

FLINK Library Manual

(Version 2.03)

CASIO Computer Co., Ltd.

Copyright ©2011. All rights reserved.

January 2011

Table of the Contents

	Editorial Record	4
Chapter 1.	Overview	5
1.1	Operating Environment	5
1.2	Development Environment	5
1.3	Supplied Files	5
1.4	Steps to Start Up	5
1.5	List of the Structures	6
1.5.1	FLK_RSPRM	7
1.5.2	FLK_STATUS	8
1.5.3	FLK_ERRINFO	10
1.5.4	FLK_DATETIME	13
1.5.5	FLK_FINFO	14
1.5.6	FLK_DINFO	15
1.5.7	FLK_SYSINFO	16
1.5.8	FLK_FINFOEX	17
1.6	Functions List	18
1.6.1	FLKOpen	19
1.6.2	FLKFileSend, FLKSendFile	21
1.6.3	FLKFileAdd, FLKAddFile	23
1.6.4	FLKFileRecv, FLKReceiveFile	25
1.6.5	FLKClose	27
1.6.6	FLKReadStatus	29
1.6.7	FLKReadErrStat, FLKReadErrorStatus	30
1.6.8	FLKIdle	31
1.6.9	FLKCmdRecv, FLKReceiveCommand	33
1.6.10	FLKFileDelete, FLKDeleteFile	35
1.6.11	FLKFileMove, FLKMoveFile	36
1.6.12	FLKMakeDir, FLKMakeDirectroy	38
1.6.13	FLKGetFileInfo	40
1.6.14	FLKSetFileInfo	42
1.6.15	FLKGetFileInfoEx	43
1.6.16	FLKSetFileInfoEx	45
1.6.17	FLKGetDiskInfo	46
1.6.18	FLKDateTime	48
1.6.19	FLKGetSysInfo, FLKGetSystemInfo	50
1.6.20	FLKMsgSend, FLKSendMessage	51
1.6.21	FLKBeep	52
Chapter 2.	Communication Specifications	53
2.1	Interfaces	53
2.2	Configurations	54
2.3	Communication Parameters	55
2.4	Operation Modes	56
2.5	Command-Transmission Right	57
2.6	Overview of the Process	58
Chapter 3.	File Transmission/Reception Basic Functions	60
3.1	Communication Basic Functions	60

3.2	File Transmission/Reception Functions	64
3.3	Remote Operation Functions	68
3.4	Retrieve/Set Up Functions for Partner Information	69
Chapter 4.	Notifying Communication Status	70
Chapter 5.	Pathnames	71
Chapter 6.	Registry	72

No part of this document may be produced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of CASIO Computer Co., Ltd. in Tokyo Japan. Information in this document is subject to change without advance notice. CASIO Computer Co., Ltd. makes no representations or warranties with respect to the contents or use of this manual and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose.

© 2011 CASIO Computer Co., Ltd. All rights reserved.

Editorial Record

Manual Version no.	Date edited	Page	Content
0.90	November 2005		Tentative version
1.00	February 2006		Original version
1.01	March 2006	5	"DT-X11" is added in Chapter 1.1 "Operating Environment".
		42	"DT-X11" is added in Table 1.6 of Chapter 1.7.1 "Interfaces".
		60	"DT-X11" is added in Table 1.15 of Chapter 1.14 "Pathnames".
1.02	August 2007	5	"DT-X7" is added in Chapter 1.1 "Operating Environment".
		43	"DT-X7" is added in Table 1.6 of Chapter 1.7.1 "Interfaces".
		61	"DT-X7" is added in Table 1.15 of Chapter 1.14 "Pathnames".
1.03	December 2007	5	In Chapter 2.1, "Visual Studio .NET 2003" is deleted and "Visual Studio 2005" is added instead.
		18	"FLKDataSend" and "FLKDataRecvEnd" functions are deleted in Table 1.5 in Chapter 1.6.
		19	Return values of FLKOpen function are corrected.
		50	The explanation about "Command-Transmission Right" in HT mode is corrected.
1.04	July 2008	5	"DT-X30" and "IT-3100" are added in Chapter 1.1.
		5	"Microsoft WindowsCE Ver. 6.0" is added in Chapter 1.1.
		5	"Visual Studio 2008" is added in Chapter 2.1.
		19 to 45	Note for return value in the Device Emulator is added.
		46	"DT-X30" and "IT-3100" are added in Table 2.1 of Chapter 2.1.
		64	"DT-X30" and "IT-3100" are added in Table 5.1 of Chapter 5.
2.00	October 2008	all	Version of the integration of FLINK Library Manual and FLINK Class Library Manual
		5	In Chapter 1.1, IT-10, IT-500, IT-3000, DT-X5 and DT-X10 are deleted.
		5	In Chapter 1.1, "WindowsCE .NET Ver. 4.1" is deleted.
		55	In Chapter 2.1, IT-10, IT-500, IT-3000, DT-X5 and DT-X10 are deleted.
		73	In Chapter 5, IT-10, IT-500, IT-3000, DT-X5 and DT-X10 are deleted in Table 5.1.
2.01	March 2009	7	In Chapter 1.5.1, the comments on calling sequence are updated.
		30	In Chapter 1.6.7, the description about the return values is corrected.
		5	In Chapter 1.1, "Microsoft® Windows Mobile 6.1" is added to "OS".
2.02	August 2009	all	"IT-800" is added.
2.03	January 2011	all	"IT-300" and "DT-X8" are added.

1. Overview

The purpose of the **FLINK Library** is to make the **FLCE** functions of **FLINK** application for WindowsCE available in user's application program.

The **FLINK Class Library - MoFlinkLib.dll** - is a wrapper library layer. The library can be directly manipulated by .NET Compact Framework application.

1.1 Operating Environment

Applicable Handheld Terminals

- IT-600
- IT-3100
- DT-X7
- DT-X11
- DT-X30
- IT-800
- IT-300
- DT-X8

OS

- Microsoft® WindowsCE 5.0
- Microsoft® WindowsCE 6.0
- Microsoft® WindowsMobile 6.1
- Microsoft® WindowsMobile 6.5
- Microsoft® WindowsMobile 6.5.3

1.2 Development Environment

- Microsoft® embedded C++ Version 4.0 + SP4
- Microsoft® Visual Studio 2005 + SP1
- Microsoft® Visual Studio 2008 + SP1

1.3 Supplied Files

- flinklib.lib : Import library
- flinklib.h : Header file
- MoFlinkLib.dll : Class library for C# .NET and VB .NET

1.4 Steps to Start Up

In the development environment, include the header file, "**flinkfib.h**", in the source program and then specify "**flinkfib.lib**" as the import library to use.

Copy **MoFlinkLib.dll** file in the same folder where the execution module resides.

1.5 List of the Structures

Table 1.1

Structure	Description
FLK_RSPRM	Stores information about the Serial port.
FLK_STATUS	Stores the statuses of FLINK Library.
FLK_ERRINFO	Stores information about errors.
FLK_DATETIME	Stores date and time.
FLK_FINFO	Stores information about disk.
FLK_FINFOEX	Stores file information.
FLK_DINFO	Stores information about disk.
FLK_SYSINFO	Stores information about the system.

1.5.1 FLK_RSPRM

This structure stores information about the Serial port.

Calling Sequence

```
Typedef struct{
    WORD  speed;           // Baud rate  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;          // Data length  FLK_CHAR8
    WORD  parity;           // Parity      FLK_PARI_NON (Non parity)
                           // Parity      FLK_PARI_ODD (Odd parity)
                           // Parity      FLK_PARI_EVN (Even parity)
    WORD  stop_bit         // Stop bit    FLK_STOP1 (1 bit)
                           // Stop bit    FLK_STOP2 (2 bits)
}FLK_RSPRM;
```

The function that uses the structure is as follows.

- **FLKOpen**

1.5.2 FLK_STATUS

This structure stores the status of the **FLINK Library**.

Calling Sequence

```
Typedef struct{
    WORD      status;      // Command execution status
    WORD      function;    // Command type
    TCHAR     *FileName;   // Name of transmission/receive
    DWORD     total_size;  // Number of total bytes
                          // for one-time transmission
    DWORD     total_count; // Number of total bytes sent already
                          // for one-time transmission
    DWORD     file_size;   // File size being transmitted
    DWORD     file_count;  // Number of bytes sent already
                          // for the file being transmitted
    Char      *Data;       // Data
    DWORD     data_size;   // Size of data being received
    DWORD     data_count;  // Number of transmitted bytes
                          // for data being received
}FLK_STATUS;
```

status

This parameter specifies one of the values listed in the table indicating either the command carried out lastly is still in the progress or it is complete.

Table 1.2

status	Description
FLK_STATUS_RUNNING	In progress.
FLK_STATUS_END	Normal end.
FLK_STATUS_ERROR	Aborted due to error.
FLK_STATUS_INDICATION_END	Receive termination order command from the partner station.

function, filename

In the *function* and *filename* parameters, the followings will be set. If "NULL" has been specified in the member of "FileName" when calling the function, none will be set in the *filename* parameter.

Table 1.3

<i>function</i>	<i>filename</i>	Description
FLK_SND_EXEC	Name of file to be transmitted	The file is now being transmitted.
FLK_RCV_EXEC	Name of file to be received	The file is now being received.
FLK_ADDSND_EXEC	Name of file to be transmitted.	The additional file is now being transmitted.
FLK_ADDRCV_EXEC	Name of original file to be transmitted	The additional file is now being received.
FLK_DELETE_EXEC	Name of file to be deleted	Request is made to delete the file at the partner station.
FLK_DELETE_RECV	Name of file to be deleted	The file at own station is now being deleted.
FLK_MOVE_EXEC	Name of source file to be moved	Request to move the file at the partner station is being made.
FLK_MOVE_RECV	Name of source file to be moved	The source file is now being moved.
FLK_MAKEDIR_RECV	Name of directory to be created	The directory is now being created.
FLK_DATETIME_EXEC	----	Request is made to change date/time at the partner station.
FLK_DATETIME_RECV	----	Request to change date/time is received.
FLK_DATETIME_SEND	----	Date/time of the partner station is transmitted.
FLK_MESSAGE_RECV	Message requested to display	Request to display message is received.
FLK_BUZZER_RECV	----	Request to sound the buzzer is received.
FLK_GETFILEINFO_RECV	Name of file in the progress	Request to retrieve file information is received.
FLK_SETFILEINFO_RECV	Name of file in the progress	Request to set file information is received.
FLK_GETDISKINFO_RECV	Name of drive	Disk information is now being received.
FLK_DATA_RECV	Data	Request to transmit data is received.

The function that uses the structure is as follows.

- **FLKReadStatus**

1.5.3 FLK_ERRINFO

This structure stores information about errors.

Calling Sequence

```
Typedef struct{
    BYTE      kind;      // Error type (See Figure 1.1)
    BYTE      command;   // Command type (See Table 1.4)
    BYTE      category;  // Category (See Table 1.5)
    BYTE      detail;    // Error detail (See Table 1.5)
    DWORD     biosStat;  // Error code retrieved with
                        // GetLastError function
}FLK_ERRINFO;
```

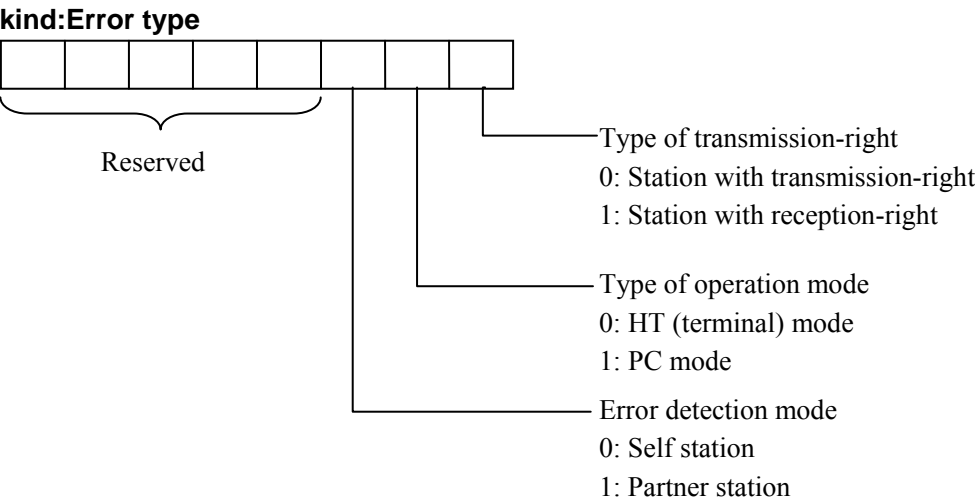


Figure 1.1

Table 1.4 Command types

Command type		
Value	Command	Description
00h	FLK_CMD_NON	No applicable command
01h	FLK_CMD_FSEND_TINFO	File transmission information command
02h	FLK_CMD_FSEND_FINFO	File information command
03h	FLK_CMD_FRECV_TREQ	File reception request command
04h	FLK_CMD_FADD	File append command
05h	FLK_CMD_FDATA	File data command
06h	FLK_CMD_FDEL	File delete command
07h	FLK_CMD_FMOV	File move command
08h	FLK_CMD_MAKEDIR	Directory create command
09h	FLK_CMD_TIME_SET	Date/time setting command
0Ah	FLK_CMD_TIME_GET	Date/time retrieve command
0Bh	FLK_CMD_DISP	Message display command
0Ch	FLK_CMD_BEEP	Buzzer ON command
0Dh	FLK_CMD_FINFO_GET	File information retrieve command
0Eh	FLK_CMD_FINFO_SET	File information setting command
0Fh	FLK_CMD_DINFO_GET	Disk information retrieve command

Continue.

10h	FLK_CMD_SYS_GET	System information retrieve command
11h	FLK_CMD_IDLE	IDLE notification command
12h	FLK_CMD_END	Termination request command
13h	FLK_CMD_DATA	Data transmission command

Table 1.5 Category and Error detail

Value		Description	
Category	Detail		
When terminate normally.			
00h	00h	Normal end	
DCh to F5h	00h	Format request command (A to Z)	
F6h	00h	Power OFF complete notification	
F7h	00h	Reset designation termination notification	
F8h	00h	Termination with BREAK key notification	
F9h to FFh	--	Reserved area	
Protocol error			
01h	00h	Reception frame function code undefined error	
	01h	Reception frame sub-function code undefined error	
	03h	Reception frame checksum error	
	04h	Sequence error	
	05h	Sequence number error	
	07h	Reception frame internal information parameter error	
	08h	Reception timeout	
	10h	Command length error	
File error (Protocol logic)			
04h	00h	Read-only file access error	
Utility error			
10h	00h	Line open error	<ul style="list-style-type: none">• The line has not been opened.• Check if any error at the time of opening the line has occurred.
	01h	Function phase error	<ul style="list-style-type: none">• The way to use the function is wrong.• Check the operation mode and the transmission-right station mode.
	02h	Function parameter error	<ul style="list-style-type: none">• The function parameter has been irregularly set.• Check the specified parameter.
	03h	Specified file not-found error	<ul style="list-style-type: none">• The specified file does not exist.• Check the specified file.
	04h	Partner station not-found	<ul style="list-style-type: none">• Session establishment wait timeout• Check the comm. settings and the line.
	05h	System date setting error	<ul style="list-style-type: none">• Check the specified date.
	06h	System time setting error	<ul style="list-style-type: none">• Check the specified time.
	09h	Fatal error	
	0Ah	Line break off error	<ul style="list-style-type: none">• The line broke off during the communication. Check the line.
	0Bh	Drive capacity not sufficient	<ul style="list-style-type: none">• The specified drive does not have enough free space.
	0Ch	Command carry out error	<ul style="list-style-type: none">• Other command was invoked while the one is executing.

Continue.

File error (File function)		
11h	00h	Create error
	01h	Open error
	02h	Read error
	03h	Write error
	04h	Seek error
	05h	File delete error
	06h	Directory delete error
	07h	File name change/move error
	08h	Timestamp setting error
	09h	Timestamp retrieve error
	0Ah	File attribute setting error
	0Bh	File attribute retrieve error
	0Ch	Directory create error
	0Dh	File size change error
System error		
0Fh	0xh	FTP internal error
	1xh	Communication utility internal error

The functions that use the structure are as follows.

- **FLKReadErrStat**
- **FLKReadErrorStatus**

1.5.4 FLK_DATETIME

This structure stores date and time.

Calling Sequence

```
Typedef struct{
    BYTE    day;        // Day (1 to 31)
    BYTE    month;      // Month (1 to 12)
    WORD    year;       // Year (1980 to 2079)
    BYTE    sec;        // Second (0 to 59)
    BYTE    min;        // Minute (0 to 59)
    BYTE    hour;       // Hour (0 to 23)
}FLK_DATETIME;
```

The functions that use the structure are as follows.

- **FLKMakeDir**
- **FLKMakeDirectroy**
- **FLKGetFileInfoEx**
- **FLKSetFileInfoEx**
- **FLKDateTime**

1.5.5 FLK_FINFO

This structure stores information about files.

Calling Sequence

```
Typedef struct{
    TCHAR        name[256] // Searched file name (in full path)
    FLK_DATETIME  datetime; // Date/time storage area
    DWORD        size      // Size
    BYTE         atr;       // Attribute
                                // FLK_A_NORMAL: Ordinary file
                                // FLK_A_HIDDEN: Hidden file
                                // FLK_A_RDONLY: Read-only file
                                // FLK_A_SYSTEM: System file
                                // FLK_A_ARCH  : Archive
}FLK_FINFO;
```

The functions that use the structure are as follows.

- FLKGetFileInfo
- FLKSetFileInfo

1.5.6 FLK_DINFO

This structure stores information about disk.

Calling Sequence

```
Typedef struct{
    DWORD      size;    // Disk capacity
    DWORD      freex;   // Disk capacity free
    BYTE       status   // Disk status
                    // FLK_DINFO_NORMAL   : Disk available (formatted)
                    // FLK_DINFO_NOFMT    : Disk available (not formatted)
                    // FLK_DINFO_NODISK   : Disk not available
}FLK_DINFO;
```

The function that uses the structure is as follows.

- **FLKGetDiskInfo**

1.5.7 FLK_SYSINFO

This structure stores information about the system.

Calling Sequence

```
typedef struct{  
    WORD    id;           // Session ID (set "0" when connecting to PC)  
    BYTE    ftpver;       // Protocol version  
    BYTE    code[3];      // Model code  
    BYTE    model;        // Model information  
}FLK_SYSINFO;
```

The functions that use the structure are as follows.

- **FLKGetSysInfo**
- **FLKGetSystemInfo**

1.5.8 FLK_FINFOEX

This structure stores file information.

Calling Sequences

```
typedef struct{
    TCHAR      name[256]; // File name searched (in full path)
    BYTE       day;       // Date (1 to 31)
    BYTE       month;     // Month (1 to 12)
    WORD       year;      // Year (1980 to 2079)
    BYTE       sec;       // Second (0 to 59)
    BYTE       min;       // Minute (0 to 59)
    BYTE       hour;      // Hour (0 to 23)
    DWORD      size;      // Size
    BYTE       atr;       // Attribute
                        // FLK_A_NORMAL   : Ordinary file
                        // FLK_A_HIDDEN   : Hidden file
                        // FLK_A_RDONLY  : Read-only file
                        // FLK_A_SYSTEM  : System file
                        // FLK_A_ARCH    : Archive
}FLK_SYSINFO;
```

The functions that use the structure are as follows.

- **FLKGetFileInfoEx**
- **FLKSetFileInfoEx**

1.6 Functions List

Table 1.6

Function	Description
FLKOpen	Opens the preset communication port.
FLKFileSend FLKSendFile	Transmits a file.
FLKFileAdd FLKAddFile	Transmits a file appended.
FLKFileRecv FLKReceiveFile	Receives a file.
FLKClose	Closes the communication port.
FLKReadStatus	Retrieves the progress information.
FLKReadErrStat FLKReadErrorStatus	Retrieves error information.
FLKIdle	Invokes the idle condition.
FLKCmdRecv FLKReceiveCommand	Waits to receive command.
FLKFileDelete FLKDeleteFile	Deletes a file.
FLKFileMove FLKMoveFile	Moves a file or changes its file name.
FLKMakeDir FLKMakeDirectroy	Creates a directory.
FLKGetFileInfo	Retrieves file information.
FLKSetFileInfo	Sets up file information.
FLKGetFileInfoEx	Retrieves file information.
FLKSetFileInfoEx	Sets up file information.
FLKGetDiskInfo	Retrieves disk information.
FLKDateTime	Retrieves date and time and sets.
FLKGetSysInfo FLKGetSystemInfo	Retrieves the system information.
FLKMsgSend FLKSendMessage	Transmits screen messages.
FLKBeep	Activates the buzzer sound.

Note:

For any operation that repeats opening and then closing line, be sure to wait for approximately five seconds to proceed the line open process again after closing. Otherwise, it may fail.

1.6.1 FLKOpen

This function opens the communication port, initializes, establishes session, and retrieves system information about partner station.

Calling Sequences

```
[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 rsPrm,
    ushort mode,
    IntPtr hWnd,
    uint message
)
```

Parameters

device

This parameter specifies one of the values listed as device to use.

"COM1"	: Serial (COM1)
"IRDA"	: IrDA
"LAN"	: LAN
"USB"	: USB

cString

This parameter sets up the communication parameter.

"COM1" : Ignore.
"IRDA" : Ignore.
"LAN" : Specify IP address of the Host PC.
"USB" : Ignore.

rsPrm

This parameter sets up the communication parameter.

FLK_DEVICE_COM1 : Specify the communication parameters of COM port.
FLK_DEVICE_IRDA : Ignore.
FLK_DEVICE_LAN : Ignore.
USB : Ignore.

mode

This parameter specifies station mode.

FLK_MODE_HT : HT (terminal) mode
FLK_MODE_PC : PC mode

hWnd

This parameter specifies window handle of the window that notifies a message. Specifying "NULL" in this parameter will not notify a message.

message

This parameter specifies a message number to be notified to the application.

Return Values

Handle value : Normal end
FLK_PRM : Parameter error
FLK_NG : Abnormal end caused by other errors excluding the parameter error. In the Device Emulator, this value is always returned.
Other return values than the above : Handle of FLINK Library (Normal end)

Messages

Table 1.7

WPARAM	Description
FLK_COMMAND_END	Terminate the opening process.
FLK_COMMAND_ERROR	Terminate the opening process abnormally.

Note:

If the connection is established in USB mode, navigate to **Control Panel → Connection to PC → Connect cable to PC** on the terminal side and then remove check in the checkbox of **Connect cable to PC**. For operations on the PC side, refer to the LMWIN manual available separately.

1.6.2 FLKFileSend, FLKSendFile

These functions transmit specified multiple files at the same time. If a directory at the destination side does not exist, the function will create a directory automatically.

Calling Sequences

```
[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
)
```

Parameters

hPort

This parameter specifies the handle retrieved at a time of opening the communication port.

mode

This parameter specifies transmission mode by selecting either one of the values below.

FLK_TRANS_NORMAL	: Normal transmission
FLK_TRANS_RECURSIVE	: Recursive transmission

fName

This parameter specifies source file name in full path. "Wild card" is possible.

dir

This parameter specifies destination folder name in full path.

protect

This parameter specifies forcibly overwrite flag. If a file with the same file name and with "READ-ONLY" attribute exists at the destination side, specify either "Enable" to allow writing file over the existing file or "Disable" it.

FLK_PROTECT_VALID : Disable forcibly overwrite a file.
FLK_PROTECT_INVALID : Enable forcibly overwrite a file.

Return Values

FLK_OK : Normal end
FLK_PRM : Parameter error
FLK_NG : Abnormal termination caused by other errors excluding the parameter error. In the Device Emulator, this value is always returned.

Messages

Table 1.8

WPARAM	Description
FLK_COMMAND_END	All specified files are transmitted.
FLK_COMMAND_ERROR	File transmission is terminated due to error.
FLK_STATUS_CHANGE1	Transmission of one file begins. The name of the file can be retrieved with FLKReadStatus function.
FLK_STATUS_CHANGE2	The number of bytes of file transmitted is changed. The number of the bytes changed can be retrieved with FLKReadStatus function.

1.6.3 FLKFileAdd, FLKAddFile

These functions append a specified file to the destination file existed at the partner station. Specifying multiple file names and in wild card is not possible for both transmission and destination sides. If the file name at the destination side does not exist, a new file will be created.

Calling Sequences

```
[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
)
```

Parameters

hPort

This parameter specifies the handle retrieved at a time of opening the communication port.

sfName

This parameter specifies source file name in full path.

rfName

This parameter specifies destination file name in full path.

Return Values

FLK_OK	: Normal end
FLK_PRN	: Parameter error
FLK_NG	: Abnormal termination caused by other errors excluding the parameter error. In the Device Emulator, this value is always returned.

Messages

Table 1.9

WPARAM	Description
FLK_COMMAND_END	All specified files are transmitted.
FLK_COMMAND_ERROR	File transmission is terminated due to error.
FLK_STATUS_CHANGE1	Transmission of one file begins. The name of the file can be retrieved with FLKReadStatus function.
FLK_STATUS_CHANGE2	The number of bytes of file transmitted has been changed. It can be retrieved with FLKReadStatus function.

1.6.4 FLKFileRecv, FLKReceiveFile

These functions receive specified multiple files at the same time. If a directory at the destination side does not exist, the functions will create a directory automatically.

Calling Sequences

```
[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
)
```

Parameters

hPort

This parameter specifies the handle retrieved at a time of opening the communication port.

mode

This parameter specifies transmission mode, normal or recursive mode, selecting either one of the values listed below.

FLK_TRANS_NORMAL	: Normal transmission
FLK_TRANS_RECURSIVE	: Recursive transmission

fName

This parameter specifies destination file name in full path. "Wild card" is possible.

dir

This parameter specifies destination folder name in full path.

protect

This parameter specifies forcibly overwrite flag. If a file with the same file name and with "READ-ONLY" attribute exists at the destination, specify either "Enable" to allow writing file over the file or "Disable" it.

FLK_PROTECT_VALID : Disable forcibly overwrite a file.
FLK_PROTECT_INVALID : Enable forcibly overwrite a file.

Return Values

FLK_OK : Normal end
FLK_PRM : Parameter error
FLK_NG : Abnormal end caused by other errors excluding the parameter error. In the Device Emulator, this value is always returned.

Messages

Table 1.10

WPARAM	Description
FLK_COMMAND_END	All specified files have been transmitted.
FLK_COMMAND_ERROR	File transmission is terminated due to error.
FLK_STATUS_CHANGE1	Transmission of one file begins. The name of the file can be retrieved with FLKReadStatus function.
FLK_STATUS_CHANGE2	The number of bytes of file transmitted has been changed. It can be retrieved with FLKReadStatus function.

1.6.5 FLKClose

This function closes the communication port. By transmitting the termination command to the partner station, the function releases the communication session. In transmission-right mode, it transmits the exit command to the partner station unless otherwise an error has occurred already.

Calling Sequences

```
[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
)
```

Parameters

hPort

This parameter specifies the handle retrieved at a time of opening the communication port.

endKind

This parameter specifies exit command to the partner station which is effect only in transmission-right mode.

FLK_CLOSE_NORMAL : Normal end.

The above parameter is only notified to the partner station. After **FLKIdle** function or **FLKCmdRecv** function has been carried out normally, a code has been specified from the partner station can be retrieved with **FLKReadErrStat** function.

Return Values

FLK_OK	: Normal end
FLK_PRN	: Parameter error
FLK_NG	: Abnormal end caused by other errors excluding the parameter error. In the Device Emulator, this value is always returned.

Messages

Table 1.11

WPARAM	Description
FLK_COMMAND_END	Terminate the closing process.
FLK_COMMAND_ERROR	Terminate the closing process abnormally.

1.6.6 FLKReadStatus

This function retrieves the communication status in the **FLINK Library**.

Calling Sequences

```
[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
)
```

Parameters

hPort

This parameter specifies the handle retrieved at a time of opening the communication port.

flkstatus

This parameter specifies communication status storage area. See Chapter 1.5.2 "FLK_STATUS".

Return Values

FLK_OK	: Normal end
FLK_PRM	: Parameter error
FLK_NG	: Abnormal end caused by other errors excluding the parameter error. In the Device Emulator, this value is always returned.

Message

None

1.6.7 FLKReadErrStat, FLKReadErrorStatus

These functions retrieve the error information. When the functions receive the exit command from the partner station, they retrieve also the category and error detailed codes. After the retrieve, the error information will be cleared.

Calling Sequences

```
[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
)
```

Parameters

hPort

This parameter specifies the handle retrieved at a time of opening the communication port.

errinfo

This parameter stores error information. See Chapter 1.5.3 "FLK_ERRINFO".

Return Values

FLK_OK	: Normal end. In the Device Emulator, this value is always returned.
FLK_PRN	: Parameter error

Message

None

1.6.8 FLKIdle

This function invokes the idle condition to wait for commands sent from the partner station after transmitting the idle notification. This function is available only in HT mode. After the commands are received, they will be carried out sequentially until either the exit command is received or when an error occurs.

Calling Sequences

```
[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
)
```

Parameters

hPort

This parameter specifies the handle retrieved at a time of opening the communication port.

script

This parameter specifies script file name (File name only. The file name including the extension must be limited to 13 characters or less.). Specify "NULL" if the script file name has not been set.

Return Values

FLK_OK	: Normal end
FLK_PRN	: Parameter error
FLK_NG	: Abnormal end caused by other errors excluding the parameter error. In the Device Emulator, this value is always returned.

Messages

Table 1.12

WPARAM	Description
FLK_COMMAND_END	All specified files have been transmitted.
FLK_COMMAND_ERROR	File transmission is terminated due to error.
FLK_STATUS_CHANGE1	Transmission of one file begins. The name of the file can be retrieved with FLKReadStatus function.
FLK_STATUS_CHANGE2	The number of bytes of file transmitted has been changed. It can be retrieved with FLKReadStatus function.
FLK_INDICATION_END	Termination order command from the partner station has been received.

1.6.9 FLKCmdRecv, FLKReceiveCommand

These functions invoke the wait condition to receive commands sent from HT (terminal). The functions are available only in PC mode. After the commands are received, they will be carried out sequentially until either the exit command is received or when an error occurs.

Calling Sequences

```
[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
)
```

Parameters

hPort

This parameter specifies the handle retrieved at a time of opening the communication port.

endKind

This parameter specifies exit command to the partner station which is effect only in transmission-right mode.

FLK_CLOSE_NORMAL : Normal end

The above parameter is only notified to the partner station. After **FLKIdle** function or **FLKCmdRecv** function has been carried out normally, a code has been specified from the partner station can be retrieved with **FLKReadErrStat** function.

script

This parameter specifies script file name (File name only. The file name including the extension must be limited to 13 characters or less.). If script is specified by the partner station, the script name is set in the area specified by "script". Specify "NULL" if the script file name has not been set.

Return Values

FLK_OK : Normal end
FLK_PRM : Parameter error
FLK_NG : Abnormal end caused by other errors excluding the parameter error. In the Device Emulator, this value is always returned.

Messages

Table 1.13

WPARAM	Description
FLK_COMMAND_END	All specified files have been transmitted.
FLK_COMMAND_ERROR	File transmission is terminated due to error.
FLK_STATUS_CHANGE1	Transmission of one file begins. The name of the file can be retrieved with FLKReadStatus function.
FLK_STATUS_CHANGE2	The number of bytes of file transmitted has been changed. It can be retrieved with FLKReadStatus function.

1.6.10 FLKFileDelete, FLKDeleteFile

These functions delete a file except "Read-Only" file.

Calling Sequences

```
[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
)
```

Parameters

hPort

This parameter specifies the handle retrieved at a time of opening the communication port.

fName

This parameter specifies the name of a file in full path that is deleted. "Wild card" is possible.

Return Values

FLK_OK	: Normal end
FLK_PRM	: Parameter error
FLK_NG	: Abnormal end caused by other errors excluding the parameter error. In the Device Emulator, this value is always returned.

Messages

Table 1.14

WPARAM	Description
FLK_COMMAND_END	Terminate deleting file successfully.
FLK_COMMAND_ERROR	Terminate deleting file abnormally.

1.6.11 FLKFileMove, FLKMoveFile

These functions move a file, not to other drive, and change the file name if the same folder is specified.

Calling Sequences

```
[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
)
```

Parameters

hPort

This parameter specifies the handle retrieved at a time of opening the communication port.

sfName

This parameter specifies original file name in full path.

dfName

This parameter specifies destination file name in full path.

Return Values

FLK_OK	: Normal end
FLK_PRN	: Parameter error
FLK_NG	: Abnormal end caused by other errors excluding the parameter error. In the Device Emulator, this value is always returned.

Messages

Table 1.15

WPARAM	Description
FLK_COMMAND_END	Terminate deleting file successfully.
FLK_COMMAND_ERROR	Terminate deleting file abnormally.

1.6.12 FLKMakeDir, FLKMakeDirectroy

While the HT mode has been set effect, these functions create a directory, not in CF and SD cards.

Calling Sequences

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

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

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

Parameters

hPort

This parameter specifies the handle retrieved at a time of opening the communication port.

mDir

This parameter specifies created directory name.

datetime

This parameter specifies storage area that stores date.

atr

This parameter specifies attribute. To specify multiple attributes, specify a sum of the values in logical OR for attributes selected in the list below.

FLK_A_NORMAL : Ordinary file
FLK_A_HIDDEN : Hidden file
FLK_A_RDONLY : Read-only file
FLK_A_SYSTEM : System file
FLK_A_SUBDIR : Directory
FLK_A_ARCH : Archive

Notes:

- Since FLK_A_NORMAL in the ***atr*** parameter has been set "0", it is not necessary to set any other value. For FLK_A_SUBDIR, the parameter is ORed automatically in the library.
- If you do not set up date and time, set 0xFFF to "year" of the datetime structure member and 0xFF to "day, month, sec, min, hour" respectively.

Return Values

FLK_OK : Normal end
FLK_PRN : Parameter error
FLK_NG : Abnormal end caused by other errors excluding the parameter error. In the Device Emulator, this value is always returned.

Messages

Table 1.16

WPARAM	Description
FLK_COMMAND_END	Terminate creating directory successfully.
FLK_COMMAND_ERROR	Terminate creating directory abnormally.

1.6.13 FLKGetFileInfo

This function retrieves information about file size, time stamp, attribute for a specified file at the partner station. The file information is set up in the file information area when the searched file with the same file is found at the partner station. If "wild card" has been specified, specify "Retrieve first information" for first time, and "Retrieve subsequent information" for 2nd time. Call this function continuously if "wild card" has been specified. However, if other communication function is executed, subsequent information cannot be retrieved.

Calling Sequence

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

Parameters

hPort

This parameter specifies the handle retrieved at a time of opening the communication port.

mode

This parameter specifies "First" or "Subsequent" flag selecting either one of the values listed below.

- | | |
|---------------|--|
| FLK_GET_FIRST | : Retrieve first information (for the first time if "one file" or "wild card" is specified.) |
| FLK_GET_NEXT | : Retrieve subsequent information (for second forth if "wild card" is specified.) |

fName

This parameter specifies area for search file names. "Wild card" is possible. Specification of multiple files is not possible. The parameter cannot be referred to if "Retrieve subsequent information" is set.

fInfo

This parameter specifies area for file information where searched file is set. "0x00" is set in each parameter if searched file is not found.

Return Values

- | | |
|---------|---|
| FLK_OK | : Normal end |
| FLK_PRN | : Parameter error |
| FLK_NG | : Abnormal end caused by other errors excluding the parameter error. In the device Emulator, this value is always returned. |

Messages

Table 1.17

WPARAM	Description
FLK_COMMAND_END	Terminate successfully.
FLK_COMMAND_ERROR	Terminate abnormally.

1.6.14 FLKSetFileInfo

While the HT mode has been set effect only, this function updates information such as file size, time stamp, attribute for a specified file at the partner station by making the content in the file information area same with the content in the file name area.

Calling Sequence

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

Parameters

hPort

This parameter specifies the handle retrieved at a time of opening the communication port.

fInfo

This parameter specifies file information area.

Return Values

FLK_OK	: Normal end
FLK_PRN	: Parameter error
FLK_NG	: Abnormal end caused by other errors excluding the parameter error. In the Device Emulator, this value is always returned.

Messages

Table 1.18

WPARAM	Description
FLK_COMMAND_END	Terminate successfully.
FLK_COMMAND_ERROR	Terminate abnormally.

1.6.15 FLKGetFileInfoEx

This function retrieves information about file size, time stamp, attribute for a specified file at the partner station. The file information is set up in the file information area when the retrieved file with the same file is found at the partner station. If "wild card" has been specified, specify "Retrieve first information" for the first time, and "Retrieve subsequent information" for 2nd time forth. Call this function continuously if "wild card" has been specified. However, if other communication function is carried out, subsequent information cannot be retrieved. This function is operable in C#, VB, C and C++ development environments.

Calling Sequences

```
[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
)
```

Parameters

hPort

This parameter specifies the handle retrieved at a time of opening the communication port.

mode

This parameter specifies "First" or "Subsequent" flag selecting either one of the values listed below.

FLK_GET_FIRST	: Retrieve information for the first time if "one file" or wild card is specified.
FLK_GET_NEXT	: Retrieve subsequent information if wild card is specified for second time forth.

fName

This parameter specifies area to retrieve file names. Wild card is possible. Specifying multiple files is not possible. The parameter cannot be referred to if FLK_GET_NEXT has been set in the *mode* parameter.

fInfoex

This parameter specifies area for file information where retrieved files are set. "0x00" is set in each parameter if files to be searched are not found.

Return Values

FLK_OK : Normal end
FLK_PRM : Parameter error
FLK_NG : Abnormal end caused by other errors excluding the parameter error. In the Device Emulator, this value is always returned.

Messages

Table 1.19

WPARAM	Description
FLK_COMMAND_END	Terminate successfully.
FLK_COMMAND_ERROR	Terminate abnormally.

1.6.16 FLKSetFileInfoEx

While the HT mode has been set effect only, this function updates information about file size, time stamp, attribute for a specified file at the partner station by making the content in the file information area same with the content in the file name area. This function is operable in C#, VB, C and C++ development environments.

Calling Sequences

```
[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
)
```

Parameters

hPort

This parameter specifies the handle retrieved at a time of opening the communication port.

fInfoex

This parameter specifies file information area.

Return Values

FLK_OK	: Normal end
FLK_PRN	: Parameter error
FLK_NG	: Abnormal end caused by other errors excluding the parameter error. In the Device Emulator, this value is always returned.

Messages

Table 1.20

WPARAM	Description
FLK_COMMAND_END	Terminate successfully.
FLK_COMMAND_ERROR	Terminate abnormally.

1.6.17 FLKGetDiskInfo

While the HT mode has been set effect only, this function retrieves the specified drive information from the partner station and sets it to the drive information area. See also a note in the next page.

Calling Sequences

```
[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
)
```

Parameters

hPort

This parameter specifies the handle retrieved at a time of opening the communication port.

drive

This parameter specifies a drive number area selecting an alphabet in the range of A to Z.

dinfo

This parameter specifies an address of drive information area where information about retrieved drive is set. The value is set in this parameter when a message informing command has been carried out completely is received.

Return Values

FLK_OK	: Normal end
FLK_PRM	: Parameter error
FLK_NG	: Abnormal end caused by other errors excluding the parameter error. In the Device Emulator, this value is always returned.

Messages

Table 1.21

WPARAM	Description
FLK_COMMAND_END	Terminate successfully.
FLK_COMMAND_ERROR	Terminate abnormally.

Note:

An accurate size of drive cannot be retrieved if its capacity is too large. This limitation is derived from the protocol.

1.6.18 FLKDateTime

While the HT mode has been set effect only, this function retrieves or sets up date and time. The function either sets system date of the partner station to the date-time area if retrieving or transmits value in the date-time area to the partner station if setting.

Calling Sequences

```
[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
)
```

Parameters

hPort

This parameter specifies the handle retrieved at a time of opening the communication port.

mode

This parameter specifies "Retrieve" or "Set" mode flag selecting either one of the values listed below.

FLK_GET_MODE	: Retrieve
FLK_SET_MODE	: Set

dateTime

This parameter specifies storage area for date and time set.

Return Values

FLK_OK	: Normal end
FLK_PRN	: Parameter error
FLK_NG	: Abnormal end caused by other errors excluding the parameter error

Messages

Table 1.22

WPARAM	Description
FLK_COMMAND_END	Terminate successfully.
FLK_COMMAND_ERROR	Terminate abnormally.

1.6.19 FLKGetSysInfo, FLKGetSystemInfo

These functions retrieve information about system on the partner station. They return connection session number if the partner station is PC (or returns "0" if it is terminal). Since the information about the partner station has been already retrieved at a time of opening FLINK communication, communication between the partner station and the terminal will not take place with the functions but, only the information retrieved and internally stored in the terminal is returned with the functions.

Calling Sequences

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

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

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

Parameters

hPort

This parameter specifies the handle retrieved at a time of opening the communication port.

sysinfo

This parameter specifies area to retrieve system information where searched information is set.

Return Values

FLK_OK	: Normal end
FLK_PRM	: Parameter error
FLK_NG	: Abnormal end caused by other errors excluding the parameter error. In the Device Emulator, this value is always returned.

Message

None

1.6.20 FLKMsgSend, FLKSendMessage

While the HT mode has been set effect only, these functions transmit a message displayed on the partner station.

Calling Sequences

```
[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
)
```

Parameters

hPort

This parameter specifies the handle retrieved at a time of opening the communication port.

msg

This parameter specifies storage area for message to display.

Return Values

FLK_OK	: Normal end
FLK_PRM	: Parameter error
FLK_NG	: Abnormal end caused by other errors excluding the parameter error. In the Device Emulator, this value is always returned.

Messages

Table 1.23

WPARAM	Description
FLK_COMMAND_END	Terminate successfully.
FLK_COMMAND_ERROR	Terminate abnormally.

1.6.21 FLKBeep

While the HT mode has been set effect only, this function requests the partner station to turn on the buzzer. The buzzer at the partner station does not actually sound with this function.

Calling Sequences

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

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

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

Parameters

hPort

This parameter specifies the handle retrieved at a time of opening the communication port.

Return Values

FLK_OK	: Normal end
FLK_PRM	: Parameter error
FLK_NG	: Abnormal end caused by other errors excluding the parameter error. In the Device Emulator, this value is always returned.

Messages

Table 1.24

WPARAM	Description
FLK_COMMAND_END	Terminate successfully.
FLK_COMMAND_ERROR	Terminate abnormally.

2. Communication Specifications

2.1 Interfaces

The **FLINK Library** supports the following communication ports.

Table 2.1

Model no.	Communication Port			
	IrDA	Serial	LAN	USB
IT-3100	Yes	Yes	Yes (see note)	No
DT-X11	Yes	No	Yes	Yes
IT-600	Yes	No	Yes	Yes
DT-X7	Yes	No	Yes	Yes
DT-X30	Yes	No	Yes	Yes
IT-800	Yes	No	Yes	Yes
IT-300	No	No	Yes	Yes
DT-X8	No	No	Yes	Yes

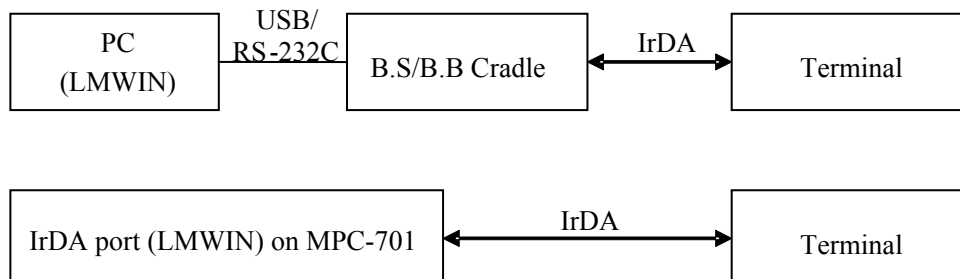
Note:

Because the model does not integrate WLAN module, a WLAN card available from third party should be employed to run WLAN operation.

2.2 Configurations

File transmissions using the **FLINK Library** can be performed in the system configurations illustrated below.

- IrDA communication with CASIO MPC-701 via the cradle



B.S : Bridge Satellite Cradle, B.B : Bridge Basic Cradle
Figure 2.1

- IrDA communication between two terminals

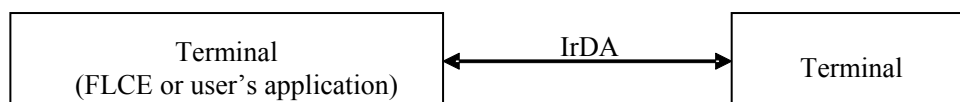


Figure 2.2

- Communication with host PC via WLAN and Wired LAN

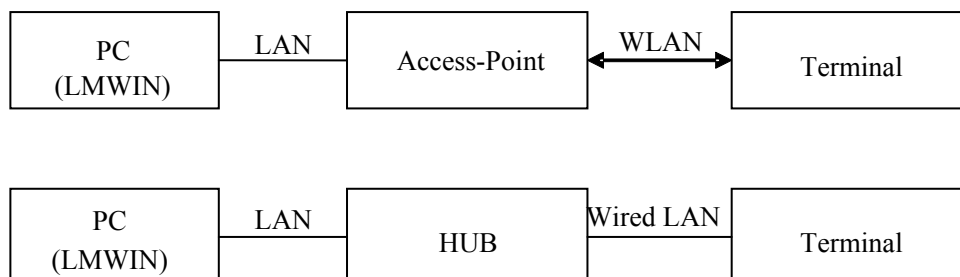


Figure 2.3

- Communication with host PC via USB Cradle

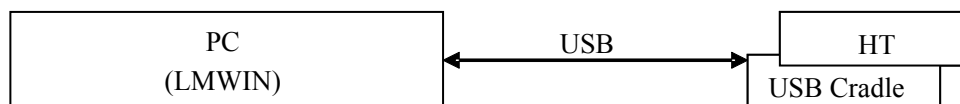


Figure 2.4

2.3 Communication Parameters

Common parameters

The parameters described in the table below can be set up in the registry.

Table 2.2

	Description	Range (in second)	Default (in second)
Timeout for session establishment	Time period to wait until the comm. session is established while the line is open.	0 to 3600	60
Timeout for wait reception	Time period to wait for command/response	0 to 600	30
Timeout for wait session end	Time period to wait until an acknowledgement for session end is returned by the partner station after the termination end command is sent by the send station.	0 to 600	10

IrDA communication parameters

Table 2.3

Parameter	Specification
Maximum baud rate	Not possible to set. The maximum baud rate is defined by the system (the registry).
Windows98 mode	This is the parameter to set when the communication is established in Windows98 OS via Bridge Satellite Cradle (or Bridge Basic Cradle). It is similar function to "/8 parameter" of FLCE.EXE. This parameter will be set in the registry.

Serial communication parameters

Table 2.4

Parameter	Specification
Baud rate	Selectable. 1200, 2400, 4800, 9600, 19200, 38400, 57600, or 115200
Data length	8 bits (fixed)
Parity bit	Selectable. None, Even, or Odd
Stop bit	Selectable. 1 bit or 2 bits
Flow control	RS/CS flow control only

LAN communication parameter

Table 2.5

Parameter	Specification
Destination IP address	Possible to set.

USB communication parameter

Table 2.6

Parameter	Specification
None	

2.4 Operation Modes

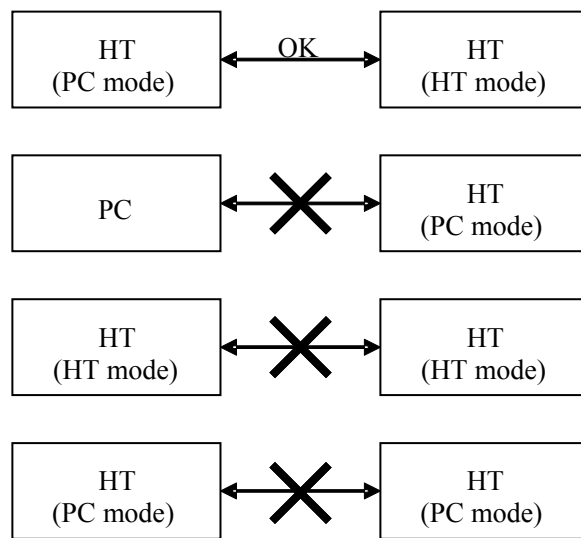
The **FLINK Library** supports the following two operation modes. Either one of the modes, HT mode or PC mode, is selected when opening the **FLINK Library**.

HT mode

This is the mode to set the right of sending commands after the communication session (a negotiation for checking the partner station when the line is open) is established. The right is set to either side of HT in the PC-and-HT configuration or in the HT-and-HT configuration.

PC mode

This is the pseudo PC mode after the communication session establishment waiting for command to be sent by the other HT.



X : not supported by the **FLINK Library**.

Figure 2.5

2.5 Command-Transmission Right

HT mode

- The command-transmission right mode (a mode in which the terminal specifies a command)
After the session has been established, the terminal retrieves the transmission right to send commands to achieve various executions. At a time when the terminal sends command to the PC, set the PC in server mode. For commands on the PC side, refer to LMWIN manual available separately.
- Idle mode (a mode in which the PC specifies a command)
To transfer the transmission right to the PC, IDLE notification command must be issued by the terminal. To set the terminal in idle mode for waiting command from the PC, **FLKIdle** function must be executed. The status on carrying out can be retrieved with **FLKReadStatus** function. When sending the IDLE notification command, script file can be specified to the PC. However, the terminal in PC mode (FLCE.EXE) does not have the ability to carry out script file.

PC mode

- After the session with the terminal has been established, the PC waits for commands to be sent by the terminal. The PC carries out commands one by one received from the terminal.
- The transmission right is transferred to the PC when IDLE notification is sent by the terminal.

2.6 Overview of the Process

The following table summarizes the operational and error handling processes provided with each function. Any time an error occurs communication will be promptly terminated. In this case, the side of the communication line that has detected the error earlier shall set the error information (category code/error detail code) in the termination instructing command and send it to the partner station, irrespective of whichever side has transmission priority. The partner station will detect an abnormal termination from the error information of the received termination instructing command.

Table 2.7

Function	Handling at the transmission-right station	Handling at the station without transmission right	Handling at an error
File transmission	After sending the command, sequentially sends the specified files.	Sequentially receives the specified files.	Deletes the files being received.
File reception	After sending the command, sequentially receives the specified files.	Sequentially sends the specified files.	
File append	After sending the command, sends the specified file.	After receiving the transferred file as a temporary file (FL. ADD), appends it to the existing file. After this append, deletes the temporary file.	Deletes the temporary file.
File delete	Sends the command.	Deletes the specified file/directory.	Does not recover the deleted file/directory.
File move	Sends the command.	Moves the specified file.	Does not recover the deleted file after move.
File create	Sends the command.	Creates the specified directory.	Does not delete the created directory.
Date/time retrieve	After sending the command, receives the date/time information.	Sends the system date/time information.	
Date/time set	Sends the command.	Sets the date/time for the system.	Does not recover the date/time after setting them.
File information retrieve	After sending the command, receives the file information.	Sends the specified file information.	
File information set	Sends the command.	Modifies the specified file information.	Does not recover the file information after setting it.
Disk information retrieve	After sending the command, receives the disk information.	Sends the specified disk information.	
System information retrieve	No communication is attempted, since the information about the partner station was received when the session was last established.	No communication is attempted, since the information about the partner station was received when the session was last established.	

Continue.

Screen message display	Sends the command.	Notifies the application that it has received the screen message display request. However, it does not display the message.	
Buzzer ON	Sends the command.	Notifies the application that it has received the buzzer sound request. However, it does not sound the buzzer.	
IDLE notification	(HT mode only) After sending the command, enters the wait mode for receiving a command.	(PC mode only) Retrieves the command transmission-right.	
Data transmission	Data transmission	Notifies the application that it has received the data transmission request.	

3. File Transmission/Reception Basic Functions

The file transmission and reception basic functions provide the capability that enables the transmission and reception of more than one file.

3.1 Communication Basic Functions

The following are the basic functions required to use the file transmission/reception and remote operation commands.

Initializing the communication port ("FLKOpen")

This function initializes the communication port. The following table lists the parameters and setup values for each communication port.

Table 3.1

COM1		
Baud rate	FLK_B1200(1200 bps), FLK_B2400(2400 bps), FLK_B9600(9600 bps), FLK_B19K(19.2 Kbps), FLK_B38K(38.4 Kbps), FLK_B57K(57.6 Kbps), FLK_B115K(115.2 Kbps)	
Data length	FLK_CHARS (Data length 8 bits)	
Parity	FLK_PARI_ODD (Odd), FLK_PARI_EVEN (Even), FLK_PARI_NON (Non-parity)	
Stop bit	FLK_STOP1 (1 stop bit), FLK_STOP2 (2 stop bits)	
IrDA		
No setting value		
LAN		
Destination IP address	Specify the IP address.	
USB		
No setting value		

Sets the APO function disabled.

Closing the communication port ("FLKClose")

- This function terminates communication and closes the communication port. For this purpose it transmits the termination instructing command to the partner station.
- However, if it has received the termination instructing command earlier than the partner station, it does not transmit the termination instructing command to the other station.
- Set the termination instructing command with the appropriate error information (category code/error detail code).
- Recovers the APO function setting that has been saved when the communication line was established.

Retrieving the error detail information ("FLKReadErrStat")

- This function retrieves error information.
- The error code, if generated, and error information (category code/error detail code) contained in the termination instructing command from the partner station, etc. can be retrieved.

Retrieving the communication status ("FLKReadStatus")

- With this function, it is possible to retrieve the progress status of communication (including the function being executed, the number of bytes transferred from the objective file, etc.).
- The application can retrieve information about whether the last executed command is terminated or continues to be executed.

Communication by transmitting the HT commands

- HT mode basic flow

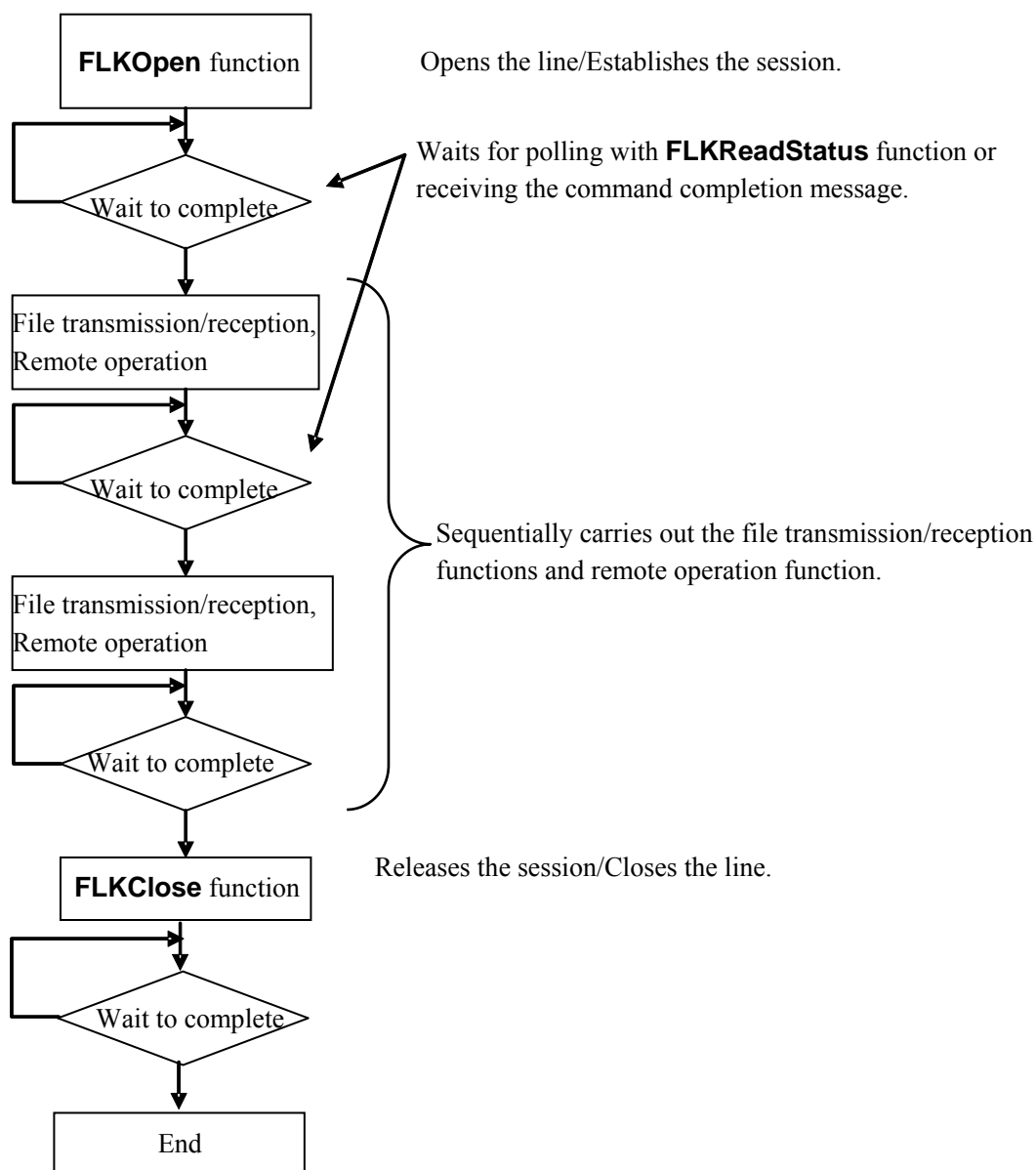


Figure 3.1

- PC mode basic flow

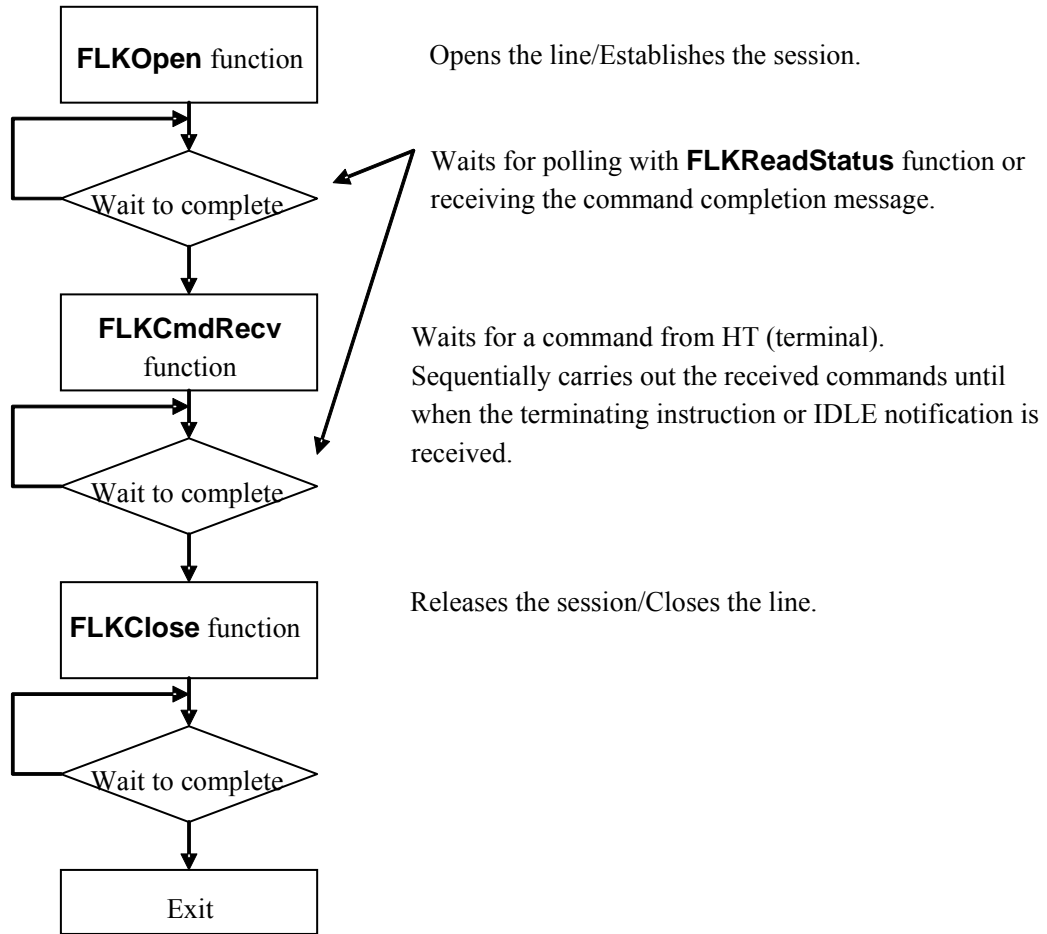


Figure 3.2

Communication by transmitting the PC commands

- HT mode basic flow

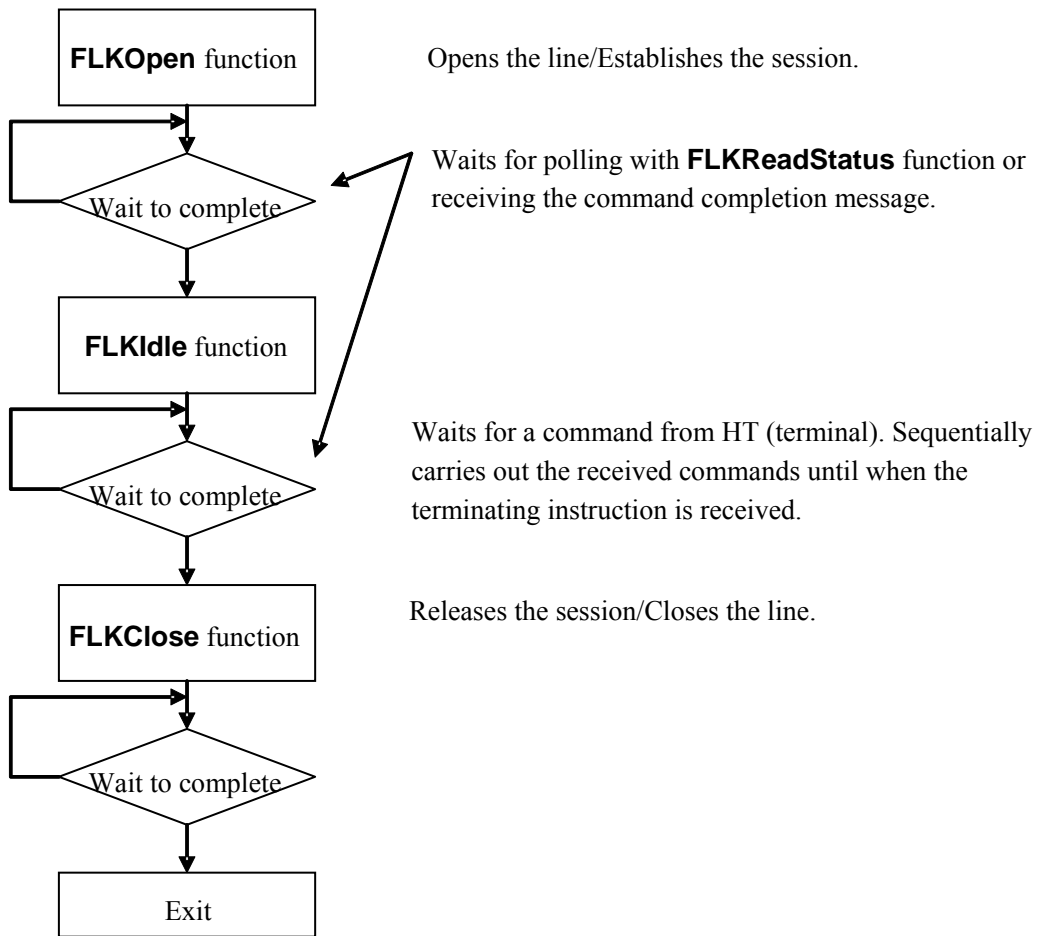


Figure 3.3

3.2 File Transmission/Reception Functions

These functions are used to implement file transfer (send, send (append), and receive) to and from the partner station. Either station with transmission priority will implement file transmission with the other station using the send file, send file (append), and receive file commands. The transmission-right station will enter the IDLE state (HT mode) or PC command wait state (PC mode) and be ready to accept the command from the partner.

This library does not include a function to display the progress of file transmission. This must be implemented as required from the application.

To indicate the progress status, calculate "total_size", "total_count", "file_size" and "file_count" of the FLK_STATUS structure by calling **FLKReadStatus** function when receiving FLK_STATUS_CHANGE2 message. The message is issued when either one of **FLKFileSend**, **FLKFileAdd**, **FLKFileRecv**, **FLKIdle**, and **FLKCmdRecv** functions is carried out.

File transmission ("FLKFileSend")

This function will batch-transmit more than one file. If the specified directory does not exist in the destination side, that directory will automatically be created there. The following options can be specified for the transmitted file.

- "Read-only file forcible overwrite" option
If the file to be transmitted already exists as a read-only file on the reception side, it can be forcibly overwritten. If this option is not specified, an attempt to overwrite the read-only file will result in an error.
- "Recursive call specification" option
The target of file transmission includes all files under the directory specified by the transmission file pathname. If the specified directory is followed by sub-directories, they will also be included in the targets of file transmission.

Example

(Transmission file name)	(Destination directory name)
"\SEND\AAA.DAT"	"C:\RECV\"

(Transmission-side directory structure)	
\---SEND\---SUB1\----AAA.DAT	C:\---RECV\---SUB1\---AAA.DAT
----SUB2\----BBB.DAT	----AAA.DAT
----AAA.DAT	
----BBB.DAT	

- Use of "wild card"
Wild card specification (*,?) is permitted for transmission file names.

File transmission (append) function ("FLKFileAdd")

With this function a file at the HT side can be appended to the specified file existing in the partner station. If the target file of this append does not exist on the partner side, that file will automatically be created. Neither a wild card specification nor multiple files specification is permitted.

File receive function ("FLKFileRecv")

This function will batch-receive more than one file. The following options can be specified for the received file.

- "Read-only file forcible overwrite" option
Even if the file to be received already exists as a read-only file on the reception side, it can be forcibly overwritten. If this option is not specified, an attempt to overwrite the read-only file will result in an error.
- "Recursive call specification" option
The target of file transfer includes all files under the directory specified by the reception file pathname. If the specified directory is followed by sub-directories, they will also be included in the targets of file reception.
- Use of "wild card"
Wild card specification (*,?) is permitted for reception file names.

IDLE notification ("FLKIdle")

This function assigns the transmission-right to the partner side, and enters the command wait state. Received commands will be sequentially executed until the termination instructing command is received or an error occurs. With this function it is also possible to specify carrying out of script files on the PC.

- IDLE transition basic flow

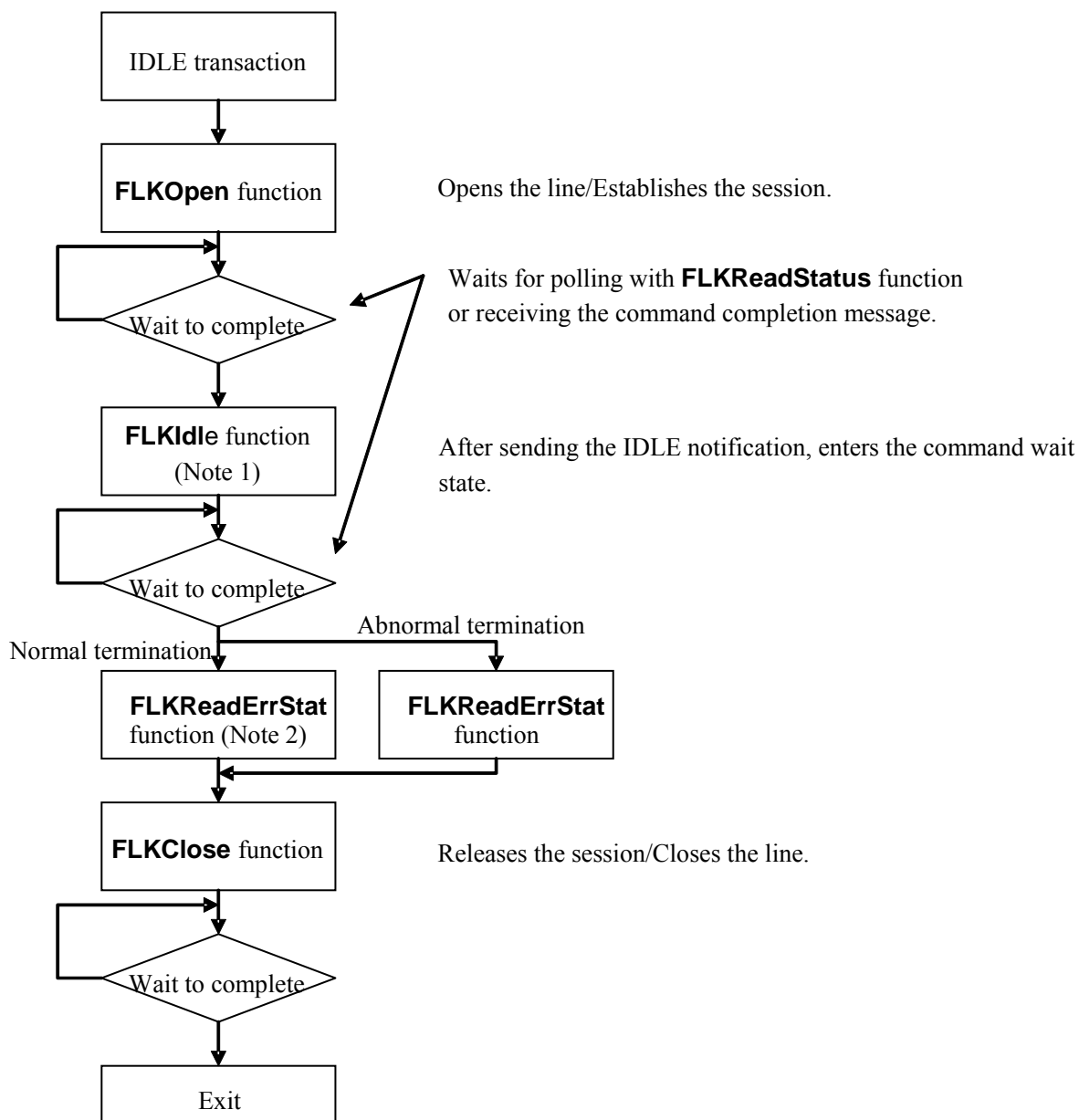


Figure 3.4

Notes:

1. Received commands will be carried out one by one until when the termination instructing command is received or an error occurs.
2. It is also possible, as required, to retrieve detailed information (specification of formatting, resetting, etc.) with the termination instructing command from the partner station.

Wait for command in the PC mode ("FLKCmdRecv")

This function enters into the wait state for receiving commands from HT (terminal). It is only available in the PC mode. As the transmission-right is assigned to the HT side just after the session is established, this terminal must, in the PC mode, use this function to enter the command wait state after the session is opened. The received commands will be executed one by one until when the termination instructing command or IDLE notification command is received or when an error occurs.

- Basic flow in the PC mode

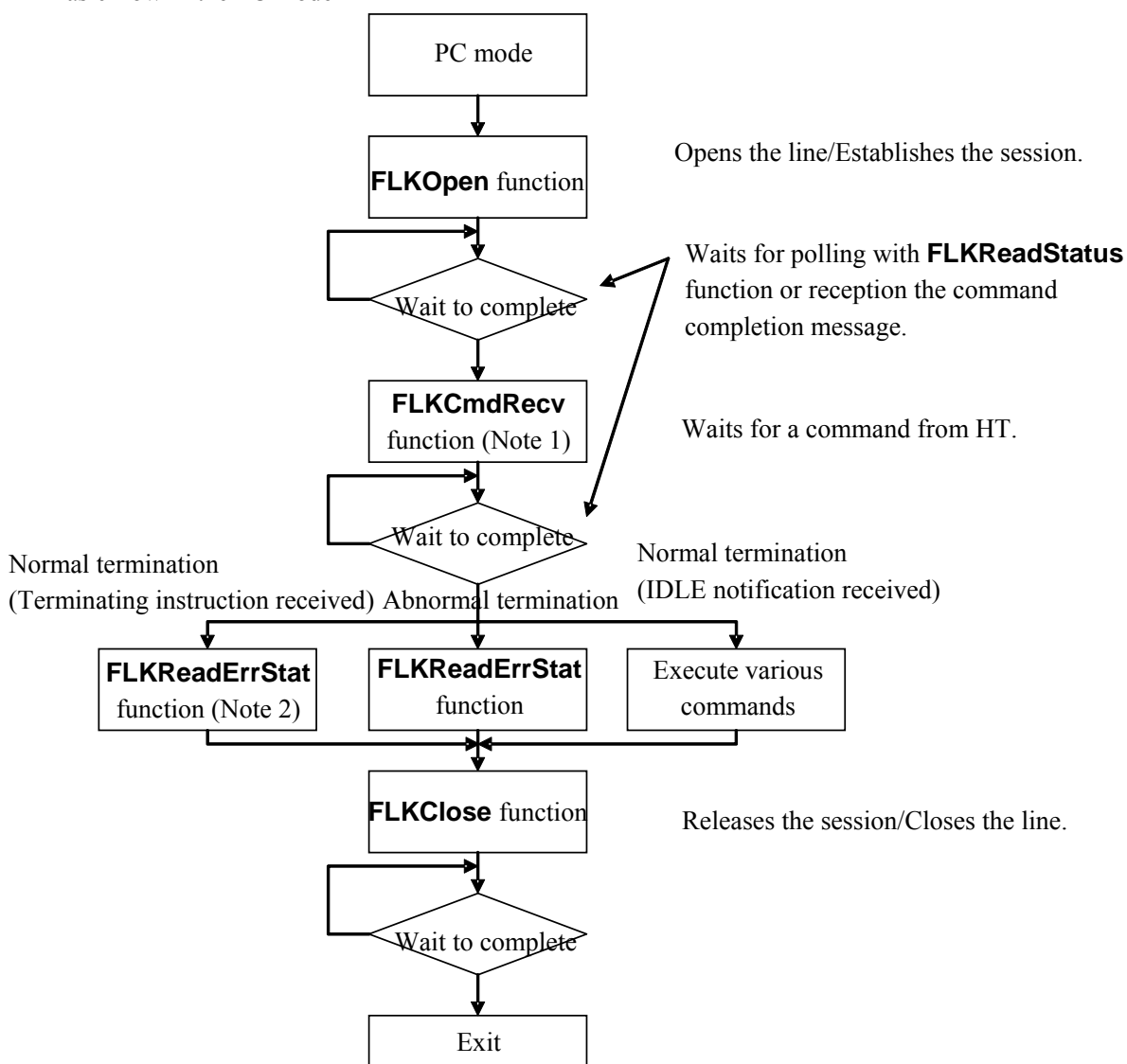


Figure 3.5

Notes:

1. Received commands will be carried out one by one until when the termination instructing command or IDLE notification command is received or an error occurs.
2. It is also possible, as required, to get detailed information (specification of formatting, resetting, etc.) with the termination instructing command from the partner station.

3.3 Remote Operation Functions

These functions are used to handle files or get/set the environment information on the partner station. However, they are only for handling files on the partner station, i.e. getting or setting the file/directory information on the partner station.

Deletion of file directory ("FLKFileDelete")

This function deletes the files and directories on the partner station. Specifying multiple files and wild card is possible. Even if the specified file does not exist on the partner, an error will not result.

Move file ("FLKFileMove")

This function moves or renames files within the same drive on the partner side. Neither specifying multiple files nor the use of wild card is possible. If the specified directory does not exist at the destination, it will be automatically created. If the drive name is different for the source and destination drives, an error will cause.

Create directory ("FLKMakeDir")

This function creates a directory on the partner station. Neither specifying multiple files nor the use of wild card is possible. The timestamp and related attributes can be set.

Retrieve file information ("FLKGetFileInfo", "FLKGetFileInfoEx")

These functions retrieve file information (timestamp, size, attribute) of the partner station. Use of wild card is permitted.

Set file information ("FLKSetFileInfo", "FLKSetFileInfoEx")

These functions update file information (timestamp, size, attribute) of the partner station.

3.4 Retrieve/Set Up Functions for Partner Information

These functions retrieve or set up the system environment information of the partner station.

Retrieve/Set date and time ("FLKDateTime")

This function retrieves or sets up the system date and time of the partner station.

Retrieve disk information ("FLKGetDiskInfo")

This function retrieves the disk information of the partner station. The disk information includes the following.

- Total disk capacity
- Amount of free disk space
- Disk conditions (Formatted/Unformatted/No disk)

Retrieve system information ("FLKGetSysInfo")

This function retrieves the system information of the partner station. The system information includes the following.

- Session ID (Session number at the time of communication)
- Protocol version (Version number of the file transfer protocol)
- Model code of the partner station (HT/PC (AT-compatible machine))
- OS model information (HT model type/OS type of PC)

The above listed information will be retrieved from the partner station immediately after the communication line is opened and the session is established.

Transmission of screen display message ("FLKMsgSend")

This function transmits messages to be displayed on the screen of the partner station.

Transmission of buzzer on request ("FLKBeep")

This function transmits the buzzer command to the partner station.

4. Notifying Communication Status

How to post notification of the communication status

Functions included in this Library are designed so that the active control will be returned to the source routine of the call after each function has been carried out. Therefore, in order to determine if carrying out the specified function is normally terminated, read the message posted by the library or the current status with **FLKReadStatus** function.

Display of the progress

This library has no function available to display the progress. However, a notification using an appropriate message takes place as the communication status changes. Make the application side display the progress information as required. In addition, it is always possible to read the current progress by carrying out **FLKReadStatus** function.

Message types

At the following timings messages will be posted to the window specified by the Library. Message numbers can be specified with the specific parameter of **FLKOpen** function.

Table 4.1

WPARAM value of message	Description	Remark
FLK_COMMAND_END	Carried out command is normally terminated.	The next command can be carried out.
FLK_COMMAND_ERROR	Carried out command is terminated as an error.	Carry out the close command.
FLK_STATUS_CHANGE1	The communication state has changed.	Reading the status with FLKReadStatus function allows the user to identify the currently carried out process.
FLK_STATUS_CHANGE2	The number of transferred bytes of the file changed during file transfer.	Reading the status with FLKReadStatus function allows the user to see the progress information of the transferred file.
FLK_INDICATION_END (available only when FLKIdle function is carried out.)	Termination order command is received from the partner station.	Carry out the close command.

5. Pathnames

For description of pathnames follow the naming rule of OS of machine (PC, terminal) where the path to be specified exists. To describe the path, follow the rules shown below for drive symbol letters.

1. Write the path name on the terminal beginning with the root directory, without using a drive symbol letter. This also applies to cases where the path of a file or directory on the terminal is specified from the LMWIN utility for Host PC.
2. If the communication partner side (PC, etc.) specifies a directory with some drive symbol letter, that drive symbol letter will be ignored by the Library (i.e. terminal as if the directory is specified so as to directly follow the root directory without a drive symbol letter).
3. To describe, on the terminal, the path of a file or directory of the communication partner (PC, etc.) on which an OS that requires a drive symbol letter runs, insert the drive symbol letter as usual.
4. However, as an exception to the rule no. 2 above, the user is permitted to make the drive symbol letter have the following meaning when the communication partner requests disk formatting or acquisition of the disk information after specifying the device on this terminal. These settings can be modified accordingly by describing them in the registry.

Table 5.1

Drive	DT-X11	IT-600	DT-X7	DT-X30	IT-3100	IT-800
C:	Windows	Windows	Windows	Windows	Windows	Windows
D:	CF card	CF card	USBStrage	USBStrage	CF card	USBStrage
E:	CF card	SD card	USBStrage	SD card	SD card	SD card
F:	FlashDisk	FlashDisk	FlashDisk	FlashDisk	FlashDisk	FlashDisk

Drive	IT-300	DT-X8
C:	Windows	Windows
D:	CF card	CF card
E:	CF card	SD card
F:	FlashDisk	FlashDisk

6. Registry

By writing a new value in the registry, it is possible to modify the following parameters. The registries will be read when the communication line is opened. Some of the registries use the same objects as for FLCE.EXE.

Table 6.1

Parameter	Default
Session establishment timeout	60 seconds
Reception wait timeout	30 seconds
Session completion wait timeout	10 seconds
Windows98 communication mode	0 second

Table 6.2 Assignment of drive symbol letters

Drive symbol letters	Default
C	\ Windows
D	\CF Card or \Storage card or \
E	\SD Card or \Storage card 2 or \
F	\FlashDisk or \

Location of the registry

[HKEY_CURRENT_USER\Software\CASIO\FLCE\]

Table 6.3

Key name	Type	Description
CONWAIT	DWORD	Session establishment timeout period
RECVWAIT	DWORD	Reception wait timeout period
DISCONWAIT	DWORD	Session completion wait timeout period
WIN98MODE	DWORD	0 (Set "1" for the Windows98 communication mode.)
DRIVE\A	STRING	Path to the device that is assigned as Drive A
DRIVE\B	STRING	Path to the device that is assigned as Drive B.
DRIVE\C	STRING	Path to the device that is assigned as Drive C.
:		
DRIVE\Z	STRING	Path to the device that is assigned as Drive Z.