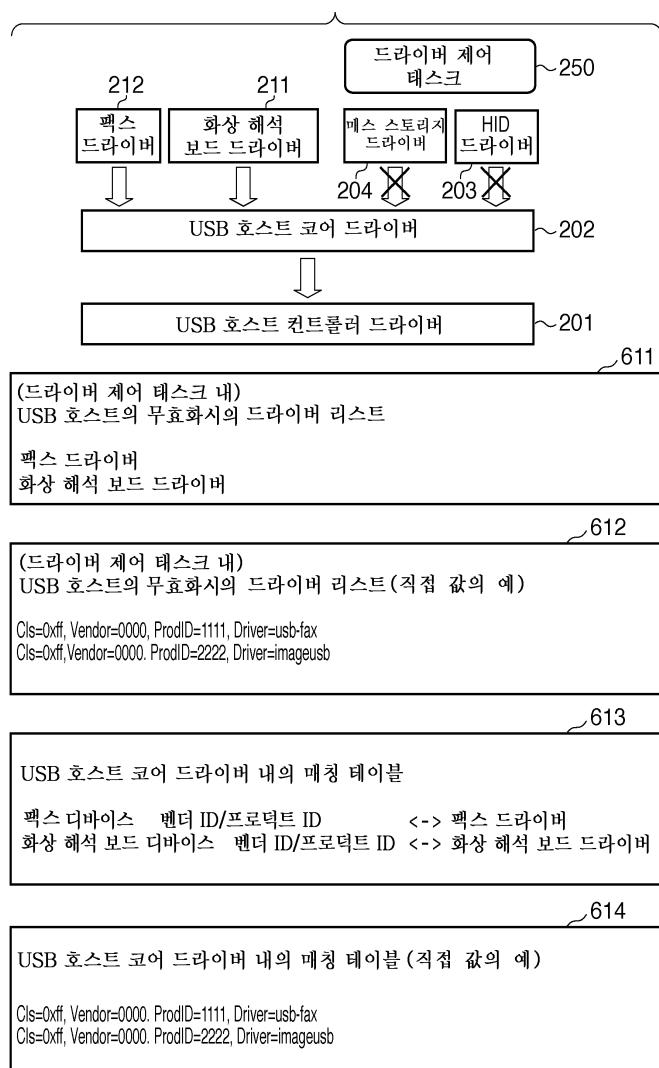
## 도면3



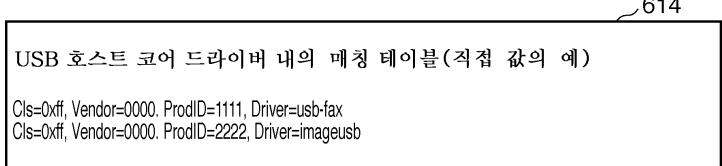
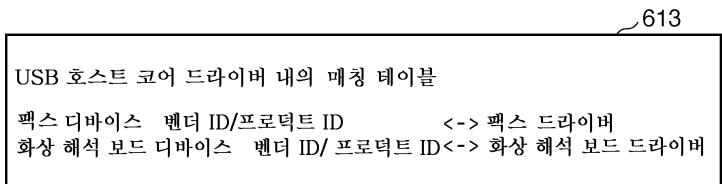
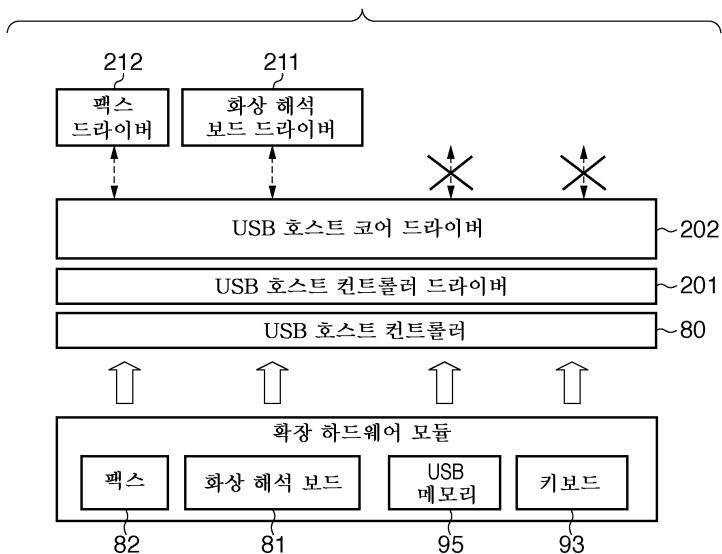
## 도면4



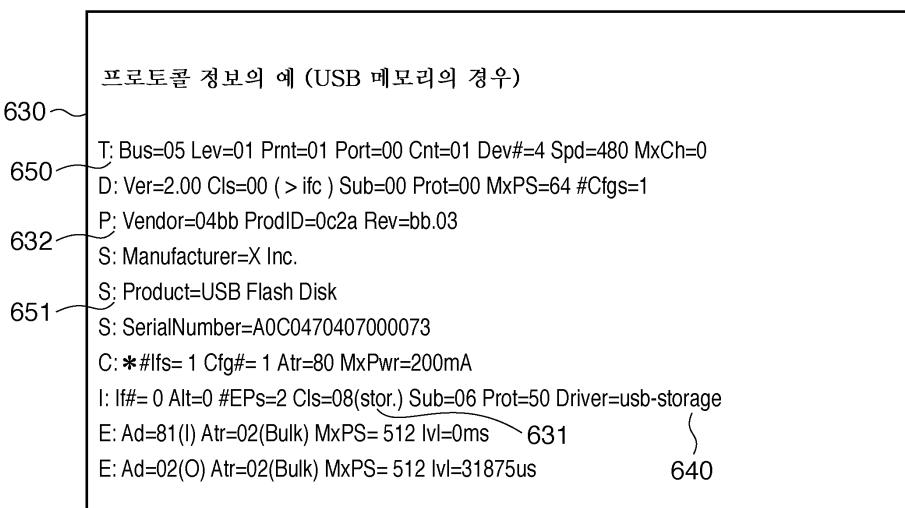
## 도면5



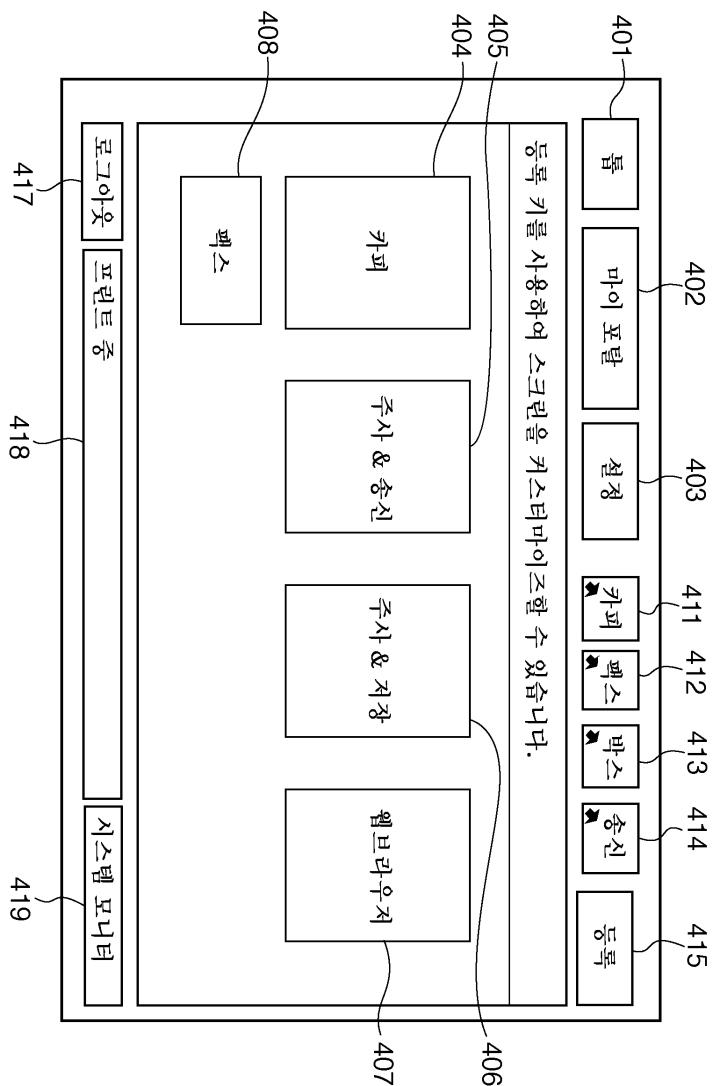
## 도면6



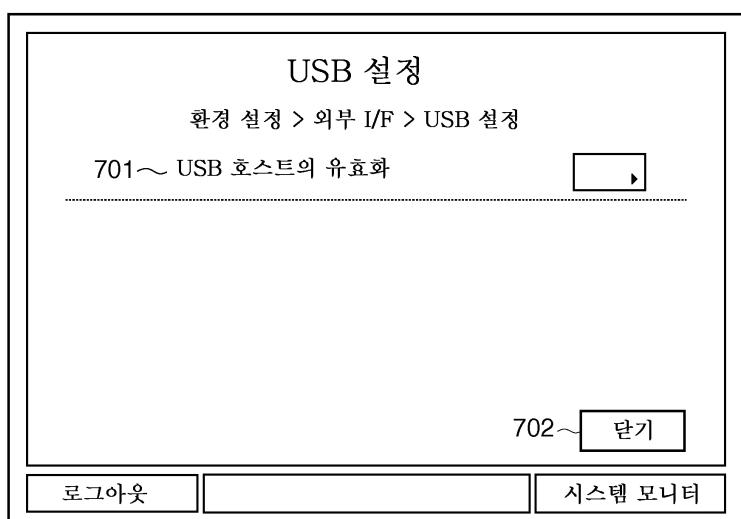
## 도면7



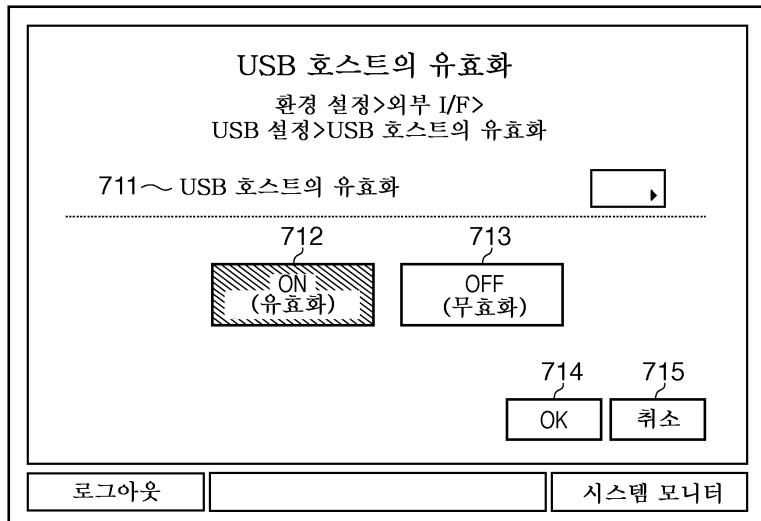
도면8



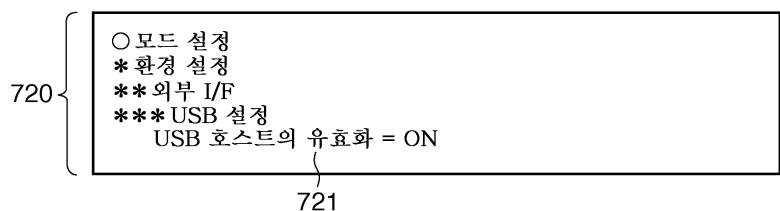
도면9



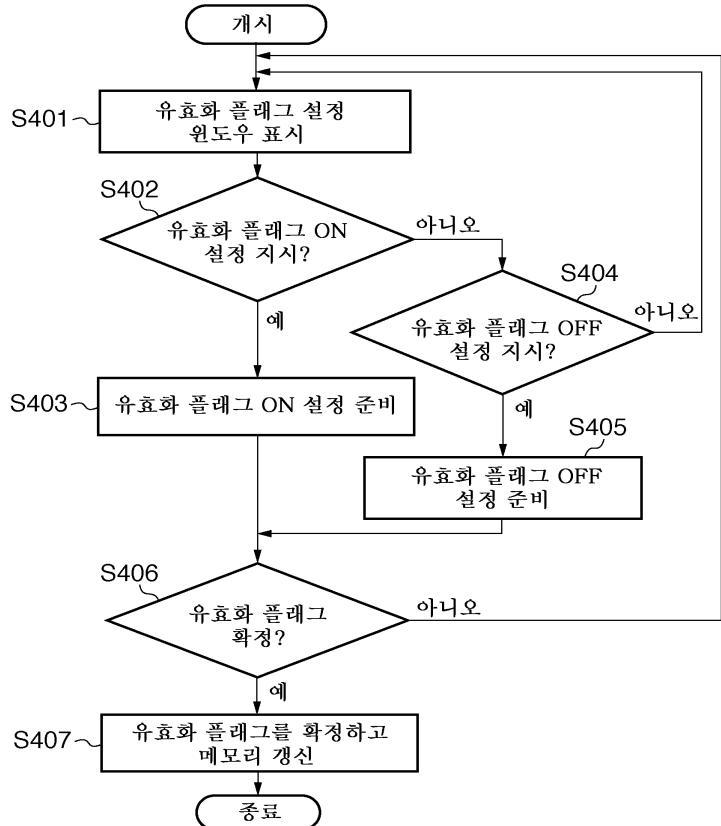
## 도면10



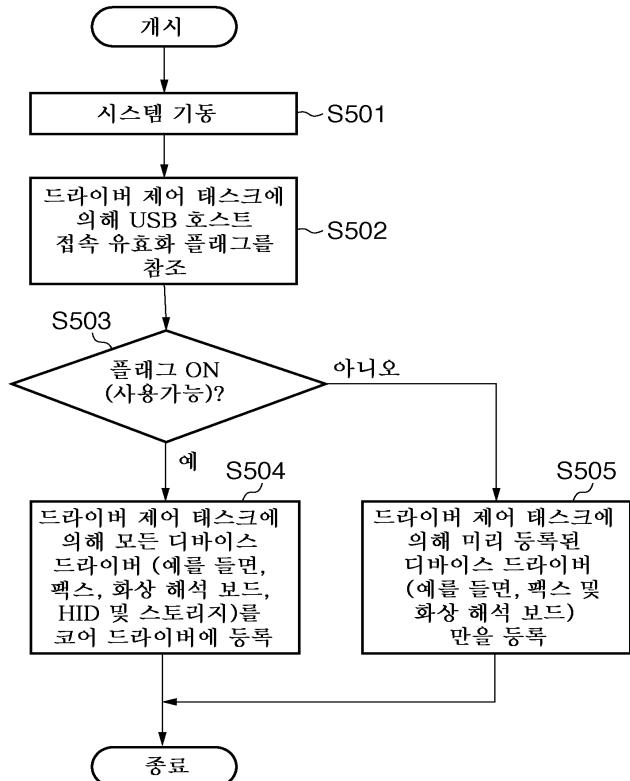
## 도면11



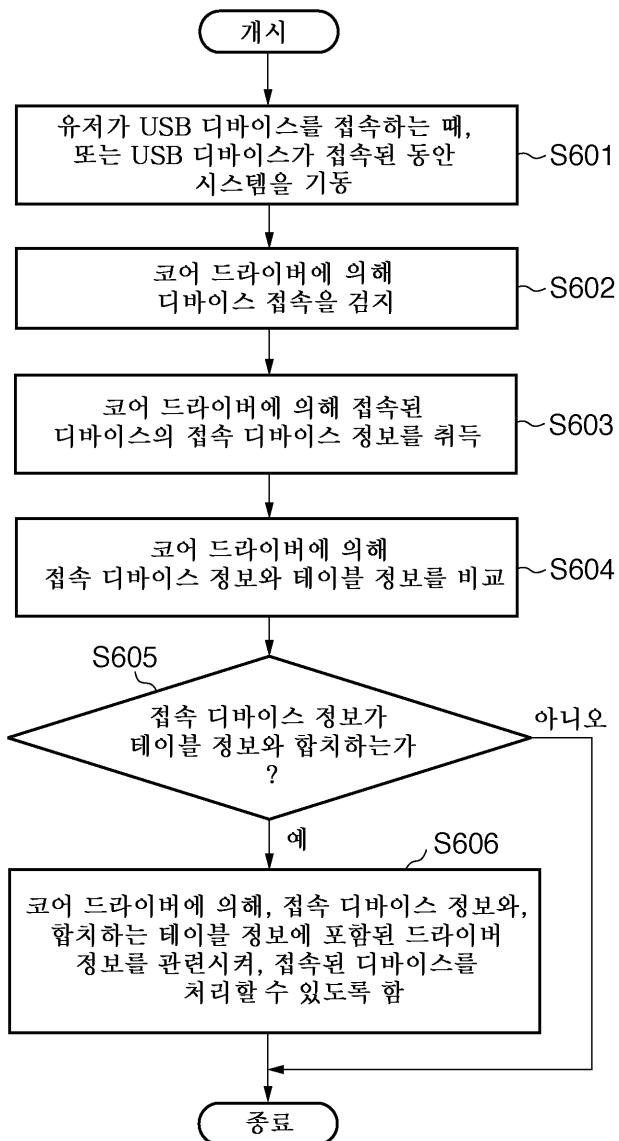
## 도면12



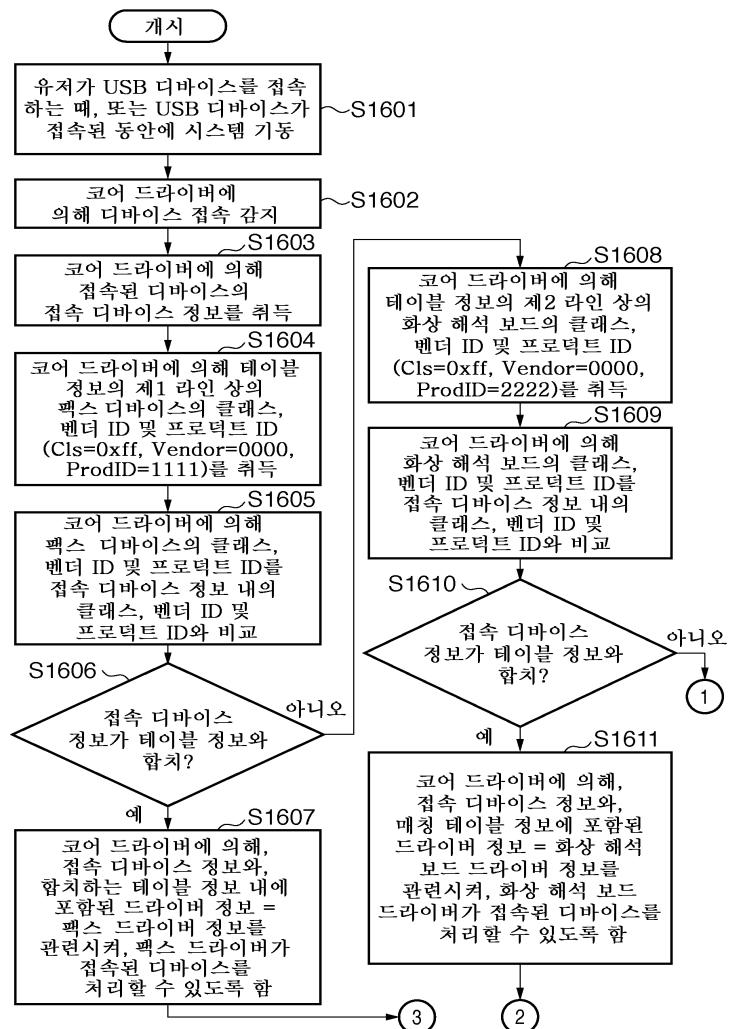
## 도면13



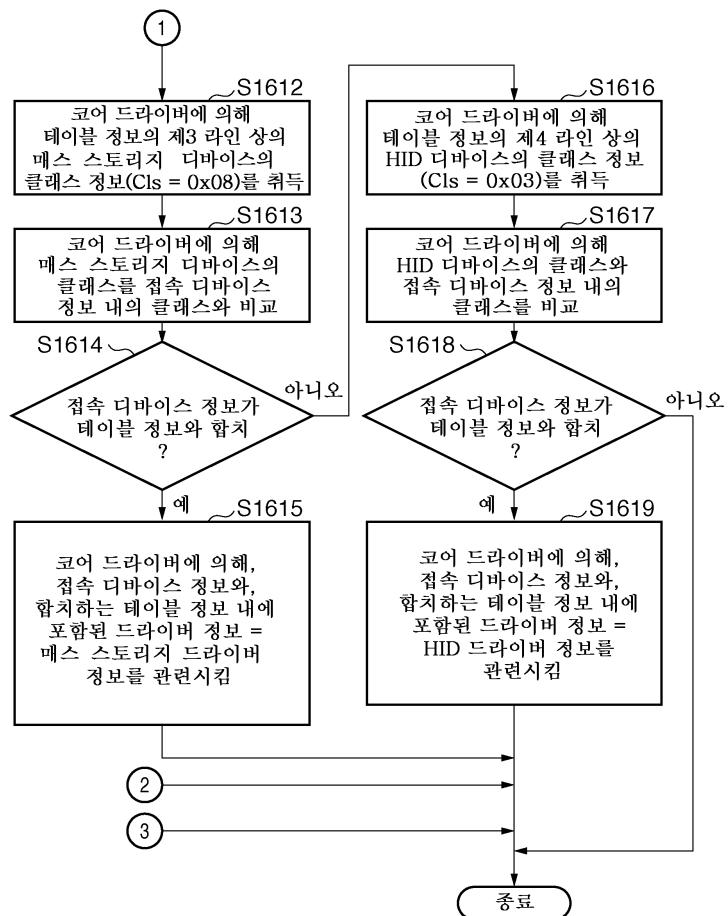
## 도면14



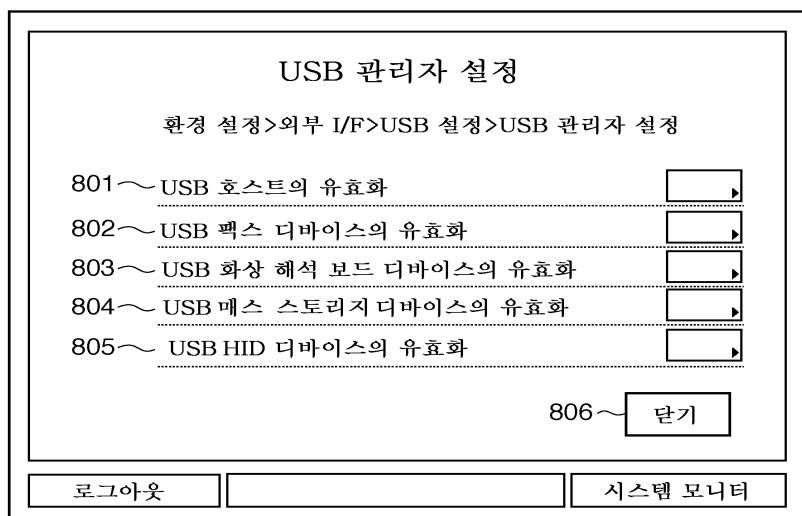
## 도면15a



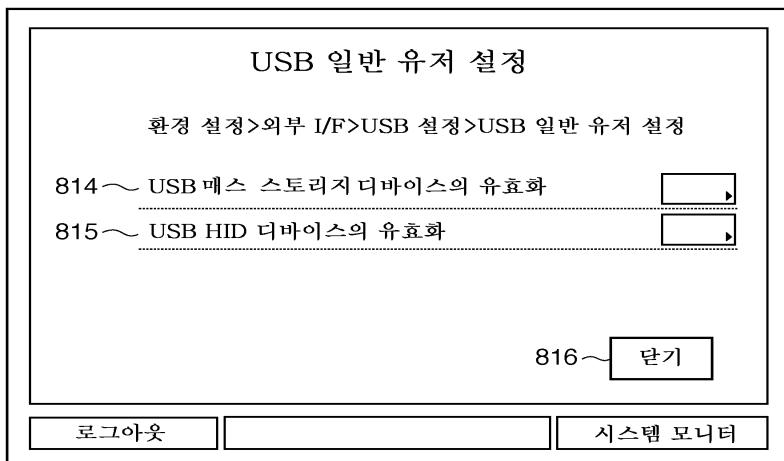
도면15b



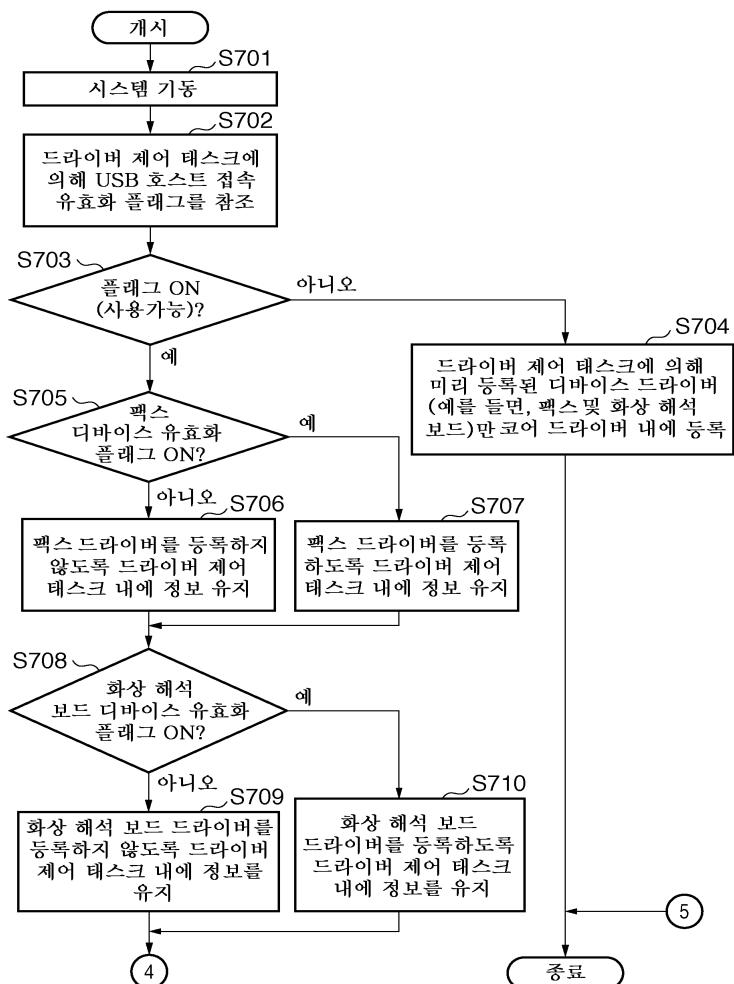
도면16



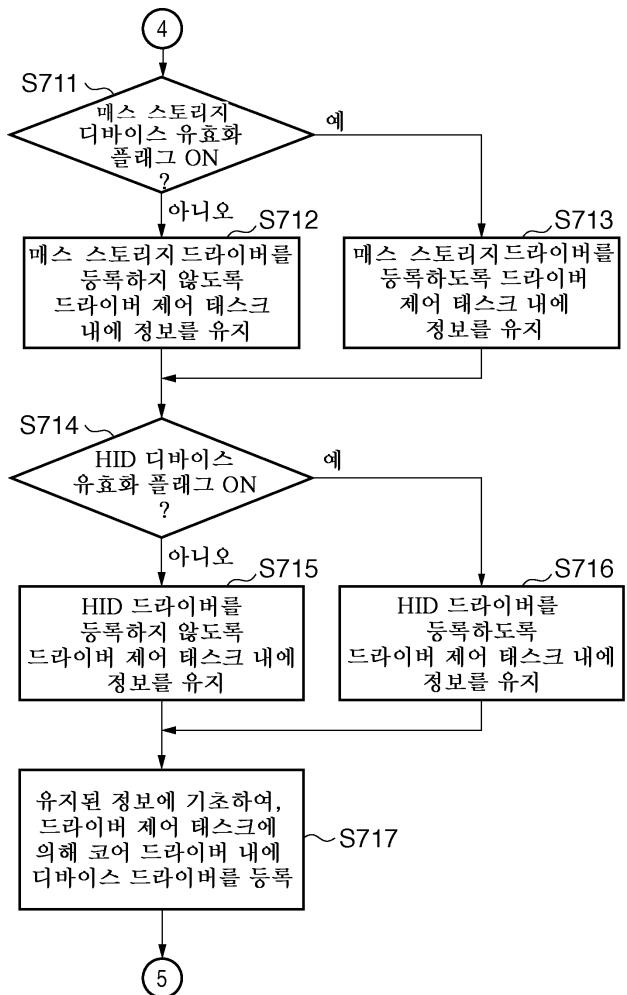
## 도면17



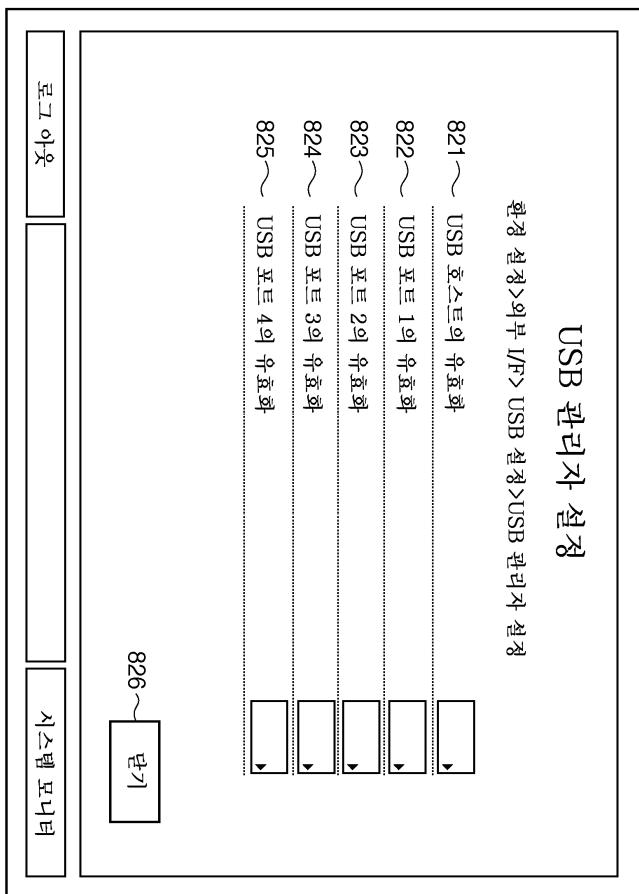
## 도면18a



## 도면18b



도면19



## 도면20

