

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3957825号
(P3957825)

(45) 発行日 平成19年8月15日(2007.8.15)

(24) 登録日 平成19年5月18日(2007.5.18)

(51) Int. Cl.	F I
G06F 3/12 (2006.01)	G06F 3/12 A
B41J 29/38 (2006.01)	G06F 3/12 C
H04N 1/00 (2006.01)	B41J 29/38 Z
	H04N 1/00 I O 7 A

請求項の数 8 (全 70 頁)

(21) 出願番号	特願平9-223597	(73) 特許権者	000001007
(22) 出願日	平成9年8月20日(1997.8.20)		キヤノン株式会社
(65) 公開番号	特開平10-177461		東京都大田区下丸子3丁目30番2号
(43) 公開日	平成10年6月30日(1998.6.30)	(74) 代理人	100090538
審査請求日	平成16年8月19日(2004.8.19)		弁理士 西山 恵三
(31) 優先権主張番号	特願平8-272310	(74) 代理人	100096965
(32) 優先日	平成8年10月15日(1996.10.15)		弁理士 内尾 裕一
(33) 優先権主張国	日本国(JP)	(72) 発明者	福長 耕司
			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
		(72) 発明者	片野 清
			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

最終頁に続く

(54) 【発明の名称】 印字制御システム及びスキャナ制御システム

(57) 【特許請求の範囲】

【請求項1】

ホストコンピュータより受信したコントロールI/Oコマンドを記憶するメモリと、
前記メモリに記憶されたコントロールI/Oコマンドを所定周期で読み出し、読み出したコントロールI/OコマンドをI/Oレジスタに書き込むことにより、印字ユニットのプリントを制御するコントローラとを有し、

前記I/Oレジスタは、前記印刷ユニットを制御するコントローラを構成するレジスタ郡であり、各ユニットを特定するアドレスが規定され、各アドレスに制御データを書き込むことにより各ユニットを制御し、

前記コントロールI/Oコマンドは、I/Oレジスタのアドレスを指示するアドレスとI/Oレジスタに書き込む制御データとを含むことを特徴とするプリンタ。

10

【請求項2】

前記コントロールI/Oコマンドは、コントロールI/Oコマンドの実行を所定時間ウェイトするウェイトコマンドを含むことを特徴とする請求項1記載のプリンタ。

【請求項3】

前記メモリを制御するメモリコントローラを有し、

前記メモリコントローラは、前記コントロールI/Oコマンドの実行を第1優先度とし、リフレッシュの実行を第2優先度とし、ホストコンピュータからのメモリのアクセスを第3優先度とすることを特徴とする請求項1記載のプリンタ。

【請求項4】

20

前記プリンタは、インクジェットプリンタを含むことを特徴とする請求項 1 記載のプリンタ。

【請求項 5】

前記印字ユニットは、印字ヘッド、印字ヘッドを搭載したキャリッジを駆動するモータ、および、紙送りモータを含むことを特徴とする請求項 1 記載のプリンタ。

【請求項 6】

ホストコンピュータより受信したコントロール I / O コマンドを記憶するメモリと、
前記メモリに記憶されたコントロール I / O コマンドを所定周期で読み出し、読み出したコントロール I / O コマンドを I / O レジスタに書き込むことにより、スキャンユニットのスキャンを制御するコントローラとを有し、

10

前記 I / O レジスタは、前記スキャンユニットを制御するコントローラを構成するレジスタ郡であり、各ユニットを特定するアドレスが規定され、各アドレスに制御データを書き込むことにより各ユニットを制御し、

前記コントロール I / O コマンドは、I / O レジスタのアドレスを指示するアドレスと I / O レジスタに書き込む 制御データとを含むことを特徴とするスキャナ。

【請求項 7】

ホストコンピュータより受信したコントロール I / O コマンドをメモリに記憶するステップと、

前記メモリに記憶されたコントロール I / O コマンドを所定周期で読み出し、読み出したコントロール I / O コマンドを I / O レジスタに書き込むことにより、印字ユニットのプリントを制御する制御ステップとを有し、

20

前記 I / O レジスタは、前記印刷ユニットを制御するコントローラを構成するレジスタ郡であり、各ユニットを特定するアドレスが規定され、各アドレスに制御データを書き込むことにより各ユニットを制御し、

前記コントロール I / O コマンドは、I / O レジスタのアドレスを指示するアドレスと I / O レジスタに書き込む 制御データとを含むことを特徴とするプリント方法。

【請求項 8】

ホストコンピュータより受信したコントロール I / O コマンドをメモリに記憶するステップと、

前記メモリに記憶されたコントロール I / O コマンドを所定周期で読み出し、読み出したコントロール I / O コマンドを I / O レジスタに書き込むことにより、スキャンユニットのスキャンを制御する制御ステップとを有し、

30

前記 I / O レジスタは、前記スキャンユニットを制御するコントローラを構成するレジスタ郡であり、各ユニットを特定するアドレスが規定され、各アドレスに制御データを書き込むことにより各ユニットを制御し、

前記コントロール I / O コマンドは、I / O レジスタのアドレスを指示するアドレスと I / O レジスタに書き込む 制御データとを含むことを特徴とするスキャン方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

40

本発明は、パーソナルコンピュータとプリンタとが接続され、ホストがプリンタを制御して印字を行うような印字制御システムに関するものである。

【0002】

【従来の技術】

従来、ホスト（パーソナルコンピュータ）とプリンタとはインターフェース（セントロニクスなど）を介して接続され、ホストから送られる印字データ、印字指示をプリンタが順次処理することにより印字を行っていた。印字データ、印字指示はプリンタ固有のコントロールコードで、プリンタドライバによって作成され、プリンタへと送られる。

【0003】

プリンタではホストから送られてくるコントロールコードを解析し印字データであれば印

50

字データを印字できる分だけ蓄える。印字データのコントロールコードはラスタ形式をしているものがあり、蓄えられた印字データに対してイメージ加工処理や変換処理を行いプリンタに搭載する印字ヘッドで印字可能な形式にイメージデータを再構成し、印字ヘッドにイメージデータを送ることで印字を行っていた。コントロールコードでは給・排紙、フィードなどの各種コントロールコードに対する処理を行っていた。マイクロソフト社のウィンドウズシステムでの印字手順については図 3 1 で説明する。

【 0 0 0 4 】

また、ホスト（パーソナルコンピュータ）とスキャナとはインターフェース（SCSI など）を介して接続され、ホストから送られるスキャンコントロールコードをスキャナが順次処理し、スキャンしたイメージデータをホストへ送信することによりスキャンを行っていた。スキャンコントロールコードはスキャナ固有のコントロールコードで、スキャナアプリケーションによって作成され、スキャナへと送られる。スキャナではホストから送られてくるコントロールコードを解析しスキャン動作とイメージデータの送信とを行っていた。スキャンコントロールコードでは給・排紙、フィードなどの各種コントロールコードに対する処理を行っていた。

【 0 0 0 5 】

【発明が解決しようとする課題】

しかしながら、従来例ではホストはアプリケーションからの印字指示により印刷イメージを作り、印刷イメージからプリンタへと送る印字データと印字指示のコントロールコードを作成していた。また、ホストとプリンタとが完全に独立しているため、プリンタは印字のためにコントロールコードの解析処理、イメージの展開処理、イメージの加工処理などを行うために中央処理装置CPU、処理手順を保持するROM、データを蓄えるRAMなどの多くの資源を有していた。また、プリンタに対してホストの性能は飛躍的に向上しており、従来プリンタで行っていた処理をホストで処理する方が有利な状況も起こってきた。

【 0 0 0 6 】

そこで、本発明の目的は、従来プリンタで処理していた処理をプリンタよりも高性能なホストで行うことにより、従来プリンタが内蔵していた資源を簡素化しコストを低く押さえたプリンタ装置とホストでの印字制御システムを提供することにある。

【 0 0 0 7 】

また、従来例ではホストはスキャンのコントロールコードを作成し送信しているだけであった。また、ホストとスキャナとが完全に独立しているため、スキャナはスキャンのためにコントロールコードの解析処理、スキャン動作制御、イメージデータの加工処理などを行うために中央処理装置CPU、処理手順を保持するROM、データを蓄えるRAMなどの多くの資源を有していた。また、スキャナに対してホストの性能は飛躍的に向上しており、従来スキャナで行っていた処理をホストで処理する方が有利な状況も起こってきた。

【 0 0 0 8 】

そこで、本発明の目的は従来スキャナで処理していた処理をスキャナよりも高性能なホストで行うことにより、従来スキャナが内蔵していた資源を簡素化しコストを低く押さえたスキャナ装置とホストでのスキャナ制御システムを提供することにある。

【 0 0 0 9 】

また、上述したホストベースプリントシステムにおいては、ホストでのプリンタの制御を実現するためにマイクロソフト社のウィンドウズ、ウィンドウズプリンティングシステムやプリンタドライバに変更を加える必要があった。

【 0 0 1 0 】

そこで、本発明の目的は従来プリンタで処理していた処理をプリンタよりも高性能なホストで行うことにより、従来プリンタが内蔵していた資源を簡素化しコストを低く押さえたプリンタ装置とホストでのプリンタ制御に関する方法を提供することにある。また、従来のプリンタドライバからのコントロールコードを入力として利用するため、ウィンドウズ、ウィンドウズプリンティングシステムやプリンタドライバに変更を加える必要が

10

20

30

40

50

ない印字制御システムを提供することにある。

【 0 0 1 1 】

また、従来例のようなシステムでは、ホストはアプリケーションからの印刷指示により印刷イメージを作り、印刷イメージからプリンタへ送る印字データと印字コマンドのコントロールコードを作成する必要があった。

【 0 0 1 2 】

また、プリンタでは印字のためにコマンドの解析処理、イメージの展開処理、イメージの加工処理などを行うために中央処理装置 C P U、処理手順を保持する R O M、データを蓄える R A M などの多くの資源を有していた。

【 0 0 1 3 】

また、プリンタに対してホストの性能は飛躍的に向上しており、従来プリンタで行っていた処理をホストで処理する方が有利な状況も起こってきた。

【 0 0 1 4 】

そこで、本発明の目的は従来プリンタで処理していた処理をプリンタよりも高性能なホストで行うことにより、従来プリンタが内蔵していた資源を簡素化しコストを低く押さえたプリンタとホストでの、プリンタ制御（時間管理）システムを提供することにある。

【 0 0 1 5 】

【課題を解決するための手段】

上記課題を解決するために、本発明のプリンタは、

ホストコンピュータより受信したコントロール I / O コマンドを記憶するメモリと、
前記メモリに記憶されたコントロール I / O コマンドを所定周期で読み出し、読み出したコントロール I / O コマンドを I / O レジスタに書き込むことにより、印字ユニットのプリントを制御するコントローラとを有し、

前記 I / O レジスタは、前記印刷ユニットを制御するコントローラを構成するレジスタ郡であり、各ユニットを特定するアドレスが規定され、各アドレスに制御データを書き込むことにより各ユニットを制御し、

前記コントロール I / O コマンドは、I / O レジスタのアドレスを指示するアドレスと I / O レジスタに書き込む 制御データとを含むことを特徴とする。

【 0 0 1 6 】

また、本発明のスキナは、

ホストコンピュータより受信したコントロール I / O コマンドを記憶するメモリと、
前記メモリに記憶されたコントロール I / O コマンドを所定周期で読み出し、読み出したコントロール I / O コマンドを I / O レジスタに書き込むことにより、スキャンユニットのスキャンを制御するコントローラとを有し、

前記 I / O レジスタは、前記スキャンユニットを制御するコントローラを構成するレジスタ郡であり、各ユニットを特定するアドレスが規定され、各アドレスに制御データを書き込むことにより各ユニットを制御し、

前記コントロール I / O コマンドは、I / O レジスタのアドレスを指示するアドレスと I / O レジスタに書き込む 制御データとを含むことを特徴とする。

【 0 0 1 7 】

また、本発明のプリント方法は、

ホストコンピュータより受信したコントロール I / O コマンドをメモリに記憶するステップと、

前記メモリに記憶されたコントロール I / O コマンドを所定周期で読み出し、読み出したコントロール I / O コマンドを I / O レジスタに書き込むことにより、印字ユニットのプリントを制御する制御ステップとを有し、

前記 I / O レジスタは、前記印刷ユニットを制御するコントローラを構成するレジスタ郡であり、各ユニットを特定するアドレスが規定され、各アドレスに制御データを書き込むことにより各ユニットを制御し、

前記コントロール I / O コマンドは、I / O レジスタのアドレスを指示するアドレスと

10

20

30

40

50

I/Oレジスタに書き込む制御データとを含むことを特徴とする。

【0018】

また、本発明のスキャン方法は、

ホストコンピュータより受信したコントロールI/Oコマンドをメモリに記憶するステップと、

前記メモリに記憶されたコントロールI/Oコマンドを所定周期で読み出し、読み出したコントロールI/OコマンドをI/Oレジスタに書き込むことにより、スキャンユニットのスキャンを制御する制御ステップとを有し、

前記I/Oレジスタは、前記スキャンユニットを制御するコントローラを構成するレジスタ群であり、各ユニットを特定するアドレスが規定され、各アドレスに制御データを書き込むことにより各ユニットを制御し、

10

前記コントロールI/Oコマンドは、I/Oレジスタのアドレスを指示するアドレスとI/Oレジスタに書き込む制御データとを含むことを特徴とする。

【0020】

【発明の実施の形態】

(実施の形態1)

図1は本発明の実施の形態であるホスト部とプリンタ部が一体となった携帯型パーソナルコンピュータ(以下パソコンと略す)を示す斜視図である。

【0021】

パソコン1は、装置本体101、キーボード102、表示部103、及びプリンタユニット2の各部によって構成されている。

20

【0022】

上カバー104は、装置本体101に対して、その後縁の両端に設けられたヒンジ104aを介して回動可能に取り付けられている。

【0023】

これにより本装置の使用時には、上カバー104は、その回動によって表示部103が見やすくなる位置まで開けられ、また不使用時には閉じられていてカバーとして機能することができる。

【0024】

また、デバイスの一例として表示部103の表示素子としては、表示部を薄く構成できることから液晶表示素子(LCD)を用いている。

30

【0025】

プリンタユニット2は、操作者が開閉可能な開口部を持ち、記録ヘッドの交換が可能になっている。

【0026】

記録紙は図示されていないが、キーボード102の下部に設けられた給紙口101aから挿入され、装置本体101内を貫通する搬送路内を搬送されて装置後方の排紙口から排出される。

【0027】

キーボード102は、装置本体101の両側に設けられたヒンジ102aを介して回動可能に取り付けられている。

40

【0028】

キーボード102の下部に記録紙の搬送路が設けられているため、記録紙をセットした状態でもキーボード102や表示部103の操作が可能である。

【0029】

図2はパソコンの内部構成を示す概略ブロック図であり、本発明の実施の形態でのホストコンピュータ部(以下ホスト部と略す)はPC/AT互換機と同等のシステムで構成されている。

【0030】

まず、ホスト部で主制御をつかさどっているのが中央処理装置(CPU)であり、その基

50

本的な制御を指示するのが B I O S - R O M である。

【 0 0 3 1 】

数値演算プロセッサ (F P U) は C P U - b u s に接続され、浮動小数点演算や対数計算などの演算命令を拡張するもので有る。

【 0 0 3 2 】

フロッピーディスクドライブ (F D D) やハードディスクドライブ (H D D) からフロッピーディスクコントローラ (F D C) やハードディスクコントローラ (H D C) を経由してアプリケーションプログラムやデータファイルを読み出し、システムメモリ (S Y S T E M R A M) を利用してアプリケーションプログラムの実行を行なう。

【 0 0 3 3 】

この時の画面の表示は、ビデオコントローラ (V G A) と表示メモリ (V R A M) により液晶表示器 (L C D) にキャラクタやグラフィック等の表示を行ない、キーボード (K B) からのキー入力は、キーボードコントローラ (K B C) を経由して取り込まれる。

【 0 0 3 4 】

リアルタイムクロック (R T C) は、現在時刻を示すタイマである。

【 0 0 3 5 】

D M A コントローラ (D M A C) は、メモリ - メモリ間、メモリ - I / O 間において、C P U の介在なしでデータ転送を行なう。

【 0 0 3 6 】

割り込みコントローラ (I R Q C) は、各 I / O からの割り込みを受け付け、優先順位に従って処理を行なう。

【 0 0 3 7 】

システムタイマ (T I M E R) は、数チャンネルのフリーランニングタイマを持ち、種々の時間管理を行なう。

【 0 0 3 8 】

その他に外部コネクタにつながるポートブロックとして、R S - 2 3 2 C 仕様のシリアルポート (S I O) 、セントロニクス仕様のパラレルポート (P I O) 、I S A 仕様の拡張ポート (E I O) があり、それぞれ S I / F コネクタ、P I / F コネクタ、E I / F コネクタにつながる。

【 0 0 3 9 】

またユーザに対して動作状況を伝える L E D 出力ポートや、操作スイッチ入力ポートがつながる、P A N E L もある。

【 0 0 4 0 】

本発明の実施の形態におけるプリンタ部は、ホスト部に対して I S A - b u s で直接つながれた形態になっていて、I S A - b u s の I / O レジスタを介してデータのやり取りを行うことができる。

【 0 0 4 1 】

図 3 に本発明の実施の形態におけるホスト部 R A M のメモリマップの配置を示す。

【 0 0 4 2 】

R A M は標準領域として 0 0 0 0 0 H ~ B F F F F H の 6 4 0 K B と、拡張領域として 1 0 0 0 0 0 H ~ F D F F F F H の 1 5 M B があり、R A M はこれらの領域に配置されるようにメモリマッピングされる。

【 0 0 4 3 】

R A M の標準領域の先頭部分 0 0 0 0 0 H ~ 0 0 3 F F H には、割り込み用のベクタを保存する領域が有り、この中に割り込みに対する各処理のエントリーアドレスが保存される。

【 0 0 4 4 】

ビデオ R A M 領域、及びビデオ B I O S 領域は、ビデオコントローラの内部に配置され、ビデオ B I O S 領域にはビデオ制御のためのプログラムが保存され、ビデオ R A M 領域にはビデオ表示データが保存される。

10

20

30

40

50

【0045】

拡張ボードROM領域はC8000H～DFFFFHまでの領域に配置され、拡張ポートに繋がるデバイスにより使用される。

【0046】

システムBIOS領域はF0000H～FFFFFFHに配置され、各種I/Oの処理を行うBIOSプログラムを保持している。

【0047】

図4に本発明の実施の形態におけるホスト部のI/Oアドレスマップを示す。

【0048】

ここで示したI/Oアドレスは、ハードウェアで設定されている各デバイスのI/Oアドレスのレジスタを介して、データのやり取り(READ・WRITE)をホストと各デバイスの間で行うものであるが、特にプリンタ部に関連する部分について説明する。

【0049】

パラレルポート1～3(PIO/IO)はホスト部からPI/Fに繋がった外部デバイスに対して汎用のパラレルデータを送出するためのレジスタである。

【0050】

図20に従来のプリンタ制御部のシステム構成を示す。

【0051】

I/Fコントローラは、ホストのパラレルポートに対してケーブルを介して繋がり、ホストから送られる印字動作を行うためのコントロールコードを受け取るブロックである。

【0052】

P-CPUは、プリンタ制御部の中で主制御を司るCPUであり、I/Fコントローラからのコントロールコードの取り込み処理と、プリンタの印字動作を行うための処理を、時間配分しながら制御を行う。

【0053】

P-ROMは、P-CPUの動作プログラムの格納されたROMである。

【0054】

P-PAMは、P-CPUが動作プログラムの実行に際してワークメモリとして使ったり、I/Fコントローラから送られてきたコントロールコードを一時保管する受信メモリとして使ったり、コントロールコードを実際の記録ヘッドに転送するデータの形態に変換したイメージデータを一時保管する印字メモリとして使ったりするRAMである。

【0055】

プリンタI/Oコントローラは、プリンタユニットを動作させるための制御ブロックで、P-CPUがPRT-busに対してプリンタI/Oコントローラに割り当てられた任意アドレスのレジスタに対するアクセス処理を行うことにより、プリンタユニットのキャリアモータやフィードモータの相切り替えや、印字ヘッドに対するイメージデータの転送などを行うブロックである。

【0056】

すなわち、従来のプリンタ制御部による印字処理は、ホストから送られてくる印刷のためのコントロールコードをP-CPUが逐次解釈しながら、キャリアを水平方向に走査して印字ヘッドに対してイメージデータを転送し印字を行っていた。

【0057】

また、従来のプリンタ制御部による画像読み取り処理は、プリンタユニットのキャリア上に印字ヘッドの代わりに画像読み取り用のスキャナヘッドが載った状態で、ホストからのスキャン実行を指示するコントロールコードを受けると、キャリアを走査しながらスキャナヘッドで画像データを読み取り、パラレルインターフェイスのニブルモード等を利用してホストに画像データを逐次転送していた。

【0058】

本発明の実施の形態で用いているプリンタ制御部は、汎用のパラレルポートを用いずにプリンタ制御部の専用ポート(PRT/IO)を用いて印字動作を行うシステムとなってい

10

20

30

40

50

る。

【0059】

PRT / IO のアドレス設定等を行うのが 100H ~ 107H のセットアップレジスタ (SET / IO) で PRT / IO を構成するスーパーバイザコマンドレジスタ (SVA / IO・SVD / IO) と制御 I / O コマンドレジスタ (BAT / IO) のベースアドレスとポートイネーブルの設定を行うものである。

【0060】

図5はセットアップレジスタ (SET / IO) の構成を示す図である。

【0061】

0102番地のセットアップレジスタ2は、PRT / IO のイネーブルビットがビット0に設けられ、イネーブルビットが “ 1 ” の時にのみホスト部からプリンタポートへのアクセスを可能としている。

10

【0062】

0103 ~ 0104番地のセットアップレジスタ3・4は、PRT・IOのベースアドレスを設定するレジスタで、設定されたアドレスから連続した8バイトをホスト部とプリンタ部の間のデータのやり取りをポートに設定するレジスタである。

【0063】

ベースアドレスは0000H ~ 03F8Hまでの範囲内において、8バイト単位で任意に設定することができる。

【0064】

20

図6はプリンタ部ポート (PRT・IO) の構成を示す図である。

【0065】

プリンタ部ポートの構成は、セットアップレジスタ (SET / IO) で設定されたベースアドレスから連続した8バイトのうち、最初の1バイト目がスーパーバイザコマンド・アドレスレジスタ (SVA / IO)、2バイト目がスーパーバイザコマンド・データレジスタ (SVD / IO) でこれらは8ビットアクセスであり、3 ~ 4バイト目が制御 I / O コマンド・データレジスタ (BAT / IO) でこれは16ビットアクセス専用のレジスタとなる。

【0066】

スーパーバイザコマンドはホストからプリンタ制御部の PRT - bus に対して直後アクセスを行うことができるレジスタであり、直接制御 I / O コマンドとも呼ぶ。

30

【0067】

SVA / IO は PRT - bus へのアクセスアドレスを設定するレジスタで、PA [7 : 0] の8ビットを任意に設定するスーパーバイザコマンド・アドレスレジスタである。

【0068】

SVD / IO は PRT - bus へのデータ入出力を行うレジスタで、PD [7 : 0] の8ビットを使って操作するスーパーバイザコマンド・データレジスタである。

【0069】

BAT / IO は MEM - bus へのデータ入出力を行うレジスタで、MD [15 : 0] の16ビットでアクセスする制御 I / O コマンド・データレジスタである。

40

【0070】

図7は本発明の実施の形態におけるプリンタ制御部の内部構成を示すブロック図である。

【0071】

プリンタ部の主要ロジックは、ホスト部の ISA - bus からステートマシンコントローラ (STATE MACHINE CONTROLLER) を経由して、プリンタバス (PRT - bus) に繋がるプリンタ I / O コントローラ (PRINTER I / O CONTROLLER) と、メモリバス (MEM - bus) に繋がるプリンタメモリ (PRINTER MEMORY) によって構成されている。

【0072】

プリンタユニットに繋がるプリンタ I / O コントローラの入力側には、プリンタユニット

50

からのセンサ情報を読み取ったプリンタステータス信号 (P r i n t e r S t a t u s) がとりこまれ、その情報をもとに出力側のモータコントロール信号 (M o t o r C o n t .) やヘッドコントロール信号 (H e a d C o n t .) によってプリンタユニット (P R I N T E R U N I T) のモータやインクヘッドを駆動し印字動作を行う。

【 0 0 7 3 】

またプリンタユニットのキャリアにスキャナヘッドの載ったシステムの場合は、スキャナヘッドで読み出した画像データをプリンタ I / O コントローラを経由してプリンタメモリに書き込むことでスキャン動作を行う。

【 0 0 7 4 】

ここで、図 2 0 に示した従来のプリンタ制御部と構成を対比させると、本発明の実施の形態ではプリンタ側に C P U と R O M が存在しない点が大きな特徴であるといえる。

10

【 0 0 7 5 】

図 8 はステートマシンコントローラの内部構成を示すブロック図である。

【 0 0 7 6 】

ステートマシンコントローラはホスト部の I S A - b u s を経由してプリンタユニットの動作をつかさどるための制御ブロックであり、セットアップレジスタで設定したプリンタ部ポートの各々のレジスタによってアクセスを行うことができる。

【 0 0 7 7 】

スーパーバイザコマンドレジスタは、ホスト部からプリンタ I / O コントローラへの I / O アクセスを P R T - b u s を介して行ったり、制御 I / O コマンドの起動・停止コマンドの発行を行うためのレジスタで、スーパーバイザコマンド・アドレスレジスタ (S V A / I O) で設定した P R T - b u s のアドレス (P A) に対してスーパーバイザコマンド・データレジスタ (S V D / I O) を使ってデータ (P D) の読み書きを行うことを行う。

20

【 0 0 7 8 】

制御 I / O コマンド・データレジスタ (B A T / I O) は、ホスト部からプリンタ部メモリを M E M - b u s を介して直接アクセスするための機能であり、「アドレス + データ」のセットでデータ書き込み動作を行うとメモリアドレスカウンタのライトアドレスに対応したプリンタ部メモリのアドレス (M A) に 2 バイトのデータ (M D) が書き込まれ、書き込まれた後にはライトアドレスが毎回 + 2 インクリメントされる。

30

【 0 0 7 9 】

制御 I / O コマンド・データレジスタを使ったデータの読み出し動作も同様に、リードアドレスに対応したプリンタ部メモリのデータが 2 バイト読み出され、読み出し動作後には毎回リードアドレスが + 2 インクリメントされる。

【 0 0 8 0 】

制御 I / O コマンド・データレジスタによってメモリに書き込まれたデータは制御 I / O コマンドを実行するために使われるデータであり、スーパーバイザコマンドレジスタによる制御 I / O コマンドの起動コマンドとして B P - s t a r t に “ 1 ” を書き込む事により、プリンタ部メモリから制御 I / O アドレスカウンタのリードアドレスで示されたデータを 2 バイトずつ読み出し、読み出したデータの下位バイトがアドレス・上位バイトがデータを表しているの、 P R T - b u s に読み出したアドレスとデータを出力してプリンタ I / O コントローラのレジスタへのデータライトアクセス動作を行うことで制御 I / O コマンドの実行を行う。

40

【 0 0 8 1 】

この制御 I / O コマンドの実行により読み出されたリードアドレスは + 2 ずつ毎回インクリメントされ、制御 I / O コマンドの停止コマンド (B p - e n d) が発行されて B P - s t a r t が “ 0 ” になるまで、制御 I / O コマンド実行動作が繰り返される。

【 0 0 8 2 】

リフレッシュコントロールは、D - R A M のリフレッシュ動作をつかさどる制御ブロックで、8 m s の間に 5 1 2 回のリフレッシュアクセスを行うようになっていて、その時のア

50

ドレスを毎回インクリメントさせるのがリフレッシュアドレスカウンタである。

【0083】

メモリアドレスコントロールは、プリンタ部メモリに対して各アクセス要求に優先順位をつけて処理を行うブロックで、制御I/Oコマンドの実行が第一優先・リフレッシュが第二優先・ホスト部からのメモリアccessが第三優先となっている。

【0084】

これは制御I/Oコマンドの実行時間の変動するのを防ぐための優先順序となっている。

【0085】

図9は、プリンタ部でスーパーバイザコマンドや制御I/Oコマンドを実行する際に必要となるステートマシンコントローラ内のレジスタのアドレスマップである。

10

【0086】

これら00番地から1F番地までのレジスタにおいて、制御I/Oコマンドの実行に必要なスーパーバイザコマンドでのみ使用するレジスタとして00~0F番地・18~1A番地と、制御I/Oコマンド上でのみ使用するレジスタの10~15番地・1C~1E番地がある。

【0087】

また20番地以降についてはプリンタI/Oコントローラに対してスーパーバイザコマンドや制御I/Oコマンドによるアクセスを行うレジスタであるが詳しくは後述する。

【0088】

00番地から1F番地までの各レジスタの機能について個々に説明すると、00番地は制御I/Oコマンドの実行開始を指示する制御I/Oコマンドコントロールレジスタ(BP-start)であり、スーパーバイザコマンドのライトアクセスにより実行される。

20

【0089】

このレジスタは、制御I/Oコマンドが実行中は常に“1”になっているので、“0”であるかどうかをスーパーバイザコマンドにより読み出して確認することで、ホスト側で制御I/Oコマンドの実行が完了したことを検出するのに使うことも可能である。

【0090】

01番地はシステム内のレジスタを全てリセットするリセットトリガ(CRES)であり、スーパーバイザコマンドのライトアクセスにより実行される。

【0091】

08~0A番地はホストからプリンタメモリの読み出しアドレス(MRA)を設定するレジスタであり、スーパーバイザコマンドにより設定を行いホストからのメモリデータの読み出しにより+2インクリメントされる。

30

【0092】

0C~0E番地はホストからのプリンタメモリへの書き込みアドレス(MWA)を設定するレジスタであり、スーパーバイザコマンドにより設定を行いホストからのメモリデータの書き込みにより+2インクリメントされる。

【0093】

10番地は制御I/Oコマンドの実行停止を指示するコマンドエンドトリガ(BP-end)であり、制御I/Oコマンドにより実行される。

40

【0094】

12番地は制御I/Oコマンドによりレジスタデータをメモリに書き込む動作を行うためのレジスタライトトリガ(REGWR)であり、制御I/Oコマンドにより実行される。

【0095】

13番地は制御I/Oコマンドによりレジスタデータをメモリに書き込む際のレジスタアドレス(RWA)を設定するレジスタであり、制御I/Oコマンドにより実行される。

【0096】

14番地はウェイト処理を行うタイマレジスタの下位バイト(WAITL)を設定するレジスタであり、制御I/Oコマンドの実行において0.5μs単位でウェイトを挿入する際に使用するレジスタで、0.5μs(01H)から127.5μs(FFH)まで設定

50

することができる。

【0097】

15番地はウエイト処理を行うタイマレジスタの上位バイト(WAITH)を設定するレジスタであり、制御I/Oコマンドの実行において128μs単位でウエイトを挿入する際に使用するレジスタで、128μs(01H)から32640μs(FFH)まで設定することができる。

【0098】

18~1A番地は制御I/Oコマンドのプリンタメモリからの実行読み出しアドレス(BRA)を設定するレジスタであり、制御I/Oコマンドの実行開始アドレスをスーパーバイザコマンドで設定した後に、制御I/Oコマンドの実行に伴ってメモリデータの読み出しごとに+2インクリメントされる。

10

【0099】

1C~1E番地はレジスタデータのプリンタメモリへの書き込みアドレス(BWA)を設定するレジスタであり、13番地のレジスタライトトリガによるレジスタデータのメモリへの書き込み動作によって+2インクリメントされる。

【0100】

図10は制御I/Oコマンドの動作を示すタイミングチャートである。

【0101】

制御I/Oコマンドの実行サイクルはプログラムカウンタの基本クロック(CLKPRG)に同期していて決められた時間間隔で実行されるが、本発明の実施の形態では500ns単位でひとつの制御I/Oコマンドが実行される様になっている。

20

【0102】

始めに一連の制御I/Oコマンドがプリンタメモリに書き込まれた状態において、起動コマンドとしてスーパーバイザコマンドでBP-startに“1”が書き込まれるとCLKPRGが動作を開始する。

【0103】

1クロック目(BRA:#00)は、MEM-busから読み出した制御I/Oコマンドのアドレスが通常のプリンタI/Oコントローラへアクセス(20番地以降)の場合で、読み出したアドレスに対してデータをPRT-busに書き込む処理を行い、CLKPRGの立ち下がりに同期して制御I/Oコマンドのリードアドレスカウンタ(BRA)が+2インクリメントされる。

30

【0104】

2クロック目(BRA:#02)は、MEM-busから読み出した制御I/OコマンドのアドレスとデータがPRT-busへのアクションではなくメモリ書き込み処理を行うレジスタライトコマンド(12番地)であった場合で、制御I/Oコマンドアドレスカウンタのレジスタライトアドレスに対応したメモリアドレス(MRA)に所望のレジスタのデータを書き込み、その後CLKPRGの立ち下がりに同期して制御I/Oコマンドのリードアドレスカウンタ(BRA)が+2インクリメントされる。

【0105】

3クロック目(BRA:#04)は、MEM-busから読み出した制御I/Oコマンドがコマンドエンドトリガ(BP~end:10番地)であった場合で、PRT~busの10番地に書き込む処理でBP~endが発行されたことをステータスコントローラが検知し、次のCLKPRGの立ち下がりのタイミングでBP-startを“0”にセットしてCLKPRGクロックを停止させ、制御I/Oコマンドのリードアドレスカウンタを+2インクリメントして制御I/Oコマンドの動作実行を完了する。

40

【0106】

図21は、制御I/Oコマンドの中のウエイトコマンドの動作を示すタイミングチャートである。

【0107】

1クロック目(BRA:#00)は、MEM-busから読み出した制御I/Oコマンド

50

がウエイトコマンドである 14 番地に 01H を書き込むコマンドであった場合で、PRT - bus の 14 番地に 01H が書き込まれたところで、CLKPRG の 1 クロック分のウエイトがかかり、その後 500ns 経過後に CLKPRG の立ち下がりに同期して制御 I/O コマンドのリードアドレスカウンタ (BRA) が +2 インクリメントされる。

【0108】

ウエイト後の 2 クロック目 (BRA : #02) は、MEM - bus から読み出した制御 I/O コマンドがコマンドエンドトリガ (BP - end : 10 番地) であった場合で、PRT - bus の 10 番地にデータを書き込む処理で BP - end が発行されたことをステータスコントローラが検知し、次の CLKPRG の立ち下がりのタイミングで BP - start を “0” にセットして CLKPRG クロックを停止させ、制御 I/O コマンドのリードアドレスカウンタを +2 インクリメントして制御 I/O コマンドの動作実行を完了する。

10

【0109】

(プリンタユニットの構成)

図 11 にプリンタユニットの外形図、図 12 に本プリンタのアクチュエータと、そのドライバ回路の構成を示す。

【0110】

図 11 中 1 はプリンタユニット本体を表す。本プリンタユニットのキャリッジ 2 には脱着可能な印字記録ヘッドユニット 3 と、印字記録ヘッドユニット 3 と同形状の画像取り込み用スキャナヘッドユニット 4 が装着可能になっている。印字記録ヘッドユニットは吐出口 128 ドットのモノクロ印字ヘッドユニットと、吐出口 136 ドットのカラー印字ヘッドユニットの 2 種類が装着可能である。

20

【0111】

図 12 で示すように本プリンタは大別して記録ヘッド 121 とその駆動系 122、キャリッジ 123 とその駆動系 124、紙送り機構 125 とその駆動系 126、各種センサー 127 により構成される。さらにこれら駆動系を制御し、プリンタ動作を司るプリンタ I/O コントローラ IC 128 が設けられている。さらに電源回路 129、ユーザーにステータスを知らしめる LED 129、ユーザーによる制御が可能であるスイッチ 1210 が設けられている。

【0112】

図 13 に記録ヘッドおよびヘッドドライバ回路の構成を示す。なお本実施例では印字ヘッドユニットとしてカラー印字ヘッドユニットの説明を行うが、モノクロ印字ヘッドユニットの動作原理も基本的には同様である。

30

【0113】

ここで本例では吐出ユニットは 136 個の吐出口を有するものとし、#1 ~ #136 は吐出ユニットに設けられた吐出口の位置に対応した番号を示すものとする。

【0114】

R1 ~ R136 はそれぞれ #1 ~ #136 の吐出口に対応して設けられた吐出エネルギー発生素子としての発熱抵抗体である。発熱抵抗体 R1 ~ R136 は 16 個を 1 単位としたブロックに分割され、プリンタ I/O コントローラ IC から出力される 4 ビットのイネーブル信号 EENB、BENB1 ~ 3 (以下ブロックイネーブル信号) をデコードすることにより生成される各ブロックのコモンイネーブル信号により、通電経路のオン・オフが制御される。

40

【0115】

また、各発熱抵抗体にはプリンタ I/O コントローラ IC から本記録ヘッドに信号線 HDATA を介し、転送クロック HCLK に同期してシリアル転送され、シリアル・パラレル変換後ヘッド内のデータラッチにて所定のタイミングでラッチされる吐出データに応じて、データに対応した位置の抵抗体の通電経路をオン・オフ制御するセグメントイネーブル信号が接続されている。発熱抵抗体はブロックイネーブル信号とセグメントイネーブル信号の両方がイネーブル状態になっている場合に通電経路がオン、すなわち発熱状態となる。

50

【 0 1 1 6 】

最後に、選択された抵抗体を発熱するパルスを送るための制御信号 H E N B (ヒートイネーブル) が各発熱抵抗体に接続されている。これによって定められたヒートパルスに従って発熱抵抗体 R 1 ~ R 1 3 6 の通電、発熱が行われる。

【 0 1 1 7 】

図 1 4 は上記構成によるヘッド駆動のタイミングチャートを示す。ヘッド走査方向上のある位置においてブロックイネーブル信号の入力に応じてコモンイネーブル信号が順次オンされる。そのオンにより一つのブロックが選択されて通電状態になるので選択されたブロック内において転送された画像データに応じてセグメント信号をそれぞれイネーブル・デイスエーブル状態にすることにより選択された発熱抵抗体がヒートイネーブル信号で与えられたパルスタイミングで通電され、発熱に応じてインクが吐出されドット記録が行われる。

10

【 0 1 1 8 】

図 1 5 はヘッドを搭載するキャリッジの走査動作を行うキャリッジモーター並びにモータードライバの構成図である。また図 1 6 はその駆動タイミングを示す。

【 0 1 1 9 】

本例においてはキャリッジモーターとしてコイル 1 ~ 4 を有するステッピングモーターを用い、駆動信号 C M a 、 C M a _、 C M b 、 C M b _と電流制御信号 P W M A 、 P W M B を備えたモータードライバにより駆動している。また、紙送り機能を構成するラインフィードモーターについても同様な構成で駆動信号 L F a 、 L F a _ L F b 、 L F b _により駆動される。

20

【 0 1 2 0 】

本例ではセンサー機能として、キャリッジが走査範囲左端の所定場所に位置したことを検出するホームポジションセンサ、記録対象物である紙を検出するペーパーセンサが設けられており、これらセンサ情報に基づいたプリンタ制御が行われる。またシステムの温度情報、モーターの駆動電圧が後述する A / D 変換機能を介し、検出可能となっている。

【 0 1 2 1 】

本プリンタにはユーザーによって操作可能であるプリントストップ・再スタートスイッチと一回の押下により定められたライン数の紙送りを実行するラインフィードスイッチが設けられている。

30

【 0 1 2 2 】

本プリンタの電源部は、プリンタの各アクチュエータへの電源供給を行う。プリンタ I O コントローラ I C の制御により、モーター駆動用電源のほかに、ヘッド駆動用のロジック電源とインク吐出用発熱抵抗体の駆動電圧 (以下ヘッドドライブ電源) のオン・オフ制御を行う。

【 0 1 2 3 】

(プリンタ I O コントローラ I C の構成)

プリンタ I O コントローラ I C のブロック図を図 1 7 に示す。本 I C は与えられた制御制御 I / O コマンドに基づいたプリンタ制御を行う。本コントローラ I C は図中に示す様に、モーター制御ブロック 1 7 1、記録ヘッド制御ブロック 1 7 2、データ転送ブロック 1 7 3、センサステータスブロック 1 7 4、パワー・I O ポートブロック 1 7 5 と、制御 I / O コマンド実行タイミング制御回路 1 7 6 により構成される。また本 I C はステートマシンコントローラと、プリンタバスと称するアドレスバス P A { 7 : 0 } ・データバス P D { 7 : 0 } とタイミング制御信号で接続されている。各制御ブロックは各々後述する機能を持っており、基本的には制御 I / O コマンド格納用入力レジスタ群と制御回路により構成される。図 1 8 に本コントローラ I C のレジスタマップを示す。以下の説明では各レジスタ名は表中の該当するアドレスを使って説明する。

40

【 0 1 2 4 】

(制御 I / O コマンド実行タイミング制御回路)

プリンタ I O コントローラ I C に接続されるプリンタバス P A { 7 : 0 }、P D { 7 : 0

50

｝は制御 I / O コマンド実行タイミング回路に接続される。プリンタバスは本回路を経由して各制御ブロックのレジスタ群にレジスタアドレスバス I P A { 7 : 0 }、レジスタデータバス I P D { 7 : 0 }として接続される。

【 0 1 2 5 】

本回路はステートマシンコントロール I C でレジスタアドレスとレジスタデータに分割された制御 I / O コマンドがプリンタバスを介して送られてくる際、本回路内でレジスタアドレス、データを保持し、プリンタコントロール部内のアドレスで指定されたレジスタに格納するタイミング、すなわち制御 I / O コマンド実行開始タイミングを一定にすることにより連続した制御 I / O コマンド処理の中で各制御 I / O コマンドの読み出しから実行までの制御 I / O コマンド実行時間を統一する機能を制御する回路である。具体的には、プリンタバスのデータは保持手段として設けられた制御 I / O コマンドラッチによりラッチされ、コマンド実行基準クロック C K P R G の立ち上がりエッジに同期したタイミングで該当レジスタにデータが格納される。このタイミングでレジスタアドレスバス、レジスタデータバスにラッチしたデータを、書き込み用制御信号 ~ I W R には書き込み信号を出力する。本回路の構成図を図 2 2 に、タイミング図を図 1 9 に表す。図中 P A、P D バスを介してレジスタ 6 7、3 1 にデータが異なるタイミングで書き込まれているが、該当レジスタに書き込まれるタイミング信号 ~ I W R は常に C K P R G に同期したタイミングで出力される。

10

【 0 1 2 6 】

(キャリッジモータ (C M) ・ラインフィードモータ (L F) モータ制御ブロック)
C M、L F であるコイル 1 ~ 4 を有するステッピングモータをモータドライバを介して 4 相 2 磁駆動するための駆動相信号各々 4 本と電流制御を行うための P W M 出力信号各々 2 本を制御するブロックである。相シフト出力パターンをレジスタ 6 6、7 6 でセットした後、レジスタ 6 7、7 7 にアクセスすることによるシフトトリガによって相信号の出力制御をおこなう。また、レジスタ 6 4、6 5、7 4、7 5 に P W M の設定値を格納し、レジスタ 6 7、7 7 にアクセスすることによるシフトトリガによって設定値に基づく P W M 出力を行う。

20

【 0 1 2 7 】

(記録ヘッド制御 (ヒート・スキャン) ブロック)

本プリンタにプリントヘッド、またはスキャナヘッドが装着されている場合、その記録ヘッドの吐出制御を行うブロックである。

30

【 0 1 2 8 】

レジスタ 3 4 ~ 3 8 はヒートモード設定用レジスタであり、ヘッドのリセット、プリント・スキャンの動作切り替えなどの制御、ヒート動作を行う際の各種オプションを設定するレジスタである。またレジスタ 3 A ~ 3 F はスキャンモード設定用レジスタであり、スキャン動作を行う際のオプション、スキャンヘッドへの制御 I / O コマンド送出手の制御を設定するレジスタである。

【 0 1 2 9 】

レジスタ 4 0 ~ 4 B はヒートイネーブル信号の出力であるヒートパルスのパルスタイミングを設定するレジスタ群である。スキャン動作時のタイミング設定には上記パルスタイミング設定用レジスタの一部を兼用する。

40

【 0 1 3 0 】

レジスタ 3 1 へのデータ 0 1 H 書き込みによるヒート・スキャントリガ制御 I / O コマンドが認識されると、記録ヘッドの 1 カラム分の吐出制御が開始される。上記レジスタの設定値に基づいたヒート・スキャン制御が行われる。

【 0 1 3 1 】

(データ転送ブロック)

本ブロックは記録ヘッド制御時に所定データをパラレル・シリアル変換し、コントローラからヘッドに向けて吐出データとしてシリアル転送クロックに同期して送出する機能を持つ。8 ビットレジスタ 3 2 に吐出データが 1 バイト格納されると、転送動作が行われる。

50

【 0 1 3 2 】

また、レジスタ 3 4 の設定により、スキャナ制御モードが指定されている場合には、スキャナユニットから転送クロックとともにシリアル送出されてくるスキャン画像データをシリアル・パラレル変換の後、16ビットデータとしてレジスタ 3 A、3 B に格納する。

【 0 1 3 3 】

(センサステータス ブロック)

プリンタドライブ回路に設けられているセンサのレベルをレジスタ情報として読みだし可能とするブロックである。この情報はレジスタ 2 4 に格納される。また、センサの内アナログ出力入力のもので A / D 入力に接続されている信号はレジスタ 2 7 の A / D 変換トリガビットをアクセスすることにより、本ブロックで A / D 変換され、デジタルデータとしてレジスタ 2 C に格納される。これらセンサー情報をフィードバックすることにより、プリンタの状況に応じた制御が可能となる。

10

【 0 1 3 4 】

(パワー・ＩＯポートブロック)

プリンタ駆動用の電源を制御するブロックである。具体的にはモーター駆動用電源のほかに、ヘッド駆動用のロジック電源とインク吐出用発熱抵抗体の駆動電圧（以下ヘッドドライブ電源）の制御をレジスタ 2 6 の設定により行う。また、プリンタスタート・ストップ用、ラインフィード用スイッチ、ステータス表示用 L E D などの入出力ポートの制御もレジスタ 2 0、2 1、2 8、2 9 を使用して行われる。

【 0 1 3 5 】

図 2 3 ~ 図 2 9 で、プリンタ駆動のための制御 I / O コマンド構成について説明する。

20

【 0 1 3 6 】

この実施例では、キャリッジモーター、ラインフィードモーターともバイポーラ型ステッピングモーターで、かつ駆動制御方式は、定電流チョッパー方式の P W M 変調によるマイクロステップ駆動である（一相あたり 6 分割して駆動）。

【 0 1 3 7 】

キャリッジ初期化動作時のキャリッジモーターの回転数は、ステッピングモーターの自起動領域（300 p . p . s）であるので、等速駆動をさせる。

【 0 1 3 8 】

また、印字動作とキャリッジ復帰動作時のおのおののモーター回転数（766 p . p . s）は、ステッピングモーターの自起動領域より高速であるので、印字（キャリッジ復帰）速度にするため、相切り替え時間を漸次小さくしていき（加速）、等速区間は、加速の最後の相切り替え時間で駆動し、また、停止するまで相切り替え時間を漸次大きくしていく（減速）必要があるので、大きく分けて、加速動作部分と等速動作部分、減速動作部分の3つの部分から構成される。

30

【 0 1 3 9 】

図 2 3 に実際のプリンタとしてのキャリッジ初期化動作（ホームポジション検出動作）のフローチャートを示す。本発明の実施の形態のようなシステムでは、従来のプリンタのようにキャリッジモーターなどのアクチュエーターを動作させながらそれにつれて変化するセンサの状態を監視し、状態変化によって、動作中のアクチュエーターに対する制御 I / O コマンドを変更することができないため、次のような構成をとる。すなわち、図の S 1 でスーパーバイザーコマンドを用いて H P センサの状態を読み込む。S 2 で読み込んだ値から H P センサの状態を判断し、O N であれば、前回の初期化が正常に完了（初期化済み位置）しているとして、S 3 に進み、キャリッジを方向にセンサ情報を毎パルスごとにバッファにダンプしながら右方向に（キャリッジ初期化済み位置から印字方向へ向いて）120 パルス移動させる。S 4 で再びスーパーバイザーコマンドを用いてキャリッジ停止後の H P センサの状態を読み込む。S 5 で、S 4 で読み込んだ値から H P センサの状態を判断し、O F F であれば、S 6 に進んで毎パルスごとにバッファにダンプしたセンサ情報から H P センサが O N から O F F へ切り替わった位置を判断し、切り替わったと判断した位置から O N 方向に 5 4 パルスの地点を初期化済み位置として、S 7

40

50

で左方向（印字位置からキャリッジ初期化済み位置へ向いて）に移動して初期化終了とする。S 5でS 4で読み込んだ値からHPセンサーの状態を判断し、ONであれば、前回の初期化が正常に完了しておらず、初期化済み位置より左側にいた可能性があるので、S 8に進んで、キャリッジをさらに右方向へ120パルス（センサー情報を毎パルスごとにバッファにダンプしながら）移動させる。S 9で再びスーパーバイザーコマンドを用いてキャリッジ停止後のHPセンサーの状態を読み込む。S 10でS 9で読み込んだ値からHPセンサーの状態を判断し、OFFであれば、S 6、S 7と進んで前述のように初期化終了とする。まだONであれば、センサーが壊れているか、キャリッジが物理的に動作していない可能性が高いので、S 11に進んでエラーとする。

【0140】

S 2でHPセンサーの状態がOFFの場合、前回の初期化が正常に完了していないで、印字位置近傍にいるとして、S 12に進み、キャリッジをセンサー情報を毎パルスごとにバッファにダンプしながら左方向（印字位置からキャリッジ初期化済み位置へ向いて）10パルス移動させる。S 13で、スーパーバイザーコマンドを用いてキャリッジ停止後のHPセンサーの状態を読み込む。S 14でS 13で読み込んだ値からHPセンサーの状態を判断し、OFFであれば、S 15に進んで毎パルスごとにバッファにダンプしたセンサー情報からHPセンサーがOFFからONへ切り替わった位置を判断し、そこからON方向にパルスの地点を初期化済み位置として、S 18で初期化位置（HPセンサーがOFFからONへ切り替わった位置からON方向に54パルスの地点）になるように、足りない分を左方向へ移動する。

【0141】

S 14でHPセンサーの状態を判断し、ONであれば、S 17に進んで何回目の左スキャンかチェックし、48回以下なら再びS 12に戻って、前述の動作を繰り返す。S 17で回数が48回を越えていたら、このプリンタの場合、右側壁から左側壁まで480パルス分の長さなので、センサーが壊れているか、キャリッジが物理的に動作していない可能性があるので、S 18に進んでエラーとする。

【0142】

図24は、前述の初期化動作中で、実際にキャリッジを動作させる際の、キャリッジモーター駆動制御I/Oコマンド群の一例を示す。制御I/Oコマンドは、初期設定の行から始まり、左から右に、また上から下の行へと進む（6B01, 6101, 6901, 6401, 6528, 1200, 1508, 1455, 6406...）。

【0143】

各制御I/Oコマンドは、アドレス・データの構成になっており、2バイトをひとかたまりとして扱い、16進数で表記する（例えば、制御I/Oコマンド6B01は、アドレス6Bhにデータ01hの書き込み）。

【0144】

キャリッジ初期化（ホームポジション検出動作）時の動作速度（相切り替え時間）を300p.p.s.とすると、1相あたりの切り替え時間は3.3333msec = 3333usec、1/6相あたりの切り替え時間では555usecになる。本実施例でのウエイトタイマー制御I/Oコマンドの基本単位が、0.5usと64usとすると、64usec × 8 + 0.5 = 512.5、0.5usec × 85 + 0.5 = 43usec、512.5 + 43usec = 555.5usecよって、ウエイト制御I/Oコマンドは、各々WAITH（64usec単位）= 1508H、WAITL（0.5usec単位）= 1455Hとなる。

【0145】

実際は、ウエイト制御I/Oコマンドの他の各々の制御I/Oコマンドを処理するのにそれぞれ約0.5usecの時間が必要になるため、正しく時間管理するためには、ウエイトタイマー制御I/Oコマンド間の他の制御I/Oコマンド数の分も計算にに入れてウエイトタイマー制御I/Oコマンドの値を設定する必要がある。

【0146】

10

20

30

40

50

ここでの説明では、便宜上ウエイトタイマー制御 I / O コマンド自身の処理時間以外は考慮にいていない。実際には、制御 I / O コマンド群を発行する時にその分も考慮して再計算する。

【 0 1 4 7 】

等速部分は、基本的には 2 行目 ~ 1 3 行目までを必要パルス数分だけ繰り返すことにより実現される。2 行目 ~ 1 3 行目までを一周期にしているのは、P W M 変調によるマイクロステップ駆動の電流変化テーブルの周期がモーターの相 2 パルス分で一周期になるからである。

【 0 1 4 8 】

制御 I / O コマンド個々の内容について説明すると、一行目の初期設定は、動作前のキャリッジモーターの回転方向の設定「ここでは右方向キャリッジ移動の例として 0 1 (左方向なら 0 0) 」(6 B 0 1)、バッファヘダンプする中身「センサーポート (2 4 H) の内容」を指示するレジスタライトアドレスの設定制御 I / O コマンド (1 3 2 4)、キャリッジモーター P W M 電流 O N 制御 I / O コマンド (6 1 0 1) である。

【 0 1 4 9 】

二行目は、最初の相シフト (スタート相の次の相励磁) 制御 I / O コマンド (6 9 0 1)、キャリッジモーター P W M 電流 (A 相) 書き込み制御 I / O コマンド (6 4 0 1)、キャリッジモーター P W M 電流 (B 相) 書き込み制御 I / O コマンド (6 5 2 8)、センサー情報ダンプのためのレジスタライトトリガー制御 I / O コマンド (1 2 0 0)、動作速度 3 0 0 p . p . s . (1 / 6 相あたりの切り替え時間 5 5 5 u s e c) を実現するためのウエイトタイマー制御 I / O コマンド H (6 4 u s e c 単位) (1 5 0 8)、およびウエイトタイマー制御 I / O コマンド L (0 . 5 u s e c 単位) (1 4 5 5)、である。

【 0 1 5 0 】

三行目はスタート相から 1 / 6 相分だけ進んだ位置でのキャリッジモーター P W M 電流 (A 相) 書き込み制御 I / O コマンド (6 4 0 6)、キャリッジモーター P W M 電流 (B 相) 書き込み制御 I / O コマンド (6 5 2 6)、センサー情報ダンプのためのレジスタライトトリガー制御 I / O コマンド (1 2 0 0)、動作速度 3 0 0 p . p . s . (1 / 6 相あたりの切り替え時間 5 5 5 u s e c) を実現するためのウエイトタイマー制御 I / O コマンド H (6 4 u s e c 単位) (1 5 0 8)、およびウエイトタイマー制御 I / O コマンド L (0 . 5 u s e c 単位) (1 4 5 5)、である。

【 0 1 5 1 】

同様に、四行目はスタート相から 2 / 6 相分進んだ位置、五行目はスタート相から 3 / 6 相分進んだ位置、六行目はスタート相から 4 / 6 相分進んだ位置、七行目はスタート相から 5 / 6 相分進んだ位置、の制御 I / O コマンドである。

【 0 1 5 2 】

基本的に三行目から七行目は、処理の内容は一緒で、キャリッジモーター P W M 電流の A 相 / B 相に書き込む電流値だけが、マイクロステップ実現のために、正弦関数的に変化している。実際には電流の流れは、A 相 / B 相とも方向性があるが、ここではその電流量の絶対値だけ書き込み、方向については、ドライバ側で変化させるので負の値はとらない。八行目はスタート相から 6 / 6 相分進んだ位置すなわちスタート相の次の相なので、再び相シフト (スタート相の次の次の相励磁) (6 9 0 1)、キャリッジモーター P W M 電流 (A 相) 書き込み (6 4 0 1)、キャリッジモーター P W M 電流 (B 相) 書き込み (6 5 2 8)、センサー情報ダンプのためのレジスタライトトリガー (1 2 0 0)、動作速度 3 0 0 p . p . s . (1 / 6 相あたりの切り替え時間 5 5 5 u s e c) を実現するためのウエイトタイマー制御 I / O コマンド H (6 4 u s e c 単位) (1 5 0 8)、およびウエイトタイマー制御 I / O コマンド L (0 . 5 u s e c 単位) (1 4 5 5)、である。

【 0 1 5 3 】

図 2 5 ~ 図 2 9 は、実際に片方向印字動作 (キャリッジ復帰動作)、紙送り (ラインフィード) 動作とを逐次行う際の、制御 I / O コマンド群の一例を示す。

【 0 1 5 4 】

10

20

30

40

50

図25～27は、片方向印字動作（キャリッジ復帰動作）時の制御I/Oコマンド群を表す。印字動作時のキャリッジモーター回転数は、813 p.p.sであり、通常のステッピングモーターの自起動領域より高速であるので、加減速動作が必要となり、大きく分けて、加速動作と等速動作部分（印字動作を含む）、減速動作部分の3つの部分から構成される。

【0155】

図25は、片方向印字動作（キャリッジ復帰動作）時の加速部分である。制御I/Oコマンドは、初期設定の行から始まり、左から右に、また上から下の行へと進む（6B01, 6101, 6401, 6528, 6528, 159C, 1400, 6901...）。

【0156】

一行目の初期設定は、動作前のキャリッジモーターの回転方向の設定「ここでは右方向キャリッジ移動（および印字）の例として01（左方向なら00）」（6B01）、キャリッジモーターPWM電流ON制御I/Oコマンド（6101）である。

【0157】

二行目はスタート相の励磁のためのキャリッジモーターPWM電流（A相）書き込み制御I/Oコマンド（6401）、キャリッジモーターPWM電流（B相）書き込み制御I/Oコマンド（6528）、スタート相の励磁（保持）時間設定のためのウェイトタイマー制御I/OコマンドH（64usec単位）（159C）、およびウェイトタイマー制御I/OコマンドL（0.5usec単位）（1400）、である。

【0158】

三行目は、最初の相シフト（スタート相の次の相励磁）制御I/Oコマンド（6901）、キャリッジモーターPWM電流（A相）書き込み制御I/Oコマンド（6401）、キャリッジモーターPWM電流（B相）書き込み制御I/Oコマンド（6528）、加速動作を実現するためのウェイトタイマー制御I/OコマンドH（64usec単位）（1519）、およびウェイトタイマー制御I/OコマンドL（0.5usec単位）（14D3）、である。

【0159】

四行目はスタート相から1/6相分だけ進んだ位置でのキャリッジモーターPWM電流（A相）書き込み制御I/Oコマンド（6406）、キャリッジモーターPWM電流（B相）書き込み制御I/Oコマンド（6526）、加速動作を実現するためのウェイトタイマー制御I/OコマンドH（64usec単位）（1518）、およびウェイトタイマー制御I/OコマンドL（0.5usec単位）（14B3）、である。

【0160】

同様に、五行目はスタート相から2/6相分進んだ位置、六行目はスタート相から3/6相分進んだ位置、七行目はスタート相から4/6相分進んだ位置、八行目はスタート相から5/6相分進んだ位置、の制御I/Oコマンドである。基本的に四行目から八行目は、処理の内容は一緒で、キャリッジモーターPWM電流のA相/B相に書き込む電流値が、マイクロステップ実現のために、正弦関数的に変化し、また、ウェイトタイマー制御I/Oコマンドの設定値が、加速動作実現のために少しずつ小さくなっている。

【0161】

通常の加速動作では、相切り替えのタイミングで徐々に時間間隔をせばめていくのであるが、本実施例の場合、マイクロステップ制御を行っているため、同じ相内でも、徐々に時間を変化させていく必要がある（加速1-1～加速1-6の間でも変化させている）。

【0162】

また、実際には電流の流れは、A相/B相とも方向性があるが、ここではその電流量の絶対値だけ書き込み、方向については、モータードライバ側で変化させるので負の値はとらない。

【0163】

九行目はスタート相から6/6相分進んだ位置すなわちスタート相の次の相なので、再び相シフト（スタート相の次の次の相励磁）（6901）、キャリッジモーターPWM電流

10

20

30

40

50

(A相)書き込み(6428)、キャリッジモーターPWM電流の(B相)書き込み(6501)、加速動作を実現するためのウェイトタイマー制御I/OコマンドH(64usec単位)(150F)、およびウェイトタイマー制御I/OコマンドL(0.5usec単位)(14ED)、である。

【0164】

以下、同様に加速動作を本実施例では22ステップ分繰り返す。加速が終了すると、次は等速部分である。

【0165】

図26は、片方向印字動作時の等速部分である。

【0166】

一行目は、最初の相シフト(等速動作スタート相の次の相励磁)制御I/Oコマンド(6901)、キャリッジモーターPWM電流(A相)書き込み制御I/Oコマンド(6401)、キャリッジモーターPWM電流(B相)書き込み制御I/Oコマンド(6528)、等速度813p.p.s(1/6相あたりの切り替え時間205usec)を実現するためのウェイトタイマー制御I/OコマンドH(64usec単位)(1508)、およびウェイトタイマー制御I/OコマンドL(0.5usec単位)(1433)、である。

【0167】

二行目は等速動作スタート相から1/6相分だけ進んだ位置でのキャリッジモーターPWM電流(A相)書き込み制御I/Oコマンド(6406)、キャリッジモーターPWM電流(B相)書き込み制御I/Oコマンド(6526)、等速度813p.p.s(1/6相あたりの切り替え時間205usec)を実現するためのウェイトタイマー制御I/OコマンドH(64usec単位)を実現するためのウェイトタイマー制御I/OコマンドH(64usec単位)(1508)、およびウェイトタイマー制御I/OコマンドL(0.5usec単位)(1433)、である。

【0168】

同様に、三行目はスタート相から2/6相分進んだ位置、四行目はスタート相から3/6相分進んだ位置、五行目はスタート相から4/6相分進んだ位置、六行目はスタート相から5/6相分進んだ位置、の制御I/Oコマンドである。

【0169】

基本的に二行目から六行目は、処理の内容は一緒で、キャリッジモーターPWM電流のA相/B相に書き込む電流値だけが、マイクロステップ実現のために、正弦関数的に変化している。実際には相を流れる電流は、A相/B相とも方向性があるが、ここではその電流量の絶対値だけ書き込み、方向については、ドライバー側で変化させるので負の値はとらない。

【0170】

七行目はスタート相から6/6相分進んだ位置、すなわち等速動作スタート相の次の相の位置にいたので、再び相シフト制御I/Oコマンド(等速動作スタート相の次の次の相励磁)(6901)、キャリッジモーターPWM電流の(A相)書き込み制御I/Oコマンド(6428)、キャリッジモーターPWM電流の(B相)書き込み制御I/Oコマンド(6501)、等速動作速度813p.p.s.(1/6相あたりの切り替え時間205usec)を実現するためのウェイトタイマー制御I/OコマンドH(64usec単位)(1508)、およびウェイトタイマー制御I/OコマンドL(0.5usec単位)(1433)、である。

【0171】

等速部分は、基本的には1行目~12行目までを必要パルス分だけ繰り返すことにより実現される。1行目~12行目までを一周期にしているのは、PWM変調によるマイクロステップ駆動の電流変化テーブルの周期がモーターの相2パルス分で一周期になるからである。

【0172】

10

20

30

40

50

本発明の実施の形態では、キャリッジモーター１パルス（１相分）で、１／６０インチキャリッジが移動する。１相をマイクロステップ制御するために６分割しているので、１マイクロステップあたり $1/60 \text{ インチ} \div 6 = 1/360 \text{ インチ}$ の移動に相当する。

【０１７３】

印字解像度が 360 d.p.i 相当の記録ヘッドでは、各マイクロステップごとに、１カラムの印字に相当するので、実際に 360 d.p.i 相当の印字を行う場合には、図 26 の各行（マイクロステップ）中に１カラム分の印字データ転送制御 I/O コマンドとヒートスキャントリガー制御 I/O コマンド等がさらに付加されるが、これについては後述する。

【０１７４】

等速（印字動作）が終了すると、次は減速部分である。

【０１７５】

図 27 は、片方向印字動作時の減速部分である。制御 I/O コマンドは、第一行から始まり、左から右に、また上から下の行へと進む（6901, 6401, 6528, 1503, 1433, 6406...）。

【０１７６】

一行目は、最初の相シフト（減速スタート相の次の相励磁）制御 I/O コマンド（6901）、キャリッジモーター PWM 電流（A 相）書き込み制御 I/O コマンド（6401）、キャリッジモーター PWM 電流（B 相）書き込み制御 I/O コマンド（6528）、減速動作を実現するためのウェイトタイマー制御 I/O コマンド H（64 usec 単位）（1503）、およびウェイトタイマー制御 I/O コマンド L（0.5 usec 単位）（1433）、である。

【０１７７】

二行目はスタート相から 1/6 相分だけ進んだ位置でのキャリッジモーター PWM 電流（A 相）書き込み制御 I/O コマンド（6406）、キャリッジモーター PWM 電流（B 相）書き込み制御 I/O コマンド（6526）、減速動作を実現するためのウェイトタイマー制御 I/O コマンド H（64 usec 単位）（1503）、およびウェイトタイマー制御 I/O コマンド L（0.5 usec 単位）（1433）、である。

【０１７８】

同様に、三行目はスタート相から 2/6 相分進んだ位置、四行目はスタート相から 3/6 相分進んだ位置、五行目はスタート相から 4/6 相分進んだ位置、六行目はスタート相から 5/6 相分進んだ位置、の制御 I/O コマンドである。基本的に二行目から六行目は、処理の内容は一緒で、キャリッジモーター PWM 電流の A 相 / B 相に書き込む電流値が、マイクロステップ実現のために、正弦関数的に変化し、また、ウェイトタイマー制御 I/O コマンドの設定は減速動作実現のために少しずつ大きくなるが、減速 1～3 パルスあたりではまだそれほど変化していない。

【０１７９】

通常の減速動作では、相切り替えのタイミングで徐々に時間間隔広げていくのであるが、本発明の実施の形態の場合、マイクロステップ制御を行っているため、同じ相内でも、徐々に時間を変化させていく必要がある（例えば、減速 3 - 1～減速 3 - 6の間でも変化させている）。

【０１８０】

また、実際には電流の流れは、A 相 / B 相とも方向性があるが、ここではその電流量の絶対値だけ書き込み、方向については、モータードライバー側で変化させるので負の値はとらない。

【０１８１】

七行目は減速スタート相から 6/6 相分進んだ位置、すなわち減速スタート相の次の相なので、再び相シフト（減速スタート相の次の次の相励磁）（6901）、キャリッジモーター PWM 電流（A 相）書き込み（6428）、キャリッジモーター PWM 電流（B 相）書き込み（6501）、加速動作を実現するためのウェイトタイマー制御 I/O コマンド

10

20

30

40

50

H (6 4 u s e c 単位) (1 5 0 3)、およびウェイトタイマー制御 I / O コマンド L (0 . 5 u s e c 単位) (1 4 3 3)、である。

【 0 1 8 2 】

以下、同様に減速動作を本実施例では 2 2 ステップ分繰り返す。減速が終了すると、減速最後の相 (停止相) を励磁し、しかるのちに、最後の行で、電流を切る。

【 0 1 8 3 】

キャリッジ復帰動作は、基本的には、前述の加速、等速、減速という制御 I / O コマンドと同じであるが、加速の初期設定で、回転方向が異なるのと、(6 B 0 1 - > 6 b 0 0)、等速部分に印字データ、ヒートトリガーとかの制御 I / O コマンドが付加されないという違いがある。

10

【 0 1 8 4 】

図 2 8 ~ 2 9 は、紙送り動作 (ここでは 2 4 パルスの送り) の加速、減速の制御 I / O コマンド群を表す。2 4 パルス送り時のラインフィードモーター回転数の最高速度は、8 0 1 p . p . s であり、通常のステッピングモーターの自起動領域より高速であるので、加速減速動作が必要となる。

【 0 1 8 5 】

本発明の実施の形態では紙送りモーターの加速減速に各々 2 0 パルス必要であるが、ここでは 2 4 パルスの紙送りなので、加速減速とも各々のテーブルから 1 2 パルスずつを用い、等速部分は存在しない。よって、加速動作と、減速動作の 2 つの部分から構成される。

【 0 1 8 6 】

20

図 2 8 は、2 4 パルス紙送り動作時の加速部分である。制御 I / O コマンドは、初期設定の行から始まり、左から右に、また上から下の行へと進む (7 B 0 1 , 7 1 0 1 , 7 4 0 1 , 7 5 2 8 , 1 5 9 C , 1 4 0 0 , 7 9 0 1 ...)。

【 0 1 8 7 】

一行目の初期設定は、動作前の紙送りモーターの回転方向の設定「ここでは正方向の紙送りの例として 0 1 (逆方向なら 0 0)」(7 B 0 1)、紙送りモーター P W M 電流 O N 制御 I / O コマンド (7 1 0 1) である。

【 0 1 8 8 】

二行目はスタート相の励磁のための紙送りモーター P W M 電流 (A 相) 書き込み制御 I / O コマンド (7 4 0 1)、紙送りモーター P W M 電流 (B 相) 書き込み制御 I / O コマンド (7 5 2 8)、スタート相の励磁 (保持) 時間設定のためのウェイトタイマー制御 I / O コマンド H (6 4 u s e c 単位) (1 5 9 C)、およびウェイトタイマー制御 I / O コマンド L (0 . 5 u s e c 単位) (1 4 0 0)、である。

30

【 0 1 8 9 】

三行目は、最初の相シフト (スタート相の次の相励磁) 制御 I / O コマンド (7 9 0 1)、紙送りモーター P W M 電流 (A 相) 書き込み制御 I / O コマンド (7 4 0 1)、紙送りモーター P W M 電流 (B 相) 書き込み制御 I / O コマンド (7 5 2 8)、加速動作を実現するためのウェイトタイマー制御 I / O コマンド H (6 4 u s e c 単位) (1 5 0 5)、およびウェイトタイマー制御 I / O コマンド L (0 . 5 u s e c 単位) (1 4 2 4)、である。

40

【 0 1 9 0 】

四行目はスタート相から 1 / 6 相分だけ進んだ位置での紙送りモーター P W M 電流 (A 相) 書き込み制御 I / O コマンド (7 4 0 6)、紙送りモーター P W M 電流 (B 相) 書き込み制御 I / O コマンド (7 5 2 6)、加速動作を実現するためのウェイトタイマー制御 I / O コマンド H (6 4 u s e c 単位) (1 5 0 5)、およびウェイトタイマー制御 I / O コマンド L (0 . 5 u s e c 単位) (1 4 1 C)、である。

【 0 1 9 1 】

同様に、五行目はスタート相から 2 / 6 相分進んだ位置、六行目はスタート相から 3 / 6 相分進んだ位置、七行目はスタート相から 4 / 6 相分進んだ位置、八行目はスタート相から 5 / 6 相分進んだ位置、の制御コマンドである。基本的に四行目から八行目は、処理の

50

内容は一緒に紙送りモーターPWM電流のA相/B相に書き込む電流値が、マイクロステップ実現のために、正弦関数的に変化し、また、ウェイトタイマー制御I/Oコマンドの設定値が、加速動作実現のために少しずつ小さくなっている。

【0192】

通常の加速動作では、相切り替えのタイミングで徐々に時間間隔をせばめていくのであるが、本発明の実施の形態の場合、マイクロステップ制御を行っているため、同じ相内でも、徐々に時間を変化させていく必要がある(加速1-1~加速1-6の間でも変化させている)。

【0193】

また、実際には電流の流れは、A相/B相とも方向性があるが、ここではその電流量の絶対値だけ書き込み、方向については、モータードライバー側で変化させるので負の値はとらない。

【0194】

九行目はスタート相から6/6相分進んだ位置すなわちスタート相の次の相なので、再び相シフト(スタート相の次の次の相励磁)(7901)、紙送りモーターPWM電流(A相)書き込み(7428)、紙送りモーターPWM電流(B相)書き込み(7501)、加速動作を実現するためのウェイトタイマー制御I/OコマンドH(64usec単位)(150F)、およびウェイトタイマー制御I/OコマンドL(0.5usec単位)(14ED)、である。

【0195】

以下、同様に加速動作を本発明の実施の形態では12ステップ分繰り返す。加速が終了すると、次は減速部分である。

【0196】

図29は、24パルス紙送り動作時の減速部分である。制御I/Oコマンドは、第一行から始まり、左から右に、または上から下へと進む(7901, 7401, 7528, 1503, 1420, 7406...)。

【0197】

一行目は、最初の相シフト(減速スタート相の次の相励磁)制御I/Oコマンド(7901)、紙送りモーターPWM電流(A相)書き込み制御I/Oコマンド(7401)、紙送りモーターPWM電流(B相)書き込み制御I/Oコマンド(7528)、減速動作を実現するためのウェイトタイマー制御I/OコマンドH(64usec単位)(1503)、およびウェイトタイマー制御I/OコマンドL(0.5usec単位)(1420)、である。

【0198】

二行目は減速スタート相から1/6相分だけ進んだ位置での紙送りモーターPWM電流(A相)書き込み制御I/Oコマンド(7406)、紙送りモーターPWM電流(B相)書き込み制御I/Oコマンド(7526)、減速動作を実現するためのウェイトタイマー制御I/OコマンドH(64usec単位)(1503)、およびウェイトタイマー制御I/OコマンドL(0.5usec単位)(1424)、である。

【0199】

同様に、三行目はスタート相から2/6相分進んだ位置、四行目はスタート相から3/6相分進んだ位置、五行目はスタート相から4/6相分進んだ位置、六行目はスタート相から5/6相分進んだ位置、の制御I/Oコマンドである。基本的に二行目から六行目は、処理の内容は一緒に紙送りモーターPWM電流のA相/B相に書き込む電流値が、マイクロステップ実現のために、正弦関数的に変化し、また、ウェイトタイマー制御I/Oコマンドの設定値が減速動作実現のために少しずつ大きくなっている。

【0200】

通常の減速動作では、相切り替えのタイミングで徐々に時間間隔を広げていくのであるが、本実施例の場合、マイクロステップ制御を行っているため、同じ相内でも、徐々に時間を変化させていく必要がある(例えば、減速3-1~加速3-6の間でも変化させている

10

20

30

40

50

）。

【0201】

また、実際には電流の流れは、A相／B相とも方向性があるが、ここではその電流量の絶対値だけ書き込み、方向については、モータドライバ側で変化させるので負の値はとらない。

【0202】

七行目は減速スタート相から6／6相分進んだ位置、すなわち減速スタート相の次の相なので、再び相シフト（減速スタート相の次の次の相励磁）（7901）、紙送りモータPWM電流（A相）書き込み（7428）、紙送りモータPWM電流（B相）書き込み（7501）、加速動作を実現するためのウェイトタイマー制御I／OコマンドH（64 usec単位）（1503）、およびウェイトタイマー制御I／OコマンドL（0.5 usec単位）（1434）、である。

10

【0203】

以下、同様に加速動作を本実施例では12ステップ分繰り返す。減速が終了すると、減速最後の相（停止相）を励磁し、しかるのちに、最後の行の停止処理で、電流を切る。

【0204】

図30はメディアの構成を示した図で、メディアにはフロッピーディスク、CR-ROM、MOなどがある。メディアの構成で（1）Installerは（2）から（7）のソフトをホストシステムにインストールするためのソフトウェア、（2）RTOS Runtime LibraryはホストシステムでRTOS環境を提供するためのソフトウェア、（3）Language Monitorはホストのスプーリングサブシステムに追加されW-RTOS Host Applicationとデータのやり取りを行うソフトウェア、（4）W-RTOS Host ApplicationはLanguage Monitorから印字コマンドを受け取り、RTOS Taskへ渡す役割と、RTOS Taskをコントロールする役割を行うソフトウェア、（5）RTOS TaskはW-RTOS Host Applicationから渡される印字、スキャンコマンドなどに従って印字やスキャンを行うソフトウェアで、コントロールコード解析Task、印字処理Task、スキャン処理Task、ヘッド交換処理Task、給・排紙処理Task、タイマー管理Task、回復系処理Task、ヘッド温調処理Task、エラー処理Task、電源管理Taskからなる。（6）VCOMM Port DriverはVCOMMからの印字コマンドをRTOS Driverに引き渡す処理を行うソフトウェア、（7）RTOS DriverはVCOMM Port Driverからの印字コマンドをRTOS Taskへ引き渡すことを行うソフトウェア、（8）Utilityはプリンタの環境設定やヘッド交換などを行うユーティリティソフトウェアである。以上のソフトウェアが（1）のInstallerによりシステムに設定され実行可能となる。

20

30

【0205】

図31は、ウィンドウズ95環境下における、ウィンドウズ・アプリケーションからLPTポートにつながるプリンタへ出力する場合の、ソフトウェア処理手順を示す。図40において、4001は印刷機能を有するウィンドウズアプリケーション、4002はプリンティング・サブシステム、4003はスプーリング・サブシステム、4004はVCOMM、4005はポート・ドライバ、4006はLPTポートであり、4001～4005の動作はすべて、ウィンドウズ・システムの指令を受ける。

40

【0206】

印刷機能をもつ一般的なウィンドウズ・アプリケーション4001から出力されたデータは、プリンティング・サブシステム4002に送られ、ウィンドウズ・オペレーティング・システムにおけるグラフィックデバイスのハードウェアに対するインターフェースでいるGDIと連動してプリンタドライバが出力し、印刷用データ（コントロールコード）に変換される。さらにスプーリング・サブシステム4003に送られ、スプーラプロセスが印刷用データ（コントロールコード）をスプールファイルに蓄え、VCOMM4004

50

に対して書き出す。VCOMM4004はLPTポート・ドライバ4005を通じて、物理的なLPTポート4006に印刷用データ(コントロールコード)を書き出す。

【0207】

本発明では、Windows95環境下で動くリアルタイムOSを使用する。RTOSは、CPU動作モードのリング0でリアルタイム環境を作り、そこで複数のシグナル・プロセッシング・タスクを実行することが出来る。RTOSタスクとリング3で実行されるウィンドウズアプリケーションとは、ウィンドウ・メッセージを用いて非同期に通信する手段がある。RTOSタスクと通信するWindowsアプリケーションを、W-RTOSホストアプリケーションと呼ぶ。また、RTOSタスク間も、非同期に通信する手段を有する。

10

【0208】

図39は、従来例のウィンドウズ95環境下における、RTOSタスクとウィンドウズ・アプリケーションの通信、またRTOSタスク間の通信を示す。図39において、4901はW-RTOSホスト・アプリケーション、4902はW-RTOSホスト・アプリケーションと通信を行うRTOSタスク、4903はRTOSタスク間の通信を行うRTOSタスクである。

【0209】

RTOSタスク4902は、データの要求、データ処理の終了等をウィンドウズ・メッセージの形でW-RTOSホスト・アプリケーション4901に通信する。W-RTOSホスト・アプリケーション4901は、RTOSタスク4901の処理開始を通信し、メッセージに応じてデータの送信を行う。

20

【0210】

図32は、本発明における印刷処理の流れを示す、ソフトウェア・ブロック図である。図41に示されるように、この印刷システムは、ウィンドウズ・アプリケーション4101、プリンティング・サブシステム4102、スプーリング・サブシステム4103、Language Monitor 4104、セマフォA 4105、セマフォB 4106、共有メモリ4107、W-RTOSホスト・アプリケーション4108、RTOSタスク4109、プリンタ・ポート4110により構成される。

【0211】

ウィンドウズ95・オペレーティング・システムの場合、スプーリング・サブシステムは複数のコンポーネントにより構成され、それらのうち、VCOMMと通信しPort Driverにデータを書き出すコンポーネントとして、Port Monitor、Language Monitorがある。本発明においては、印刷用データ(コントロールコード)をスプーリング・サブシステムからフックするために、独自のLanguage Monitor 4104を作成し、スプーリング・サブシステム4103に追加する。

30

【0212】

Language Monitor 4104はいくつかの関数をエクスポートするが、双方向のプリンタを考慮しない場合、印刷用データ(コントロールコード)を処理するとき、ウィンドウズ95・システムは、StartDocPort WritePort(複数回) EndDocPortの様にLanguage Monitorの関数を呼び出す。

40

【0213】

図31で示したのと同様に、印刷機能をもつ一般的なウィンドウズ・アプリケーション4101から出力されたデータは、プリンティング・サブシステム4102で処理され、スプーリング・サブシステム4103に送られる。スプーリング・サブシステム4103を構成するコンポーネントの一つで、今回新たに追加したLanguage Monitor 4104において印刷用データ(コントロールコード)を処理する際、まず図33の動作が行われる。StartDocPort 0命令がコールされ、図33のステップ4201、4202においてセマフォA 4105とセマフォB 4106が作成される。次にステップ4203において共有メモリ4107がアロケートされる。さらにステップ4204

50

において、Language Monitor 4104の子プロセスとして、WinSPOXホスト・アプリケーション4108が生成される。最後にステップ4205においてPort MonitorのStartDocPort0命令をコールする。

【0214】

図33のステップ4204で子プロセスとして生成されたW-RTOSホスト・アプリケーション4108は、図36に示す動作を行う。すなわち、ステップ4501においてRTOSタスク4109を起動し、ステップ4502において、W-RTOSホスト・アプリケーション4108とRTOSタスク4109との通信を初期化し、ステップ4503、4504においてセマフォA4105、セマフォB4106をオープンし、ステップ4505においてセマフォA4105を待つ。

10

【0215】

一方、図33の動作を行った後、Language Monitor 4104においては、図34の動作が行われる。Write Port0命令がコールされ、図34のステップ4301において共有メモリ4107がマッピングされ、ステップ4302において印刷用データ(コントロールコード)が共有メモリ4107に対して書き出される。そしてステップ4303においてセマフォA4105を解放し、ステップ4304でセマフォBを待つ。

【0216】

他方、図36の動作を行った後、W-RTOSホスト・アプリケーション4108は、図37の動作を行う。すなわち、ステップ4601において共有メモリ4107をオープンし、ステップ4602において共有メモリ4107をW-RTOSホスト・アプリケーション4108のプロセス空間にマッピングし、ステップ4603において、この時点で図34のステップ4302で既に共有メモリ4107に対して書き出されている印刷用データ(コントロールコード)を読み出し、ステップ4604においてRTOSタスク4109に対してこの印刷用データ(コントロールコード)を書き出す。さらにステップ4605においてセマフォB4106を解放し、ステップ4606においてセマフォA4105を待つ。RTOSタスク4109において、書き出された印刷用データ(コントロールコード)は、プリンタヘッドイメージに変換され、プリンタ・ポート4110に書き出される。

20

【0217】

ウィンドウズ95・システムは、スプールされたデータがなくなるまで図34の動作を繰り返す。これにより、図34、図37の動作が繰り返されて、一連の印字処理が行われる。

30

【0218】

スプールされたデータがなくなると、ウィンドウズ95・システムは図35の動作を行う。EndDocPort0命令がコールされ、図35のステップ4401、4402においてセマフォA、Bが破棄され、ステップ4403において共有メモリが破棄される。最後にステップ4404においてPort MonitorのEndDocPort0命令をコールする。

【0219】

図38は、本発明におけるスキャナ処理の流れを示すソフトウェアブロック図である。

40

【0220】

図38に示されるように、このスキャナ・システムは、ウィンドウズ・スキャナアプリケーション4701、スキャナドライバ4702、セマフォC4703、セマフォD4704、共有メモリ4705、W-RTOSホスト・アプリケーション4706、RTOSタスク4707、スキャナ・ポート4708により構成される。

【0221】

発明の実施の形態1の図40で示したのと同様に、スキャナ機能をもつ一般的なウィンドウズ・アプリケーション4701がスキャン開始指令を出すと、スキャナ・ドライバ4702がセマフォC4703とセマフォD4704を作成し、共有メモリ4705をアロケ

50

ートする。そして、スキャナ・ドライバ4702の子プロセスとして、W-R T O S ホスト・アプリケーション4706を生成する。その後、スキャン開始コントロールコードが、R T O S タスク4707、スキャナ・ポート4708を通じてスキャナ装置に出される。

【0222】

スキャナ装置から、スキャナ・ポート4708を経由したスキャン・データは、R T O S タスク4707においてバッファリングされ、W-R T O S ホスト・アプリケーション4706に送られる。

【0223】

W-R T O S ホスト・アプリケーション4706は、共有メモリ4705を介し、セマフ
ォC4703、セマフォD4704を用いて同期を取り、スキャナ・ドライバ4702から
ウィンドウズ・スキャナ・アプリケーションにスキャン・データを送る。

10

【0224】

図40はタスクの構成を示す図で、タスクを管理するTask Controlと実行されるTaskから構成される。Task Controlはタスクの実行、実行順序を管理するTask Dispatcher60-01と時間管理を行うTimer60-02からなる。Task ControlはリアルタイムOS R T O S のシステムの一部で、Taskの管理を行う。Taskはメインの実行を行う親タスクと親タスクから生成される子タスクから構成される。親タスクは3つのフェーズを持ち、タスクの初期化を行うCreate Phase60-03、タスクの本体となるExecute Phase60-04、タスクの終了処理を行うDelete Phase60-05からなり、Create, Execute, Deleteの順で実行される。子タスクは親タスクから実行時に生成されTask1-Taskn60-06までのn個のタスクを生成することができる。親タスクのExecute Phaseと子タスク1-nはそれぞれ個別にタスクの順位を表すプライオリティを持つ。これらのタスクは、図32の4109で示される。Task Dispatcherは各タスクのプライオリティに従ってタスクの遷移状態が変わったときにタスク実行の切り換えを行う。Timerは一定の時間間隔で実行されるタスクのために、一定時間ごとに特定のタスクに実行の切り替えを行う。タスクは他のタスクとの間で同期をとるためにセマフォを持ち、セマフォの管理もTask Dispatcherで行われる。セマフォはタスクで生成されたセマフォに設定される
値によりタスクの実行の中断再開を管理することができる。セマフォの値が変更されるのを待つことをセマフォ待ち、セマフォに値を設定することをセマフォ送信と呼ぶ。

20

30

【0225】

図41はタスクの遷移関係を示した図で、まずタスクは矢印createで生成されReady状態となる。Ready状態となったタスクはTask Dispatcherにより管理されキューに登録される、キューはタスクがReady状態にされた順でプライオリティごとに登録し、登録されたタスクの中で最もプライオリティの高いタスクでキューの先頭にあるタスクがRunning状態となる。図中の矢印Runとなる。矢印changeはRunning状態のタスクを他のタスクに切り換えるためにReady状態に切り換える。Running状態のタスクがセマフォ待ちを行い、中断状態になるのが
矢印waitでWait状態となる。矢印readyはWait状態でセマフォ送信を受けるか、中断が終了することでWait状態からReady状態となりタスクのキューの最後に登録される。Running状態のタスクが終了するのが矢印exitでTerminate状態となる。Ready, Wait状態からタスクを終了するのが矢印deleteでTerminate状態となる。矢印change, exit, wait, delete, readyの各状態遷移が起きた場合にTask Dispatcherにより次のタスクがReady状態からRunning状態とされ次のタスクが実行される。Timerで実行に移されるタスクも同じ遷移を行いRunning状態とされる。ただし、Timerの場合には設定された一定時間でタスクの遷移が行われるが、通常のタスクでは矢印change, exit, wait, delete, readyの各状態遷移

40

50

が起きた場合にのみタスクの遷移が実行される。以上のタスクの管理によりタスクの実行、切り換えと、時間によるタスクの実行が行える。

【0226】

図42は、本発明の実施の形態で実行されるタスクを表した図である。図40のTask - Task nに当たる。コントロールコードに従って各タスクに遷移を変更するコントロールコード解析Task 62 - 01、コントロールコード解析Taskにより実行され印字を行う印字処理Task 52 - 02、スキャナでスキャンを行うスキャン処理Task 62 - 03、印字ヘッド、スキャンヘッドの交換を行うヘッド交換処理Task 62 - 04、給紙・排紙を行う給、排紙処理Task 62 - 05がある。また、Timerにより一定時間ごとに呼び出され各タスクに遷移を変更するタイマー管理Task 62 - 06、タイマー管理Taskにより実行される印字ヘッドのメンテナンス処理を行う回復系処理Task 62 - 07、印字・スキャンヘッドの温度管理を行うヘッド温度処理Task 62 - 08、様々なエラーを処理するエラー処理Task 62 - 09、電源の状態を管理する電源処理Task 62 - 10がある。コントロールコード解析Task、印字処理Task、スキャン処理Task、ヘッド交換処理Task、給、排紙処理Taskは同じプライオリティを持つ。タイマー管理Task、回復系処理Task、ヘッド温度処理Task、エラー処理Task、電源管理Taskは同じプライオリティを持ち、コントロールコード解析Task、印字処理Task、スキャン処理Task、ヘッド交換処理Task、給、排紙処理Taskより高いプライオリティを持つ。各タスクの詳細については後述のフローチャートに従って説明する。

10

20

【0227】

従来のプリンタ装置ではプリンタ装置内の中央演算装置CPUでこれらとほぼ同等のタスクが実行され、印字が行われていた。本発明では、これらのタスクはホストコンピュータ内の中央演算装置CPUで実行される。

【0228】

図43はコントロールコードの基本形式を示しており、ESCはコントロールコードの識別子(1Bh)でコントロールコードの開始を表す。コントロールコードには、EPSON、ESCP、HP、PCL、Canon color native commandなどがあり、印字、スキャンを行うための制御指示を行う。次にcommandでコントロールコードの種類を表す。commandの内容は図44に示す。lengthは以降に続くデータのサイズを表し、バイト単位の数値となる。length = 0は以降に続くデータのないことを表す。dataはイメージデータ、モード、フィード量などの指定を行うデータとなる。コントロールコードはアプリケーションから印字・スキャンを行うための各タスクに送られてくる。実施例1でコントロールコードは、図32のウィンドウ・アプリケーション4101、プリンティング・サブシステム4102、スプーリング・サブシステム4103、Language Monitor 4104、共有メモリ4107、W-RTO Sホスト・アプリケーション4108を経て各タスクに渡される。実施例2でコントロールコードは、図78のウィンドウ・アプリケーション90 - 01、プリンティング・サブシステム90 - 02、スプーリング・サブシステム90 - 03、VCOMM 90 - 08、VCOMM Port Driver 90 - 09、RTOS Driver 90 - 10を経て各タスクに渡される。

30

40

【0229】

図44に本発明の実施の形態の具体的なコントロールを示す。(1)ESC P0は印字処理Taskへの切り換えを行うコントロールコード。(2)ESC S0はスキャン処理Taskへの切り換えを行うコントロールコード。(3)ESC E0は印字・スキャンTaskの終了コントロールコード。(4)ESC H 1modeはヘッド交換指示コントロールコード。mode = 0でヘッド交換開始、mode = 1でヘッド交換終了。(5)ESC I0は給紙指示コントロールコード。(6)ESC 00は排紙コントロールコード。(7)ESC icolor length dataは印字イメージコントロールコードでcolorは色の指定を表し、K = ブラック、C = シアン、M = マゼンタ

50

、Y = イエローを指示し以降のデータの色が `color` で決まる。`length` がデータのサイズを表す。`data` がイメージのデータで、1 バイトは 8 ビット、1 ビットが 1 ドットを表す。このコントロールコードはラストイメージのコントロールコードでラスト単位で各色のラストデータを指定する。(8) `ESCs0` はラストスキップコントロールコードで 1 ラスタ分ラスト位置を下へ更新する。(9) `ESC f 2 feed` はフィードを指定するコントロールコードで、`feed` にフィード量を指定する。

【0230】

図 45 に 1 ラスタ分のイメージデータとラストスキップコントロールコードを連続して送る場合の例を示している。4 色分のラストデータのイメージコントロールコードとラストスキップコントロールコードで 1 ラスタ分のイメージが出来る。この順でイメージデータを 1 ページ分繰り返し送り送ることにより 1 ページのカラー印字を行う。

10

【0231】

図 46 は、印字ヘッド図 11 の 2 のノズル部分の概略構成を示した図である。符号 (A) はカラーインクジェットヘッド、(B) はモノクロインクジェットヘッドを示している。カラーヘッド (A) は、三色のカラーヘッド部分とブラックヘッドの部分からなり、上からイエロー (Y)、マゼンタ (M)、シアン (C)、ブラック (Bk) の順で並び、三色は 24 ドット、ブラックは 64 ドットから成り、それぞれの色の間には色の干渉を防ぐために 8 ドットづつの間隔が取られている。インクジェットのヘッドでは 1 ドットごとに一本のノズルが配置されており、この場合では 1 ドットは 1 / 360 DPI (ドット/インチ) の分解能を持つ。カラーデータの印字は四色のインクジェットヘッドからそれぞれの色のインクを噴出して行い、それぞれのインクの色を重ねることによりインクジェットヘッドにない色の印字も可能となる。ブラックインクジェットのヘッド部分はモノクロの印字に使われる。モノクロインクジェットヘッド (B) はモノクロ専用印字のインクジェットヘッドでブラックインクジェットヘッドだけで構成される。モノクロインクジェットヘッドは 128 ドットで構成され、一度の印字で 128 ドットの全てのインクジェットヘッドを使うことが出来る。

20

【0232】

図 47 ~ 53 は図 45 で示したイメージコントロールコードのイメージデータを蓄え加工するためのバッファを示した図である。図 47 から図 51 までのバッファは横方向 5760 ドット分のドット幅を持ち、本実施例では横方向の分解能が 720 DPI (ドット/インチ) であり、横方向は 5760 / 720 で 8 インチの幅が印字領域となる。図 52 は 8 インチ + の幅を持つ。図 47 から図 49 までのバッファはイメージをラストイメージとして保持する。図 50 から図 52 はラストイメージデータを縦横変換したカラムイメージとして保持する。YMCk は各色に対応し、ラスタ数はバッファの保持できるラスタ数を表す。ドット数は縦横変換後の縦ドット数を表し、たとえば 24 ドットは 3 バイトとなる。図 53 にラスタとカラムのデータの配置を示す。ラスタでは 1 バイト (8 ドット) 単位のデータが横方向に 1 > 2 > 3 > 4 > ... > n の順でバッファに保持され、カラムでは 1 バイト (8 ドット) 単位のデータが縦方向に 1 > 2 > 3 > 4 > ... > n の順でバッファに保持される。これは縦 24 ドットの場合である。

30

【0233】

図 47 はラストイメージデータを 8 ラスタ単位で蓄えるためのバッファ。図 48 は図 47 で蓄えられたラストイメージデータを 2 回分蓄えるバッファで、後で説明するイメージデータを加工するために使用され 16 ラスタ分の大きさを持つ。図 47 で蓄えられたデータが 8 ラスタ分一杯になると、図 48 のバッファへ送られる。図 48 のバッファにまだデータがセットされていない場合図 47 のバッファは図 48 のバッファの前半 8 ラスタに送られ、すでに図 48 の前半 8 ラスタにデータがセットされていれば図 48 のバッファの後半 8 ラスタにセットされる。図 49 は図 48 に蓄えられた加工されたデータを印字に従って保持するバッファである。印字ヘッドは図 46 の (A) に示したように一番下にブラック、シアン、マゼンタ、イエローの順で配置されているため、上に配置される色ほど後から印字されるためデータを多く蓄える必要があり K < C < M < Y の順でバッファのラスタ数

40

50

も多くなる。印字方法については後述する。図48のバッファから図49のバッファへは図48のバッファの前半8ラスタが送られ、図48のバッファは後半8ラスタが前半に移動される。これは、図48のバッファで行われる加工処理が16ラスタ単位で行われるため、加工の終了した図48の前半8ラスタが図49へ送られる。図50は図49で蓄えられたラスタデータを印字ヘッドに合わせた形で縦横変換したカラムデータを蓄えるバッファである。図50の形式のデータを図46の(A)で示した印字ヘッドに送ることで印字が出来る。図49のバッファから図50のバッファへは図49のKバッファに24ラスタ蓄えられた時点で行われる。図51は図50で作り出された印字ヘッドに対応したデータに印字パスの処理を行うためのバッファである。印字パスとは複数回印字イメージを重ねて印字する(ファイン印字)、印字データを間引いて印字する(ドラフト印字)などを行うことで、本発明の実施の形態では重ね合わせ、間引きを行わない(ハイクオリティ印字)を印字方法としているので図50のバッファと図51は同じデータとなる。図52は図51で印字パス処理を実施されたデータに対して印字補正処理を行うバッファである。印字バッファの処理補正は印字ヘッド特有の補正を行う処理で、本発明の実施の形態では印字ヘッドは図46で示したように縦一列で一直線に構成されているが、印字ヘッドが斜めに配置されているようなヘッドではヘッドの斜行に合わせて印字データをずらす(分散斜め打ち処理)処理、また印字ヘッドが複数列で構成されている場合に各々ヘッドにデータに対応させる処理などがある。図52はこれら処理を行ったデータを蓄えるために図51のバッファに処理用の領域を+している。図52でできあがったデータが印字ヘッドに送られ印字が行われる。

【0234】

図54はハイクオリティ印字の場合の印字パスを示した図で、(1)から(15)までの印字パスと示している。(2)から(15)までのパスは図を見やすくするために右にずらして図示しているが、実際の印字では(1)と同じ横位置に印字される。(1)から(15)は印字ヘッドに対応している。印字パス(1)ではブラックインクジェットヘッドの下から24ドットで1、2、3のブラックデータを印字している。1、2、3...の数字で表されているのは8ドット単位(1バイト)である。ブラックヘッドの残り40ドット(常にイメージデータなし)と他の三色ではイメージデータなし(ヌルデータ)、結果としてバンドb1にブラック(1)、(2)、(3)のみ印字される。このブラックデータは図56のKバッファの(1)、(2)、(3)当たり、他三色はC、M、Yバッファの(-)、(-)、(-)に当たる。図56のカッコ内の数字は実際に設定されているデータを意味し、-はイメージデータがヌルデータであることを示している。次に24ドットのフィールドを行い、印字パス(2)で図56のKバッファの(4)、(5)、(6)のデータを印字する。バンドb2のブラックが印字される。同様に繰り返して印字パス(4)でシアンのインクジェットヘッド位置がバンドb1に到達し、(A)、(B)、(C)のブラックデータと(1)、(2)、(3)のシアンデータを印字する。図56のCバッファのA、B、Cにセットされているイメージデータ(1)、(2)、(3)を印字する。印字パス(5)でマゼンタのインクジェットヘッド位置がバンドb1の2/3に到達し、(D)、(E)、(F)のブラックデータと(4)、(5)、(6)のシアンデータと(-)、(1)、(2)のマゼンタデータを印字する。図56のCバッファのD、E、Fにセットされているイメージデータ(4)、(5)、(6)とMバッファのD、E、Fにセットされているイメージデータ(-)、(1)、(2)を印字する。印字パス(6)でイエローのインクジェットヘッド位置がバンドb1の1/3に到達し、(10)、(11)、(12)のブラックデータと(7)、(8)、(9)のシアンデータと(3)、(4)、(5)のマゼンタデータと(-)、(-)、(1)のイエローデータを印字する。図56のCバッファの10、11、12にセットされているイメージデータ(7)、(8)、(9)とMバッファの10、11、12にセットされているイメージデータ(3)、(4)、(5)とYバッファの10、11、12にセットされているイメージデータ(-)、(-)、(1)を印字する。印字パス(7)で(13)、(14)、(15)のブラックデータと(A)、(B)、(C)のシアンデータと(6)、(7)、(8)のマゼンタデ

ータと(2)、(3)、(4)のイエローデータを印字する。図56のCバッファの13、14、15にセットされているイメージデータ(A)、(B)、(C)とMバッファの13、14、15にセットされているイメージデータ(6)、(7)、(8)とYバッファの13、14、15にセットされているイメージデータ(2)、(3)、(4)を印字する。印字バンドに印字ヘッドが到達するまではヌルデータが印字され、結果として印字が行われないことになる。

【0235】

図54の印字パス(7)が終了してバンドb1は、4色ともに印字されたことになる。バンドb1とバンドb2の1/3は、カラー印字が完全に終了したことになる。同様に印字パスとフィードを繰り返すことにより、印字パス(14)まで終了するとバンドb8までの印字が完成する。

10

【0236】

図55は図49にイメージデータがセットされた状態を示した図である。図中KSPはKデータセットポインタの開始位置を示し、CSPはCデータセットポインタの開始位置を示し、MSPはMデータセットポインタの開始位置を示し、YSPはYデータセットポインタの開始位置を示す。KSPはKバッファの先頭を示し、CSP、MSP、YSPは各バッファの最終から24ラスタの前の位置を示している。イメージデータのセットはこれら4つのポインタの示す先に行われる。8ラスタ単位で1から順にセットされバッファの最終までセットされると各データセットポインタはバッファの先頭に戻され、バッファが繰り返し利用される。C、M、Yの各バッファではバッファ先頭からCSP、MSP、YSPの初期位置まではヌルデータがあらかじめセットされている。図48にセットされたイメージデータは8ラスタ単位で取り出され、図55のバッファにセットされる。

20

【0237】

図56は図55にセットされたイメージデータを取り出して図50のバッファにセットするために図49のバッファからデータを取り出す順序を示した図で、図中KGPはKデータゲットポインタの開始位置を示し、CGPはCデータゲットポインタの開始位置を示し、MGPはMデータゲットポインタの開始位置を示し、YGPはYデータゲットポインタの開始位置を示す。イメージデータの取り出しは各ポインタの指し示す先から行われる。KバッファからはKGP、CバッファからはCGP、MバッファからはMGP、YバッファからはYGPのそれぞれ指し示す先からデータが取り出され、図54の印字パス(1)では図56の1、2、3がそれぞれ取り出される。バッファの設定内容は図55で示されており、C、M、Yの各色では1、2、3のイメージデータにヌルデータがセットされており、イメージデータの取り出しではこのヌルデータが取り出される。()はヌルデータを表している。Cバッファでは1-9、Mバッファでは1-D、Yバッファでは1-11までがヌルデータとなる。

30

【0238】

図57はポインタとカウンタを示した図で、KSP71-01はKデータバッファにイメージデータをセットする位置を示すポインタで、CSP71-02はCデータバッファにイメージデータをセットする位置を示すポインタで、MSP71-03はMデータバッファにイメージデータをセットする位置を示すポインタで、YSP71-04はYデータバッファにイメージデータをセットする位置を示すポインタ。KGP71-05はKデータバッファからイメージデータを取り出す位置を示すポインタで、CGP71-06はCデータバッファからイメージデータを取り出す位置を示すポインタで、MGP71-07はMデータバッファからイメージデータを取り出す位置を示すポインタで、YGP71-08はYデータバッファからイメージデータを取り出す位置を示すポインタ。RCNT71-09はKSPが示す位置にイメージデータがセットされるごとにカウントアップされ、イメージデータが取り出されることでクリアされる。取り出しに必要なイメージデータがセットされたかどうかを判定するために使用される。これらの領域は図2のRAM52に配置される。

40

【0239】

50

以下、フローチャートに従って本実施例のシステム動作を詳細に説明する。なお、以下に述べるフローチャートに示す手順は、ホストコンピュータの中央演算装置CPUにおいて実行される。

【0240】

図58は図42に示したコントロールコード解析Task62-01を詳細に説明するフローチャートで、まずステップS201でコントロールコード取得を行う。コントロールコード取得は発明の実施の形態1では図32の4108のW-RTOShost・アプリケーションからコントロールコードを受け取る。また発明の実施の形態2では図78の90-09のVCOMM Port Driverから90-10のRTOS Driverを経てコントロールコードを受け取る。

10

【0241】

コントロールコードを受け取るとステップS202へ進み、印字コントロールコードかどうかのチェックを行う。印字コントロールコードは図44の(1)で示したコントロールコードでステップS202はこのコントロールコードとの比較を行う。ステップS202で印字コントロールコードであればステップS203へ進み、印字Taskへセマフォを送信する。次にステップS210へ進みタスクの終了待ちとなる。各タスクは自分の処理が終了するとき終了のセマフォを送信する。ステップS210で待ち状態となっているコントロールコード解析タスクはステップS201へ進む。ステップS202で印字コントロールコードでない場合ステップS204へ進み、スキャンコントロールコードかどうかのチェックを行う。スキャンコントロールコードは図44の(2)で示したコントロールコードでステップS204はこのコントロールコードとの比較を行う。ステップS204でスキャンコントロールコードであればステップS205へ進み、スキャンTaskへセマフォを送信する。次にステップS210へ進みタスクの終了待ちとなる。ステップS204でスキャンコントロールコードでない場合ステップS206へ進み、ヘッド交換コントロールコードかどうかのチェックを行う。ヘッド交換コントロールコードは図44の(4)で示したコントロールコードでステップS206はこのコントロールコードとの比較を行う。ステップS206でヘッド交換コントロールコードであればステップS207へ進み、ヘッド交換Taskへセマフォを送信する。次にステップS210へ進みタスクの終了待ちとなる。ステップS206でヘッド交換コントロールコードでない場合ステップS208へ進み、給・排紙コントロールコードかどうかのチェックを行う。給・排紙コントロールコードは図44の(5)、(6)で示したコントロールコードでステップS208はこのコントロールコードとの比較を行う。ステップS208で給・排紙コントロールコードであればステップS209へ進み、給・排紙Taskへセマフォを送信する。次にステップS210へ進みタスクの終了待ちとなる。給・排紙コントロールコードでない場合ステップS201へ進みコントロールコード取得となる。以上の処理によりコントロールコードを取得しコントロールコードの内容に対応したタスクへセマフォを送信し各タスクの終了セマフォ待ちとなりコントロールコード解析TaskがRunning状態からWait状態となる。Task Dispatcherによりコントロールコード解析Taskがセマフォを送信したタスクがRunning状態となる。

20

30

【0242】

図59は図42に示した印字処理Task62-02を詳細に説明するフローチャートで、まずステップS221で初期化を行う。初期化では図47～図53で示したバッファの確保、初期化、図57で示したポインタの初期設定、キャリッジ初期化動作(図23で説明)を行う。次にステップS222へ進み、印字のセマフォ待ちとなる。印字処理Taskは印字に対するセマフォが送信されるまでステップS222で待ち状態となる。印字セマフォは図58で説明したコントロールコード解析TaskのステップS202で印字コントロールコードを解釈し、ステップS203で送信される。それまで印字処理TaskはステップS222で待ち状態となる。これは図41で説明したWait状態からReady状態を経てRunning状態にタスクの状態が変化したことを表している。印字セマフォが送信されるとステップS223へ進み印字コントロールコードの取得となる。印

40

50

字コントロールコード受け取ると印字コントロールコードを解析するためにステップ S 2 2 4 へ進みイメージコントロールコードかどうかのチェックを行う。イメージコントロールコード図 4 4 の (7) であればステップ S 2 2 5 へ進みイメージ展開処理を実行する。イメージ展開処理は送られてきたイメージデータをバッファに展開し、データが整ったところでプリンタ部へ送る処理を行う。イメージ展開処理は図 6 0 で詳細に説明する。ステップ S 2 2 5 で 1 ラスタ分のイメージデータが処理されると、ステップ S 2 2 3 へ戻り次のコントロールコード取得となる。ステップ S 2 2 4 でイメージコントロールコードでなければステップ S 2 2 6 へ進み、フィードコントロールコードかどうかのチェックを行う。フィードコントロールコード図 4 4 の (9) であればステップ S 2 2 7 へ進み、フィード処理を行う。フィード処理はフィードモーターを回して紙をフィードする処理を行う。フィード処理は図 6 3 で詳細に説明する。フィード処理が行われて紙のフィードが実行されると、ステップ S 2 2 3 へ戻り次のコントロールコード取得となる。ステップ S 2 2 6 でフィードコントロールコードでなければステップ S 2 2 8 へ進み、給・排紙コントロールコードかどうかのチェックを行う。給・排紙コントロールコード図 4 4 の (5)、(6) であればステップ S 2 2 9 へ進み、給・排紙処理を行う。給・排紙処理は新しい紙の給紙、あるいは紙の排紙を行う処理で、給・排紙処理は図 6 4 で詳細に説明する。給・排紙処理が行われて紙の給紙、排紙が実行されると、ステップ S 2 2 3 へ戻り次のコントロールコード取得となる。ステップ S 2 2 8 で給・排紙コントロールコードでなければステップ S 2 3 0 へ進み、終了コントロールコードかどうかのチェックを行う。終了コントロールコード図 4 4 の (3) でなければ、ステップ S 2 2 3 へ戻り次のコントロールコード取得となる。ステップ S 2 3 0 で終了コントロールコード図 4 4 の (3) であればステップ S 2 3 1 へ進み、印字処理の終了を通知する終了セマフォを送信し印字処理 T a s k は終了となる。この終了により図 4 1 の R u n n i n g 状態にあった印字タスクが W a i t 状態に変化することを意味している。これら一連の処理によりイメージデータが展開され印字が行われる。

【 0 2 4 3 】

図 6 0 は図 5 9 のステップ S 2 2 5 を詳細に説明するフローチャートで、ステップ S 2 4 1 でコントロールコードの所得を行う。イメージ展開処理はコントロールコード図 4 4 の (7) あるいは (8) の場合に呼び出される。次にステップ S 2 4 2 へ進み、取得したコントロールコードがイメージコントロールコード図 4 4 (7) どうかをチェックする。イメージコントロールコードであればステップ S 2 4 3 へ進み、ラスタデータのセットを行う。ラスタイメージデータは図 4 4 の (7) に示したように 4 つの色に区別され、それぞれの色に対応したバッファにセットされる。セットされるバッファは図 4 7 のバッファでそれぞれ 4 色 Y、M、C、K のバッファを 8 ラスタ分ずつ持っている。8 ラスタ中の図 5 7 に示したラスタカウント R C N T に設定されている値のラスタにイメージデータがセットされる。イメージデータのセットが終わると戻りとなる。ステップ S 2 4 2 でイメージコントロールコードでなければステップ S 2 4 4 へ進みラスタスキップコントロールコードかどうかのチェックを行う。ラスタスキップコントロールコードは図 4 4 の (8) で示され、このコントロールコードによりラスタ位置を 1 ラスタ分先へ進める。ラスタスキップコントロールコードでなければ戻りとなる。ラスタスキップコントロールコードであれば、ステップ S 2 4 5 へ進みデータの切り出しを行うかどうかのチェックを行う。データの切り出しを行うかどうかのチェックは図 4 9 のバッファに K データが 2 4 ラスタ分新たにセットされたかどうかで行われ、データがセットされた場合イメージの切り出し可能となる。これは図 5 5 の K データセットポインタ K S P がデータのセットにより図 5 5 では 1、2、3 あるいは 4、5、6 のように 8 ラスタデータが 3 個分 K のバッファにセットされたかどうかでチェックされる。ステップ S 2 4 5 でデータ切り出し可能でなければ、ステップ S 2 4 6 へ進みラスタカウント R C N T が 8 ラスタかどうかをチェックする。8 ラスタになっていなければステップ S 2 4 7 へ進みラスタカウント R C N T を 1 カウントアップし戻りとなる。ステップ S 2 4 6 で 8 ラスタとなればステップ S 2 4 8 へ進み 8 ラスタ分のバッファ図 4 7 から図 4 8 へデータの転送を行う。図 4 8 のバッファは 1 6 ラ

10

20

30

40

50

タ分の大きさを持ち、図47のバッファ2つ分の大きさとなる。はじめ図47のバッファのデータは図48のバッファの前半の8ラスタにセットされる。次のデータのセットの際には16ラスタの後半にセットされる。次にステップS249へ進み図48のバッファの後半にデータが転送されたかどうかをチェックする。後半への転送が行われていなければステップS252へ進みラスタカウンタRCNTをクリアして戻りとなる。ステップS249で後半へのデータ転送が行われていれば、ステップS250へ進みイメージ加工の処理を行う。イメージ加工の処理は、図48のバッファにセットされた16ラスタのデータの前半8ラスタに対して行われ、各4色の色データの配置によりイメージデータの変換を行う処理を示している。これら処理はたとえば境界検知処理と呼ばれ隣接する黒データをシアン、マゼンタ、イエローデータに置き換える事によりイメージデータの濃度を下げインクのべたつきを防ぐ処理である。これらの処理を送られてきた元データに対して行いデータの加工を行った後のデータを以降の処理に用いる。次にステップS251へ進みステップS250で作られ出したデータをイメージ展開用のバッファ図49へ転送する。データの転送は図48のバッファの前半8ラスタが図49のバッファに送られる。バッファにセットされる位置は図55のKデータセットポイントKSP、CデータセットポイントCSP、MデータセットポイントMSP、YデータセットポイントYSPのそれぞれの色に対応した位置で行われ、図55の一つの番号分のラスタデータが図48の前半から図49のバッファへ転送される。転送後の図48のバッファは後半の8ラスタデータが前半に転送される。以降図48のバッファには後半のデータが図47のバッファからセットされる。次にステップS252へ進みラスタカウンタRCNTをクリアして戻りとなる。ステップS245でデータ切り出し可能であればステップS253へ進みイメージの切り出し処理を実行する。イメージの切り出し処理はステップS251でセットされたイメージデータを印字ヘッドに必要とされる分だけ図49のバッファから取り出してくる処理で、それは図56に示したKデータゲットポイントKGP、CデータゲットポイントCGP、MデータゲットポイントMGP、YデータゲットポイントYGPから8ラスタ単位で番号付けされた3個分(24ラスタ)を取り出すことで行われる。取り出されたデータはラスタデータであり印字ヘッドにセットするデータはカラムデータとなる。このためラスタデータをカラムデータに変換するための縦横変換が行われる。縦横変換されたデータは図50のバッファにセットされる。黒データは印字ヘッド分の64ドットが確保されているが切り出されたデータ24ドットは64ドットの下から24ドットにセットされる。黒データの残り40ドットはクリアされデータはセットされない。次にステップS254へ進み印字パスの処理を行う。印字パスの処理は図50にステップS253で切り出されたデータに対して行われ、処理されたデータが図51のバッファにセットされる。印字パスの処理は、一つの印字バンドを複数回に分けて印字することにより高精細な印字を行うファイン印字、印字データを間引いて濃度を下げるドラフト印字などがあり、これらの処理は印字データに加工を行うことで実現される。ファイン印字であればあらかじめ決めたファインマスクパターンと元のイメージデータとでAND処理を行うことで加工され、ドラフト処理であれば一定のパターンとで間引きを行う。次にステップS255へ進み印字補正処理を行う。印字補正処理はステップS254で印字パス処理が行われた図51のバッファに対して行われ、図52のバッファに補正処理が実行されたデータがセットされる。印字補正処理には、印字ヘッドが斜めに配置された印字ヘッドを用いる場合、印字ヘッドが複数列に配置されているなどの印字ヘッドに特有の特性を補正する処理がある。本実施例ではヘッドが斜めに配置されている場合で、図51にセットされたデータを印字ヘッドの斜行した分だけ印字データを逆方向に斜行することを行う。これは印字データをその分だけシフトすることで行いシフトしたデータを図52にセットする。次にステップS256へ進み印字制御I/Oコントロールコード作成処理を実行する。印字制御I/Oコントロールコード作成処理はステップS255で完成した印字イメージの印字ヘッドに対応したデータに印字制御I/Oコマンドを付加することで行う。一連の印字は印字データを印字し24ドット分のフィードを行い印字ヘッドを元の印字位置へ戻す処理で構成される。詳細は図61のフローチャートで説明する。次にステップS257へ進み制御I/Oコマンド転送処

10

20

30

40

50

理を実行する。制御 I / O コマンド転送処理はステップ S 2 5 6 で完成したイメージデータと印字の制御 I / O コマンドを送信する。詳細は図 6 2 のフローチャートで説明する。次に戻りとなる。

【 0 2 4 4 】

図 6 1 は図 6 0 のステップ S 2 5 6 を詳細に説明するフローチャートで、印字制御 I / O コマンド作成処理を示している。印字制御 I / O コマンド作成処理は印字のためのキャリッジモーター C M を制御して印字イメージデータを印字し、ラインフィードモーターを制御してラインフィードを行い、キャリッジモーター C M を印字前の位置に戻す処理を行う。

【 0 2 4 5 】

まず、ステップ S 2 6 1 でキャリッジモーター C M の加速制御 I / O コマンドの作成を行う。C M 加速制御 I / O コマンドは図 2 7 で示した制御 I / O コマンドで、これら制御 I / O コマンドを使用する。印字のためのキャリッジモーター C M の制御はキャリッジモーター C M を加速し、次に等速として等速時に印字データを印字ヘッドから吐出し、イメージデータを印字後減速する。次にステップ S 2 6 2 でキャリッジモーター C M の等速制御 I / O コマンドと図 6 0 ステップ S 2 5 5 で作成されたイメージデータに合わせてキャリッジモーター C M の等速制御と印字ヘッドの吐出制御の制御 I / O コマンドを作成する。C M 等速制御 I / O コマンドは図 2 6 で示した制御 I / O コマンドで、これら制御 I / O コマンドを使用しイメージデータにヒートデータレジスタ制御 I / O コマンド 3 2 x x (図 1 8) x x はイメージデータを付加し、ヒート・スキャン駆動トリガ (図 1 8) 3 0 0 1 を付加することで印字イメージを含んだ C M 等速制御 I / O コマンドを作成する。次にステップ S 2 6 3 でキャリッジモーター C M の減速制御 I / O コマンドを作成する。C M 減速制御 I / O コマンドは図 2 9 で示した制御 I / O コマンドで、これら制御 I / O コマンドを使用する。ステップ S 2 6 1 ~ S 2 6 3 でキャリッジモーター C M の印字制御 I / O コマンドが作成される。

【 0 2 4 6 】

次にステップ S 2 6 4 でラインフィードモーター L F の加速制御 I / O コマンドを作成する。L F 加速制御 I / O コマンドは図 2 8 で示した制御 I / O コマンドで、これら制御 I / O コマンドを使用する。次にステップ S 2 6 5 でラインフィードモーター L F の等速制御 I / O コマンドを作成する。ただし、2 4 ドットのフィードを行う場合には、等速制御 I / O コマンドはない。次にステップ S 2 6 6 でラインフィードモーター L F の減速制御 I / O コマンドを作成する。L F 減速制御 I / O コマンドは図 3 1 で示した制御 I / O コマンドで、これらコマンドを使用する。ステップ S 2 6 4 ~ S 2 6 6 でラインフィードモーター L F で 2 4 ドットのフィードを行う分のフィード制御 I / O コマンドが作成される。

【 0 2 4 7 】

次にステップ S 2 6 7 でキャリッジモーター C M の加速制御 I / O コマンドを作成する。C M 加速制御 I / O コマンドは図 2 5 で示した制御 I / O コマンドで、これら制御 I / O コマンドを使用する。次にステップ S 2 6 8 でキャリッジモーター C M の等速制御 I / O コマンドを作成する。C M 等速制御 I / O コマンドは図 2 6 で示した制御 I / O コマンドで、これら制御 I / O コマンドを使用する。次にステップ S 2 6 9 でキャリッジモーター C M の減速制御 I / O コマンドを作成する。C M 減速制御 I / O コマンドは図 2 7 で示した制御 I / O コマンドで、これら制御 I / O コマンドを使用する。ステップ S 2 6 7 ~ S 2 6 9 でキャリッジモーター C M の戻り制御 I / O コマンドが作成される。戻り制御 I / O コマンドでは印字制御とは逆方向にキャリッジモーター C M を動かすためステップ S 2 6 7 にはキャリッジモーター C M の方向を逆転する制御 I / O コマンドを含む。以上の処理により印字してラインフィードを行いキャリッジモーター C M を元の位置に戻す制御 I / O コマンドが作成される。

【 0 2 4 8 】

図 6 2 は図 6 0 のステップ S 2 5 7 を詳細に説明するフローチャートで、制御 I / O コマ

10

20

30

40

50

ンド送信処理を示している。制御 I / O コマンドの送信処理は図 60 ステップ S 2 5 6 で作成された印字制御 I / O コマンドをプリンタ部へ送信するための処理である。まず、ステップ S 2 7 1 でプリンタ部が制御 I / O コマンド受信可能であるかをチェックする。受信可能であるかどうかのチェックは図 10 の B P S T A R T の値をチェックすることで行う。B P S T A R T の値が 0 であれば受信可能となる。受信可能でなければステップ S 2 7 1 を繰り返す。ステップ S 2 7 1 で制御 I / O コマンド受信可能であれば、ステップ S 2 7 2 へ進み、プリンタ部と接続するポートのアドレス制御 I / O コマンドを書き込む。制御 I / O コマンド書き込みのポートは図 6 の (c) で示されこのポートアドレスにデータを書き込むことで行える。データ書き込み後の図 7 P R I N T E R M E M O R Y の詳細を図 7 2 ~ 図 7 4 に示す。図 7 2 で P R I N T E R M E M O R Y には印字制御 (1 - 1) - (1 - 3) とラインフィード制御 (2 - 1) - (2 - 2) と戻り制御 (3 - 1) - (3 - 3) の制御 I / O コマンドがセットされている。(1 - 1) は図 6 1 ステップ S 2 6 1 で作成される C M 加速制御 I / O コマンドで、その内容は図 2 5 に示される。(1 - 2) は図 6 1 ステップ S 2 6 2 で作成される制御 I / O コマンドで C M 等速と印字イメージを付加したもので図 7 3 で詳細に説明する。(1 - 3) は図 6 1 ステップ S 2 6 3 で作成される C M 減速制御 I / O コマンドで、その内容は図 2 7 に示される。(2 - 1) は図 6 1 ステップ S 2 6 4 で作成される L F 加速制御 I / O コマンドで、その内容は図 2 8 に示される。(2 - 2) は図 6 1 ステップ S 2 6 6 で作成される L F 減速制御 I / O コマンドで、その内容は図 2 9 に示される。(3 - 1) は図 6 1 ステップ S 2 6 7 で作成される C M 加速制御 I / O コマンドで、その内容は図 2 5 に示される。(3 - 2) は図 6 1 ステップ S 2 6 8 で作成される C M 等速制御 I / O コマンドで、その内容は図 2 6 に示される。(3 - 3) は図 6 1 ステップ S 2 6 9 で作成される C M 減速制御 I / O コマンドで、その内容は図 2 7 に示される。(4) は終了制御 I / O コマンドがセットされる。終了制御 I / O コマンドは 1 0 0 0 で図 10 の B P E N D で制御 I / O コマンドの実行は B P E N D で終了する。図 7 3 は図 7 2 の (1 - 2) の内容を示した図で、図 2 6 の等速制御にイメージのデータを付加して印字を実行するように構成される。第 1 カラムから第 n カラムまでの制御を示しており、図中 6 9 0 1 から 6 5 2 8 は図 2 5 と同じで 3 2 0 0 + d a t a の部分がイメージデータを吐出するためにヒートレジスタにデータを送る制御 I / O コマンドとなる。制御 I / O コマンド 3 2 0 0 は図 1 8 のアドレス 3 2 H D R でヒートデータレジスタである。ヒートデータレジスタは印字のためのヒータレジスタで印字ヘッドに対応する。その詳細は図 1 3 で説明されている。d a t a (1 - 1) から d a t a (1 - 1 7) は印字ヘッド 1 列分のデータでデータの構成は図 7 4 に示す。3 2 0 0 + d a t a のつぎの 1 4 0 6 はデータを送る時間間隔を指定する制御 I / O コマンドで 2 マイクロ秒コンドの時間をとる。3 2 0 0 + d a t a (1 - 1 7) の後に 1 5 0 0 , 1 4 E 3 で 1 カラムごとの時間設定を行う。ここまでの時間でインクを吐出する時間をキャリッジの移動時間に合わせる。3 1 0 1 は図 1 8 のアドレス 3 1 H D R U N でセットしたイメージデータの吐出を指定する。これで 1 カラム分の印字制御が構成され、同時に n カラムまで繰り返すことで n カラム分の印字が出来る。図 7 4 は印字データの印字位置を表した図で 1 から n カラムの印字となる。これは図 6 0 のステップ S 2 5 5 までで作成された印字イメージとなる。制御 I / O コマンド書き込み後ステップ S 2 7 3 へ進み制御 I / O コマンドの実行を指定する。制御 I / O コマンド実行は制御 I / O コマンド制御ポート図 6 (a) で示されこのポートに実行を指示して書き込むことにより行える。これは図 10 の B P S T A R T に 1 をライトする事で行う。制御 I / O コマンドの実行は図 9 のタイミングチャートで説明した手順に従って行われる。以上の処理により制御 I / O コマンドのプリンタ部への送信と制御 I / O コマンドの実行が行える。

【 0 2 4 9 】

図 6 3 は図 5 9 のステップ S 2 2 7 を詳細に説明するフローチャートで、フィード処理を示している。フィード処理はフィードコントロールコード図 4 4 (9) で示されるコントロールコードに対する処理でまずステップ S 2 8 1 でラインフィードモーター L F の加速制御 I / O コマンドを作成する。次にステップ S 2 8 2 でラインフィードモーター L F の

10

20

30

40

50

等速制御 I / O コマンドを作成する。等速制御 I / O コマンドは図 4 4 (9) で示されたコントロールコードの f e e d に設定されている値分の等速が継続するように作成される。次にステップ S 2 8 3 ラインフィードモーター L F の減速制御 I / O コマンドを作成する。次に、ステップ S 2 8 4 でプリンタ部が制御 I / O コマンド受信可能であることをチェックする。受信可能でなければステップ S 2 8 4 を繰り返す。ステップ S 2 8 4 で制御 I / O コマンド受信可能であれば、ステップ S 2 8 5 へ進み、プリンタ部と接続するポートのアドレスに制御 I / O コマンドを書き込む。これは図 6 2 のステップ S 2 7 2 と同様になる。次にステップ S 2 8 6 で制御 I / O コマンド実行を行う。これは図 6 2 のステップ S 2 7 3 と同様になる。以上の処理によりフィードコントロールコードに対する処理が行われフィードが実行される。

10

【 0 2 5 0 】

図 6 4 は図 5 9 のステップ S 2 2 9 を詳細に説明するフローチャートで、給・排紙処理を示している。まずステップ S 2 9 1 で給紙コントロールコードあるいは排紙コントロールコードに対する制御 I / O コマンドを作成する。次に、ステップ S 2 9 2 でプリンタ部が制御 I / O コマンド受信可能であることをチェックする。受信可能でなければステップ S 2 9 2 を繰り返す。ステップ S 2 9 2 で制御 I / O コマンド受信可能であれば、ステップ S 2 9 3 へ進み、プリンタ部と接続するポートのアドレスに制御 I / O コマンドを書き込む。これは図 6 2 のステップ S 2 7 2 と同様になる。次にステップ S 2 9 4 で制御 I / O コマンド実行を行う。これは図 6 2 のステップ S 2 7 3 と同様になる。以上の処理により給紙・排紙に対する処理が行われる。

20

【 0 2 5 1 】

以上、図 5 9 から図 6 4 で印字処理について説明した。

【 0 2 5 2 】

図 6 5 は図 4 2 に示したスキャン処理 T a s k 6 2 - 0 3 を詳細に説明するフローチャートで、まず、ステップ S 3 0 1 で初期化を行う。初期化ではバッファの確保、初期化などを行う。次にステップ S 3 0 2 へ進み、スキャンのセマフォ待ちとなる。スキャン処理 T a s k はスキャンに対するセマフォが送信されるまでステップ S 3 0 2 で待ち状態となる。スキャンセマフォは図 5 8 で説明したコントロールコード解析 T a s k のステップ S 2 0 4 でスキャンコントロールコードを解釈し、ステップ S 2 0 5 で送信される。それまでスキャン処理 T a s k はステップ S 3 0 2 で待ち状態となる。これは図 4 1 で説明した W a i t 状態から R e a d y 状態を経て R u n n i n g 状態にタスクの状態が変化したことを表している。印字セマフォが送信されるとステップ S 3 0 3 へ進みスキャンコントロールコードの取得となる。スキャンコントロールコード受けるとスキャンコントロールコードを解析するためにステップ S 3 0 4 へ進みスキャンコントロールコードかどうかのチェックを行う。スキャンコントロールコードは図 4 4 (1 0) で示されこのコントロールコードであればステップ S 3 0 5 へ進む。ステップ S 3 0 5 ではスキャンの処理が行われスキャンしたイメージデータをスキャンアプリケーションに戻すことができる。詳細は図 6 6 のフローチャートで説明する。ステップ S 3 0 6 からステップ S 3 1 1 までは図 5 9 の印字処理 T a s k のステップ S 2 2 6 からステップ S 2 3 1 と同様で説明は省略する。

30

【 0 2 5 3 】

図 6 6 は図 6 5 のステップ S 3 0 5 を詳細に説明するフローチャートでまずステップ S 3 2 1 でスキャン制御 I / O コマンドの作成を行う。スキャン制御 I / O コマンドはスキャン動作を行うために制御 I / O コマンドでキャリッジモーター C M を加速し、等速とした後スキャナーヘッドからスキャンデータを読み出して保存するための制御 I / O コマンドをセットし、減速制御 I / O コマンドをセットする。次にスキャンヘッド分のフィードを行い、キャリッジを戻す処理をセットする。次に、ステップ S 3 2 2 でプリンタ部が制御 I / O コマンド受信可能であることをチェックする。受信可能でなければステップ S 3 2 2 を繰り返す。ステップ S 3 2 2 で制御 I / O コマンド受信可能であれば、ステップ S 3 2 3 へ進み、プリンタ部と接続するポートのアドレスに制御 I / O コマンドを書き込む。これは図 6 2 のステップ S 2 7 2 と同様になる。

40

50

【0254】

データ書き込み後の図7 PRINTER MEMORYの詳細を図75～図77に示す。図75でPRINTER MEMORYにはスキャン制御(1-1)-(1-3)とラインフィード制御(2-1)-(2-3)と戻り制御(3-1)-(3-3)の制御I/Oコマンドがセットされている。(1-1)は図61ステップS261で作成されるCM加速制御I/Oコマンドと同じで、その内容は図25に示される。(1-2)はスキャン制御I/OコマンドでCM等速とイメージデータ取得制御I/Oコマンドを付加したもので図76で詳細を説明する。(1-3)は図61ステップS263で作成されるCM減速制御I/Oコマンドと同じで、その内容は図27に示される。(2-1)は図61ステップS264で作成されるLF加速制御I/Oコマンドと同じで、その内容は図28に示される。(2-2)は図63ステップS282で作成されるLF等速制御I/Oコマンドと同じで、その内容は図28に示される。(2-3)は図61ステップS266で作成されるLF減速制御I/Oコマンドと同じで、その内容は図29に示される。(3-1)は図61ステップS267で作成されるCM加速制御I/Oコマンドと同じで、その内容は図25に示される。(3-2)は図61ステップS268で作成されるCM等速制御I/Oコマンドと同じで、その内容は図26に示される。(3-3)は図61ステップS269で作成されるCM減速制御I/Oコマンドと同じで、その内容は図27に示される。(4)は終了制御I/Oコマンドがセットされる。終了制御I/Oコマンドは1000で図10のB PENDで制御I/Oコマンドの実行はB PENDで終了する。(5)はスキャンイメージデータ保存エリアでスキャンしたイメージデータを1ライン分スキャンが終了するまで保存する領域である。図76は図75の(1-2)の内容を示した図で、図26の等速制御にイメージデータ取得制御I/Oコマンドを付加してスキャンを実行するように構成される。第1カラムから第2カラムまでの制御を示しており以降同様にnカラムまで繰り返す。図中6901から6528は図25と同じで、3101から始まって1500, 1404, 133A, 1200, 133B, 1200までが1カラムのスキャンを指示し、スキャンヘッドの2バイト分のデータを取り出してスキャンイメージデータ保存エリアに保存する制御I/Oコマンドとなる。3101は図18のアドレス31H DRUNでイメージデータの取り込みタイミングを指定する。133A, 1200, 133B, 1200はスキャンしたイメージデータをイメージデータが格納されているレジスタ3A, 3B(図18)から図75(5)のスキャンイメージデータ保存エリアに保存する。

【0255】

スキャンされたイメージデータの構成は図77に示す。1カラムのスキャン動作、16バイトイメージデータ保存後の1502, 14CFで1カラムごとの時間設定を行う。ここまでのトータル時間でスキャンする時間をキャリッジの移動時間に合わせる。これで1カラム分のスキャン制御が構成され、同様にnカラムまで繰り返すことでnカラム分のスキャンが出来る。(3)はスキャンデータの構成を表した図で横方向に1からnカラムで縦1から16バイトの構成で128ドットとなる。

【0256】

次にステップS324で制御I/Oコマンド実行を行う。これは図62のステップS273と同様になる。次にステップS325でプリンタ部へ送った制御I/Oコマンドが実行されてスキャンが終了したかどうかのチェックを行う。終了していなければステップS325へ戻る。ステップS325でスキャンが終了すると、ステップS326へ進み、スキャンしたイメージデータをスキャンアプリケーションへ送信する。スキャンしたイメージデータはプリンタ部に保存されステップS326でプリンタ部から読み出される。イメージデータの読み出しは図6(c)のポートで行う。図38にスキャン動作を行うタスクとWindowsスキャナ・アプリケーションの構成を示す。

【0257】

図65、図66のスキャン処理よりプリンタ部でスキャンされたイメージデータをスキャンアプリケーションへ送り出すことができる。

【0258】

10

20

30

40

50

図 6 7 は図 4 2 に示したタイマー管理 Task 6 2 - 0 6 を詳細に示したフローチャートでタイマー管理 Task は図 4 0 の Task Control の Timer により一定の時間ごとに呼び出される。ステップ S 3 3 1 で経過時間の算出を行う。タイマー管理 Task は一定時間で呼び出されるため呼び出される回数から経過時間を算出することが出来る。次にステップ S 3 3 2 へ進み回復系処理時間かどうかをチェックする。ステップ S 3 3 1 で算出された時間を基に一定の時間ごとに回復系の処理を実行するためにチェックが行われる。ステップ S 3 3 2 で回復系の処理時間であれば、ステップ S 3 3 3 へ進み回復系処理 Task へセマフォを送信する。次にステップ S 3 4 0 へ進み終了セマフォ待ちとなる。ステップ S 3 3 2 で回復系処理時間でなければステップ S 3 3 4 へ進み、印字ヘッド温調処理時間かどうかをチェックする。ステップ S 3 3 4 で印字ヘッド温調処理時間であれば、ステップ S 3 3 5 へ進み印字ヘッド温調処理 Task へセマフォを送信する。次にステップ S 3 4 0 へ進み終了セマフォ待ちとなる。ステップ S 3 3 4 で印字ヘッド温調処理時間でなければステップ S 3 3 6 へ進み、エラーをチェックする。ステップ S 3 3 6 でエラーがあれば、ステップ S 3 3 7 へ進みエラー処理 Task へセマフォを送信する。次にステップ S 3 4 0 へ進み終了セマフォ待ちとなる。ステップ S 3 3 6 でエラーでなければステップ S 3 3 8 へ進み、電源管理変更をチェックする。ステップ S 3 3 8 で電源管理変更があれば、ステップ S 3 3 9 へ進み電源管理 Task へセマフォを送信する。次にステップ S 3 4 0 へ進み終了セマフォ待ちとなる。次にステップ S 3 4 1 へ進み、タイマー管理 Task の終了セマフォを送信する。

10

【 0 2 5 9 】

20

図 6 8 は図 4 2 の回復系処理 Task 6 2 - 0 7 の詳細を示したフローチャートで、ステップ S 3 5 1 で初期化を行う。次にステップ S 3 5 2 で回復系処理 Task へのセマフォ待ちとなる。図 6 7 のステップ S 3 3 3 で回復系処理 Task へのセマフォが送信されるとステップ S 3 5 3 へ進み回復系処理へと進む。ステップ S 3 5 3 でキャッピング時間であるかどうかをチェックする。キャッピング時間であるかどうかのチェックは最後に印字した時間からの経過時間で行われ、キャッピング時間であればステップ S 3 5 4 へ進み、キャッピング処理を実行する。キャッピングは印字ヘッドが印字されずに放置され、インクが乾くことを防止するためにキャッピング位置にキャリッジを移動して密閉することでインクの乾燥を防ぐ処理である。印字が連続して行われる場合には、毎回キャッピングすると印字時間が延びるため、最後に印字を行った時点から一定の時間印字が行われなかった場合にキャッピングする。次にステップ S 3 6 1 へ進み、終了セマフォを送信する。ステップ S 3 5 3 でキャッピング時間でなければ、ステップ S 3 5 5 へ進みワイピング時間かどうかのチェックを行う。ワイピング時間かどうかのチェックは前回のワイピング時間からの経過時間で行われ、ワイピング時間であればステップ S 3 5 6 へ進みワイピング処理実行する。ワイピングは印字ヘッドに付着した余分なインクを拭き取る処理である。ワイピング処理は一定の印字時間ごとに行われ、キャリッジをワイピング位置へ移動し、ワイピングブレードで余分なインクをふき取り、余分なインクによる混色等を防ぐ。次にステップ S 3 6 1 へ進み、終了セマフォを送信する。ステップ S 3 5 5 でワイピング時間でなければ、ステップ S 3 5 7 へ進み予備吐時間かどうかのチェックを行う。予備吐時間かどうかのチェックは前回の予備吐時間からの経過時間で行われ、予備吐時間であればステップ S 3 5 8 へ進み予備吐処理実行する。予備吐は印字ヘッドが目詰まりする事を防止する処理である。予備吐処理は一定の印字時間ごとに行われ予備吐位置へキャリッジを移動し一定量のインクを吐出して、インクがヘッドで目詰まりすることを防ぐ。次にステップ S 3 6 1 へ進み、終了セマフォを送信する。ステップ S 3 5 7 で予備吐時間でなければ、ステップ S 3 5 9 へ進みクリーニング時間かどうかのチェックを行う。クリーニング時間かどうかのチェックは前回のクリーニング時間からの経過時間と印字起動の一回目などで行われ、クリーニング時間であればステップ S 3 6 0 へ進みクリーニング処理実行する。クリーニングは印字ヘッドが目詰まりする事を防止する予備吐処理と印字ヘッドに付着した余分なインクを拭き取るワイピング処理を合わせて行う処理である。クリーニング処理は予備吐処理とワイプ処理を行うことでヘッドのクリーニングが行え、ヘッドを最適な

30

40

50

状態に保つ。次にステップ S 3 6 1 へ進み、終了セマフォを送信する。

【 0 2 6 0 】

図 6 9 は図 4 2 のヘッド温調処理 T a s k 6 2 - 0 8 の詳細を示したフローチャートで、ステップ S 3 7 1 で初期化を行う。次にステップ S 3 7 2 でヘッド温調処理 T a s k へのセマフォ待ちとなる。図 6 7 のステップ S 3 3 5 でヘッド温調処理 T a s k へのセマフォが送信されるとステップ S 3 7 3 へ進み各ヘッド温調処理へと進む。ステップ S 3 7 3 で装着されているヘッドが印字ヘッドであるかスキャナヘッドであるかどうかをチェックする。ステップ S 3 7 3 でスキャナであれば、ステップ S 3 7 4 へ進みスキャナヘッドの温調処理を実行する。スキャナヘッドの温調処理はスキャナヘッドに搭載されている光源がいつも一定の光量以上で点灯できるようにスキャンヘッドの光源を前もって必要時間分点灯する処理である。次にステップ S 3 7 6 へ進み、終了セマフォを送信する。ステップ S 3 7 3 でスキャナヘッドでない場合、ステップ S 3 7 5 へ進み、印字ヘッド温調処理を行う。印字ヘッド温調処理は印字ヘッドが一定の状態インクを吐出出来るように印字ヘッドのヒータを前もって暖めたり、印字が連続すると印字ヘッドの温度が上昇し、ヘッドの温度変化に対応した印字ヘッドの温度調整を行う。次にステップ S 3 7 6 へ進み、終了セマフォを送信する。

10

【 0 2 6 1 】

図 7 0 は図 4 2 のエラー処理 T a s k 6 2 - 0 9 の詳細を示したフローチャートで、ステップ S 3 8 1 で初期化を行う。次にステップ S 3 8 2 でエラー処理 T a s k へのセマフォ待ちとなる。図 6 7 のステップ S 3 3 7 でエラー処理 T a s k へのセマフォが送信されるとステップ S 3 8 3 へ進みエラー処理へと進む。ステップ S 3 8 3 でエラーがあるかどうかをチェックする。ステップ S 3 8 3 でエラーであれば、ステップ S 3 8 4 へ進みエラー処理を実行する。エラー処理は各種プリンタのエラーを行う処理で、エラーにはモーター、ヘッド、紙関係などがあり、それらエラーに対する処理を行う。次にステップ S 3 8 5 へ進み、終了セマフォを送信する。

20

【 0 2 6 2 】

図 7 1 は図 4 2 の電源管理 T a s k 6 2 - 1 0 の詳細を示したフローチャートで、ステップ S 1 4 0 1 で初期化を行う。次にステップ S 3 9 2 で電源管理 T a s k へのセマフォ待ちとなる。図 6 7 のステップ S 3 3 9 で電源管理 T a s k へのセマフォが送信されるとステップ S 3 9 3 へ進み電源管理処理へと進む。ステップ S 3 9 3 で電源変動があるかどうかをチェックする。ステップ S 3 9 3 で電源変動であれば、ステップ S 3 9 4 へ進み電源管理処理を実行する。電源管理処理はプリンタの電源に関する処理で、電圧の低下など、それら電源変動に対する処理を行う。次にステップ S 3 9 5 へ進み、終了セマフォを送信する。

30

【 0 2 6 3 】

以上、図 6 7 から図 7 1 でタイマー管理 T a s k について説明したが、一定の時間ごとにタイマー管理 T a s k が起動され、タイマー管理 T a s k により時間の経過に従って行う処理が実行される。

【 0 2 6 4 】

(実施の形態 2)

40

図 7 8 は本発明の実施の形態の処理を示すブロック図である。

【 0 2 6 5 】

図中 9 0 - 0 1 はウィンドウズ・アプリケーションであり、9 0 - 0 2 はウィンドウズ・オペレーティング・システムにおけるプリンティングサブシステムであり、9 0 - 0 3 はウィンドウズ・オペレーティング・システムにおけるスプーリングサブシステムであり、通常のウィンドウズのプリントシーケンスに従い、印刷用データ(コントロールコード)を生成しスプールファイルに蓄える。

【 0 2 6 6 】

9 0 - 0 5 は W - R T O S ホスト・アプリケーションである。W - R T O S ホスト・アプリケーション 9 0 - 0 5 は、R T O S と通信する機能を有する。通常のウィンドウズ・ア

50

アプリケーションである。

【0267】

90-06はRTOSタスクであり、90-07はDCPE Portである。RTOSタスク90-06は、W-RTOSホスト・アプリケーション90-05により起動され、発明の実施の形態1RTOSタスクと同様に印刷用データ(コントロールコード)をプリンタヘッドイメージ・プリンタ制御I/Oコマンドに変換し、DCPE Port 90-07に書き出す。

【0268】

90-08はウィンドウズ・オペレーティング・システムにおけるVCOMMであり、90-09はウィンドウズ・オペレーティング・システムにおけるVCOMMポートドライバである。VCOMM 90-08は通常のウィンドウズのプリントシーケンスに従い、スプーリングサブシステム90-03より印刷用データ(コントロールコード)を受け取り、VCOMMポートドライバ90-09にこれを引き渡す。

【0269】

90-10はRTOSドライバであり、RTOSの提供するストリーム入出力(SIO)を介してRTOSタスク90-06に印刷用データ(コントロールコード)を引き渡す。RTOSタスク及びRTOSドライバはウィンドウズ環境下で動作するリアルタイムオペレーティングシステムRTOSの下で動作しており、CPU動作モードRing 0で動作している。

【0270】

VCOMM及びVCOMMポートドライバはともにウィンドウズ・オペレーティング・システムにおけるVxDであり、CPU動作モードRing 0で動作し、構造体DDbによりシステムに認識される。構造体DDbには、デバイス名、サービス・エントリーポイントなどが含まれる。VCOMMポートドライバは、VCOMMに対してポートドライバファンクションと呼ばれる関数群を提供し、これらは概ねVCOMMのサービスに対応している。スプーリングサブシステムはウィンドウズ・システムAPIを介してVCOMMのサービスを呼び出し、VCOMMはサービスに対応するポートドライバファンクションを呼び出して印刷用データ(コントロールコード)を出力する。印刷用データ(コントロールコード)の出力に関係する主なポートドライバファンクションは、Port Open(ポートのオープン)、Port Close(ポートのクローズ)、Port Setup(送受信キューのセットアップ)、Port Write(出力キューへの書込み)、Port Get Queue Status(キューの状態の引出し)、などである。通常のウィンドウズのプリントシーケンスにおいては、VCOMMポートドライバは、印刷用データ(コントロールコード)をプリンタが接続された物理ポートに書き出す。本発明の実施の形態においては、VCOMMポートドライバ90-09は、印刷用データ(コントロールコード)をRTOSドライバ90-10に引き渡す。

【0271】

図79はRTOSのストリーム入出力の働きを示すブロック図で、本発明の実施の形態で用いるストリーム入力について説明する。図中91-01はRTOSタスク、91-02はSIO、91-03はRTOSドライバ、91-04は空バッファ、91-05は入力データが格納されたバッファである。RTOSタスク91-01はSIO 91-02の入力ファンクションSIO__get()を呼び出し、空バッファをSIO 91-02に渡す。SIO 91-02は渡された空バッファを空バッファキューに入れてRTOSドライバ91-03の入力ファンクションDxx__input()を呼び出す。RTOSドライバ91-03は空バッファキューから空バッファを取り出し、このバッファに入力データを格納する。バッファが入力データで満たされるとバッファを入力データバッファキューに入れる。SIO 91-02はDxx__input()から戻るとSIO__get()を終了し入力データバッファキューから入力データが格納されたバッファをRTOSタスク91-01に返す。以下、フローチャートを用いてRTOSドライバ91-03の処理を説明する。

10

20

30

40

50

【 0 2 7 2 】

図 8 0 は、`D x x _ i n p u t ()` のフローチャートである。ステップ S 4 0 0 で入力ジョブを開始する。入力ジョブは非同期に起動され処理をするので、RTOS ドライバ 9 1 - 0 3 は同期のためのセマフォを持つ。ステップ S 4 0 1 でこのセマフォがポストされるのを待つ。ポストされると処理を終え戻る。

【 0 2 7 3 】

図 8 1 は、入力ジョブのフローチャートである。ステップ S 4 1 0 でデバイスから入力しバッファにデータを格納する。本発明の実施の形態においては前述の通り V C O M M ポートドライバ 9 0 - 0 9 からデータを受け取ることになる。ステップ S 4 1 1 でバッファがフルになったかどうかを判定し、フルでなければ処理を終える。バッファがフルになると、ステップ S 4 1 2 でバッファを入力データバッファキューに入れて、ステップ S 4 1 3 でセマフォをポストする。

10

【 0 2 7 4 】

以下、V C O M M ポートドライバ 9 0 - 0 9 から R T O S ドライバ 9 0 - 1 0 へのデータの引き渡しについて説明する。

【 0 2 7 5 】

R T O S ドライバ 9 0 - 1 0 は R T O S の提供するファンクション `I A _ a d d D D B ()` により、自身の持つ構造体 D D B をウィンドウズ・オペレーティングシステムの V x D 層の D D B チェーンに加えることによりエントリーポイントをエクスポートする。V C O M M ポートドライバ 9 0 - 0 9 はウィンドウズの提供するサービス `g e t D D B ()` により、R T O S ドライバ 9 0 - 1 0 の D D B を検索し、エントリーポイントを取得する。V C O M M ポートドライバ 9 0 - 0 9 は、前述したポートドライバファンクション `P o r t W r i t e` (出力キューへの書込み) が呼び出されると前記エントリーポイントにより R T O S ドライバ 9 0 - 1 0 を呼び出して V C O M M から渡された印刷用データ (コントロールコード) を R T O S ドライバ 9 0 - 1 0 に引き渡す。R T O S ドライバ 9 0 - 1 0 はこれを入力データとして、前記入力ジョブを行う。

20

【 0 2 7 6 】

(実施の形態 3)

図 8 2 は本発明の実施の形態の処理を示すブロック図である。

【 0 2 7 7 】

図中 9 4 - 0 1 はウィンドウズ・アプリケーションであり、9 4 - 0 2 はウィンドウズ・オペレーティング・システムにおけるプリンティングサブシステム・スプーリングサブシステムであり、9 4 - 0 3 はウィンドウズ・オペレーティング・システムにおけるプリンタドライバであり、9 4 - 0 4 はデータ変換タスクであり、9 4 - 0 5 はウィンドウズ・オペレーティング・システムにおける V C O M M であり、9 4 - 0 6 はウィンドウズ・オペレーティング・システムにおける V C O M M ポートドライバであり、9 4 - 0 7 は D C P E P o r t である。本発明の実施の形態においても通常のウィンドウズのプリントシーケンスに従い、印刷用データ (コントロールコード) を生成しスプールファイルに蓄える。ただし、プリンタドライバ 9 4 - 0 3 は発明の実施の形態 1 R T O S タスクと同様に処理するデータ変換タスク 9 4 - 0 4 と通信してこれを印刷用データ (コントロールコード) をプリンタヘッドイメージ・プリンタ制御 I / O コマンドに変換し、これを V C O M M 、V C O M M ポートドライバを介して、D C P E P o r t に出力する。

30

40

【 0 2 7 8 】

(実施の形態 4)

第 1 の発明の実施の形態では、ホスト部とプリンタ部の間のインターフェイスを I S A バスを介して接続した構成についての説明をしたが、インターフェイスの構造についてはこれに限定されるものではなく、第 3 の形態として I S A バスの代わりに汎用の高速インターフェイスとして I E E E 1 3 9 4 を用いた実施の形態について説明する。

【 0 2 7 9 】

まず初めに I E E E 1 3 9 4 の技術の概要についての説明を行うと、家庭用デジタル V T

50

RやDVDの登場も伴って、ビデオデータやオーディオデータなどのリアルタイムでかつ高情報量のデータ転送のサポートが必要になっている。こういったビデオデータやオーディオデータをリアルタイムで転送し、パソコン（PC）に取り込んだり、またはその他のデジタル機器に転送を行うには、必要な転送機能を備えた高速データ転送可能なインタフェースが必要になってくるものであり、そういった観点から開発されたインタフェースがIEEE 1394 - 1995（以下1394シリアルバス）である。

【0280】

図83に1394シリアルバスを用いて構成されるネットワーク・システムの例を示す。このシステムは機器A、B、C、D、E、F、G、Hを備えており、A - B間、A - C間、B - D間、D - E間、C - F間、C - G間、及びC - H間をそれぞれ1394シリアルバスのツイスト・ペア・ケーブルで接続されている。この機器A～Hは例としてPC、デジタルVTR、DVD、デジタルカメラ、ハードディスク、モニタ等である。各機器間の接続方式は、ディジーチェーン方式とノード分岐方式とを混在可能としたものであり、自由度の高い接続が可能である。

10

【0281】

また、各機器は各自固有のIDを有し、それぞれが認識し合うことによって1394シリアルバスで接続された範囲において、1つのネットワークを構成している。各デジタル機器間をそれぞれ1本の1394シリアルバスケーブルで順次接続するだけで、それぞれの機器が中継の役割を行い、全体として1つのネットワークを構成するものである。また、1394シリアルバスの特徴でもある、Plug & Play機能でケーブルを機器に接続した時点で自動で機器の認識や接続状況などを認識する機能を有している。

20

【0282】

また、図83に示したようなシステムにおいて、ネットワークからある機器が削除されたり、または新たに追加されたときなど、自動的にバスリセットを行い、それまでのネットワーク構成をリセットしてから、新たなネットワークの再構築を行う。この機能によって、その時々ネットワークの構成を常時設定、認識することができる。

【0283】

またデータ転送速度は、100 / 200 / 400 Mbpsと備えており、上位の転送速度を持つ機器が下位の転送速度をサポートし、互換をとるようになっている。

【0284】

データ転送モードとしては、コントロール信号などの非同期データ（Asynchronousデータ：以下Asyncデータ）を転送するAsynchronous転送モード、リアルタイムなビデオデータやオーディオデータ等の同期データ（Isochronousデータ：以下Isoデータ）を転送するIsochronous転送モードがある。このAsyncデータとIsoデータは各サイクル（通常1サイクル125μs）の中において、サイクル開始を示すサイクル・スタート・パケット（CSP）の転送に続き、Isoデータの転送を優先しつつサイクル内で混在して転送される。

30

【0285】

次に、図84に1394シリアルバスの構成要素を示す。1394シリアルバスは全体としてレイヤ（階層）構造で構成されている。図3に示したように、最もハード的なのが1394シリアルバスのケーブルであり、そのケーブルのコネクタが接続されるコネクタポートがあり、その上にハードウェアとしてフィジカル・レイヤとリンク・レイヤがある。

40

【0286】

ハードウェア部はインターフェイスチップの部分であり、そのうちフィジカル・レイヤは符号化やコネクタ関連の制御等を行い、リンク・レイヤはパケット転送やサイクルタイムの制御等を行う。

【0287】

ファームウェア部のトランザクション・レイヤは、転送（トランザクション）すべきデータの管理を行い、ReadやWriteといった命令を出す。マネージメント・レイヤは、接続されている各機器の接続状況やIDの管理を行い、ネットワークの構成を管理する

50

部分である。このハードウェアとファームウェアまでが実質上の 1 3 9 4 シリアルバスの構成である。

【 0 2 8 8 】

またソフトウェア部のアプリケーション・レイヤは使うソフトによって異なり、インタフェース上にどのようにデータをのせるか規定する部分であり、A V プロトコルなどのプロトコルによって規定されている。以上が 1 3 9 4 シリアルバスの構成である。

【 0 2 8 9 】

次に、図 8 5 に 1 3 9 4 シリアルバスにおけるアドレス空間の図を示す。

【 0 2 9 0 】

1 3 9 4 シリアルバスに接続された各機器（ノード）には必ず各ノード固有の、6 4 ビットアドレスを持たせておく。そしてこのアドレスを R O M に格納しておくことで、自分や相手のノードアドレスを常時認識では相手を指定した通信も行える。1 3 9 4 シリアルバスのアドレッシングは、I E E E 1 2 1 2 規格に準じた方式であり、アドレス設定は、最初の 1 0 b i t がバスの番号の指定用に、次の 6 b i t がノード I D 番号の指定用に使われる。それぞれの機器内で使用できる 4 8 ビットのアドレスについても 2 0 ビットと 2 8 ビットに分けられ、2 5 6 M バイト単位の構造を持って利用される。

10

【 0 2 9 1 】

最初の 2 0 ビットの 0 ~ 0 x F F F F D の部分はメモリ空間と呼ばれる。0 x F F F F E の部分はプライベート空間と呼ばれ、機器内で自由に利用できるアドレスである。0 x F F F F F の部分はレジスタ空間と呼ばれ、バスに接続された機器間で共通な情報が置かれ、各機器間のコミュニケーションに使われる。レジスタ空間の最初の 5 1 2 バイトには、C S R アーキテクチャのコアになるレジスタ（C S R コア）がある。次の 5 1 2 バイトにはシリアルバスのレジスタがある。その次の 1 0 2 4 バイトには C o n f i g u r a t i o n R O M が置かれる。残りはユニット空間で機器固有のレジスタがある。以上が 1 3 9 4 シリアルバスの技術の概要である。

20

【 0 2 9 2 】

図 8 6 は本発明の第 3 の実施の形態として、ホスト部とプリンタ部のインターフェイスを 1 3 9 4 シリアルバスを使って接続したシステムのブロック図である。ホスト部とプリンタ部の主要構成については、図 2 の概略ブロック図と同等であるので詳しい説明は省略するが、相違点としてはホスト部とプリンタ部の各々に 1 3 9 4 I F ブロックが付加されていて、この 1 3 9 4 I F を介した 1 3 9 4 シリアルバスケーブルによってお互いのデータのやり取りが行われる。1 3 9 4 I F は、1 3 9 4 シリアルバスで転送されるシリアルデータを各機器の内部バスと接続するために、シリアル - パラレル変換やデータパケットの構成等を行うブロックである。

30

【 0 2 9 3 】

すなわち本発明の実施の形態におけるプリンタ部は、ホスト部に対して 1 3 9 4 シリアルバスでつながれた形態になっていて、1 3 9 4 シリアルバス上のレジスタ空間を介してホスト部とのデータのやり取りを行うことができる。

【 0 2 9 4 】

図 8 7 は本発明の実施の形態におけるプリンタ部の内部構成を示すブロック図である。主要ロジックについては図 7 のブロックと同等であるが、ステートマシンコントローラにつながっていた I S A - b u s の部分が、1 3 9 4 I F からの I F - b u s に変わっている点が特徴である。1 3 9 4 シリアルバスから送られてきたデータは、1 3 9 4 I F によってプリンタ部に対してレジスタアクセスを行うよう I F - b u s が出力され、ホスト部からはあたかもホスト部の C P U 2 1 がプリンタ部に I O アクセスを行っているような環境を提供することができる。

40

【 0 2 9 5 】

図 8 8 はステートマシンコントローラの内部構成を示すブロック図である。

【 0 2 9 6 】

ステートマシンコントローラは 1 3 9 4 I F からの I F - b u s を経由してホスト部との

50

データのやり取りを行い、プリンタユニットの制御を行うブロックである。

【0297】

基本的な動作については図8のブロック図と同等であるのが、相違点となるのはISA-busによってCPU21が直接アクセスしていたI/Oレジスタが、1394シリアルバスを介して間接的にアクセスが行われるようになった点である。

【0298】

ここで基本動作を説明すると、スーパーバイザコマンドレジスタはホスト部からプリンタI/OコントローラへのI/OアクセスをPRT-busを介して行ったり、制御I/Oコマンドの起動・停止コマンドの発行を行うためのレジスタで、スーパーバイザコマンド・アドレスレジスタ(SVA/I/O)で設定したPRT-busのアドレス(PA)に対してスーパーバイザコマンド・データレジスタ(SVD/I/O)を使ってデータ(PD)の読み書きを行うことを行う。

10

【0299】

制御I/Oコマンド・データレジスタ(BAT/I/O)は、ホスト部からプリンタ部メモリをMEM-busを介して直接アクセスするための機能であり、「アドレス+データ」のセットでデータ書き込み動作を行うとメモリアドレスカウンタのライトアドレスに対応したプリンタ部メモリのアドレス(MA)に2バイトのデータ(MD)が書き込まれ、書き込まれた後にはライトアドレスが毎回+2インクリメントされる。

【0300】

制御I/Oコマンド・データレジスタを使ったデータの読み出し動作も同様で、リードアドレスに対応したプリンタ部メモリのデータが2バイト読み出され、読み出し動作後には毎回リードアドレスが+2インクリメントされる。

20

【0301】

制御I/Oコマンド・データレジスタによってメモリに書き込まれたデータは制御I/Oコマンドを実行するために使われるデータであり、スーパーバイザコマンドレジスタによる制御I/Oコマンドの起動コマンドとしてBP-startに“1”を書き込む事により、プリンタ部メモリから制御I/Oアドレスカウンタのリードアドレスで示されたデータを2バイトづつ読み出し、読み出したデータの下位バイトがアドレス・上位バイトがデータを表しているので、PRT-busに読み出したアドレスとデータを出力してプリンタI/Oコントローラのレジスタへのデータライトアクセス動作を行うことで制御I/Oコマンドの実行を行う。

30

【0302】

この制御I/Oコマンドの実行により読み出されたリードアドレスは+2づつ毎回インクリメントされ、制御I/Oコマンドの停止コマンド(BP-end)が発行されてBP-startが“0”になるまで、制御I/Oコマンド実行動作が繰り返される。

【0303】

リフレッシュコントロールは、D-RAMのリフレッシュ動作をつかさどる制御ブロックで、8msの間に512回のリフレッシュアクセスを行うようになっていて、その時のアドレスを毎回インクリメントさせるのがリフレッシュアドレスカウンタである。

【0304】

メモリアドレスコントロールは、プリンタ部メモリに対して各アクセス要求に優先順位をつけて処理を行うブロックで、制御I/Oコマンドの実行が第一優先・リフレッシュが第二優先・ホスト部からのメモリアccessが第三優先となっている。これは制御I/Oコマンドの実行時間が変動するのを防ぐための優先順序となっている。以上がステートマシンコントローラの動作説明である。

40

【0305】

この様にホスト部とプリンタ部の接続の形態は一つのものに限定されることはなく汎用のインターフェイスに置きかえることが可能であるので、その際の制御方法については常に共通の環境を用いることができ、更にシステムのハードウェアの構成は状況に合わせて最適なものを使うことが出来るという点が大きな特徴である。

50

【 0 3 0 6 】

【 発明の効果 】

ホストベースプリンティングシステムの簡素化されコストダウンされたプリンタ装置を提供することが出来る。

【 0 3 0 7 】

ホストベースプリンティングシステムのホストでのプリンタ制御方式を提供することが出来る。

【 0 3 0 8 】

ホストベーススキャンニングシステムの簡素化されコストダウンされたスキャナ装置を提供することが出来る。

10

【 0 3 0 9 】

ホストベーススキャンニングシステムのホストでのスキャナ制御方式を提供することが出来る。

【 0 3 1 0 】

既存のプリンティングシステム、プリンタドライバに変更を加えないホストベースプリンティングシステムを提供することが出来る。

【 図面の簡単な説明 】

【 図 1 】 本発明を実施したパソコンを示す斜視図である。

【 図 2 】 パソコンの概略ブロック図である。

【 図 3 】 ホスト部のメモリアドレスマップである。

20

【 図 4 】 ホスト部の I / O アドレスマップである。

【 図 5 】 S E T / I O レジスタの構成図である。

【 図 6 】 P R T / I O レジスタの構成図である。

【 図 7 】 プリンタ制御部のブロック図である。

【 図 8 】 ステートマシンコントローラのブロック図である。

【 図 9 】 ステートマシンコントローラ内のレジスタマップである。

【 図 1 0 】 制御 I / O コマンド実行時のタイミング図である。

【 図 1 1 】 プリンタユニットを表してる図である。

【 図 1 2 】 プリンタ制御装置のブロック図である。

【 図 1 3 】 カラー印字用記録ヘッドの内部構成図である。

30

【 図 1 4 】 記録ヘッド制御用タイミング図である。

【 図 1 5 】 キャリッジモーター制御回路を表す図である。

【 図 1 6 】 キャリッジモーター制御用タイミング図である。

【 図 1 7 】 プリンタ I O コントローラ I C のブロック図である。

【 図 1 8 】 プリンタ I O コントローラ I C のレジスタマップである。

【 図 1 9 】 プリンタ I O コントローラ I C の制御 I / O コマンド実行タイミング制御回路のタイミング図である。

【 図 2 0 】 従来例のプリンタ制御部のブロック図である。

【 図 2 1 】 ウェイトコマンド実行時のタイミング図である。

【 図 2 2 】 プリンタ I O コントローラ I C の制御 I / O コマンド実行タイミング制御回路の構成図である。

40

【 図 2 3 】 プリンタ初期化フローチャートを表す図である。

【 図 2 4 】 キャリッジ初期化動作時の自起動領域等速動作制御 I / O コマンド群を表す図である。

【 図 2 5 】 片方向印字動作時（加速部分）の制御 I / O コマンド群を表す図である。

【 図 2 6 】 片方向印字動作時（等速部分）の制御 I / O コマンド群を表す図である。

【 図 2 7 】 片方向印字動作時（減速部分）の制御 I / O コマンド群を表す図である。

【 図 2 8 】 紙送り動作時（加速部分）の制御 I / O コマンド群を表す図である。

【 図 2 9 】 紙送り動作時（減速部分）の制御 I / O コマンド群を表す図である。

【 図 3 0 】 メディアの構成を示した図である。

50

【図 3 1】従来例のウィンドウズ・アプリケーションからプリンタへ出力する場合の、ソフトウェア処理手順を示す図である。

【図 3 2】印刷処理のブロック図である。

【図 3 3】Language Monitor の Start Doc Port () のフローチャートである。

【図 3 4】Language Monitor の Write Port () のフローチャートである。

【図 3 5】Language Monitor の End Doc Port () のフローチャートである。

【図 3 6】W - R T O S ホスト・アプリケーション中の、初期化関数のフローチャートである。 10

【図 3 7】W - R T O S ホスト・アプリケーション中の、非同期通信関数のフローチャートである。

【図 3 8】スキャナ処理の流れを示すソフトウェアブロック図である。

【図 3 9】従来例の R T O S タスクとウィンドウズ・アプリケーションの通信、また R T O S タスク間の通信を示す図である。

【図 4 0】タスク制御を示した図である。

【図 4 1】タスク遷移状態を示した図である。

【図 4 2】タスクの構成を示した図である。

【図 4 3】コントロールコードの基本構成を示した図である。 20

【図 4 4】コントロールコード一覧を示した図である。

【図 4 5】イメージコントロールコードの順序を示した図である。

【図 4 6】印字ヘッドのドット構成を示した図である。

【図 4 7】バッファ構成を示した図である。

【図 4 8】バッファの構成を示した図である。

【図 4 9】バッファの構成を示した図である。

【図 5 0】バッファの構成を示した図である。

【図 5 1】バッファの構成を示した図である。

【図 5 2】バッファの構成を示した図である。

【図 5 3】バッファの構成を示した図である。 30

【図 5 4】印字順序を示した図である。

【図 5 5】バッファへのイメージデータをセットする順序を示した図である。

【図 5 6】バッファからのイメージデータをゲットする順序を示した図である。

【図 5 7】ポインタ、カウンタを示した図である。

【図 5 8】コントロールコード解析 T a s k を示したフローチャートである。

【図 5 9】印字処理を示したフローチャートである。

【図 6 0】イメージ展開処理を示したフローチャートである。

【図 6 1】印字制御 I / O コマンド作成処理を示したフローチャートである。

【図 6 2】制御 I / O コマンド転送処理を示したフローチャートである。

【図 6 3】フィード処理を示したフローチャートである。 40

【図 6 4】給・排紙処理を示したフローチャートである。

【図 6 5】スキャン処理 T a s k を示したフローチャートである。

【図 6 6】スキャン処理を示したフローチャートである。

【図 6 7】タイマー管理 T a s k を示したフローチャートである。

【図 6 8】回復系処理 T a s k を示したフローチャートである。

【図 6 9】ヘッド温調処理 T a s k を示したフローチャートである。

【図 7 0】エラー処理 T a s k を示したフローチャートである。

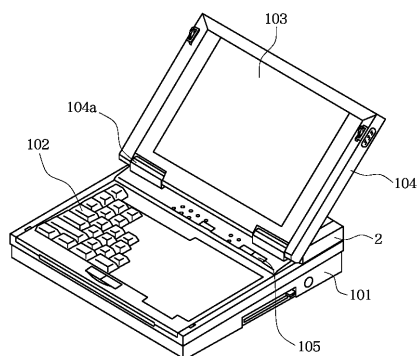
【図 7 1】電源管理 T a s k を示したフローチャートである。

【図 7 2】印字の制御 I / O コマンドを受信したバッファの内容を示した図である。

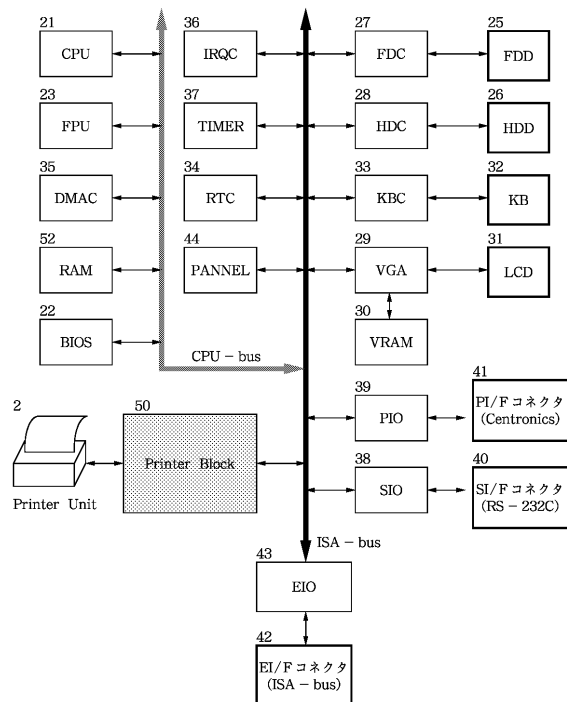
【図 7 3】印字制御 I / O コマンドを受信したバッファの内容を示した図である。 50

- 【図 7 4】印字制御 I / O コマンドを受信したバッファの内容を示した図である。
- 【図 7 5】スキャンの制御 I / O コマンドを受信したバッファの内容を示した図である。
- 【図 7 6】スキャン制御の I / O コマンドを受信したバッファの内容を示した図である。
- 【図 7 7】スキャン制御の I / O コマンドを受信したバッファの内容を示した図である。
- 【図 7 8】発明の実施の形態 2 の処理を示すブロック図である。
- 【図 7 9】RTOS ストリーム入出力のブロック図である。
- 【図 8 0】RTOS ドライバ 9 1 - 0 3 の入力ファンクションのフローチャートである。
- 【図 8 1】RTOS ドライバ 9 1 - 0 3 の入力ジョブのフローチャートである。
- 【図 8 2】発明の実施の形態 3 の処理を示すブロック図である。
- 【図 8 3】IEEE 1394 シリアルバスを用いて構成されるネットワークシステムを示す図である。 10
- 【図 8 4】IEEE 1394 シリアルバスの構成要素を示す図である。
- 【図 8 5】IEEE 1394 シリアルバスにおけるアドレス空間を示す図である。
- 【図 8 6】IEEE 1394 シリアルバスを用いたシステムのブロック図である。
- 【図 8 7】プリンタ部の内部構成を示すブロック図である。
- 【図 8 8】ステートマシンコントローラの内部構成を示すブロック図である。

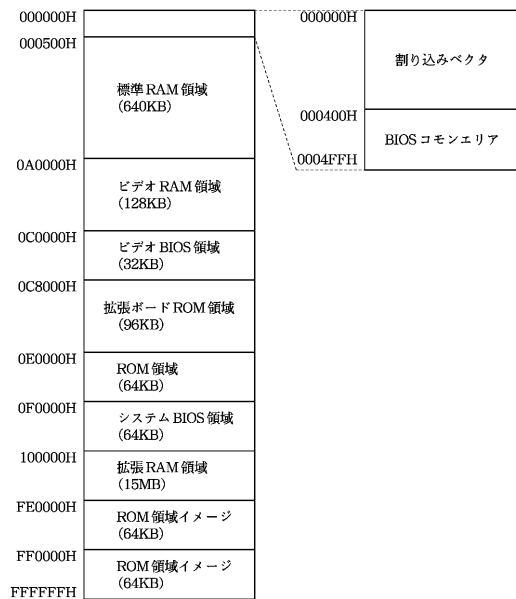
【図 1】



【図 2】



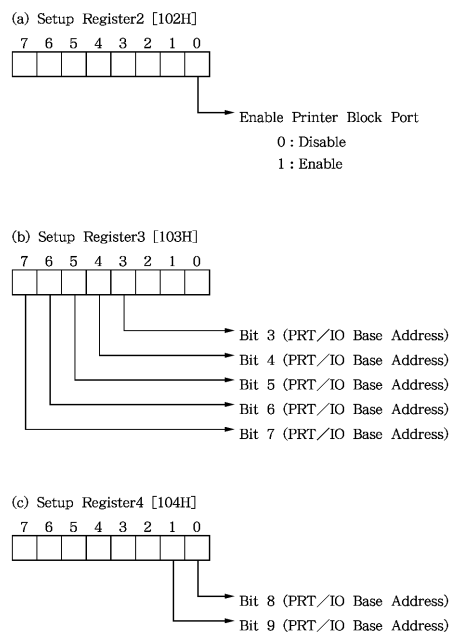
【 図 3 】



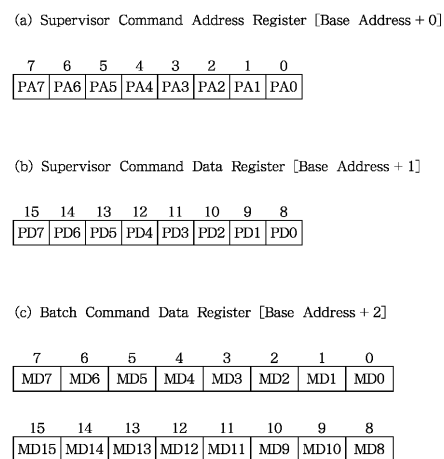
【 図 4 】

アドレス	デバイス
0000H-001FH	DMA Controller
0020H-0021H	Interrupt Controller
0040H-0043H	Timer
0060H-0064H	Keyboard
0070H-0071H	Real Time Clock
0080H-008FH	DMA Bank Reg.
00A0H-00A1H	DMA Controller
00C0H-00DFH	Interrupt Controller
00F0H-00FFH	FPU
0100H-0107H	Setup Reg. [SET/IO]
01F0H-01FFH	HDD Controller
0278H-027FH	Parallel Port3[P10/I03]
02F8H-02FFH	Serial Port2
0378H-037FH	Parallel Port2[P10/I02]
03B0H-03BBH	Video Controller
03BCH-03BFH	Parallel Port1[P10/I01]
03C0H-03DFH	Video Controller
03F0H-03F7H	FDD Controller
03F8H-03FFH	Serial Port1
0XXXH-0XXXH	Printer Block Port[PRT/I0]

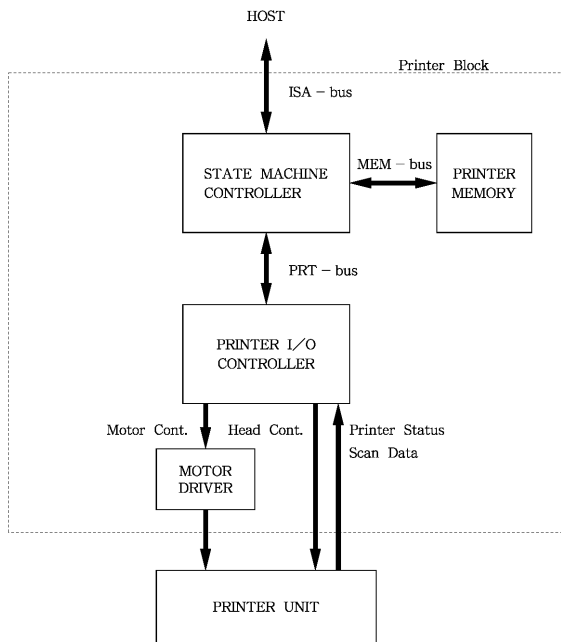
【 図 5 】



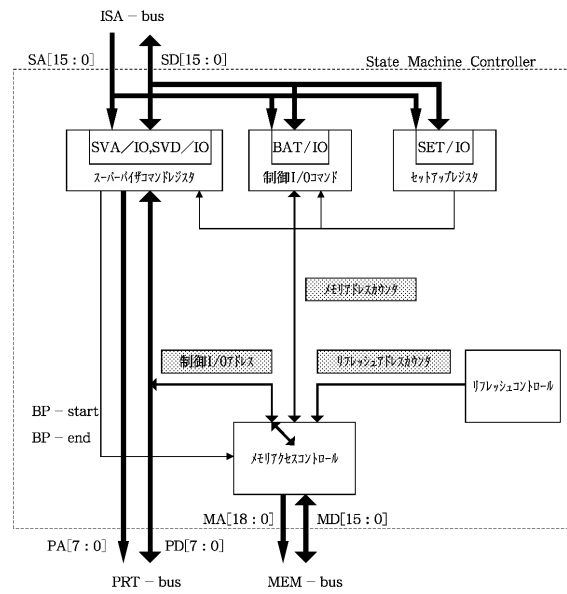
【 図 6 】



【図 7】



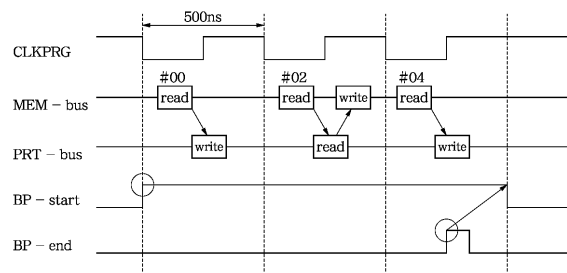
【図 8】



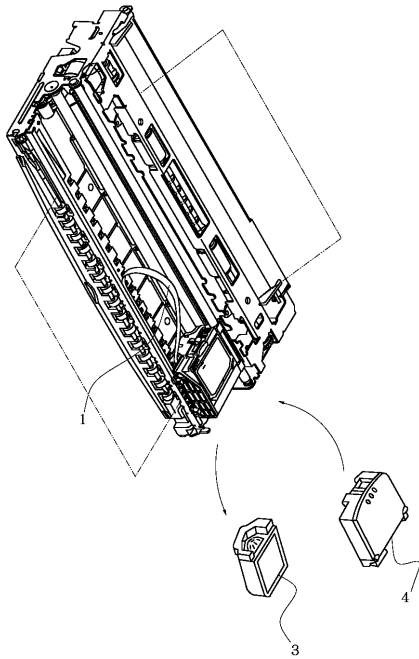
【図 9】

アドレス	記号	名称	
		Upper	Lower
**01	**00 CRES	BPSTART	制御I/Oコマンドレジスタ
**03	**02		
**05	**04		
**07	**06		
**09	**08 MVALH	R/W	R/W
**0B	**0A MVALH	R/W	R/W
**0D	**0C MVALH	R/W	R/W
**0F	**0E MVALH	R/W	R/W
**11	**10	BPEND	制御I/Oコマンドレジスタ
**13	**12 RWA	R/W	R/W
**15	**14 WAITL	R/W	R/W
**17	**16		
**19	**18 BVALH	R/W	R/W
**1B	**1A BVALH	R/W	R/W
**1D	**1C BVALH	R/W	R/W
**1F	**1E BVALH	R/W	R/W

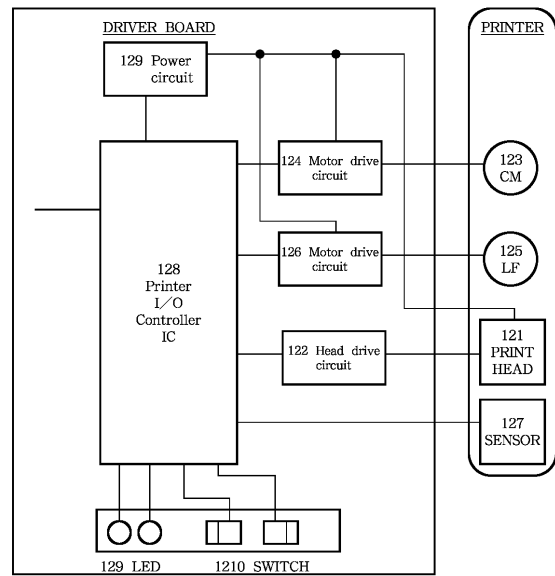
【図 10】



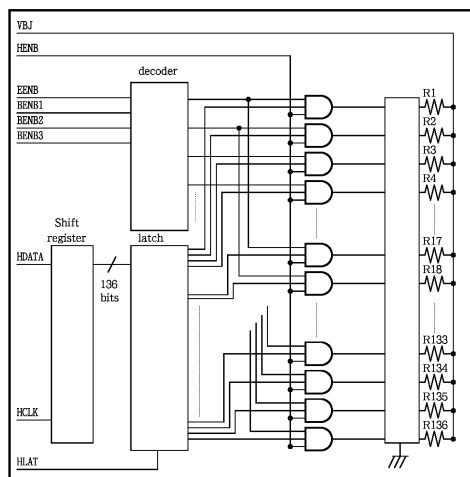
【図 1 1】



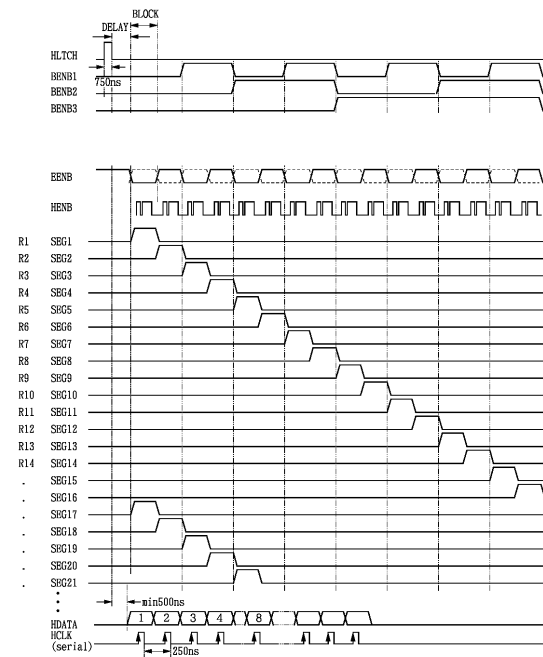
【図 1 2】



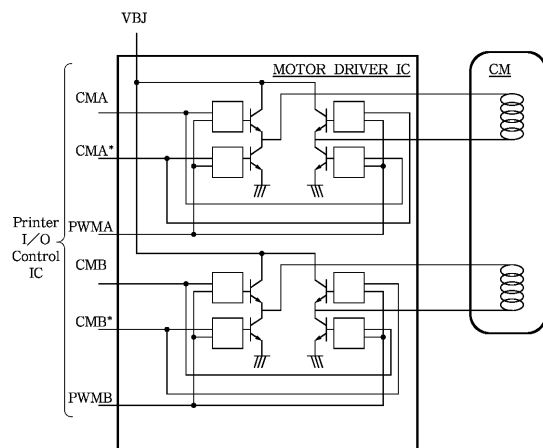
【図 1 3】



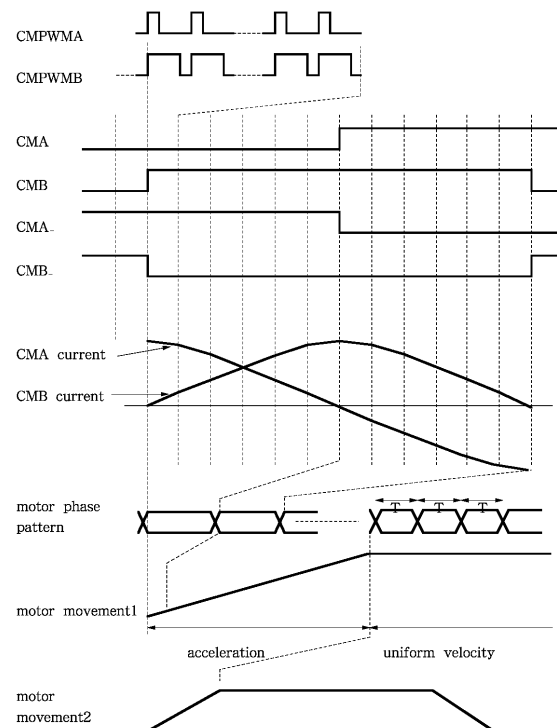
【図 1 4】



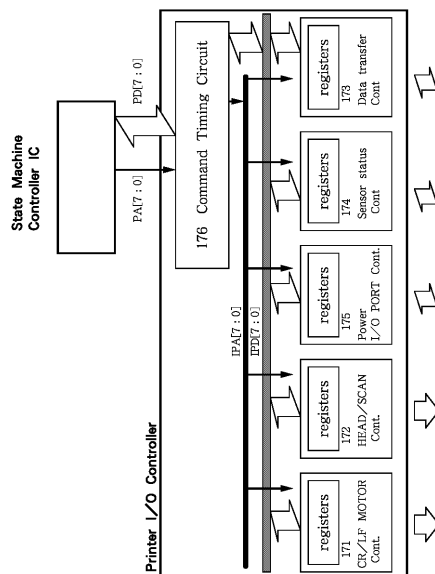
【図 15】



【図 16】



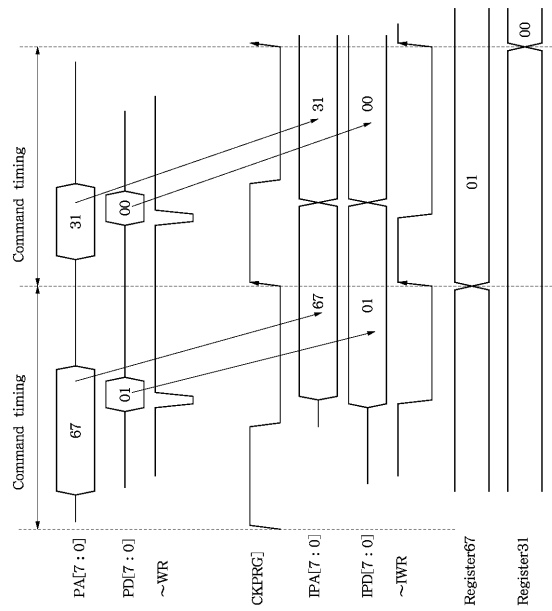
【図 17】



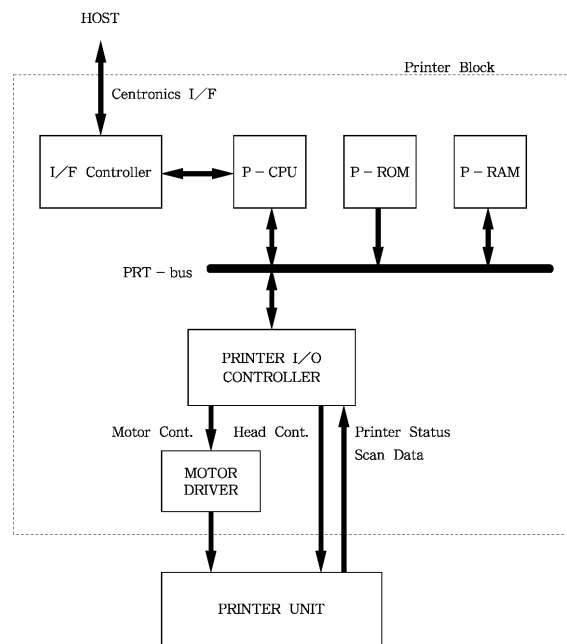
【図 18】

アドレス		記号		R/W		名称	
U	L	Upper	Lower	U	L	Upper	Lower
**21	**20	PORTB	PORTA	R/W	R/W	汎用ポートB	汎用ポートA
**23	**22	SUBH		R/W		サブヒートレジスタ	
**25	**24	PNCPRTB	PSTATUS	R/W	R	ポートBファンクションレジスタ	センサステータスポート
**27	**26	ADONCVNT	POWER	R/W	R/W	A/Dコンバータコントロールレジスタ	電源制御ポート
**29	**28	SWITCH	LED	R	R/W	スイッチレジスタ	LEDレジスタ
**2B	**2A						
**2D	**2C		ADDATA		R		A/Dデータレジスタ
**2F	**2E						
**31	**30	HDRUN		R/W		ヒート・スキャン駆動トリガ	
**33	**32		HDR		W		ヒートデータレジスタ
**35	**34	HCR1	HCR0	R/W	R/W	ヒートコントロールレジスタ1	ヒートコントロールレジスタ0
**37	**36	HCR3	HCR2	R/W	R/W	ヒートコントロールレジスタ3	ヒートコントロールレジスタ2
**39	**38		HCR4		R/W		ヒートコントロールレジスタ4
**3B	**3A	SDRH	SDRL	R	R	スキャンデータレジスタ (H)	スキャンデータレジスタ (L)
**3D	**3C	SCR1	SCRO	R/W	R/W	スキャンコントロールレジスタ1	スキャンコントロールレジスタ0
**3F	**3E		SCCLR		W		スキャンクリアレジスタ
**41	**40	BOTH	BDTL	R/W	R/W	ブロックデータレジスタ (H)	ブロックデータレジスタ (L)
**43	**42	BLTH	BLTL	R/W	R/W	ブロックデータレジスタ (H)	ブロックデータレジスタ (L)
**45	**44	MHT	SUT	R/W	R/W	メインヒートタイム	セッティングタイム
**47	**46	IVT	PHT	R/W	R/W	インターバルタイム	プリヒートタイム
**49	**48	CMHT	CSUT	R/W	R/W	COLメインヒートタイム	COLセッティングタイム
**4B	**4A	CIVT	CPHT	R/W	R/W	COLインターバルタイム	COLプリヒートタイム
**4D	**4C						
**4F	**4E						
**61	**60	CMRUN		R/W		CM PWMトリガ	
**63	**62	CMASF	CMCTL1	R/W	R/W	CM/ASPセレクトレジスタ	CMコントロール
**65	**64	TCMCB	TCMCA	R/W	R/W	CM/CM PWM時間	CM/CM PWM時間
**67	**66	CMSHFT	CMPAT	R/W	R/W	CM/CM PWMトリガ	CMフェーズ切り替え
**69	**68	SELTCM		R/W		PWM/相セレクトレジスタ	
**6B	**6A	SFTCMDIR		R/W		CM/CM PWM方向	
**6D	**6C						
**6F	**6E						
**71	**70	LPRUN		R/W		LFPWMトリガ	
**73	**72		LFCTL1		R/W		LFコントロール
**75	**74	TLFCB	TLFCA	R/W	R/W	LF/CM PWM時間	LF/CM PWM時間
**77	**76	LFSHFT	LFPAT	R/W	R/W	LFトリガ	LFフェーズ切り替え
**79	**78	SELTLF		R/W		PWM/相セレクトレジスタ	
**7B	**7A	SFTLFDIR		R/W		LFトリガ方向	
**7D	**7C						
**7F	**7E						

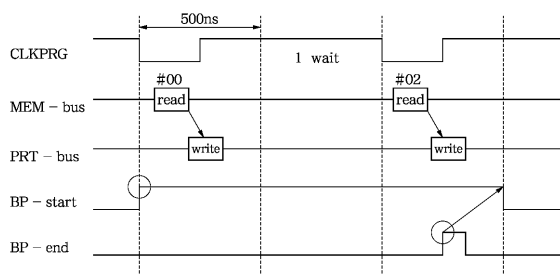
【図 19】



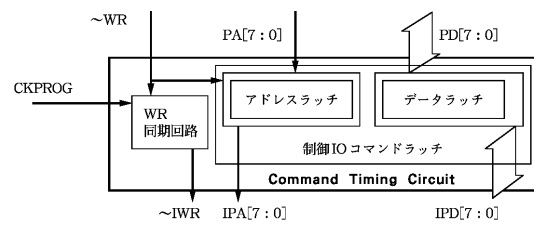
【図 20】



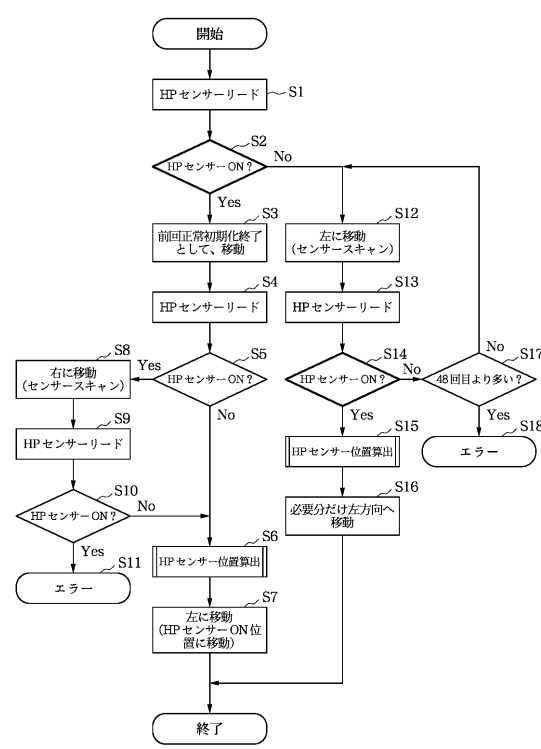
【図 21】



【図 22】



【図 2 3】



【図 2 4】

機能	相セレクト /回転方向	PWMA /PWMRUN	PWMB	レジスタライト アドレス/トリガー	WAITH 64usec	WAITL 0.5usec
初期設定	6B01	6101		1324		
等速 1-1	6901	6401	6528	1200	1508	1455
等速 1-2		6406	6526	1200	1508	1455
等速 1-3		6414	6523	1200	1508	1455
等速 1-4		6419	6519	1200	1508	1455
等速 1-5		6423	6514	1200	1508	1455
等速 1-6		6426	6506	1200	1508	1455
等速 2-1	6901	6428	6501	1200	1508	1455
等速 2-2		6426	6506	1200	1508	1455
等速 2-3		6423	6514	1200	1508	1455
等速 2-4		6419	6519	1200	1508	1455
等速 2-5		6414	6523	1200	1508	1455
等速 2-6		6406	6526	1200	1508	1455
等速 3-1 (等速 1-1)	6901	6401	6528	1200	1508	1455
□						

【図 2 5】

機能	CR相セレクト /回転方向	CRPWMA /PWMRUN	CRPWMB	WAITH 64usec	WAITL 0.5usec
初期設定	6B01	6101			
自相		6401	6528	159C	1400
加速 1-1	6901	6401	6528	1519	14D3
加速 1-2		6406	6526	1518	14B3
加速 1-3		6414	6523	1516	14E0
加速 1-4		6419	6519	1514	14E6
加速 1-5		6423	6514	1513	1400
加速 1-6		6426	6506	1511	145A
加速 2-1	6901	6428	6501	150F	14ED
加速 2-2		6426	6506	150E	14B3
加速 2-3		6423	6514	150D	14A6
加速 2-4		6419	6519	150C	14C0
加速 2-5		6414	6523	150B	14FA
加速 2-6		6406	6526	150B	14A0
加速 3-1	6901	6401	6528	150A	1453
加速 3-2		6406	6526	150A	140D
加速 3-3		6414	6523	1509	14CD
加速 3-4		6419	6519	1509	148D
加速 3-5		6423	6514	1508	145A
加速 3-6		6426	6506	1508	1426
加速 4-1	6901	6428	6501	1508	14F3
□					
□					
加速 22-1	6901	6428	6501	1503	1433
加速 22-2		6426	6506	1503	1433
加速 22-3		6423	6514	1503	1433
加速 22-4		6419	6519	1503	1433
加速 22-5		6414	6523	1503	1433
加速 22-6		6406	6526	1503	1433

【図 2 6】

機能	CR相セレクト /回転方向	CRPWMA /PWMRUN	CRPWMB	WAITH 64usec	WAITL 0.5usec
等速 1-1	6901	6401	6528	1503	1433
等速 1-2		6406	6526	1503	1433
等速 1-3		6414	6523	1503	1433
等速 1-4		6419	6519	1503	1433
等速 1-5		6423	6514	1503	1433
等速 1-6		6426	6506	1503	1433
等速 2-1	6901	6428	6501	1503	1433
等速 2-2		6426	6506	1503	1433
等速 2-3		6423	6514	1503	1433
等速 2-4		6419	6519	1503	1433
等速 2-5		6414	6523	1503	1433
等速 2-6		6406	6526	1503	1433
等速 3-1	6901	6401	6528	1503	1433
□					

【図 27】

機能	CR相セレクト /回転方向	CRPWWA /PWWRUN	CRPWMB	WAITH 64usec	WAITL 0.5usec
減速 1-1	6901	6401	6528	1503	1433
減速 1-2		6406	6526	1503	1433
減速 1-3		6414	6523	1503	1433
減速 1-4		6419	6519	1503	1433
減速 1-5		6423	6514	1503	1433
減速 1-6		6426	6506	1503	1433
減速 2-1	6901	6428	6501	1503	1433
減速 2-2		6426	6506	1503	1433
減速 2-3		6423	6514	1503	143A
減速 2-4		6419	6519	1503	143A
減速 2-5		6414	6523	1503	143A
減速 2-6		6406	6526	1503	143A
減速 3-1	6901	6401	6528	1503	143A
減速 3-2		6406	6526	1503	143A
減速 3-3		6414	6523	1503	143A
減速 3-4		6419	6519	1503	1440
減速 3-5		6423	6514	1503	1440
減速 3-6		6426	6506	1503	1440
減速 4-1	6901	6428	6501	1503	1440
□					
□					
減速 22-1		6428	6501	1511	145A
減速 22-2		6426	6506	1513	1400
減速 22-3		6423	6514	1514	14B6
減速 22-4	6901	6419	6519	1516	14E0
減速 22-5		6414	6523	1518	14B3
減速 22-6		6406	6526	1519	14D3
自相		6401	6528	1527	1400
		6100			

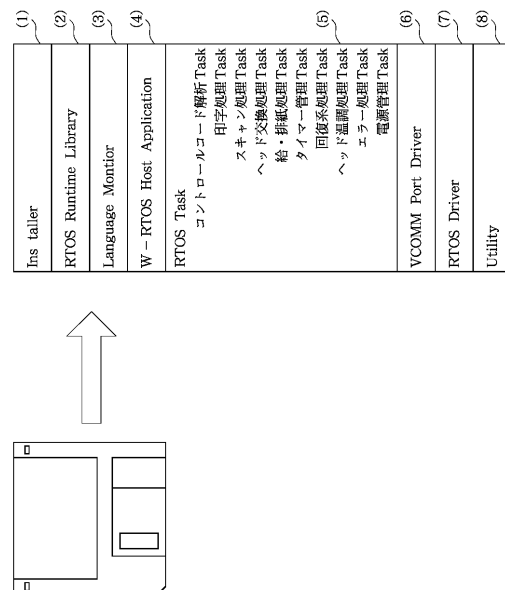
【図 28】

機能	LF相セレクト /回転方向	LFPWWA /PWWRUN	LFPWMB	WAITH 64usec	WAITL 0.5usec
初期設定	7B01	7101			
自相		7401	7528	159C	1400
加速 1-1		7901	7401	1505	1424
加速 1-2			7406	1505	141C
加速 1-3			7414	1505	140C
加速 1-4			7419	1505	1400
加速 1-5	7901		7423	1504	14F0
加速 1-6			7426	1504	14E4
加速 2-1			7428	1504	14D4
加速 2-2			7426	1504	14C8
加速 2-3			7423	1504	14BC
加速 2-4			7419	1504	14B0
加速 2-5	7901		7414	1504	14A4
加速 2-6			7406	1504	1498
加速 3-1			7401	1504	148C
加速 3-2			7406	1504	1480
加速 3-3			7414	1504	1478
加速 3-4			7419	1504	146C
加速 3-5	7901		7423	1504	1464
加速 3-6			7426	1504	1458
加速 4-1			7428	1504	1450
□					
□					
加速 12-1	7901		7728	1503	1430
加速 12-2			7426	1503	142C
加速 12-3			7423	1503	1428
加速 12-4			7419	1503	1424
加速 12-5			7414	1503	1424
加速 12-6			7406	1503	1420

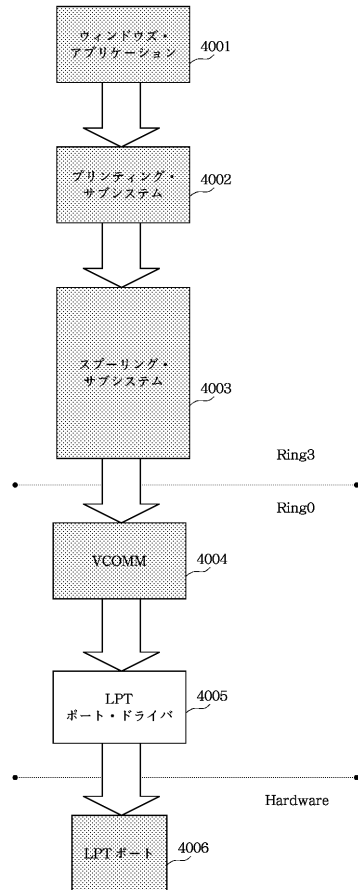
【図 29】

機能	LF相セレクト /回転方向	LFPWWA /PWWRUN	LFPWMB	WAITH 64usec	WAITL 0.5usec
減速 1-1	7901	7401	7528	1503	1420
減速 1-2		7406	7526	1503	1424
減速 1-3		7414	7523	1503	1424
減速 1-4		7419	7519	1503	1428
減速 1-5		7423	7514	1503	142C
減速 1-6		7426	7506	1503	1430
減速 2-1	7901	7428	7501	1503	1434
減速 2-2		7426	7506	1503	1438
減速 2-3		7423	7514	1503	143E
減速 2-4		7419	7519	1503	1440
減速 2-5		7414	7523	1503	1444
減速 2-6		7406	7526	1503	1448
減速 3-1	7901	7401	7528	1503	144C
減速 3-2		7406	7526	1503	1450
減速 3-3		7414	7523	1503	1458
減速 3-4		7419	7519	1503	145C
減速 3-5		7423	7514	1503	1460
減速 3-6		7426	7506	1503	1464
減速 4-1	7901	7428	7501	1503	1468
□					
□					
減速 12-1		7428	7501	1504	14E4
減速 12-2		7426	7506	1504	14F0
減速 12-3		7423	7514	1505	1400
減速 12-4	7901	7419	7519	1505	140C
減速 12-5		7414	7523	1505	141C
減速 12-6		7406	7526	1505	1424
自相		7401	7528	159C	1400
停止処理		7100			

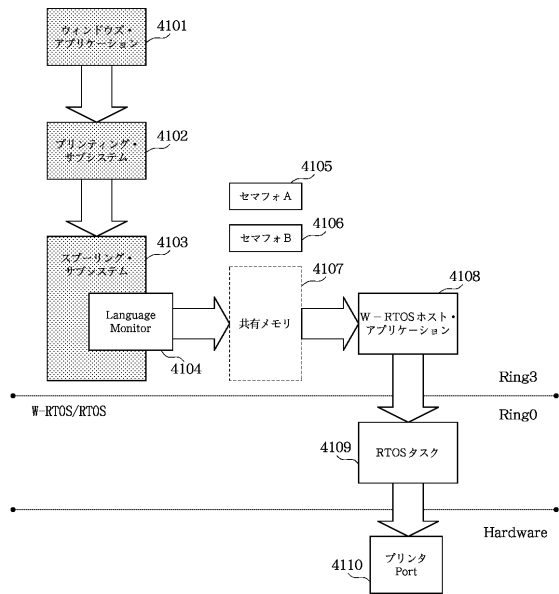
【図 30】



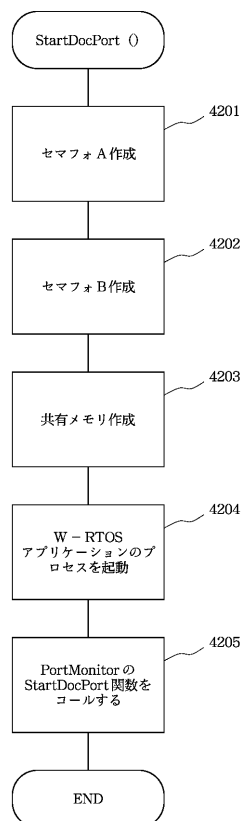
【図 3 1】



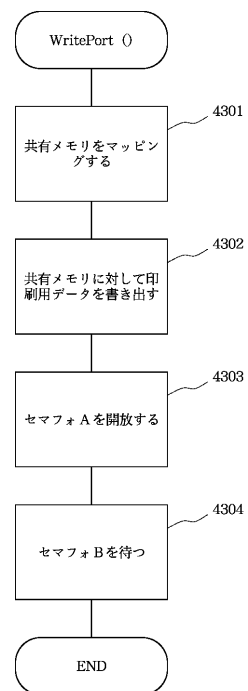
【図 3 2】



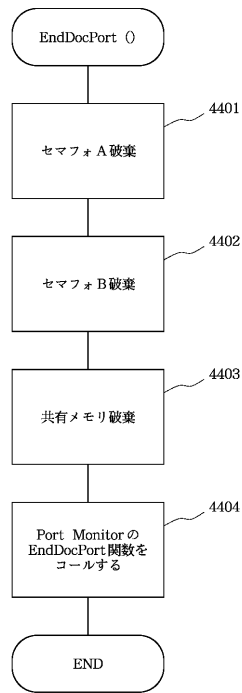
【図 3 3】



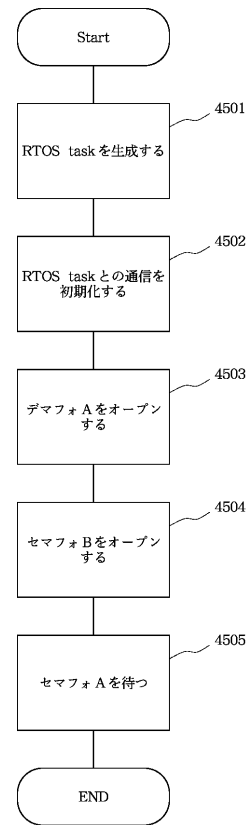
【図 3 4】



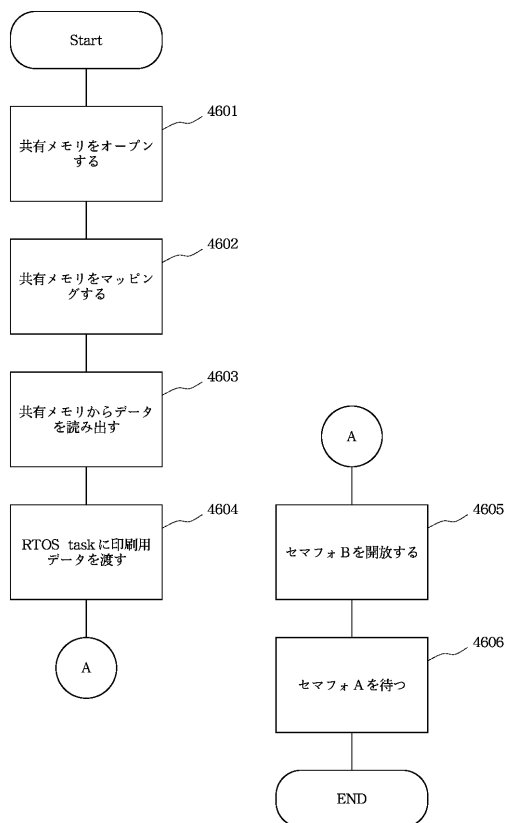
【図 35】



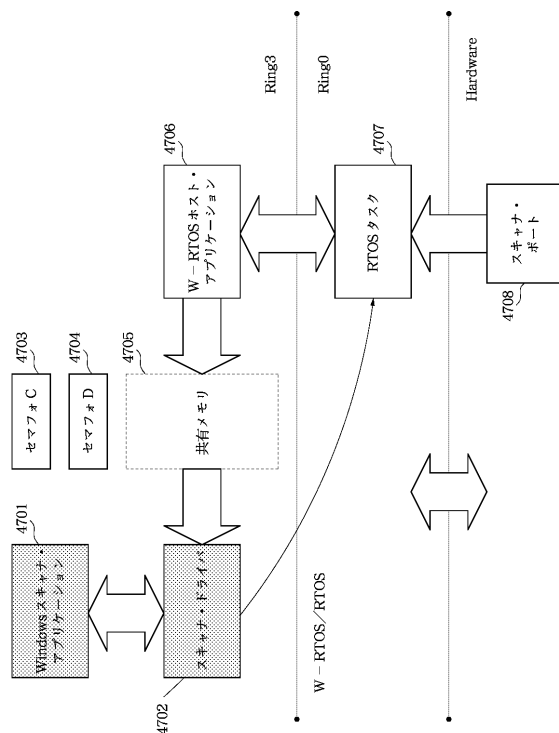
【図 36】



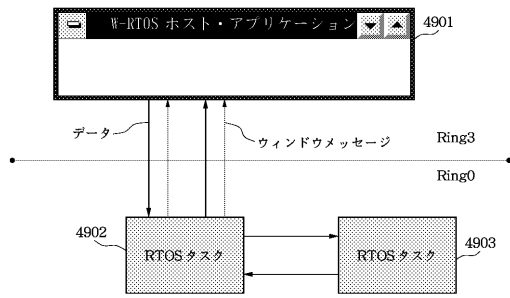
【図 37】



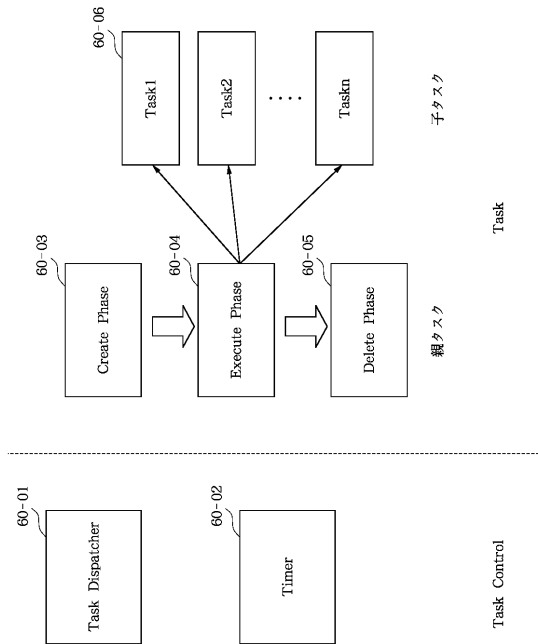
【図 38】



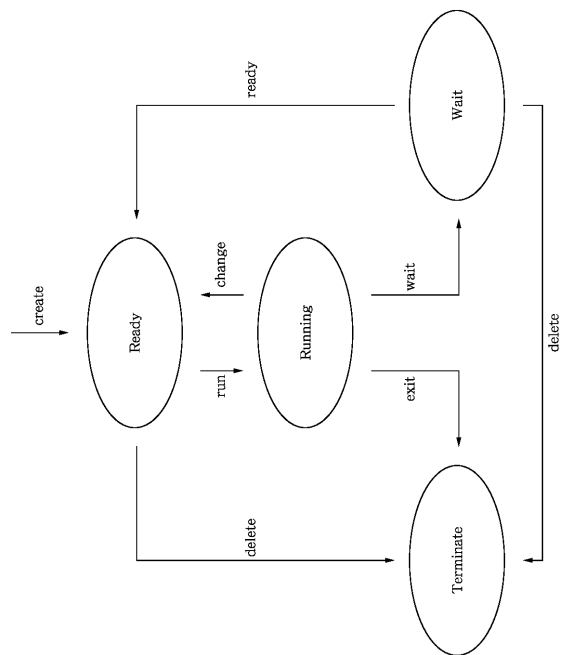
【図 39】



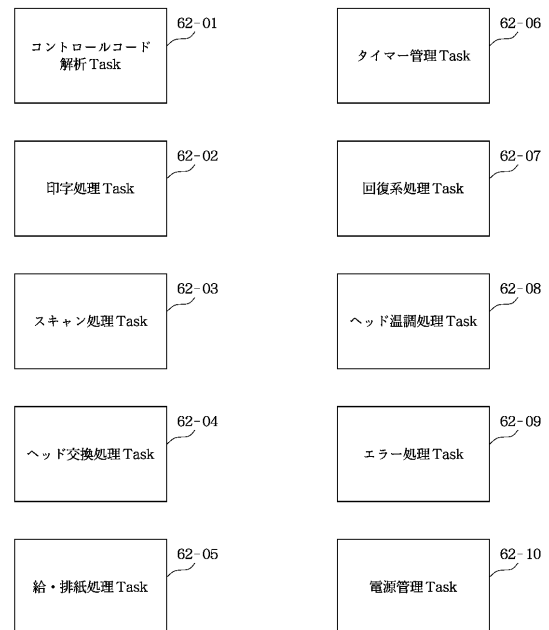
【図 40】



【図 41】



【図 42】



【 図 4 3 】

ESC command length data

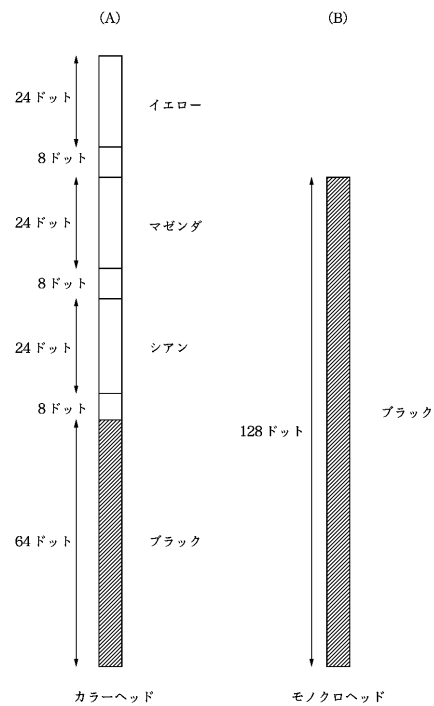
【 図 4 4 】

- (1) ESC P 0
- (2) ESC S 0
- (3) ESC E 0
- (4) ESC H 1 mode
mode = 0 : start
mode = 1 : end
- (5) ESC I 0
- (6) ESC 0 0
- (7) ESC i color length data ...
color = K : ブラック
color = C : シアン
color = M : マゼンタ
color = Y : イエロー
- (8) ESC s 0
- (9) ESC f 2 feed
- (10) ESC r 0

【 図 4 5 】

ESC i K length data ...
ESC i C length data ...
ESC i M length data ...
ESC i Y length data ...
ESC s 0

【 図 4 6 】



【図 4 7】

Y
M
C
K

8 ラスタ
8 ラスタ
8 ラスタ
8 ラスタ

【図 4 8】

Y
M
C
K

1 6 ラスタ
1 6 ラスタ
1 6 ラスタ
1 6 ラスタ

【図 4 9】

Y
M
C
K

1 6 0 ラスタ

1 2 8 ラスタ

9 6 ラスタ

6 4 ラスタ

5760 ドット

【図 5 0】

Y
M
C
K

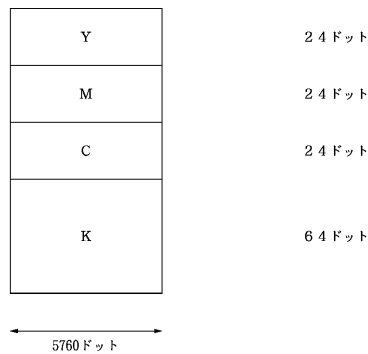
2 4 ドット

2 4 ドット

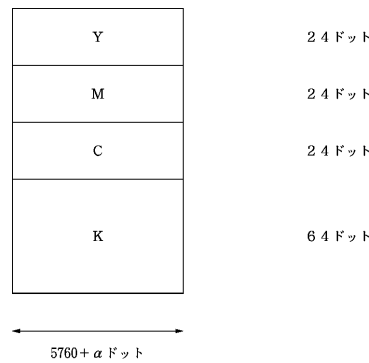
2 4 ドット

6 4 ドット

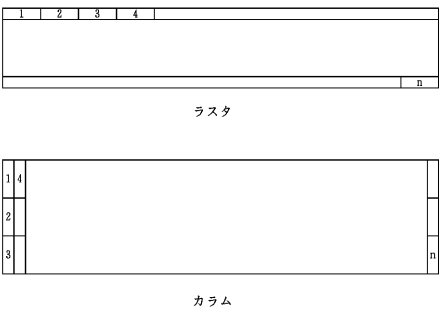
【 図 5 1 】



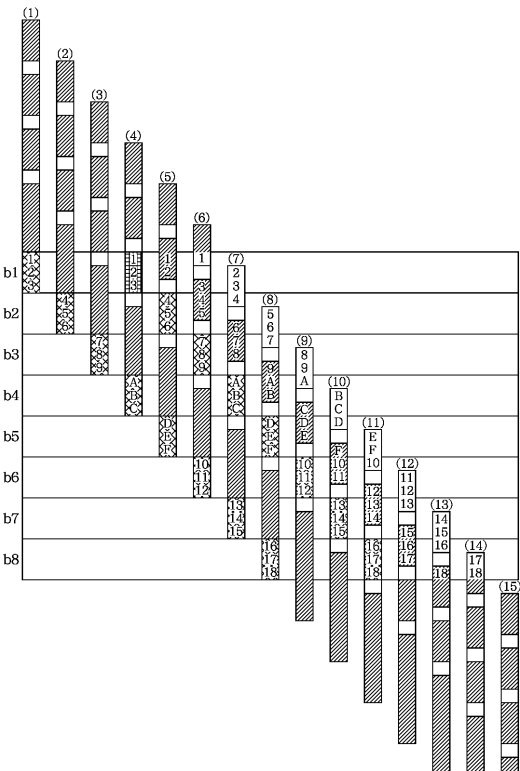
【 図 5 2 】



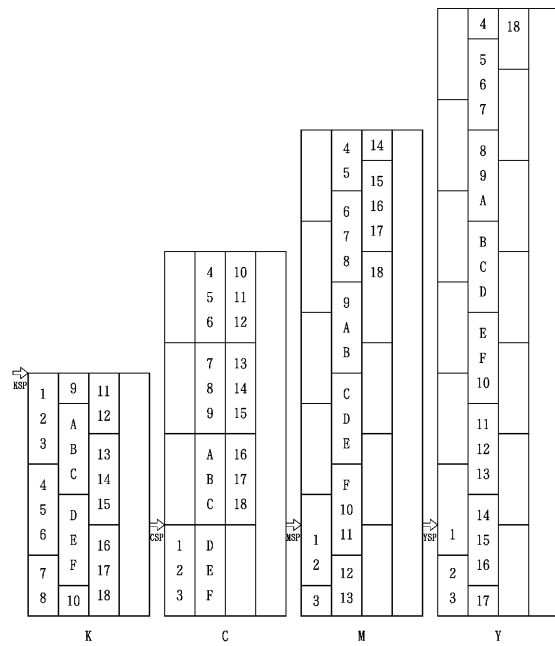
【 図 5 3 】



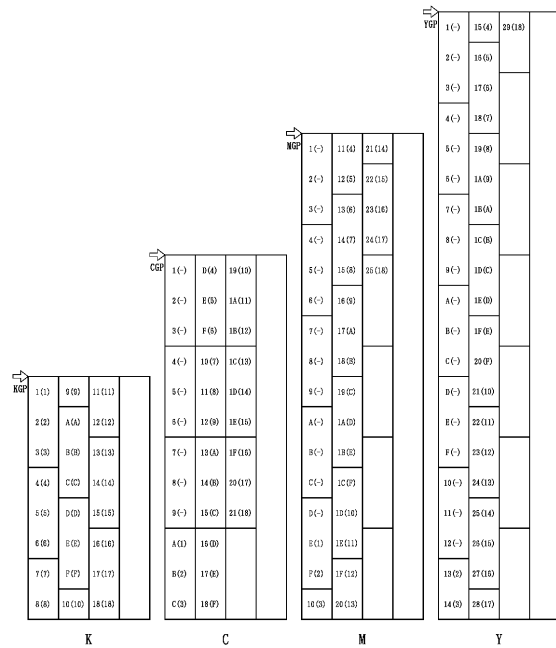
【 図 5 4 】



【図 5 5】



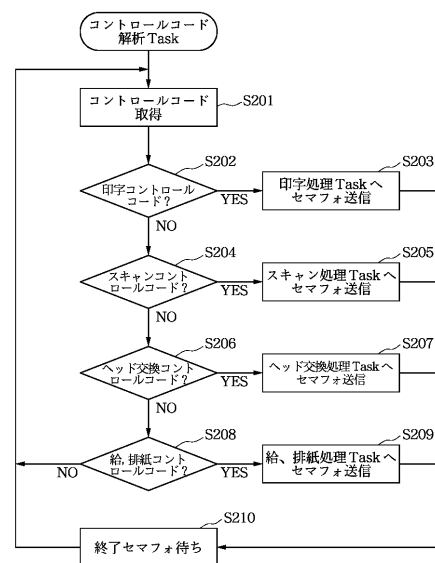
【図 5 6】



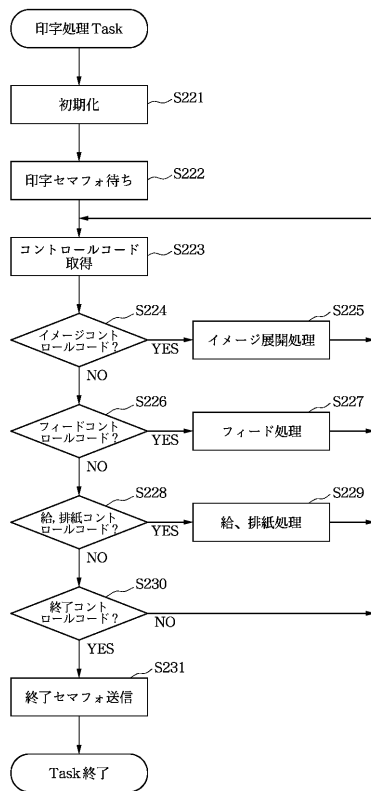
【図 5 7】

KデータセットポイントKSP	71-01
CデータセットポイントCSP	71-02
MデータセットポイントMSP	71-03
YデータセットポイントYSP	71-04
KデータゲットポイントKGP	71-05
CデータゲットポイントCGP	71-06
MデータゲットポイントMGP	71-07
YデータゲットポイントYGP	71-08
ラスタカウンタRCNT	71-09

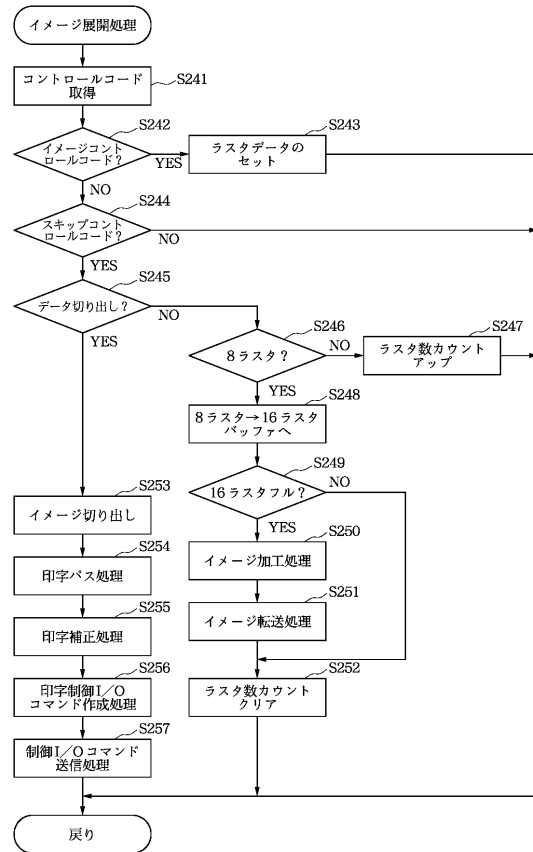
【図 5 8】



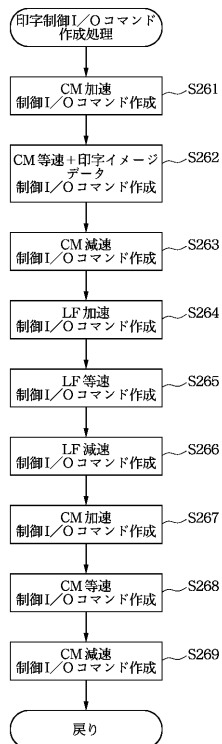
【図 59】



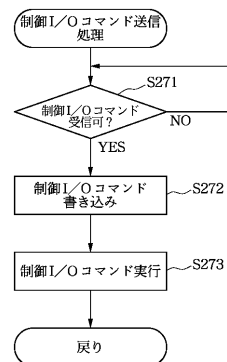
【図 60】



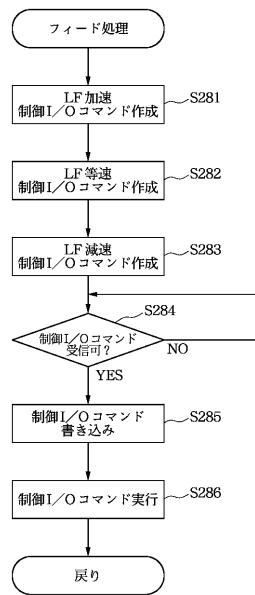
【図 61】



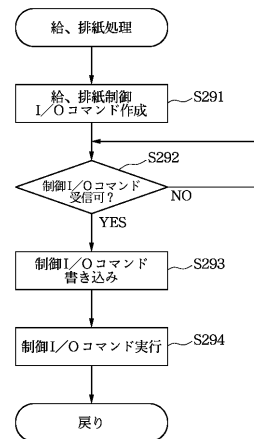
【図 62】



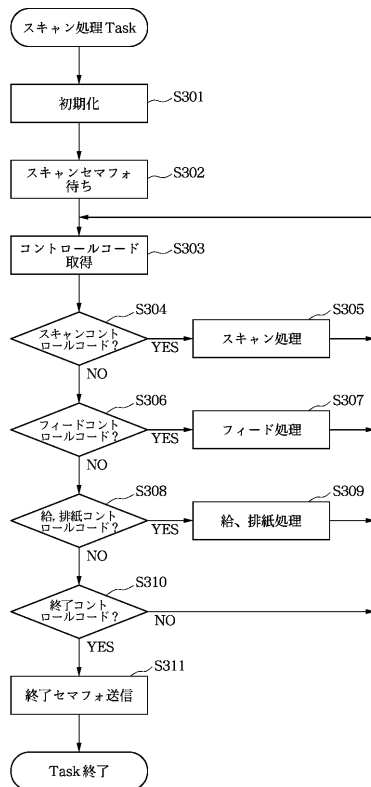
【図 63】



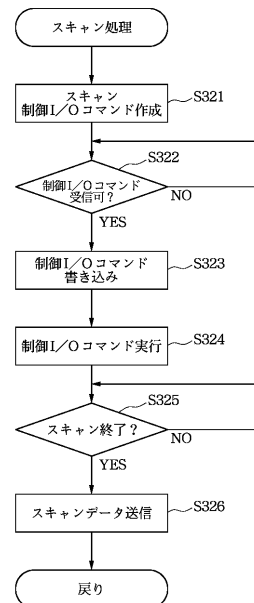
【図 64】



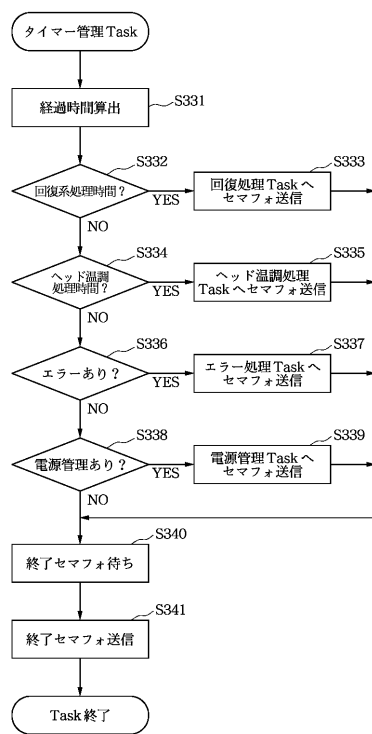
【図 65】



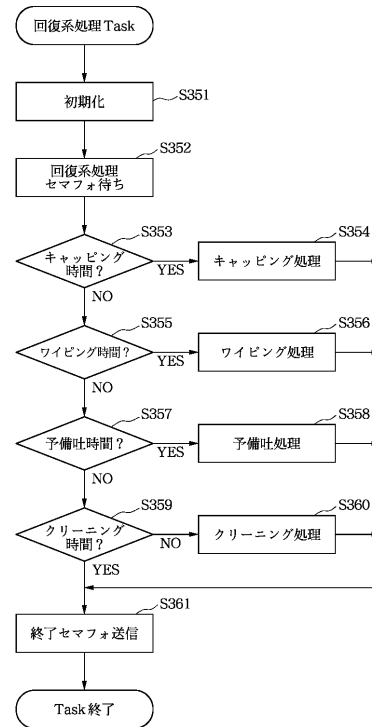
【図 66】



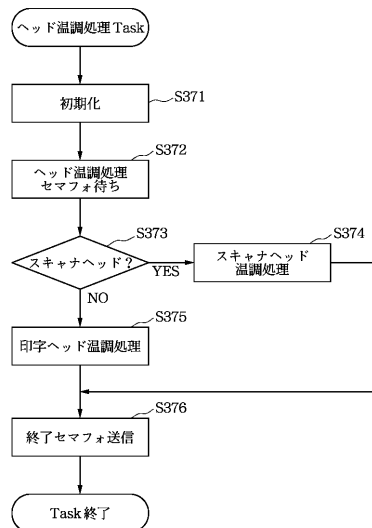
【図 67】



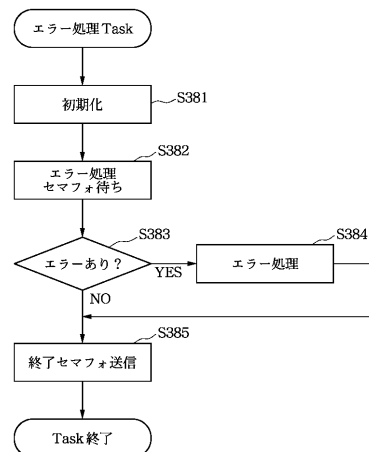
【図 68】



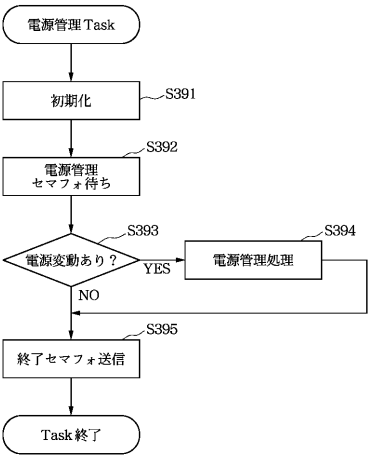
【図 69】



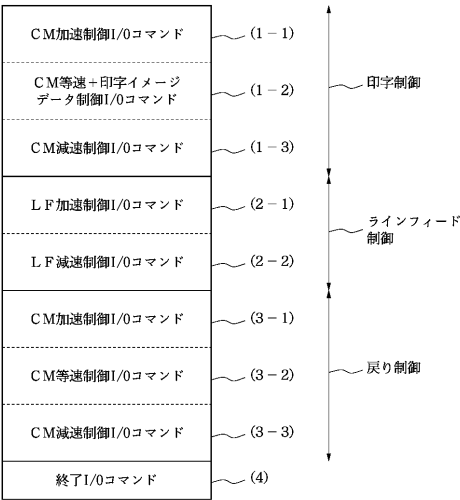
【図 70】



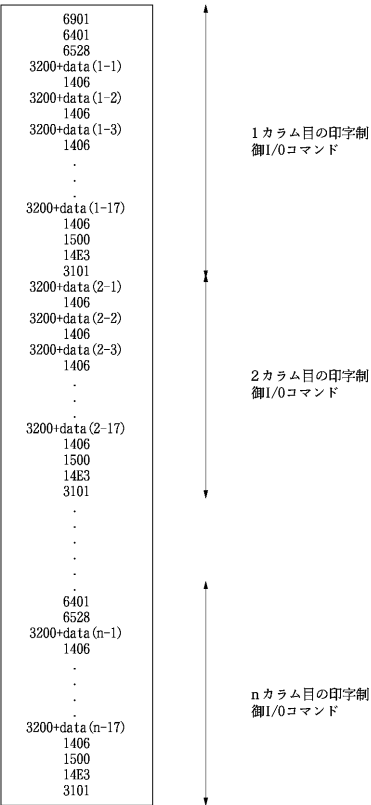
【図 7 1】



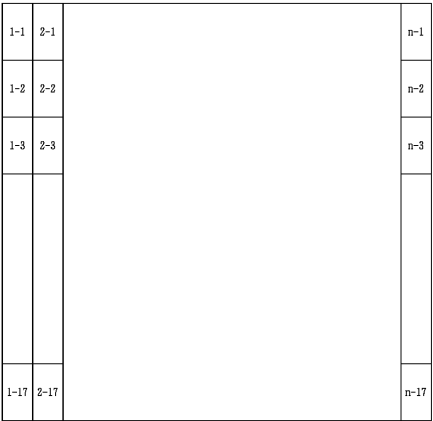
【図 7 2】



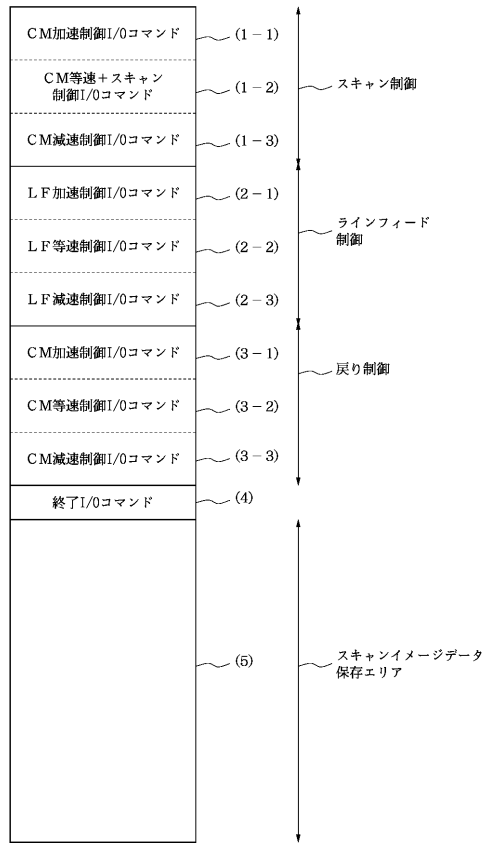
【図 7 3】



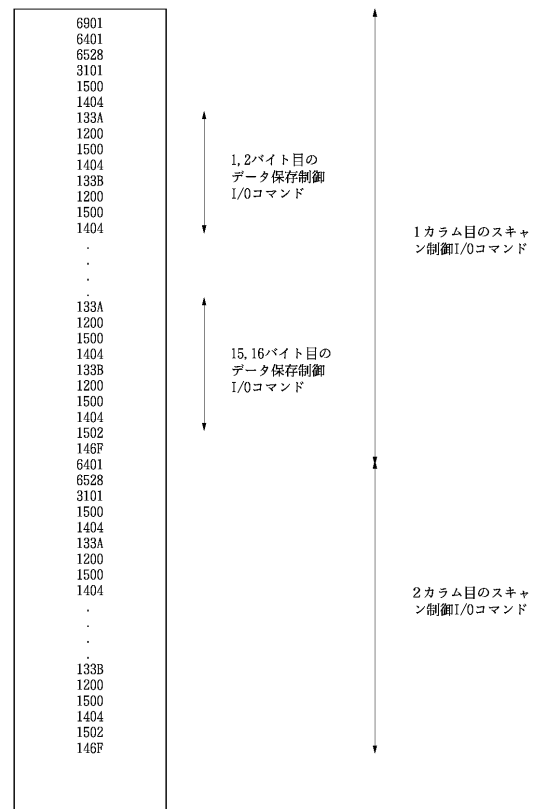
【図 7 4】



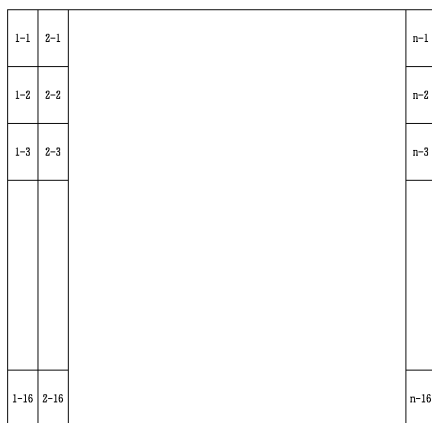
【図 75】



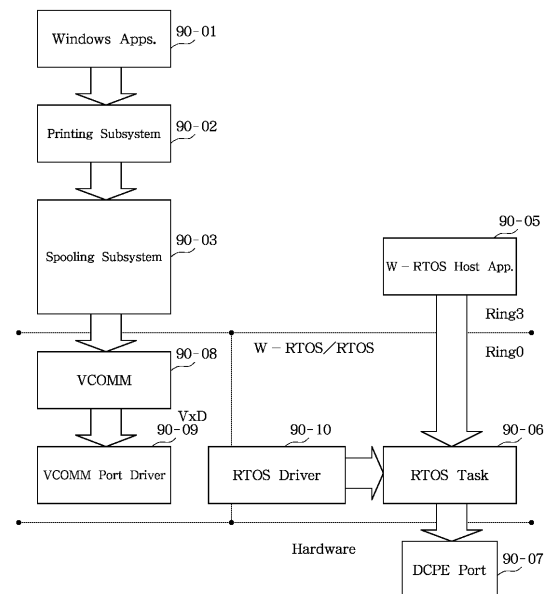
【図 76】



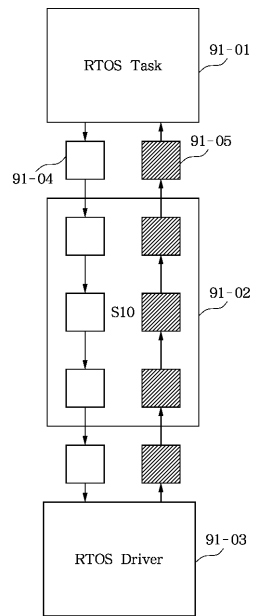
【図 77】



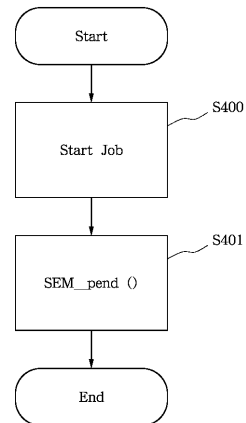
【図 78】



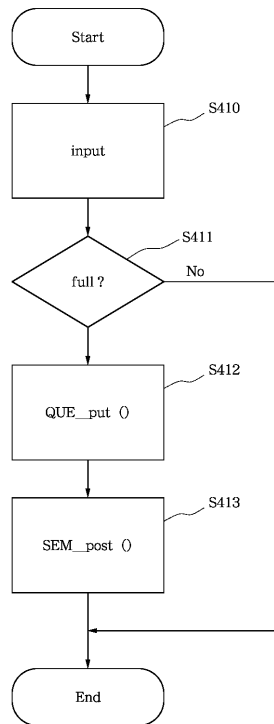
【図 79】



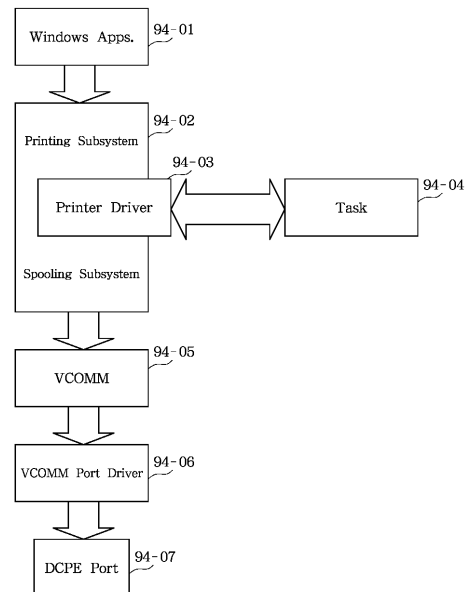
【図 80】



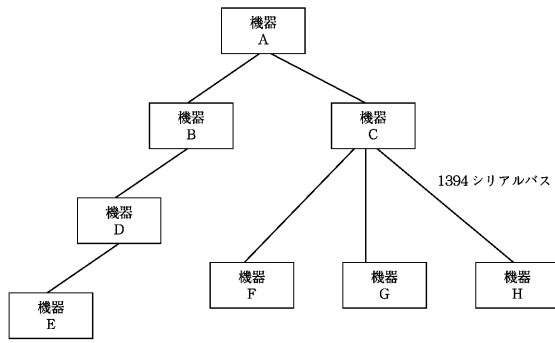
【図 81】



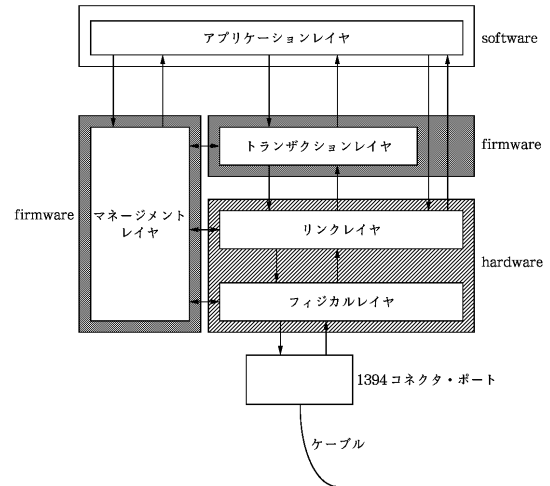
【図 82】



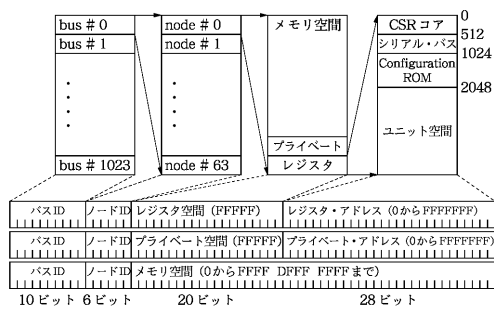
【図 8 3】



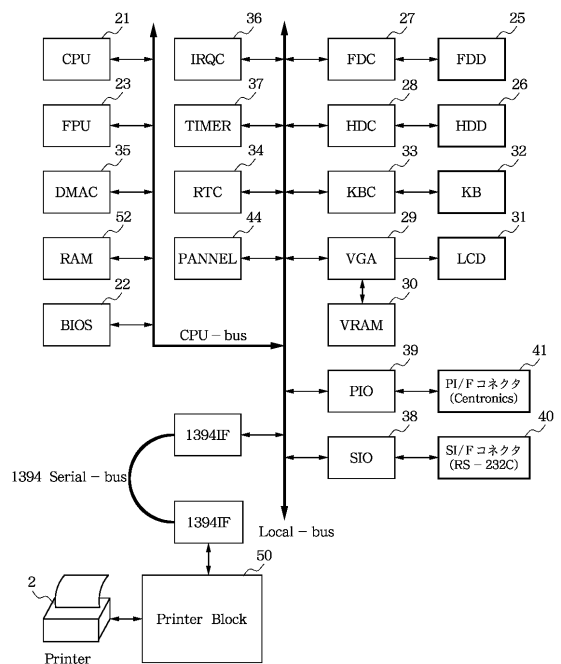
【図 8 4】



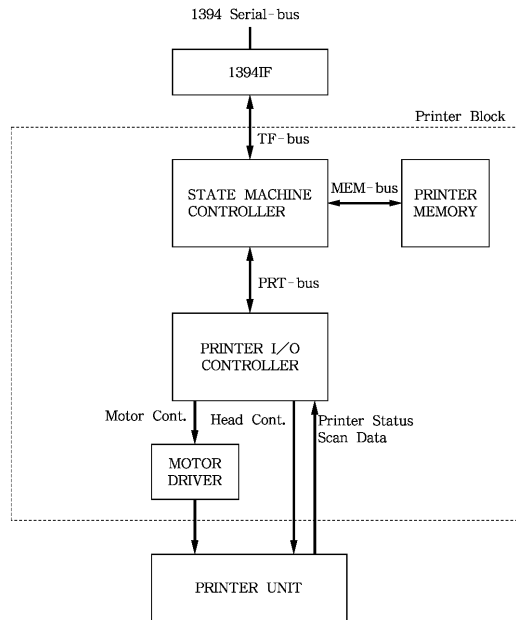
【図 8 5】



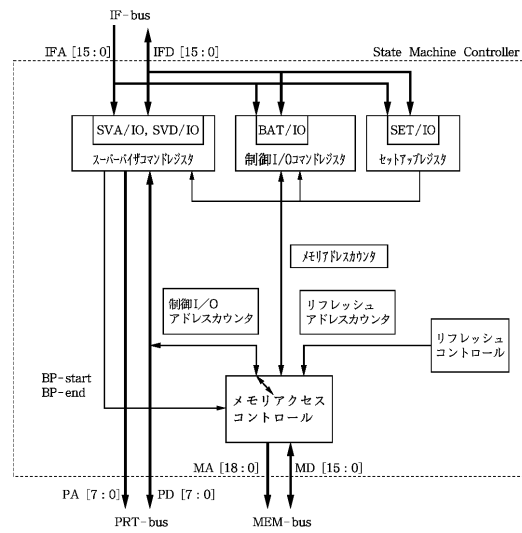
【図 8 6】



【図 87】



【図 88】



フロントページの続き

- (72)発明者 小林 真琴
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
- (72)発明者 立山 二郎
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
- (72)発明者 中村 敦
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
- (72)発明者 内藤 久嗣
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

審査官 内田 正和

- (56)参考文献 特開平07-250228(JP,A)
特開平02-032421(JP,A)
特開平06-324811(JP,A)
特開平01-241482(JP,A)
特開平6-75905(JP,A)
特開平5-158713(JP,A)

(58)調査した分野(Int.Cl., DB名)

G06F 3/12
B41J 29/38
G09G 5/00
H04N 1/00