

アロハ DLL ユーザーズ・マニュアル

第4.1版

2015年6月4日

目 次

1	概要.....	1
2	動作環境.....	1
3	インストールについて.....	1
3.1	インストール方法.....	1
3.2	旧バージョンとの共存.....	1
4	DLL 機能仕様.....	1
4.1	機能一覧.....	1
4.2	通信関連機能説明.....	3
4.2.1	シリアル通信.....	3
4.2.2	多回線対応.....	4
5	API 仕様.....	5
5.1	公開 API 一覧.....	5
5.1.1	USB 接続検出時のアロハとの通信確立/切断.....	5
5.1.2	指定した COM ポートでのアロハとの通信確立.....	5
5.1.3	アロハとの通信切断.....	6
5.1.4	接続状態取得.....	6
5.1.5	発 ID メモリ情報の読み込み.....	6
5.1.6	発 ID+オフ/未応答メモリ情報の読み込み.....	7
5.1.7	時刻設定.....	8
5.1.8	時刻同期.....	8
5.1.9	管理コード設定.....	9
5.1.10	管理コードの読み込み.....	9
5.1.11	イベントの読み込み.....	10
5.1.12	現在コールバックされているイベントのメッセージを取得する.....	10
5.1.13	一次応答する/一次応答しない設定.....	11
5.1.14	着信拒否モード/着信許可モード にする設定.....	11
5.1.15	着信拒否する電話番号 30 件の一括設定.....	11
5.1.16	着信許可する電話番号 30 件の一括設定.....	11
5.1.17	設定状況（一次応答 ON か OFF/着信拒否モードか着信許可モード）の読み込み.....	12
5.1.18	着信拒否設定 30 件の電話番号の読み込み.....	12
5.1.19	着信許可設定 30 件の電話番号の読み込み.....	12
5.1.20	PC からアロハへの接続確認.....	13
5.2	イベントの受信方法.....	14
5.2.1	コールバック関数 I/F 仕様.....	14
5.2.2	イベントポーリング仕様.....	15
5.3	エラーコード.....	15

6	サンプル・コード	16
6.1	発 ID 情報表示	16
6.1.1	コールバックを使う場合	16
6.1.2	ポーリングを使う場合	16
6.2	アプリ起動時のアロハ検出と初期設定	17
6.3	時刻設定	17
6.4	管理コード表示	17
7	サンプルアプリケーション	18
7.1	VC++版	18
7.1.1	制限事項	18
7.1.2	画面仕様	18
7.1.3	使用方法	20

1 概要

本書は「アロハ DLL」の API 仕様及びサンプル・アプリの使用方法等を記したものである。

2 動作環境

OS

Microsoft Windows Vista (32bit) Service Pack 2

Microsoft Windows 7 (32bit/64bit) Service Pack 1

Microsoft Windows 8 (32bit/64bit)

※各 Service Pack の適用は必須ではないが、弊社における動作保証は行わない。

DLL は OEM 文字セットを使用する為、アプリケーションが Unicode アプリケーションの場合は、適宜文字コード変換が必要です。

3 インストールについて

3.1 インストール方法

OS が 32bit の場合は

DLL4 版¥SDK&サンプルアプリインストーラ¥AlohaDLLSetup_Win32¥setup.exe

OS が 64bit の場合は

DLL4 版¥ SDK&サンプルアプリインストーラ¥AlohaDLLSetup_x64¥setup.exe

を実行する。

3.2 旧バージョンとの共存

既に古い DLL がインストールされている場合は、コントロールパネルの「プログラムの追加と削除」から、古いアロハ DLL を削除してからインストールする。削除せずに新しい DLL をインストールすると、ファイルが上書きされ、正常に動作しない可能性がある。

新旧を共存させたい場合は、古い DLL と異なるフォルダ名をつけてインストールする。ただし、新旧を同時に動作させることはできない。

4 DLL 機能仕様

4.1 機能一覧

※適用機器: アロハ ND3／アロハ ND4／アロハ ND5／アロハ ND6

No.	カテゴリー	内容	備考
1	シリアル通信	COM ポート Open/Close 制御 (シリアル通信設定を含む)	
2		メッセージ(コマンド/レスポンス、イベント)送受信 (ACK/NAK 応答を含む)	
3		アロハとの接続確認 アロハに対して 'CON' コマンドを送信し、'ALOHA' をアプリに返す。	アロハ ND5 アロハ ND6
4	多回線対応	1～8 回線に対応する。 USB ポートに装填されたアロハは自動検出し、OS 制御による COM ポートの割り当てを意識することなく、アプリ固有(任意ではない)の識別番号で制御可能とする。	
5	着信連携	発 ID 情報の連携	

		アロハから「発 ID 情報」を受信後、着信日時と発信電話番号をアプリに通知(コールバック)する。	
6		電話機フック情報の連携 アロハから「フック情報」を受信後、日時情報とフック情報(オフフック・オンフック)をアプリに通知(コールバック)する。	アロハ ND3 アロハ ND6
7		未応答情報の連携 アロハから「未応答情報」を受信後、着信日時と未応答をアプリへ通知(コールバック)する。	アロハ ND3 アロハ ND6
8		着信情報の連携 アロハから「着信情報」を受信後、着信日時(PC から取得)と着信情報をアプリへ通知(コールバック)する。	アロハ ND3
9	メモリ連携	発 ID メモリ情報の連携 アロハに対して 'M' コマンドを送信し、指定のメモリ番地に保存されている発 ID 情報をアプリに返す。	
10		発 ID + オフフック/未応答情報の連携 アロハに対して 'MT' コマンドを送信し、指定のメモリ番地に保存されている発 ID + オフフック/未応答情報をアプリに返す。	アロハ ND3 アロハ ND6
11		管理コード読み込み アロハに対して 'N' コマンドを送信し、登録番号をアプリに返す。	アロハ ND3 アロハ ND6
12		管理コード設定 アロハに対して 'USNR' コマンドを送信し、指定の管理コードを設定する。	アロハ ND3 アロハ ND6
13		一次応答設定 アロハに対して 'ON' または 'OFF' コマンドを送信し、一次応答するかを設定する。	アロハ ND5 アロハ ND6
14		着信タイプ設定 アロハに対して 'BLOC' または 'NUMB' コマンドを送信し、着信拒否モードと着信許可モードのどちらを使用するかを設定する。	アロハ ND6
15		着信拒否する電話番号 30 件を一括設定 アロハに対して 'BLOC' コマンドと電話番号 30 件を送信し、着信拒否番号に登録する。	アロハ ND6
16		着信許可する電話番号 30 件を一括設定 アロハに対して 'NUMB' コマンドと電話番号 30 件を送信し、着信許可番号に登録する。	アロハ ND6
17		設定状況読み込み アロハに対して 'SETUP' コマンドを送信し、一次応答と着信タイプ設定の設定状況をアプリに返す。	アロハ ND5 アロハ ND6
18		着信拒否された 30 件の番号を一括読み込み アロハに対して 'BL' コマンドを送信し、着信拒否された電話番号 30 件をアプリに返す。	アロハ ND6
19		着信許可された 30 件の番号を一括読み込み アロハに対して 'NU' コマンドを送信し、着信許可された電話番号 30 件をアプリに返す。	アロハ ND6
20	時刻設定連携	一括時刻設定 接続されているすべてのアロハに対して 'C' コマンドを送信し、PC から取得する日時を時刻情報として設定する。	アロハ ND3 アロハ ND6

4.2 通信関連機能説明

4.2.1 シリアル通信

1) COM ポート Open/Close 制御

i) シリアル通信設定

アロハ本体の通信仕様に従い、以下の通りにシリアル通信に関する設定を行う。

同期方式	非同期
通信速度	9600bps
ビット長	7bit
パリティ	偶数パリティ
ストップビット	1 ストップビット

ii) COM ポート Open

アロハが接続されている指定の COM ポートの Open 処理を行う。

iii) COM ポート Close

アロハが接続されている指定の COM ポートの Close 処理を行う。(一括してすべての COM ポートを対象とすることも可能)

2) メッセージ(コマンド/レスポンス、イベント)送受信

i) コマンド送信

アプリからの要求を受け、アロハに対してコマンドを送信する。

アロハからのコマンドに対するレスポンスを待ち、受信後、受信データをアプリに返す。

コマンド送信時のレスポンス待ちタイマーは 2 秒です。

レスポンスが返ってこない場合は、エラーコード(-11:ALOHA_ERROR_TIMEOUT)をアプリに返します。

ii) メッセージ(イベント)受信

アロハから発 ID 情報等の非同期のメッセージを受信し、受信データをアプリに通知する。

アプリからは、予めコールバック関数を設定させ、その関数を呼び出すことで、通知する。

3) 接続テスト

i) コマンド送信

アプリから 'CON' コマンドを送信し、アロハとの接続確認を行う。

ii) メッセージ受信

アロハから 'ALOHA' というメッセージを受信する。

コマンドのフォーマットは以下の通り。

STX (02H) (1 キャラクタ)	CON (3 キャラクタ)	ETX (03H) (1 キャラクタ)
------------------------	------------------	------------------------

アロハからレスポンス・データを受信し、レスポンス・データに含まれる 'ALOHA' をアプリに受け渡す。レスポンス・データのフォーマットは、以下の通り。

STX (02H) (1 キャラクタ)	ALOHA (5 キャラクタ)	ETX (03H) (1 キャラクタ)
------------------------	--------------------	------------------------

4.2.2 多回線対応

1) 最大 8 回線まで対応する。

2) 自動検出

USB ポートに装填されたアロハは、Windows(OS)によって自動検出され、COM ポートを割り当てられるが、OS が割り当てる COM ポートは不定値である為、アプリがこれを意識することなく任意のアロハを制御できるよう、DLL にて固有のポート番号(連番)を割り当てる。

[割り当てのルール]

- ・0～7 を順に割り当てる。
- ・装置が外された場合、既に割り当てたポート番号を詰めることはしない。
- ・装置が装填された場合、0 から空きポートを検索し、見つければ、その番号を採用する。

※OS が接続を検出したことを知るのは、アプリの役割とする。アプリは、接続を認識した後、DLL の COM ポート Open/Close 制御を呼び出す。

5 API 仕様

5.1 公開 API 一覧

5.1.1 USB 接続検出時のアロハとの通信確立/切断

概要:	ホットプラグイベントに応じて、アロハへ接続または切断する		
名称:	Aloha_Hotplug		
引数:	No.	型	内容
	1	WPARAM	USB 接続検出時に OS からアプリが受け取る Windows メッセージの第 1 引数。
	2	LPARAM	USB 接続検出時に OS からアプリが受け取る Windows メッセージの第 2 引数。
	3	Aloha_Callback_t	接続したアロハからのイベントを受け取るイベントハンドラへのポインタ。 または、イベントをポーリングで検出する場合は NULL。
	4	void*	イベントハンドラへ渡す追加引数。
戻り値:	型	値	内容
	int	0～7	機器接続だった場合: 接続成功。値は API 内部で管理する機器番号。 機器切断だった場合: 切断成功。 接続/切断イベントでなかった場合 常に 0 を返す。
		負	失敗。値は ALOHA_ERROR_XXXX で定義されるエラー番号。

5.1.2 指定した COM ポートでのアロハとの通信確立

概要:	すでに存在する COM ポートを通じてアロハへ接続します。		
名称:	Aloha_SetupPort		
引数:	No.	型	内容
	1	char*	COM ポート名("COM1"等)を表す文字列へのポインタ。
	2	Aloha_Callback_t	接続したアロハからのイベントを受け取るイベントハンドラへのポインタ。 または、イベントをポーリングで検出する場合は NULL。
	3	void*	イベントハンドラへ渡す追加引数。
戻り値:	型	値	内容
	int	0～7	接続成功。値は API 内部で管理する機器番号。
		負	接続失敗。値は ALOHA_ERROR_XXXX で定義されるエラー番号。

5.1.3 アロハとの通信切断

概要:	Aloha_Set や Aloha_Hotplug で確立した通信を切断する。		
名称:	Aloha_TearDown		
引数:	No.	型	内容
	1	int	接続 API が戻した機器番号。
戻り値:	型	値	内容
	int	ALOHA_OK (0)	成功。
		負	失敗。値は ALOHA_ERROR_XXXX で定義されるエラー番号。

5.1.4 接続状態取得

概要:	機器番号 0～7 について接続状態を取得する		
名称:	Aloha_GetConnStatus		
引数:	No.	型	内容
	1	int	0～7 の機器番号
	2	char*	対応する COM ポート名を受け取るバッファ。 接続されていない場合は空文字列が設定される。
	3	unsigned int	COM ポート名を受け取るバッファの長さ。
戻り値:	型	値	内容
	int	ALOHA_OK (0)	成功。
		ALOHA_INVALID_DEVICE_NUMBER(-2)	機器番号が無効です。

5.1.5 発 ID メモリ情報の読み込み

概要:	アロハに”M”コマンドを送信し、指定番地のメモリ内容(発 ID)を読み込む。		
名称:	Aloha_ReadMemory		
引数:	No.	型	内容
	1	int	接続 API が戻した機器番号。
	2	unsigned int	読み込む 1～30 のメモリ番地。
	3	Aloha_MemoryMessage_t*	読み込んだメモリ内容を受け取るバッファへのポインタ。
戻り値:	型	値	内容
	int	ALOHA_OK (0)	成功。 読み込んだメモリ内容は、第 3 引数で指定したバッファへ格納される。
		負	失敗。値は ALOHA_ERROR_XXXX で定義されるエラー番号。

Aloha_MomeryMessage_t 型の構造

Aloha_MemoryMessage_t の構造は以下のとおり。

No	名称	型	バイト数	内容
1	datetime	Aloha_DateTime_t	9	着信日時
2	telno	Aloha_Telno_t	20	電話番号

Aloha_Datetime_t 型の構造

Aloha_Memory_Message_t の構造は以下のとおり。

No	名称	型	バイト数	内容
1	month	char[2]	2	月
2	day	char[2]	2	日
3	weekday	char[1]	1	曜日 (“0”が日曜日)
4	hour	char[2]	2	時
5	minute	char[2]	2	分

5.1.6 発 ID+オフ/未応答メモリ情報の読み込み

概要:	アロハに”MT”コマンドを送信し、指定番地のメモリ内容(発 ID+オフ/未応答)を読み込む。		
名称:	Aloha_ReadMemoryEx		
引数:	No.	型	内容
	1	int	0～7 の機器番号
	2	unsigned int	読み込む 1～30 のメモリ番地。
	3	Aloha_MemoryMessageEx_t*	読み込んだメモリ内容を受け取るバッファへのポインタ。
戻り値:	型	値	内容
	int	ALOHA_OK (0)	成功。
		ALOHA_INVALID_DEVICE_NUMBER(-2)	機器番号が無効です。

Aloha_MomeryMessageEx_t 型の構造

Aloha_MemoryMessageEx_t の構造は以下のとおり。

No	名称	型	バイト数	内容
1	memory_msg	Aloha_MemoryMessage_t	29	発 ID メモリ情報。 ※ 6 ページ参照
2	hook_msg	Aloha_HookStatusMessage_t	11	オフ/未応答メモリ情報。

Aloha_HookStatusMessage_t 型の構造

Aloha_HookStatusMessage_t の構造は以下のとおり。

No	名称	型	バイト数	内容
1	datetime	Aloha_DateTime_t	9	日時。 ※ 7 ページ参照
2	hookstatus	Aloha_HookStatus_t	2	フックステータス。 “ON”: オンフック “OF”: オフフック “UR”: 未応答

5.1.7 時刻設定

概要:	アロハに”C”コマンドを送信し、時刻を設定する。		
名称:	Aloha_SetCalendar		
引数:	No.	型	内容
	1	int	0～7 の機器番号
	2	int	2010～2110 の年
	3	int	1～12 の月。
	4	int	1～31 の日
	5	int	0～23 の時
	6	int	0～59 の分
戻り値:	型	値	内容
	int	ALOHA_OK (0)	成功。
		負	失敗。値は ALOHA_ERROR_XXXX で定義されるエラー番号。

5.1.8 時刻同期

概要:	アロハに”C”コマンドを送信し、時刻を PC と同期する。 現在日時を引数に取って Aloha_SetCalendar を呼び出すのと同等。		
名称:	Aloha_SyncCalendar		
引数:	No.	型	内容
	1	int	0～7 の機器番号
戻り値:	型	値	内容
	int	ALOHA_OK (0)	成功。
		負	失敗。値は ALOHA_ERROR_XXXX で定義されるエラー番号。

5.1.9 管理コード設定

概要:	アロハに”USNR”コマンドを送信し、機器の管理コードを設定する。		
名称:	Aloha_SetDeviceID		
引数:	No.	型	内容
	1	int	0～7 の機器番号
	2	Aloha_DeVID_t*	機器管理コードを受け取る 17 文字(16 文字 + ‘¥0’)のバッファへのポインタ。 末尾の‘¥0’は自動的に付加される。
戻り値:	型	値	内容
	int	ALOHA_OK (0)	成功。
		負	失敗。値は ALOHA_ERROR_XXXX で定義されるエラー番号。

5.1.10 管理コードの読み込み

概要:	アロハに”N”コマンドを送信し、機器の管理コードを読み込む。		
名称:	Aloha_GetDeviceID		
引数:	No.	型	内容
	1	int	0～7 の機器番号
	2	Aloha_DeVID_t*	機器管理コードを格納した 17 バイト(16 文字 + ‘¥0’)のバッファ。 16 文字に満たない部分は空白で埋められ、逆に長すぎる部分は無視される。
戻り値:	型	値	内容
	int	ALOHA_OK (0)	成功。
		負	失敗。値は ALOHA_ERROR_XXXX で定義されるエラー番号。

5.1.11 イベントの読み込み

概要:	アロハ DLL 内部のキューに溜まっているイベントを取り出す。		
名称:	Aloha_GetEvent		
引数:	No.	型	内容
	1	int*	0~7 の機器番号を受け取る変数へのポインタ。
	2	unsigned int*	イベント番号を受け取る変数へのポインタ。
	3	char*	イベントのメッセージを受け取る変数へのポインタ。
	4	unsigned int*	イベントのメッセージの長さを受け取る変数へのポインタ
戻り値:	型	値	内容
	int	ALOHA_OK (0)	成功。
		ALOHA_QUEUE_EMPTY(-10)	イベントは発生していない。

Aloha_Setup や Aloha_Hotplug でコールバック関数のポインタに NULL を指定した場合、到着したイベントを取り出す為に Aloha_GetEvent を定期的呼び出す必要がある。内部キューには最大 1024 個のイベントを溜める事ができるが、溢れたイベントは捨てられてしまう。

Aloha_GetEvent で取り出す事のできるイベントは、コールバック関数のポインタに NULL を指定した機器に対するものだけで、コールバック関数が登録されている場合は、コールバック後に破棄されて、内部キューには溜まらない。

5.1.12 現在コールバックされているイベントのメッセージを取得する

概要:	コールバック関数内から、発生したイベントのメッセージ本文を取り出す。		
名称:	Aloha_GetCurrentCallbackMessage		
引数:	No.	型	内容
	1	int	コールバック関数に渡された機器番号。
	2	char*	イベントのメッセージを受け取る変数へのポインタ。
戻り値:	型	値	内容
	int	ALOHA_OK (0)	成功。
		ALOHA_ERROR_INVALID_DEVICE_NUMBER;(-2)	機器番号が無効。

イベントの受信にコールバック I/F を使う場合、アプリケーション側の言語によっては、コールバック関数にポインタを渡されても、そのポインタの指している先の内容にアクセスする手段が無い場合(その言語にポインタ型が無い等)がある。そのような場合には、本 API を使用して、コールバック関数の第3引数に渡されたメッセージのコピーを改めて取得する事ができる。

取得されるメッセージは、コールバック関数の第 3 引数に渡されるものと同じ(コピー)であり、コールバック関数の第 4 引数で渡された長さ以上の領域を確保する必要がある。

アプリケーションの開発言語でポインタを問題無く扱える場合や、イベントの受信にポーリングを使う場合には、本 API を使う必要は無い。

コールバック関数以外の場所から本 API を呼び出してはならない。

5.1.13 一次応答する/一次応答しない設定

概要:	一次応答設定コマンドに応じて、一次応答設定を行う。		
名称:	Aloha_SetFirstResponse		
引数:	No.	型	内容
	1	int	0～7 の機器番号
	2	int	ALOHA_ON なら一次応答する。 ALOHA_OFF なら一次応答しない
戻り値:	型	値	内容
	int	ALOHA_OK (0)	成功。
		負	失敗。値は ALOHA_ERROR_XXXX で定義されるエラー番号。

5.1.14 着信拒否モード/着信許可モード にする設定

概要:	着信タイプ設定コマンドに応じて、着信モードの設定を行う。		
名称:	Aloha_SetRecvMode		
引数:	No.	型	内容
	1	int	0～7 の機器番号
	2	int	ALOHA_NUMB なら着信許可する。 ALOHA_BLOC なら着信拒否する。
戻り値:	型	値	内容
	int	ALOHA_OK (0)	成功。
		負	失敗。値は ALOHA_ERROR_XXXX で定義されるエラー番号。

5.1.15 着信拒否する電話番号 30 件の一括設定

概要:	アロハに” BLOC”コマンドを送信し、着信拒否する電話番号 30 件を一括設定する。		
名称:	Aloha_SetBlockTelno		
引数:	No.	型	内容
	1	int	0～7 の機器番号
	2	Aloha_Telno_all_t *	電話番号 30 件
戻り値:	型	値	内容
	int	ALOHA_OK (0)	成功。
		負	失敗。値は ALOHA_ERROR_XXXX で定義されるエラー番号。

5.1.16 着信許可する電話番号 30 件の一括設定

概要:	アロハに” NUMB”コマンドを送信し、着信許可する電話番号 30 件を一括設定する。		
名称:	Aloha_SetPermitTelno		
引数:	No.	型	内容
	1	int	0～7 の機器番号
	2	Aloha_Telno_all_t *	電話番号 30 件
戻り値:	型	値	内容
	int	ALOHA_OK (0)	成功。
		負	失敗。値は ALOHA_ERROR_XXXX で定義されるエラー番号。

5.1.17 設定状況（一次応答 ON か OFF/着信拒否モードか着信許可モード）の読み込み

概要:	アロハに”SETUP”コマンドを送信し、設定状況を呼出す。		
名称:	Aloha_GetStatus		
引数:	No.	型	内容
	1	int	0～7 の機器番号
	2	Aloha_SetupMessage_t*	読み込んだメモリ内容を受け取るバッファへのポインタ
戻り値:	型	値	内容
	int	ALOHA_OK (0)	成功。
		負	失敗。値は ALOHA_ERROR_XXXX で定義されるエラー番号。

Aloha_SetupMessage_t 型の構造

Aloha_SetupMessage_t の構造は以下のとおり。

No	名称	型	バイト数	内容
1	setup_stat	char [7]	7	設定内容 “ON” “OFF” + “BLOC” “NUMB”の文字列

5.1.18 着信拒否設定 30 件の電話番号の読み込み

概要:	アロハに”BL”コマンドを送信し、着信拒否された 30 件の電話番号を呼出す。		
名称:	Aloha_GetBlockTelno		
引数:	No.	型	内容
	1	int	0～7 の機器番号
	2	Aloha_TelnoAllMessage_t*	電話番号 30 件分の内容を受け取るバッファへのポインタ
戻り値:	型	値	内容
	int	ALOHA_OK (0)	成功。
		負	失敗。値は ALOHA_ERROR_XXXX で定義されるエラー番号。

Aloha_TelnoAllMessage_t 型の構造

Aloha_TelnoAllMessage_t の構造は以下のとおり。

No	名称	型	バイト数	内容
1	telno_all	Aloha_Telno_all_t	630	電話番号 20 桁 30 件分 区切り用のスラッシュ 29 個分 終端 NULL 1 個分

5.1.19 着信許可設定 30 件の電話番号の読み込み

概要:	アロハに”NU”コマンドを送信し、着信許可された 30 件の電話番号を呼出す。		
名称:	Aloha_GetPermitTelno		
引数:	No.	型	内容
	1	int	0～7 の機器番号
	2	Aloha_TelnoAllMessage_t*	電話番号 30 件分の内容を受け取るバッファへのポインタ
戻り値:	型	値	内容
	int	ALOHA_OK (0)	成功。
		負	失敗。値は ALOHA_ERROR_XXXX で定義されるエラー番号。

5.1.20 PC からアロハへの接続確認

概要:	アロハに” CON”コマンドを送信し、アロハとの接続確認を行う。		
名称:	Aloha_PingAloha		
引数:	No.	型	内容
	1	int	0～7 の機器番号
	2	Aloha_PingMessage_t*	アロハからの応答メッセージ内容を受け取るバッファへのポインタ
戻り値:	型	値	内容
	int	ALOHA_OK (0)	成功。
		負	失敗。値は ALOHA_ERROR_XXXX で定義されるエラー番号。

Aloha_PingMessage_t 型の構造

Aloha_PingMessage_t の構造は以下のとおり。

No	名称	型	バイト数	内容
1	recv_msg	char [5]	5	アロハからの応答メッセージ

5.2 イベントの受信方法

アロハで発生したイベントを受信する方法には、コールバック関数を使う手段と、イベントポーリングを行う手段の2種類がある。

5.2.1 コールバック関数 I/F 仕様

Aloha_Setup や Aloha_Hotplug にコールバック関数を指定した場合、接続しているアロハでイベントが発生したタイミングで指定したコールバック関数が呼ばれる。

コールバック関数の仕様は以下のとおり。

概要:	アロハから送信されたイベントを受信する		
名称:			
引数:	No.	型	内容
	1	int	機器番号。
	2	unsigned int	イベント番号。
	3	char*	イベントのメッセージが格納されたバッファ。
	4	unsigned int	イベントのメッセージの長さ
	5	void*	コールバック関数登録時に指定した追加引数。
戻り値:	型	内容	
	void	戻り値は無い	

コールバック関数は、アロハ DLL のデータ受信スレッドから呼び出される。

この為、コールバック関数内で、Aloha_GetCurrentCallbackMessage 以外の API 関数呼び出しや、スレッドがブロックするような操作を行うとデッドロックや通信の停止を引き起こすので、それらの操作を行ってはならない。

第2引数の値でどのようなイベントが発生したかを判別でき、受信したメッセージは第3引数に渡ってくるバッファに格納されている。

名称	値	メッセージの内容	内容
ALOHA_EVENT_UNKNOWN	0	受信した生データ。	不明なイベント。 通信状態が正常の場合は通常発生しない。
ALOHA_EVENT_FOREIGN_TELNO	1	Aloha_MemoryMessage_t ※ 6ページ参照	発 ID 情報送信。
ALOHA_EVENT_HOOK_STATUS	2	Aloha_HookStatusMessage_t ※ 7ページ参照	フック情報送信。
ALOHA_EVENT_UNRESPONDED	3	Aloha_HookStatusMessage_t ※ 7ページ参照	未応答情報送信。
ALOHA_EVENT_RING	4	Aloha_RingMessage_t	着信情報送信。
ALOHA_EVENT_COMM_ERROR	255	メッセージは存在しないので、NULL が渡される。	通信エラーが発生した。

Aloha_RingMessage_t 型の構造

Aloha_MemoryMessage_t の構造は以下のとおり。

No	名称	型	バイト数	内容
1	rc	char[2]	2	常に”RC”
2	datetime	Aloha_DateTime_t	9	日時。 ※ 7 ページ参照

5.2.2 イベントポーリング仕様

アプリケーション側で使用している言語上の制約、またはコールバック関数仕様上の制約が問題になる場合、イベントをポーリングで取得する事もできる。

Aloha_SetupPort や Aloha_Hotplug API に渡すコールバック関数に NULL を指定した場合、アロハ DLL は発生したイベントのコールバックを行う代わりに、DLL 内部のイベントキューにデータを記録する。

アプリケーションは、適当な周期で Aloha_GetEvent API によってこのイベントキューを監視する事でイベントを取得できる。

ただし、キューをポーリングするという仕組み上、以下の制約がある。

1. イベントの検出がポーリング周期分遅れる可能性がある。
2. キューに溜める事のできるイベントの数は 1024 と十分大きいですが、ポーリング周期が長すぎてキューを溢れた場合は、以降のイベントは捨てられてしまう。

5.3 エラーコード

アロハ DLL の各 API が戻す可能性のあるエラーコードとその意味は以下の通り。

値	名称	内容
0	ALOHA_OK	成功
-1	ALOHA_ERROR_GENERIC	不明なエラー（このエラーが戻る事はありません）
-2	ALOHA_ERROR_INVALID_DEVICE_NUMBER	機器番号(0～7)が不正
-3	ALOHA_ERROR_COMM_CANNOT_OPEN	COM ポートのオープンに失敗しました
-4	ALOHA_ERROR_COMM_SETUP	COM ポートの初期化に失敗しました
-5	ALOHA_ERROR_COMIO	COM ポートのアクセス時に OS からエラーが戻りました
-7	ALOHA_ERROR_TOO_MANY_DEVICES	これ以上機器を開けません
-8	ALOHA_ERROR_HOTPLUG_IS_NOT_COMM	挿された USB 機器は COM ポートではありませんでした
-9	ALOHA_ERROR_HOTPLUG_NOT_OPENED	抜かれた USB 機器は DLL の管理下にありませんでした
-10	ALOHA_ERROR_QUEUE_EMPTY	イベントキューは空です
-11	ALOHA_ERROR_TIMEOUT	アロハに送信したコマンドがタイムアウトしました
-12	ALOHA_ERROR_INVALID_PARAM	設定値(0、1)が不正。

6 サンプル・コード

6.1 発 ID 情報表示

6.1.1 コールバックを使う場合

```
static void __stdcall MyCallback(int devnum, unsigned int event_id, char const* message, unsigned int msg_length, void* userdata)
{
    // NOTE: コールバック関数は、アロハ DLL 内の受信スレッドから呼び出されます。
    //       可能であれば、メッセージをバッファへコピーしてから速やかにリターンしてください。

    ((CAlohaDLLTestDlg*)userdata)->OnAlohaEvent(devnum, event_id, message, msg_length); // ダイアログメンバにアクセス
    // するため、一旦転送します。
}

//
// アロハからイベント通知があった時に、コールバック関数から転送されてきます。
//
void CAlohaDLLTestDlg::OnAlohaEvent(int devnum, unsigned int event_id, char const* message, unsigned int msg_length)
{
    (略)
    switch (event_id)
    {
    {
        case ALOHA_EVENT_FOREIGN_TELNO: // 発 ID 情報
            this->SetDlgItemText(event_name_id, _T("発 ID 情報"));

            {
                Aloha_MemoryMessage_t* p = (Aloha_MemoryMessage_t*)message;
                this->SetDlgItemText(param1_id, LPCTSTR(CString(p->datetime.str, sizeof(p->datetime))));
                this->SetDlgItemText(param2_id, LPCTSTR(CString(p->telno, sizeof(p->telno))));
            }
            break;
    }
    (略)
    }
}
```

6.1.2 ポーリングを使う場合

```
Sub 周期起動プロシジャ()
(略)
Dim devnum As Long
Dim event_id As Long
Dim msg As String * 128
Dim msg_len As Long
If Aloha_GetEvent(devnum, event_id, msg, msg_len) = 0 Then
    (略)
    Select Case event_id
    (略)
    Case 1:
        Cells(row, 4).Value = "発 ID 情報"
        Cells(row, 5).Value = Mid(msg, 1, 9)
        Cells(row, 7).Value = Mid(msg, 10, 20)
    (略)
    End Select
End Sub
```

```

End If

Application.OnTime Now + TimeValue("1 秒"), "周期起動プロシジャー"
End Sub

```

6.2 アプリ起動時のアロハ検出と初期設定

```

Bool CAlohaDLLTestDlg::OnInitDialog()
{
    (略)
    // 存在するアロハに接続を試みる
    for (int portno = 0; portno < 8; portno++) {
        CString portname;
        portname.Format(_T("COM%d"), portno);
        Aloha_SetupPort(LPCTSTR(portname), MyCallback, this);
    }
    (略)
}

```

6.3 時刻設定

```

Aloha_SyncCalendar(devno):// 時刻設定 呼び出し

```

6.4 管理コード表示

```

Aloha_DevID_t devid;
if (Aloha_GetDeviceID(devno, &devid) == ALOHA_OK)
    this->SetDlgItemText(IDC_DEVICEID_R, A2T(devid)); // // 管理コード読込 呼び出し

```

7 サンプルアプリケーション

7.1 VC++版

7.1.1 制限事項

サンプルアプリケーションを起動するには、Microsoft Visual C++ 2010 のランタイムが必要になる。アロハ DLL のインストーラはこのランタイムを自動的にインストールするが、EXE 単体で配布する場合には別途 Microsoft Visual C++ 2010 のランタイムをマイクロソフトのサイトからダウンロードおよびインストールを行う必要がある。

アロハ DLL は DLL 検索パス上に存在する必要がある。

7.1.2 画面仕様

アロハDLL (4版) サンプルアプリケーション

受信イベント(1)
ポート:
種類:
パラメータ(1):
パラメータ(2):

受信イベント(2)
ポート:
種類:
パラメータ(1):
パラメータ(2):

ポート: 0

☐ 発IDメモリ情報読込
☐ 発ID + オフ/未応答メモリ情報読込

番地:
結果:
着信日時:
着信電話番号:
日時:
オフ/未応答:

☐ 管理コード設定
番号:

☐ 時刻設定 (現在のPCの時刻をアロハに設定します)
☐ 時刻リセット (2014/1/1 0時0分に設定します)

☐ 一次応答設定 (電話機にナリバーディスプレイ機能がなし/あり)
なし

ポート COM番号 管理コード
0 COM4
1
2
3
4
5
6
7

☐ 着信拒否/着信許可番号設定
☐ 着信拒否/着信許可番号30件読込

拒否設定

No.	拒否リスト	許可リスト
1	01234567890123456789	01234567890123456789
2	02	02
3	03	03
4	04	04
5	05	05
6	06	06
7	07	07

☐ 設定状況読込
一次応答設定: 着信タイク設定:

☐ 接続テスト(ALOHA)
応答文字列:

実行 閉じる

図 1 VC++版サンプルアプリケーション

No	項目	内容
1	受信イベント(1)/(2)	<p>発 ID 情報等のアロハからの受信イベントの内容を表示する。</p> <p>2 ポート分のデータを表示する。</p> <p>ポート…イベントを受信したポート(0~7)</p> <p>種類…発 ID 情報/フック情報/未応答情報/着信情報</p> <p>パラメーター(1),(2)…</p> <p>1) 発 ID 情報 パラメーター(1): 着信日時 /パラメーター(2): 発信電話番号</p> <p>2) フック情報 ※アロハ ND3/ND6 のみの機能 パラメーター(1): 日時 /パラメーター(2): フック情報(OF/ON)</p> <p>3) 未応答情報 ※アロハ ND3/ND6 のみの機能 パラメーター(1): 日時 /パラメーター(2): UR(未応答)</p> <p>4) 着信情報 ※アロハ ND3 のみの機能 パラメーター(1): 着信日時(PC より) /パラメーター(2): RC(着信)</p>
2	ポート	コマンドを送信する機器番号を指定する。(0~7)
3	機能選択	実行する機能(コマンド)を選択する。
4	番地	「発 ID メモリ情報読込」、「発 ID+オフ/未応答メモリ情報読込」を実行する際に、読み込み対象とするメモリ番地を指定する(1~30)
5	「発 ID メモリ情報読込」、「発 ID+オフ/未応答メモリ情報読込」結果	<p>「発 ID メモリ情報読込」、「発 ID+オフ/未応答メモリ情報読込」を実行した結果を表示する。</p> <p>※「発 ID+オフ/未応答メモリ情報読込」はアロハ ND3/ND6 のみの機能</p>
6	設定する管理コード	<p>「管理コード設定」を実行する際に、管理コード(16 桁)として設定する内容を指定する。</p> <p>※アロハ ND3/ND6 のみの機能</p>
7	管理コード読み込み結果	<p>「管理コード読込」を実行した結果を表示する。</p> <p>※アロハ ND3/ND6 のみの機能</p>
8	一次応答設定	<p>「一次応答設定」を実行する際に、一次応答をする設定にするのか一次応答をしない設定にするのかをナンバーディスプレイの有無にあわせて指定する。</p> <p>※アロハ ND5/ND6 のみの機能</p>
9	着信タイプ設定	<p>「着信タイプ設定」を実行する際に、着信拒否モードと着信許可モードのどちらに設定するのかを指定する。</p> <p>※アロハ ND6 のみの機能</p>
10	拒否設定/許可設定	<p>「着信拒否/着信許可番号設定」を実行する際に、⑪に入力された電話番号を、着信拒否と着信許可のどちらに設定するのかを指定する。</p> <p>※アロハ ND6 のみの機能</p>
11	電話番号入力欄	<p>「着信拒否/着信許可番号設定」を実行する際に、設定する電話番号を入力する。拒否/許可のどちらに設定するかは⑩で指定する。</p> <p>1 件目(左上)から詰めて入力する。Tab キーで次の欄に移動可能。</p> <p>20 桁までの半角数字か、1 文字の「P」(非通知)、「C」(公衆電話)、「O」(表示圏外) のいずれかのみ入力可能。</p> <p>空白のまま実行すると、登録してあった電話番号が削除される。</p> <p>※アロハ ND6 のみの機能</p>
12	番号リスト表示	<p>「着信拒否/着信許可番号 30 件読込」を実行した結果を表示する。</p> <p>※アロハ ND6 のみの機能</p>
13	一次応答設定読み込み結果	<p>「設定状況読込」を実行した際に、一次応答設定の結果を表示する。</p> <p>一次応答する場合はナンバーディスプレイが「なし」、一次応答しない場合はナンバーディスプレイが「あり」。</p> <p>※アロハ ND5/ND6 のみの機能</p>

14	着信タイプ設定読み込み結果	「設定状況読込」を実行した際に、着信タイプ設定の結果を表示する。 ※アロハ ND6 のみの機能
15	接続テスト結果	「接続テスト」を実行した結果を表示する。「ALOHA」と出力される。接続できない場合は、何も表示されない状態になる。 ※アロハ ND5/ND6 のみの機能
16	接続状態表示	接続状態を表示する。
17	実行ボタン	選択された機能を実行する。
18	閉じるボタン	アプリケーションを終了する。 終了時には、接続済みのアロハから自動的に切断する。

7.1.3 使用方法

- 上図②欄にコマンド送信先となるポートを指定する。(0～7)
- 上図③のラジオボタンから実行する機能を選択する。
- 事前に入力しておくべき項目がある場合は、入力する。
 - 発 ID メモリ情報読込 ④
 - 発 ID+オフ/未応答メモリ情報読込 ④
 - 管理コード設定 ⑥
 - 一次応答設定 ⑧
 - 着信タイプ設定 ⑨
 - 着信拒否/着信許可番号設定 ⑩⑪
- 上図⑦の「実行」ボタンをクリックする。
- 出力される内容がある場合は、出力される。
 - 発 ID メモリ情報読込 ⑤
 - 発 ID+オフ/未応答メモリ情報読込 ⑤
 - 管理コード読込 ⑦
 - 着信拒否/着信許可番号 30 件読込 ⑫
 - 設定状況読込 ⑬⑭
 - 接続テスト ⑮
- 以上を繰り返す。
- 上図⑮の「閉じる」ボタンを押すと、アプリケーションを終了する。

※ 接続されているアロハの状態は上図⑮に表示されている。

※ アプリ起動中に着信等のイベントを受信した場合には、上図①欄にイベント内容が表示される。

※ USB へアロハを挿抜すると、自動的に接続/切断し、結果をメッセージボックスで表示する。

ただし Windows8 では、USB からアロハを取り外しても切断されず、メッセージボックスも表示されない。その後、USB にアロハを挿入した際に、取り外されているアロハが切断される(切断のメッセージボックスは表示されない)。

以上