

参考資料

DP-DS820/DP-DS820 (A) システム実現方法の検討(4台接続)

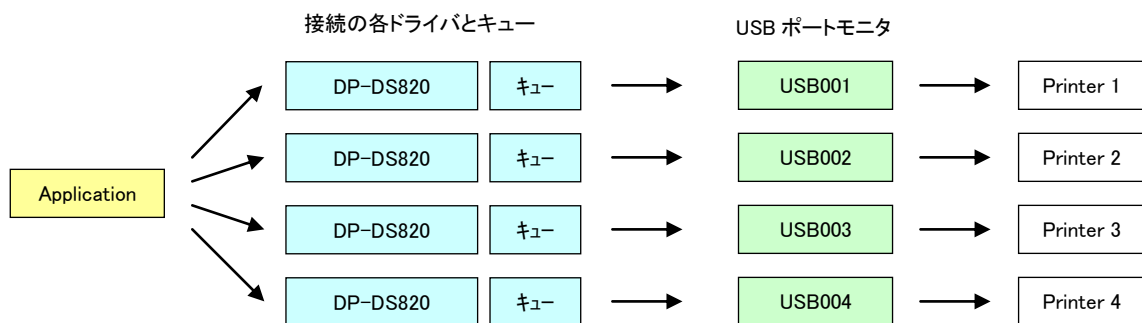
2016 年 3 月 18 日
大日本印刷株式会社

- ⚠ 本書の著作権は権利者にあります。一部または全部の無断転載・無断複製を禁止します。
- ⚠ 本書の内容は予告なく変更することがあります。
- ⚠ 本書に記載されている会社名、製品名は各社の商標または登録商標です。
- ⚠ 仕向けを表す(A)の付かない DP-DS820 を米国へ転売したり、米国で使用したりすることはできません。

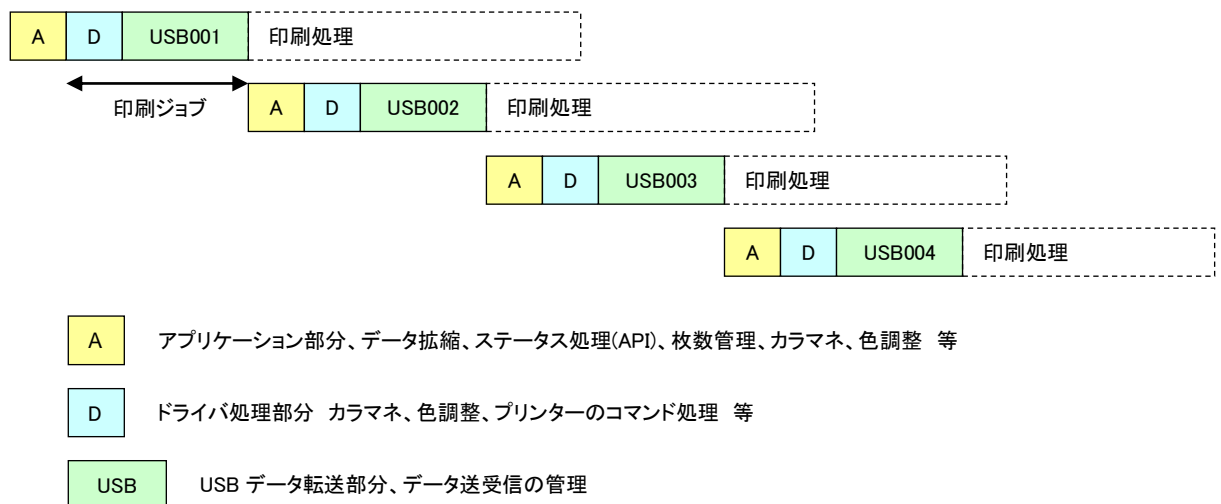
1. Windows 上でのデータの流れについて

プリンタードライバの設定で、スプール ON/OFF の設定がありますが、この設定によりデータ送信がどの様に行われるかを以下に示します。

1-1. スプールOFF時の動作



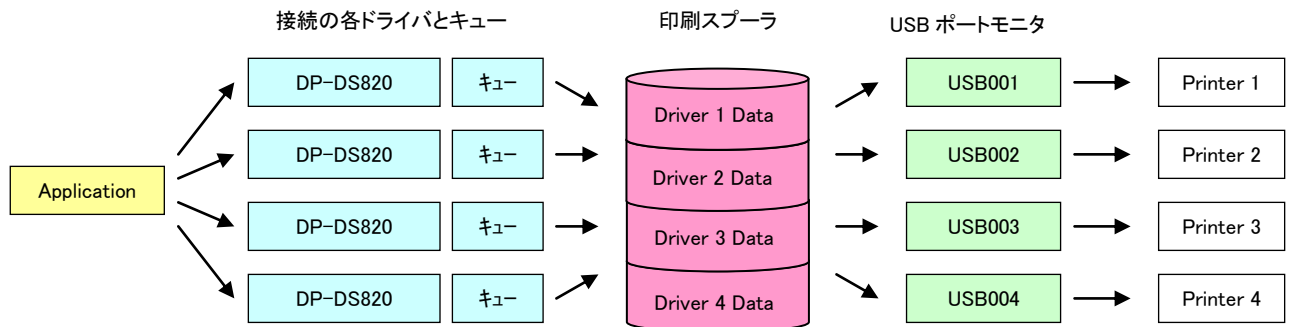
アプリケーションが各ドライバにデータを出力すると、カレントドライバはUSBポートにデータが出力されるまで印刷ジョブを解放しません。したがって、アプリケーションはカレントドライバのUSBポートへのデータ出力終了を待って、次のドライバへデータを出力することが出来ます。



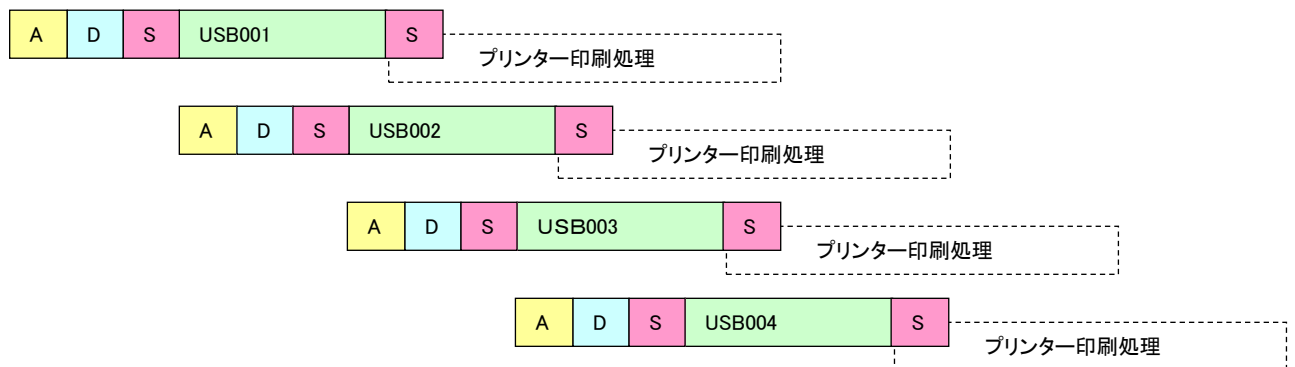
【ステータスの確認について】

スプールをOFFにした場合、データ送信中はアプリケーションは他の処理を行うことが出来ません。したがって、データ送信している間は、他のプリンターのステータスを確認することは出来ません。

1-2. スプールON時の動作



アプリケーションが各ドライバにデータを出力すると、各ドライバは印刷スプーラへデータを出力し印刷ジョブを解放します。アプリケーションは各 USB ポートへのデータ出力を待たずに、次のドライバへデータ出力を行う事が出来ます。印刷スプーラはデータを取り込むと各 USB Port Monitor の管理を行い、各ポートへデータ出力を開始します。印刷スプーラに複数のデータが存在する場合、各 USB ポートへのデータ出力は並列処理で行われます。



- A** アプリケーション部分、データ拡張、ステータス処理(API)、枚数管理、カラマネ、色調整 等
- D** ドライバ処理部分 カラマネ、色調整、プリンターのコマンド処理 等
- S** 印刷スプーラへのファイル書き込み、ファイル削除処理部分
- USB** USB データ転送部分、データ送受信の管理

【ステータスの確認について】

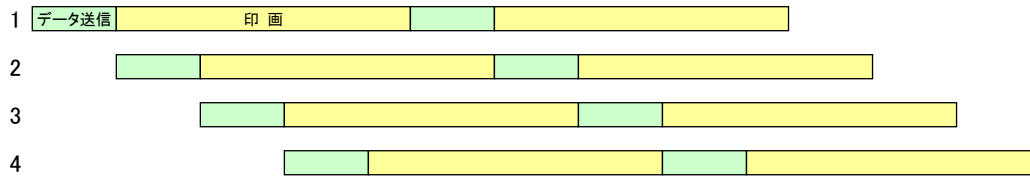
スプールをONにした場合、スプーラへデータ出力が行われるとアプリケーションは印刷処理から解放され、他の処理が行えるようになります。したがって、データ送信している間も、他のプリンターのステータスを確認することが可能です。

2. プリンターのバッファ動作(シングル/ダブル)とスプールON/OFFの関係

プリンターのバッファ動作(シングル/ダブル)と、スプールON/OFFの関係を以下に示します。

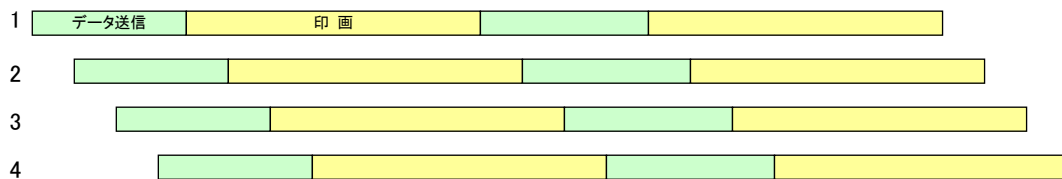
(1) シングルバッファ スプールOFF

スプールOFFであるため1台毎にデータが送信され、更に印画が終了したのを待って次のデータが送信されます。



(2) シングルバッファ スプール ON

スプールをONすることでデータ送信の処理がオーバーラップしますが、スプールの処理時間によりトータルの処理時間は伸びてしまいます。

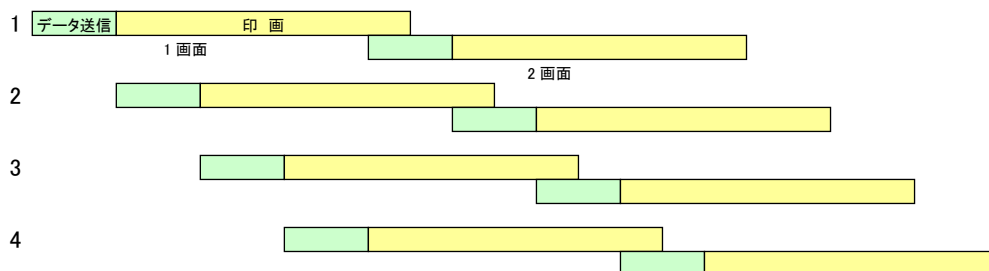


【備考】

- ・理論的にはスプール ON した場合の方がスループットの時間は短くなるはずですが、現在のハード、OS では平行してデータ送信した場合に通信時間が伸びてしまうため、上記に示すようにスプール OFF の時よりも時間が長くなっているものと推測されます。
- ・今後、ハードとOSの改良により平行して送信した場合でも高速で通信できるようになれば、スプール ON にした場合の方が時間は短くなると思われます。

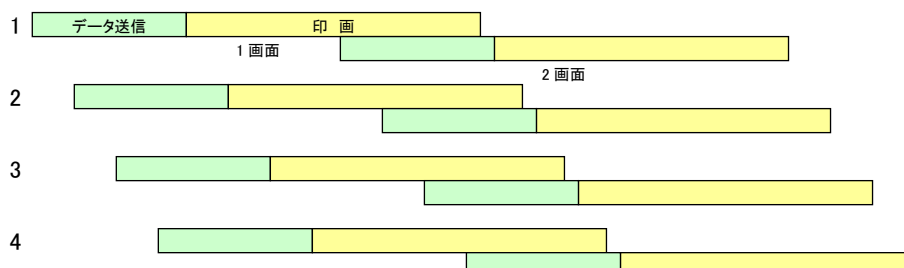
(3) ダブルバッファ スプール OFF

ダブルバッファで処理するため、1画面目を印画中に2画面目のデータを送信することが出来ます。その分処理時間が短くなります。



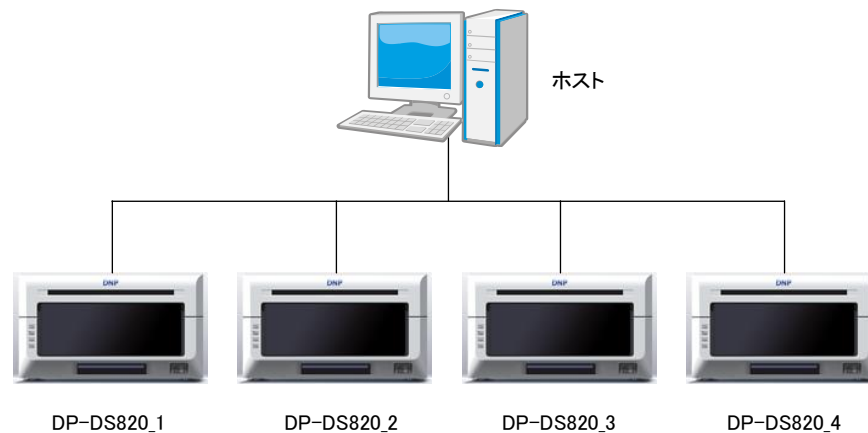
(4) ダブルバッファ スプールON

スプールをONすることでデータ送信がオーバーラップ出来、更にダブルバッファ処理により印画中のデータ送信が可能となるため、トータルの処理時間は最短となります。



3. システム構成例

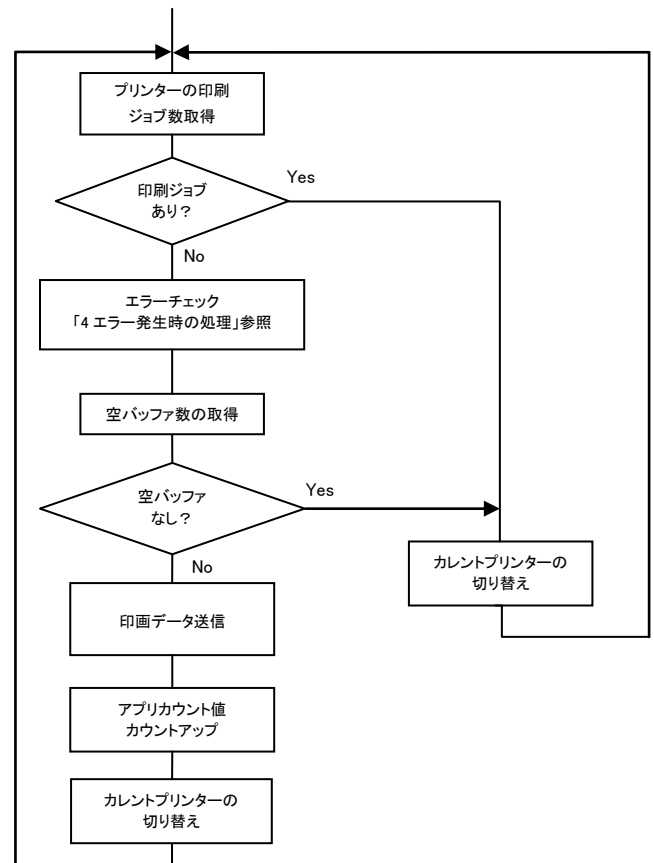
- ・プリンター接続台数 : 4 台
- ・プリンター動作モード : ダブルバッファ動作
- ・スプール設定 : スプールON



【制御方法】

処理の大きな流れとしては、印画要求があった場合に、データを送信できる状態にあるプリンターを探し、データを送信します。具体的な処理は下記の様に行います。

- ① 印画要求がある場合に、カレントプリンターの印刷ジョブがあるか確認し (Windows API EnumJobs)、印刷ジョブがある場合にはカレントプリンターを他のプリンターに切替えます
- ② エラーチェックを行います。エラーが発生している場合には、エラーリカバリー処理を行います (「4.エラー発生時の処理」参照)。
- ③ 印刷ジョブがなく、エラーが発生していない場合にはプリンターの空バッファ数を取得します (ステータス API GetFreeBuffer)。
- ④ 空きバッファがない場合は、カレントプリンターを切替えて、送信可能なプリンターを探します。
- ⑤ 空きバッファが1つ以上ある場合に、プリンターへ印画データを送信します。
- ⑥ 印画データを送信したら、アプリケーション側のカウント値をカウントアップし、カレントプリンターを切替えます。



4. エラー発生時の処理

エラー発生時には下記の様にプリンターのカウント値を取得して(ステータス API GetCounterL)、その値を元にどこまで印画が終了しているかを判断し、破棄されたデータを再度印画処理する必要があります。

- ① 印刷ジョブを確認します。
- ② 印刷ジョブがない場合は、プリンターのステータスを取得してエラーが発生していないか確認します。
- ③ エラーが発生している場合には、プリンターのカウント値を取り込みます。
- ④ プリンターのカウント値とアプリケーション側のカウント値を比較します。
- ⑤ カウント値の差が 2 の場合には、2 つ前のデータと1つ前のデータを再度印画処理します(4 台接続の場合は、他の正常動作しているプリンターへデータ送信。1 台接続の場合は、エラー解除後にプリンターへデータ送信)。カウント値の差が 1 の時は、1 つ前のデータを再度印画処理します。

【備考】

- ・印刷ジョブがいつまでも残っている場合には、通信が停止している可能性があるため、エラー処理をする必要があります。
- ・印刷ジョブが無く、何らかの原因で通信が停止していた場合、DP-DS820 プリンターステータスAPIでステータスを取得しようとすると約 1 分後に取得失敗のコードが返ってきます(この間アプリケーションはステータス待ち状態にあります)。そこでステータス確認時は、確認開始から終了までの時間をチェックして、明らかに時間が掛かっている場合には、エラー処理をする必要があります。

