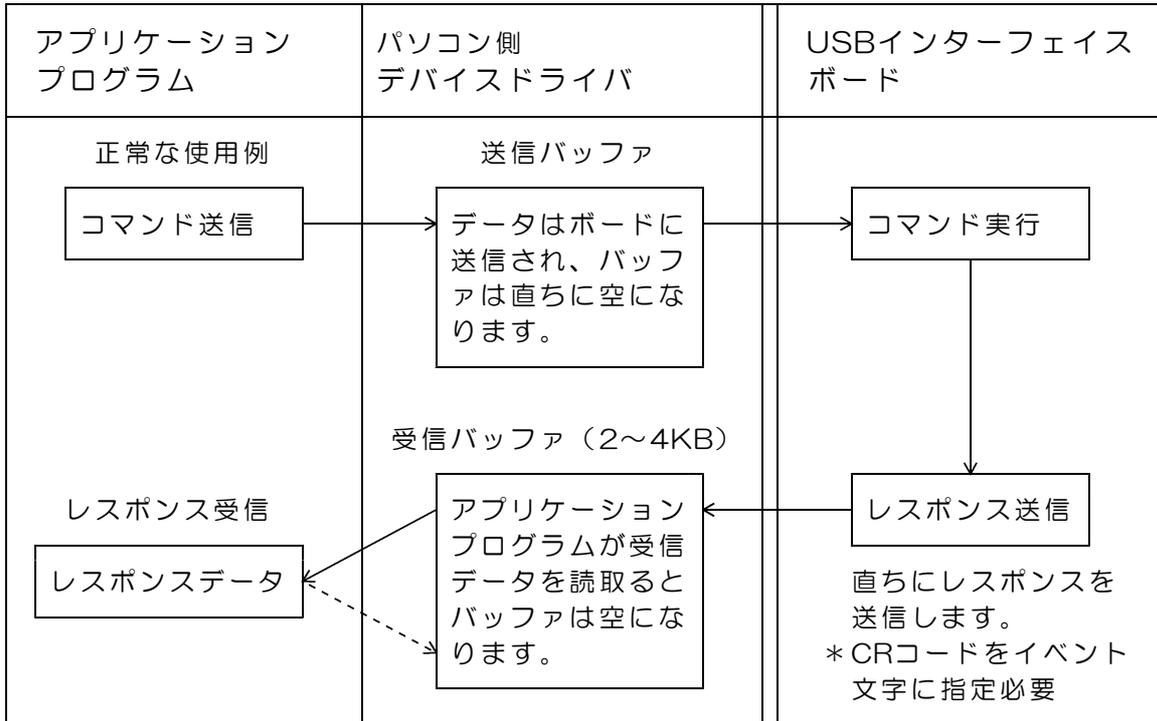
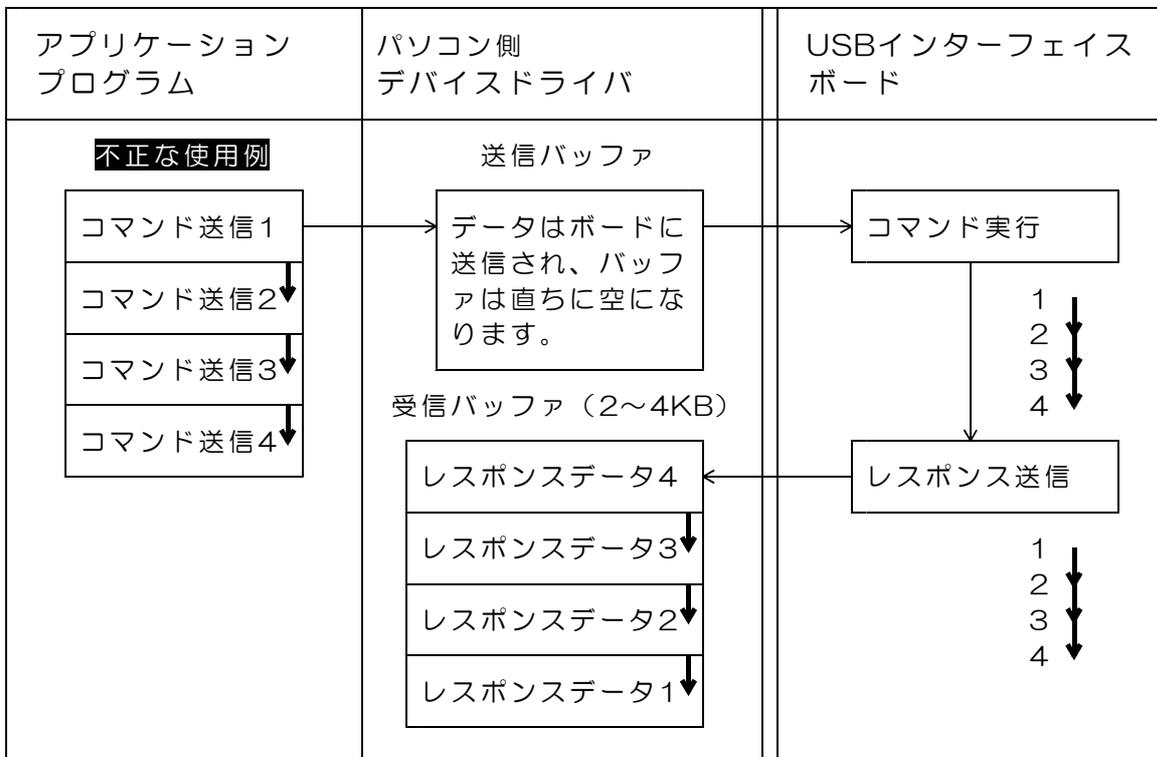


DACS USBインターフェイス製品のコマンド送信とレスポンス受信

コマンド送信とレスポンス受信が正常にハンドシェイクを繰り返している場合

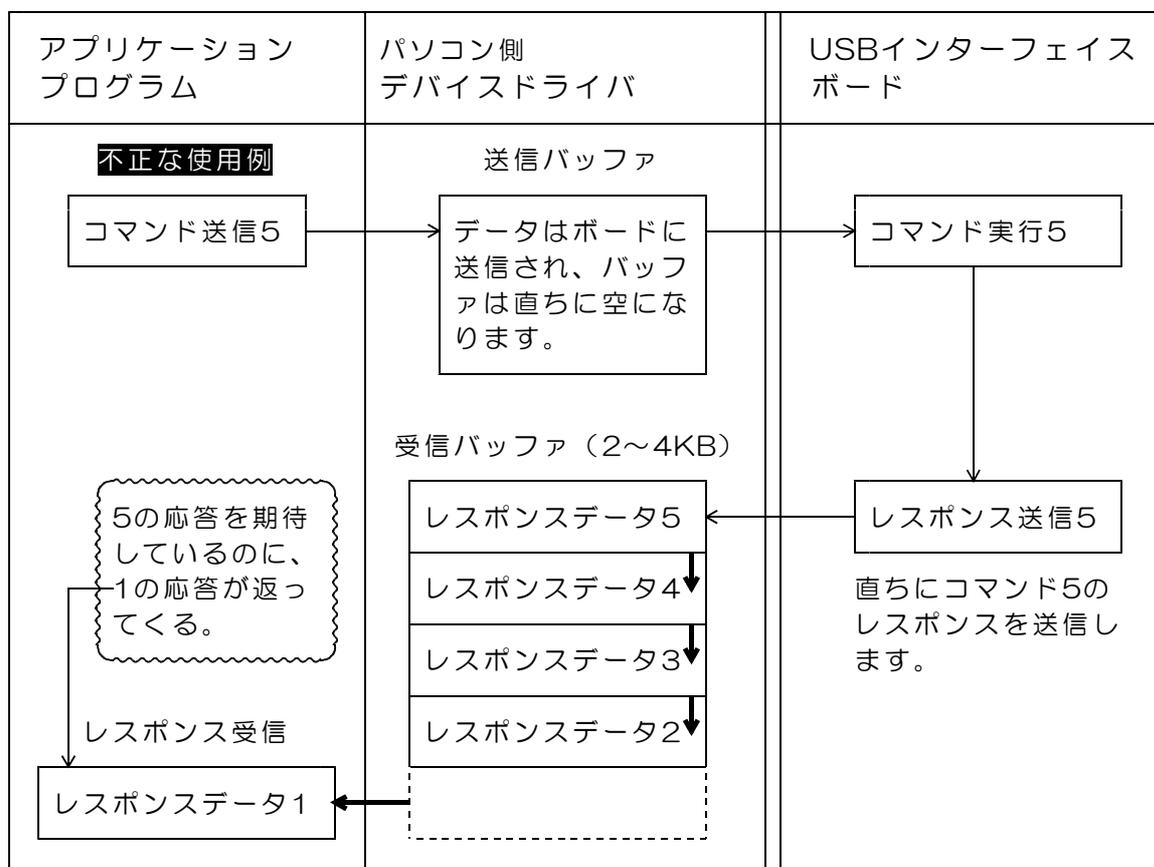


コマンド送信のみを続けると、パソコン側の受信バッファにレスポンスデータがたまってゆきます。



*コマンド送信のみを続けると、受信バッファがいつかはオーバーフローします。

受信バッファにレスポンスデータがたまった状態で、次のコマンドを送って、コマンド分1個のレスポンスを読取ると、期待しているレスポンスとは異なったデータを読取ることになります。

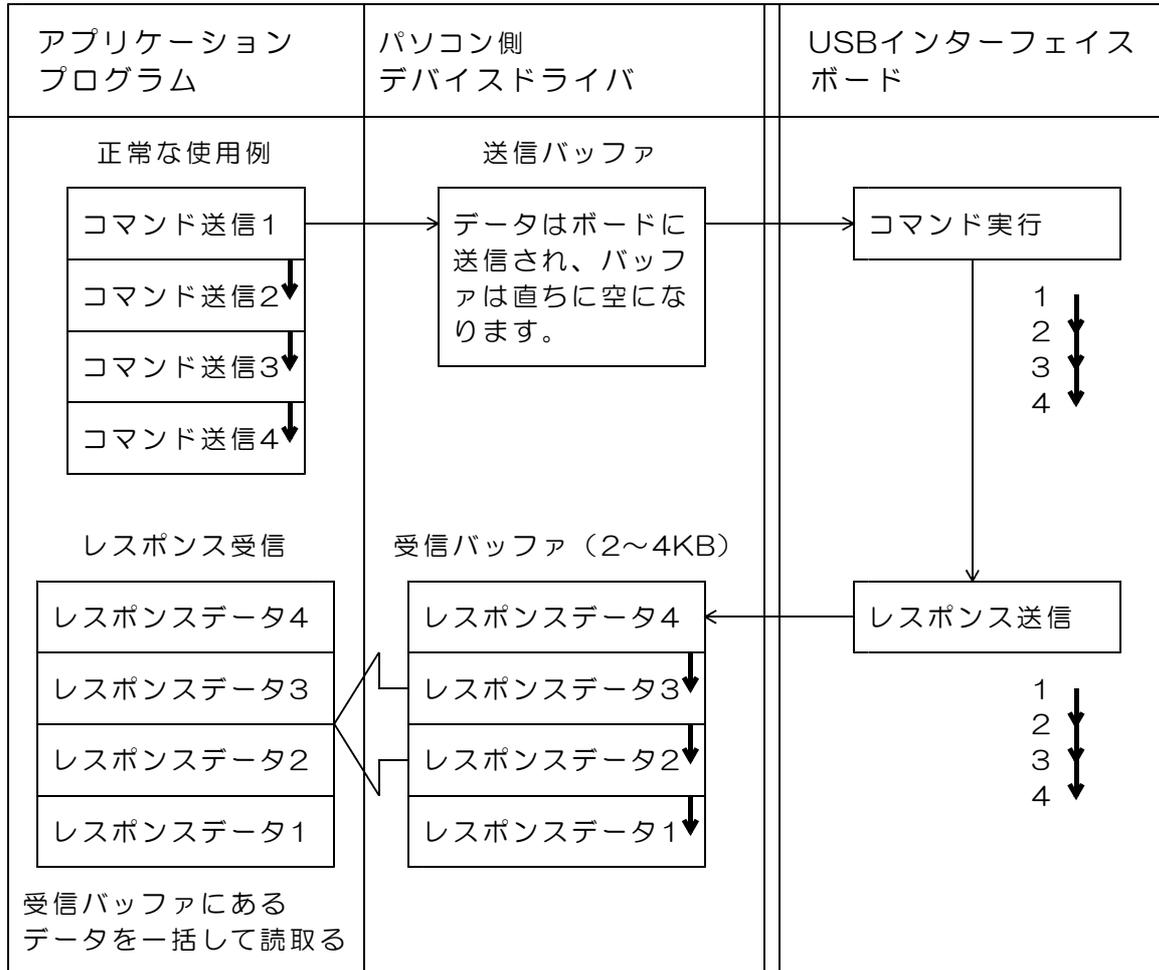


*上図のような問題を起こさないためには、(1) コマンド送信の後、必ず応答を受信すること。(2) 受信データ読取の前に、受信バッファにあるデータ数(バイト数)を確認することが肝要です。プログラミングの詳細はサンプルプログラムを参照ください。

カウンタで問題が生じる間違ったプログラミング例

- ①エンコーダカウンタモードとするため、M018 (CR) を送信
必要ないのでボードからの応答を読取らない。 <----- 間違い ×
- ②カウンタをスタートするため、M008 (CR) を送信
必要ないのでボードからの応答を読取らない。 <----- 間違い ×
- ③カウント値 (low word) 読取りのため M00 (CR) を送信
- ④カウント値 (low word) を読取り (9文字分を読取り) --> ①の応答 ×
- ⑤カウント値 (high word) 読取りのため M01 (CR) を送信
- ⑥カウント値 (high word) を読取り (9文字分を読取り) --> ②の応答 ×
- ⑦カウント値 (low word) 読取りのため M00 (CR) を送信
- ⑧カウント値 (low word) を読取り (9文字分を読取り) --> ③の応答 ×
- ⑨カウント値 (high word) 読取りのため M01 (CR) を送信
- ⑩カウント値 (high word) を読取り (9文字分を読取り) --> ⑤の応答 ×

複数のコマンドに対する応答を、一括して読取るプログラミング方法もあります。処理時間短縮に有効な手法です。



カウンタで複数チャンネルのカウンタ値を読取る高度なプログラミング例

- ① I (アイ) コマンドでコマンド実行間隔 (10 μs程度) を設定する。
必ず9バイト分の応答を読取る
- ② 3ch分のカウンタ値読取りのため
MO0&MO1&MO2&MO3&MO4&MO5 (CR) を送信
- ③ 受信バッファに54バイト分の受信データあることを確認し、一括してデータを読取る。
---- カウンタ値読取りは、②~③の繰返し。
①は一度実行すれば、その後は必要なし。