(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2010-15267
(P2010-15267A)
(43)公開日　平成22年1月21日(2010.1.21)

(51)Int.Cl.		F I			テーマコード（参考）
G06F 11/00	(2006.01)	G06F	9/06	630E	5B176
G06F 9/445	(2006.01)	G06F	9/06	640A	
G06F 13/00	(2006.01)	G06F	13/00	530B	

審査請求 未請求　請求項の数 17　ＯＬ　（全 15 頁）

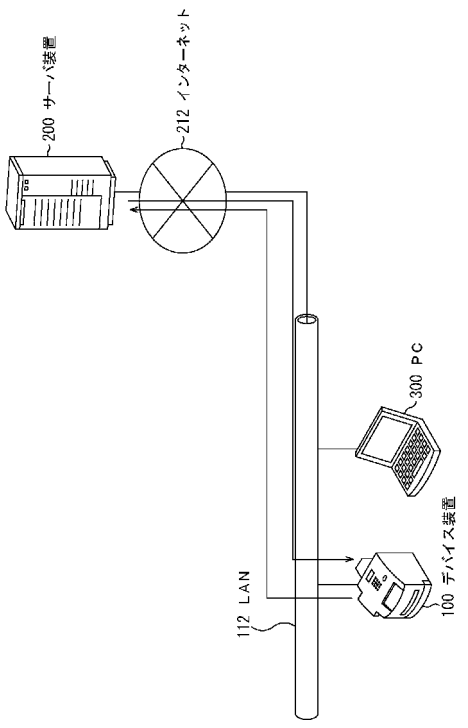
(21)出願番号　特願2008-172804 (P2008-172804)
(22)出願日　平成20年7月1日 (2008.7.1)

(71)出願人　000001007
キヤノン株式会社
東京都大田区下丸子３丁目３０番２号
(74)代理人　100090273
弁理士　國分　孝悦
(72)発明者　横倉　秀則
東京都大田区下丸子３丁目３０番２号　キヤノン株式会社内

Ｆターム(参考)　5B176 BB06　EA06

(54)【発明の名称】ソフトウェア配信システム、ソフトウェア配信方法、及びコンピュータプログラム

(57)【要約】
【課題】　ソフトウェアを配信するサーバが、従来よりも効率よく、当該ソフトウェアを多数のデバイスに配信することができるようにする。
【解決手段】　デバイス装置１００に発生する障害と、ファームウェアの更新時間とを相互に関連付けて、サーバ装置２００のダウンロード記述テーブルに予め登録しておく。デバイス装置１００は、ファームウェアの更新がサーバ装置２００から指示されると、ダウンロード記述テーブルをサーバ装置２００から取得する。そして、デバイス装置１００は、自身に発生した障害の記録と、ダウンロード記述テーブルとを照合し、自身に発生した障害がダウンロード記述テーブルに登録されている場合には、その障害に関連付いている更新時間に基づくタイミングで、ファームウェアを更新する。
【選択図】　図３



【特許請求の範囲】
【請求項１】
配信装置からデバイスにソフトウェアを配信するソフトウェア配信システムであって、
前記デバイスで生じた動作状態に関する動作状態情報を取得する取得手段と、
前記取得手段により取得された動作状態情報に基づく動作状態に応じて、前記デバイスに前記ソフトウェアを配信するタイミングを決定する決定手段と、
前記決定手段により決定されたタイミングに基づいて、前記配信装置から前記デバイスに前記ソフトウェアを配信する配信手段とを有することを特徴とするソフトウェア配信システム。
【請求項２】
前記デバイスで生じる動作状態を識別する識別子と、前記ソフトウェアを当該動作状態に応じて配信するタイミングを判別するための配信タイミング判別情報とが相互に対応付けられて登録された配信タイミング管理テーブルを保持する保持手段を有し、
前記決定手段は、前記取得手段により取得された動作状態情報に基づく動作状態を識別する識別子に対応付けられて前記配信タイミング管理テーブルに登録されている配信タイミング判別情報に基づいて、前記デバイスに前記ソフトウェアを配信するタイミングを決定することを特徴とする請求項１に記載のソフトウェア配信システム。
【請求項３】
前記配信装置が、前記保持手段を有し、
前記デバイスが、前記取得手段と、前記決定手段とを有し、
前記デバイスは、前記保持手段により保持されている配信タイミング管理テーブルを、前記配信装置から取得する第２の取得手段を更に有することを特徴とする請求項２に記載のソフトウェア配信システム。
【請求項４】
前記決定手段は、前記デバイスにおける動作を示す情報に基づいて、前記デバイスが、次に前記取得手段により取得された動作状態になる時間を算出し、算出した時間を、当該デバイスに前記ソフトウェアを配信するタイミングとして決定することを特徴とする請求項２又は３に記載のソフトウェア配信システム。
【請求項５】
前記配信装置が、前記取得手段と、前記保持手段と、前記決定手段とを有することを特徴とする請求項２に記載のソフトウェア配信システム。
【請求項６】
前記決定手段は、前記取得手段により取得された動作状態情報に基づく動作状態を識別する識別子が、前記配信タイミング管理テーブルに登録されていない場合には、予め定められたタイミングを、前記デバイスに前記ソフトウェアを配信するタイミングとして決定することを特徴とする請求項２～５の何れか１項に記載のソフトウェア配信システム。
【請求項７】
前記配信装置から前記デバイスへのソフトウェアのダウンロードに失敗すると、前記取得手段により取得された動作状態情報に基づく動作状態を識別する識別子に対応付けられて前記配信タイミング管理テーブルに登録されている配信タイミング判別情報に応じて、当該ソフトウェアの再配信を前記配信装置に要求する頻度を決定する第２の決定手段を有し、
前記第２の決定手段により決定された頻度で、前記配信装置に前記ソフトウェアの配信を要求する要求手段とを有することを特徴とする請求項１～６の何れか１項に記載のソフトウェア配信システム。
【請求項８】
前記取得手段は、前記デバイスで発生したエラーに関するエラー情報を取得し、
前記決定手段は、前記取得手段により取得されたエラー情報に基づくエラーの内容に応じて、前記デバイスに前記ソフトウェアを配信するタイミングを決定することを特徴とする請求項１～７の何れか１項に記載のソフトウェア配信システム。

【請求項９】
配信装置からデバイスにソフトウェアを配信するソフトウェア配信方法であって、
前記デバイスで生じた動作状態に関する動作状態情報を取得する取得ステップと、
前記取得ステップにより取得された動作状態情報に基づく動作状態に応じて、前記デバイスに前記ソフトウェアを配信するタイミングを決定する決定ステップと、
前記決定ステップにより決定されたタイミングに基づいて、前記配信装置から前記デバイスに前記ソフトウェアを配信することを指示する配信ステップとを有することを特徴とするソフトウェア配信方法。
【請求項１０】
前記デバイスで生じる動作状態を識別する識別子と、前記ソフトウェアを当該動作状態に応じて配信するタイミングを判別するための配信タイミング判別情報とが相互に対応付けられて登録された配信タイミング管理テーブルを保持する保持ステップを有し、
前記決定ステップは、前記取得ステップにより取得された動作状態情報に基づく動作状態を識別する識別子に対応付けられて前記配信タイミング管理テーブルに登録されている配信タイミング判別情報に基づいて、前記デバイスに前記ソフトウェアを配信するタイミングを決定することを特徴とする請求項９に記載のソフトウェア配信方法。
【請求項１１】
前記配信装置が、前記保持ステップを行い、
前記デバイスが、前記取得ステップと、前記決定ステップとを行い、
前記デバイスは、前記保持ステップにより保持されている配信タイミング管理テーブルを、前記配信装置から取得する第２の取得ステップを更に有することを特徴とする請求項１０に記載のソフトウェア配信方法。
【請求項１２】
前記決定ステップは、前記デバイスにおける動作を示す情報に基づいて、前記デバイスが、次に前記取得ステップにより取得された動作状態情報に基づく動作状態になる時間を算出し、算出した時間を、当該デバイスに前記ソフトウェアを配信するタイミングとして決定することを特徴とする請求項１０又は１１に記載のソフトウェア配信方法。
【請求項１３】
前記配信装置が、前記取得ステップと、前記保持ステップと、前記決定ステップとを行うことを特徴とする請求項１０に記載のソフトウェア配信方法。
【請求項１４】
前記決定ステップは、前記取得ステップにより取得された動作状態情報に基づく動作状態を識別する識別子が、前記配信タイミング管理テーブルに登録されていない場合には、予め定められたタイミングを、前記デバイスに前記ソフトウェアを配信するタイミングとして決定することを特徴とする請求項１０～１３の何れか１項に記載のソフトウェア配信方法。
【請求項１５】
前記配信装置から前記デバイスへのソフトウェアのダウンロードに失敗すると、前記取得ステップにより取得された動作状態情報に基づく動作状態を識別する識別子に対応付けられて前記配信タイミング管理テーブルに登録されている配信タイミング判別情報に応じて、当該ソフトウェアの再配信を前記配信装置に要求する頻度を決定する第２の決定ステップを有し、
前記第２の決定ステップにより決定された頻度で、前記配信装置に前記ソフトウェアの配信を要求する要求ステップとを有することを特徴とする請求項９～１４の何れか１項に記載のソフトウェア配信方法。
【請求項１６】
前記取得ステップは、前記デバイスで発生したエラーに関するエラー情報を取得し、
前記決定ステップは、前記取得ステップにより取得されたエラー情報に基づくエラーの内容に応じて、前記デバイスに前記ソフトウェアを配信するタイミングを決定することを特徴とする請求項９～１５の何れか１項に記載のソフトウェア配信方法。

【請求項１７】
請求項９～１６の何れか１項に記載のソフトウェア配信方法の各ステップをコンピュータに実行させることを特徴とするコンピュータプログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、ソフトウェア配信システム、ソフトウェア配信方法、及びコンピュータプログラムに関し、特に、ファームウェア等のソフトウェアをサーバからデバイスに配信するために用いて好適なものである。
【背景技術】
【０００２】
ＭＦＰ（Multi Function Peripheral）のようなデバイスのファームウェアの不具合への対応や機能の追加を行うため、即時に又は不定期にファームウェアの更新を行うようなシステムが開発されてきている。このようなシステムを実現する形態としては、インターネットを通じて、デバイスがサーバからファームウェアを受信して更新する形態や、イントラネット上で、サーバがデバイスにファームウェアを送信して更新する形態等、様々な形態がある。また、このようなシステムでサーバが監視するデバイスの数は年々増加する傾向にある。また、このようなシステムは今後ますます増加することが予想される。
【０００３】
そして、このようなシステムにおける従来の技術として特許文献１に記載の技術がある。特許文献１に記載の技術では、トラブルが発生したファームウェアを新たなファームウェアに更新するに際し、新たなファームウェアがどのモジュールを制御するものであるかを判定し、そのモジュールを停止させた後に、新たなファームウェアを更新している。
【０００４】
【特許文献１】特開２００４－１６５７３４号公報
【発明の開示】
【発明が解決しようとする課題】
【０００５】
しかしながら、前述した従来の技術では、多数のデバイスがファームウェアの配信対象となる場合には、それら多数のデバイスに一度にファームウェアを配信して更新することになる。したがって、ファームウェアを配信するサーバの負荷が集中してしまい、サーバをダウンさせてしまう虞がある。また、デバイスがファームウェアの更新を無造作に行ってしまうと、ファームウェアを頻繁に更新することになり、ネットワークのトラフィックを高めてしまう可能性もある。
【０００６】
本発明は、このような問題点に鑑みてなされたものであり、ファームウェア等のソフトウェアを配信するサーバが、従来よりも効率よく、当該ソフトウェアを多数のデバイスに配信することができるようにすることを目的とする。
【課題を解決するための手段】
【０００７】
本発明のソフトウェア配信システムは、配信装置からデバイスにソフトウェアを配信するソフトウェア配信システムであって、前記デバイスで生じた動作状態に関する動作状態情報を取得する取得手段と、前記取得手段により取得された動作状態情報に基づく動作状態に応じて、前記デバイスに前記ソフトウェアを配信するタイミングを決定する決定手段と、前記決定手段により決定されたタイミングに基づいて、前記配信装置から前記デバイスに前記ソフトウェアを配信する配信手段とを有することを特徴とする。
【０００８】
本発明のソフトウェア配信方法は、配信装置からデバイスにソフトウェアを配信するソフトウェア配信方法であって、前記デバイスで生じた動作状態に関する動作状態情報を取得する取得ステップと、前記取得ステップにより取得された動作状態情報に基づく動作状

態に応じて、前記デバイスに前記ソフトウェアを配信するタイミングを決定する決定ステップと、前記決定ステップにより決定されたタイミングに基づいて、前記配信装置から前記デバイスに前記ソフトウェアを配信することを指示する配信ステップとを有することを特徴とする。

【０００９】

本発明のコンピュータプログラムは、前記ソフトウェア配信方法の各ステップをコンピュータに実行させることを特徴とする。

【発明の効果】

【００１０】

本発明によれば、デバイスの動作状態に応じて、ソフトウェアを配信するタイミングを決定するようにしたので、ソフトウェアが多数のデバイスに同時に配信されることを可及的に防止することができる。したがって、デバイスで使用されるソフトウェアを配信するサーバが、従来よりも効率よく、当該ソフトウェアを多数のデバイスに配信することができる。

【発明を実施するための最良の形態】

【００１１】

（第１の実施形態）

以下に、図面を参照しながら、本発明の第１の実施形態について説明する。

図１は、デバイス装置の内部構成の一例を示した図である。

本実施形態では、デバイス装置１００は、例えばＭＦＰであり、ＣＰＵ１０１においてネットワーク制御部とデバイス（プリンタ、コピー、スキャナ、ＦＡＸ）制御部とが動作する形態で実現されている。このＣＰＵ１０１は、ＲＯＭ１０３に格納されているファームウェアを実行する。また、ＣＰＵ１０１は、システムバス１１１を介して相互に接続される各装置１０２～１０７を総括的に制御する。また、ＣＰＵ１０１は、ＬＡＮ１１２等のネットワークを介して相互に接続されているサーバ装置２００（図２、図３を参照）からデータを受信して、ＲＯＭ１０３に格納されているファームウェアを更新させることが可能である。

【００１２】

ＲＡＭ１０２は、ＣＰＵ１０１の主メモリ、ワークエリア等として機能する。また、ＲＡＭ１０２は、デバイス装置１００の設定値を保存するバックアップＲＡＭとしても機能する。ＬＣＤＣコントローラ（ＬＣＤＣ）１０４は、液晶ディスプレイ（ＬＣＤ）１０８の表示を制御する。液晶ディスプレイ（ＬＣＤ）１０８は、タッチパネルの機能を有している。

ディスクコントローラ（ＤＫＣ）１０５は、ハードディスク（ＨＤ）１０９を制御し、画像や様々なユーザデータ等を保存する。デバイスコントローラ（ＤＶＣ）１０６は、プリンタ等のデバイス（ＤＶ）１１０を制御する。ネットワークインタフェースカード（ＮＩＣ）１０７は、ＬＡＮ１１２を介して相互に接続されているサーバ装置２００との通信を行い、自身のファームウェア等を受信することができる。

尚、ハードディスク（ＨＤ）１０９はなくてもよいし、また、データを保存することができれば、ハードディスク（ＨＤ）１０９の代わりにＵＳＢメモリ等を用いてもよい。また、ファームウェア等のソフトウェアは、ＲＯＭ１０３に格納されていてもハードディスク（ＨＤ）１０９に格納されていてもよい。また、デバイス装置１００には、ユーザが操作する各種のスイッチが設けられており、ＣＰＵ１０１は、スイッチの操作の内容に応じた処理を実行する。

【００１３】

図２は、サーバ装置２００の内部構成の一例を示した図である。

ＣＰＵ２０１は、ＲＯＭ２０３に格納されているプログラムを実行し、システムバス１１１を介して相互に接続される各装置２０２～２０７を総括的に制御する。ＲＡＭ２０２は、ＣＰＵ２０１の主メモリ、ワークエリア等として機能する。また、ＲＡＭ２０２は、サーバ装置２００の設定値を保存するバックアップＲＡＭとしても機能する。

キーボードコントローラ（ＫＢＣ）２０４は、キーボード（ＫＢ）２０８からの入力を制御する。ＣＲＴコントローラ（ＣＲＴＣ）２０５は、ＣＲＴディスプレイ（ＣＲＴ）２０９の表示を制御する。

【００１４】

ディスクコントローラ（ＤＫＣ）２０６は、ハードディスク（ＨＤ）２１０を制御する。このディスクコントローラ（ＤＫＣ）２０６によって、ハードディスク（ＨＤ）２１０にプログラムを保存しておくことが可能である。ネットワークインタフェースカード（ＮＩＣ）２０７は、インターネット２１２を介して相互に接続されているデバイス装置１００にファームウェアを送信することができる。

尚、ハードディスク（ＨＤ）２１０はなくてもよいし、また、データを保存することができれば、ハードディスク（ＨＤ）２１０の代わりにＵＳＢメモリ等を用いてもよい。また、プログラムは、ＲＯＭ２０３に格納されていてもハードディスク（ＨＤ）２１０に格納されていてもよい。

【００１５】

図３は、本実施形態におけるファームウェア配信システムの全体構成の一例を示す図である。図３において、デバイス装置１００とサーバ装置２００との間に示されている矢印は、データ又は指示の流れを示している。

デバイス装置１００は、ＰＣ３００から印刷ジョブを受信したり、ユーザによる操作に基づいてコピーを行ったりすることが可能である。そして、デバイス装置１００は、自身のファームウェアの更新指示があるかどうかを、サーバ装置２００に定期的に確認する。サーバ装置２００は、図４を用いて後述するダウンロード記述ファイルを更新したり、更新可能なファームウェアをデバイス装置１００に送信したりすることができる。

尚、サーバ装置２００、デバイス装置１００、ＰＣ３００共、図１～図３に示す構成に限定されるものではなく、夫々１又は複数の組み合わせでもよい。また、インターネット２１２を経由せずに、サーバ装置２００を構成するようにしてもよい。

以上のように本実施形態では、例えば、デバイス装置１００によりデバイスの一例が実現され、サーバ装置２００により配信装置の一例が実現される。

【００１６】

図４は、サーバ装置２００に格納されているダウンロード記述テーブルの一例を示す図である。本実施形態では、このダウンロード記述テーブルは、サーバ装置２００が管理するデバイス装置１００毎に作成されている。

図４において、ＫＥＹ４０１は、デバイス装置１００に発生する障害（エラー）に関するキーワード（エラー発生キーワード）であり、"モジュール名：動作状態"で表される。また、UPDATE TIME４０２は、ＫＥＹ４０１として登録されている夫々のキーワードに対応する"ファームウェアの更新時間"である。

"SLP：REBOOT"は、ＳＬＰ（Serial Link Protocol）パケットを受けるたびにリブートされるという障害の発生を示している。図４に示す例では、"SLP：REBOOT"に対しては、"即時"にダウンロードを開始することを示す"Immedately"を登録している。また、"RDS：COUNTER ERROR"は、２４時間以内にカウンタ送信を行うカウンタ送信モジュールにおける送信時のエラーの発生を示している。図４に示す例では、"RDS：COUNTER ERROR"に対しては、次回エラーが発生するであろう"２４時間以内"にダウンロードを行うことを示す"Less than 24 hours"を登録している。

【００１７】

"PCL：PROCESS ERROR"は、ＰＣＬ（Printer Control Language）の印刷エラーの発生を示している。図４に示す例では、"PCL：PROCESS ERROR"に対しては、"即時"にダウンロードを開始することを示す"Immedately"を登録している。また、"SCON：SLEEP ERROR"は、電力制御系のエラーの発生を示している。図４に示す例では、"SCON：SLEEP ERROR"に対しては、次回エラーが発生するであろう"スリープモードへの移行時"までにダウンロードを行うことを示す"Before Sleeping"を登録している。

【００１８】

"UI：MESSAGE ERROR"は、ＵＩ（User Interface）に表示するメッセージ（文言）に関するエラーの発生を示している。図４に示す例では、"UI：MESSAGE ERROR"に対しては、ダウンロードを行う緊急性を落として、"２４時間以内"にダウンロードを行うことを示す"Less than 24 hours"を登録している。また、"REPORT：PRINT ERROR"は、レポート出力に関するエラーの発生を示している。図４に示す例では、"REPORT：PRINT ERROR"に対しては、"次のレポート出力時間の前"までにダウンロードを行うことを示す"Before next time"を登録している。

【００１９】

尚、本実施形態では、ＫＥＹ４０１に対応付けて、ファームウェアの更新時間であるUPDATE TIME４０２を登録する場合を例に挙げて説明した。しかしながら、ファームウェアの更新時間を判別できる情報であれば、必ずしもファームウェアの更新時間そのものを登録しなくてもよい。例えば、UPDATE TIME４０２の代わりに更新レベルを登録してもよい。このようにした場合、その更新レベルの値に応じて、ファームウェアをダウンロードするタイミングが決定されるようにする。例えば、"Immedately"に相当する更新レベルをレベル１とする。また、"Before Sleeping"、"Before next time"に相当する更新レベルをレベル２とするまた、"Less than 24 hours"に相当する更新レベルをレベル３とする。そして、レベル１が登録されている場合には即時、レベル２が登録されている場合には１時間以内、レベル３が登録されている場合には２４時間以内にファームウェアをダウンロードするような制御を行うようにする。

【００２０】

次に、図５のフローチャートを参照しながら、デバイス装置１００における処理の一例を説明する。図５のフローチャートは、例えば、デバイス装置１００のＣＰＵ１０１がＲＯＭ１０３に記憶されているファームウェア等のプログラム（ソフトウェア）を実行することにより実現される。

まず、ステップＳ５０１では、ＣＰＵ１０１は、デバイス装置１００内部の動作状態を記録したログの収集を順次行う。

以上のように本実施形態では、例えば、デバイス装置１００内部の動作状態を記録したログにより、動作状態情報（エラー情報）の一例が実現され、ステップＳ５０１の処理を行うことにより取得手段の一例が実現される。

次に、ステップＳ５０２では、ＣＰＵ１０１は、ファームウェアの更新指示があるかどうかを、サーバ装置２００に確認する。尚、サーバ装置２００に確認せずに、デバイス装置１００側でファームウェアの更新指示があるかどうかを判断することもできる。このようにする場合には、ファームウェアの更新指示をデバイス装置１００がサーバ装置２００に行うことになる。

次に、ステップＳ５０３では、ＣＰＵ１０１は、サーバ装置２００から、ファームウェアの更新指示があるかどうかを判別する。この判別の結果、ファームウェアの更新指示がない場合には、図５のフローチャートによる処理を終了する。

【００２１】

一方、ファームウェアの更新指示がある場合には、ステップＳ５０４へ進む。ステップＳ５０４では、ＣＰＵ１０１は、図４で示したダウンロード記述テーブルを、サーバ装置２００から取得する。

次に、ステップＳ５０５では、ＣＰＵ１０１は、ステップＳ５０１で収集したログの内容と、ステップＳ５０４で取得したダウンロード記述ファイルにあるＫＥＹ４０１の内容から、夫々エラー発生キーワードを検索する。

次に、ステップＳ５０６では、ＣＰＵ１０１は、ステップＳ５０５の検索の結果に基づいて、ログから得られたエラー発生キーワードに一致するエラー発生キーワードが、ダウンロード記述ファイル（ＫＥＹ４０１）に登録されているかどうかを判別する。

【００２２】

以上のように本実施形態では、例えば、図４に示すダウンロード記述テーブルにより配信タイミング管理テーブルの一例が実現される。また、例えば、ダウンロード記述テーブ

ルのＫＥＹ４０１の情報、UPDATE TIME４０２の情報により、夫々、識別子、配信タイミング判別情報が実現される。そして、サーバ装置２００がダウンロード記述テーブルを記憶することにより保持手段の一例が実現される。また、例えば、ステップＳ５０４の処理を行うことにより第２の取得手段の一例が実現される。

【００２３】

ステップＳ５０６の判別の結果、ログから得られたエラー発生キーワードに一致するエラー発生キーワードが、ダウンロード記述ファイル（ＫＥＹ４０１）に登録されていなければ、図５のフローチャートによる処理を終了する。

【００２４】

一方、ログから得られたエラー発生キーワードに一致するエラー発生キーワードが、ダウンロード記述ファイル（ＫＥＹ４０１）に登録されている場合には、ステップＳ５０７へ進む。ステップＳ５０７では、ＣＰＵ１０１は、ログから得られたエラー発生キーワードに一致するエラー発生キーワードに対応付けられてダウンロード記述ファイルに登録されているUPDATE TIME４０２の内容（更新時間）を確認する。

次に、ステップＳ５０８では、ＣＰＵ１０１は、ステップＳ５０７で確認した更新時間が"Before Next time"であるかどうかを判別する。この判別の結果、ステップＳ５０７で確認した更新時間が"Before Next time"でなければ、ステップＳ５０９へ進み、ステップＳ５０７で確認した更新時間が"Before Next time"であれば、ステップＳ５１０へ進む。

【００２５】

ステップＳ５０９では、ＣＰＵ１０１は、ステップＳ５０７で確認した更新時間に従って、サーバ装置２００から配信されたファームウェアの更新を行う。

一方、ステップＳ５１０では、ＣＰＵ１０１は、デバイスの動作を示すデバイス動作情報の設定に基づいて、次回のエラー発生時刻を算出する。図４に示す例では、次回のレポート出力時間を算出する。

次に、ステップＳ５１１では、ＣＰＵ１０１は、ステップＳ５１０で取得した次回のエラー発生時刻よりも前に、サーバ装置２００から配信されたファームウェアの更新を行う。

以上のように本実施形態では、例えば、ステップＳ５０６～５１１の処理を行うことにより決定手段の一例が実現される。また、例えば、ステップＳ５０９、Ｓ５１１で、サーバ装置２００がファームウェアをデバイス装置１００に配信することにより配信手段の一例が実現される。

【００２６】

以上のように本実施形態では、デバイス装置１００に発生する障害と、ファームウェアの更新時間とを相互に関連付けて、サーバ装置２００のダウンロード記述テーブルに予め登録しておく。デバイス装置１００は、ファームウェアの更新がサーバ装置２００から指示されると、ダウンロード記述テーブルをサーバ装置２００から取得する。そして、デバイス装置１００は、自身に発生した障害の記録と、ダウンロード記述テーブルとを照合し、自身に発生した障害がダウンロード記述テーブルに登録されている場合は、その障害に関連付いている更新時間に基づくタイミングで、ファームウェアを更新する。

【００２７】

したがって、サーバ装置２００からファームウェアを多数のデバイス装置１００に配信する場合であっても、それまでのデバイス装置１００の動作に応じて、ファームウェアを更新するタイミングを変えることが可能になる。したがって、サーバ装置２００の負荷を軽減させることができ、効率的にファームウェアを配信することができる。また、デバイス装置１００も効率的にファームウェアのダウンロードを行うことができ、デバイス装置１００とサーバ装置２００とを相互に接続するネットワークのトラフィックを軽減させることが可能となる。

【００２８】

尚、ステップＳ５０９では、UPDATE TIME４０２で指定された更新時間に従ってファームウェアを更新するようにしている。したがって、UPDATE TIME４０２が"Immediately"で

あれば、ＣＰＵ１０１が即時にファームウェアの更新を行うことになる。
　また、本実施形態では、ステップＳ５０６でエラー発生キーワードが一致していない場合には、何もせずに処理を終了するようにしたが、このような場合には、定期的（例えば１カ月に１回）にファームウェアのダウンロードを行ってもよい。
【００２９】
　また、本実施形態では、ステップＳ５０１では、ログの収集を行うようにしたが、ログの収集を行う代わりに、デバイス装置１００のエラーを単に監視するようにしてもよい。
　また、前述したように、UPDATE TIME４０２（更新時間）の代わりに更新レベルを用いて、更新レベルに応じたダウンロードの時間を決定してダウンロードを実行してもよい。
　また、サーバ装置２００からのファームウェアのダウンロードが失敗した場合には、エラーが発生した場合にデバイス装置１００が処理（例えばファームウェアの配信要求）をリトライする頻度を、更新時間や更新レベルに応じて変更するようにしてもよい。このようにした場合には、デバイス装置１００は、変更された頻度に従って、ファームウェアの再配信をサーバ装置２００に要求することになる。本実施形態では、例えば、このような変形をすることによって、要求手段、第２の決定手段の一例が実現される。
　また、本実施形態では、ファームウェアを配信する場合を例に挙げて説明したが、ファームウェア以外のソフトウェア（例えばアプリケーションプログラム）を配信するようにしてもよい。
【００３０】
（第２の実施形態）
　次に、本発明の第２の実施形態について説明する。前述した第１の実施形態では、ファームウェアを更新するタイミングをデバイス装置１００で判別する場合を例に挙げて説明した。これに対し、本実施形態では、ファームウェアを更新するタイミングをサーバ装置２００で判別する場合を例に挙げて説明する。このように本実施形態と前述した第１の実施形態とは、ファームウェアを更新するタイミングを判別する主体が主として異なる。したがって、本実施形態の説明において、前述した第１の実施形態と同一の部分については、図１～図７に付した符号と同一の符号を付すこと等により、詳細な説明を省略する。例えば、サーバ装置及びデバイス装置の内部構成は、第１の実施形態のものと同様の構成で実現することができる。
【００３１】
　図６は、本実施形態におけるファームウェア配信システムの全体配置の一例を示す図である。図６において、デバイス装置６００ａ、６００ｂとサーバ装置７００との間に示されている矢印は、データ又は指示の流れを示している。
　デバイス装置６００は、ＰＣ３００から印刷ジョブを受信したり、デバイス装置１００のユーザによる操作に基づいてコピー等を行ったりすることが可能である。そして、デバイス装置６００は、自身のファームウェアの更新指示があるかどうかを、サーバ装置７００に定期的に確認する。サーバ装置７００は、図７を用いて後述するダウンロード記述ファイルを更新したり、更新可能なファームウェアをデバイス装置６００に送信したりすることができる。
　尚、サーバ装置７００、デバイス装置６００、ＰＣ３００共、図１、図２、図６に示す構成に限定されるものではなく、夫々１又は複数の組み合わせでもよい。また、図３と同様に、サーバ装置７００と、デバイス装置６００、ＰＣ３００とが、インターネット２１２を経由して通信するようにシステムを構成するようにしてもよい。
　以上のように本実施形態では、例えば、デバイス装置６００によりデバイスの一例が実現され、サーバ装置７００により配信装置の一例が実現される。
【００３２】
　図７は、サーバ装置７００に格納されているダウンロード記述テーブルの一例を示す図である。本実施形態でも、第１の実施形態と同様に、このダウンロード記述テーブルは、サーバ装置７００が管理するデバイス装置６００毎に作成されている。
　図７に示すダウンロード記述テーブルは、基本的に図４に示したものと同様である。す

なわち、ＫＥＹ７０１は、デバイス装置６００に発生する障害に関するキーワード（エラー発生キーワード）であり、"モジュール名：動作状態"で表される。また、UPDATE TIME７０２は、ＫＥＹ７０１として登録されている夫々のキーワードに対応する"ファームウェアの更新時間"である。

【００３３】

"SLP：REBOOT"は、ＳＬＰパケットを受けるたびにリブートされるという障害の発生を示している。図７に示す例では、"SLP：REBOOT"に対しては、"即時"にダウンロードを開始することを示す"Immedately"を登録している。"RDS：COUNTER ERROR"は、２４時間以内にカウンタ送信を行うカウンタ送信モジュールにおける送信時のエラーの発生を示している。図７に示す例では、"RDS：COUNTER ERROR"に対しては、次回エラーが発生するであろう"２４時間以内"にダウンロードを行うことを示す"Less than 24 hours"を登録している。

【００３４】

"PCL：PROCESS ERROR"は、ＰＣＬの印刷エラーの発生を示している。図７に示す例では、"PCL：PROCESS ERROR"に対しては、"即時"にダウンロードを開始することを示す"Immedately"を登録している。"UI：MESSAGE ERROR"は、ＵＩ（User Interface）に表示するメッセージ（文言）に関するエラーの発生を示している。図７に示す例では、"UI：MESSAGE ERROR"に対しては、ダウンロードを行う緊急性を落として、"２４時間以内"にダウンロードを行うことを示す"Less than 24 hours"を登録している。

尚、本実施形態においても、ＫＥＹ７０１に対応付けて、ファームウェアの更新時間であるUPDATE TIME７０２を登録する場合を例に挙げて説明した。しかしながら、第１の実施形態で説明した様に、UPDATE TIME７０２の代わりに、更新時間以外の更新レベル等を登録してもよい。

【００３５】

次に、図８のフローチャートを参照しながら、サーバ装置７００における処理の一例を説明する。

まず、ステップＳ８０１では、ＣＰＵ２０１は、ファームウェアの更新指示をデバイス装置６００に対して行うかどうかを確認する。尚、ファームウェアの更新指示が各デバイス装置６００からきているかどうかをサーバ装置７００側で判断することもできる。

【００３６】

次に、ステップＳ８０２では、ＣＰＵ２０１は、ファームウェアの更新指示がなされているデバイスであるかどうかを判別する。この判別の結果、ファームウェアの更新指示がなされているデバイスがない場合には、図８のフローチャートによる処理を終了する。

一方、ファームウェアの更新指示がなされているデバイスがある場合には、ステップＳＳ８０３へ進む。ステップＳ８０３では、ＣＰＵ２０１は、デバイス装置６００から、デバイス装置６００内部の動作状態を記録したログを取得する。

以上のように本実施形態では、例えば、デバイス装置１００内部の動作状態を記録したログにより、動作状態情報（エラー情報）の一例が実現され、ステップＳ８０３の処理を行うことにより取得手段の一例が実現される。

【００３７】

次に、ステップＳ８０４では、ＣＰＵ２０１は、図７に示したダウンロード記述テーブルを解析する。

次に、ステップＳ８０５では、ＣＰＵ２０１は、ステップＳ８０３で取得したログの内容と、ステップＳ８０４で解析したダウンロード記述ファイルにあるＫＥＹ７０１の内容から、夫々エラー発生キーワードを検索する。

次に、ステップＳ８０６では、ＣＰＵ２０１は、ステップＳ８０５の検索の結果に基づいて、ログから得られたエラー発生キーワードに一致するエラー発生キーワードが、ダウンロード記述ファイル（ＫＥＹ７０１）に登録されているかどうかを判別する。

以上のように本実施形態では、例えば、図７に示すダウンロード記述テーブルにより配信タイミング管理テーブルの一例が実現される。また、例えば、ダウンロード記述テーブ

ルのＫＥＹ７０１の情報、UPDATE TIME７０２の情報により、夫々、識別子、配信タイミング判別情報が実現される。そして、サーバ装置７００がダウンロード記述テーブルを記憶することにより保持手段の一例が実現される。

【００３８】

ステップＳ８０６の判別の結果、ログから得られたエラー発生キーワードに一致するエラー発生キーワードが、ダウンロード記述ファイル（ＫＥＹ７０１）に登録されていなければ、図５のフローチャートによる処理を終了する。

【００３９】

一方、ログから得られたエラー発生キーワードに一致するエラー発生キーワードが、ダウンロード記述ファイル（ＫＥＹ７０１）に登録されている場合には、ステップＳ８０７へ進む。ステップＳ８０７では、ＣＰＵ２０１は、ログから得られたエラー発生キーワードに一致するエラー発生キーワードに対応付けられてダウンロード記述ファイルに登録されているUPDATE TIME７０２の内容（更新時間）を確認する。

次に、ステップＳ８０８では、ＣＰＵ２０１は、ステップＳ８０７で確認した更新時間が"Immediately"であるかどうかを判別する。この判別の結果、ステップＳ８０７で確認した更新時間が"Immediately"であれば、ステップＳ８０９へ進み、ステップＳ８０７で確認した更新時間が"Immediately"でなければ、ステップＳ８１０へ進む。

【００４０】

ステップＳ８１０では、ＣＰＵ２０１は、ステップＳ８０７で確認した更新時間に従って該当するデバイス装置６００に、ファームウェアを配信して、ファームウェアの更新を指示する。

一方、ステップＳ８０９では、ＣＰＵ２０１は、該当するデバイス装置６００にファームウェアを即時に配信して、ファームウェアの更新を指示する。

以上のように本実施形態では、例えば、ステップＳ８０６～８１０の処理を行うことにより決定手段の一例が実現される。

以上のように、ファームウェアを更新するタイミングをサーバ装置７００側で判別するようにしても、前述した第１の実施形態と同様の効果を得ることができる。

【００４１】

尚、本実施形態では、ステップＳ８０６でエラー発生キーワードが一致していない場合には、何もせずに処理を終了するようにしたが、このような場合には、定期的（例えば１カ月に１回）にファームウェアのダウンロードを行ってもよい。

また、ステップＳ８０３では、ログの取得を行うようにしたが、ログの取得を行う代わりに、デバイス装置１００のエラーを単に監視するようにしてもよい。

また、前述したように、UPDATE TIME７０２（更新時間）の代わりに更新レベルを用いて、更新レベルに応じたダウンロードの時間を決定してダウンロードを実行してもよい。

【００４２】

また、第１の実施形態と同様に、サーバ装置７００からのファームウェアのダウンロードが失敗した場合には、エラーが発生した場合にデバイス装置６００が処理をリトライする頻度を、更新時間や更新レベルに応じて変更してもよい。このようにした場合には、デバイス装置１００は、変更された頻度に従って、ファームウェアの再配信をサーバ装置２００に要求することになる。本実施形態では、例えば、このような変形をすることによって、要求手段、第２の決定手段の一例が実現される。　また、第１の実施形態と同様に、ファームウェア以外のソフトウェア（例えばアプリケーションプログラム）を配信するようにしてもよい。

【００４３】

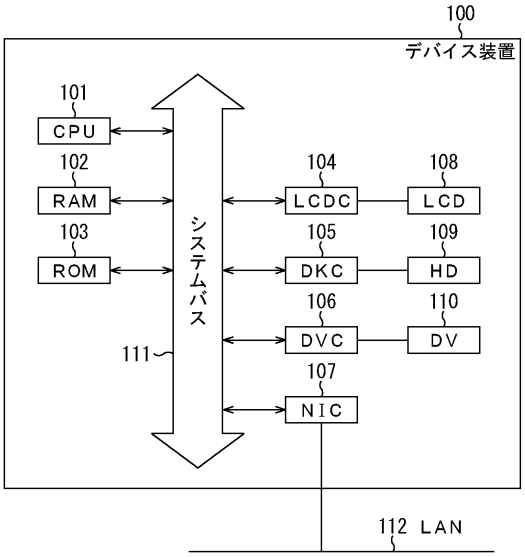
（本発明の他の実施形態）

前述した本発明の実施形態におけるソフトウェア配信システムを構成する各手段、並びにソフトウェア配信方法の各ステップは、コンピュータのＲＡＭやＲＯＭなどに記憶されたプログラムが動作することによって実現できる。このプログラム及び前記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は本発明に含まれる。

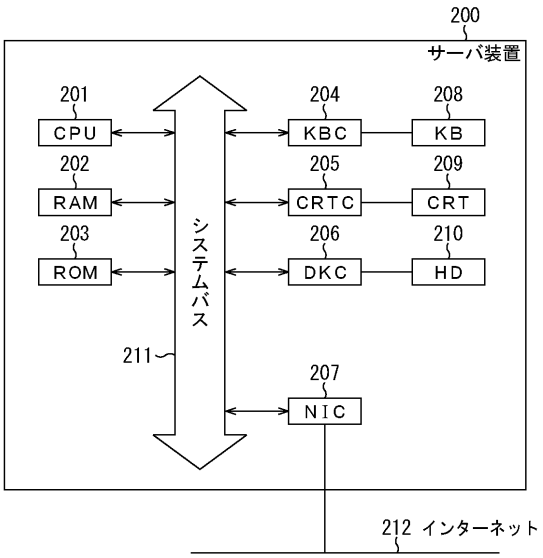
【００４４】
　また、本発明は、例えば、システム、装置、方法、プログラム若しくは記憶媒体等としての実施形態も可能であり、具体的には、複数の機器から構成されるシステムに適用してもよいし、また、一つの機器からなる装置に適用してもよい。
【００４５】
　尚、本発明は、前述した実施形態の機能を実現するソフトウェアのプログラム（実施形態では図５、図８に示すフローチャートに対応したプログラム）を、システムあるいは装置に直接、あるいは遠隔から供給するものを含む。そして、そのシステムあるいは装置のコンピュータが前記供給されたプログラムコードを読み出して実行することによっても達成される場合も本発明に含まれる。
【００４６】
　したがって、本発明の機能処理をコンピュータで実現するために、前記コンピュータにインストールされるプログラムコード自体も本発明を実現するものである。つまり、本発明は、本発明の機能処理を実現するためのコンピュータプログラム自体も含まれる。
【００４７】
　その場合、プログラムの機能を有していれば、オブジェクトコード、インタプリタにより実行されるプログラム、ＯＳに供給するスクリプトデータ等の形態であってもよい。
【００４８】
　プログラムを供給するための記録媒体としては、例えば、フロッピー（登録商標）ディスク、ハードディスク、光ディスク、光磁気ディスク、ＭＯ、ＣＤ－ＲＯＭ、ＣＤ－Ｒ、ＣＤ－ＲＷなどがある。また、磁気テープ、不揮発性のメモリカード、ＲＯＭ、ＤＶＤ（ＤＶＤ－ＲＯＭ，ＤＶＤ－Ｒ）などもある。
【００４９】
　その他、プログラムの供給方法としては、クライアントコンピュータのブラウザを用いてインターネットのホームページに接続する。そして、前記ホームページから本発明のコンピュータプログラムそのもの、若しくは圧縮され自動インストール機能を含むファイルをハードディスク等の記録媒体にダウンロードすることによっても供給できる。
【００５０】
　また、本発明のプログラムを構成するプログラムコードを複数のファイルに分割し、それぞれのファイルを異なるホームページからダウンロードすることによっても実現可能である。つまり、本発明の機能処理をコンピュータで実現するためのプログラムファイルを複数のユーザに対してダウンロードさせるＷＷＷサーバも、本発明に含まれるものである。
【００５１】
　また、本発明のプログラムを暗号化してＣＤ－ＲＯＭ等の記憶媒体に格納してユーザに配布し、所定の条件をクリアしたユーザに対し、インターネットを介してホームページから暗号化を解く鍵情報をダウンロードさせる。そして、ダウンロードした鍵情報を使用することにより暗号化されたプログラムを実行してコンピュータにインストールさせて実現することも可能である。
【００５２】
　また、コンピュータが、読み出したプログラムを実行することによって、前述した実施形態の機能が実現される。その他、そのプログラムの指示に基づき、コンピュータ上で稼動しているＯＳなどが、実際の処理の一部又は全部を行い、その処理によっても前述した実施形態の機能が実現され得る。
【００５３】
　さらに、記録媒体から読み出されたプログラムが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書き込まれる。その後、そのプログラムの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるＣＰＵなどが実際の処理の一部又は全部を行い、その処理によっても前述した実施形態の機能が実現される。

【００５４】
尚、前述した各実施形態は、何れも本発明を実施するにあたっての具体化の例を示したものに過ぎず、これらによって本発明の技術的範囲が限定的に解釈されてはならないものである。すなわち、本発明はその技術思想、又はその主要な特徴から逸脱することなく、様々な形で実施することができる。
【図面の簡単な説明】
【００５５】
【図１】本発明の第１の実施形態を示し、デバイス装置の内部構成の一例を示した図である。
【図２】本発明の第１の実施形態を示し、サーバ装置の内部構成の一例を示した図である。
【図３】本発明の第１の実施形態を示し、ファームウェア配信システムの全体構成の一例を示す図である。
【図４】本発明の第１の実施形態を示し、ダウンロード記述テーブルの一例を示す図である。
【図５】本発明の第１の実施形態を示し、デバイス装置における処理の一例を説明するフローチャートである。
【図６】本発明の第２の実施形態を示し、ファームウェア配信システムの全体配置の一例を示す図である。
【図７】本発明の第２の実施形態を示し、ダウンロード記述テーブルの一例を示す図である。
【図８】本発明の第２の実施形態を示し、サーバ装置における処理の一例を説明するフローチャートである。
【符号の説明】
【００５６】
１００、６００　デバイス装置
１１２　ＬＡＮ
２００、７００　サーバ装置
２１２　インターネット

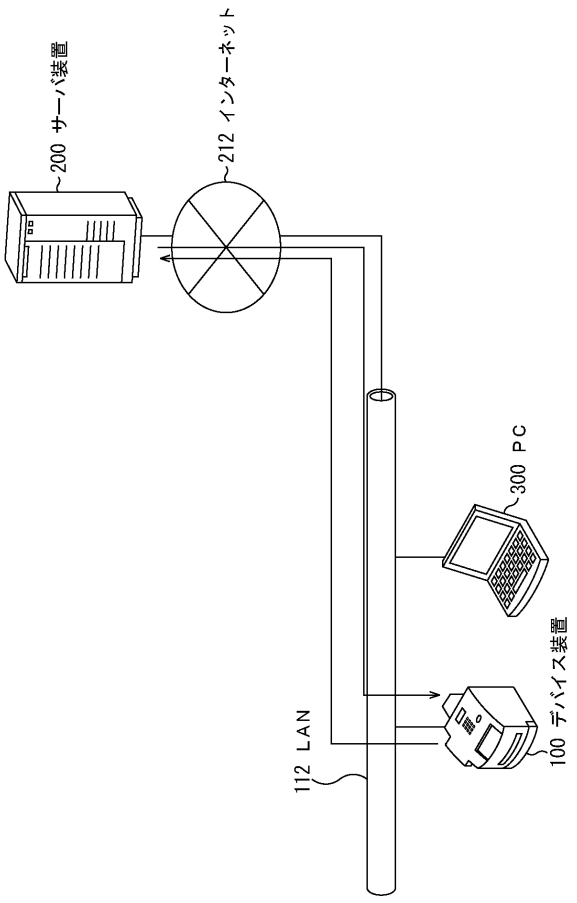
【図１】



【図２】



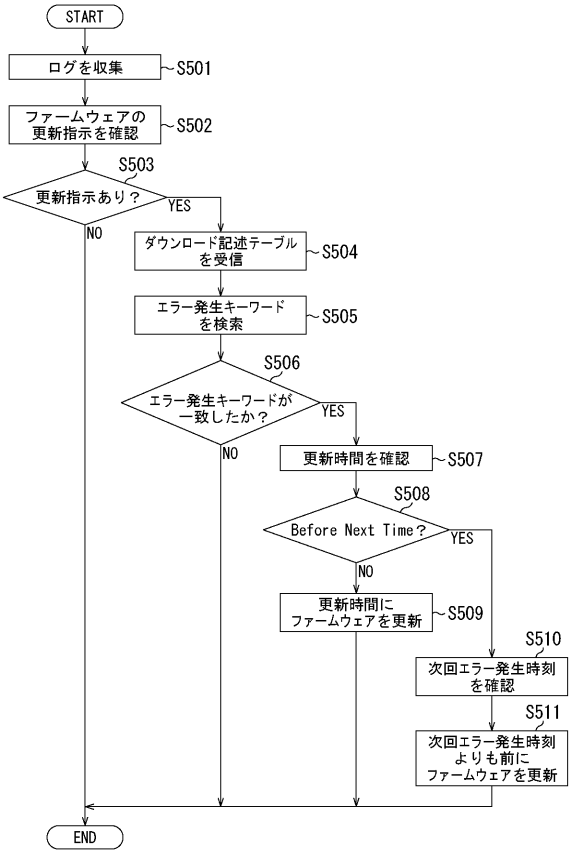
【図３】



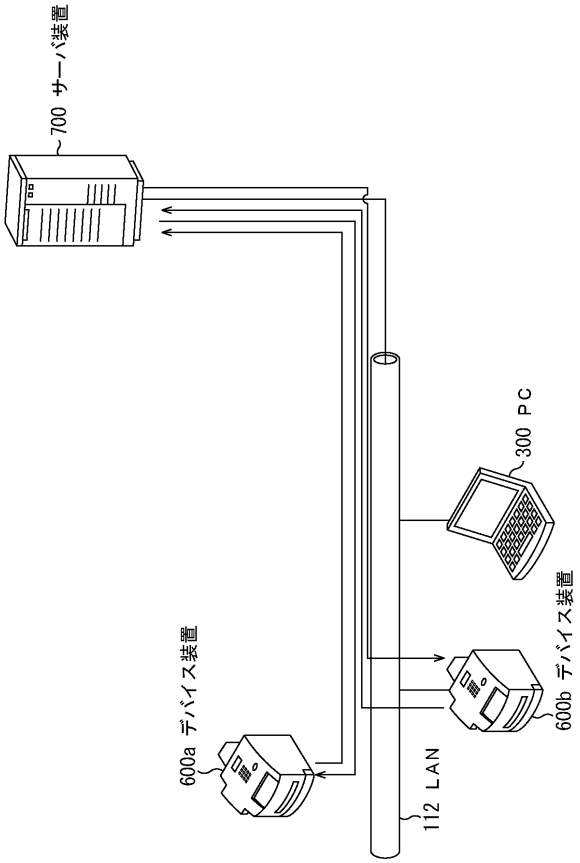
【図４】

401 KEY	402 UPDATE TIME
“SLP:REBOOT”	Immediately
“RDS:COUNTERERROR”	Less than 24 hours
“PCL:PROCESS ERROR”	Immediately
“SCON:SLEEP ERROR”	Before sleeping
“UI:MESSAGE ERROR”	Less than 24 hours
“REPORT:PRINT ERROR”	Before next time

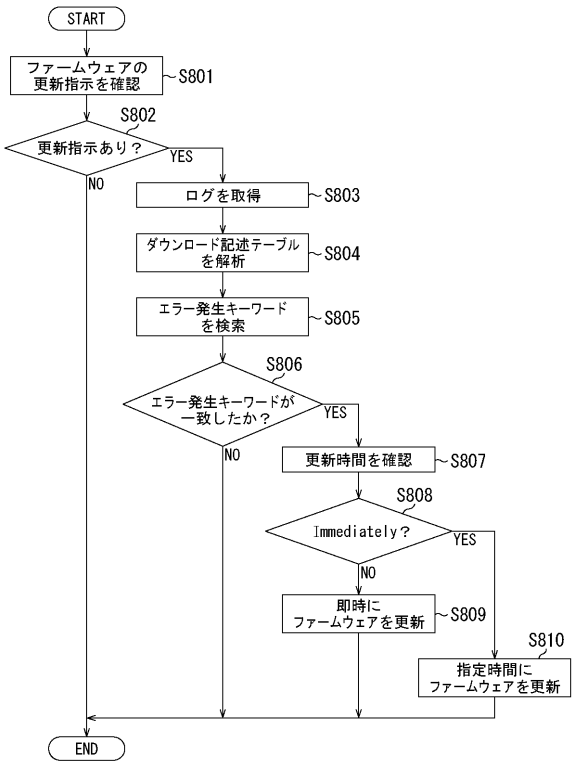
【図５】



【図６】



【図７】

701 KEY	702 UPDATE TIME
“SLP:REBOOT”	Immediately
“RDS:COUNTERERROR”	Less than 24 hours
“PCL:PROCESS ERROR”	Immediately
“UI:MESSAGE ERROR”	Less than 24 hours

【図８】