



FLINK ライブラリマニユアル

このマニュアルは、FLCE 機能を操作するライブラリの仕様について記載します。

ご注意

- このソフトウェアおよびマニュアルの、一部または全部を無断で使用、複製することはできません。
- このソフトウェアおよびマニュアルは、本製品の使用許諾契約書のもとでのみ使用することができます。
- このソフトウェアおよびマニュアルを運用した結果の影響については、一切の責任を負いかねますのでご了承ください。
- このソフトウェアの仕様、およびマニュアルに記載されている事柄は、将来予告なしに変更することがあります。
- このマニュアルの著作権はカシオ計算機株式会社に帰属します。
- 本書中に含まれている画面表示は、実際の画面とは若干異なる場合があります。予めご了承ください。

© 2019 カシオ計算機株式会社

Microsoft, MS, ActiveSync, Active Desktop, Outlook, Windows, Windows NT, および Windows ロゴは、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。Microsoft 社の製品は、OEM 各社に、Microsoft Corporation の 100%出資子会社である Microsoft Licensing, Inc.によりライセンス供与されています。

変更履歴

バージョン	変更日付	ページ	内容
0.90	2005.11		新規作成
1.00	2006.06	62	機種名修正
1.01	2007.08	1、44、 47、65、 66	機種名に DT-X7 を追加
1.02	2007.12	16	FLKOpen 関数正常終了時の戻り値を修正
		51	HT モードの説明を修正
1.03	2008.07	1、44、 47、65、 66	機種名に DT-9800 を追加
		16～46	Device Emulator における注意書きを追加
2.00	2008.11		「FLINK ライブラリマニュアル」と「FLINK クラスライブラリ マニュアル」を統合
		1	開発環境に Microsoft Visual Studio 2008 を追加
		1、44、 47、65、 66	機種名から DT-950 / DT-9700 / DT-10 / DT-5100 / DT-870 / E-3100 を削除
2.01	2009.03	1	開発環境とプログラミング言語の対応表を訂正
		4	FLK_RSPRM 構造体宣言のコメントを追加
		5	FLK_STATUS 構造体宣言の誤記を訂正
		18	FLKFileSend / FLKSendFile 関数のパラメータ説明を 訂正
		22	FLKFileRecv / FLKReceiveFile 関数のパラメータ説明 を訂正
		26	FLKReadErrStat / FLKReadErrorStatus 関数の戻り値 の説明を訂正
2.02	2009.07	—	対象機種に DT-5300 を追加
2.03	2011.01	—	対象機種に IT-300 および DT-X8 を追加
2.04	2011.08	5、6、7、 27、64	不要なパラメータを削除
2.05	2011.11	—	対象機種に IT-9000 を追加
	2012.02	27	FLKIdle 関数のメッセージを追加
		54～63	ファイル送受信基本機能の各フローチャートを修正
		64	通信状態の通知にメッセージを追加
2.06	2014.11	—	対象機種に IT-G500 を追加
2.07	2015.02	—	対象機種に DT-X100 および DT-X200 を追加
2.08	2016.12	47	一覧表に IT-G500、DT-X100、DT-X200 を追加
2.09	2019.03	1、2	DLL のファイル名を修正

目次

1.	概要	1
1.1	構造体一覧	3
1.1.1	FLK_RSPRM	4
1.1.2	FLK_STATUS	5
1.1.3	FLK_ERRINFO	7
1.1.4	FLK_DATETIME	10
1.1.5	FLK_FINFO	11
1.1.6	FLK_FINFOEX	12
1.1.7	FLK_DINFO	13
1.1.8	FLK_SYSINFO	14
1.2	関数一覧	15
1.2.1	FLKOpen	16
1.2.2	FLKFileSend / FLKSendFile	18
1.2.3	FLKFileAdd / FLKAddFile	20
1.2.4	FLKFileRecv / FLKReceiveFile	22
1.2.5	FLKClose	24
1.2.6	FLKReadStatus	25
1.2.7	FLKReadErrStat / FLKReadErrorStatus	26
1.2.8	FLKIdle	27
1.2.9	FLKCmdRecv / FLKReceiveCommand	28
1.2.10	FLKFileDelete / FLKDeleteFile	30
1.2.11	FLKFileMove / FLKMoveFile	31
1.2.12	FLKMakeDir / FLKMakeDirectory	33
1.2.13	FLKGetFileInfo	35
1.2.14	FLKSetFileInfo	36
1.2.15	FLKGetFileInfoEx	37
1.2.16	FLKSetFileInfoEx	39
1.2.17	FLKGetDiskInfo	40
1.2.18	FLKDateTime	42
1.2.19	FLKGetSysInfo / FLKGetSystemInfo	44
1.2.20	FLKMsgSend / FLKSendMessage	45
1.2.21	FLKBeep	46
2.	通信仕様	47
2.1	通信構成	47
2.2	通信パラメータ	49
2.3	動作モード	50
2.4	コマンド送信権	51
2.5	処理概要	52
3.	ファイル送受信基本機能	54
3.1	通信基本関数	54
3.2	ファイル送受信関数	58
3.3	リモート操作関数	62
4.	通信状態の通知	64
5.	パス名の記述方法	65

1. 概要

FLINK ライブラリは、FLCE 機能をユーザアプリケーションにて利用することを可能とします。

動作環境

対象機種

- DT-5200
- DT-X7
- DT-9800
- DT-5300
- IT-300
- DT-X8
- IT-9000
- IT-G500
- DT-X100
- DT-X200

対象 OS

- Microsoft Windows CE 5.0
- Microsoft Windows CE 6.0
- Microsoft Windows Embedded Compact 7
- Microsoft Windows Mobile 6.5
- Microsoft Windows Mobile 6.5.3
- Microsoft Windows Embedded Handheld 6.5

開発環境とプログラミング言語

開発環境	Visual C++	Visual Basic, Visual C#
Microsoft embedded Visual C++ Version 4.0 + SP4	○	-
Microsoft Visual Studio.NET 2003 + SP1	×	○
Microsoft Visual Studio 2005 + SP1	○	○
Microsoft Visual Studio 2008 + SP1	○	○

(○:利用可、×:利用不可、-:機能なし)

提供ファイル

ファイル	Visual C++	Visual Basic, Visual C#
FlinkLib.h	○	-
FlinkLib.lib	○	-
FlinkLib.dll	○	○
MoFlinkLibNet.dll (クラスライブラリ)	-	○

(○:必要、-:不要)

使用方法

Visual C++ の場合

- プログラムソース内に `FlinkLib.h` をインクルードし、リンカの依存ファイルとして `FlinkLib.lib` を指定してください
- `FlinkLib.dll` は本体に内蔵されています。

Visual Basic または Visual C# の場合

- `MoFlinkLibNet.dll` をプロジェクトの参照に追加してください。
- `FlinkLib.dll` は本体に内蔵されています。
- `MoFlinkLibNet.dll` を実行モジュールと同じフォルダにコピーしてください。

1.1 構造体一覧

構造体	内容
FLK_RSPRM	シリアルポートの情報を格納します
FLK_STATUS	ライブラリの状態を格納します
FLK_ERRINFO	エラー情報を格納します
FLK_DATETIME	日付時刻を格納します
FLK_FINFO	ファイル情報を格納します
FLK_FINFOEX	ファイル情報を格納します
FLK_DINFO	ディスク情報を格納します
FLK_SYSINFO	システム情報を格納します

1.1.1 FLK_RSPRM

シリアルポートの情報を格納します。

```
typedef struct {  
    WORD    speed;        // 転送速度      FLK_B1200      1200 BPS  
                        //              FLK_B2400      2400 BPS  
                        //              FLK_B4800      4800 BPS  
                        //              FLK_B9600      9600 BPS  
                        //              FLK_B19K       19200 BPS  
                        //              FLK_B38K       38400 BPS  
                        //              FLK_B57K       57600 BPS  
                        //              FLK_B115K      115200 BPS  
    WORD    length;      // データ長      FLK_CHAR8  
    WORD    parity;      // パリティ      FLK_PARI_NON   パリティなし  
                        //              FLK_PARI_ODD     奇数  
                        //              FLK_PARI_EVN     偶数  
    WORD    stop_bit     // ストップビット FLK_STOP1      1 ビット  
                        //              FLK_STOP2      2 ビット  
} FLK_RSPRM;
```

以下の関数で使します。

- FLKOpen

1.1.2 FLK_STATUS

ライブラリの状態を格納します。

```
typedef struct {  
    WORD    status;           // コマンド実行ステータス  
    WORD    function;        // コマンド種類  
    TCHAR   *FileName;       // 送受信の名前  
    DWORD   total_size;      // 1回の転送の総バイト数  
    DWORD   total_count;     // 1回の転送の転送済みバイト数  
    DWORD   file_size;       // 転送中のファイルサイズ  
    DWORD   file_count;      // 転送中のファイルの転送済みバイト数  
    char    *Data;           // データ  
    DWORD   data_size;       // データ受信中のデータサイズ  
    DWORD   data_count;      // データ受信中の転送済みバイト数  
} FLK_STATUS;
```

コマンド実行ステータス: status

値	意味
FLK_STATUS_RUNNING	実行中
FLK_STATUS_END	正常終了
FLK_STATUS_ERROR	エラー発生により中断

コマンド種類: function
 送受信の名前: filename
 データ: data

function	filename	data	内容
FLK_SND_EXEC	送信ファイル名	-	ファイル送信中
FLK_RCV_EXEC	受信ファイル名	-	ファイル受信中
FLK_ADDSND_EXEC	送信ファイル名	-	ファイル追加送信中
FLK_ADDRCV_EXEC	送信元ファイル名	-	ファイル追加受信中
FLK_DELETE_EXEC	削除ファイル名	-	相手局のファイル削除要求中
FLK_DELETE_RECV	削除ファイル名	-	自局のファイル削除中
FLK_MOVE_EXEC	移動元のファイル名	-	相手局のファイルの移動要求中
FLK_MOVE_RECV	移動元のファイル名	-	自局のファイル移動中
FLK_MAKEDIR_RECV	作成ディレクトリ名	-	ディレクトリ作成中
FLK_DATETIME_EXEC	-	-	相手局の日付時刻変更要求中
FLK_DATETIME_RECV	-	-	日付時刻変更要求受信
FLK_DATETIME_SEND	-	-	相手局の日付時刻送信
FLK_MESSAGE_RECV	要求された表示メッセージ	-	メッセージ表示要求受信
FLK_BUZZER_RECV	-	-	ブザー鳴動要求受信
FLK_GETFILEINFO_RECV	処理中のファイル名	-	ファイル情報取得要求受信
FLK_SETFILEINFO_RECV	処理中のファイル名	-	ファイル情報設定要求中
FLK_GETDISKINFO_RECV	ドライブ名	-	ディスク情報受信中

以下の関数で使します。

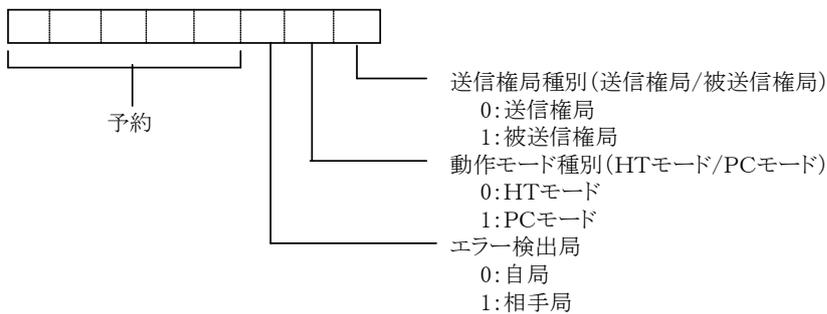
- FLKReadStatus

1.1.3 FLK_ERRINFO

エラー情報を格納します。

```
typedef struct {
    BYTE    kind;        // エラー種別
    BYTE    command;     // コマンド種別
    BYTE    category;   // カテゴリ
    BYTE    detail;     // エラー詳細
    DWORD   biosStat;   // GetLastError() で取得したエラーコード
} FLK_ERRINFO;
```

エラー種別: kind



コマンド種別: command

値	シンボル	意味
00h	FLK_CMD_NON	該当コマンドなし
01h	FLK_CMD_FSEND_TINFO	ファイル転送情報コマンド
02h	FLK_CMD_FSEND_FINFO	ファイル情報コマンド
03h	FLK_CMD_FRECV_TREQ	ファイル受信要求コマンド
04h	FLK_CMD_FADD	ファイル追加コマンド
05h	FLK_CMD_FDATA	ファイルデータコマンド
06h	FLK_CMD_FDEL	ファイル削除コマンド
07h	FLK_CMD_FMOV	ファイル移動コマンド
08h	FLK_CMD_MAKEDIR	ディレクトリ作成コマンド
09h	FLK_CMD_TIME_SET	日付時刻設定コマンド
0Ah	FLK_CMD_TIME_GET	日付時刻取得コマンド
0Bh	FLK_CMD_DISP	メッセージ表示コマンド
0Ch	FLK_CMD_BEEP	ブザー鳴動コマンド
0Dh	FLK_CMD_FINFO_GET	ファイル情報取得コマンド
0Eh	FLK_CMD_FINFO_SET	ファイル情報設定コマンド
0Fh	FLK_CMD_DINFO_GET	ディスク情報取得コマンド
10h	FLK_CMD_SYS_GET	システム情報取得コマンド
11h	FLK_CMD_IDLE	IDLE 通知コマンド
12h	FLK_CMD_END	終了指示コマンド

カテゴリ: category

エラー詳細: detail

カテゴリとエラー詳細コードの組み合わせによりエラー状態を表す。

値		意味	
カテゴリ	詳細		
正常終了状態			
00h	00h	正常終了	
DCh~F5h	00h	フォーマット指示コマンド(A~Z)	
F6h	00h	電源 OFF 終了通知	
F7h	00h	リセット指定終了通知	
F8h	00h	中断キーによる終了通知	
F9h~FFh	-	予約領域	
プロトコルエラー			
01h	00h	受信フレームファンクションコード未定義エラー	
	01h	受信フレームサブファンクションコード未定義エラー	
	03h	受信フレームチェックサムエラー	
	04h	シーケンスエラー	
	05h	シーケンス番号エラー	
	07h	受信フレーム内情報パラメータエラー	
	08h	受信タイムアウト	
	10h	コマンドレンダスエラー	
ファイルエラー[プロトコル論理]			
04h	00h	リードオンリファイルアクセスエラー	
ユーティリティエラー			
10h	00h	回線オープンエラー	回線がオープンされていない オープン時のエラーを確認
	01h	使用関数フェーズエラー	関数の使い方に誤りがある。 動作モード/送信権局モードを確認
	02h	使用関数パラメータエラー	関数パラメータに誤りがある 指定パラメータを確認
	03h	指定ファイル未検出エラー	指定されたファイルが存在しない 指定ファイルを確認
	04h	相手局未検出	セッション確立待ちタイムアウト 通信設定、回線経路を確認
	05h	システム日付設定エラー	指定日付を確認
	06h	システム時刻設定エラー	指定時刻を確認
	09h	致命的エラー	
	0Ah	通信中回線断エラー	通信中に回線が切断された。 回線経路を確認
	0Bh	ドライブ容量不足	指定ドライブの容量が足りない
0Ch	コマンド実行中エラー	コマンド実行中に、次のコマンドを実行した	

値		意味
カテゴリ	詳細	
ファイルエラー[ファイル関数]		
11h	00h	クリエートエラー
	01h	オープンエラー
	02h	リードエラー
	03h	ライトエラー
	04h	シークエラー
	05h	ファイル削除エラー
	06h	ディレクトリ削除エラー
	07h	ファイル名変更移動エラー
	08h	タイムスタンプ設定エラー
	09h	タイムスタンプ取得エラー
	0Ah	ファイル属性設定エラー
	0Bh	ファイル属性取得エラー
	0Ch	ディレクトリ作成エラー
0Dh	ファイルサイズ変更エラー	
システム異常エラー		
0Fh	0xh	FTP部内部エラー
	1xh	通信ユーティリティ内部エラー

以下の関数で使用します。

- FLKReadErrStat / FLKReadErrorStatus

1.1.4 FLK_DATETIME

日付時刻を格納します。

```
typedef struct {  
    BYTE    day;    // 日 (1-31)  
    BYTE    month; // 月 (1-12)  
    WORD    year;   // 年 (1980-2079)  
    BYTE    sec;    // 秒 (0-59)  
    BYTE    min;    // 分 (0-59)  
    BYTE    hour;   // 時 (0-23)  
} FLK_DATETIME;
```

以下の関数で使します。

- FLKMakeDir / FLKMakeDirectory
- FLKGetFileInfo
- FLKSetFileInfo
- FLKDateTime

1.1.5 FLK_FINFO

ファイル情報を格納します。

```
typedef struct {
    TCHAR      name[256] // 検索されたファイル名(フルパス)
    FLK_DATETIME  datetime; // 日付時刻格納エリア
    DWORD      size // サイズ
    BYTE      atr; // 属性
                // FLK_A_NORMAL :通常ファイル
                // FLK_A_HIDDEN  :不可視ファイル
                // FLK_A_RDONLY  :読み出し専用ファイル
                // FLK_A_SYSTEM  :システムファイル
                // FLK_A_ARCH   :アーカイブ
} FLK_FINFO;
```

以下の関数で使します。

- FLKGetFileInfo
- FLKSetFileInfo

1.1.6 FLK_FINFOEX

ファイル情報を格納します。

```
typedef struct {
    TCHAR    name[256]    // 検索されたファイル名(フルパス)
    BYTE     day;         // 日(1-31)
    BYTE     month;       // 月(1-12)
    WORD     year;        // 年(1980-2079)
    BYTE     sec;         // 秒(0-59)
    BYTE     min;         // 分(0-59)
    BYTE     hour;        // 時(0-23)
    DWORD    size;        // サイズ
    BYTE     atr;         // 属性
                                // FLK_A_NORMAL:通常ファイル
                                // FLK_A_HIDDEN :不可視ファイル
                                // FLK_A_RDONLY :読み出し専用ファイル
                                // FLK_A_SYSTEM :システムファイル
                                // FLK_A_ARCH  :アーカイブ
} FLK_FINFOEX;
```

以下の関数で使します。

- FLKGetFileInfoEx
- FLKSetFileInfoEx

1.1.7 FLK_DINFO

ディスク情報を格納します。

```
typedef struct {  
    DWORD    size;           // ディスク容量  
    DWORD    freex;         // ディスク空き容量  
    BYTE     status;        // ディスク状態  
                                // FLK_DINFO_NORMAL : ディスク有り(フォーマット済み)  
                                // FLK_DINFO_NOFMT  : ディスク有り(未フォーマット)  
                                // FLK_DINFO_NODISK : ディスクなし  
} FLK_DINFO;
```

以下の関数で使します。

- FLKGetDiskInfo

1.1.8 FLK_SYSINFO

システム情報を格納します。

```
typedef struct {  
    WORD    id;           // セッション ID(PC との接続時は 0)  
    BYTE    ftpver;      // プロトコルバージョン  
    BYTE    code[3];     // 機種コード  
    BYTE    model;       // モデル情報  
} FLK_SYSINFO;
```

以下の関数で使します。

- FLKGetSysInfo / FLKGetSystemInfo

1.2 関数一覧

関数名	機能
FLKOpen	回線オープン
FLKFileSend FLKSendFile	ファイル送信
FLKFileAdd FLKAddFile	ファイル追加
FLKFileRecv FLKReceiveFile	ファイル受信
FLKClose	回線クローズ
FLKReadErrStat FLKReadErrorStatus	エラー情報の取得
FLKReadStatus	進捗情報のリード
FLKIdle	アイドル遷移
FLKCmdRecv FLKReceiveCommand	コマンド受信待ち
FLKFileDelete FLKDeleteFile	ファイル削除
FLKFileMove FLKMoveFile	ファイル移動
FLKMakeDir FLKMakeDirectory	ディレクトリ作成
FLKGetFileInfo	ファイル情報の取得(C、C++)
FLKSetFileInfo	ファイル情報の設定(C、C++)
FLKGetFileInfoEx	ファイル情報の取得(C#、VB、C、C++)
FLKSetFileInfoEx	ファイル情報の設定(C#、VB、C、C++)
FLKGetDiskInfo	ディスク情報の取得
FLKDateTime	日付時刻の取得および設定
FLKGetSysInfo FLKGetSystemInfo	システム情報の取得
FLKMsgSend FLKSendMessage	画面表示メッセージの送信
FLKBeep	ブザー鳴動要求送信

1.2.1 FLKOpen

回線オープンし、通信ポートの初期化およびセッションの確立を行います。
相手局のシステム情報の取得を行います。

```
[C++]
HANDLE FLKOpen(
    TCHAR*    device,
    char*     cString,
    FLK_RSPRM* rsPrm,
    WORD      mode,
    HWND      hWnd,
    UINT      message
)
```

```
[Visual Basic]
FLKOpen(
    ByVal device As String, _
    ByVal cString() As Char, _
    ByVal rsprm As FLK_RSPRM,
    ByVal mode As Short, _
    ByVal hWnd As IntPtr,
    ByVal message As UInt32
) As IntPtr
```

```
[C#]
IntPtr FLKOpen(
    string    device, ,
    char []   cString,
    FLK_RSPRM sprm,
    ushort    mode
    IntPtr    hWnd,
    uint      message
)
```

パラメータ

device

使用デバイスを指定します。

"COM1" : シリアル(COM1)
"IRDA" : 赤外線
"LAN" : LAN
"USB" : USB

cString

通信パラメータを設定します。

COM1 指定時 : 無視します
IRDA 指定時 : 無視します
LAN 指定時 : ホスト PC の IP アドレスを指定します
USB 指定時 : 無視します

rsPrm

通信パラメータを設定します。

COM1 指定時 : COM ポート通信パラメータを指定します
IRDA 指定時 : 無視します
LAN 指定時 : 無視します
USB 指定時 : 無視します

mode

局モード

FLK_MODE_HT : HT モード
FLK_MODE_PC : PC モード

hWnd

メッセージを通知するウインドウの、ウインドウハンドルを指定します。
NULL を指定した場合はメッセージを通知しません。

message

アプリケーションへ通知するメッセージ番号を指定します。

戻り値

ハンドル値 : 正常終了
FLK_PRM : パラメータエラー
FLK_NG : パラメータ以外の異常終了
Device Emulator では、常に FLK_NG を返します。

メッセージ

WPARAM	内容
FLK_COMMAND_END	オープン処理が正常終了した
FLK_COMMAND_ERROR	オープン処理が異常終了した

1.2.2 FLKFileSend / FLKSendFile

指定された複数ファイルを一括して送信します。転送先ディレクトリが存在しない場合は、ディレクトリを自動的に生成します。

```
[C++]
DWORD FLKFileSend(
    HANDLE  hPort,
    WORD    mode,
    TCHAR*  fName,
    TCHAR*  dir,
    WORD    protect
)
```

```
[Visual Basic]
FLKSendFile(
    ByVal hPort      As IntPtr, _
    ByVal mode       As Short, _
    ByVal fName      As String, _
    ByVal dir        As String, _
    ByVal protect    As Short
) As Int32
```

```
[C#]
uint FLKSendFile(
    IntPtr  hPort,
    ushort mode,
    string  fName,
    string  dir,
    ushort protect
)
```

パラメータ

hPort

オープン時に取得したハンドル

mode

転送モード(通常転送か再起呼び出し転送かを指定します)

FLK_TRANS_NORMAL :通常転送

FLK_TRANS_RECURSIVE :再起呼び出し転送

fName

送信元ファイル名(フルパスで指定、ワイルドカード可能)

Dir

送信先フォルダ名(フルパスで指定)

protect

強制上書きフラグ(受信側に同一ファイルが書き込み禁止属性で存在した場合、属性変更して書き込む)

かを指定します)

FLK_PROTECT_VALID :強制書き込みしない

FLK_PROTECT_INVALID :強制書き込みする

戻り値

FLK_OK :正常終了

FLK_PRM :パラメータエラー

FLK_NG :パラメータ以外の異常終了

Device Emulator では、常に FLK_NG を返します。

メッセージ

WPARAM	内容
FLK_COMMAND_END	指定した全ファイルの送信が完了した
FLK_COMMAND_ERROR	ファイル送信がエラーにより終了した
FLK_STATUS_CHANGE1	1つのファイルの送信が始まった FLKReadStatus()関数で送信ファイル名を取得できます。
FLK_STATUS_CHANGE2	ファイル転送バイト数に変化があった FLKReadStatus()関数で転送バイト数を取得できます。

1.2.3 FLKFileAdd / FLKAddFile

指定したファイルを相手局の既存ファイルに追加します。
送信元、追加先ファイル名とも、複数ファイルの指定、およびワイルドカードの指定はできません。
追加先ファイル名が相手局側に存在しない場合は、新規にファイルを作成します。

```
[C++]
DWORD FLKFileAdd(
    HANDLE hPort,
    TCHAR* sfName,
    TCHAR* rfName
)
```

```
[Visual Basic]
FLKAddFile(
    ByVal hPort As IntPtr, _
    ByVal sfName As String, _
    ByVal rfName As String
) As Int32
```

```
[C#]
uint FLKAddFile(
    IntPtr hPort,
    string sfName,
    string rfName
)
```

パラメータ

hPort
オープン時に取得したハンドル

sfName
送信元ファイル名(フルパスで指定)

rfName
追加先ファイル名(フルパスで指定)

戻り値

FLK_OK : 正常終了
FLK_PRM : パラメータエラー
FLK_NG : パラメータ以外の異常終了
Device Emulator では、常に FLK_NG を返します。

メッセージ

WPARAM	内容
FLK_COMMAND_END	指定したファイルの追加が完了した
FLK_COMMAND_ERROR	ファイル追加がエラーにより終了した
FLK_STATUS_CHANGE1	ファイル追加送信が始まった

	FLKReadStatus()関数で送信ファイル名を取得できます。
FLK_STATUS_CHANGE2	ファイル転送バイト数に変化があった FLKReadStatus()関数で転送バイト数を取得できます。

1.2.4 FLKFileRecv / FLKReceiveFile

指定された複数ファイルを一括して受信します。受信先ディレクトリが存在しない場合は、ディレクトリを自動的に作成します。

```
[C++]
DWORD FLKFileRecv (
    HANDLE  hPort,
    WORD    mode,
    TCHAR*  fName,
    TCHAR*  dir,
    WORD    protect
)
```

```
[Visual Basic]
FLKReceiveFile (
    ByVal hPort      As IntPtr, _
    ByVal mode       As Short, _
    ByVal fName      As String, _
    ByVal dir        As String, _
    ByVal protect    As Short
) As Int32
```

```
[C#]
uint FLKReceiveFile (
    IntPtr  hPort,
    ushort mode,
    string  fName,
    string  dir,
    ushort  protect
)
```

パラメータ

hPort

オープン時に取得したハンドル

mode

転送モード(通常転送か再起呼び出し転送かを指定します)

FLK_TRANS_NORMAL :通常転送

FLK_TRANS_RECURSIVE :再起呼び出し転送

fName

受信ファイル名(フルパスで指定、ワイルドカード可能)

Dir

送信先フォルダ名(フルパスで指定)

protect

強制上書きフラグ(受信側に同一ファイルが書き込み禁止属性で存在した場合、属性変更して書き込む)

かを指定します)

FLK_PROTECT_VALID :強制書き込みしない

FLK_PROTECT_INVALID :強制書き込みする

戻り値

FLK_OK :正常終了

FLK_PRM :パラメータエラー

FLK_NG :パラメータ以外の異常終了

Device Emulator では、常に FLK_NG を返します。

メッセージ

WPARAM	内容
FLK_COMMAND_END	ファイルの受信が完了した
FLK_COMMAND_ERROR	ファイル受信がエラーにより終了した
FLK_STATUS_CHANGE1	1 ファイルの受信が始まった FLKReadStatus()関数でファイル名を取得できます。
FLK_STATUS_CHANGE2	ファイル転送バイト数に変化があった FLKReadStatus()関数で転送バイト数を取得できます。

1.2.5 FLKClose

終了指示コマンドを相手に送信することにより、セッションを開放します。その際、送信権モード時に限り、相手局に対して終了指示コマンドを送信します。ただし、すでにエラーが発生していた場合は送信しません。

```
[C++]
DWORD FLKClose(
    HANDLE hPort,
    WORD endKind
)
```

```
[Visual Basic]
FLKClose(
    ByVal hPort As IntPtr, _
    ByVal endKind As Integer
) As Int32
```

```
[C#]
uint FLKClose(
    IntPtr hPort,
    ushort endKind
)
```

パラメータ

hPort

オープン時に取得したハンドル

endKind

相手局への終了指示(送信権局モードのみ有効)

FLK_CLOSE_NORMAL: 正常終了

これらのパラメータは相手局へ通知するだけです。FLKIdle または FLKCmdRecv 関数が正常終了したとき、FLKReadErrStat 関数で相手局からどのコードを指定されたかを知ることができます。

戻り値

FLK_OK : 正常終了

FLK_PRM : パラメータエラー

FLK_NG : パラメータ以外の異常終了

Device Emulator では、常に FLK_NG を返します。

メッセージ

WPARAM	内容
FLK_COMMAND_END	クローズ処理が正常終了した。
FLK_COMMAND_ERROR	クローズ処理が異常終了した。

1.2.6 FLKReadStatus

ライブラリ内の通信状態を取得します。

```
[C++]
DWORD FLKReadStatus(
    HANDLE      hPort,
    FLK_STATUS* flkstatus
)
```

```
[Visual Basic]
FLKReadStatus(
    ByVal hPort      As IntPtr, _
    ByVal flkstatus As FLK_STATUS
) As Int32
```

```
[C#]
uint FLKReadStatus(
    IntPtr      hPort,
    FLK_STATUS flkstatus
)
```

パラメータ

hPort

オープン時に取得したハンドル

flkstatus

通信状態

詳細は FLK_STATUS 構造体を参照してください。

戻り値

FLK_OK : 正常終了

FLK_PRM : パラメータエラー

FLK_NG : パラメータ以外の異常終了

Device Emulator では、常に FLK_NG を返します。

メッセージ

メッセージはありません。

1.2.7 FLKReadErrStat / FLKReadErrorStatus

エラー情報を取得します。また、相手局からの終了指示コマンド受信時、カテゴリコード、エラー詳細コードを取得します。取得後エラー情報はクリアされます。

```
[C++]
DWORD FLKReadErrStat(
    HANDLE      hPort,
    FLK_ERRINFO* errinfo
)
```

```
[Visual Basic]
FLKReadErrorStatus(
    ByVal hPort As IntPtr, _
    ByVal errinfo As FLK_ERRINFO
) As Int32
```

```
[C#]
uint FLKReadErrorStatus(
    IntPtr      hPort,
    FLK_ERRINFO errinfo
)
```

パラメータ

hPort

オープン時に取得したハンドル

errinfo

エラー情報

詳細は FLK_ERRINFO 構造体を参照してください。

戻り値

FLK_OK : 正常終了

FLK_PRM : パラメータエラー

Device Emulator では、常に FLK_OK を返します。

メッセージ

メッセージはありません。

1.2.8 FLKIdle

アイドル通知送信後、相手局からのコマンド受信待ち状態となります。

HT モード時のみ使用可能です。

1 つのコマンドが終了するか、エラーが発生するまで実行します。

```
[C++]
DWORD FLKIdle(
    HANDLE hPort,
    TCHAR* script
)
```

```
[Visual Basic]
FLKIdle(
    ByVal hPort As IntPtr, _
    ByVal script As String
) As Int32
```

```
[C#]
uint FLKIdle(
    IntPtr hPort,
    string script
)
```

パラメータ

hPort

オープン時に取得したハンドル

script

スクリプトファイル名(ファイル名のみ。拡張子を含め 13 文字以内)

未設定時は NULL を指定します。

戻り値

FLK_OK : 正常終了

FLK_PRM : パラメータエラー

FLK_NG : パラメータ以外の異常終了

Device Emulator では、常に FLK_NG を返します。

メッセージ

WPARAM	内容
FLK_COMMAND_END	1 つのコマンドが終了した。
FLK_COMMAND_ERROR	エラーが発生により、コマンドが終了した。
FLK_STATUS_CHANGE1	コマンドに変化があったときに通知されます。 コマンドの内容は、FLKReadStatus 関数で取得できます。
FLK_STATUS_CHANGE2	ファイル転送バイト数に変化があった FLKReadStatus()関数で転送バイト数を取得できます。
FLK_INDICATION_END	相手局から終了指示コマンドを受信した。

1.2.9 FLKCmdRecv / FLKReceiveCommand

HT からのコマンド受信待ち状態となります。

PC モード時のみ使用可能です。

アイドル通知コマンド、終了指示コマンドを受信するか、エラーが発生するまで、相手局から受信したコマンドを順次実行します。

```
[C++]
DWORD FLKCmdRecv (
HANDLE  hPort,
WORD    *endKind,
TCHAR*  script
)
```

```
[Visual Basic]
FLKReceiveCommand (
ByVal hPort          As IntPtr, _
ByRef endKind        As Short, _
ByVal script         As String
) As Int32
```

```
[C#]
uint FLKReceiveCommand (
IntPtr      hPort,
ref ushort  endKind,
string      script
)
```

パラメータ

hPort

オープン時に取得したハンドル

endKind

相手局への終了指示(送信権局モードのみ有効)

FLK_CLOSE_NORMAL: 正常終了

これらのパラメータは相手局へ通知するだけです。FLKIdle または FLKCmdRecv 関数が正常終了したとき、FLKReadErrStat 関数で相手局からどのコードを指定されたかを知ることができます。

script

スクリプトファイル名(ファイル名のみ。拡張子を含め 13 文字以内)

相手局からスクリプトが指定された場合には、**script** で指定されたエリアヘスクリプト名が設定されます。スクリプトが指定されない場合は、NULL を設定します。

戻り値

FLK_OK :正常終了
FLK_PRM :パラメータエラー
FLK_NG :パラメータ以外の異常終了
Device Emulator では、常に FLK_NG を返します。

メッセージ

WPARAM	内容
FLK_COMMAND_END	相手局から終了指示又は IDLE 通知コマンドを受信した。
FLK_COMMAND_ERROR	エラー発生によりコマンドが終了した。
FLK_STATUS_CHANGE1	コマンドに変化があったときに通知されます。 コマンドの内容は、FLKReadStatus 関数で取得できます。
FLK_STATUS_CHANGE2	ファイル転送バイト数に変化があった FLKReadStatus 関数で転送バイト数を取得できます。

1.2.10 FLKFileDelete / FLKDeleteFile

ファイルを削除します。

```
[C++]
DWORD FLKFileDelete(
    HANDLE hPort,
    TCHAR* fName
)
```

```
[Visual Basic]
FLKDeleteFile(
    ByVal hPort As IntPtr, _
    ByVal fName As String
) As Int32
```

```
[C#]
uint FLKDeleteFile(
    IntPtr hPort,
    string fName
)
```

パラメータ

hPort

オープン時に取得したハンドル

fName

削除するファイル名(フルパスで指定。ワイルドカード可能)

戻り値

FLK_OK : 正常終了

FLK_PRM : パラメータエラー

FLK_NG : パラメータ以外の異常終了

Device Emulator では、常に FLK_NG を返します。

メッセージ

WPARAM	内容
FLK_COMMAND_END	ファイル削除が正常終了した。
FLK_COMMAND_ERROR	ファイル削除がエラー終了した。

備考

読み取り専用ファイルを、削除する事は出来ません。

1.2.11 FLKFileMove / FLKMoveFile

ファイルを移動します。同一フォルダを指定している場合は、ファイル名を変更します。

```
[C++]
DWORD FLKFileMove(
    HANDLE hPort,
    TCHAR* sfName,
    TCHAR* dfName
)
```

```
[Visual Basic]
FLKMoveFile(
    ByVal hPort As IntPtr, _
    ByVal sfName As String, _
    ByVal dfName As String
) As Int32
```

```
[C#]
uint FLKMoveFile(
    IntPtr hPort,
    string sfName,
    string dfName
)
```

パラメータ

hPort

オープン時に取得したハンドル

sfName

移動元ファイル名(フルパスで指定)

dfName

移動先ファイル名(フルパスで指定)

戻り値

FLK_OK : 正常終了

FLK_PRM : パラメータエラー

FLK_NG : パラメータ以外の異常終了

Device Emulator では、常に FLK_NG を返します。

メッセージ

WPARAM	内容
FLK_COMMAND_END	ファイル削除が正常終了した。
FLK_COMMAND_ERROR	ファイル削除がエラー終了した。

備考

別ドライブへの移動は出来ません。

1.2.12 FLKMakeDir / FLKMakeDirectory

ディレクトリを作成します。

```
[C++]
DWORD FLKMakeDir(
    HANDLE          hPort,
    TCHAR*          mDir,
    FLK_DATETIME*  datetime,
    BYTE            atr
)
```

```
[Visual Basic]
FLKMakeDirectory(
    ByVal hPort    As IntPtr, _
    ByVal mDir     As String, _
    ByVal datetime As FLK_DATETIME, _
    ByVal atr      As Byte
) As Int32
```

```
[C#]
uint FLKMakeDirectory(
    IntPtr          hPort,
    string          mDir,
    FLK_DATETIME  datetime, _
    byte            atr
)
```

パラメータ

hPort

オープン時に取得したハンドル

mDIR

作成ディレクトリ名

datetime

日付格納エリア

atr

属性(OR 指定による複数指定可)

FLK_A_NORMAL : 通常ファイル
FLK_A_HIDDEN : 不可視ファイル
FLK_A_RDONLY : 読み出し専用ファイル
FLK_A_SYSTEM : システムファイル
FLK_A_SUBDIR : ディレクトリ
FLK_A_ARCH : アーカイブ

- FLK_A_NORMAL の値は 0 なので、設定する必要はありません。また、FLK_A_SUBDIR はライブラリ内で自動的に OR されます。

- 日付時刻を指定しない場合には、`datetime` 構造体のメンバーの、`year` に `0xFFFF`、`day,month,sec,min,hour` に `0xFF` を設定してください。

戻り値

FLK_OK : 正常終了
FLK_PRM : パラメータエラー
FLK_NG : パラメータ以外の異常終了
Device Emulator では、常に FLK_NG を返します。

メッセージ

WPARAM	内容
FLK_COMMAND_END	ディレクトリ作成が正常終了した。
FLK_COMMAND_ERROR	ディレクトリ作成がエラー終了した。

備考

CF、SD カードフォルダを作成する事は出来ません。
HT 対向時のみ有効です。

1.2.13 FLKGetFileInfo

相手局側の指定ファイル情報(ファイルサイズ・タイムスタンプ・属性)の取得を行います。
検索ファイル名と一致するファイルの情報がファイル情報エリアに設定されます。
ワイルドカード指定時は 1 回目に"最初の取得"、2 回目以降に"次情報取得"を指定します。
ワイルドカード指定時は、この関数を連続的に呼ぶ必要があります。
他の通信関数を使用すると、次情報取得は行なえません。

```
[C++]
DWORD FLKGetFileInfo(
    HANDLE      hPort,
    WORD        mode,
    TCHAR*      fName,
    FLK_FINFO*  fInfo
)
```

パラメータ

hPort

オープン時に取得したハンドル

mode

最初/次フラグ

FLK_GET_FIRST : 最初の取得(1 ファイル指定又はワイルドカード指定時の 1 回目)

FLK_GET_NEXT : 次情報取得(ワイルドカード指定時の 2 回目以降)

fName

検索ファイル名エリア(ワイルドカード指定可。複数指定不可。"次情報取得"では参照しません。)

finfo

ファイル情報エリア(検索したファイルの情報が設定されます)

該当ファイルが存在しない場合には、各パラメータに 0x00 が設定されます。

戻り値

FLK_OK : 正常終了

FLK_PRM : パラメータエラー

FLK_NG : パラメータ以外の異常終了

Device Emulator では、常に FLK_NG を返します。

メッセージ

WPARAM	内容
FLK_COMMAND_END	正常終了した。
FLK_COMMAND_ERROR	エラー終了した。

1.2.14 FLKSetFileInfo

相手局側の指定ファイル情報(ファイルサイズ・タイムスタンプ・属性)の更新を行います。
ファイル情報エリアの内容をファイル名エリアと一致するファイルに設定します。

```
[C++]
DWORD FLKSetFileInfo(
HANDLE      hPort,
FLK_FINFO*  fInfo
)
```

パラメータ

hPort

オープン時に取得したハンドル

fInfo

ファイル情報エリア

戻り値

FLK_OK : 正常終了

FLK_PRM : パラメータエラー

FLK_NG : パラメータ以外の異常終了

Device Emulator では、常に FLK_NG を返します。

メッセージ

WPARAM	内容
FLK_COMMAND_END	正常終了した。
FLK_COMMAND_ERROR	エラー終了した。

備考

HT 対向時のみ有効です。

1.2.15 FLKGetFileInfoEx

相手局側の指定ファイル情報(ファイルサイズ・タイムスタンプ・属性)の取得を行います。検索ファイル名と一致するファイルの情報がファイル情報エリアに設定されます。

ワイルドカード指定時は 1 回目に“最初の取得”、2 回目以降に“次情報取得”を指定します。ワイルドカード指定時は、この関数を連続的に呼ぶ必要があります。

他の通信関数を使用すると、次情報取得は行なえません。

```
[C++]
DWORD FLKGetFileInfoEx(
    HANDLE      hPort,
    WORD        mode,
    TCHAR*      fName,
    FLK_FINFOEX* fInfoex
)
```

```
[Visual Basic]
FLKGetFileInfoEx(
    ByVal hPort As IntPtr, _
    ByVal mode As Short, _
    ByVal fName As String, _
    ByVal fInfoex As FLK_FINFOEX
) As Int32
```

```
[C#]
uint FLKGetFileInfoEx(
    IntPtr      hPort,
    ushort     mode,
    string      fName,
    FLK_FINFOEX fInfoex
)
```

パラメータ

hPort

オープン時に取得したハンドル

mode

最初/次フラグ

FLK_GET_FIRST : 最初の取得(1 ファイル指定又はワイルドカード指定時の 1 回目)

FLK_GET_NEXT : 次情報取得(ワイルドカード指定時の 2 回目以降)

fName

検索ファイル名エリア(ワイルドカード指定可。複数指定不可。“次情報取得”では参照しません。)

fInfoex

ファイル情報エリア(検索したファイルの情報が設定されます)

該当ファイルが存在しない場合には、各パラメータに 0x00 が設定されます。

戻り値

FLK_OK :正常終了
FLK_PRM :パラメータエラー
FLK_NG :パラメータ以外の異常終了
Device Emulator では、常に FLK_NG を返します。

メッセージ

WPARAM	内容
FLK_COMMAND_END	正常終了した。
FLK_COMMAND_ERROR	エラー終了した。

1.2.16 FLKSetFileInfoEx

相手局側の指定ファイル情報(ファイルサイズ・タイムスタンプ・属性)の更新を行います。
ファイル情報エリアの内容をファイル名エリアと一致するファイルに設定します。

```
[C++]
DWORD FLKSetFileInfoEx(
    HANDLE      hPort,
    FLK_FINFOEX* fInfoex
)
```

```
[Visual Basic]
FLKSetFileInfoEx(
    ByVal hPort As IntPtr, _
    ByVal fInfoex As FLK_FINFOEX
) As Int32
```

```
[C#]
uint FLKSetFileInfoEx(
    IntPtr      hPort,
    FLK_FINFOEX fInfoex
)
```

パラメータ

hPort

オープン時に取得したハンドル

fInfoex

ファイル情報エリア

戻り値

FLK_OK : 正常終了

FLK_PRM : パラメータエラー

FLK_NG : パラメータ以外の異常終了

Device Emulator では、常に FLK_NG を返します。

メッセージ

WPARAM	内容
FLK_COMMAND_END	正常終了した。
FLK_COMMAND_ERROR	エラー終了した。

備考

HT 対向時のみ有効です。

1.2.17 FLKGetDiskInfo

相手局側の指定ドライブ情報の取得を行います。指定ドライブの情報がドライブ情報エリアへ設定されます。

```
[C++]
DWORD FLKGetDiskInfo(
    HANDLE    hPort,
    BYTE      drive,
    FLK_DINFO* dinfo
)
```

```
[Visual Basic]
FLKGetDiskInfo(
    ByVal hPort    As IntPtr, _
    ByVal drive    As Byte, _
    ByVal dinfo    As FLK_DINFO
) As Int32
```

```
[C#]
uint FLKGetDiskInfo(
    IntPtr    hPort,
    byte      drive,
    FLK_DINFO dinfo
)
```

パラメータ

hPort

オープン時に取得したハンドル

drive

ドライブ名エリア、'A'~'Z'のいずれか

dinfo

ドライブ情報エリアアドレス(検索したドライブの情報が設定されます)
コマンド正常終了メッセージ受信時に、**dinfo** に値が設定されます。

戻り値

FLK_OK : 正常終了

FLK_PRM : パラメータエラー

FLK_NG : パラメータ以外の異常終了

Device Emulator では、常に FLK_NG を返します。

メッセージ

WPARAM	内容
FLK_COMMAND_END	正常終了した。
FLK_COMMAND_ERROR	エラー終了した。

備考

ディスク容量が大きい時はプロトコル上、正しいサイズを取得する事ができません。
HT 対向時のみ有効です。

1.2.18 FLKDateTime

日時を取得/設定します。

取得の場合は、日付時刻エリアへ相手局のシステム日付が設定されます。設定の場合は、日付時刻エリアの値を相手局へ送信します。

```
[C++]
DWORD FLKDateTime(
    HANDLE          hPort,
    WORD            mode,
    FLK_DATETIME*  dateTime
)
```

```
[Visual Basic]
FLKDateTime(
    ByVal hPort    As IntPtr, _
    ByVal mode     As Short, _
    ByVal dateTime As FLK_DATETIME
) As Int32
```

```
[C#]
uint FLKDateTime(
    IntPtr          hPort,
    ushort         mode,
    FLK_DATETIME  datetime
)
```

パラメータ

hPort

オープン時に取得したハンドル

mode

取得/設定フラグ

FLK_GET_MODE :取得

FLK_SET_MODE :設定

datetime

設定日付時刻格納エリア

戻り値

FLK_OK :正常終了

FLK_PRM :パラメータエラー

FLK_NG :パラメータ以外の異常終了

Device Emulator では、常に FLK_NG を返します。

メッセージ

WPARAM	内容
FLK_COMMAND_END	正常終了した。

FLK_COMMAND_ERROR

エラー終了した。

備考

設定は、HT 対向時のみ有効です。

1.2.19 FLKGetSysInfo / FLKGetSystemInfo

相手局側のシステム情報を取得します。相手局が PC の場合は、接続セッション番号も返します。(相手が HT の場合は 0)

これらの情報はオープン時に取得しているため、通信は行わずに情報のみ返します。

```
[C++]
DWORD FLKGetSysInfo(
    HANDLE      hPort,
    FLKSYSINFO  sysinfo
)
```

```
[Visual Basic]
FLKGetSystemInfo(
    ByVal hPort      As IntPtr, _
    ByVal sysinfo    As FLK_SYSINFO
) As Int32
```

```
[C#]
uint FLKGetSystemInfo(
    IntPtr      hPort,
    FLK_SYSINFO  sysinfo
)
```

パラメータ

hPort

オープン時に取得したハンドル

sysinfo

取得システム情報エリア (検索された情報が設定されます)

本ライブラリでは、機種コードが DT-5200 では"021"、DT-X7 では"055"、DT-9800 では"780"、DT-5300 では"022"、IT-300 では"031"、DT-X8 では"052"、相手局が PC では"AT"が取得されます。

戻り値

FLK_OK : 正常終了

FLK_PRM : パラメータエラー

FLK_NG : パラメータ以外の異常終了

Device Emulator では、常に FLK_NG を返します。

メッセージ

メッセージはありません。

1.2.20 FLKMsgSend / FLKSendMessage

相手局に表示するメッセージを送信します。

```
[C++]
DWORD FLKMsgSend(
    HANDLE hPort,
    TCHAR* msg
)
```

```
[Visual Basic]
FLKSendMessage(
    ByVal hPort As IntPtr,
    ByVal msg As String
) As Int32
```

```
[C#]
uint FLKSendMessage(
    IntPtr hPort,
    string msg
)
```

パラメータ

hPort
オープン時に取得したハンドル

msg
表示メッセージ格納エリア

戻り値

FLK_OK : 正常終了
FLK_PRM : パラメータエラー
FLK_NG : パラメータ以外の異常終了
Device Emulator では、常に FLK_NG を返します。

メッセージ

WPARAM	内容
FLK_COMMAND_END	正常終了した。
FLK_COMMAND_ERROR	エラー終了した。

備考

HT 対向時のみ有効です。

1.2.21 FLKBeep

相手局側にブザー鳴動要求の送信をします。

```
[C++]
DWORD FLKBeep(
    HANDLE hPort
)
```

```
[Visual Basic]
FLKBeep(
    ByVal hPort As IntPtr
) As Int32
```

```
[C#]
uint FLKBeep(
    IntPtr hPort
)
```

パラメータ

hPort

オープン時に取得したハンドル

戻り値

FLK_OK : 正常終了

FLK_PRM : パラメータエラー

FLK_NG : パラメータ以外の異常終了

Device Emulator では、常に FLK_NG を返します。

メッセージ

WPARAM	内容
FLK_COMMAND_END	正常終了した。
FLK_COMMAND_ERROR	エラー終了した。

備考

HT 対向時のみ有効です。

ブザーは鳴動しません

2. 通信仕様

本ライブラリは、以下の通信ポートをサポートします。

	赤外線(IrDA)	シリアル	LAN	USB
DT-5200	○	×	○	○
DT-X7	○	×	○	○
DT-9800	○	○	△※	×
DT-5300	×	×	○	○
IT-300	×	×	○	○
DT-X8	×	×	○	○
IT-9000	×	×	○	○
IT-G500	×	×	○	○
DT-X100	×	×	○	○
DT-X200	×	×	○	○

○ 該当デバイス搭載機種

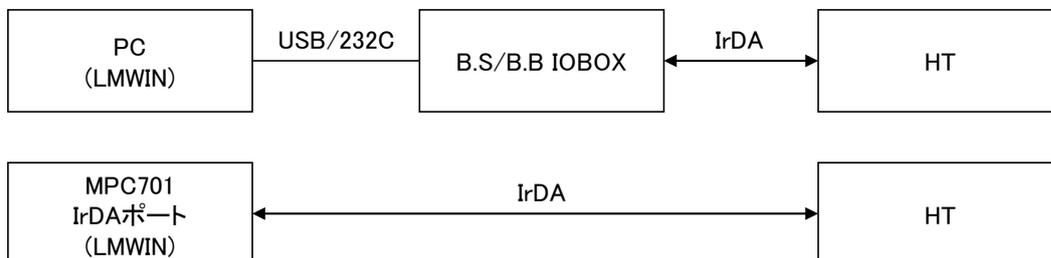
×

※ 無線 LAN を内蔵していない機種のため、無線 LAN カードを使用

2.1 通信構成

本ライブラリでは、以下の構成でのファイル転送機能を提供します。

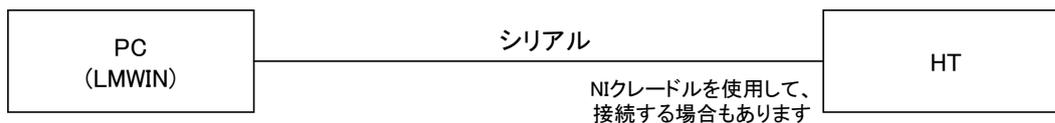
1. B.S/B.B IOBOX、及び FIVA(MPC701)を使用した IrDA 通信



2. IrDA による本体間通信



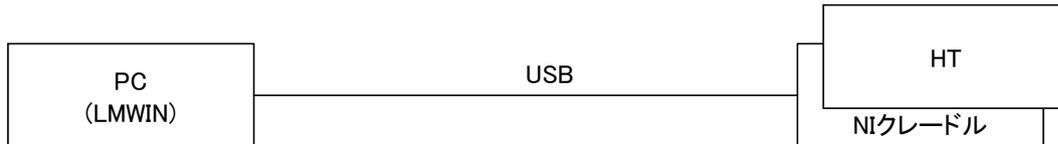
3. シリアルケーブルによるホスト PC 間通信



4. 無線 LAN/有線 LAN を使用したホスト PC 間通信



5. USB ケーブルによるホスト PC 間通信



2.2 通信パラメータ

共通パラメータ

以下のパラメータはレジストリにより設定を行います。

	機能	設定範囲	デフォルト値
セッション確立タイムアウト	回線オープン時、セッション確立までの待ち時間	0-3600 秒	60 秒
受信待ちタイムアウト	コマンド/レスポンス受信待ち時間	0-600 秒	30 秒
セッション終了待ちタイムアウト	終了指示コマンド送信側が、相手局セッション終了を確認するまでの時間	0-600 秒	10 秒

※ レジストリの設定内容は、レジストリの項目を参照してください。

IrDA パラメータ

パラメータ	サポートする値
最大通信速度	設定不可 最大通信速度は、システムの設定(レジストリ)によります。
Windows98 モード	B.S/B.BIOBOX を使用して、Windows98 と通信する場合に設定するパラメータです。 FLCE.EXE の /8 パラメータに相当する機能です。 本パラメータは、レジストリにより設定します。

シリアル通信パラメータ

パラメータ	サポートする値
回線速度	選択可能(1200/2400/4800/9600/19200/38400/57600/115200)
データ長	8 ビット固定
パリティビット	選択可能(無し/偶数/奇数)
ストップビット	選択可能(1 ビット/2 ビット)
フロー制御	RS/CS フロー制御のみ

LAN 通信パラメータ

パラメータ	サポートする値
接続先 IP アドレス	設定可能

USB 通信パラメータ

パラメータ	サポートする値
無し	

2.3 動作モード

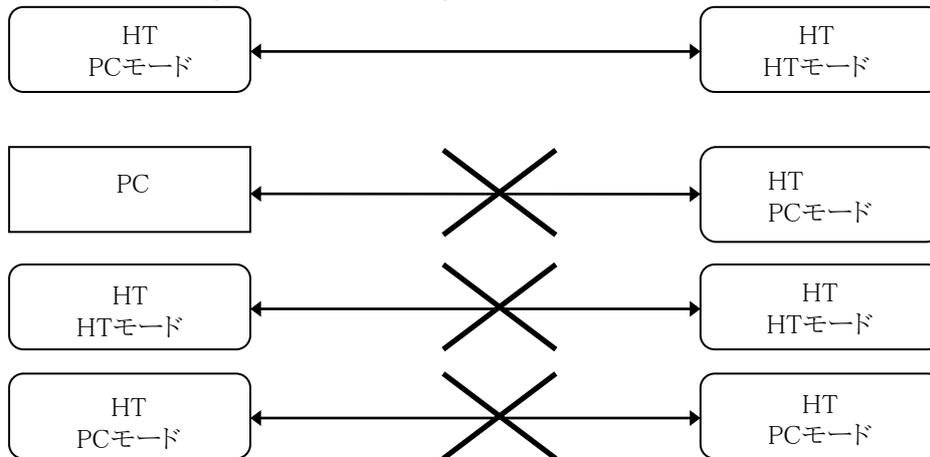
本ライブラリでは接続構成により以下のモードをサポートします。
これらのモードはオープン時に選択します。

HT モード

セッション※確立後、コマンドを送信する権利(以後、送信権とする)を持つモードです。
PC-HT 間通信時及び HT-HT 間通信時(どちらか一方の HT)に選択します。

PC モード

セッション確立後、HT からのコマンド待ちとなる、擬似 PC モードです。
HT-HT 間通信時(どちらか一方の HT)に選択します。



※セッションとは回線オープン時に、相手局確認等のネゴシエーションをさします。

2.4 コマンド送信権

HT モード

1. コマンド送信権モード(HT 側がコマンド指定を行うモード)

セッション確立後、HT 側は送信権を有し、PC(PC モードを含む。以後、PC とする)に各コマンドを送信することにより、各機能を実現します。この時、PC 側はサーバーモードにする必要があります。

2. アイドルモード(PC 側がコマンド指定を行うモード)

送信権を PC 側に譲渡する場合は IDLE 通知コマンドを送信します。その後、HT は PC からのコマンド受信待ち状態となります。この時、HT はアイドルモード(FLKIdle 関数)にする必要があります。

実行状態は通信状態取得関数(FLKReadStatus)で取得します。

IDLE 通知コマンド送信の際、PC ヘスクリプトファイルの指定が行えます。但し、PC モードの HT(FLCE.EXE)にはスクリプトファイル実行機能はありません。

PC モード

PC はセッション確立後に HT からのコマンド受信待ちとなり、以後受信したコマンドに従い、処理を実行します。

HT から IDLE 通知コマンドを受信した場合、PC に送信権が移ります。

2.5 処理概要

以下に各関数内の処理概要及びエラー発生時の処理を示します。

エラー発生時は、直ちに通信を終了します。この場合、送信権の有無に関らず、先にエラーを検出した側がエラー情報(カテゴリコード・エラー詳細コード)を終了指示コマンドに設定し、相手局へ送信します。相手局は、受信した終了指示コマンドのエラー情報により、異常終了を検出します。

関数	送信権局の処理	被送信権局の処理	エラー発生時の処理
ファイル送信	コマンド送信後、指定ファイルを順次送信。	指定ファイルを順次受信。	受信中ファイルの削除を行う。
ファイル受信	コマンド送信後、指定ファイルを順次受信。	指定ファイルを順次送信。	
ファイル追加	コマンド送信後、指定ファイルを送信。	転送ファイルをテンポラリファイル (FL.ADD)に受信後、ファイルを追加する。追加完了後、テンポラリファイルを削除。	テンポラリファイルを削除。
ファイル削除	コマンド送信。	指定ファイル/ディレクトリを削除。	削除したファイル/ディレクトリの復旧は行わない。
ファイル移動	コマンド送信。	指定ファイルを移動。	移動後の削除ファイルは復旧しない。
ディレクトリ作成	コマンド送信。	指定ディレクトリを作成。	作成したディレクトリは削除しない。
日付時刻の取得	コマンド送信後、日付時刻情報を受信する。	システム日付時刻情報を送信。	
日付時刻の設定	コマンド送信。	日付時刻をシステムに設定。	設定後の日付時刻は復旧しない。
ファイル情報の取得	コマンド送信後、ファイル情報を受信する。	指定ファイル情報を送信する。	
ファイル情報の設定	コマンド送信。	指定ファイルの情報を変更する。	設定後のファイル情報は復旧しない。
ディスク情報の取得	コマンド送信後、ディスク情報を受信する。	指定ディスク情報を送信する。	
システム情報の取得	セッション確立時に相手局情報を取得するため、通信は行わない	セッション確立時に相手局情報を取得するため、通信は行わない	
画面メッセージ表示	コマンド送信。	画面メッセージ表示要求を受信したことをアプリケーションに通知する。メッセージ表示は行わない。	
ブザー鳴動	コマンド送信。	ブザー鳴動要求を受信したことをアプリケーションに通知する。ブザー鳴動は行わない。	

IDLE 通知	(HT モードのみ) コマンドを送信後、 コマンド受信待ちとなる。	(PC モードのみ) コマンド送信権を取得する。	
データ転送	データ転送。	データ転送要求を受信したことをアプリケーションに通知する。	

3. ファイル送受信基本機能

複数ファイルの送信および受信を行うための、基本機能を提供します。

3.1 通信基本関数

ファイル送受信およびリモート操作関数を使用する際に必要となる基本関数を以下に示します。

通信ポートの初期化(FLKOpen 関数)

通信ポートの初期化を行います。

通信ポート毎のパラメータおよび設定値を以下に示します。

COM1	
通信速度	FLK_B1200(1200bps)/FLK_B2400(2400bps)/FLK_B19K(19.2kbps) FLK_B38K(38.4kbps)/FLK_B57K(57.6kbps)/FLK_B115K(115.2kbps)
データ長	FLK_CHAR8(データ長 8 ビット)
パリティ	FLK_PARI_ODD(奇数)/FLK_PARI_EVEN(偶数)/FLK_PARI_NON(パリティなし)
ストップビット	FLK_STOP1(ストップビット 1)/FLK_STOP2(ストップビット 2)
IrDA	
設定値なし	
LAN	
接続先 IP アドレス	IP アドレスを指定します。
USB	
設定値なし	

APO を禁止に設定します。

通信ポートのクローズ(FLKClose 関数)

- 通信終了および回線ポートのクローズを行います。通信を終了するために相手局に終了指示コマンドを送信します。
- ただし、既に相手局より終了指示コマンドを受信していた場合は、終了指示コマンドの送信は行いません。
- 終了指示コマンドにはエラー情報(カテゴリコード・エラー詳細コード)を設定します。
- 回線オープン時にセーブした APO 設定を復旧します。

エラー詳細情報の取得(FLKReadErrStat 関数)

エラー情報の取得を行います。

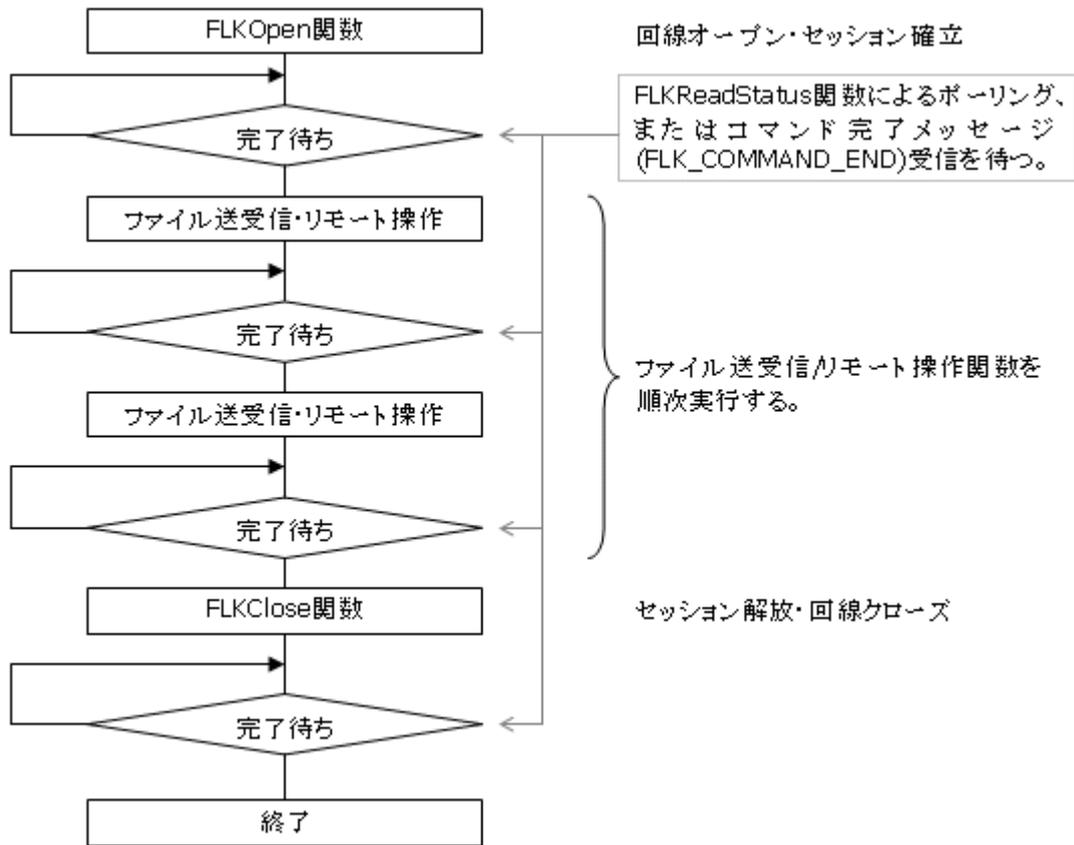
発生エラーコード、相手局からの終了指示コマンドのエラー情報(カテゴリコード・エラー詳細コード)等の取得が可能です。

通信状態の取得(FLKReadStatus 関数)

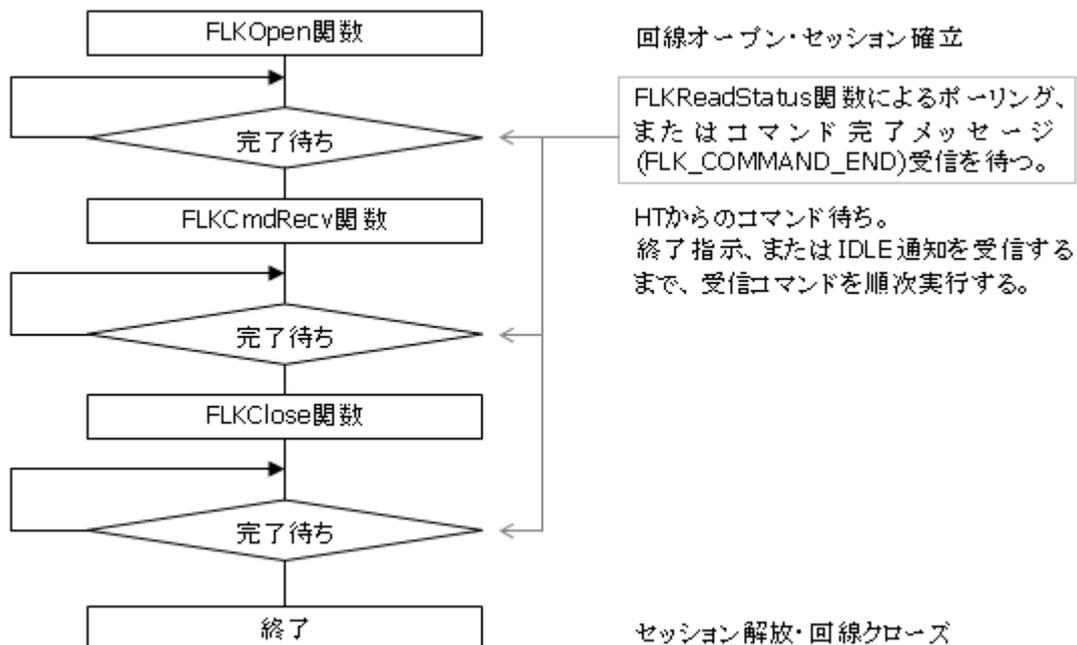
通信の進捗状態(現在実行中の機能、ファイルの転送バイト数など)を取得できます。
最後に実行したコマンドが終了したか、実行中かを知ることができます。

(1) HT コマンド送信による通信

【HT モード基本フロー】

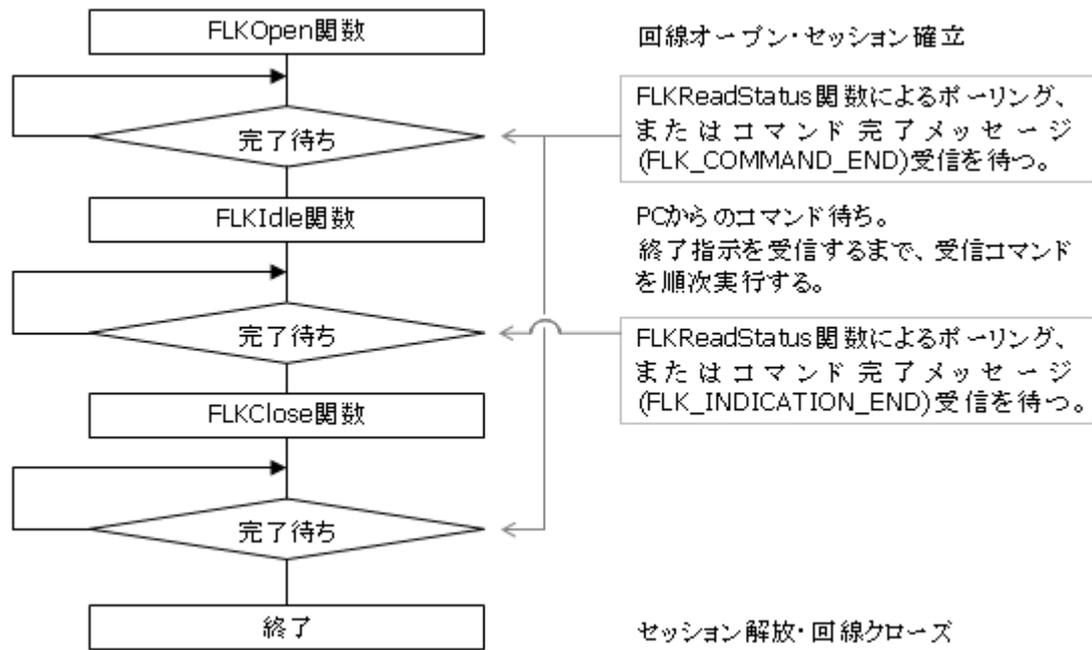


【PC モード基本フロー】



(2) PCコマンド送信による通信

【HT モード基本フロー】



3.2 ファイル送受信関数

相手局とのファイル転送(送信、追加、受信)を行うための関数です。

送信権局はファイル送信、追加および受信コマンドを使用して、相手局とのファイル転送を実現します。被送信権局は、IDLE 状態(HT モード)、PC モードコマンド待ち状態(PC モード)にて相手からのコマンドを受け付けます。

なお、ライブラリには、ファイル転送時の進捗表示を行う機能はありません。アプリケーションで必要に応じて表示を行ってください。

進捗表示は、メッセージ『FLK_STATUS_CHANGE2』を受信した時に、FLKReadStatus 関数を呼び FLK_STATUS 構造体の「total_size」、「total_count」、「file_size」、「file_count」を元に計算を行い進捗表示を行って下さい。

メッセージ『FLK_STATUS_CHANGE2』は、ファイル送信関数(FLKFileSend)、ファイル追加関数(FLKFileAdd)、ファイル受信関数(FLKFileRecv)、コマンド受信待ち関数(FLKIdle、FLKCmdRecv)の時に受信する事が可能です。

ファイル送信(FLKFileSend 関数)

複数ファイルの送信を一括して行います。

送信先に指定ディレクトリが存在しない場合は、自動的に作成します。

送信ファイルに対して以下のオプションを選択することができます。

- リードオンリーファイル強制上書きオプション

送信ファイルが、既に受信側にリードオンリーファイルとして存在していた場合、強制的に上書きすることができます。この指定が無い場合にリードオンリーファイルへの書き込みを行うと、エラーとなります。

- 再帰呼び出し指定オプション

送信ファイルパス名で指定したディレクトリ傘下のすべてのファイルが転送対象となります。指定ディレクトリ傘下にサブディレクトリが存在した場合はサブディレクトリ名を付加してファイルの送信を行います。

(例)

[送信ファイル名]

"¥SEND¥AAA.DAT"

[送信先ディレクトリ名]

"¥RECV¥"

(送信側ディレクトリ構成)

¥---SEND¥----SUB1¥----AAA.DAT

| ----SUB2¥----BBB.DAT

| ----AAA.DAT

| ----BBB.DAT

C:¥---RECV¥---SUB1¥----AAA.DAT

| ----AAA.DAT

- ワイルドカードの使用

送信ファイル名にはワイルドカード(*,?)を使用することができます。

ファイル追加関数(FLKFileAdd 関数)

HT 側ファイルを相手局側ファイルにアペンドすることができます。

相手局側に指定したアペンドファイルが存在しない場合は、新規作成となります。

複数ファイルおよびワイルドカードの指定はできません。

ファイル受信(FLKFileRecv 関数)

複数ファイルの受信を一括して行うことができます。

受信ファイルに対して以下のオプションを選択することができます。

- リードオンリーファイル強制ライトオプション

受信ファイルが、既に受信側にリードオンリーファイルとして存在していた場合でも、上書きすることができます。この指定が無い場合にリードオンリーファイルへの書き込みを行うと、エラーとなります。

- 再帰呼び出し指定オプション

受信ファイルパス名で指定したディレクトリ傘下のすべてのファイルが転送対象となります。指定ディレクトリ傘下にサブディレクトリが存在した場合はサブディレクトリ名を付加してファイルの受信を行います。

- ワイルドカードの使用

受信ファイル名にはワイルドカード(*,?)を使用することができます。

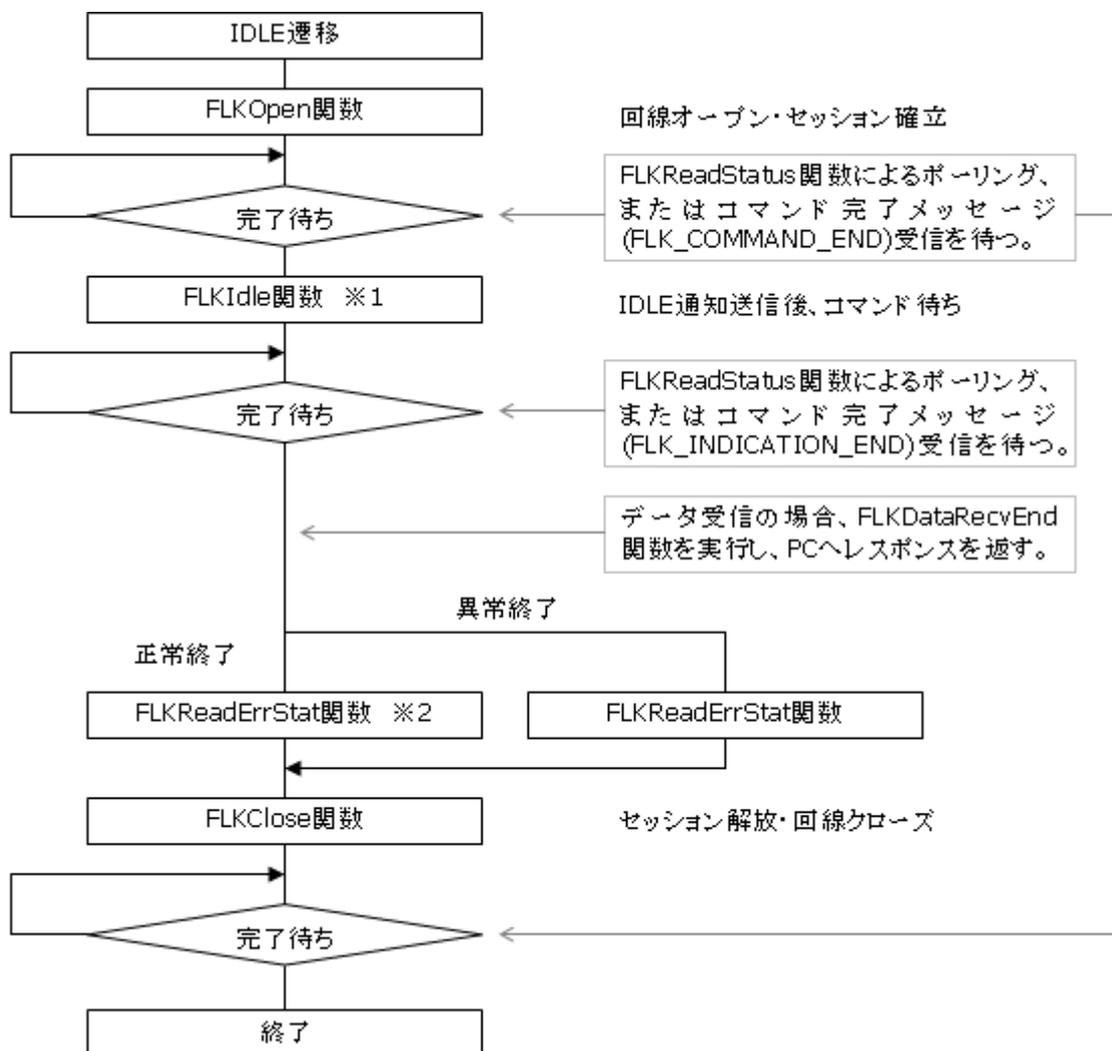
アイドル通知(FLKIdle 関数)

相手局側に送信権を渡し、コマンド待ち状態となります。

終了指示コマンドを受信またはエラー発生まで、受信したコマンドを順次実行します。

また、PC のスクリプトファイルの実行を指示することができます。

[IDLE 遷移基本フロー]



相手局からの終了指示コマンド受信またはエラー発生まで、受信コマンドを順次実行する。必要に応じて相手局からの終了指示コマンド詳細情報(フォーマット指示、リセット指示等)の取得が可能。

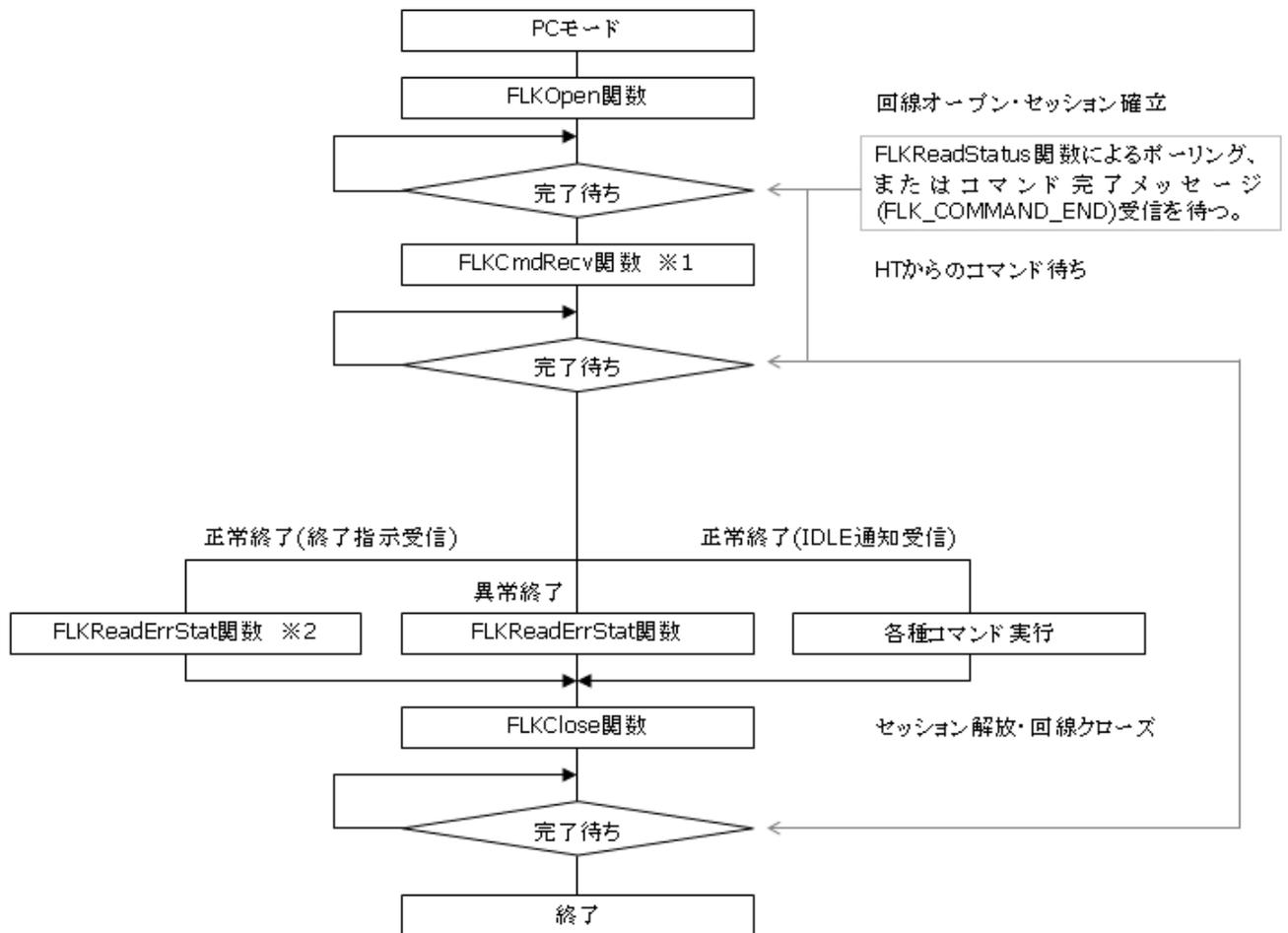
PC モードコマンド待ち(FLKCmdRecv 関数)

HT からのコマンド受信待ち状態となります。PC モードでのみ使用可能です。

セッション確立直後は、HT 側に送信権があるため、PC モードではオープン直後にこの関数を用いてコマンド待ちとなる必要があります。

終了指示コマンドか、IDLE 通知コマンドを受信またはエラー発生まで、受信したコマンドを順次実行します。

[PC モード時の基本フロー]



HT から終了指示コマンドまたは IDLE 通知コマンドを受信するか、エラーが発生するまで、受信コマンドを順次実行する。

必要に応じて相手局からの終了指示コマンド詳細情報(フォーマット指示、リセット指示等)の取得が可能。

3.3 リモート操作関数

相手局側のファイル操作、環境情報の取得/設定を行います。

ファイル操作関数

相手局のファイル/ディレクトリ情報の取得および設定、相手局上でのファイル操作を行うための関数です。

ファイルディレクトリ削除(FLKFileDelete 関数)

相手局側ファイルおよびディレクトリの削除を行います。
複数ファイル指定、ワイルドカード使用が可能です。
指定ファイルが存在しない場合でもエラーになりません。

ファイル移動(FLKFileMove 関数)

相手局側ファイルの同一ドライブ内での移動またはファイル名の変更を行います。
複数ファイル指定、ワイルドカード使用はできません。
移動先ディレクトリが存在しない場合は自動的に作成します。
移動元と移動先のドライブ名が異なる場合はエラーとなります。

ディレクトリ作成(FLKMakeDir 関数)

相手局側ディレクトリ作成を行います。
複数ファイル指定、ワイルドカード使用はできません。
タイムスタンプ、属性の設定が可能です。

ファイル情報の取得(FLKGetFileInfo 関数、FLKGetFileInfoEx 関数)

相手局のファイル情報(タイムスタンプ、サイズ、属性)の取得を行います。
ワイルドカード使用が可能です。

ファイル情報の設定(FLKSetFileInfo 関数、FLKSetFileInfoEx 関数)

相手局のファイル情報(タイムスタンプ、サイズ、属性)の更新を行います。

相手局環境情報取得/設定関数

相手局のシステム環境情報の取得および設定を行うための関数です。

日付時刻の取得/設定(FLKDateTime 関数)

相手局のシステム日付時刻の取得/設定を行います。

ディスク情報の取得(FLKGetDiskInfo 関数)

相手局側ディスク情報の取得を行います。
ディスク情報の項目は以下の通りです。

- ディスク総容量
- ディスク空き容量
- ディスク状態(フォーマット済み/未フォーマット/ディスク無し)

システム情報の取得(FLKGetSysInfo 関数)

相手局側のシステム情報の取得を行います。
システム情報の項目は以下の通りです。

- セッション ID(通信時のセッション番号)
- プロトコルバージョン(ファイル転送プロトコルのバージョン番号)
- 相手局機種コード(HT/PC(AT 互換機)/PC(98 シリーズ))
- OS モデル情報(HT モデル種別/PC の OS 種別)

なお、上記情報は回線オープン時のセッション確立直後に相手局より取得します。

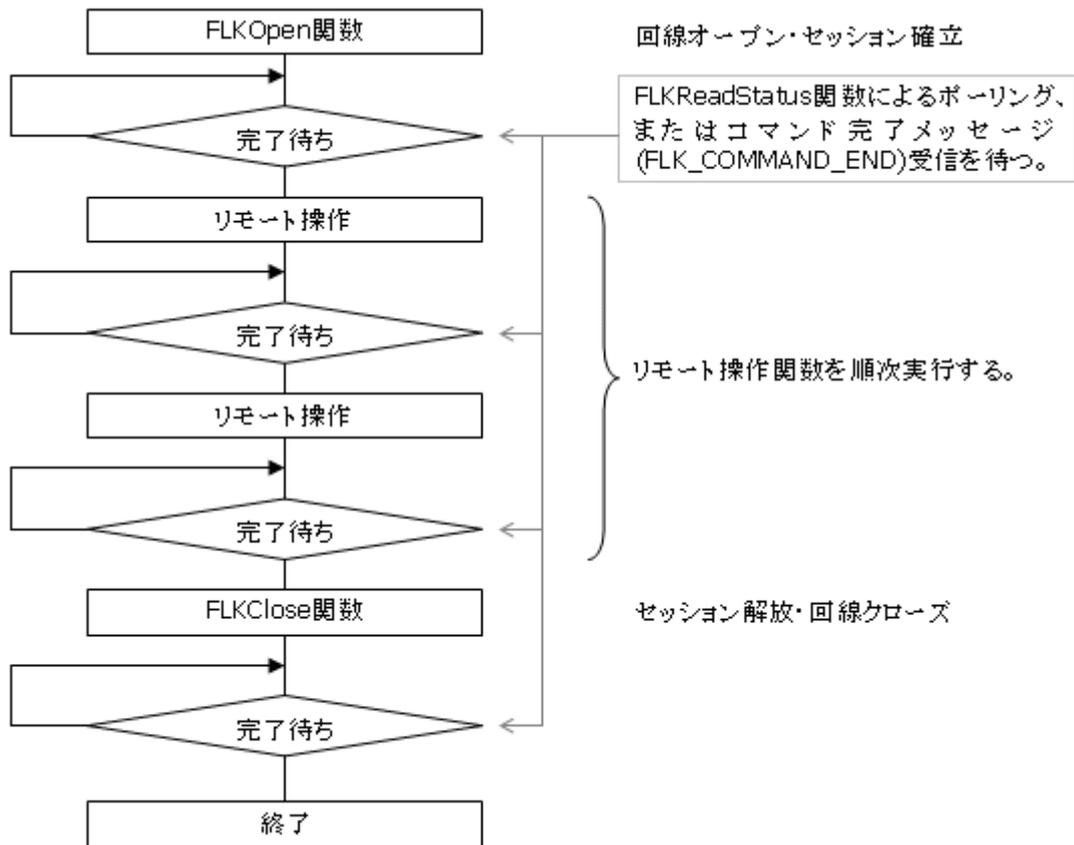
画面表示メッセージの送信(FLKMsgSend 関数)

相手局へ画面表示用のメッセージを送信します。

ブザー鳴動要求の送信(FLKBeep 関数)

相手局へブザー鳴動コマンドを送信します。

[リモート操作関数の基本フロー]



4. 通信状態の通知

通信状態の通知方法

本ライブラリの関数は、関数実行後すぐに呼び出し元に制御が戻ります。

実行した関数の機能が正常終了したかどうかは、ライブラリからのメッセージ通知、または、`FLKReadStatus` 関数により状態を読み出して判断します。

進捗表示

本ライブラリには、進捗状態を表示する機能はありません。

通信状態が変化した場合に、メッセージ通知するので、アプリケーション側で必要に応じて進捗情報を表示してください。

また、`FLKReadStatus` 関数を実行することで、現在の進捗状況をいつでも読み出すことができます。

メッセージの種類

以下のタイミングで、本ライブラリから指定した `Window` にメッセージを通知します。

メッセージ番号は、`FLKOpen` 関数のパラメータで指定します。

WPARAM の値	種類	備考
<code>FLK_COMMAND_END</code>	実行したコマンドが正常終了した。	次のコマンドを実行できます。
<code>FLK_COMMAND_ERROR</code>	実行したコマンドがエラー終了した。	クローズを実行してください。
<code>FLK_STATUS_CHANGE1</code>	通信状態が変化した。	<code>FLKReadStatus</code> 関数で状態を読み出すことで、現在実行中の処理 (Function、送受信ファイル名等)を知ることができます。
<code>FLK_STATUS_CHANGE2</code>	ファイル転送中に、ファイルの転送バイト数に変化があった。	<code>FLKReadStatus</code> 関数で状態を読み出すことで、転送中のファイルの進捗情報を知ることができます。
<code>FLK_INDICATION_END</code> (<code>FLKIdle</code> 関数実行時のみ)	相手局から終了指示コマンドを受信した。	クローズを実行してください。

5. パス名の記述方法

パス名の記述は、対象マシン OS のパス名命名規則に従ってください。

パスを表現する場合、ドライブレターについて以下に従った記述をしてください。

1. HT 上のパス名はドライブレターを用いず、ルートディレクトリから記述してください。
(ホスト PC 用のアップダウンユーティリティから HT 上のファイルまたはディレクトリのパスを指定する場合も同様です)
2. 通信相手から、ドライブレター付で指定されても、本ライブラリではドライブレターを無視します。
(ルートディレクトリからドライブレターなしで指定されたものと同等に扱われます)
3. HT からドライブレターの必要な OS の動作している通信相手(PC 等)上のファイルまたはディレクトリのパスを表現する場合は、ドライブレターをつけて記述してください。

ただし、上記 2. にかかわらず例外として、通信相手から本機のデバイスを指定して、フォーマット、ディスク情報取得を行う場合、ドライブレターに以下の意味をもたせます。ただし、これらの設定はレジストリに記述することで変更可能です。

デフォルト設定

	DT-5200	DT-X7	DT-9800	DT-5300	IT-300	DT-X8
C:	内部 RAM					
D:	CF カード*	USB メモリ	CF カード*	USB メモリ	USB メモリ	USB メモリ
E:	SD カード*	USB メモリ	SD カード*	SD カード*	SD カード*	SD カード*
F:	FlashDisk	FlashDisk	FlashDisk	FlashDisk	FlashDisk	FlashDisk

	IT-9000	IT-G500	DT-X100	DT-X200
C:	内部 RAM	内部 RAM	内部 RAM	内部 RAM
D:	USB メモリ	USB メモリ	USB メモリ	USB メモリ
E:	SD カード*	SD カード*	USB メモリ	SD カード*
F:	FlashDisk	FlashDisk	FlashDisk	FlashDisk

6. レジストリ

レジストリに値を書き込むことにより以下の設定を変更することができます。

レジストリは、オープン時に読み込みます。

一部のレジストリは、FLCE.EXE と同じ物を使用します。

設定項目

- セッション確立タイムアウト レジストリに設定がない場合 60 秒
- 受信待ちタイムアウト レジストリに設定がない場合 30 秒
- セッション終了待ちタイムアウト レジストリに設定がない場合 10 秒
- Windows98 通信モード レジストリに設定がない場合は 0
- ドライブレターの意味付け対応 レジストリに設定がなかった場合のデフォルト

	DT-5200	DT-X7	DT-9800	DT-5300	IT-300	DT-X8
C:	¥Windows	¥Windows	¥Windows	¥Windows	¥Windows	¥Windows
D:	¥CF カード [*]	¥USBStorage	¥メモリ カード [*]	¥USBStorage	¥USBStorage	¥USBStorage
E:	¥SD カード [*]	¥USBStorage	¥SD カード [*]	¥SD カード [*]	¥SD カード [*]	¥SD カード [*]
F:	¥FlashDisk	¥FlashDisk	¥FlashDisk	¥FlashDisk	¥FlashDisk	¥FlashDisk

	IT-9000	IT-G500	DT-X100	DT-X200
C:	¥Windows	¥Windows	¥Windows	¥Windows
D:	¥USBStorage	¥USBStorage	¥USBStorage	¥USBStorage
E:	¥SD カード [*]	¥SD カード [*]	¥USBStorage	¥SD カード [*]
F:	¥FlashDisk	¥FlashDisk	¥FlashDisk	¥FlashDisk

機種により、名称が全角、半角、スペースの有無があります。

機種名	ドライブ	名称	詳細説明
DT-5200	D	CF カード	文字は半角(スペース無し)
	E	SD カード	文字は半角(スペース無し)
DT-X7	D	USBStorage	文字は半角(スペース無し)
	E	USBStorage	文字は半角(スペース無し)
DT-9800	D	メモリ カード	文字は全角、リとカの間は半角スペース
	E	SD カード	SD は半角、カードは全角(スペース無し)
DT-5300	D	USBStorage	文字は半角(スペース無し)
	E	SD カード	文字は半角(スペース無し)
IT-300	D	USBStorage	文字は半角(スペース無し)
	E	SD カード	文字は半角(スペース無し)
DT-X8	D	USBStorage	文字は半角(スペース無し)
	E	SD カード	文字は半角(スペース無し)
IT-9000	D	USBStorage	文字は半角(スペース無し)
	E	SD カード	文字は半角(スペース無し)
IT-G500	D	USBStorage	文字は半角(スペース無し)
	E	SD カード	文字は半角(スペース無し)
DT-X100	D	USBStorage	文字は半角(スペース無し)
	E	USBStorage	文字は半角(スペース無し)
DT-X200	D	USBStorage	文字は半角(スペース無し)
	E	SD カード	文字は半角(スペース無し)

• レジストリ位置

[HKEY_CURRENT_USER¥Software¥CASIO¥FLCE]

キー名	型	値
CONWAIT	DWORD	セッション確立タイムアウト時間
RECVWAIT	DWORD	受信待ちタイムアウト時間
DISCONWAIT	DWORD	セッション終了タイムアウト時間
WIN98MODE	DWORD	0(1を設定すると Windows98 通信モード)
DRIVE¥A	STRING	Aドライブとして意味付けするデバイスのパス
DRIVE¥B	STRING	Bドライブとして意味付けするデバイスのパス
DRIVE¥C	STRING	Cドライブとして意味付けするデバイスのパス
DRIVE¥Z	STRING	Zドライブとして意味付けするデバイスのパス

カシオ計算機お問い合わせ窓口

製品に関する最新情報

- 製品サポートサイト（カシオペア・ハンディターミナル）

<https://casio.jp/support/ht/>

カシオ計算機株式会社

〒151-8543 東京都渋谷区本町 1-6-2

TEL 03-5334-4638(代)