

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2020-35242
(P2020-35242A)

(43) 公開日 令和2年3月5日(2020.3.5)

(51) Int. Cl.	F 1	テーマコード (参考)
G06F 13/14 (2006.01)	G06F 13/14 320A	5B077
G06F 13/38 (2006.01)	G06F 13/14 330A	
	G06F 13/38 350	

審査請求 未請求 請求項の数 8 O L (全 15 頁)

(21) 出願番号	特願2018-162175 (P2018-162175)	(71) 出願人	000002369 セイコーエプソン株式会社 東京都新宿区新宿四丁目1番6号
(22) 出願日	平成30年8月30日 (2018.8.30)	(74) 代理人	110001623 特許業務法人真菱国際特許事務所
		(72) 発明者	田中 大 長野県諏訪市大和3丁目3番5号 セイコーエプソン株式会社内
		Fターム(参考)	5B077 NN02

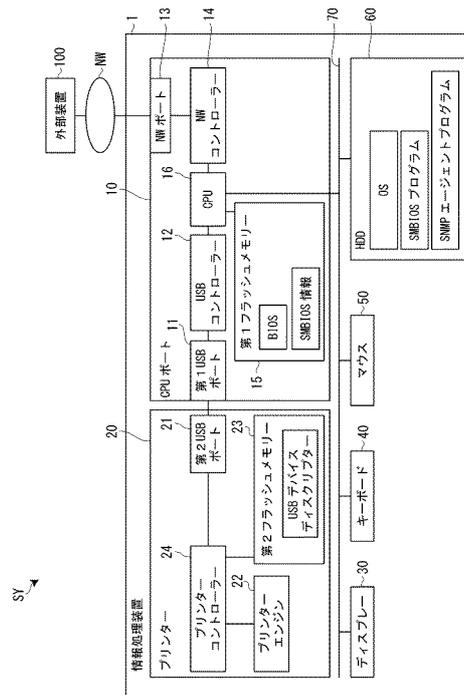
(54) 【発明の名称】 情報処理装置および情報処理装置の制御方法

(57) 【要約】

【課題】 ユーザーの手を煩わせることなく、ネットワークNWに対応していない内部機器に関する識別情報を付与する。

【解決手段】 情報処理装置1は、外部装置100とネットワークNWを介して通信可能であり、CPU16を有するCPUボード10と、CPUボード10とネットワークNWとは異なる通信方式で通信可能であり、USBデバイスディスクリプターを有する内部機器であるプリンター20と、を備え、CPU16は、プリンター20からUSBデバイスディスクリプターを取得し、取得したUSBデバイスディスクリプターに基づいてプリンター20に対応する識別情報を生成し、外部装置100からプリンター20に対する照会要求を受信すると、生成した識別情報を示す第1応答情報を外部装置100に送信する。

【選択図】 図1



【特許請求の範囲】**【請求項 1】**

外部装置とネットワークを介して通信可能であり、制御部を有する制御機器と、前記制御機器と前記ネットワークとは異なる通信方式で通信可能であり、機器情報を有する内部機器と、を備え、

前記制御部は、

前記内部機器から前記機器情報を取得し、取得した前記機器情報に基づいて前記内部機器に対応する識別情報を生成し、

前記ネットワークを介して、前記外部装置から前記内部機器に対する照会要求を受信すると、生成した前記識別情報を示す第 1 応答情報を前記外部装置に送信することを特徴とする情報処理装置。

10

【請求項 2】

前記制御機器は、BIOSの設定値が記憶される第 1 記憶部を有し、

前記制御部は、生成した前記識別情報を前記第 1 記憶部に記憶させることを特徴とする請求項 1 に記載の情報処理装置。

【請求項 3】

前記第 1 記憶部とは異なる第 2 記憶部を備え、

前記制御部は、

所定の通信プロトコルに基づく通信を実現するための通信プログラムを、前記第 2 記憶部から読み出して実行し、

前記ネットワークを介して、前記外部装置から前記内部機器に対する通信要求を受信すると、前記第 1 記憶部に記憶されている前記識別情報を前記所定の通信プロトコルで利用可能な形式の第 2 応答情報に変換し、前記第 2 応答情報を前記外部装置に送信することを特徴とする請求項 2 に記載の情報処理装置。

20

【請求項 4】

前記制御機器と前記内部機器の間の通信方式はUSB通信規格に対応したものであり、

前記内部機器は、USB機器であり、

前記機器情報は、USBデバイスディスクリプターであり、

前記通信プログラムは、SMBIOSプログラムおよびSNMPエージェントプログラムの少なくとも一方を含むことを特徴とする請求項 3 に記載の情報処理装置。

30

【請求項 5】

前記制御部は、

少なくとも、前記制御機器の初回起動時に、前記識別情報を生成することを特徴とする請求項 1 ないし 4 のいずれか 1 項に記載の情報処理装置。

【請求項 6】

前記制御部は、

前記制御機器の起動時に、前記第 1 記憶部に前記識別情報が記憶されているか否かを判別し、前記第 1 記憶部に前記識別情報が記憶されていないと判定した場合、前記識別情報を生成し、前記第 1 記憶部に前記識別情報が記憶されていると判定した場合、前記機器情報を生成しないことを特徴とする請求項 2 ないし 4 のいずれか 1 項に記載の情報処理装置。

40

【請求項 7】

前記内部機器は、印刷機構を有することを特徴とする請求項 1 ないし 6 のいずれか 1 項に記載の情報処理装置。

【請求項 8】

外部装置とネットワークを介して通信可能であり、制御部を有する制御機器と、

前記制御機器と前記ネットワークとは異なる通信方式で通信可能であり、機器情報を有する内部機器と、を備えた情報処理装置の制御方法であって、

前記内部機器から前記機器情報を取得し、取得した前記機器情報に基づいて前記内部機器に対応する識別情報を生成し、

50

前記ネットワークを介して、前記外部装置から前記内部機器に対する照会要求を受信すると、生成した前記識別情報を示す第1応答情報を前記外部装置に送信することを特徴とする情報処理装置の制御方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、外部装置とネットワークを介して接続可能な情報処理装置および情報処理装置の制御方法に関する。

【背景技術】

【0002】

特許文献1には、複数台のUSB機器と接続されると共に、ネットワークを介してPCと接続されたデバイスサーバーであって、USB機器がシリアル番号を備えていない場合、デバイスサーバーのデバイスサーバー情報と、デバイスサーバーとUSB機器を接続するインターフェース情報と、USB機器の機器情報と、に基づいて、USB機器の識別情報を生成する技術が開示されている。

【0003】

一方、ネットワーク通信に対応していないUSB機器等の内部機器を、外部装置からネットワークを介して制御する方法として、CPUボード等の制御機器を、外部装置と内部機器との間に介在させる方法が知られている。

【先行技術文献】

【特許文献】

【0004】

【特許文献1】特開2013-109466号公報

【発明の概要】

【発明が解決しようとする課題】

【0005】

ネットワーク通信に対応していない内部機器を、外部装置からネットワークを介して制御する場合、内部機器に対応する識別情報を制御機器に付与する必要がある。特許文献1の技術では、内部機器に識別情報を付与することはできるが、制御機器に、内部機器に対応する識別情報を付与することはできない。もちろん、ユーザーが手入力により、内部機器に対応する識別情報を制御機器に付与することも考えられるが、手間がかかる。

【課題を解決するための手段】

【0006】

本発明の情報処理装置は、外部装置とネットワークを介して通信可能であり、制御部を有する制御機器と、制御機器とネットワークとは異なる通信方式で通信可能であり、機器情報を有する内部機器と、を備え、制御部は、内部機器から機器情報を取得し、取得した機器情報に基づいて内部機器に対応する識別情報生成し、ネットワークを介して、外部装置から内部機器に対する照会要求を受信すると、生成した識別情報を示す第1応答情報を外部装置に送信する。

【0007】

本発明の情報処理装置の制御方法は、外部装置とネットワークを介して通信可能であり、制御部を有する制御機器と、制御機器とネットワークとは異なる通信方式で通信可能であり、機器情報を有する内部機器と、を備えた情報処理装置の制御方法であって、内部機器から機器情報を取得し、取得した機器情報に基づいて内部機器に対応する識別情報を生成し、ネットワークを介して、外部装置から内部機器に対する照会要求を受信すると、生成した識別情報を示す第1応答情報を外部装置に送信する。

【図面の簡単な説明】

【0008】

【図1】ネットワークシステムのブロック図である。

【図2】BIOSメニューの一例を示す図である。

10

20

30

40

50

【図3】BIOS処理を示すフローチャートである。

【図4】図3に続くフローチャートである。

【図5】Auto Fill処理を示すフローチャートである。

【図6】図5に続くフローチャートである。

【図7】第1応答処理を示すフローチャートである。

【図8】第2応答処理を示すフローチャートである。

【発明を実施するための形態】

【0009】

以下、一実施形態に係る情報処理装置および情報処理装置の制御方法について、添付図面を参照して説明する。図1は、ネットワークシステムSYのブロック図である。ネットワークシステムSYは、情報処理装置1と、ネットワークNWを介して情報処理装置1と接続された外部装置100と、を備える。

10

【0010】

情報処理装置1は、特に用途を限定しないが、例えば、会計処理を行うPOS(Point of Sale System)端末として機能する。また、外部装置100は、例えば、POS端末を管理するPOSサーバーとして機能する。この場合、外部装置100は、情報処理装置1から取得した会計情報に基づいて、レシート印刷データやクーポン印刷データを生成し、生成したこれらの印刷データを情報処理装置1に送信するなどの処理を行う。

【0011】

ネットワークNWは、例えばインターネット網である。また、外部装置100は、例えばWebサーバー装置、PC(Personal Computer)またはタブレット端末である。

20

【0012】

情報処理装置1は、CPU(Central Processing Unit)ボード10と、プリンター20と、ディスプレイ30と、キーボード40と、マウス50と、HDD(Hard Disk Drive)60と、を備える。CPUボード10は、「制御機器」の一例である。また、プリンター20は、「内部機器」および「USB機器」の一例である。また、HDD60は、「第2記憶部」の一例である。なお、情報処理装置1のプリンター20以外の構成、すなわち、CPUボード10、ディスプレイ30、キーボード40、マウス50およびHDD60は、例えばPCである。

【0013】

プリンター20は、例えば、レシートやクーポンを印刷するレシートプリンターである。また、プリンター20は、ネットワーク通信に対応していないことを前提とする。本実施形態のネットワークシステムSYは、このプリンター20を外部装置100から制御するため、プリンター20と外部装置100との間にCPUボード10を介在させた構成となっている。

30

【0014】

CPUボード10は、第1USBポート11と、USBコントローラー12と、ネットワークポート13と、ネットワークコントローラー14と、第1フラッシュメモリー15と、CPU16(プロセッサ)と、を備える。第1フラッシュメモリー15は、「第1記憶部」の一例である。また、CPU16は、「制御部」の一例である。なお、図1において、ネットワークポート13は、「NWポート」と表記し、ネットワークコントローラー14は、「NWコントローラー」と表記する。

40

【0015】

第1USBポート11は、CPUボード10がプリンター20とUSB通信を行うためのポートである。第1USBポート11は、プリンター20の第2USBポート21と接続される。つまり、CPUボード10とプリンター20の間の通信方式はUSB通信規格に対応したものである。USBコントローラー12は、CPU16の命令に基づき、USB通信を制御する。

【0016】

ネットワークポート13は、CPU16が外部装置100とネットワーク通信を行うためのポートである。ネットワークポート13は、外部装置100の不図示のネットワークポ

50

ートと接続される。なお、CPUボード10と外部装置100は、SNMP(Simple Network Management Protocol)に基づいて通信が行われる。SNMPは、「所定の通信プロトコル」の一例である。ネットワークコントローラ14は、CPU16の命令に基づき、ネットワーク通信を制御する。ネットワークコントローラ14は、例えばネットワークカードとしてCPUボード10に実装される。

【0017】

第1フラッシュメモリー15は、BIOS(Basic Input / Output System)およびSMBIOS情報を記憶する。BIOSは、CPUボード10の起動時に実行されるプログラムである。CPU16は、第1フラッシュメモリー15からBIOSを読み出して実行し、CPUボード10に接続された各種ハードウェアの初期化や、HDD60内のOS(Operating System)の読み出しおよび起動、などを行う。

10

【0018】

SMBIOS情報は、BIOS情報、システム情報およびUUID(Universally Unique Identifier)など、SMBIOSで定義される各種情報を含む。例えば、BIOS情報は、BIOSの設定値を含む。また、システム情報は、プリンター20に対応する識別情報を含む。この識別情報は、プリンター20から取得したUSBデバイスディスクリプターを用いて生成される。詳細については、後述する。

【0019】

CPU16は、バス70を介して、ディスプレイ30、キーボード40、マウス50およびHDD60と接続され、これらのハードウェアを制御する。また、CPU16は、これらのハードウェアと協働して、BIOSの起動により実行されるBIOS処理や、プリンター20および外部装置100との通信処理など、各種処理を制御する。

20

【0020】

プリンター20は、第2USBポート21と、プリンターエンジン22と、第2フラッシュメモリー23と、CPU(プロセッサ)を有するプリンターコントローラ24と、を備える。第2USBポート21は、プリンター20がCPUボード10とUSB通信を行うためのポートである。プリンターエンジン22は、印刷ヘッド、用紙搬送機構、用紙切断機構などの印刷機構を含み、プリンターコントローラ24の制御下で、レシート用紙などの印刷媒体に印刷を行う。

【0021】

第2フラッシュメモリー23は、プリンターエンジン22を制御するファームウェアとUSBデバイスディスクリプターを記憶する。USBデバイスディスクリプターは、「機器情報」の一例である。USBデバイスディスクリプターは、「iManufacturer」、「iProduct」、「iSerial Number」などを含む。

30

【0022】

プリンターコントローラ24のCPUは、第2フラッシュメモリー23からファームウェアを読み出して実行し、CPUボード10からの印刷命令にしたがい、プリンターエンジン22に印刷を実行させる。また、プリンターコントローラ24は、CPUボード10からの情報要求にしたがい、第2フラッシュメモリー23に記憶されているUSBデバイスディスクリプターを読み出して、CPUボード10に送信する。

40

【0023】

ディスプレイ30は、BIOSメニューM(図2参照)など、各種情報を表示する。キーボード40およびマウス50は、BIOSメニューMの操作など、ユーザーによる各種操作に用いられる。

【0024】

HDD60は、OS、SMBIOSプログラムおよびSNMPエージェントプログラムを記憶する。SMBIOSプログラムおよびSNMPエージェントプログラムは、「所定の通信プロトコルに基づく通信を実現するための通信プログラム」の一例である。

【0025】

OSは、CPU16が実行する基本ソフトウェアである。SMBIOSプログラムは、CPU16がSM

50

BIOS情報の読み取りや書き込みを行うために用いられる。また、SNMPエージェントプログラムは、CPU 1 6 がSNMPに基づくネットワーク通信を行うために用いられる。なお、特に図示しないが、外部装置 1 0 0 には、SNMPエージェントプログラムに対応するSNMPマネージャプログラムが記憶される。外部装置 1 0 0 は、このSNMPマネージャプログラムによりSNMPマネージャとして機能し、SNMPエージェントとなる情報処理装置 1 を管理する。

【 0 0 2 6 】

上記の構成により、CPUボード 1 0 のCPU 1 6 は、BIOS処理の一部として実行されるAuto fill処理（図 5 および図 6 参照）において、プリンター 2 0 に対し、USBデバイスディスクリプターを要求する。CPU 1 6 は、プリンター 2 0 から返信されたUSBデバイスディスクリプターを、第 1 フラッシュメモリー 1 5 の所定の記憶領域に記憶する。CPU 1 6 は、SMBIOSプログラムを用いて、第 1 フラッシュメモリー 1 5 に記憶されたUSBデバイスディスクリプターから、プリンター 2 0 に対応する識別情報を生成し、SMBIOS情報の一部として記憶する。

10

【 0 0 2 7 】

また、CPU 1 6 は、外部装置 1 0 0 からプリンター 2 0 に対する照会要求を受信すると、SMBIOS情報の一部として記憶されている識別情報を示す第 1 応答情報を外部装置 1 0 0 に送信する（図 7 参照）。この識別情報は、外部装置 1 0 0 が、CPUボード 1 0 を介してプリンター 2 0 を制御するために必要な情報である。また、CPU 1 6 は、外部装置 1 0 0 からプリンター 2 0 に対する通信要求を受信すると、SMBIOS情報の一部として記憶されている識別情報を示す第 2 応答情報を外部装置 1 0 0 に送信する（図 8 参照）。

20

【 0 0 2 8 】

また、CPU 1 6 は、外部装置 1 0 0 からXML (Extensible Markup Language) 形式の印刷データを受信した場合、XML形式の印刷データを解析して、プリンター 2 0 のファームウェアで実行可能な印刷命令である文字コードや画像データ、コマンドに変換する。CPU 1 6 は、変換後の印刷命令を、プリンター 2 0 に送信する。

【 0 0 2 9 】

図 2 は、BIOSメニュー M の一例を示す図である。BIOSメニュー M は、ユーザーの操作にしたがって、ディスプレイ 3 0 に表示される。BIOSメニュー M は、SMBIOSメニュー M 1 と、SMBIOS Auto fillメニュー M 2 と、を含む。

30

【 0 0 3 0 】

SMBIOSメニュー M 1 は、Product Nameの情報 8 1 と、Serial Numberの情報 8 2 と、UIDの情報 8 3 と、を表示する。ここで、Product Nameの情報 8 1 と、Serial Numberの情報 8 2 が、プリンター 2 0 に対応する識別情報に相当する。また、UIDの情報 8 3 は、ネットワークコントローラー 1 4 のMAC (Media Access Control) アドレスを用いて生成される情報である。

【 0 0 3 1 】

SMBIOS Auto fillメニュー M 2 は、Auto fill nowボタン 8 4 と、Auto fill on bootの情報 8 5 と、Target USB Portの情報 8 6 と、Product suffixの情報 8 7 と、S/N conversionの情報 8 8 と、を表示する。

40

【 0 0 3 2 】

Auto fill nowボタン 8 4 は、ユーザーが手動でAuto fill処理を実行させる場合に選択されるボタンである。Auto fill nowボタン 8 4 が選択されると、SMBIOS情報が更新され、情報処理装置 1 が再起動される。

【 0 0 3 3 】

Auto fill on bootの情報 8 5 としては、"Enabled"と"Disabled"のいずれかが選択される。"Enabled"が選択された場合、所定の条件を満たすとAuto fill処理を実行する。また、"Disabled"が選択された場合、所定の条件を満たす場合でもAuto fill処理を実行しない。所定の条件とは、後述する図 3 の S 0 2 で N o となり、S 0 3 で Y e s となる条件である。

50

【 0 0 3 4 】

Target USB Portの情報 8 6 は、第 1 USBポート 1 1 のポート名である。Product suffixの情報 8 7 は、識別情報の一部となるProduct Nameの情報 8 1 を生成する際の接尾辞である。例えば、CPU 1 6 は、プリンター 2 0 から取得したUSBデバイスディスクリプターの「iProduct」が「TM-T1234」である場合であって、Product suffixの情報 8 7 が「-DT2」の場合、「iProduct」の後にProduct suffixの情報 8 7 を付加した「TM-T1234-DT2」を、Product Nameの情報 8 1 として生成する。

【 0 0 3 5 】

S/N conversionの情報 8 8 としては、「Enabled」と「Disabled」のいずれかが選択される。「Enabled」が選択された場合、プリンター 2 0 から取得したUSBデバイスディスクリプターの「iSerial Number」に対しシリアル番号変換処理を行って、識別情報の一部となるSerial Numberの情報 8 2 を生成する。図 2 の例では、シリアル番号変換処理が行われた場合を示している。シリアル番号変換処理は、所定の変換規則にしたがって、USBデバイスディスクリプターの「iSerial Number」をSerial Numberの情報 8 2 に変換する。また、S/N conversionの情報 8 8 として「Disabled」が選択された場合、USBデバイスディスクリプターの「iSerial Number」に対しシリアル番号変換処理を行うことなく、「iSerial Number」をそのままSerial Numberの情報 8 2 とする。

10

【 0 0 3 6 】

図 3 および図 4 は、BIOS処理を示すフローチャートである。BIOS処理は、第 1 フラッシュメモリー 1 5 に記憶されているBIOSに基づいて、CPU 1 6 により実行される処理である。CPU 1 6 は、BIOSが起動されると、Auto fill on bootの情報 8 5 が「Enabled」であるかを判別する（S 0 1）。CPU 1 6 は、Auto fill on bootの情報 8 5 が「Enabled」ではないと判定した場合（S 0 1 : N o）、S 0 6 に進む。

20

【 0 0 3 7 】

CPU 1 6 は、Auto fill on bootの情報 8 5 が「Enabled」と判定した場合（S 0 1 : Y e s ）、 「Auto fill loop guard flag」が設定されているかを判別する（S 0 2）。この「Auto fill loop guard flag」は、Auto fill処理を実行しても、Product Nameの情報 8 1 と、Serial Numberの情報 8 2 が空文字列（空白）の場合、すなわち、これらの情報が入力されていない状態の場合に無限ループとなることを防ぐためのフラグである。

30

【 0 0 3 8 】

CPU 1 6 は、「Auto fill loop guard flag」が設定されていると判定した場合（S 0 2 : Y e s ）、S 0 6 に進み、「Auto fill loop guard flag」が設定されていないと判定した場合（S 0 2 : N o ）、Product Nameの情報 8 1 とSerial Numberの情報 8 2 がいずれも空文字列であるかを判別する（S 0 3）。CPU 1 6 は、Product Nameの情報 8 1 とSerial Numberの情報 8 2 がいずれも空文字列ではないと判定した場合（S 0 3 : N o ）、すなわちProduct Nameの情報 8 1 とSerial Numberの情報 8 2 の少なくとも一方が入力された状態であると判定した場合は、S 0 6 に進む。

【 0 0 3 9 】

CPU 1 6 は、Product Nameの情報 8 1 とSerial Numberの情報 8 2 がいずれも空文字列であると判定した場合（S 0 3 : Y e s ）、Auto fill処理（図 5 および図 6 参照）を実行する（S 0 4）。その後、CPU 1 6 は、「Auto fill loop guard flag」を設定し（S 0 5）、S 0 1 に戻る。

40

【 0 0 4 0 】

図 4 に示すように、CPU 1 6 は、ユーザーがBIOSメニュー M（図 2 参照）を表示するための操作を行ったか否かを判別する（S 0 6）。CPU 1 6 は、ユーザーがBIOSメニュー Mを表示するための操作を行っていないと判定した場合（S 0 6 : N o ）、BIOS処理を終了する。また、CPU 1 6 は、ユーザーがBIOSメニュー Mを表示するための操作を行ったと判定した場合（S 0 6 : Y e s ）、BIOSメニュー Mを表示する（S 0 7）。

【 0 0 4 1 】

50

CPU 1 6 は、BIOSメニュー M において、ユーザーが Auto fill now ボタン 8 4 を選択したか否かを判別する (S 0 8)。CPU 1 6 は、ユーザーが Auto fill now ボタン 8 4 を選択したと判定した場合 (S 0 8 : Y e s)、Auto fill 処理を実行し (S 0 9)、S 0 1 に戻る。また、CPU 1 6 は、ユーザーが Auto fill now ボタン 8 4 を選択していないと判定した場合 (S 0 8 : N o)、ユーザーが SMBIOSメニュー M 1 を手動で編集したか否かを判別する (S 1 0)。CPU 1 6 は、ユーザーが SMBIOSメニュー M 1 を手動で編集したと判定した場合 (S 1 0 : Y e s)、SMBIOS 情報を更新して情報処理装置 1 を再起動し (S 1 1)、S 1 2 に進む。

【 0 0 4 2 】

CPU 1 6 は、ユーザーが BIOSメニュー M を終了するための操作を行ったか否かを判別し (S 1 2)、終了するための操作を行っていないと判定した場合は (S 1 2 : N o)、S 0 8 に戻る。また、CPU 1 6 は、ユーザーが BIOSメニュー M を終了するための操作を行ったと判定した場合は (S 1 2 : Y e s)、BIOSメニュー M を終了し (S 1 3)、BIOS 処理を終了する。なお、CPU 1 6 は、BIOS 処理の終了後、OS を起動する。

10

【 0 0 4 3 】

図 5 および図 6 は、Auto fill 処理を示すフローチャートである。Auto fill 処理は、図 3 の S 0 4 および図 4 の S 0 9 に相当するサブフローチャートである。CPU 1 6 は、Auto fill 処理を開始すると、Target USB Port の値を取得し (S 2 1)、プリンター 2 0 が Target USB Port、すなわち第 1 USBポート 1 1 に接続されているか否かを判別する (S 2 2)。

20

【 0 0 4 4 】

CPU 1 6 は、プリンター 2 0 が Target USB Port に接続されていないと判定した場合 (S 2 2 : N o)、Product Name の情報 8 1 と Serial Number の情報 8 2 を空文字列にして (S 2 3)、S 3 4 に進む。また、CPU 1 6 は、プリンター 2 0 が Target USB Port に接続されていると判定した場合 (S 2 2 : Y e s)、USB デバイスディスクリプターの「iProduct」を取得する (S 2 4)。

【 0 0 4 5 】

CPU 1 6 は、USB デバイスディスクリプターの「iProduct」を取得できたか否かを判別し (S 2 5)、取得できたと判定した場合 (S 2 5 : Y e s)、「iProduct」の後に Product suffix の情報 8 7 を付加し、Product Name の情報 8 1 を設定する (S 2 6)。ここで、「Product Name の情報 8 1 を設定する」とは、SMBIOS のデータ構造に Product Name の情報 8 1 を入力し、SNBIOS 情報として記憶させることを意味する。また、CPU 1 6 は、USB デバイスディスクリプターの「iProduct」を取得できなかったと判定した場合 (S 2 5 : N o)、Product suffix の情報 8 7 を空文字列にする (S 2 7)。なお、USB デバイスディスクリプターの「iProduct」を取得できない場合としては、プリンター 2 0 が故障している場合や、プリンター 2 0 に「iProduct」が記憶されていない場合、などが考えられる。

30

【 0 0 4 6 】

図 6 に示すように、CPU 1 6 は、S 2 6 または S 2 7 の後、USB デバイスディスクリプターの「iSerial Number」を取得する (S 2 8)。CPU 1 6 は、USB デバイスディスクリプターの「iSerial Number」を取得できたか否かを判別し (S 2 9)、取得できなかったと判定した場合 (S 2 9 : N o)、Serial Number の情報 8 2 を空文字列にし (S 3 0)、S 3 4 に進む。USB デバイスディスクリプターの「iSerial Number」を取得できない場合としては、プリンター 2 0 が故障している場合や、プリンター 2 0 に「iSerial Number」が記憶されていない場合、などが考えられる。また、CPU 1 6 は、USB デバイスディスクリプターの「iSerial Number」を取得できたと判定した場合 (S 2 9 : Y e s)、S/N conversion の情報 8 8 が "Enabled" であるか否かを判別する (S 3 1)。

40

【 0 0 4 7 】

CPU 1 6 は、S/N conversion の情報 8 8 が "Enabled" であると判定した場合 (S 3 1 : Y e s)、プリンター 2 0 から取得した USB デバイスディスクリプターの「iSerial Number」に対しシリアル番号変換処理を行い (S 3 2)、Serial Number の情報 8 2 を設定する

50

(S 3 3)。つまり、「 iSerial Number」に対してシリアル番号変換処理を行った情報を、SNBIOS情報の一部として記憶する。また、CPU 1 6 は、S/N conversionの情報 8 8 が"Enabled"ではないと判定した場合 (S 3 1 : N o)、「 iSerial Number」に対してシリアル番号変換処理を行うことなく、Serial Numberの情報 8 2 を設定する (S 3 3)。つまり、USBデバイスディスクリプターの「 iSerial Number」を、そのままSNBIOS情報の一部として記憶する。その後、CPU 1 6 は、ネットワークコントローラー 1 4 のMACアドレスを用いてUUIDを生成し、UUIDの情報 8 3 を設定して (S 3 4)、Auto fill処理を終了する。

【 0 0 4 8 】

図 7 は、第 1 応答処理を示すフローチャートである。第 1 応答処理は、図 3 および図 4 に示したBIOS処理が実行された後、外部装置 1 0 0 からの照会要求に基づいて実行される処理である。ここで、「照会要求」とは、プリンター 2 0 に対応する識別情報の送信要求を意味する。CPU 1 6 は、ネットワークNWを介して、外部装置 1 0 0 からプリンター 2 0 に対する照会要求を受信すると (S 4 1 : Y e s)、SMBIOSプログラムを用いて、第 1 フラッシュメモリー 1 5 からSMBIOS情報を読み出し、識別情報を取得する (S 4 2)。つまり、CPU 1 6 は、識別情報として、Auto fill処理で設定されたProduct Nameの情報 8 1 およびSerial Numberの情報 8 2 を取得する。

10

【 0 0 4 9 】

また、CPU 1 6 は、SNMPエージェントプログラムを用いて、取得した識別情報をSNMPで利用可能な形式に変換し、変換済み識別情報を生成する (S 4 3)。CPU 1 6 は、変換済み識別情報に、ヘッダーおよびフッターを付加して第 1 応答情報を生成し (S 4 4)、生成した第 1 応答情報を外部装置 1 0 0 に送信する (S 4 5)。外部装置 1 0 0 は、第 1 応答情報を受信し、プリンター 2 0 に対応する識別情報を取得することで、以後、プリンター 2 0 の制御が可能となる。

20

【 0 0 5 0 】

図 8 は、第 2 応答処理を示すフローチャートである。第 2 応答処理は、図 7 に示した第 1 応答処理の後、外部装置 1 0 0 からの通信要求に基づいて実行される処理である。CPU 1 6 は、ネットワークNWを介して、外部装置 1 0 0 からプリンター 2 0 の通信要求を受信すると (S 5 1 : Y e s)、SMBIOSプログラムを用いて、第 1 フラッシュメモリー 1 5 からSMBIOS情報を読み出し、識別情報を取得する (S 5 2)。また、CPU 1 6 は、SNMPエージェントプログラムを用いて、取得した識別情報をSNMPで利用可能な形式に変換し、変換済み識別情報を生成する (S 5 3)。CPU 1 6 は、変換済み識別情報に、ヘッダーおよびフッターを付加して第 2 応答情報を生成し (S 5 4)、生成した第 2 応答情報を外部装置 1 0 0 に送信する (S 5 5)。

30

【 0 0 5 1 】

以上説明したとおり、本実施形態によれば、ネットワーク通信に対応しておらず、情報処理装置 1 の内部に収納され、情報処理装置 1 の一部として一体的に構成されたプリンター 2 0 を、外部装置 1 0 0 からネットワークNWを介して制御する場合において、CPUボード 1 0 が、プリンター 2 0 から取得したUSBデバイスディスクリプターに基づいて、プリンター 2 0 に対応する識別情報を生成するため、ユーザーの手を煩わせることなく、CPUボード 1 0 に識別情報を付与することができる。

40

【 0 0 5 2 】

また、BIOSの設定値が記憶される第 1 フラッシュメモリー 1 5 に識別情報を記憶させるため、情報処理装置 1 のOSが変更された場合でも、識別情報を変更する必要がない。また、HDD 6 0 に識別情報を記憶させると、HDD 6 0 が取り外された場合に識別情報を付与し直す必要があるが、第 1 フラッシュメモリー 1 5 に識別情報を記憶させるため、そのような問題がない。

【 0 0 5 3 】

また、第 1 フラッシュメモリー 1 5 に識別情報を記憶させるため、プリンター 2 0 がCPUボード 1 0 に接続されていない場合でも、外部装置 1 0 0 に応答することができる。また、外部装置 1 0 0 に応答する際は、SMBIOSプログラムを用いてSMBIOS情報を読み出し、

50

SNMPエージェントプログラムを用いてSNMPに基づく通信を行うため、汎用のプログラムを用いて本実施形態を実現することができる。

【 0 0 5 4 】

また、CPUボード10は、第1フラッシュメモリー15に識別情報が記憶されていないと判定した場合（図3のS03でYesと判定した場合）に、Auto Fill処理を実行して識別情報を生成するため、CPUボード10の初回起動時や、ユーザーがBIOSメニューMで意図的に識別情報を削除した場合に、CPUボード10に識別情報を付与することができる。また、第1フラッシュメモリー15に識別情報が記憶されていると判定した場合は、Auto Fill処理を実行しないため、誤って識別情報が変更されることがない。

【 0 0 5 5 】

上記の実施形態によらず、以下の変形例を採用可能である。

[変形例 1]

上記の実施形態では、CPUボード10が生成した識別情報を、SMBIOS情報の一部として第1フラッシュメモリー15に記憶したが、SMBIOS情報以外の情報を識別情報として記憶してもよい。例えば、SNMPのMIB情報を識別情報として記憶してもよい。この場合、MIB情報のsysDescrとして、USBデバイスディスクリプターの「iProduct」に、Product suffixの情報87を付加した情報を記憶し、MIB情報のchassisIdとして、「iSerial Number」を記憶してもよい。

また、識別情報を、例えばHDD60など、第1フラッシュメモリー15以外の記憶装置に記憶してもよい。また、識別情報を記憶する記憶装置は、必ずしも情報処理装置1内に存在しなくてもよく、CPUボード10が通信可能な装置内に存在してもよい。

【 0 0 5 6 】

[変形例 2]

上記の実施形態では、「所定の通信プロトコルに基づく通信を実現するための通信プログラム」として、SMBIOSプログラムおよびSNMPエージェントプログラムを例示したが、必ずしも両プログラムを用いる必要はない。例えば、HDD60に記憶されたOS上で動作する所定のアプリケーションプログラムにより、SMBIOS情報の読み取りおよび書き込みを行ってもよい。また、SNMP以外の通信プロトコルを用いて、CPUボード10と外部装置100が通信を行う構成でもよい。

【 0 0 5 7 】

[変形例 3]

また、上記の実施形態では、CPU16は、SNMPエージェントプログラムに基づいて、SMBIOS情報が記憶されている記憶領域から識別情報を読み出したが、他の方法で識別情報を取得してもよい。例えば、CPU16は、第1フラッシュメモリー15の所定の記憶領域にアクセスして、USBデバイスディスクリプターを取得し、取得したUSBデバイスディスクリプターに基づいて識別情報を生成し、生成した識別情報を取得してもよい。つまり、CPU16は、外部装置100から照会要求や通信要求を受信したとき、一次情報となる所定の記憶領域内のUSBデバイスディスクリプターを用いて識別情報を生成してもよいし、二次情報となるSMBIOS情報を読み出してもよい。

【 0 0 5 8 】

[変形例 4]

上記の実施形態では、USBデバイスディスクリプターの「iProduct」に、Product suffixの情報87を付加したProduct Nameの情報81と、「iSerial Number」に基づくSerial Numberの情報82と、を識別情報としたが、識別情報の生成アルゴリズムは、これに限らない。例えば、「iProduct」に、Product suffixの情報87と、「iSerial Number」と、を付加した情報や「iManufacturer」を、Product Nameの情報81としてもよい。このように、USBデバイスディスクリプターの各情報を利用し、所定の生成アルゴリズムに基づいて、識別情報を生成してもよい。

【 0 0 5 9 】

[変形例 5]

10

20

30

40

50

また、上記の実施形態では、BIOSを用いて識別情報を生成したが、プリンター 20 を制御するデバイスドライバーなどのアプリケーションプログラムを用いて識別情報を生成してもよい。この場合、アプリケーションプログラムは、HDD 60 に記憶される。

【0060】

[変形例 6]

上記の実施形態では、CPUボード 10 とプリンター 20 が、USB通信規格に基づいて通信する場合を例示したが、他の通信規格に基づいて通信する構成でもよい。例えば、CPUボード 10 とプリンター 20 が、RS-232などのシリアル通信規格に基づいて通信する構成でもよい。

【0061】

[変形例 7]

上記の実施形態では、「内部機器」としてプリンター 20 を例示したが、プリンター 20 に限らず、他の機器を用いてもよい。例えば、情報処理装置 1 がPOS端末である場合、スキャナー、表示機器、キーボード、キャッシュドローワーなどを「内部機器」として用いてもよい。また、1台のCPUボード 10 に、複数の機器が接続される構成でもよい。

【0062】

[変形例 8]

上記の実施形態では、「制御機器」としてCPUボード 10 を例示したが、CPUボード 10 に限らず、他の機器を用いてもよい。例えば、各種集積回路やインターフェースボードを「制御機器」として用いてもよい。

【0063】

[変形例 9]

上記の実施形態では、CPU 16 は、第 1 フラッシュメモリ 15 に識別情報が記憶されていないと判定した場合に、Auto Fill処理を実行して識別情報を生成したが、CPUボード 10 の初回起動時以降は、識別情報を書き換え不能としてもよい。

【0064】

[その他の変形例]

上記の各実施形態および各変形例に示した情報処理装置 1 の各処理を実行する方法、情報処理装置 1 の各処理を実行するためのプログラム、またそのプログラムを記録したコンピュータ読み取り可能な記録媒体も、発明の権利範囲に含まれる。また、各実施形態および各変形例を組み合わせた構成としてもよい。その他、ハードウェアとソフトウェアの協業により情報処理装置 1 の各処理を実現するなど、発明の要旨を逸脱しない範囲で、適宜変更が可能である。

【0065】

[付記]

以下、情報処理装置 1 および情報処理装置 1 の制御方法について付記する。

情報処理装置 1 は、外部装置 100 とネットワークNWを介して通信可能であり、制御部 16 を有する制御機器 10 と、制御機器 10 とネットワークNWとは異なる通信方式で通信可能であり、機器情報を有する内部機器 20 と、を備え、制御部 16 は、内部機器 20 から機器情報を取得し、取得した機器情報に基づいて内部機器 20 に対応する識別情報を生成し、ネットワークNWを介して、外部装置 100 から内部機器 20 に対する照会要求を受信すると、生成した識別情報を示す第 1 応答情報を外部装置 100 に送信する。

【0066】

情報処理装置 1 の制御方法は、外部装置 100 とネットワークNWを介して通信可能であり、制御部 16 を有する制御機器 10 と、制御機器 10 とネットワークNWを介さず通信可能であり、機器情報を有する内部機器 20 と、を備えた情報処理装置 1 の制御方法であって、内部機器 20 から機器情報を取得し、取得した機器情報に基づいて内部機器 20 に対応する識別情報を生成し、ネットワークNWを介して、外部装置 100 から内部機器 20 に対する照会要求を受信すると、生成した識別情報を示す第 1 応答情報を外部装置 100 に送信する。

10

20

30

40

50

【 0 0 6 7 】

この構成によれば、ネットワーク通信に対応していない内部機器 2 0 を、外部装置 1 0 0 からネットワーク NW を介して制御する場合において、制御機器 1 0 が、内部機器 2 0 から取得した機器情報に基づいて、内部機器 2 0 に対応する識別情報を生成するため、ユーザーの手を煩わせることなく、制御機器 1 0 に識別情報を付与することができる。

【 0 0 6 8 】

上記の情報処理装置 1 において、制御機器 1 0 は、BIOS の設定値が記憶される第 1 記憶部 1 5 を有し、制御部 1 6 は、生成した識別情報を第 1 記憶部 1 5 に記憶させることが好ましい。

【 0 0 6 9 】

この構成によれば、BIOS の設定値が記憶される第 1 記憶部 1 5 に識別情報を記憶させるため、情報処理装置 1 の OS が変更された場合でも、識別情報を変更する必要がない。また、HDD 6 0 など、取り外し可能な記憶装置に識別情報を記憶させると、記憶装置が取り外された場合に識別情報を付与し直す必要があるが、BIOS の設定値が記憶される第 1 記憶部 1 5 に識別情報を記憶させるため、そのような問題がない。また、制御機器 1 0 内に識別情報を記憶させるため、内部機器 2 0 が接続されていない場合でも、外部装置 1 0 0 に応答することができる。

【 0 0 7 0 】

上記の情報処理装置 1 において、第 1 記憶部 1 5 とは異なる第 2 記憶部 6 0 を備え、制御部 1 6 は、所定の通信プロトコルに基づく通信を実現するための通信プログラムを、第 2 記憶部 6 0 から読み出して実行し、ネットワーク NW を介して、外部装置 1 0 0 から内部機器 2 0 に対する通信要求を受信すると、第 1 記憶部 1 5 に記憶されている識別情報を所定の通信プロトコルで利用可能な形式の第 2 応答情報に変換し、第 2 応答情報を外部装置 1 0 0 に送信することが好ましい。

【 0 0 7 1 】

この構成によれば、BIOS の設定値が記憶される第 1 記憶部 1 5 とは異なる第 2 記憶部 6 0 に記憶されている通信プログラムを用いて、プロトコル変換および外部装置 1 0 0 との通信を行うことができる。これにより、使用する通信プログラムが、第 1 記憶部 1 5 に記憶可能なものに制約されることがない。

【 0 0 7 2 】

上記の情報処理装置 1 において、制御機器 1 0 と内部機器 2 0 の間の通信方式は USB 通信規格に対応したものであり、内部機器 2 0 は、USB 機器であり、機器情報は、USB デバイスディスクリプターであり、通信プログラムは、SMBIOS プログラムおよび SNMP エージェントプログラムの少なくとも一方を含むことが好ましい。

【 0 0 7 3 】

この構成によれば、ネットワーク通信に対応していない USB 機器を、外部装置 1 0 0 からネットワーク NW を介して制御することができる。また、制御機器 1 0 は、SMBIOS プログラムおよび SNMP エージェントプログラムの少なくとも一方を用いて、外部装置 1 0 0 に第 2 応答情報を送信することができる。

【 0 0 7 4 】

上記の情報処理装置 1 において、制御部 1 6 は、少なくとも、制御機器 1 0 の初回起動時に、識別情報を生成することが好ましい。

【 0 0 7 5 】

この構成によれば、少なくとも、制御機器 1 0 の初回起動時に、制御機器 1 0 に識別情報を付与することができる。

【 0 0 7 6 】

上記の情報処理装置 1 において、制御部 1 6 は、制御機器 1 0 の起動時に、第 1 記憶部 1 5 に識別情報が記憶されているか否かを判別し、第 1 記憶部 1 5 に識別情報が記憶されていないと判定した場合、識別情報を生成し、第 1 記憶部 1 5 に識別情報が記憶されていると判定した場合、機器情報を生成しないことが好ましい。

10

20

30

40

50

【0077】

この構成によれば、第1記憶部15に識別情報が記憶されていないと判定した場合に、識別情報を生成するため、初回起動時や、ユーザーが意図的に識別情報を削除した場合に、制御機器10に識別情報を付与することができる。また、第1記憶部15に識別情報が記憶されていると判定した場合は、識別情報を生成しないため、誤って識別情報が変更されることがない。

【0078】

上記の情報処理装置1において、内部機器20は、印刷機構22を有することが好ましい。

【0079】

この構成によれば、内部機器20としてネットワークNWに対応していないプリンター20を一体的に構成した情報処理装置1に適応できる。外部装置100からネットワークNWを介して情報処理装置1に印刷データを送って印刷することができる。

【符号の説明】

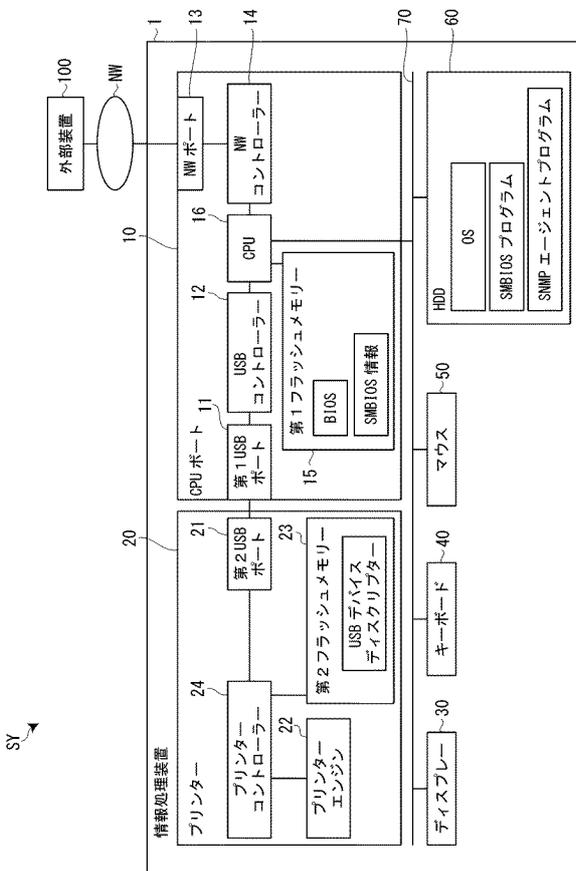
【0080】

1...情報処理装置、10...CPUボード、11...第1USBポート、12...USBコントローラー、13...ネットワークポート、14...ネットワークコントローラー、15...第1フラッシュメモリー、20...プリンター、21...第2USBポート、22...プリンターエンジン、23...第2フラッシュメモリー、24...プリンターコントローラー、30...ディスプレイ、40...キーボード、50...マウス、60...HDD、70...バス。

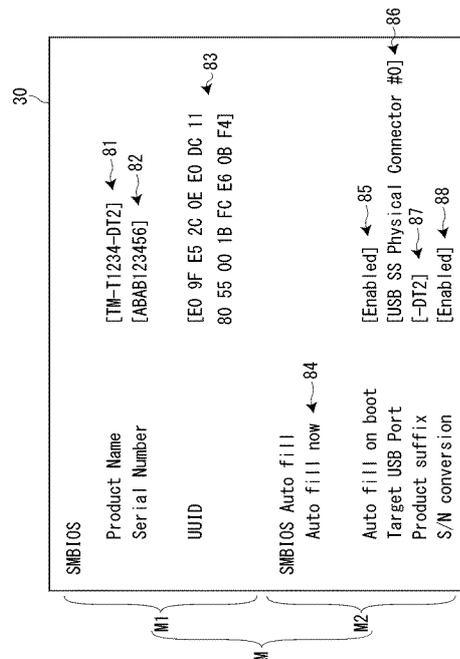
10

20

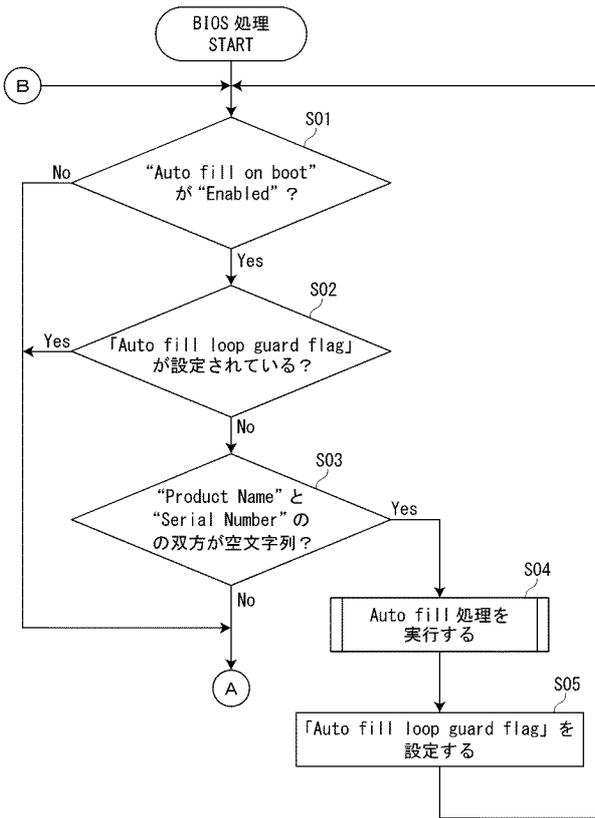
【図1】



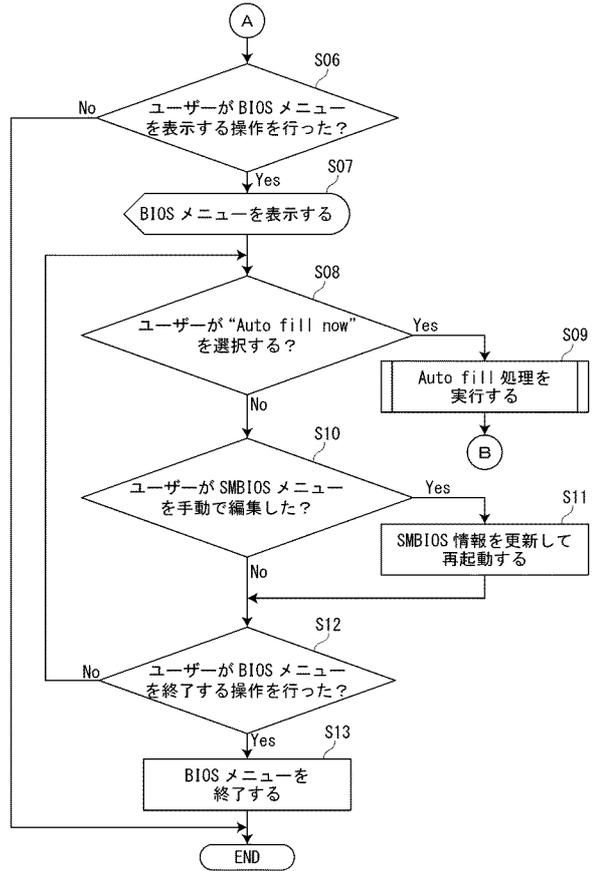
【図2】



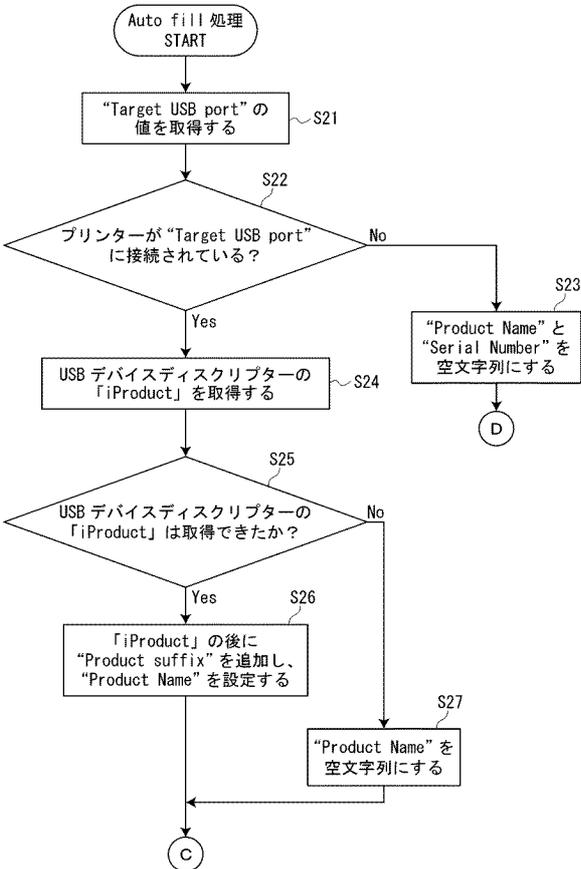
【 図 3 】



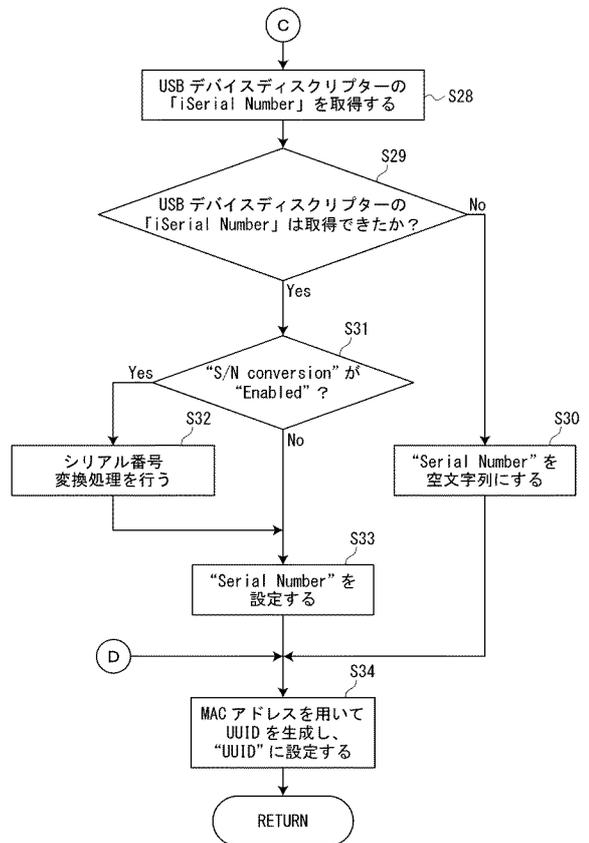
【 図 4 】



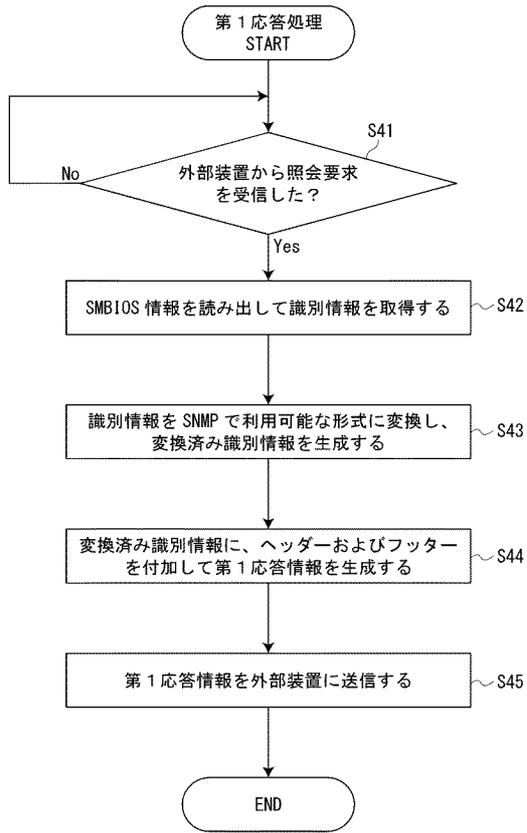
【 図 5 】



【 図 6 】



【 図 7 】



【 図 8 】

