

# *H-15BJ* シリーズ

*HANDHELD TERMINAL*

---

## システム制御ライブラリ仕様書

株式会社 オプトエレクトロニクス

## 變更履歷

製品名 : H-15BJ シリーズ

仕様書名： H-15BJ システム制御ライブラリ仕様書

発行番号： Rev. 1.1

管理番号： SI14044

[illegible]

## 目 次

1. はじめに.....	1
1-1. 対象者 .....	1
1-2. 関連資料.....	1
1-3. 注意事項.....	1
2. 概要 .....	2
2-1. ライブラリ .....	2
2-2. 定義.....	3
2-3. ファイル構成.....	3
3. 開発環境構築.....	4
4. システム制御ライブラリ .....	5
4-1. DEVICE INFORMATION API.....	5
4-1-1. <i>SysGetSerialNumber</i> .....	6
4-1-2. <i>SysGetConfigNumber</i> .....	7
4-1-3. <i>SysGetModelName</i> .....	8
4-2. KEY API.....	9
4-2-1. <i>SysSetFunctionKeyPrograms</i> .....	10
4-2-2. <i>SysGetFunctionKeyPrograms</i> .....	11
4-2-3. <i>SysSetFunctionKeySettingState</i> .....	12
4-2-4. <i>SysGetFunctionKeySettingState</i> .....	13
4-2-5. <i>SysSetAlternateKeyCode</i> .....	14
4-2-6. <i>SysGetAlternateKeyCode</i> .....	15
4-3. POWER MANAGEMENT API.....	16
4-3-1. <i>SysSetSuspendTime</i> .....	17
4-3-2. <i>SysGetSuspendTime</i> .....	18
4-3-3. <i>SysSetPowerDownDisableMode</i> .....	19
4-3-4. <i>SysGetPowerDownDisableMode</i> .....	20
4-3-5. <i>SysPowerDown</i> .....	21
4-3-6. <i>SysSoftReset</i> .....	22
4-3-7. <i>SysSetWakeSources</i> .....	23
4-3-8. <i>SysGetWakeSources</i> .....	24
4-4. BACKLIGHT AND TOUCHPANEL API .....	25
4-4-1. <i>SysSetBacklightLevel</i> .....	26
4-4-2. <i>SysGetBacklightLevel</i> .....	27

---

4-4-3. <i>SysSetBacklightTimeout</i> .....	28
4-4-4. <i>SysGetBacklightTimeout</i> .....	29
4-4-5. <i>SysSetTouchPanelState</i> .....	30
4-4-6. <i>SysGetTouchPanelState</i> .....	31
4-5. WIRELESS LAN API .....	32
4-5-1. <i>SysSetWLANPower</i> .....	33
4-5-2. <i>SysGetWLANPower</i> .....	34
4-5-3. <i>SysGetWLANConnectStatus</i> .....	35
4-5-4. <i>SysGetWLANSignalLevel</i> .....	36
4-6. BLUETOOTH API .....	37
4-6-1. <i>SysSetBluetoothPower</i> .....	38
4-6-2. <i>SysGetBluetoothPower</i> .....	39
4-7. VIBRATOR API .....	40
4-7-1. <i>SysSetVibrator</i> .....	41
4-7-2. <i>SysStopVibrator</i> .....	42
4-8. LED API .....	43
4-8-1. <i>SysSetLED</i> .....	44
4-9. OTHER API .....	45
4-9-1. <i>SysGetLibraryVersion</i> .....	46
付録 A：キー配置と仮想キーコード .....	47
付録 B：ファンクションキーおよびショートカットキー .....	49

## 図 目 次

図 1 : キーパッドレイアウト .....	47
------------------------	----

## 表 目 次

表 1	: ライブラリ機能一覧.....	2
表 2	: ファイルの構成.....	3
表 3	: Device Information API 関数一覧.....	5
表 4	: Key API 関数一覧.....	9
表 5	: SysSetAlternateKeyCode のパラメータ.....	14
表 6	: Power Management API 関数一覧.....	16
表 7	: SysSetWakeSources のパラメータ.....	23
表 8	: Backlight and TouchPanel API 関数一覧.....	25
表 9	: Wireless LAN API 関数一覧.....	32
表 10	: Bluetooth API 関数一覧.....	37
表 11	: Vibrator API 関数一覧.....	40
表 11	: Vibrator API 関数一覧.....	43
表 12	: Other API 関数一覧.....	45
表 13	: 仮想キーコード.....	48
表 14	: ファンクションキーおよびショートカットキー.....	49

## 1. はじめに

本書は、デバイスのシステム関係を制御するライブラリ仕様書です。提供する API に含まれる関数のパラメータや戻り値などの詳細情報を提示します。

### 1-1. 対象者

本書は、H-15BJ シリーズのソフトウェアの開発者を対象としています。  
また、C および C++ 言語の文法に関する知識を持っていることを前提としています。

### 1-2. 関連資料

H-15BJ シリーズに関連するライブラリは大別して 2 種類、本書に記述するシステム制御関係と、バーコードリーダー制御関係があります。また、C# や VB.NET を使用する .NET 環境でのプログラミングに対応したクラスライブラリも提供しています。

- ・ H-15BJ システム制御ライブラリ仕様書（本書）
- ・ H-15BJ バーコードリーダー制御ライブラリ仕様書
- ・ H-15BJ システム制御クラスライブラリ仕様書
- ・ H-15BJ バーコードリーダー制御クラスライブラリ仕様書

### 1-3. 注意事項

- ・ 本書にある「バッテリー」とは H-15BJ の主電池「リチウムイオン充電池」を示しております。

## 2. 概要

### 2-1. ライブラリ

本ライブラリに含まれる機能を以下の表に示します。

表 1 : ライブラリ機能一覧

分類	機能	備考
Device Information API	デバイス型番の取得	
	デバイスシリアル番号の取得	
	構成番号の取得	
Key API	ファンクションキーの設定／取得	
	ファンクションキーの使用許可設定／取得	
	仮想キーコードの切り替え	
Power management API	自動電源オフ時間の設定／取得	
	電源キー未使用モードの設定／取得	
	電源オフ（サスペンド）	
	ウェイクアップ要因の設定／取得	
Backlight API	バックライトレベルの設定／取得	
	バックライトタイムアウト時間の設定／取得	
Wireless LAN API	無線 LAN 電源の設定／取得	
	無線 LAN 規制の設定／取得	
Bluetooth API	Bluetooth 電源の設定／取得	
Vibrator API	バイブレータの時間指定振動	
	バイブレータの停止	
LED API	LED の制御	
Others	システム制御ライブラリのバージョン情報取得	



## 2-2. 定義

キーボード入力に対してキーコードが設定されております。詳細は付録をご参照ください。

- ・ 仮想キーコード
- ・ ファンクションキー
- ・ ショートカットキー

## 2-3. ファイル構成

表 2 : ファイルの構成

ファイル名	解説	対象	提供方法	備考
sysctl.h	システム制御用ヘッダファイルです。	PC(開発環境)	開発キット	
sysctl.dll	システム制御用 DLL ファイルです。	HHT	標準実装	
sysctl.lib	システム制御用ライブラリファイルです。	PC(開発環境)	開発キット	

上記の表で HHT は H-15BJ 本体を示しております。

### 3. 開発環境構築

アプリケーションを開発するためには、ご使用の PC 上に適切な環境を整える必要があります。

環境の構築、また、本ライブラリを使用する方法については「H-15BJ アプリケーション開発キット」に添付されている「H-15BJ アプリケーション開発ガイド」に詳しく記述されていますので、そちらをご参照ください。

## 4. システム制御ライブラリ

本ライブラリで提供する API 関数について紹介します。

### 4-1. Device Information API

表 3 : Device Information API 関数一覧

関数名	解説	備考
SysGetSerialNumber	シリアル番号を取得します。	
SysGetConfigNumber	構成番号を取得します。	
SysGetModelName	デバイスの型番を取得します。	

#### 4-1-1. SysGetSerialNumber

デバイスのシリアル番号を取得します。

##### 構文

```
BOOL SysGetSerialNumber(  
    LPTSTR lpszSerialNumber  
);
```

##### パラメータ

lpszSerialNumber  
[out] シリアル番号を示す文字列へのポインタです。

##### 戻り値

関数が成功したとき、TRUE を返します。  
関数が失敗したとき、FALSE を返します。

##### 解説

文字列の領域はアプリケーションによって割り当てられます。

##### 対応情報

ヘッダファイル: sysctl.h  
インポートライブラリ: sysctl.lib

#### 4-1-2. SysGetConfigNumber

構成番号を取得します。

##### 構文

```
BOOL SysGetConfigNumber(  
    LPTSTR lpszConfigNumber  
);
```

##### パラメータ

*lpszConfigNumber*  
[out] 構成番号を示す文字列へのポインタです。

##### 戻り値

関数が成功したとき、TRUE を返します。  
関数が失敗したとき、FALSE を返します。

##### 解説

文字列の領域はアプリケーションによって割り当てられます。

##### 対応情報

ヘッダファイル: sysctl.h  
インポートライブラリ: sysctl.lib

### 4-1-3. SysGetModelName

デバイスの型番を取得します。

#### 構文

```
BOOL SysGetModelName(  
    LPTSTR lpszModelName  
);
```

#### パラメータ

*lpszModelName*

[out] デバイスの型番を示す文字列へのポインタです。

#### 戻り値

関数が成功したとき、TRUE を返します。

関数が失敗したとき、FALSE を返します。

#### 解説

なし

#### 対応情報

ヘッダファイル: sysctl.h

インポートライブラリ: sysctl.lib

## 4-2. Key API

表 4 : Key API 関数一覧

関数名	解説	備考
SysGetFunctionKeyPrograms	ファンクションキーおよびショートカットキーの割り当ての情報を取得します。	
SysSetFunctionKeyPrograms	ファンクションキーおよびショートカットキーに割り当てるプログラムを設定します。	
SysGetFunctionKeySettingState	ファンクションキーおよびショートカットキー使用有無の状態を取得します。	
SysSetFunctionKeySettingState	ファンクションキーおよびショートカットキーを使用するかどうかを設定します。	
SysSetAlternateKeyCode	センタキー（アクションキー）および ENT キーに割り当てる仮想キーコードを切り替えます。	
SysGetAlternateKeyCode	センタキー（アクションキー）および ENT キーに割り当てられている仮想キーコードについての情報を取得します。	

### 4-2-1. SysSetFunctionKeyPrograms

あるファンクションキーあるいはショートカットキーによって実行されるプログラム、または事前定義関数を設定します。

#### 構文

```
BOOL SysSetFunctionKeyPrograms(  
    DWORD dwFuncKey,  
    LPCTSTR lpszAppName,  
    LPCTSTR lpszParameters  
);
```

#### パラメータ

*dwFuncKey*

[in] ファンクションキーあるいはショートカットキーのインデックスです。  
詳細は「ファンクションキーおよびショートカットキー」をご参照ください。

*lpszAppName*

[in] 割り当てるプログラムのパス、または事前定義関数を示す文字列へのポインタです。NULL に設定することが可能です。

*lpszParameters*

[in] 実行するプログラム（コマンドライン）のパラメータを示す文字列へのポインタです。

#### 戻り値

関数が成功したとき、TRUE を返します。  
関数が失敗したとき、FALSE を返します。

#### 解説

使用可能な事前定義関数を以下に示します。

FN_CUT	// カット (Ctrl+X)
FN_COPY	// コピー (Ctrl+C)
FN_PASTE	// 貼り付け (Ctrl+V)
FN_VOLUME_UP	// 音量上昇
FN_VOLUME_DOWN	// 音量低下
FN_BACKLIGHT_UP	// バックライト輝度上昇
FN_BACKLIGHT_DOWN	// バックライト輝度低下

それぞれの関数はヘッダファイル sysctl.h に定義されます。

#### 対応情報

ヘッダファイル: sysctl.h

インポートライブラリ: sysctl.lib



## 4-2-2. SysGetFunctionKeyPrograms

あるファンクションキーあるいはショートカットキーに割り当てられたプログラム、または事前定義関数についての情報を取得します。

### 構文

```
BOOL SysGetFunctionKeyPrograms(  
    DWORD dwFuncKey,  
    LPCTSTR lpszAppName,  
    LPCTSTR lpszParameters  
);
```

### パラメータ

*dwFuncKey*

[in] ファンクションキーあるいはショートカットキーのインデックスです。

*lpszAppName*

[out] 割り当てるプログラムのパス、または事前定義関数を示す文字列へのポインタです。

*lpszParameters*

[out] 実行するプログラム（コマンドライン）のパラメータを示す文字列へのポインタです。

### 戻り値

関数が成功したとき、TRUE を返します。

関数が失敗したとき、FALSE を返します。

### 解説

SysSetFunctionKeyProgramsの解説をご参照ください。

### 対応情報

ヘッダファイル: sysctl.h

インポートライブラリ: sysctl.lib

### 4-2-3. SysSetFunctionKeySettingState

ファンクションキーおよびショートカットキーを有効にするか否かの設定をおこないます。

#### 構文

```
BOOL SysSetFunctionKeySettingState(  
    BOOL bEnable  
);
```

#### パラメータ

*bEnable*

[in] 有効無効を示すステータスです。TRUE のとき有効、FALSE のとき無効になります。

#### 戻り値

関数が成功したとき、TRUE を返します。

関数が失敗したとき、FALSE を返します。

#### 解説

なし

#### 対応情報

ヘッダファイル: sysctl.h

インポートライブラリ: sysctl.lib

#### 4-2-4. SysGetFunctionKeySettingState

ファンクションキーあるいはショートカットキーが有効であるか否かの状態を取得します。

##### 構文

```
BOOL SysGetFunctionKeySettingState(  
    PBOOL pEnable  
);
```

##### パラメータ

*pEnable*

[in] 有効無効を示すステータスへのポインタです。TRUE は有効、FALSE は無効を示します。

##### 戻り値

関数が成功したとき、TRUE を返します。

関数が失敗したとき、FALSE を返します。

##### 解説

なし

##### 対応情報

ヘッダファイル: sysctl.h

インポートライブラリ: sysctl.lib

#### 4-2-5. SysSetAlternateKeyCode

センタキー（アクションキー）および ENT キーの仮想キーコードを切り替えます。

##### 構文

```
BOOL SysSetAlternateKeyCode(  
    DWORD dwKey,  
    BOOL bAlternate  
);
```

##### パラメータ

*dwKey*

[in] 対象となるキーを示す変数です。値は下の表に示す定義値 *dwKey* のいずれかです。

表 5 : SysSetAlternateKeyCode のパラメータ

対象	dwKey	仮想キーコード	
		bAlternate: FALSE のとき	bAlternate: TRUE のとき
センタキー (アクションキー)	SYS_KEY_ACTION	VK_RETURN (0x0D)	0xEB
ENT キー	SYS_KEY_ENT	VK_RETURN (0x0D) SHIFT	0xEC

*bAlternate*

[in] 切り替え状況を示すブール値です。各設定の組合せによるキーコードの変化は上の表をご参照ください。

##### 戻り値

関数が成功したとき、TRUE を返します。

関数が失敗したとき、FALSE を返します。

##### 解説

センタキーをトリガキーとして用いる場合、切り替えの設定に関わらず仮想キーコードは 0xEA となります。

##### 対応情報

ヘッダファイル: sysctl.h

インポートライブラリ: sysctl.lib

#### 4-2-6. SysGetAlternateKeyCode

センタキー（アクションキー）および ENT キーの仮想キーコードの切り替え状況を取得します。

##### 構文

```
BOOL SysGetAlternateKeyCode(  
    DWORD dwKey,  
    PBOOL bAlternate  
);
```

##### パラメータ

*dwKey*

[in] 対象となるキーを示す変数です。値は以下に示す定義値のいずれかです。

SYS_KEY_ACTION	// センタキー（アクションキー）
SYS_KEY_ENT	// ENT キー

*bAlternate*

[out] 切り替え状況を示すブール値へのポインタです。

##### 戻り値

関数が成功したとき、TRUE を返します。

関数が失敗したとき、FALSE を返します。

##### 解説

各設定の組合せに関してはSysSetAlternateKeyCodeの頁をご参照ください。

##### 対応情報

ヘッダファイル: sysctl.h

インポートライブラリ: sysctl.lib

## 4-3. Power Management API

表 6 : Power Management API 関数一覧

関数名	解説	備考
SysSetSuspendTime	自動電源オフの時間を設定します。	
SysGetSuspendTime	自動電源オフ時間の現在値を取得します。	
SysSetPowerDownDisableMode	電源キーによる電源オフ無効モードを設定します。	
SysGetPowerDownDisableMode	電源キーによる電源オフ無効モードの状態を取得します。	
SysPowerDown	次に電源オンされたときの起動方法を指定して、電源オフします。	
SysSoftReset	デバイスをソフトリセットします。	
SysSetWakeSources	起動要因を設定します。	
SysGetWakeSources	起動要因の状態を取得します。	

### 4-3-1. SysSetSuspendTime

何の操作も行われないとき、自動で電源オフするまでの時間を設定します。バッテリー使用時、AC 電源使用時の 2 種類があります。

#### 構文

```
BOOL SysSetSuspendTime(  
    DWORD dwPowerMode,  
    DWORD dwSuspendTime  
);
```

#### パラメータ

##### *dwPowerMode*

[in] 電源モード（バッテリー、AC 電源）を示す変数です。値は以下に示す定義値のいずれかになります。

```
SYS_POWER_BATTERY_MODE    // バッテリー使用時  
SYS_POWER_EXTERNAL_MODE    // AC 電源使用時
```

##### *dwSuspendTime*

[in] 電源オフまでの時間を示す変数です。値は以下に示す定義値のいずれかになります。

```
SUSPEND_TIME_ONE_MIN  
SUSPEND_TIME_TWO_MIN  
SUSPEND_TIME_THREE_MIN    // バッテリー使用時のみ  
SUSPEND_TIME_FOUR_MIN     // バッテリー使用時のみ  
SUSPEND_TIME_FIVE_MIN  
  
SUSPEND_TIME_TEN_MIN  
SUSPEND_TIME_FIFTEEN_MIN  // AC 電源使用時のみ  
SUSPEND_TIME_THIRTY_MIN  
  
SUSPEND_TIME_INDEFINITELY // 自動電源オフなし
```

#### 戻り値

関数が成功したとき、TRUE を返します。  
関数が失敗したとき、FALSE を返します。

#### 解説

バッテリー使用時と AC 電源使用時で設定できる値が異なることにご注意ください。3、4 分はバッテリー使用時のみ、15 分は AC 電源使用時のみ設定可能です。

#### 対応情報

ヘッダファイル: sysctl.h  
インポートライブラリ: sysctl.lib

### 4-3-2. SysGetSuspendTime

自動で電源オフするまでの時間に設定されている現在値を取得します。バッテリー使用時と AC 電源使用時の 2 種類があります。

#### 構文

```
BOOL SysGetSuspendTime(  
    DWORD dwPowerMode,  
    LPDWORD lpdwSuspendTime  
);
```

#### パラメータ

*dwPowerMode*

[in] 電源モード（バッテリー、AC 電源）を示す変数です。値は以下に示す定義値のいずれかになります。

```
SYS_POWER_BATTERY_MODE  
SYS_POWER_EXTERNAL_MODE
```

*lpdwSuspendTime*

[out] 電源オフするまでの時間を示す変数へのポインタです。値は定義値のいずれかです。  
SysSetSuspendTime の頁をご参照ください。

#### 戻り値

関数が成功したとき、TRUE を返します。  
関数が失敗したとき、FALSE を返します。

#### 解説

なし

#### 対応情報

ヘッダファイル: sysctl.h  
インポートライブラリ: sysctl.lib



### 4-3-3. SysSetPowerDownDisableMode

電源キーによる電源オフ無効モードを設定します。

#### 構文

```
BOOL SysSetPowerDownDisableMode(  
    BOOL bDisableMode  
);
```

#### パラメータ

bDisableMode

[in] モードを指定するステータスです。TRUE のとき電源キー無効（電源オフ不可）、FALSE のとき電源キー有効（電源オフ可）となります。

#### 戻り値

関数が成功したとき、TRUE を返します。

関数が失敗したとき、FALSE を返します。

#### 解説

電源キーによる電源オフを無効にしたとき、電源キーを押すと、WM\_POWERBROADCAST メッセージ（wParam 値 PBT\_APMSUSPEND）を送信します。この場合、アプリケーションは電源オフする前に必要な操作を行うことができます。このメッセージを取得する場合は、コード内で以下の宣言を行ってください。

```
#define WM_POWERBROADCAST    0x0218  
#define PBT_APMSUSPEND      0x0004
```

この設定はデバイスの電源が入る度に初期化されます。

また、電源キーが無効に設定されていても、バッテリーカバーが開けられた、あるいはバッテリーの電圧が一定以下に下がったとき、デバイスは設定に拘らず緊急停止します。

#### 対応情報

ヘッダファイル: sysctl.h

インポートライブラリ: sysctl.lib

#### 4-3-4. SysGetPowerDownDisableMode

電源キーによる電源オフ無効モードの状態を取得します。

##### 構文

```
BOOL SysGetPowerDownDisableMode(  
    PBOOL pbDisableMode  
);
```

##### パラメータ

*pbDisableMode*

[out] モードを指定するステータスへのポインタです。TRUE のとき電源キー無効（電源オフ不可）、FALSE のとき電源キー有効（電源オフ可）となります。

##### 戻り値

関数が成功したとき、TRUE を返します。

関数が失敗したとき、FALSE を返します。

##### 解説

なし

##### 対応情報

ヘッダファイル: sysctl.h

インポートライブラリ: sysctl.lib

### 4-3-5. SysPowerDown

次に電源オンされたときの起動方法を設定したうえで、電源オフします。

#### 構文

```
BOOL SysPowerDown (  
    DWORD dwOnMode  
);
```

#### パラメータ

*dwOnMode*

[in] 次回の起動方法を示す変数です。値は以下に示す定義値のいずれかになります。

SYS_POWERON_RESUME	// 通常の電源キーによる起動と同じです。
SYS_POWERON_RESET	// ソフトリセットします。

#### 戻り値

関数が成功したとき、TRUE を返します。

関数が失敗したとき、FALSE を返します。

#### 解説

アプリケーションプログラムでサスペンドを行う場合は、必ず本関数を使用して下さい。Windows CE の API (GwesPowerOffSystem、SetSystemPowerState など) で直接サスペンドを行うとシステムが不安定になる場合があります。

#### 対応情報

ヘッダファイル: sysctl.h

インポートライブラリ: sysctl.lib

#### 4-3-6. SysSoftReset

デバイスをソフトリセットします。

##### 構文

```
BOOL SysSoftReset (void);
```

##### パラメータ

*None.*

##### 戻り値

関数が実行されると、デバイスはソフトリセットされ、再起動します。  
関数が失敗したとき、FALSE を返します。

##### 解説

アプリケーションプログラムでリセットを行う場合は、本関数を使用してください。

##### 対応情報

ヘッダファイル: sysctl.h

インポートライブラリ: sysctl.lib

### 4-3-7. SysSetWakeSources

デバイスを起動させる要因を設定します。

#### 構文

```
BOOL SysSetWakeSources(  
    DWORD SourceFlags  
);
```

#### パラメータ

*SourceFlag*

[in] 指定したフラグ以外のビットをマスクします。フラグは以下の表に示すいずれか、またはその論理和となります。

表 7 : SysSetWakeSources のパラメータ

フラグ	解説
SYS_WAKESOURCE_USB_CRADLE	PC とのクレードルを介する USB 接続で起動する
SYS_WAKESOURCE_TRIGGER	(Reserve)
SYS_WAKESOURCE_BLUETOOTH	(Reserve)

0 に設定すると、デバイスは上記の方法では起動しません。

#### 戻り値

関数が成功したとき、TRUE を返します。

関数が失敗したとき、FALSE を返します。

#### 解説

初期設定は 0 で、いずれの方法でも起動しません。

#### 対応情報

ヘッダファイル: sysctl.h

インポートライブラリ: sysctl.lib

#### 4-3-8. SysGetWakeSources

デバイスを起動させる要因を取得します。

##### 構文

```
BOOL SysGetWakeSources(  
    LPDWORD lpFlags  
);
```

##### パラメータ

*lpFlag*

[out] 起動要因を示すフラグへのポインタです。この値は sysctl.h に定義されるフラグ SYS\_WAKESOURCE\_XXX のいずれか、またはその論理和となります。

##### 戻り値

関数が成功したとき、TRUE を返します。

関数が失敗したとき、FALSE を返します。

##### 解説

なし

##### 対応情報

ヘッダファイル: sysctl.h

インポートライブラリ: sysctl.lib

## 4-4. Backlight and TouchPanel API

表 8 : Backlight and TouchPanel API 関数一覧

関数名	解説	備考
SysSetBacklightLevel	バックライト輝度レベルを設定します。	
SysGetBacklightLevel	バックライト輝度レベルを取得します。	
SysSetBacklightTimeout	バックライトの自動オフまでの時間を設定します。	
SysGetBacklightTimeout	バックライトの自動オフまでの時間の現在値を取得します。	
SysSetTouchPanelState	タッチパネルの有効／無効を設定します。	
SysGetTouchPanelState	タッチパネルの有効／無効についての状態を取得します。	

#### 4-4-1. SysSetBacklightLevel

バックライト輝度レベルを設定します。

##### 構文

```
BOOL SysSetBacklightLevel(  
    BYTE dwLevel  
);
```

##### パラメータ

*dwLevel*

[in] バックライト輝度レベルを示す変数です。値は以下に示す定義値のいずれかです。

SYS_BACKLIGHT_LEVEL_MINIMUM	// 最小輝度です。
SYS_BACKLIGHT_LEVEL_1	// 輝度レベル 1 です。
SYS_BACKLIGHT_LEVEL_2	// 輝度レベル 2 です。
SYS_BACKLIGHT_LEVEL_3	// 輝度レベル 3 です。
SYS_BACKLIGHT_LEVEL_4	// 輝度レベル 4 です。
SYS_BACKLIGHT_LEVEL_MAXIMUM	// 最大輝度です。

##### 戻り値

関数が成功したとき、TRUE を返します。  
関数が失敗したとき、FALSE を返します。

##### 解説

なし

##### 対応情報

ヘッダファイル: sysctl.h  
インポートライブラリ: sysctl.lib



#### 4-4-2. SysGetBacklightLevel

バックライト輝度レベルを取得します。

##### 構文

```
BOOL SysGetBacklightLevel(  
    PBYTE lpdwLevel  
);
```

##### パラメータ

*lpdwLevel*

[out] バックライト輝度レベルを示す変数へのポインタです。値は以下に示す定義値のいずれかです。

SYS_BACKLIGHT_LEVEL_MINIMUM	// 最小輝度です。
SYS_BACKLIGHT_LEVEL_1	// 輝度レベル 1 です。
SYS_BACKLIGHT_LEVEL_2	// 輝度レベル 2 です。
SYS_BACKLIGHT_LEVEL_3	// 輝度レベル 3 です。
SYS_BACKLIGHT_LEVEL_4	// 輝度レベル 4 です。
SYS_BACKLIGHT_LEVEL_MAXIMUM	// 最大輝度です。

##### 戻り値

関数が成功したとき、TRUE を返します。

関数が失敗したとき、FALSE を返します。

##### 解説

なし

##### 対応情報

ヘッダファイル: sysctl.h

インポートライブラリ: sysctl.lib

### 4-4-3. SysSetBacklightTimeout

何も操作しないとき、バックライトが自動オフするまでの時間を設定します。バッテリー使用時、AC 電源使用時の 2 種類があります。

#### 構文

```
BOOL SysSetBacklightTimeout(
    DWORD dwPowerMode,
    DWORD dwBacklightTimeout
);
```

#### パラメータ

*dwPowerMode*

[in] 電源モード（バッテリー、AC 電源）を指定します。値は以下に示す定義値のいずれかです。

```
SYS_POWER_BATTERY_MODE
SYS_POWER_EXTERNAL_MODE
```

*dwBacklightTimeout*

[in] 自動オフまでの時間を示す変数です。値は以下に示す定義値のいずれかです。

```
BACKLIGHT_TIMEOUT_TEN_SEC           // バッテリー使用時のみ
BACKLIGHT_TIMEOUT_THIRTY_SEC        // バッテリー使用時のみ
BACKLIGHT_TIMEOUT_ONE_MIN
BACKLIGHT_TIMEOUT_TWO_MIN
BACKLIGHT_TIMEOUT_THREE_MIN
BACKLIGHT_TIMEOUT_FOUR_MIN
BACKLIGHT_TIMEOUT_FIVE_MIN
BACKLIGHT_TIMEOUT_SIX_MIN           // AC 電源使用時のみ
BACKLIGHT_TIMEOUT_SEVEN_MIN         // AC 電源使用時のみ
BACKLIGHT_TIMEOUT_EIGHT_MIN         // AC 電源使用時のみ
BACKLIGHT_TIMEOUT_NINE_MIN          // AC 電源使用時のみ
BACKLIGHT_TIMEOUT_TEN_MIN           // AC 電源使用時のみ
BACKLIGHT_TIMEOUT_INDEFINITELY
```

#### 戻り値

関数が成功したとき、TRUE を返します。  
関数が失敗したとき、FALSE を返します。

#### 解説

自動オフまでの時間設定で有効な範囲は、バッテリー使用時 10 秒から 5 分、AC 使用時 1 分から 10 分です。また、どちらの場合も無限（自動オフしない）に設定することもできます。

#### 対応情報

ヘッダファイル: sysctl.h  
インポートライブラリ: sysctl.lib

#### 4-4-4. SysGetBacklightTimeout

何も操作しないとき、バックライトが自動オフするまでの時間の現在値を取得します。バッテリー使用時、AC 電源使用時の 2 種類があります。

##### 構文

```
BOOL SysGetBacklightTimeout(  
    DWORD dwPowerMode,  
    LPDWORD lpdwBacklightTimeout  
);
```

##### パラメータ

*dwPowerMode*

[in] 電源モード（バッテリー、AC 電源）を指定します。値は以下に示す定義値のいずれかです。

```
SYS_POWER_BATTERY_MODE  
SYS_POWER_EXTERNAL_MODE
```

*lpdwBacklightTimeout*

[out] 自動オフまでの時間を示す変数へのポインタです。

##### 戻り値

関数が成功したとき、TRUE を返します。

関数が失敗したとき、FALSE を返します。

##### 解説

自動オフ時間の設定には定義値を用いる必要があります。

SysSetBacklightTimeout の頁をご参照ください。

##### 対応情報

ヘッダファイル: sysctl.h

インポートライブラリ: sysctl.lib

#### 4-4-5. SysSetTouchPanelState

タッチパネルの有効／無効を設定します。

##### 構文

```
BOOL SysSetTouchPanelState(  
    BOOL bEnable  
);
```

##### パラメータ

*bEnable*

[in] タッチパネルの有効／無効を示すステータスです。TRUE のとき有効、FALSE のとき無効です。

##### 戻り値

関数が成功したとき、TRUE を返します。

関数が失敗したとき、FALSE を返します。

##### 解説

なし

##### 対応情報

ヘッダファイル: sysctl.h

インポートライブラリ: sysctl.lib

#### 4-4-6. SysGetTouchPanelState

タッチパネルの有効／無効についての状態を取得します。

##### 構文

```
BOOL SysGetTouchPanelState(  
    PBOOL pbEnable  
);
```

##### パラメータ

*pbEnable*

[out] タッチパネルの有効／無効を示す変数へのポインタです。TRUE のとき有効、FALSE のとき無効です。

##### 戻り値

関数が成功したとき、TRUE を返します。

関数が失敗したとき、FALSE を返します。

##### 解説

なし

##### 対応情報

ヘッダファイル: sysctl.h

インポートライブラリ: sysctl.lib

## 4-5. Wireless LAN API

表 9 : Wireless LAN API 関数一覧

関数名	解説	備考
SysSetWLANPower	無線 LAN の電源状態を設定します。	
SysGetWLANPower	無線 LAN の電源状態を取得します。	
SysSetWLANRegulatory	無線 LAN の規制について設定します。	
SysGetWLANRegulatory	無線 LAN の規制状態を取得します。	
SysGetWLANConnectStatus	無線 LAN の接続状態を取得します。	
SysGetWLANSignalLevel	無線 LAN の信号強度を取得します。	

#### 4-5-1. SysSetWLANPower

無線 LAN の電源状態を設定します。

##### 構文

```
BOOL SysSetWLANPower(  
    DWORD dwPowerStatus  
);
```

##### パラメータ

*dwPowerStatus*

[in] 電源状態を示す変数です。値は以下に示す定義値のいずれかになります。

```
SYS_WLAN_POWER_OFF        // 電源 OFF  
SYS_WLAN_POWER_ON         // 電源 ON
```

##### 戻り値

関数が成功したとき、TRUE を返します。

関数が失敗したとき、FALSE を返します。

##### 解説

なし

##### 対応情報

ヘッダファイル: sysctl.h

インポートライブラリ: sysctl.lib

## 4-5-2. SysGetWLANPower

無線 LAN の電源状態を取得します。

### 構文

```
BOOL SysGetWLANPower(  
    LPDWORD lpdwPowerStatus,  
);
```

### パラメータ

*lpdwPowerStatus*

[out] 電源状態を示す変数へのポインタです。

### 戻り値

関数が成功したとき、TRUE を返します。

関数が失敗したとき、FALSE を返します。

### 解説

パラメータの値については SysSetWLANPower の頁をご参照ください。

### 対応情報

ヘッダファイル: sysctl.h

インポートライブラリ: sysctl.lib



### 4-5-3. SysGetWLANConnectStatus

無線 LAN の接続状態を取得します。

#### 構文

```
BOOL SysGetWLANConnectStatus(  
    LPDWORD lpStatus  
);
```

#### パラメータ

*lpStatus*

[out] 現在の接続状態を示す変数へのポインタです。 値は以下に示す定義値のいずれかとなります。

```
SYS_WLAN_STATUS_UNAVAILABLE  
SYS_WLAN_STATUS_DISCONNECTED  
SYS_WLAN_STATUS_CONNECTED
```

#### 戻り値

関数が成功したとき、TRUE を返します。  
関数が失敗したとき、FALSE を返します。

#### 解説

なし

#### 対応情報

ヘッダファイル: sysctl.h  
インポートライブラリ: sysctl.lib

#### 4-5-4. SysGetWLANSignalLevel

接続中の無線 LAN の信号強度を取得します。

##### 構文

```
BOOL SysGetWLANSignalLevel(  
    LPDWORD lpLevel  
);
```

##### パラメータ

*lpLevel*

[out] 現在の信号強度を示す変数へのポインタです。最小値は 0、最大値は 100 です。

##### 戻り値

関数が成功したとき、TRUE を返します。

関数が失敗したとき、FALSE を返します。

##### 解説

回線が接続中でない場合は、本関数で信号強度を取得することはできません。

##### 対応情報

ヘッダファイル: sysctl.h

インポートライブラリ: sysctl.lib

## 4-6. Bluetooth API

表 10 : Bluetooth API 関数一覧

関数名	解説	備考
SysSetBluetoothPower	Bluetooth の電源状態を設定します。	
SysGetBluetoothPower	Bluetooth の電源状態を取得します。	

#### 4-6-1. SysSetBluetoothPower

Bluetooth の電源状態を設定します。

##### 構文

```
BOOL SysSetBluetoothPower(  
    DWORD dwPowerStatus  
);
```

##### パラメータ

*dwPowerStatus*

[in] 電源状態を示す変数です。値は以下に示す定義値のいずれかになります。

```
SYS_BT_POWER_ON  
SYS_BT_POWER_OFF
```

##### 戻り値

関数が成功したとき、TRUE を返します。

関数が失敗したとき、FALSE を返します。

##### 解説

なし

##### 対応情報

ヘッダファイル：sysctl.h

インポートライブラリ：sysctl.lib

## 4-6-2. SysGetBluetoothPower

Bluetooth の電源状態を取得します。

### 構文

```
BOOL SysGetBluetoothPower(  
    LPDWORD lpdwPowerStatus  
);
```

### パラメータ

*lpdwPowerStatus*

[out] 電源状態を示す変数へのポインタです。

### 戻り値

関数が成功したとき、TRUE を返します。

関数が失敗したとき、FALSE を返します。

### 解説

なし

### 対応情報

ヘッダファイル: sysctl.h

インポートライブラリ: sysctl.lib

## 4-7. Vibrator API

表 11 : Vibrator API 関数一覧

関数名	解説	備考
SysSetVibrator	バイブレータを指定した時間、振動させます。	
SysStopVibrator	バイブレータを停止します。	

#### 4-7-1. SysSetVibrator

バイブレータを指定した時間、振動させます。

##### 構文

```
BOOL SysSetVibrator(  
    DWORD dwTimeMilliSecond  
);
```

##### パラメータ

*dwTimeMilliSecond*

[in] 振動時間をミリ秒単位で指定します。

##### 戻り値

関数が成功したとき、TRUE を返します。

関数が失敗したとき、FALSE を返します。

##### 解説

なし

##### 対応情報

ヘッダファイル: sysctl.h

インポートライブラリ: sysctl.lib

#### 4-7-2. SysStopVibrator

バイブレータを停止します。

##### 構文

```
BOOL SysStopVibrator(void);
```

##### パラメータ

なし

##### 戻り値

関数が成功したとき、TRUE を返します。  
関数が失敗したとき、FALSE を返します。

##### 解説

なし

##### 対応情報

ヘッダファイル: sysctl.h

インポートライブラリ: sysctl.lib



## 4-8. LED API

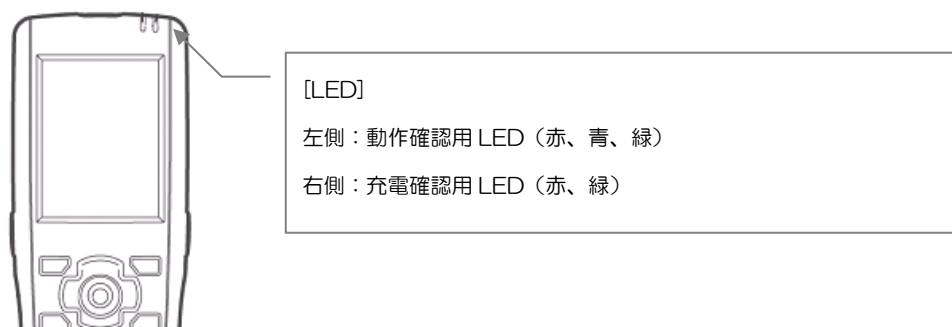
表 12 : LED API 関数一覧

関数名	解説	備考
SysSetLED	LED を制御します。	

### 4-8-1. SysSetLED

指定した LED を点灯、または消灯します。LED の制御は下図の左側の動作確認用 LED のみ行う事ができます。

※ Wireless LAN や Bluetooth の電源をオンした場合やバーコードの読み取りが成功した場合に同 LED が点滅/点灯します。特定のレジストリ値を変更する事により、本制御を行わない事ができます。詳細はアプリケーション開発ガイドをご参照下さい。



#### 構文

```
BOOL SysSetBlueLED(  
    DWORD dwColor,  
    BOOL On  
);
```

#### パラメータ

*dwColor*

[in] 点灯/消灯する。LED の色を指定します。下記より選択し設定します。

- SYS\_LED\_BLUE     // 青色 LED を指定します。
- SYS\_LED\_GREEN    // 緑色 LED を指定します。
- SYS\_LED\_RED       // 赤色 LED を指定します。

*On*

[in] 点灯消灯を示すステータスです。TRUE のとき点灯、FALSE のとき消灯になります。

#### 戻り値

関数が成功したとき、TRUE を返します。  
関数が失敗したとき、FALSE を返します。

#### 解説

なし

#### 対応情報

ヘッダファイル: sysctl.h

インポートライブラリ: sysctl.lib

## 4-9. Other API

表 13 : Other API 関数一覧

関数名	解説	備考
SysGetLibraryVersion	システム制御ライブラリのバージョン情報を取得します。	

#### 4-9-1. SysGetLibraryVersion

システム制御ライブラリのバージョン情報を取得します。

##### 構文

```
BOOL SysGetLibraryVersion(  
    LPTSTR lpVersionString  
);
```

##### パラメータ

*lpVersionString*  
[out] バージョン情報を示す文字列へのポインタです。

##### 戻り値

関数が成功したとき、TRUE を返します。  
関数が失敗したとき、FALSE を返します。

##### 解説

文字列の領域はアプリケーションによって割り当てられます。また、領域のサイズはヘッダファイルに規定されます。

##### 対応情報

ヘッダファイル: sysctl.h  
インポートライブラリ: sysctl.lib

## 付録 A：キー配置と仮想キーコード

H-15BJ デバイスにあるキーの配置を下図に示します。

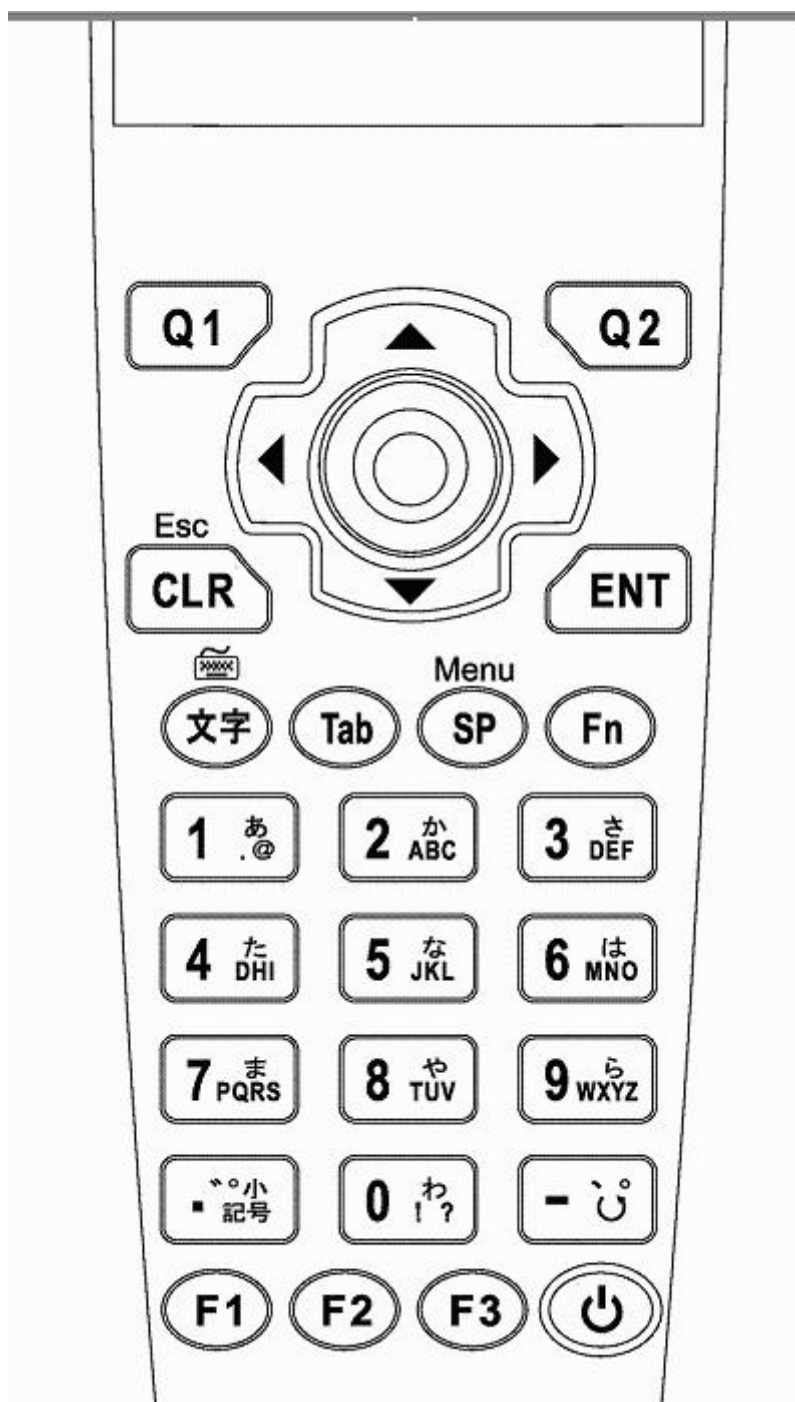


図 1：キーパッドレイアウト

また、各キーの仮想キーコードは次の通りです。

表 14 ： 仮想キーコード

キー	通常			Fn キーを押した場合		
	仮想キーコード	状態	繰返し	仮想キーコード	状態	繰返し
Q1	0xE3	-	No	0xE3	SHIFT	No
Q2	0xE4	-	No	0xE4	SHIFT	No
左	0x25 (VK_LEFT)	-	Yes	0x24 (VK_HOME)	-	No
上	0x26 (VK_UP)	-	Yes	0x21 (VK_PRIOR)	-	No
右	0x27 (VK_RIGHT)	-	Yes	0x23 (VK_END)	-	No
下	0x28 (VK_DOWN)	-	Yes	0x22 (VK_NEXT)	-	No
センタ	0x0D (VK_RETURN)	-	No	0x0D (VK_RETURN)	-	No
	0xEB (*)	-	No	0xEB (*)	-	No
左トリガ	0xE1	-	No	0xE1	-	No
右トリガ	0xE6	-	No	0xE6	-	No
CLR	0x08 (VK_BACK)	-	Yes	0x1B (VK_ESCAPE)	-	No
ENT	0x0D (VK_RETURN)	SHIFT	No	0x0D (VK_RETURN)	SHIFT	No
	0xEC (*)	-	No	0xEC (*)	-	No
文字	0xE9	-	No	0xE9	SHIFT	No
TAB	0x09 (VK_TAB)	-	No	0x09 (VK_TAB)	SHIFT	No
SP	0x20 (VK_SPACE)	-	Yes	0x12 (VK_MENU)	-	No
Fn						
1	0x31	-	Yes	0x7C (VK_F13)	-	No
2	0x32	-	Yes	0x7D (VK_F14)	-	No
3	0x33	-	Yes	0x7E (VK_F15)	-	No
4	0x34	-	Yes	0x7F (VK_F16)	-	No
5	0x35	-	Yes	0x80 (VK_F17)	-	No
6	0x36	-	Yes	0x81 (VK_F18)	-	No
7	0x37	-	Yes	0x82 (VK_F19)	-	No
8	0x38	-	Yes	0x83 (VK_F20)	-	No
9	0x39	-	Yes	0x84 (VK_F21)	-	No
0	0x30	-	Yes	0x86 (VK_F23)	-	No
.	0xBE (VK_OEM_PERIOD)	-	Yes	0x85 (VK_F22)	-	No
-	0xBD (VK_OEM_MINUS)	-	Yes	0x87 (VK_F24)	-	No
F1	0x70 (VK_F1)	-	No	0x70 (VK_F1)	SHIFT	No
F2	0x71 (VK_F2)	-	No	0x71 (VK_F2)	SHIFT	No
F3	0x72 (VK_F3)	-	No	0x72 (VK_F3)	SHIFT	No
Power						

※:センタキーおよび ENT キーは 2 種類のキーコードを持ちます。これらのキーコードは関数 SysSetAlternateKeycode によって切り替えることができます。

## 付録 B：ファンクションキーおよびショートカットキー

ファンクションキーおよびショートカットキーの定義名の一覧とそのサポート情報です。  
以下の定義は SysSetFunctionKeyPrograms() と SysGetFunctionKeyPrograms() で使用されます。

表 15：ファンクションキーおよびショートカットキー

	定義名	サポート (*1)	備考 (*2)
ファンクションキー	SYS_KEY_F1	✓	
	SYS_KEY_F2	✓	
	SYS_KEY_F3	✓	
	SYS_KEY_F4	–	
	SYS_KEY_F5	–	
	SYS_KEY_F6	–	
	SYS_KEY_F7	–	
	SYS_KEY_F8	–	
	SYS_KEY_Q1	✓	
	SYS_KEY_Q2	✓	
ショートカットキー	SYS_KEY_FN_F1	✓	“Fn” + “F1”
	SYS_KEY_FN_F2	✓	“Fn” + “F2”
	SYS_KEY_FN_F3	✓	“Fn” + “F3”
	SYS_KEY_FN_F4	–	“Fn” + “F4”
	SYS_KEY_FN_Q1	✓	“Fn” + “Q1”
	SYS_KEY_FN_Q2	✓	“Fn” + “Q2”
	SYS_KEY_FN_1	✓	“Fn” + “1”
	SYS_KEY_FN_2	✓	“Fn” + “2”
	SYS_KEY_FN_3	✓	“Fn” + “3”
	SYS_KEY_FN_4	✓	“Fn” + “4”
	SYS_KEY_FN_5	✓	“Fn” + “5”
	SYS_KEY_FN_6	✓	“Fn” + “6”
	SYS_KEY_FN_7	✓	“Fn” + “7”
	SYS_KEY_FN_8	✓	“Fn” + “8”
	SYS_KEY_FN_9	✓	“Fn” + “9”
	SYS_KEY_FN_0	✓	“Fn” + “0”
	SYS_KEY_FN_PERIOD	✓	“Fn” + “.”
	SYS_KEY_FN_MINUS	✓	“Fn” + “-”

\*1: ✓ のとき、関数は TRUE を返し、– のとき、関数は FALSE を返します。

\*2: “Fn” + “xx” とは、Fn キーを押した後に xx キーを押すという意味です。

製品名 : H-15BJ シリーズ  
仕様書名: H-15BJ システム制御ライブラリ仕様書  
発行番号: Rev.1.1  
管理番号: SI14044

---

株式会社オプトエレクトロニクス

本社 〒335-0002 埼玉県蕨市塚越 4-12-17

TEL 048-446-1181 FAX 048-434-2820

大阪営業所 〒541-0058 大阪府大阪市中央区南久宝寺町 3-4-14 三興ビル 7F

TEL 06-6251-0072 FAX 06-6251-0076

URL:<http://home.opto.co.jp>

e-mail:[sales@opto.co.jp](mailto:sales@opto.co.jp)

---