

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5147520号  
(P5147520)

(45) 発行日 平成25年2月20日(2013.2.20)

(24) 登録日 平成24年12月7日(2012.12.7)

(51) Int.Cl. F I  
**HO4N 1/00 (2006.01)** HO4N 1/00 C

請求項の数 4 (全 25 頁)

<p>(21) 出願番号 特願2008-114059 (P2008-114059)                  (22) 出願日 平成20年4月24日 (2008.4.24)                  (65) 公開番号 特開2009-267705 (P2009-267705A)                  (43) 公開日 平成21年11月12日 (2009.11.12)                  審査請求日 平成23年4月25日 (2011.4.25)</p>	<p>(73) 特許権者 000001007                  キヤノン株式会社                  東京都大田区下丸子3丁目30番2号                  (74) 代理人 100066061                  弁理士 丹羽 宏之                  (74) 代理人 100177437                  弁理士 中村 英子                  (74) 代理人 100143340                  弁理士 西尾 美良                  (72) 発明者 中村 彰浩                  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 山内 裕史</p>
--	--

最終頁に続く

(54) 【発明の名称】 通信方法

(57) 【特許請求の範囲】

【請求項1】

第一の制御部と第二の制御部とを有し、前記第一の制御部または前記第二の制御部が、コマンド信号を発行し、前記コマンド信号に対する応答信号またはステータス信号を受信することにより、前記第一の制御部と前記第二の制御部との間で通信を行う通信装置の通信方法において、

前記コマンド信号に対する、前記応答信号または前記ステータス信号が返信されるまでは新たなコマンド信号の発行を停止する工程と、

前記第一の制御部または前記第二の制御部が、前記コマンド信号を発行していない状態で、前記応答信号または前記ステータス信号を受信した場合、または、通信エラーを検知した場合には、受信不良を示す信号を発行する工程と、

前記第一の制御部または前記第二の制御部が、前記コマンド信号を発行してから前記応答信号または前記ステータス信号を受信するまでの間に、前記通信エラーを検知し、または、前記受信不良を示す信号を受信した場合には、前記コマンド信号を再発行する工程と、を有し、

前記第一の制御部または前記第二の制御部が、前記コマンド信号を発行していない状態で、前記受信不良を示す信号を受信した場合には、前記受信不良を示す信号を無視する工程を備えることを特徴とする通信方法。

【請求項2】

前記第一の制御部または前記第二の制御部が、前記コマンド信号を発行してから一定時

間経過しても前記応答信号または前記ステータス信号が返信されなかった場合には、前記コマンド信号を再発行する工程を備えることを特徴とする請求項 1 に記載の通信方法。

【請求項 3】

前記第一の制御部または前記第二の制御部が発行する前記コマンド信号に識別コードを付加し、前記識別コードを記憶する工程と、

前記第一の制御部または前記第二の制御部が、前記識別コードが付加された前記コマンド信号を受信した場合に、前記識別コードを前記応答信号または前記ステータス信号に付加する工程と、

前記第一の制御部または前記第二の制御部が、前記識別コードが付加された前記応答信号または前記ステータス信号を受信した場合に、前記記憶する工程により記憶された前記識別コードと前記応答信号または前記ステータス信号に付加された前記識別コードとが異なる場合には、前記受信不良を示す信号を送信する工程と、

を備えることを特徴とする請求項 1 又は 2 に記載の通信方法。

【請求項 4】

前記第一の制御部または前記第二の制御部が、識別コードが付加されたコマンド信号を受信した場合に、前記識別コードを記憶する工程と、

前記第一の制御部または前記第二の制御部が、識別コードが付加された次のコマンド信号を受信した場合に、前記記憶する工程により記憶された識別コードと、前記次のコマンド信号に付加された識別コードとが一致する場合には、前記次のコマンド信号に対応する動作を起動せず、前記次のコマンド信号に対する前記応答信号またはステータス信号を返

信する工程と、

を備えることを特徴とする請求項 3 に記載の通信方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、通信方法に関し、詳しくは記録媒体上に画像を形成する画像形成装置に関するものであり、特に、ハンドシェイクプロトコルを有した機内通信を持つ画像形成装置に関するものである。

【背景技術】

【0002】

従来、通信のプロトコルは、受信不良を示すところの NACK ( N e g a t i v e A C K n o w l e d g e m e n t 信号 ) ( ネガティブアクナリッジ信号 ) ( 以下 NACK ) を受信したら再送信手段を持つように設計されている ( 例えば、特許文献 1 参照 ) 。

【0003】

すなわち、特許文献 1 では、ホストから与えられた送信データを P I O データ保持部に保持する。パケット生成部は送信データをもとにこの送信データの記憶場所の情報を含む送信パケットを生成し、これを通信リンク制御部がネットワークへ送出する。このとき、送信データの記憶領域は解放しない。この送信パケットに対して送信データの記憶場所の情報を含む NACK が受信された場合には、その記憶場所に送信データが未だ記憶されているので、これをもとに送信パケット生成して再送する。ACK ( A C K n o w l e d g e m e n t 信号 ) ( アクナリッジ信号 ) パケット ( 以下 ACK ) が受信された場合には、送信データの記憶領域を開放する。

【0004】

このような構成とすることで、再送による信頼性のある通信を遅延時間の増加やハードウェア使用量の増加を抑制しつつ実現することを可能にしている。

【特許文献 1】特開 2004 - 274376 号公報

【発明の開示】

【発明が解決しようとする課題】

【0005】

しかしながら、上記従来例では、NACK に対し必ず再送信を実行するようになってい

10

20

30

40

50

るため、以下のような課題があった。

【 0 0 0 6 】

A - C P U が B - C P U に N A C K を送信した場合に、通信エラーが発生すると、B - C P U から A - C P U に N A C K が返信され、A - C P U の記憶場所に記憶されている N A C K が再送信される。B - C P U の記憶場所にも N A C K が記憶されているので、B - C P U から N A C K が再送信される状況となっている。そして結果的には、N A C K をお互いに送信しつづけることにより、正常に通信ができなくなってしまう（通信のハングアップとも言う）こととなる。

【 0 0 0 7 】

本発明は以上の点に着目して成されたもので、ネガティブアクナリッジ信号に対しネガティブアクナリッジ信号を再送する状況を避け、正常に通信ができなくなる状況を回避する通信方法を提供することを課題とする。

【 0 0 0 8 】

また、コマンド信号とアクナリッジ信号とステータス信号に識別コードを付加し、通信エラーの検知性能を向上させる通信方法を提供することを課題とする。

【 0 0 0 9 】

さらに、対応する動作が二重起動され、不必要な動作が引き起こされることを防止する通信方法を提供することを課題とする。

【課題を解決するための手段】

【 0 0 1 0 】

前記課題を解決するために、本発明は以下の構成を備える。

【 0 0 1 1 】

( 1 ) 第一の制御部と第二の制御部とを有し、前記第一の制御部または前記第二の制御部が、コマンド信号を発行し、前記コマンド信号に対する応答信号またはステータス信号を受信することにより、前記第一の制御部と前記第二の制御部との間で通信を行う通信装置の通信方法において、前記コマンド信号に対する、前記応答信号または前記ステータス信号が返信されるまでは新たなコマンド信号の発行を停止する工程と、前記第一の制御部または前記第二の制御部が、前記コマンド信号を発行していない状態で、前記応答信号または前記ステータス信号を受信した場合、または、通信エラーを検知した場合には、受信不良を示す信号を発行する工程と、前記第一の制御部または前記第二の制御部が、前記コマンド信号を発行してから前記応答信号または前記ステータス信号を受信するまでの間に、前記通信エラーを検知し、または、前記受信不良を示す信号を受信した場合には、前記コマンド信号を再発行する工程と、を有し、前記第一の制御部または前記第二の制御部が、前記コマンド信号を発行していない状態で、前記受信不良を示す信号を受信した場合には、前記受信不良を示す信号を無視する工程を備えることを特徴とする通信方法。

【発明の効果】

【 0 0 1 3 】

本発明によれば、ネガティブアクナリッジ信号の再送によって、正常に通信できない状況の発生が回避されるので、通信の品位が向上する。さらに、2個のCPUから独立して双方向に通信が実行できる。また、通信ごとの識別コードを付加することにより通信エラーの発生をより詳しく検知することができる。さらに、通信ごとの識別コードを確認することにより、不必要な動作を誤って二重起動することがなくなるので、通信エラーを起因とするハードウェアトラブルを防止することができる。

【発明を実施するための最良の形態】

【 0 0 1 4 】

以下本発明を実施するための最良の形態を、実施例により詳しく説明する。

【実施例 1】

【 0 0 1 5 】

本実施例の通信プロトコルは、以下の〔 1 〕～〔 1 8 〕に示す 1 8 項目で定義されるものである。

〔 1 〕 2 個の C P U を接続する通信である。

〔 2 〕 2 個の C P U を A - C P U ( 後述する C P U 1 1 ) , B - C P U ( 後述する C P U 1 6 ) とする。

〔 3 〕 相手側に命令をするところの C o m m a n d ( 以下 C o m m a n d ) を受信した A - C P U は、その C o m m a n d に対応した受信完了通知であるところの A C K ( 以下 A C K ) または、状況を返信するところの S t a t u s ( 以下 S t a t u s ) を返信する。

〔 4 〕 C o m m a n d を受信した B - C P U は、その C o m m a n d に対応した A C K または、 S t a t u s を返信する。

〔 5 〕 A - C P U から B - C P U に対し C o m m a n d を発行した場合には、 B - C P U よりその C o m m a n d に対応した正常な A C K または正常な S t a t u s が返信されるまでは、 A - C P U は B - C P U に対し新たな C o m m a n d は発行しない。

10

〔 6 〕 B - C P U から A - C P U に対し C o m m a n d を発行した場合には、 A - C P U よりその C o m m a n d に対応した正常な A C K または正常な S t a t u s が返信されるまでは、 B - C P U は A - C P U に対し新たな C o m m a n d は発行しない。

〔 7 〕 A - C P U が C o m m a n d を発行していない状態で、通信エラーを検知したら A - C P U は受信不良を示すところの N A C K ( 以下 N A C K ) を発行する。

〔 8 〕 B - C P U が C o m m a n d を発行していない状態で、通信エラーを検知したら B - C P U は N A C K を発行する。

〔 9 〕 A - C P U が C o m m a n d を発行し、 A C K または、 S t a t u s を受信するまでの間に、通信エラーを検知したら最後に発行した C o m m a n d を再発行する。

20

〔 1 0 〕 B - C P U が C o m m a n d を発行し、 A C K または、 S t a t u s を受信するまでの間に、通信エラーを検知したら最後に発行した C o m m a n d を再発行する。

〔 1 1 〕 A - C P U が C o m m a n d を発行していない状態で、 N A C K を受信したら N A C K を無視しなにもしない。

〔 1 2 〕 B - C P U が C o m m a n d を発行していない状態で、 N A C K を受信したら N A C K を無視しなにもしない。

〔 1 3 〕 A - C P U が C o m m a n d を発行し、 A C K または、 S t a t u s を受信するまでの間に、 N A C K を受信したら最後に発行した C o m m a n d を再発行する。

〔 1 4 〕 B - C P U が C o m m a n d を発行し、 A C K または、 S t a t u s を受信するまでの間に、 N A C K を受信したら最後に発行した C o m m a n d を再発行する。

30

〔 1 5 〕 A - C P U が C o m m a n d を発行していない状態にもかかわらず A C K または S t a t u s を受信した場合には、 N A C K を発行する。

〔 1 6 〕 B - C P U が C o m m a n d を発行していない状態にもかかわらず A C K または S t a t u s を受信した場合には、 N A C K を発行する。

〔 1 7 〕 A - C P U が C o m m a n d を発行したにもかかわらず一定時間経過しても A C K または、 S t a t u s が返信されなかった場合には、最後に発行した C o m m a n d を再発行する。

〔 1 8 〕 B - C P U が C o m m a n d を発行したにもかかわらず一定時間経過しても A C K または、 S t a t u s が返信されなかった場合には、最後に発行した C o m m a n d を再発行する。

40

【 0 0 1 6 】

以下、図 1 ~ 図 1 2 を用いて、本実施例を詳しく説明する。

【 0 0 1 7 】

[ 画像形成装置の構成 ]

図 1 は本発明の特徴を最もよく表す図面であり、同図において 1 は画像形成を行う画像形成装置本体である。 2 は画像を形成するための用紙を保持するカセットである。 3 は画像形成済みの用紙を排出する排紙トレイである。 4 は用紙を一枚ずつ取り出すため(以下ピックアップと称す)のピックアップローラである。 5 は画像を形成する現像転写部である。 6 は用紙上に転写された画像を定着させる定着器である。 7 は、定着済みの用紙を搬送する搬送ローラであり、 8 はピックアップした用紙を現像転写部へ搬送する搬送ローラ

50

である。10は外部機器であるPC（パーソナルコンピュータ）等と接続しかつ画像情報を生成するビデオコントローラ部（Video-Controller部）である。11はビデオコントローラ部10のCPUである。12は画像情報を生成するビデオ生成部（Video生成部）であり、ビデオ信号（Video信号）により電子写真を用い画像を形成する。13は外部のPCと接続するための通信制御部であり、例えばPCとUSB（Universal Serial Bus）にて接続する場合には、13はUSBコントローラを示す。14は画像形成装置本体内のもう一つのCPUと通信で接続するための通信制御部である。

#### 【0018】

15は画像形成装置の搬送、現像、転写、定着等の機械部分の制御を担当するエンジン基板である。16はエンジン基板15の制御を司るエンジンCPUである。17は、電子写真制御部、18は定着制御部、19は搬送制御部、20はビデオコントローラ部10のCPU11と通信で接続するための通信制御部である。21はPCと接続するプリンタケーブルであり、例えばUSBケーブルが使用される。22は、PC等であり、画像形成装置にプリントの命令を出す。PC22から印字命令を受信したビデオコントローラ部10は、エンジン基板15に印刷命令を出す。それと同時にビデオ生成部12にて画像情報を生成する。用紙は、カセット2からピックアップローラ4によりピックアップされ、搬送ローラ8により現像転写部5に送り込まれる。用紙上に画像が形成され、定着器6に送られる。そして定着が終了した用紙は、排紙トレイ3に排出される。

#### 【0019】

[ビデオコントローラ部10のCPU11とエンジン基板15のCPU16の接続]

図2は、エンジン基板15のCPU16とビデオコントローラ部10のCPU11の接続図である。

#### 【0020】

ビデオコントローラ部10のCPU11は、USBコントローラ13と接続され、プリンタケーブル21を介してPC22とのUSB接続を可能とする。ビデオコントローラ部10のCPU11は、ビデオ生成部12に接続され、ビデオ生成に必要な情報をビデオ生成部12に供給する。ビデオ信号は、差動ドライバ31により差動信号に変換され出力される。差動信号となったビデオ信号は、現像転写部5に伝えられ画像を形成する。ビデオコントローラ部10とエンジン基板15は、GNDを接続する。ビデオコントローラ部10のCPU11とエンジン基板15のCPU16との通信は、それぞれのCPUに内蔵するシリアル通信機能を使用する。CPU11のシリアル通信機能の送信ポートであるところのTXポート（以下TXポート）から通信データが送出される。そしてCPU16のシリアル通信機能の受信ポートであるところのRXポート（以下RXポート）より受信される。CPU16のシリアル通信機能のTXポートから通信データが送出される。そしてCPU11のシリアル通信機能のRXポートより受信される。32, 33, 34, 35は、シリアル通信用のバッファICである。エンジン基板15のCPU16は、ビデオコントローラ部10のCPU11とシリアル通信バッファ34, 35で接続されている。

#### 【0021】

エンジン基板15のCPU16は、電子写真制御部17と接続され電子写真の現像転写を制御する。エンジン基板15のCPU16は定着制御部18と接続され定着器6の温度を制御することにより用紙への画像の定着を制御する。エンジン基板15のCPU16は搬送制御部19と接続され、ピックアップローラ4と搬送ローラ7, 8を制御し、用紙の搬送を制御する。ビデオ生成部12で生成されたビデオ信号は、電子写真制御部17を通過し、レーザ基板38へ伝えられる。そしてレーザ基板38上にある差動レシーバ36により受信され、レーザ37を駆動する。

#### 【0022】

ピックアップローラ4、搬送ローラ7, 8を駆動する搬送モータは、搬送制御部19に接続されている。定着器6は、定着制御部18に接続されている。現像転写部5は、電子写真制御部17に接続されている。

## 【 0 0 2 3 】

[ CPU 1 1 に適用される通信タスクについて ]

次に CPU 1 1 に適用される F / W (ファームウェア) である通信タスクを説明する。

## 【 0 0 2 4 】

まず、状態定義と F / W の制御ブロックであるところのタスク (以下 T A S K ) とハードウェアとのデータの受け渡しをるところのドライバ (以下ドライバ) とデータの一時記憶場所であるバッファ (以下 B u f f e r ) の説明をする。

## 【 0 0 2 5 】

< 定義 >

同期状態と通常状態について、以下のように定義する。

10

同期状態： CPU 1 1 からステータス要求であるところのステータスリクエスト (以下ステータス R e q u e s t ) を送信し、そのステータス R e q u e s t に対応する正しいステータスを CPU 1 1 が受信するまでの間または CPU 1 1 から実動作を要求するところの実行コマンド (以下実行コマンド) を送信し、その実行コマンドに対応する正しいステータスを CPU 1 1 が受信するまでの間。

通常状態： CPU 1 1 が上記同期状態にない場合。

## 【 0 0 2 6 】

< T A S K とドライバ >

T A S K 2 個とドライバ 2 個から通信ソフトが構成されている。

各々の名前を、“コマンド・ステータス制御 T A S K ” ( 1 0 1 )、“第 1 受信 T A S K ” ( 1 0 2 )、“送信ドライバ” ( 1 0 3 )、“受信ドライバ” ( 1 0 4 ) とする。これら、1 0 1 ~ 1 0 4 の機能は以下のとおりである。

20

## 【 0 0 2 7 】

( 1 0 1 ) “コマンド・ステータス制御 T A S K ”

以下にこの T A S K の機能を列挙する。

\* 送信すべきステータス R e q u e s t または実行コマンド (コマンド信号) を制御し、なおかつ、どのコマンドを送信したかを記憶する。

\* 返信されたステータスを所定のメモリに格納する。

\* 再送すべきステータス R e q u e s t または実行コマンドを制御する。

\* N A C K (ネガティブアクナリッジ信号) を返信する。

30

\* 受信した N A C K を無視する。

## 【 0 0 2 8 】

( 1 0 2 ) “第 1 受信 T A S K ”

以下にこの T A S K の機能を列挙する。

\* R x D a t a (後述) の内容が CPU 1 6 からのステータスを表示するための S T C o m m a n d (以下 S T C o m m a n d ) であると判断したら A C K (アクナリッジ信号) を返信する。

\* R x D a t a (後述) の内容が CPU 1 6 からの S T C o m m a n d でない場合には、そのデータを T x R x B u f f e r (後述) を通じてコマンド・ステータス制御 T A S K 1 0 1 にデータ内容を渡す。

40

## 【 0 0 2 9 】

( 1 0 3 ) “送信ドライバ”

以下にこの T A S K の機能を列挙する。

\* コマンド・ステータス制御 T A S K 1 0 1 から依頼のあった送信を実行する。

\* 第 1 受信 T A S K 1 0 2 から依頼のあった A C K を送信する。

## 【 0 0 3 0 】

( 1 0 4 ) “受信ドライバ”

以下にこの T A S K の機能を列挙する。

\* 受信したデータを R x D a t a を通じて第 1 受信 T A S K 1 0 2 に渡す。

## 【 0 0 3 1 】

50

## &lt; Buffer &gt;

各TASK間での情報の受け渡しに使用するBufferを説明する。

## 【0032】

各々の名前を、RxData、TxRxBuffer、TxQueue、SR\_CommandQueue、NormalStatus、STCommandSaveAreaとする。

201RxData：受信ドライバ104が受信したデータをRxDataに格納し、第1受信TASK102に受け渡すために使用するメモリ領域である。

202TxRxBuffer：第1受信TASK102からコマンド・ステータス制御TASK101に受信データを受け渡すために使用するメモリ領域である。

203TxQueue：コマンド・ステータス制御TASK101または、第1受信TASK102から送信データを送信ドライバ103に通知するために使用するメモリ領域である。

204SR\_CommandQueue：通信TASK以外の部分(151)からのステータスRequest、実行コマンドの送信要求を格納するメモリ領域である。

205NormalStatus：通常の状態の格納エリアである。通信TASK以外の部分151から参照可能となっている。

206STCommandSaveArea：CPU16から到着したSTCommandを格納するメモリ領域である。通信TASK以外の部分から参照可能となっている。

## 【0033】

## &lt; CPU11の通信TASK構造の説明 &gt;

図3に従って、CPU11の通信TASK構造を説明する。

## 【0034】

CPU11のRXポートからのデータを処理する部分が受信ドライバ104である。受信ドライバ104で受信されたデータはRxData201に格納される。次に、RxData201のデータを解析するのが、第1受信TASK102である。この第1受信TASK102は、RxData201に格納されたデータがCPU16から送信されたSTCommandであるかどうかをチェックする(110)。STCommandであった場合はSTCommandSaveArea206に格納する(109)。通信TASK以外の部分151がSTCommandSaveArea206の内容を参照する。そして、ACKをCPU16に返信する。具体的には、送信バッファであるTxQueue203にACKをセットし、送信ドライバ103からTxQueue203に格納された送信データを送信する。

## 【0035】

RxData201がSTCommandでなかった場合には、そのデータのエラー発生状況を含めて、TxRxBuffer202に格納する。次のコマンド・ステータス制御TASK101では、TxRxBuffer202に格納されたデータを解析する。そして、NACKを返信するか(108)、ステータスRequestまたは、実行コマンドの再送信をするか(107)、正常であると判断し、NormalStatus205にステータスを格納するか(106)を選択する。また、通信TASK以外の部分151がNormalStatus205の内容を参照する。また、SR\_CommandQueue204に通信TASK以外の部分151から書き込まれたデータがある場合には(105)、送信バッファであるTxQueue203に送信データをセットする。最後に送信ドライバ103では、TxQueue203に格納された送信データを順次送信する。送信ドライバ103は、CPU11のTXポートへのデータを処理する部分である。

## 【0036】

## ~受信ドライバ104~

図4にて、受信ドライバ104のフローチャートを説明する。

## 【0037】

受信ドライバ104では、常に、ハードウェアからの受信割り込みであるところの受信

10

20

30

40

50

Interrupt (受信Interrupt)が発生したかどうかを監視している(ステップ354、以下ステップをSとする)。もしも受信Interruptが発生したら(Interrupt受信)、受信データをRxData201に格納する(S355)。

【0038】

～第1受信TASK102～

図5にて第1受信TASK102のフローチャートを説明する。

【0039】

まず、RxData201にデータがあるかどうか、すなわちRxData201が空か否かを監視する(S321)。もしも、空ではない、すなわちデータが格納されたRxData201が到着したらデータの解析を開始する。

10

【0040】

まず、RxData201を確認し、通信エラーが発生していないかどうかを確認する(S322)。もしも、エラーが発生している場合には、TxRxBuffer202にRxData201のデータを格納する。そしてその場合には、エラーが発生している情報も含める(S329)。

【0041】

通信エラーが発生していない場合には、RxData201がCPU16から通知されたSTCommandであるかどうかを確認する(S323)。もしもSTCommandであった場合には、STCommandSaveArea206に到着したSTCommandを格納する(S324)。そして、CPU16にSTCommandを正常に受信したことを通知するために、ACKを返信する。そのために、TxQueue203にACKをセットする(S325)。

20

【0042】

RxData201がSTCommandでない場合には、NACKであるかどうかを確認する(S326)。もしもNACKであった場合には、TxRxBuffer202にNACKを格納し、コマンド・ステータス制御TASK101に渡す(S328)。もしもRxData201がNACKでない場合には、格納されたRxData201は、通信エラーでもNACKでもないことが確定するので、TxRxBuffer202にエラー発生なしの情報も含めてRxData201の内容を格納する(S327)。

30

【0043】

なお、このフローチャートの中のS323, S324, S325の処理が前述の〔3〕に該当する処理である。

【0044】

～コマンド・ステータス制御TASK101～

図6にてコマンド・ステータス制御TASK101のフローチャートを説明する。

【0045】

SR\_CommandQueue204の中に、ステータスRequestまたは実行コマンドの送信要求があるかどうかを確認する(S301)。

【0046】

40

もしもSR\_CommandQueue204が空の場合には、TxRxBuffer202の確認を行う(S309)。もしもTxRxBuffer202にも到着データがない場合には、SR\_CommandQueue204の確認に戻る(S309)。もしも、TxRxBuffer202に受信データが格納されていた場合には、まず、通信エラーが発生していないかを確認する(S310)。もしも通信エラーが発生していた場合には、TxQueue203にNACKをセットする(S312)。通信エラーが発生していない場合には、TxRxBuffer202がNACKかどうかを確認する(S311)。そして、データがNACKの場合には、そのNACKを無視し、SR\_CommandQueue204の確認に戻る(S311)。

【0047】

50

もしも、TxRxBuffer 202のデータがNACKでない場合には、CPU 16にNACKを返信しなければいけないので、TxQueueにNACKをセットする(S 312)。

【0048】

なお、S 309, S 310, S 311, S 312の処理を実行している状態が通常状態である。

【0049】

そして、S 310, S 312の処理が前述の〔7〕に該当する処理である。また、S 311, S 312の処理が前述の〔15〕に該当する処理である。また、S 311からS 301へ戻る処理が前述の〔11〕に該当する処理である。

10

【0050】

次に同期状態の説明をする。

【0051】

SR\_CommandQueue 204の中に、ステータスRequestまたは実行コマンドの送信要求がある場合には(S 301 No)、TxQueue 203にステータスRequestまたは実行コマンドをセットする(S 302)。このとき、再送信が発生した場合に備えて発行したステータスRequestまたは実行コマンドを記憶する(S 302)。

【0052】

CPU 16から返信であるTxRxBuffer 202の到着を確認する(S 303)。そして、このデータの確認はタイムオーバーであるところのTime-Over(以下Time-Over)も同時に監視する(S 303)。

20

【0053】

もしも、Time-Overが発生したら(S 303 Yes, S 304 Yes)、再送の準備をする。すなわち、TxQueue 203に再送データを格納できるような準備をする(S 307)。再送は、S 302にて記憶したデータを使用する(S 307)。

【0054】

S 304でTime-Overが発生しなかった場合には、次に通信エラーが発生しているかどうか確認する(S 305)。

【0055】

通信エラーが発生している場合には(S 305 Yes)、再送の準備をする。すなわち、TxQueue 203に再送データを格納できるような準備をする(S 307)。再送は、S 302にて記憶したデータを使用する(S 307)。

30

【0056】

次に、TxRxBuffer 202を確認し、NACKであるかどうかを確認する(S 306)。NACKであった場合には、再送の準備をする。すなわち、TxQueue 203に再送データを格納できるような準備をする(S 307)。再送は、S 302にて記憶したデータを使用する(S 307)。

【0057】

そして、NACKでない場合は(S 306 No)、TxRxBuffer 202に通常のステータスが格納されているので、それをNormalStatus 205に格納する(S 308)。

40

【0058】

S 308を通過しない限り、次のSR\_CommandQueue 204のチェックを実施しない。すなわち、正常なACKまたはStatusが受信されない限り次の新しいステータスRequestまたは実行コマンドが発行できないようになっている。すなわち、正常なACKまたは正常なStatusが返信されるまでは実行コマンドの発行を停止する。

【0059】

なお、S 308が前述の〔5〕に該当する処理である。また、S 304, S 307の処

50

理が前述の〔17〕に該当する処理である。S304, S305, S307の処理が前述の〔9〕に該当する処理である。S304, S305, S306, S307の処理が前述の〔13〕に該当する処理である。S304, S305, S306, S308, S301の処理が前述の〔5〕に該当する処理である。

【0060】

～送信ドライバ103～

図7にて送信ドライバ103のフローチャートを説明する。

【0061】

TxQueue203にデータが格納されているかどうか、すなわち、空か否かを確認する(S351)。

10

【0062】

もしも、TxQueue203にデータがある場合には、送信データをハードウェアに書き込み、実際の送信動作を実行する(S352)。

【0063】

そして、送信動作が完了するのを待つ(S353)。送信動作が完了したらTxQueue203の確認動作に戻る(S353)。

【0064】

[CPU16に適用される通信タスクについて]

次にCPU16に適用されるF/Wである通信タスクを説明する。

【0065】

まず、定義とTASKとドライバとBufferの説明をする。

20

【0066】

<定義>

同期状態と通常状態について、以下のように定義する。

同期状態：CPU16からSTCommandを送信し、そのコマンドに対応する正しいACKが返信されるまでの間。

通常状態：CPU16が上記同期状態にない場合。

【0067】

<TASKとドライバ>

TASK2個とドライバ2個から通信ソフトが構成されている。

30

【0068】

(401) コマンド・ステータス制御TASK

以下にこのTASKの機能を列挙する。

\*送信すべきSTCommandを制御し、なおかつ、どのSTCommandを送信したかを記憶する。

\*返信されたACKを確認する。

\*再送すべきSTCommandを制御する。

\*NACKを返信する。

\*受信したNACKを無視する。

【0069】

(402) 第1受信TASK

以下にこのTASKの機能を列挙する。

\*RxDataの内容がCPU16からのステータスRequestであると判断したら対応するステータスを返信する。

\*Rxdataの内容がCPU16からの実行コマンドであると判断したら実行コマンド受信情報505に格納し、さらに対応するステータスを返信する。

\*RxDataの内容がCPU16からのステータスRequestまたは実行コマンドでない場合には、そのデータをTxRxBuffer502を通じてコマンド・ステータス制御TASK401に渡す。

40

【0070】

50

## (403) 送信ドライバ

以下にこのT A S Kの機能を列挙する。

- \* コマンド・ステータス制御T A S K 4 0 1から依頼のあった送信を実行する。
- \* 第1受信T A S K 4 0 2から依頼のあったステータスを送信する。

## 【0071】

## (404) 受信ドライバ

以下にこのT A S Kの機能を列挙する。

- \* 受信したデータをR x D a t a 5 0 1を通じて第1受信T A S K 4 0 2に渡す。

## 【0072】

< B u f f e r >

各T A S K間での情報の受け渡しに使用するB u f f e rを説明する。

R x D a t a 5 0 1 : 受信ドライバ4 0 4が受信したデータをR x D a t a 5 0 1に格納し、第1受信T A S K 4 0 2に受け渡すために使用するメモリ領域である。

T x R x B u f f e r 5 0 2 : 第1受信T A S K 4 0 2からコマンド・ステータス制御T A S K 4 0 1に受信データを受け渡すために使用するメモリ領域である。

T x Q u e u e 5 0 3 : コマンド・ステータス制御T A S K 4 0 1または、第1受信T A S K 4 0 2から送信データを送信ドライバに通知するために使用するメモリ領域である。

S T C o m m a n d Q u e u e 5 0 4 : 通信T A S K以外の部分4 5 1からのS T C o m m a n dの送信要求データを格納するために使用するメモリ領域である。

実行コマンド受信情報5 0 5 : 実行コマンドの受信データ格納エリアである。通信T A S K以外の部分4 5 1から参照可能となっている。

## 【0073】

< C P U 1 6の通信T A S K構造の説明 >

図8に従って、C P U 1 6の通信T A S K構造を説明する。

## 【0074】

C P U 1 6のR Xポートからのデータを処理する部分が受信ドライバ4 0 4である。受信ドライバ4 0 4で受信されたデータはR x D a t a 5 0 1に格納される。次に、R x D a t a 5 0 1のデータを解析するのが、第1受信T A S K 4 0 2である。この第1受信T A S K 4 0 2は、R x D a t a 5 0 1に格納されたデータがC P U 1 1から送信された実行コマンドまたはステータスR e q u e s tであるかどうかをチェックする(410)。実行コマンドであった場合は、実行コマンド受信情報5 0 5のB u f f e rに情報を格納する。そしてその実行コマンドに対応するステータスを返信する。ステータスR e q u e s tであった場合は対応するステータスを返信する。

## 【0075】

通信T A S K以外の部分4 5 1が実行コマンド受信情報5 0 5を参照する。R x D a t a 5 0 1が実行コマンドまたはステータスR e q u e s tでなかった場合には、そのデータのエラー発生状況を含めて、T x R x B u f f e r 5 0 2に格納する。次のコマンド・ステータス制御T A S K 4 0 1では、T x R x B u f f e r 5 0 2に格納されたデータを解析する。そして、N A C Kを返信するか(408)、S T C o m m a n dの再送信をするか(407)、正常にA C Kを受信したと判断し(406)、次のS T C o m m a n dの送信を許可するかを選択する。また、S T C o m m a n d Q u e u e 5 0 4に通信T A S K以外の部分4 5 1から書き込まれたデータがある場合には、送信バッファであるT x Q u e u e 5 0 3に送信データをセットする。最後に送信ドライバ4 0 3では、T x Q u e u e 5 0 3に格納された送信データを順次送信する。

## 【0076】

~ 受信ドライバ4 0 4 ~

図9にて、受信ドライバ4 0 4のフローチャートを説明する。

## 【0077】

受信ドライバ4 0 4では、常に、ハードウェアからの受信I n t e r r u p tが発生し

10

20

30

40

50

たかどうかを監視している ( S 6 5 4 )。

【 0 0 7 8 】

もしも受信 `Interrupt` が発生したら、受信データを `RxDat a 5 0 1` に格納する ( S 6 5 5 )。

【 0 0 7 9 】

～ 第 1 受信 `T A S K 4 0 2` ～

図 1 0 にて第 1 受信 `T A S K 4 0 2` のフローチャートを説明する。

【 0 0 8 0 】

まず、`RxDat a 5 0 1` にデータがあるかどうか、すなわち空か否かを監視する ( S 6 2 1 )。もしも、空ではない `RxDat a 5 0 1` が到着したらデータの解析を開始する。

10

【 0 0 8 1 】

まず、`RxDat a 5 0 1` を確認し、通信エラーが発生していないかどうかを確認する ( S 6 2 2 )。もしも、エラーが発生している場合には、`T x R x B u f f e r 5 0 2` に `RxDat a 5 0 1` のデータを格納する。そしてその場合には、エラーが発生している情報も含める ( S 6 2 9 )。

【 0 0 8 2 】

通信エラーが発生していない場合には、`RxDat a 5 0 1` が `C P U 1 1` から通知された実行コマンドまたはステータス `Request` であるかどうかを確認する ( S 6 2 3 )。もしも実行コマンドまたはステータス `Request` であった場合には、さらに、実行コマンドであるかステータス `Request` であるかを判断する ( S 6 8 1 )。実行コマンドが到着した場合には、実行コマンド受信情報 `5 0 5` にデータを格納する ( S 6 8 2 )。そして、その実行コマンドに対応するステータスを通信 `T A S K` 以外の部分 `4 5 1` から取得する ( S 6 8 4 )。そしてそのステータスを `T x Q u e u e 5 0 3` にセットする ( S 6 8 5 )。

20

【 0 0 8 3 】

ステータス `Request` が到着した場合には、そのステータス `Request` に対応するステータスを通信 `T A S K` 以外の部分 `4 5 1` から取得する ( S 6 2 4 )。そしてそのステータスを `T x Q u e u e 5 0 3` にセットする ( S 6 2 5 )。

【 0 0 8 4 】

`RxDat a 5 0 1` が実行コマンドまたはステータス `Request` でない場合には ( S 6 2 3 `No` )、`NACK` であるかどうかを確認する ( S 6 2 6 )。もしも `NACK` であった場合には、`T x R x B u f f e r 5 0 2` に `NACK` を格納し、コマンド・ステータス制御 `T A S K 4 0 1` に渡す ( S 6 2 8 )。もしも `RxDat a 5 0 1` が `NACK` でない場合には、格納された `RxDat a 5 0 1` は、通信エラーでも `NACK` でもないことが確定するので、`T x R x B u f f e r 5 0 2` にエラー発生なしの情報も含めて `RxDat a 5 0 1` の内容を格納する ( S 6 2 7 )。

30

【 0 0 8 5 】

このフローチャートの中の `S 6 2 3` , `S 6 8 1` , `S 6 2 4` , `S 6 2 5` の処理、および `S 6 2 3` , `S 6 8 1` , `S 6 8 2` , `S 6 8 4` , `S 6 8 5` の処理が前述の〔 4 〕に該当する処理である。

40

【 0 0 8 6 】

～ コマンド・ステータス制御 `T A S K 4 0 1` ～

図 1 1 にてコマンド・ステータス制御 `T A S K 4 0 1` のフローチャートを説明する。

【 0 0 8 7 】

`S T C o m m a n d Q u e u e 5 0 4` の中に、`S T C o m m a n d` の送信要求があるかどうかを確認する ( S 6 0 1 )。

【 0 0 8 8 】

もしも `S T C o m m a n d Q u e u e 5 0 4` が空の場合には、`T x R x B u f f e r 5 0 2` の確認を行う ( S 6 0 9 )。もしも `T x R x B u f f e r 5 0 2` にも到着データがな

50

い場合には、STCommandQueue504の確認に戻る(S609)。もしも、TxRxBuffer502に受信データが格納されていた場合には、まず、通信エラーが発生していないかを確認する(S610)。もしも通信エラーが発生していた場合にはTxQueue503にNACKをセットする(S612)。通信エラーが発生していない場合には、TxRxBuffer502がNACKかどうかを確認する(S611)。そして、データがNACKの場合には、そのNACKを無視し、STCommandQueue504の確認に戻る(S611)。

【0089】

もしも、TxRxBuffer502のデータがNACKでない場合には、CPU11にNACKを返信しなければいけないので、TxQueue503にNACKをセットする(S612)。

10

【0090】

なお、S609、S610、S611、S612の処理を実行している状態が通常状態である。

【0091】

そして、S610、S612の処理が前述の〔8〕に該当する処理である。また、S611、S612の処理が前述の〔16〕に該当する処理である。S611からS601へ戻る処理が前述の〔12〕に該当する処理である。

【0092】

次に同期状態の説明をする。

20

【0093】

STCommandQueue504の中に、STCommandの送信要求がある場合には(S601 No)、TxQueue503にSTCommandをセットする(S602)。このとき、再送信が発生した場合に備えて発行したSTCommandを記憶する(S602)。

【0094】

CPU11からの返信であるTxRxBuffer502を確認する(S603)。そして、このデータの確認はTime-Overも同時に監視する(S603)。

【0095】

もしも、Time-Overが発生したら(S603 Yes、S604 Yes)、再送の準備をする。すなわち、TxQueue503に再送データを格納できるような準備をする(S607)。再送は、S602にて記憶したデータを使用する(S607)。

30

【0096】

Time-Overが発生しなかった場合には(S604 No)、次に通信エラーが発生しているかどうか確認する(S605)。

【0097】

通信エラーが発生している場合には(S605 Yes)再送の準備をする。すなわち、TxQueue503に再送データを格納できるような準備をする(S607)。再送は、S602にて記憶したデータを使用する(S607)。

【0098】

次に、TxRxBuffer502を確認し、NACKであるかどうかを確認する(S606)。NACKであった場合には、再送の準備をする。すなわち、TxQueue503に再送データを格納できるような準備をする(S607)。再送は、S602にて記憶したデータを使用する(S607)。

40

【0099】

そして、NACKでない場合は(S606 No)、TxRxBuffer502に通常のステータスが格納されている(S606)。

【0100】

S606を通過しない限り、次のSTCommandQueue504のチェックを実施しない。すなわち、正常なACKが受信されない限り次の新しいSTCommandが

50

発行できないようになっている（S606）。この部分が前述の〔6〕に該当する処理である。

【0101】

なお、S604、S607の処理が前述の〔18〕に該当する処理である。また、S604、S605、S607の処理が前述の〔10〕に該当する処理である。S604、S605、S606、S607の処理が前述の〔14〕に該当する処理である。S604、S605、S606、S601の処理が前述の〔6〕に該当する処理である。

【0102】

～送信ドライバ403～

図12にて送信ドライバ403のフローチャートを説明する。

10

【0103】

TxQueue503にデータが格納されているかどうか、すなわち空か否かを確認する（S651）。

【0104】

もしも、TxQueueにデータがある場合には、送信データをハードウェアに書き込み、実際の送信動作を実行する（S652）。

【0105】

そして、送信動作が完了するのを待つ（S653）。送信動作が完了したら（S653）、TxQueue503の確認動作に戻る（S653）。

【0106】

このように本実施例によれば、ネガティブアクナリッジ信号（NACK）に対しネガティブアクナリッジ信号（NACK）を再送する状況を避け、通信のハングアップを防止することができる。また、片方からの通信によりハンドシェイクを実行中にも、完全に双方向の通信を実行できる。

20

【実施例2】

【0107】

実施例2の構成は実施例1と同一であり、本実施例の画像形成装置の構成は図1で示される。またCPUの接続も同一であり、図2で示される。

【0108】

なお、本実施例の通信プロトコルは、実施例1の〔1〕～〔18〕の通信形態を有し、さらに、下記〔19〕～〔22〕の通信プロトコルが付加されたものである。

30

〔19〕A-CPUがCommandを発行する場合には、Commandに識別コードをA-CPUが割り振り、B-CPUが、ACKおよびStatusを返信するときには、受信した識別コードをACK、および、Statusの中にコピーして返信する。

〔20〕B-CPUがCommandを発行する場合には、Commandに識別コードをB-CPUが割り振り、A-CPUが、ACKおよびStatusを返信するときには、受信した識別コードをACK、および、Statusの中にコピーして返信する。

〔21〕A-CPUが識別コードを保存した状態で、Commandを発行しその後、受信したACK、またはStatusの中に記載された識別コードが保存してあったCommand用の識別コードと異なっていた場合には、NACKを送信する。

40

〔22〕B-CPUが識別コードを保存した状態で、Commandを発行しその後、受信したACK、またはStatusの中に記載された識別コードが保存してあったCommand用の識別コードと異なっていた場合には、NACKを送信する。

【0109】

<CPU11の通信TASK構造の説明>

図13から図17が実施例2におけるCPU11のTASK構成およびフローチャートである。

【0110】

パケット識別コード（識別コード）であるところのパケットID（以下パケットID）の付加およびパケットIDの不一致の発生状況確認以外の処理は実施例1と同一である。

50

以下、実施例 1 と同一のものについては同一の符号を用いて説明する。

【 0 1 1 1 】

図 1 3 で示すように、S T C o m m a n d、A C K、ステータス、ステータス R e q u e s t、実行コマンド等には、すべてパケットを特定するための固有番号であるパケット I D が付加されている（図 3 と比較参照）。パケットに付加されたパケット I D により通信が正常に実施されたかどうかの確認を行う。

【 0 1 1 2 】

パケット I D の決定は、1 0 5 の処理の中で実施される。具体的には、パケット I D が 1 ずつ増加される。すなわち I n c r e m e n t（以下 I n c r e m e n t）される。

【 0 1 1 3 】

～受信ドライバ 1 0 4～

図 1 4 の S 3 7 5 に示すように、受信はパケット単位で行われ、R x D a t a 2 0 1 への格納もパケット単位で行われ、以降のデータ解析もパケット単位で実施される。

【 0 1 1 4 】

～第 1 受信 T A S K 1 0 2～

図 1 5 で示すように、S 3 8 5 で A C K を返信する場合には、S T C o m m a n d にて通知されたパケット I D をそのまま付加する。この処理が前述の〔 1 9 〕に対応する処理である。

【 0 1 1 5 】

～コマンド・ステータス制御 T A S K 1 0 1～

図 1 6 で示すように、S 3 8 2 で送信したパケット I D を記憶するとともに、次に使用するパケット I D が I n c r e m e n t されたものとなるように準備する。

【 0 1 1 6 】

また、S 3 8 8 では、受信したパケット I D と送信したパケット I D の比較が実施され、より精度のよい通信エラーの検知を実施している。もしも、パケット I D の不一致が発生した場合には（S 3 8 8 N o）、N A C K を返信するために、T x Q u e u e 2 3 0 に N A C K をセットする（S 3 8 9）。この処理が前述の〔 2 1 〕に対応する処理である。

【 0 1 1 7 】

～送信ドライバ 1 0 3～

図 1 7 で示すように、送信ドライバ 1 0 3 の動作は実施例 1 と同一である。

【 0 1 1 8 】

< C P U 1 6 の通信 T A S K 構造の説明 >

図 1 8 から図 2 2 が実施例 2 における C P U 1 6 の T A S K 構成およびフローチャートである。

【 0 1 1 9 】

パケット I D の付加およびパケット I D の不一致の発生状況確認以外の処理は実施例 1 と同一である。

【 0 1 2 0 】

図 1 8 で示すように、ステータス R e q u e s t、実行コマンド、ステータス、A C K、S T C o m m a n d 等にはすべてパケット I D が付加されている。パケットに付加されたパケット I D により通信が正常に実施されたかどうかの確認を行う。

【 0 1 2 1 】

パケット I D の決定は、4 0 5 の処理の中で実施される。具体的には、パケット I D が I n c r e m e n t される。

【 0 1 2 2 】

～受信ドライバ 4 0 4～

図 1 9 の S 6 7 5 で示すように、受信はパケット単位で行われ、R x D a t a 5 0 1 への格納もパケット単位で行われ、以降のデータ解析もパケット単位で実施される。

【 0 1 2 3 】

10

20

30

40

50

～ 第 1 受信 T A S K 4 0 2 ～

図 2 0 で示すように、S 6 7 5、S 6 7 6 で、返信すべきステータスには、ステータス R e q u e s t のパケット I D あるいは実行コマンドのパケット I D をそのまま付加する。この処理が前述の〔 2 0 〕に対応する処理である。

【 0 1 2 4 】

～ コマンド・ステータス制御 T A S K 4 0 1 ～

図 2 1 で示すように、S 6 4 2 では、送信したパケット I D を記憶するとともに、次に使用するパケット I D が I n c r e m e n t されたものとなるように準備する。

【 0 1 2 5 】

S 6 9 7 では、受信したパケット I D と送信したパケット I D の比較が実施され、より精度のよい通信エラーの検知を実施している。もしも、パケット I D の不一致が発生した場合には ( S 6 9 7 N o )、N A C K を返信するために、T x Q u e u e 2 3 0 に N A C K をセットする ( S 6 9 8 )。この処理が前述の〔 2 2 〕に対応する処理である。

【 0 1 2 6 】

～ 送信ドライバ 4 0 3 ～

図 2 2 で示すように、送信ドライバ 4 0 3 の動作は実施例 1 と同一である。

【 0 1 2 7 】

このように、本実施例によれば、通信エラーの発生をより詳しく検知することができる。

【 実施例 3 】

【 0 1 2 8 】

実施例 3 は実施例 2 と比較して、第 1 受信 T A S K 1 0 2 と第 1 受信 T A S K 4 0 2 のフローチャートのみが変更となっている。以下、実施例 2 と同じものについては同じ符号を用いて説明する。

【 0 1 2 9 】

なお、本実施例の通信プロトコルは、実施例 1 の〔 1 〕～〔 1 8 〕、実施例 2 の〔 1 9 〕～〔 2 2 〕の通信形態を有し、さらに、下記〔 2 3 〕、〔 2 4 〕の通信プロトコルが付加されたものである。

〔 2 3 〕 A - C P U が C o m m a n d を受信した時点で、識別コードを保存する。A - C P U は C o m m a n d に対応した動作を起動する。A - C P U は C o m m a n d に対応する A C K または S t a t u s を B - C P U に返信する。A - C P U が次の C o m m a n d を受信した場合に、もしも、C o m m a n d に記載された識別コードが前回の C o m m a n d 受信時に保存した識別コードと一致した場合には、A - C P U は、C o m m a n d に対応した動作は起動しない。そして C o m m a n d に対応する A C K または S t a t u s を返信する。

〔 2 4 〕 B - C P U が C o m m a n d を受信した時点で、識別コードを保存する。B - C P U は C o m m a n d に対応した動作を起動する。B - C P U は C o m m a n d に対応する A C K または S t a t u s を A - C P U に返信する。B - C P U が次の C o m m a n d を受信した場合に、もしも、C o m m a n d に記載された識別コードが前回の C o m m a n d 受信時に保存した識別コードと一致した場合には、B - C P U は、C o m m a n d に対応した動作は起動しない。そして C o m m a n d に対応する A C K または S t a t u s を返信する。

【 0 1 3 0 】

～ 第 1 受信 T A S K 1 0 2 ～

図 2 3 の S 3 4 3 で示すように、到着した S T C o m m a n d のパケット I D が前回 ( 直前に ) 受信したパケット I D と同一であるかどうかを確認する。もしも、パケット I D が前回受信したパケット I D と同一であれば、再送信が発生したのものとして、S T C o m m a n d S a v e A r e a 2 0 6 には格納しない ( S 3 4 3 Y e s , S 3 4 4 )。この処理が前述の〔 2 3 〕に対応する処理である。この処理をすることにより、再送信によるステータスの破壊を防止する。

10

20

30

40

50

【0131】

一方、S343で、直前に受信したパケットIDと今回のパケットIDが同一ではない場合、S344でSTCommandSaveArea206にSTCommandを格納し、パケットIDも記憶する。

【0132】

～第1受信TASK402～

図24のS699で示すように、今回通知された到着した実行コマンドのパケットIDが記憶している前回受信した実行コマンドのパケットIDと同一であるかどうかを確認する。もしも、パケットIDが前回受信した実行コマンドのパケットIDと同一であれば、再送信が発生したものととして、実行コマンド受信情報505には書き込まない(S699 Yes)。このように、実行コマンド受信情報505には書き込まず、記憶した前回のステータスをTxQueue503にセットする準備をする(S701)。この処理が前述の〔24〕に対応する処理である。

10

【0133】

この処理をすることにより、再送信による不必要な動作の起動を回避できる。そして、再送信が発生したと認識した場合には、記憶してある前回のステータスを返信する。

【0134】

一方、S699で記憶しているパケットIDと今回通知されたパケットIDが同一ではない場合は、S639で実行コマンド受信情報505に実行コマンド情報を格納し、今回通知されたパケットIDも記憶する。

20

【0135】

以上説明したように、本実施例によれば、通信ごとのパケット識別コード(パケットID)を確認することにより、不必要な動作を誤って二重起動することがなくなるので、通信エラーを起因とするハードウェアトラブルを防止することができる。

【図面の簡単な説明】

【0136】

【図1】本発明の画像形成装置本体を説明する図

【図2】エンジン基板15のCPU16とビデオコントローラ部10のCPU11の接続図

【図3】実施例1のCPU11の通信TASK構造図

30

【図4】実施例1の受信ドライバ104のフローチャート

【図5】実施例1の第1受信TASK102のフローチャート

【図6】実施例1のコマンド・ステータス制御TASK101のフローチャート

【図7】実施例1の送信ドライバ103のフローチャート

【図8】実施例1のCPU16の通信TASK構造図

【図9】実施例1の受信ドライバ404のフローチャート

【図10】実施例1の第1受信TASK402のフローチャート

【図11】実施例1のコマンド・ステータス制御TASK401のフローチャート

【図12】実施例1の送信ドライバ403のフローチャート

【図13】実施例2のCPU11の通信TASK構造図

40

【図14】実施例2の受信ドライバ104のフローチャート

【図15】実施例2の第1受信TASK102のフローチャート

【図16】実施例2のコマンド・ステータス制御TASK101のフローチャート

【図17】実施例2の送信ドライバ103のフローチャート

【図18】実施例2のCPU16の通信TASK構造図

【図19】実施例2の受信ドライバ404のフローチャート

【図20】実施例2の第1受信TASK402のフローチャート

【図21】実施例2のコマンド・ステータス制御TASK401のフローチャート

【図22】実施例2の送信ドライバ403のフローチャート

【図23】実施例3の第1受信TASK102のフローチャート

50

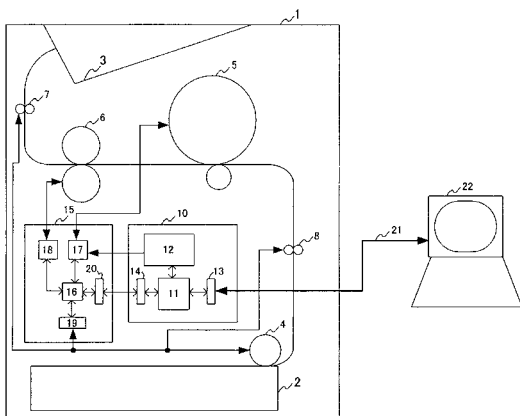
【図24】実施例3の第1受信TASK402のフローチャート

【符号の説明】

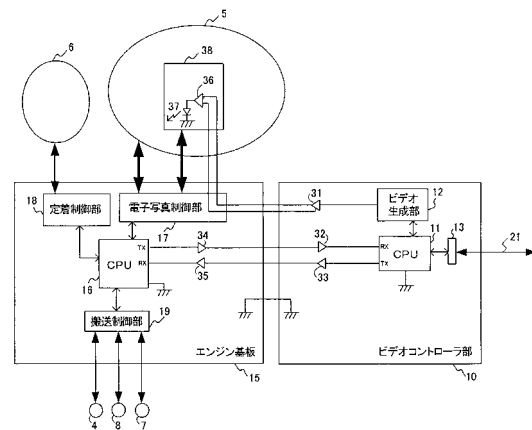
【0137】

- 1 画像形成装置本体
- 10 ビデオコントローラ部
- 11 CPU
- 15 エンジン基板
- 16 エンジンCPU
- 32, 33, 34, 35 シリアル通信用バッファIC

【図1】

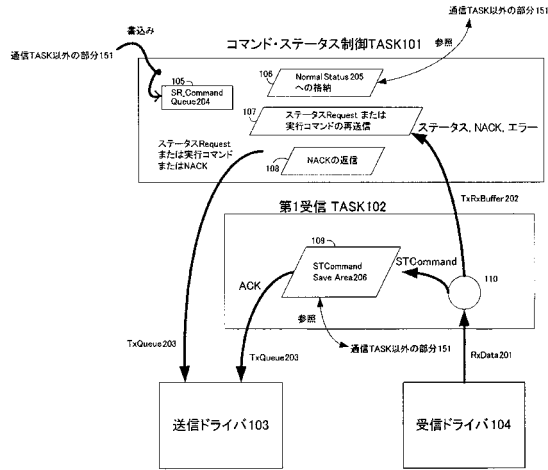


【図2】



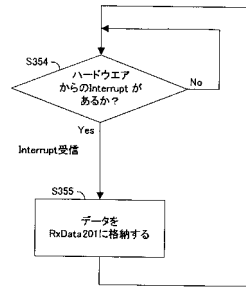
【図3】

CPU11 通信TASK構造



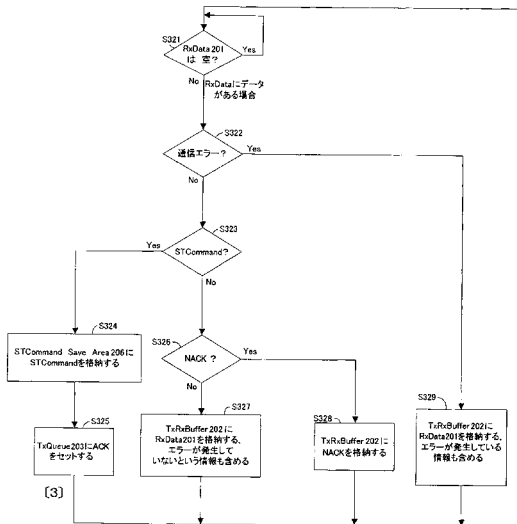
【図4】

受信ドライバ104



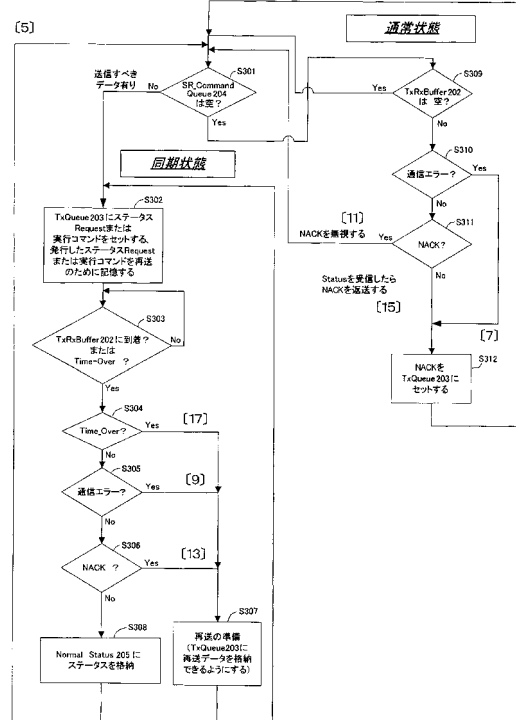
【図5】

第1受信 TASK102



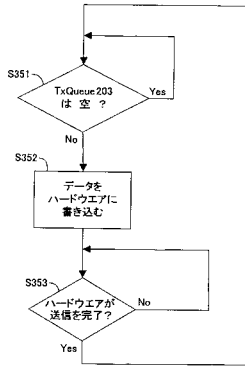
【図6】

コマンド・ステータス制御TASK101



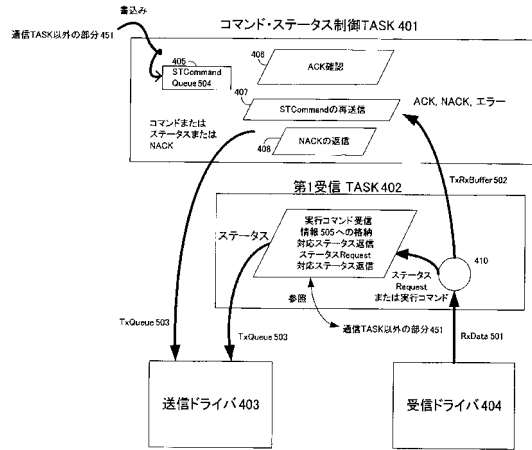
【 図 7 】

送信ドライバ103



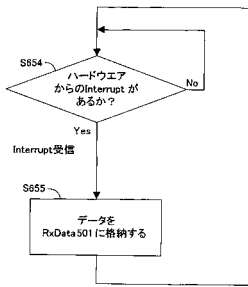
【 図 8 】

CPU16 通信TASK構造



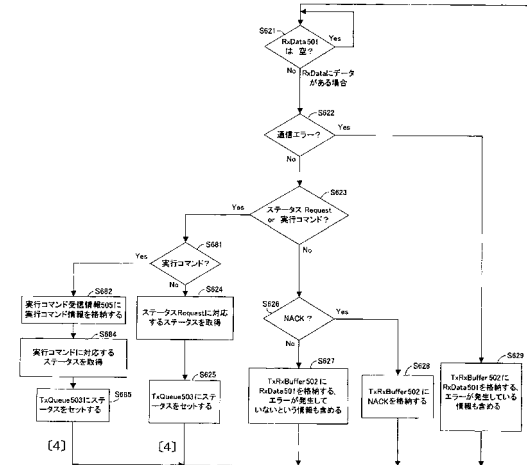
【 図 9 】

404受信ドライバ

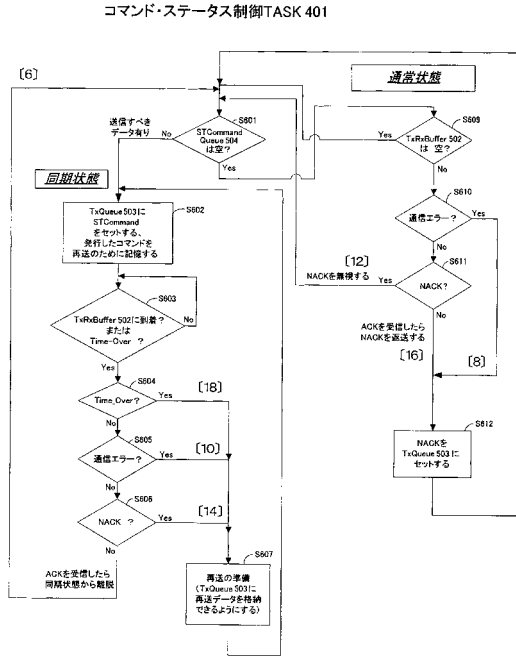


【 図 10 】

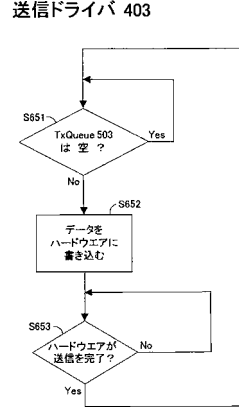
第1受信 TASK 402



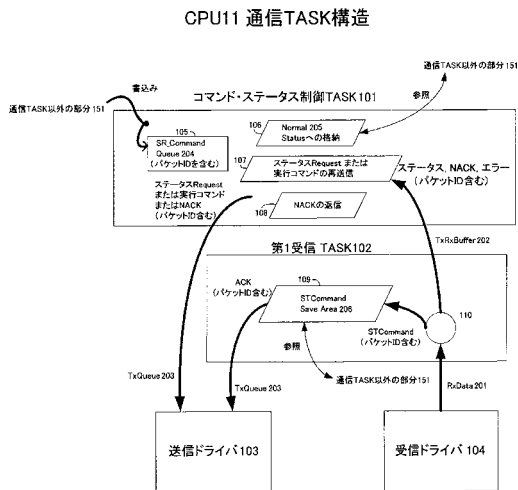
【図 1 1】



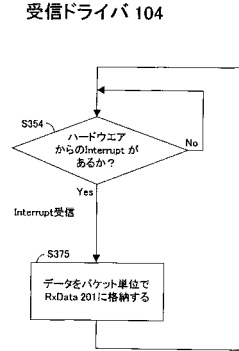
【図 1 2】



【図 1 3】

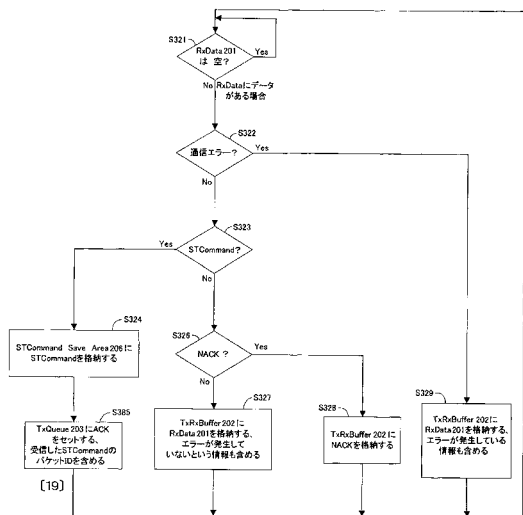


【図 1 4】



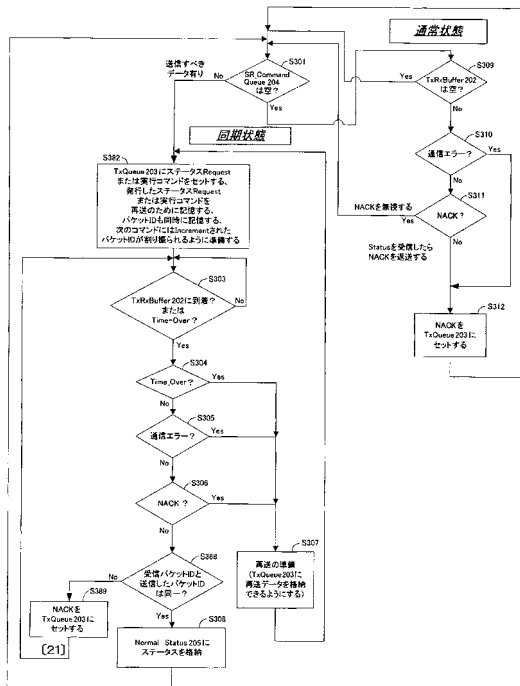
【図15】

第1受信 TASK 102



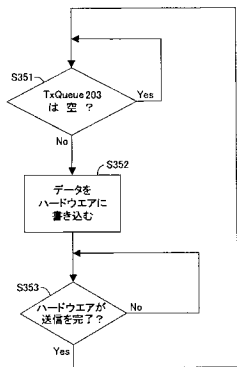
【図16】

コマンド・ステータス制御TASK 101



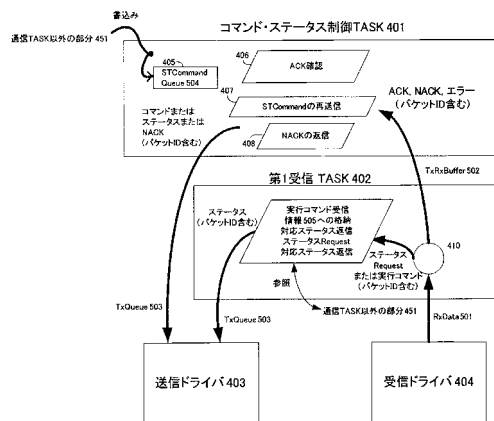
【図17】

送信ドライバ 103



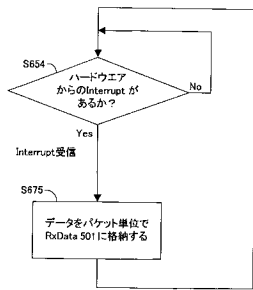
【図18】

CPU16 通信TASK構造



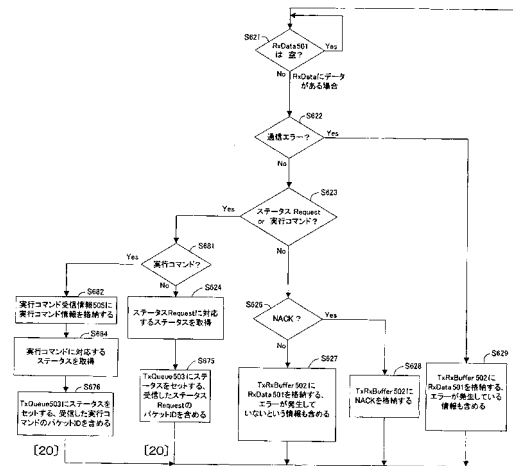
【図19】

受信ドライバ404



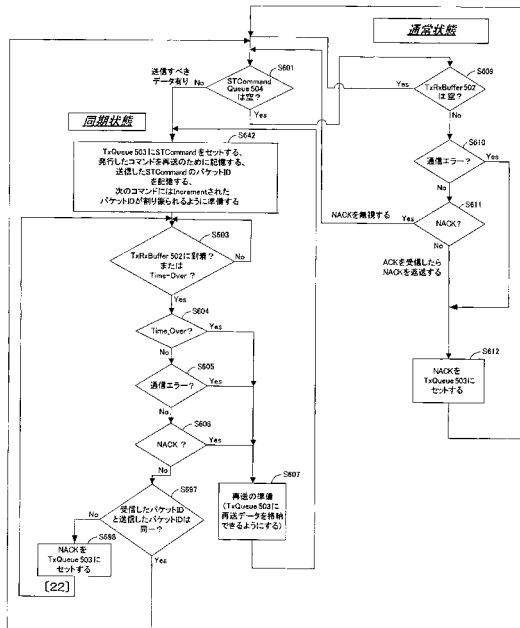
【図20】

第1受信 TASK 402



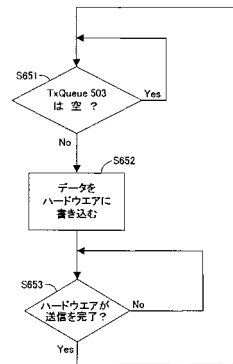
【図21】

コマンド・ステータス制御TASK 401



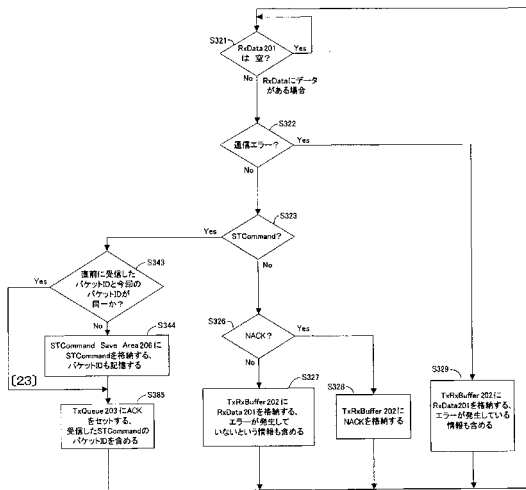
【図22】

送信ドライバ 403



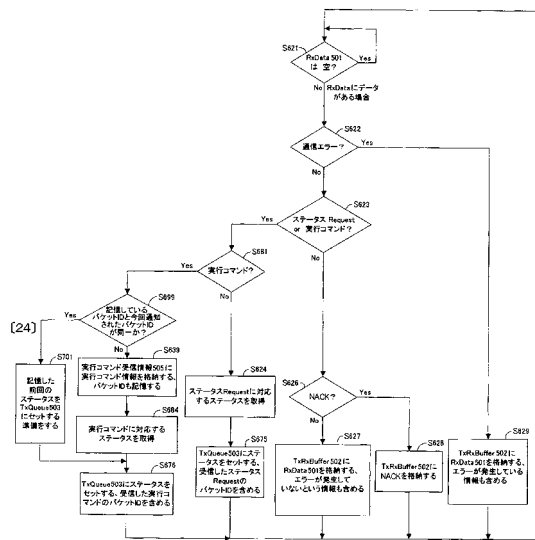
【図 23】

第1受信 TASK 102



【図 24】

第1受信 TASK 402



---

フロントページの続き

- (56)参考文献 特開2004-274376(JP,A)  
特開2007-018121(JP,A)  
特開2005-346175(JP,A)  
特開2007-133747(JP,A)  
特開2006-209392(JP,A)  
特開2004-102622(JP,A)  
特開平10-190565(JP,A)  
特開2005-039601(JP,A)  
特開2007-028214(JP,A)  
特開2006-165758(JP,A)  
国際公開第2008/041329(WO,A1)

(58)調査した分野(Int.Cl., DB名)

H04N 1/00