

PM250 UnitAPI SDK

コマンドリファレンス・マニュアル

Ver 1.3



ここに含まれる情報は、浜松東亜電機株式会社が独占的な権利を持ち、浜松東亜電機株式会社の書面による事前の許諾を得ずに全部もしくは一部を配布・複製または開示してはならないものとする。

目次

1 ドキュメントについて	4
1.1 改定履歴	4
2 序章	5
2.1 関数の説明	5
2.2 モジュールのサンプル	6
3. 各機能詳細	7
BTTurnon	7
WLANTurnon	7
ScannerTurnon	7
VibratorTurnon	8
SetBacklightValues	8
GetBacklightValues	8
BacklightTurnOn	9
UseBackLightAlways	9
GetSuspendTimeout	9
SetSuspendTimeout	10
EnablePowerButton	10
ScanLEDTurnOn	11
WarmBoot	11
HideTaskBar	11
GetModelName	12
SetIPMMode	12
GetIPMMode	13
GetSystemInfo	13
DisplayRotate	14
GetSystemInfoEx	15
ボタン再配置	17
GetScannerType	19
GetBTData	19
GetBootLoaderVerMajor	20
GetBootLoaderVerMinor	20
GetBootLoaderVerSuffix	20
GetBootLoaderVerDate	21
GetBootLoaderVerTime	21
GetKernelVerMajor	22

GetKernelVerMinor	22
GetKernelVerSuffix.....	22
GetKernelVerDate.....	23
GetKernelVerTime	23
RebootSystem	23
GetSbootVerMajor	24
GetSbootVerMinor	24
GetSbootVerSuffix.....	25
GetSbootVerDate.....	25
GetSbootVerTime	26
QueryRadioType	26
GetUnitApiVer	27
TouchPannelControl	27
VibratorOn_wDelay.....	27

1 ドキュメントについて

このドキュメントは、PM250 (PM250TA/PM250TAH) の UnitAPI に関する SDK (コマンドリファレンスマニュアル) です。

OS イメージ版数 (DLL バージョン) と UnitAPI SDK との対応を下表に示します。

このドキュメントの内容は、下表の最新版数に対応しています。

(版数・バージョンの確認方法は、PM250 ユーザーマニュアルを参照して下さい。)

OS イメージ 版数	(unitapi.dll バージョン)	対応 UnitAPI SDK
25.04 (Sep 01 2009)	V1.05.00	V1.4.zip
25.05 (Nov 12 2009) 以降	V1.07.00	V1.71.zip

SDK は、本書以外に下記のものがあります。

- ・PM250TA (1 次元モデル) スキャナ SDK (コマンドリファレンスマニュアル)
- ・PM250TAH (2 次元モデル) スキャナ SDK (コマンドリファレンスマニュアル)

1.1 改定履歴

バージョン	日付	説明	作者
1.0	2009 年 10 月 1 日	1. 日本語版 初版発行	浜松東亜電機株式会社
1.1	2009 年 11 月 18 日	1. OS (V25.05) に対応	浜松東亜電機株式会社
1.2	2010 年 8 月 31 日	1. OS イメージとの対応について追記 2. GetSystemInfoEX 関数の説明に追記	浜松東亜電機株式会社
1.3	2010 年 9 月 8 日	1. OS イメージとの対応について変更 ・UnitControl.h の修正に伴い SDK を再構成した	浜松東亜電機株式会社

2 序章

基本関数 UnitAPI は PDA のいくつかのユニット・コントロールを可能にします。この関数を使用することにより、電源の ON/OFF や詳細な制御ができます。

一般に、DLL ファイルは、多くの言語で書かれた他のモジュールにインポートするライブラリです。

本書では、いくつかのサンプルで Unitapi.dll の使用方法、PDA の制御にどの関数を使用すべきかを示します。

2.1 関数の説明.

説明	関数型	関数名
Bluetooth ON/OFF	typedef void (WINAPI *PFN_BTTURNON)(BOOL);	BTTurnon
無線 LAN ON/OFF	typedef void (WINAPI *PFN_WLANTURNON)(BOOL);	WLANTurnon
バックライトコントラスト制御	typedef void (WINAPI *PFN_SETBACKLIGHTVALUE)(PBYTE);	SetBackLightValues
スキャナー ON/OFF	typedef void (WINAPI *PFN_SCANNERTURNON)(BOOL);	ScannerTurnon
バイブレータ ON/OFF	typedef void (WINAPI *PFN_VIBRATORTURNON)(BOOL);	VibratorTurnon
バックライトコントラスト値取得	typedef void (WINAPI *PFN_GETBACKLIGHTVALUE)(PBYTE);	GetBackLightValues
バックライト ON/OFF	typedef void (WINAPI *PFN_BACKLIGHTTURNON)(bool);	BackLightTurnOn
バックライト 常時 ON	typedef void (WINAPI *PFN_USEBACKLIGHTALWAYS)(bool);	UseBackLightAlways
サスペンド時間取得	typedef void (WINAPI *PFN_GETSUSPENDTIME)(DWORD *,DWORD *);	GetSuspendTimeout
サスペンド時間設定	typedef void (WINAPI *PFN_SETSUSPENDTIME)(DWORD *,DWORD *);	SetSuspendTimeout
電源ボタン有効/無効	typedef void (WINAPI *PFN_ENABLEPOWERBUTTON)(bool);	EnablePowerButton
スキャン LED ON/OFF	typedef void (WINAPI *PFN_SCANLEDTURNON)(BYTE,BYTE);	ScanLEDTurnOn
ウォームブート(リセット)	typedef bool (WINAPI *PFN_WARMBOOT)();	WarmBoot
タスクバー非表示	typedef bool (WINAPI *PFN_HIDETASKBAR)(bool);	HideTaskBar
モデル名取得	typedef DWORD (WINAPI *PFN_GETMODELNAME)(WCHAR* pName,DWORD dwNameLen);	GetModelName
CPU 電力モード変更	typedef void (WINAPI *PFN_SETIPMMODE)(IPM_PRODUCT_OP ipm_mode);	SetIPMMode
CPU 電力モード取得	typedef void (WINAPI *PFN_GETIPMMODE)(IPM_PRODUCT_OP *ipm_mode);	GetIPMMode
システム情報取得	typedef BOOL (WINAPI *PFN_GETSYSTEMINFO)(SYSINFO_T sys_info,TCHAR* pszBuffer);	GetSystemInfo
ディスプレイ回転	typedef BOOL (WINAPI *PFN_DISPLAYROTATE)(DISPLAY_ANGLE_T angle);	DisplayRotate
システム情報取得 (ドライバ、OSを含む)	typedef BOOL (WINAPI *PFN_GETSYSTEMINFOEX)(SYSINFOVALUE* pSysInfo);	GetSystemInfoEx
ボタンマッピング	Can update by modifying registry key	
スキャナータイプ取得	typedef BOOL (WINAPI *PFN_GetScannerType)(DWORD*);	GetScannerType
Bluetooth データ取得	typedef BOOL (WINAPI *PFN_GetBTData)(CSR_NVRam*);	GetBTData
Boot ロダーメジャーバージョン取得	typedef BOOL (WINAPI *PFN_GetBootLoaderVerMajor)(DWORD*);	GetBootLoaderVerMajor
Boot ロダーマイナーバージョン取得	typedef BOOL (WINAPI *PFN_GetBootLoaderVerMinor)(DWORD*);	GetBootLoaderVerMinor
Boot ロダーサフィックス取得	typedef BOOL (WINAPI *PFN_GetBootLoaderVerSuffix)(wchar_t*,int);	GetBootLoaderVerSuffix
Boot ロダービルド日付取得	typedef BOOL (WINAPI *PFN_GetBootLoaderVerDate)(wchar_t*,int);	GetBootLoaderVerDate
Boot ロダービルド時刻取得	typedef BOOL (WINAPI *PFN_GetBootLoaderVerTime)(wchar_t*,int);	GetBootLoaderVerTime
カーネルメジャーバージョン取得	typedef BOOL (WINAPI *PFN_GetKernelVerMajor)(DWORD*);	GetKernelVerMajor

カーネルマイナーバージョン取得	typedef BOOL (WINAPI *PFN_GetKernelVerMinor)(DWORD*);	GetKernelVerMinor
カーネルサフィックス取得	typedef BOOL (WINAPI *PFN_GetKernelVerSuffix)(wchar_t*, int);	GetKernelVerSuffix
カーネルビルド日付取得	typedef BOOL (WINAPI *PFN_GetKernelVerDate)(wchar_t*, int);	GetKernelVerDate
カーネルビルド時刻取得	typedef BOOL (WINAPI *PFN_GetKernelVerTime)(wchar_t*, int);	GetKernelVerTime
リブートシステム取得	typedef BOOL (WINAPI *PFN_RebootSystem)();	RebootSystem
Sboot メジャーバージョン取得	typedef BOOL (WINAPI *PFN_GetSbootVerMajor)(DWORD*);	GetSbootVerMajor
Sboot マイナーバージョン取得	typedef BOOL (WINAPI *PFN_GetSbootVerMinor)(DWORD*);	GetSbootVerMinor
Sboot サフィックス取得	typedef BOOL (WINAPI *PFN_GetSbootVerSuffix)(wchar_t*, int);	GetSbootVerSuffix
Sboot ビルド日付取得	typedef BOOL (WINAPI *PFN_GetSbootVerDate)(wchar_t*, int);	GetSbootVerDate
Sboot ビルド時刻取得	typedef BOOL (WINAPI *PFN_GetSbootVerTime)(wchar_t*, int);	GetSbootVerTime
無線モジュール情報取得	typedef BOOL (WINAPI *PFN_QueryRadioType)(DWORD, DWORD*);	QueryRadioType
UnitApi dll バージョン取得	typedef BOOL (WINAPI *PFN_GetUnitApiVer)(wchar_t*);	GetUnitApiVer
タッチパネル制御	typedef BOOL (WINAPI *PFN_TouchPanelControl)(BOOL);	TouchPanelControl
バイブレータ遅延起動	typedef BOOL (WINAPI *PFN_VibratorOn_wDelay)(DWORD);	VibratorOn_wDelay

2.2 モジュールのサンプル..

```

HINSTANCE hInstance;
PFN_WLANTURNON pBTTurnOn;
HINSTANCE hInstance = LoadLibrary (_T( " unitapi.dll " ));
If(hInstance)
pBTTurnOn=(PFN_WLANTURNON)GetProcAddress(hInstance, L"BTTurnon");
// IF you want turn on the Bluetooth
pBTTurnOn(TRUE);
FreeLibrary(hInstance);

```

3. 各機能詳細..

BTTurnon

Bluetooth のパワーを制御します。

void BTTurnon(BOOL bON)

パラメータ

bON

TRUE は bluetooth 機能 ON を示します。.

復帰値

なし

WLanTurnon

無線 LAN のパワーを制御します。

void WLanTurnon(BOOL bON)

パラメータ

bON

TRUE は無線 LAN 機能 ON を示します。.

復帰値

なし

ScannerTurnon

スキャナーのパワーを制御します。

void ScannerTurnon(BOOL bON)

パラメータ

bON

TRUE はスキャナー機能 ON を示します。.

復帰値

なし

VibratorTurnon

バイブレータのパワーを制御します。

```
void VibratorTurnon(BOOL bON)
```

パラメータ

bON

TRUE はバイブレータ機能 ON を示します。

復帰値

なし

SetBacklightValues

バックライトの輝度を設定します。

```
void SetBacklightValues( PBYTE pValues)
```

パラメータ

pValues

pValues は 2 バイト確保する必要があります。

pValue[0] : AC コントラスト値 (範囲 0~100)

pValue[1] : バッテリコントラスト値 (範囲 0~100)

復帰値

なし

GetBacklightValues

現在のバックライト輝度を取得します。

```
void GetBacklightValues( PBYTE pValues)
```

パラメータ

pValues

pValues は 2 バイト確保する必要があります。

pValue[0] : AC コントラスト値 (範囲 0~100)

pValue[1] : バッテリコントラスト値 (範囲 0~100)

復帰値

なし

BacklightTurnOn

バックライトのパワーを制御します。

```
void BacklightTurnOn( BOOL bON)
```

パラメータ

bON

TRUE はバックライト機能 ON を示します。

復帰値

なし

UseBackLightAlways

バックライト常時 ON にするか否かを制御します。

```
void UseBackLightAlways( BOOL bON)
```

パラメータ

bON

TRUE はバックライト機能を OFF しません。

復帰値

なし

GetSuspendTimeout

デバイスがサスペンドモードに入るまでの時間を取得します。

```
void GetSuspendTimeout( DWORD *pdwBatteryPowerOffMin, DWORD *pdwExtPowerOffMin)
```

パラメータ

pdwBatteryPowerOffMin

バッテリー状態でのパワーOFF 時間を取得する変数(単位: 秒)

pdwExtPowerOffMin

AC 接続状態でのパワーOFF 時間を取得する変数(単位: 秒)

復帰値

なし

SetSuspendTimeout

サスペンドモードに入るまでの時間を設定します。

```
void SetSuspendTimeout( DWORD *pdwBatteryPowerOffMin, DWORD *pdwExtPowerOffMin)
```

パラメータ

pdwBatteryPowerOffMin

バッテリー状態でのパワーOFF 時間の値(単位:秒、60 の倍数で設定要)

本パラメータが NULL の場合、サスペンドモードになりません。

pdwExtPowerOffMin

AC 接続状態でのパワーOFF 時間の値(単位:秒、60 の倍数で設定要)

本パラメータが NULL の場合、サスペンドモードになりません。

注意事項

コントロールパネルの設定に同期させるために下記の値のみ使用可能。

電源状態	パラメータ値	実際の時間
バッテリー モード	60	1 分
	120	2 分
	180	3 分
	240	4 分
	300	5 分

電源状態	パラメータ値	実際の時間
外部電源 (AC)モード	60	1 分
	120	2 分
	300	5 分
	600	10 分
	900	15 分
	1800	30 分

復帰値

なし

EnablePowerButton

電源ボタンを使用するか否か制御します。

```
void EnablePowerButton(BOOL bON)
```

パラメータ

bON

TRUE は電源ボタンの有効を示します。

本パラメータが FALSE の場合、電源ボタンを使用してサスペンドモードにできません。

復帰値

なし

ScanLEDTurnOn

専用 LED を制御します。

```
void ScanLEDTurnOn(BYTE nLED, BYTE nStatus )
```

パラメータ

nLED

LED の色を示します。

LED_RED(0x0) : 赤LED.

LED_GREEN(0x1) : 緑LED.

LED_BLUE(0x2) : 青 LED.

nStatus

LED の状態を制御します。

LED_ON : nLED を ON します。

LED_OFF : nLED を OFF します。

LED_BLINK : nLED を 1 回点滅します。

復帰値

なし

WarmBoot

デバイスをソフトウェアリセットできます。

```
void WarmBoot()
```

パラメータ

なし

復帰値

なし

HideTaskBar

タスクバーのオプションを制御します。

```
void HideTaskBar( BOOL bHide )
```

パラメータ

bHide : TRUE はタスクバーの自動非表示機能を有効にします。

復帰値

なし

GetModelName

モデル名を取得します。

```
void GetModelName( WCHAR* pName,DWORD dwNameLen)
```

パラメータ

pName：モデル名を取得する変数。

dwNameLen：pName バッファの最大サイズ。

本値が24以上の場合、OEM 名を返します。しかし24に満たない場合はプラットフォーム名を返します。

復帰値

なし

SetIPMMode

CPU パワーモードを設定します。

```
void SetIPMMode( IPM_PRODUCT_OP IPM_mode)
```

パラメータ

IPM_mode

PDA はアイドル時間中にイベントが発生しなければ、

CPU は自動的に下位のパワーモードに変更されます。

しかし、本機能では下位パワーモードに入らないように設定できます。

IPM_OP_D0CS	CPU Lowest power mode (52MHz)
IPM_OP_104MHz	CPU 104MHz Mode
IPM_OP_416MHz	CPU 416MHz Mode
IPM_OP_624MHz	CPU 624MHz Mode

復帰値

なし

GetIPMMode

CPU パワーモードを取得します。

```
void GetIPMMode( IPM_PRODUCT_OP* IPM_mode)
```

パラメータ

IPM_mode

PDA はアイドル時間中にイベントが発生しなければ、

CPU は自動的に下位のパワーモードに変更されます。

本関数では現在の CPU の IPM モードを取得できます。

IPM_OP_D0CS	CPU Lowest power mode (52MHz)
IPM_OP_104MHz	CPU 104MHz Mode
IPM_OP_416MHz	CPU 416MHz Mode
IPM_OP_624MHz	CPU 624MHz Mode

復帰値

なし

GetSystemInfo

システムのすべての情報を取得します。

```
void GetSystemInfo( SYSINFO_T sys_info,TCHAR* pszBuffer)
```

パラメータ

sys_info : どのシステム情報を取得するかを指定します。

SYSINFO_T メンバ	説明
SYSINFO_MODELNAME	デバイスモジュール名
SYSINFO_BOOTVER	Boot バージョン
SYSINFO_OSVER	OS バージョン
SYSINFO_SERIALNO	シリアルナンバー
SYSINFO_WLAN_MODULE_NAME	無線 LAN モジュール名
SYSINFO_WLAN_DRIVER_VER	無線 LAN デバイスバージョン
SYSINFO_WLAN_FIRMWARE_VER	無線 LAN ファームウェアバージョン
SYSINFO_WLAN_MAC_ADDRESS	無線 LAN MAC アドレス
SYSINFO_BT_MODULE_NAME	Bluetooth モジュール名
SYSINFO_BT_MAC_ADDRESS	Bluetooth MAC アドレス
SYSINFO_SCANNER_MODULE_NAME	スキャナーモジュール名

PszBuffer : システム情報を得るためのバッファ、50 バイト以上なければなりません。

復帰値

なし

DisplayRotate

ディスプレイ表示角度を指定できます。

```
void DisplayRotate( DISPLAY_ANGLE_T angle)
```

パラメータ

angle :

DISPLAY_ANGLE_T メンバ	説明
ANGLE_0	0 度回転
ANGLE_90	90 度回転
ANGLE_180	180 度回転
ANGLE_270	270 度回転

復帰値

なし

GetSystemInfoEx

システムのすべての情報を取得します。

```
void GetSystemInfoEx( SYSINFOVALUE* pSysInfo)
```

パラメータ

pSysInfo : システム情報構造体

Types	SYSINFOVALUE メンバ	説明
DATETIME	os_datetime	OS ビルド日時
DATETIME	boot_datetime	BOOT ビルド日時
HW_DEVICE_VERSION	hw_revision	ボードバージョン
Char*	info	詳細バージョンと各ドライブの情報

DATETIME 構造体

Types	DATETIME メンバ	説明
DWORD	dwYear	年
DWORD	dwMonth	月
DWORD	dwDay	日
DWORD	dwHour	時
DWORD	dwMinute	分
DWORD	dwSecond	秒

HW_DEVICE_VERSION 構造体

Types	HW_DEVICE_VERSION メンバ	説明
BYTE	MODEL	現在の PCB のモデル D6100/PM250: 0 D6500/PM350: 1
BYTE	STATE	現在の PCB イベントバージョン ES : 0 EVT : 0 DVT : 1 MVT : 2 MP : 3
BYTE	REVISION	現在の PCB リビジョン ES の場合、0 固定 EVT の場合、1~7 その他の場合、1~8

info value : ソフトウェア情報を特定する値。1 バイト文字で構成。

SYSINFO_LIST メンバ	説明
SYSINFO_MajorVersion	メジャーバージョン
SYSINFO_KernelMinorVersion	カーネルマイナーバージョン
SYSINFO_EBootMinorVersion	Eboot マイナーバージョン
SYSINFO_SBootMinorVersion	Sboot マイナーバージョン
SYSINFO_Distribution	配布
SYSINFO_QFE	QFE
SYSINFO_MagicNumber	未使用 (SYS_INFO 構造検証用)
SYSINFO_ModelNumber	モデル番号
SYSINFO_SerialNumber	シリアルナンバー
SYSINFO_PartNumber	部品番号
SYSINFO_Manufacture	製造番号
SYSINFO_KeyboardType	キーボードタイプ
SYSINFO_TouchType	タッチタイプ
SYSINFO_DisplayType	ディスプレイタイプ
SYSINFO_BluetoothType	Bluetooth タイプ

SYSINFO_WLANType	無線 LAN タイプ
SYSINFO_WWANType	無線 WAN タイプ
SYSINFO_LaserNumber	レーザー番号

復帰値

なし

ボタン再配置

“UserAssignApplication”配下の[HKEY_CURRENT_USER¥ControlPanel¥Keybd¥KEY_NAME]を修正するとキーパッドボタンを再配置することができます。

例) “ESC”キーの場合

[HKEY_CURRENT_USER¥ControlPanel¥Keybd¥ESC]

“UserAssignApplication” = “” ;	初期値(ESC)機能に再配置します。
“UserAssignApplication” = “##” ;	機能なしに再配置します。
“UserAssignApplication” = “\$\$” ;	前方SCAN機能に再配置します。
“UserAssignApplication” = “%%” ;	左側SCAN key機能に再配置します。
“UserAssignApplication” = “^^” ;	右側SCAN key機能に再配置します。
“UserAssignApplication” = “¥¥Windows¥¥pWord.exe” ;	pWord.実行ファンクションキーに再配置します。

KEY_NAME	説明	機能
ESC	ESC key	デフォルト
BACKSPACE	Back space key	デフォルト
ENTER	Enter key	デフォルト
FRONT_SCAN	Front scan key	デフォルト
LEFT_SCAN	Left scan key	デフォルト
RIGHT_SCAN	Right scan key	デフォルト
REAR_SCAN	Rear scan key	デフォルト
LEFT	Left key	デフォルト
RIGHT	Right key	デフォルト
UP	Up key	デフォルト
DOWN	Down key	デフォルト
App1	F1 key	pWord.exe
App2	F2 key	pmail.exe
App3	F3 key	explorer.exe
App4	F4 key	iesample.exe
App5	F5 key	cmd.exe
App6	F6 key	control.exe
App7	F7 key	Transcriber.exe
WLAN	WiFi 機能制御 key	デフォルト
BRIGHT+	バックライト輝度制御 key	デフォルト
VOLUM+	音量制御 key	デフォルト
TAB	Tab key	デフォルト
BRIGHT-	バックライト輝度制御 key	デフォルト
VOLUM-	音量制御 key	デフォルト
WINDOW	Window key	デフォルト
DELETE	Delete key	デフォルト
CALIB	Touch calibration key	デフォルト

備考:レジストリキーを変更した後、即時に適用するために、名前付きイベント呼び出す必要があります。
イベント名は"UpdateUserActionTableEvent"で、次のように呼出さなければならない。

```
HANDLE hKeyAssignEvent;  
hKeyAssignEvent = CreateEvent(NULL, FALSE, FALSE,  
TEXT("UpdateUserActionTableEvent"));  
if(hKeyAssignEvent != NULL)  
{  
    SetEvent(hKeyAssignEvent);  
    CloseHandle(hKeyAssignEvent);  
}
```

GetScannerType

イメージスキャナのタイプコードを取得します。

BOOL GetScannerType(DWORD* pdwType)

パラメータ

pdwType : スキャナータイプを取得する変数。

復帰値

問題無い場合は、TRUE を返します。

備考

スキャナータイプ	リビジョン値	説明
不明	(-1)	
なし	0	
IT5100	8	-
IT5300	9	HHP 2D スキャナー

GetBTData

Bluetooth データを取得します。

BOOL GetBTData(CSR_NVRam* pbBtData)

パラメータ

pbBtData : Bluetooth データを取得する変数。

復帰値

問題無い場合は、TRUE を返します。

備考

```
Implement CSR_NVRam structure
typedef struct CSR_NVRam{
    union{
        struct{
            DWORD valid;           // "BTDO", when bt struct is initialized
            BYTE btaddress[8];     // Element 0x01
            BYTE btcountrycode;    // Element 0x02
        }; // anonymous struct
        char pad[256];
    }; // anonymous union
} CSR_NVRam;
```

GetBootLoaderVerMajor

Bootloader のメジャーファームウェアバージョンを取得します。

BOOL GetBootLoaderVerMajor(DWORD* pdwVersion)

パラメータ

pdwVersion : bootloader のメジャーバージョンを取得する変数。

復帰値

問題無い場合は、TRUE を返します。

備考

モデル	値
D6100	23
D6500	24
PM250	25
PM350	35

GetBootLoaderVerMinor

Bootloader のマイナーファームウェアバージョンを取得します。

BOOL GetBootLoaderVerMinor(DWORD* pdwVersion)

パラメータ

pdwVersion : bootloader のマイナーバージョンを取得する変数。

復帰値

問題無い場合は、TRUE を返します。

GetBootLoaderVerSuffix

Bootloader ファームウェアバージョンのサフィックス文字列を取得します。

BOOL GetBootLoaderVerSuffix(wchar_t* pszSuffix, int szLen)

パラメータ

pdwVersion : サフィックス文字バッファ(文字列 WCHAR[16])を指定します。

SzLen : サフィックス文字バッファのサイズ

復帰値

問題無い場合は、TRUE を返します。

備考

接尾文字列	目的
NB	Internal test
Tx	Test version
R1	RAM バージョン
(none)	MP

GetBootLoaderVerDate

Bootloader のビルド日付を取得します。

BOOL GetBootLoaderVerDate(wchar_t* pszDate, int szLen)

パラメータ

pszDate : Bootloader ビルド日付 (文字列 WCHAR[16]) を指定します。

SzLen : Bootloader ビルド日付のサイズ

復帰値

問題無い場合は、TRUE を返します。

備考

Bootloader ビルド日付のフォーマットは、“Jan 01 2005”を使用しています。

GetBootLoaderVerTime

Bootloader のビルド時刻を取得します。

BOOL GetBootLoaderVerTime(wchar_t* pszTime, int szLen)

パラメータ

pszTime : Bootloader ビルド時刻 (文字列 WCHAR[16]) を指定します。

SzLen : Bootloader ビルド時刻のサイズ

復帰値

問題無い場合は、TRUE を返します。

備考

Bootloader ビルド時刻のフォーマットは、“23:59:59”を使用しています。

GetKernelVerMajor

カーネルのメジャーファームウェアバージョンを取得します。

BOOL GetKernelVerMajor(DWORD* pdwVersion)

パラメータ

pdwVersion：カーネルのメジャーバージョンを取得する変数。

復帰値

問題無い場合は、TRUE を返します。

備考

モデル	値
D6100	23
D6500	24
PM250	25
PM350	35

GetKernelVerMinor

カーネルのマイナーファームウェアバージョンを取得します。

BOOL GetKernelVerMinor(DWORD* pdwVersion)

パラメータ

pdwVersion：カーネルのマイナーバージョンを取得する変数。

復帰値

問題無い場合は、TRUE を返します。

GetKernelVerSuffix

カーネルのファームウェアバージョンのサフィックス文字列を取得します。

BOOL GetKernelVerSuffix(wchar_t* pszSuffix, int szLen)

パラメータ

pdwVersion：サフィックス文字列バッファを取得する変数。(文字列 WCHAR[16])

szLen：サフィックス文字列バッファサイズ

復帰値

問題無い場合は、TRUE を返します。

備考

サフィックス文字列	目的
NB	Internal test
Tx	Test version
R1	RAM バージョン
(None)	MP

GetKernelVerDate

カーネルのビルド日付を取得します。

BOOL GetKernelVerDate(wchar_t* pszDate, int szLen)

パラメータ

pszDate : カーネルビルド日付を取得する変数。(文字列 WCHAR[16])

szLen : カーネルビルド日付のサイズ

復帰値

問題無い場合は、TRUE を返します。

備考

カーネルビルド日付のフォーマットは、“Jan 01 2005”を使用しています。

GetKernelVerTime

カーネルのビルド時刻を取得します。

BOOL GetKernelVerTime(wchar_t* pszTime, int szLen)

パラメータ

pszTime : カーネルビルド時刻を取得する変数。(文字列 WCHAR[16])

szLen : カーネルビルド時刻のサイズ

復帰値

問題無い場合は、TRUE を返します。

備考

カーネルビルド時刻のフォーマットは、“23:59:59”を使用しています。

RebootSystem

デバイスをウォームブートします。

BOOL RebootSystem()

パラメータ

なし

復帰値

なし

GetSbootVerMajor

Sboot のメジャーファームウェアバージョンを取得します。

BOOL GetSbootVerMajor(DWORD* pdwVersion)

パラメータ

pdwVersion : Sboot のメジャーバージョンを取得する変数。

復帰値

0 以外は成功を示します。

備考

モデル	値
D6100	23
D6500	24
PM250	25
PM350	35

GetSbootVerMinor

Sboot のマイナーファームウェアバージョンを取得します。

BOOL GetSbootVerMinor(DWORD* pdwVersion)

パラメータ

pdwVersion : Sboot のマイナーバージョンを取得する変数。

復帰値

問題無い場合は、TRUE を返します。

GetSbootVerSuffix

Sboot のファームウェアバージョンのサフィックス文字列を取得します。

BOOL GetSbootVerSuffix(wchar_t* pszSuffix, int szLen)

パラメータ

pdwVersion : サフィックス文字列バッファを指定します。(文字列 WCHAR[16])

szLen : サフィックス文字列バッファのサイズ

復帰値

0 以外は成功を示します。

備考

接尾文字列	目的
NB	Internal test
Tx	Test version
R1	RAM バージョン
(None)	MP

GetSbootVerDate

Sboot のビルド日付文字列を取得します。

BOOL GetSbootVerDate(wchar_t* pszDate, int szLen)

パラメータ

pszDate : Sboot のビルド日付を取得する変数。(文字列 WCHAR[16])

szLen : pszDate バッファのサイズ

復帰値

0 以外は成功を示します。

備考

Sboot ビルド日付のフォーマットは、“Jan 01 2005”を使用しています。

GetSbootVerTime

Sboot のビルド時刻文字列を取得します。

```
BOOL GetSbootVerTime(wchar_t* pszTime, int szLen)
```

パラメータ

pszTime : Sboot のビルド時刻を取得する変数。(文字列 WCHAR[16])

szLen : pszTime バッファのサイズ

復帰値

問題無い場合は、TRUE を返します。

備考

Sboot ビルド時刻のフォーマットは、“23:59:59”を使用しています。

QueryRadioType

無線ハードウェアの種類を列挙します。

```
BOOL QueryRadioType(DWORD mask, DWORD* pdwType)
```

パラメータ

mask : 取得する無線対応のマスク

pdwType : ラジオモジュールのタイプ

復帰値

0 以外は成功を示します。

備考

Mask type

Module	Mask
Bluetooth	0x01
WiFi	0x02
WAN(GSM)	0x04

Module type

WiFi	値
(None)	0
USI SD8686	3

Bluetooth	値
(None)	0
CSR Bluecore4Rom	3

WAN	値
(None)	0

GetUnitApiVer

UnitApi.dll のバージョンを返します。

BOOL GetUnitApiVer(wchar_t* pszVersion)

パラメータ

pszVersion : NULL を含む UnitApi.dll のバージョン

復帰値

0 以外は成功を示します。

TouchPannelControl

タッチパネルの有効／無効を制御します。

BOOL TouchPannelControl(BOOL bEnable)

パラメータ

bEnable : タッチパネルの有効 / 無効を指定します。

復帰値

0 以外は成功を示します。

VibratorOn_wDelay

指定された時間バイブレータを ON します。

BOOL VibratorOn_wDelay(DWORD ms)

パラメータ

ms :

=1 : 100ミリ秒 ON

=2 : 200ミリ秒 ON

=3 : 300ミリ秒 ON

復帰値

0 以外は成功を示します。