リモートパネル アプリケーションインターフェース 説明書

ご注意

1. 本書の内容の一部または全部を無断転載することは禁止されています。
2. 本書の内容に関しては将来予告なしに変更することがあります。
3. NECの許可なく複製・改変などを行うことはできません。
4. 本書は内容について万全を期して作成致しましたが、万一ご不審な点や誤り、記載もれなどお気づきのことがありましたら、お買い求めの販売店にご連絡ください。
5. 運用した結果の影響については4項にかかわらず責任を負いかねますのでご了承ください。
6. 本製品を第三者に売却・譲渡する際は必ず本書も添えてください。

853-810932-431-A
第4版

目　次

1. はじめに

本書が対象とするプリンターは次のとおりです。

- MultiCoder 500Lシリーズ
- MultiCoder 502Lシリーズ

プリンターに付属のリモートパネルをインストールすると、プリンターに対して、パラメーター設定・微調整設定・IPアドレス設定の「取得・設定」とDIP SW状態の取得を行うことができるAPI（以下、RpApi）が同時にインストールされます。また、サンプルプログラムもリモートパネルのインストールされた次のいずれかのフォルダーにインストールされています。

- MultiCoder 500Lシリーズの場合
  「¥Program Files¥RP for MC500L¥SAMPLE」
- MultiCoder 502Lシリーズの場合
  「¥Program Files¥RP for MC502L¥SAMPLE」

本書では、RpApiが提供する機能を、アプリケーションから使用するための説明をします。
サンプルプログラムは、RpApiを使用して、プリンターから設定値を読み込み、印刷濃度微調整（発色モード）の設定を行うものです。
本書と合わせてサンプルプログラムをご覧ください。
動作環境は、リモートパネルが動作する環境に準じます。

2. 注意

- 動作環境は、リモートパネルが動作する環境に準じます。
- プリンタードライバーの［ポート］シートで［双方向サポートを有効にする］にチェックが付いていることを確認してください。
- USER権限で使用する場合は、権限を与える必要があります。プリンタードライバーの［セキュリティ］シートから［プリンタの管理］を許可してください。

3. RpApiが提供する機能

RpApiは、アプリケーションに対して以下の機能を提供します。
RpApiは、アプリケーションに対して以下の機能を提供するDynamic Link Library（DLL）です。

関数名	機能説明
DllGetPrinterStatus	プリンターから「ステータス」を取得します。
DllGetPrinterParam	プリンターから「パラメーター設定値・微調整設定値・IPアドレス設定値・DIP SW状態」を取得します。
DllSetPrinterParam	プリンターの「パラメーター設定」を行います。
DllSetPrinterAdjust	プリンターの「微調整値設定」を行います。
DllSetPrinterAdd	プリンターの「IPアドレス設定」を行います。
DllResetPrinter	プリンターを「リセット」します。

4. 関数詳細

4.1 DllGetPrinterStatus

プリンターのステータスを取得します。

チェック

プリンターとコンピューターがパラレルインターフェースまたはRS-232Cインターフェースで接続されている場合は、ステータスを取得することはできません。

書式

```
int DllGetPrinterStatus(
  RP_INFO           *pRpInfo,
  DWORD             *dwPrinterStatusID,
  RP_ERROR          *pRpError);
```

引数

```
RP_INFO           *pRpInfo            // 接続設定情報
DWORD             *dwPrinterStatusID  // プリンターステータスID
RP_ERROR          *pRpError           // エラー情報
```

戻り値

0　　：　処理成功
その他　：　エラー発生

取得ステータス一覧

ID	ステータス情報
0	オンライン
1	オンライン状態でのヘッドオープンを検出
2	動作中（印刷中・フィード中）
4	ポーズ状態
5	剥離待ち状態
6	実行動作解析中にエラーを検出
11	紙づまりを検出
12	カッター部の異常を検出
13	ラベルエンドを検出
14	リボンエンドを検出
17	ヘッド断線エラーを検出
18	ヘッド高温を検出

4.2 DllGetPrinterParam

プリンターのパラメーター設定値・微調整設定値・IPアドレス設定値・DIP SW状態を取得します。

チェック

プリンターとコンピューターがパラレルインターフェースまたはRS-232Cインターフェースで接続されている場合は、情報を取得することはできません。

書式

```
int DllGetPrinterParam(
  RP_INFO           *pRpInfo,
  RP_PARAM          *pRpParam,
  RP_ADJUST         *pRpAdjust,
  RP_IPADD          *pRpIpAdd,
  RP_ERROR          *pRpError);
```

引数

```
RP_INFO           *pRpInfo          // 接続設定情報
RP_PARAM          *pRpParam         // パラメーター設定値
RP_ADJUST         *pRpAdjust        // 微調整設定値
RP_IPADD          *pRpIpAdd         // IPアドレス設定値
RP_ERROR          *pRpError         // エラー情報
```

戻り値

0　　：　処理成功
その他　：　エラー発生

4.3 DllSetPrinterParam

プリンターのパラメーター設定を行います。

書式

```
int DllSetPrinterParam(
  RP_INFO          *pRpInfo,
  RP_PARAM         *pRpParam,
  RP_ERROR         *pRpError);
```

引数

```
RP_INFO            *pRpInfo          // 接続設定情報
RP_PARAM           *pRpParam         // パラメーター設定値
RP_ERROR           *pRpError         // エラー情報
```

戻り値

0　　：　処理成功
その他　：　エラー発生

4.4 DllSetPrinterAdjust

プリンターの微調整設定を行います。

書式

```
int DllSetPrinterAdjust(
  RP_INFO          *pRpInfo,
  RP_ADJUST        *pRpAdjust,
  RP_ERROR         *pRpError);
```

引数

```
RP_INFO            *pRpInfo          // 接続設定情報
RP_ADJUST          *pRpAdjust        // 微調整設定値
RP_ERROR           *pRpError         // エラー情報
```

戻り値

0　　：　処理成功
その他　：　エラー発生

4.5 DllSetPrinterAdd

プリンターのIPアドレス設定を行います。

書式

```
int DllSetPrinterAdd(
  RP_INFO          *pRpInfo,
  RP_IPADD         *pRpIpAdd,
  RP_ERROR         *pRpError);
```

引数

```
RP_INFO            *pRpInfo          // 接続設定情報
RP_IPADD           *pRpIpAdd         // IPアドレス設定値
RP_ERROR           *pRpError         // エラー情報
```

戻り値

0　　：　処理成功
その他　：　エラー発生

4.6 DllResetPrinter

プリンターのリセットを行います。
このコマンドを使用してプリンターをリセットするか、プリンターの電源をON/OFFすることで設定した各設定値が有効になります。

書式

```
int DllResetPrinter(
  RP_INFO          *pRpInfo,
  RP_ERROR         *pRpError);
```

引数

```
RP_INFO*pRpInfo                      // 接続設定情報
RP_ERROR*pRpError                    // エラー情報
```

戻り値

0　　：　処理成功
その他　：　エラー発生

5. 関数戻り値一覧

関数は処理成功時に「0」を戻します。
関数の引数に不正値が確認された場合や、関数内部でエラーが発生した場合は、下表に示す値を戻します。
Win32API関数でエラーが発生した場合は、RP_ERROR構造体のメンバーのdwGetLastErrorにGetLastError関数で取得したWin32APIエラー値を設定します。

戻り値	ファンクション	原　因
0	処理成功	ー
-10	接続先設定異常	ローカル接続・LAN接続の指定がされていない
-20	ステータス取得失敗	ステータスの取得ができない
-30	プリンター設定値取得失敗	プリンター設定値の取得ができない
-700	スプーラーエラー検出	ローカル接続時、スプーラーでエラーを検出した
-701	双方向通信エラー検出	ドライバー設定が、双方向通信になっていない
-800	ソケットエラー検出	LAN接続時、ソケットでエラーを検出した
-801	接続テストエラー検出	指定されたアドレスに接続を試みたが、接続できなかった
-900	その他エラー検出	RpApi使用時の引数などに誤りがある

6. 構造体一覧

情報名	構造体名	参 照
接続設定情報	RP_INFO	11ページ
パラメーター設定情報	RP_PARAM	12ページ
微調整設定情報	RP_ADJUST	15ページ
IPアドレス設定情報	RP_IPADD	16ページ
エラー情報	RP_ERROR	17ページ

6.1 RP_INFO構造体

接続先を設定する構造体です。

```
RP_INFO {
   DWORD          dwConnectSendType;     // コマンド出力方法
   DWORD          dwPrinterNameSize;     // 論理プリンター名のバッファーサイズ
   LPSTR          pPrinterName;          // 論理プリンター名
   BYTE           pnPrinterIPAdress[4]; // 論理プリンター IPアドレス
   UINT           nSocketPort;           // ソケットポート番号
};
```

設定値

論理名称	機能名称	設定値
dwConnectSendType	出力方法	0 : ローカル接続の送受信 1 : LAN接続の送受信
dwPrinterNameSize	論理プリンター名称サイズ*1	2 ～ 65536
pPrinterName	論理プリンター名称*1	論理プリンター名称
pnPrinterIPAdress[4]	論理プリンター IPアドレス*2	[0] 000 ～ 255
		[1] 000 ～ 255
		[2] 000 ～ 255
		[3] 000 ～ 255
nSocketPort	論理プリンターソケット通信ポート番号*2	10000 ～ 65535

＊1　出力方法 =「0」を指定した場合は、必ず設定してください。

＊2　出力方法 =「1」を指定した場合は、必ず設定してください。

6.2 RP_PARAM構造体

プリンターのパラメーター設定値取得と設定をする構造体です。

MultiCoder 500Lシリーズの場合

```
RP_PARAM {
  UINT          nDipSw[10];        // DIP SW状態
  UINT          nDummy;            // 未使用
  UINT          nAutoShortCheck;   // 電源ON時の断線エラー表示設定
  UINT          nYoushiUMU;        // 電源ON・カバー閉用紙確認設定
  UINT          nBuzzOnOff;        // エラー発生時ブザー音設定
  UINT          nCut;              // カッターオプション制御方法設定
  UINT          nYoushi;           // 自動用紙位置合わせ設定
  UINT          nBackFeed;         // バックフィード速度設定
  UINT          nStopType;         // カバークローズ後停止位置設定
  UINT          nTearOff;          // 自動カット位置戻し設定
  UINT          nPrintSts;         // 印刷方向設定
  UINT          nMirrorPrint;      // ミラー印刷設定
  UINT          nThinPaperMode;    // 薄紙印刷モード設定
};
```

設定値*1

論理名称	機能名称	設定値（取得値）
nDipSw[10]	DIP SW状態	[0]をDIP SW-1、[9]をDIP SW-10とする。 ONは[1]、OFFは[0] *2
nDummy	未使用	0固定
nAutoShortCheck	電源ON時の断線エラー表示	0：ヘッド断線チェックする 1：ヘッド断線チェックしない
nYoushiUMU	電源ON・カバー閉用紙確認	0：用紙確認する 1：用紙確認しない
nBuzzOnOff	エラー発生時ブザー音	0：ブザー ON 1：ブザー OFF
nCut	カッターオプション制御方法	0：フルカット 1：パーシャルカット
nYoushi	自動用紙位置合わせ	0：用紙位置合わせする 1：用紙位置合わせしない
nBackFeed	バックフィード速度	0：4ips 1：2ips 2：3ips
nStopType	カバークローズ後停止位置	0：停止位置を保持する 1：停止位置をクリアする
nTearOff	自動カット位置戻し時間	0：無制限 1：8秒 2：15秒
nPrintSts	印刷方向	0：コマンド優先 1：通常印刷 2：180° 回転印刷
nMirrorPrint	ミラー印刷	0：コマンド優先 1：通常印刷 2：ミラー印刷
nThinPaperMode	薄紙印刷モード	0：無効 1：有効

＊1　下記以外の設定値を設定した場合、プリンターの動作は保障いたしません。

＊2　DllGetPrinterParam 関数使用時のみ取得値は有効になります。DIP SW を設定することはできません。

MultiCoder 502Lシリーズの場合

```
RP_PARAM {
  UINT              nDipSw[10];          // DIP SW状態
  UINT              nHeadCheck;          // 電源ON時の断線エラー表示設定
  UINT              nPaperCheck;         // 電源ON・カバー閉用紙確認設定
  UINT              nBuzzer;             // エラー発生時ブザー音設定
  UINT              nCut;                // カッターオプション制御方法設定
  UINT              nPaperTopSet;        // 自動用紙先頭位置合わせ設定
  UINT              nAutoBackFeedTime;   // 自動カット位置戻し時間設定
  UINT              nThinPaperMode;      // 薄紙印刷モード設定
  UINT              nBlackMarkPos;       // 黒マーク検出時の頭出し制御設定
  UINT              nZeroSlash;          // ゼロスラッシュ設定
  UINT              nOffLineBoot;        // プリンター起動時の状態設定
  UINT              nFeedPrint;          // ［FEED］スイッチによる印刷設定
  UINT              nAutoCutFeed;        // 自動カット位置送り機能設定
  UINT              nParallelAck;        // パラレルI/F ACK幅変更設定
  UINT              nInputPrime;         // インプットプライム設定
  UINT              nAutostatusSend;     // ステータス自動送信設定
  UINT              nBackFeed;           // バックフィード速度設定
  UINT              nStopType;           // カバークローズ後停止位置設定
  UINT              nPrintDirection;     // 印刷方向設定
  UINT              nMirrorPrint;        // ミラー印刷設定
};
```

設定値*1

論理名称	機能名称	設定値（取得値）
nDipSw[10]	DIP SW状態	[0]をDIP SW-1、[9]をDIP SW-10とする。 ONは[1]、OFFは[0] *2
nHeadCheck	電源ON時の断線エラー表示	0：ヘッド断線チェックする 1：ヘッド断線チェックしない
nPaperCheck	電源ON・カバー閉用紙確認	0：用紙確認する 1：用紙確認しない
nBuzzer	エラー発生時ブザー音	0：ブザー ON 1：ブザー OFF
nCut	カッターオプション制御方法	0：フルカット 1：パーシャルカット
nPaperTopSet	自動用紙先頭位置合わせ	0：用紙位置合わせする 1：用紙位置合わせしない
nAutoBackFeedTime	自動カット位置戻し時間	0：無制限 1：8秒 2：15秒
nThinPaperMode	薄紙印刷モード	0：無効 1：有効
nBlackMarkPos	黒マーク検出時の頭出し制御	0：黒マーク上端 1：黒マーク下端
nZeroSlash	ゼロスラッシュ	0：なし 1：あり
nOffLineBoot	プリンター起動時の状態	0：オフライン起動 1：オンライン起動
nFeedPrint	［FEED］スイッチによる印刷	0：行わない（用紙送り） 1：行う（印刷）
nAutoCutFeed	自動カット位置送り機能	0：有効*3 1：無効
nParallelAck	パラレルI/F ACK幅変更*4	0：0.5μsec 1：8μsec
nInputPrime	インプットプライム*4	0：無効 1：有効

設定値*1

論理名称	機能名称	設定値（取得値）
nAutostatusSend	ステータス自動送信*5	0：無効 1：有効 2：コマンド優先
nBackFeed	バックフィード速度	0：4ips 1：3ips 2：2ips
nStopType	カバークローズ後停止位置	0：停止位置を保持する 1：停止位置をクリアする
nPrintDirection	印刷方向	0：コマンド優先 1：通常印刷 2：180° 回転印刷
nMirrorPrint	ミラー印刷	0：コマンド優先 1：通常印刷 2：ミラー印刷

＊1　下記以外の設定値を設定した場合、プリンターの動作は保障いたしません。

＊2　DllGetPrinterParam 関数使用時のみ取得値は有効になります。DIP SW を設定することはできません。

＊3　「連続印刷」を指定しているときのみ機能します。

＊4　パラレルインターフェースに接続して使用しているときに有効な機能です。

＊5　RS-232C インターフェース、LAN インターフェース、および無線 LAN インターフェース接続時に有効な機能です。

6.3 RP_ADJUST構造体

プリンターの微調整設定値の取得と設定をする構造体です。

```
RP_ADJUST {
   int              iDensity;          // 濃度微調整値（感熱）
   int              iDensity2;         // 濃度微調整値（熱転写）
   int              iFeed;             // フィード量微調整値
   int              iCutPeelOff;       // カット・ハクリ位置微調整値
   int              iAutoCutPos;       // 自動カット位置送り位置微調整値
   int              iBackPos;          // 戻し位置微調整値
   int              iAutoForwardWait;  // 自動正転待機位置微調整値
   int              iXPos;             // X位置微調整値
};
```

設定値＊1

論理名称	機能名称	設定値（取得値）
iDensity	濃度微調整値（感熱）	-10 ～ 10(1ステップ単位)
iDensity2	濃度微調整値（熱転写）	-10 ～ 10(1ステップ単位)
iFeed	フィード量微調整値	-99 ～ 99(0.1mm単位)＊2
iCutPeelOff	カット・ハクリ位置微調整値	-99 ～ 99(0.1mm単位)＊2
iAutoCutPos	自動カット位置送り位置微調整値	-99 ～ 99(0.1mm単位)＊2
iBackPos	戻し位置微調整値	-99 ～ 99(0.1mm単位)＊2
iAutoForwardWait	自動正転待機位置微調整値	-99 ～ 99(0.1mm単位)＊2
iXPos	X位置微調整値	-1057 ～ 1057(0.1mm単位)＊2

＊1　下記以外の設定値を設定した場合、プリンターの動作は保障いたしません。

＊2　10倍値で設定してください。

6.4 RP_IPADD構造体

プリンターのIPアドレス設定値取得と設定をする構造体です。

```
RP_IPADD {
  BYTE              pnIPAdress[4];      // IPアドレス設定値
  BYTE              pnSubNet[4];        // サブネットマスク設定値
  BYTE              pnGateway[4];       // デフォルトゲートウェイ設定値
  UINT              nDhcpBootp;         // DHCP・BOOTP機能設定値
  UINT              nRaRp;              // RARP設定値
  UINT              nSocketNo;          // ソケット通信ポート番号
};
```

設定値＊1

論理名称	機能名称	設定値（取得値）
pnIPAdress	IPアドレス設定値	[0] 000 ～ 255
		[1] 000 ～ 255
		[2] 000 ～ 255
		[3] 000 ～ 255
pnSubNet	サブネットマスク設定値	[0] 000 ～ 255
		[1] 000 ～ 255
		[2] 000 ～ 255
		[3] 000 ～ 255
pnGateway	デフォルトゲートウェイ設定値	[0] 000 ～ 255
		[1] 000 ～ 255
		[2] 000 ～ 255
		[3] 000 ～ 255
nDhcpBootp	DHCP・BOOTP機能設定値	0：DHCP/BOOTP機能有効 1：DHCP/BOOTP機能無効
nRaRp	RARP設定値	0：RARP機能有効 1：RARP機能無効
nSocketNo	ソケット通信ポート番号	10000 ～ 65525

＊1　下記以外の設定値を設定した場合、プリンターの動作は保障いたしません。

6.5 RP_ERROR構造体

RpApiで異常を検出した場合、エラーコードが設定される構造体です。

```
RP_ERROR {
  DWORD              dwGetLastError;      // GetLastError関数で取得したエラー値
  };
```

設定値

論理名称	機能名称	設定値（取得値）
dwGetLastError	GetLastError関数で取得したエラー値	