

M3069 マイコンボード  
キャラクタ LCD 制御ライブラリ  
関数リファレンス



テクノウェーブ株式会社

---

## 目次

<b>1. はじめに</b> .....	<b>2</b>
□ 「M3069 マイコンボード キャラクタ LCD 制御ライブラリ」について.....	2
□ 本リファレンスの対応ファイルとバージョンについて.....	3
□ 本マニュアル内の表記について.....	3
<b>2. 使用準備</b> .....	<b>4</b>
□ 実行時に必要なファイル.....	4
□ 開発時に必要なファイル.....	4
□ ATF ファイルの情報.....	5
<b>3. 使用方法</b> .....	<b>6</b>
□ ハードウェア接続.....	6
□ プログラミング.....	7
初期化作業.....	7
表示方法.....	7
<b>4. 関数リファレンス</b> .....	<b>9</b>
□ 関数の戻り値.....	9
□ 初期化／終了用関数.....	10
<i>ATF_LCDAttach()</i> .....	10
<i>ATF_LCDDetach()</i> .....	10
<i>ATF_LCDSetPort()</i> .....	10
<i>ATF_LCDInit()</i> .....	10
□ 制御用関数.....	11
<i>ATF_LCDReadReg()</i> .....	11
<i>ATF_LCDWriteReg()</i> .....	11
<i>ATF_LCDReadData()</i> .....	12
<i>ATF_LCDWriteData()</i> .....	12
<i>ATF_LCDPuts()</i> .....	12
サポート情報.....	13

# 1. はじめに

## □ 「M3069 マイコンボード キャラクタ LCD 制御ライブラリ」について

「M3069 マイコンボード キャラクタ LCD 制御ライブラリ」は、パソコン用のアプリケーションプログラムから、M3069 シリーズのマイコンボード(『USBM3069(F)』、『LANM3069』)を介して、キャラクタタイプの LCD を制御するためのライブラリです。本ライブラリは、M3069 マイコンボード独自の アタッチメントファーム という機構を利用しています。LCD への制御信号はボード上のマイコンが直接操作されるため、パソコンからポート操作で制御するよりも高速に通信することができます。

本ライブラリによる機能拡張は、「USBM ライブラリ」による I/O 機能に追加して行われますので、シリアルインタフェース用として使用しない端子は、そのまま入出力ポートとして利用可能です。また、アナログ入出力、タイマ、外部バスなどの機能も通常どおりご利用いただけます。

### アタッチメントファームとは

M3069 シリーズのマイコンボードでは、マイコン用の小さなプログラムを、パソコンからボードの RAM 上にダウンロードして、付属のファームウェア (システムファーム) に新たな機能を追加することができます。このマイコン用プログラムのことをアタッチメントファームと呼んでいます (図 1)。また、アタッチメントファームを格納したファイルを ATF ファイルと呼びます。

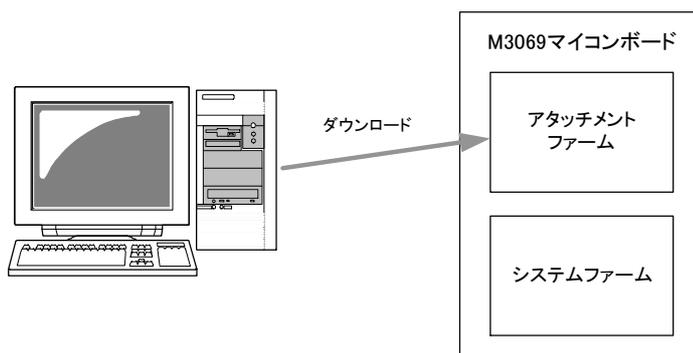


図 1 アタッチメントファームのダウンロード

アタッチメントファームは、マイコン側で実行されますので、通信のための遅延時間が無く、リアルタイム性の高い処理が可能となります。また、普段はパソコン上にファイルとして保存され、必要に応じてダウンロードされるため、機能変更やメンテナンスが容易です。

注意点として、通常、アタッチメントファームは、ユーザーメモリの一部にダウンロードされるため、メモリの一部が使用できなくなります。アプリケーションの中で、ユーザーメモリを利用する場合には、アタッチメントファームのダウンロード領域と重複しないようにする必要があります。

アタッチメントファームについての詳細は別紙「M3069 マイコンボード ユーザーファーム開発マニュアル」を参照してください。

---

□ **本リファレンスの対応ファイルとバージョンについて**

本リファレンスは下記のバージョンのファイル内容を元に記載されています。過去のバージョンや、将来のバージョンについては記載内容と異なる場合がございますのでご注意ください。

表 1 ライブラリの対応バージョン

ファイル名	バージョン番号	対応 OS
M3069Lcd.dll	1.0.x.x	Windows 98SE 以降

表 2 ATF ファイルの対応バージョン

ファイル名	バージョン番号	対応システムファーム
M3069Lcd.atf	1.0.x	3.0.1 以降

□ **本マニュアル内の表記について**

本マニュアル内では対応製品 (『USBM3069(F)』、『LANM3069』) を「デバイス」と表記する場合があります。インタフェースを明記する場合には「USB デバイス」、「LAN デバイス」などのように表記します。

本リファレンス内でハードウェアの電気的状態について記述する必要がある場合には、下記のように表記します。

表 3 電気的状態の表記方法

表記	状態
“ON”	電流が流れている状態、スイッチが閉じている状態、オープンコレクタ(オープンドレイン)出力がシンク出力している状態。
“OFF”	電流が流れていない状態、スイッチが開いている状態、オープンコレクタ(オープンドレイン)出力がハイインピーダンスの状態。
“Hi”	電圧がロジックレベルのハイレベルに相当する状態。
“Lo”	電圧がロジックレベルのローレベルに相当する状態。
“Z”	端子がハイインピーダンスの状態。

また、数値について「0x」、「&H」、「H」はいずれもそれに続く数値が 16 進数であることを表します。「0x10」、「&H1F」、「H' 20」などはいずれも 16 進数です。同様に「B」に続く数値は 2 進数であることを表します。例えば“B'01000001”のように表記されます。数値の最初に特別な表記が無い場合は 10 進数です。

---

## 2. 使用準備

### □ 実行時に必要なファイル

本ライブラリを使用するには、以下のファイルが必要になります。パソコンのシステムフォルダ (C:\Windows\System32 など)、または、ライブラリを利用する「.exe」ファイルと同じフォルダにコピーしてください。

表 4 実行時に必要なファイル

ファイル名	バージョン	説明
USBM3069.dll	3.3.0.1 以降	製品付属のライブラリ(USBM ライブラリ)。通常、ドライバやツールのインストール時にシステムフォルダにコピーされます。
M3069Lcd.atf <sup>1</sup>	1.0.2 以降	マイコンにダウンロードするアタッチメントファーム
M3069Lcd.dll	1.0.0.1 以降	ライブラリ本体

<sup>1</sup> 使用するにはボードのシステムファーム 3.0.1 以降が必要です。

### □ 開発時に必要なファイル

以下のファイルはアプリケーションプログラムの開発時に必要になるファイルです。適宜、プロジェクトに追加して使用してください。

表 5 開発に必要なファイル

開発環境	ファイル名	使用方法
Visual C++ など	USBM3069.lib	リンクされるようにプロジェクトに追加してください。
	M3069Lcd.lib	リンクされるようにプロジェクトに追加してください。
	USBM3069.h	ソースファイルでインクルードします。
	M3069Lcd.h	ソースファイルでインクルードします。
Visual Basic 6.0, VBA など	USBM3069.bas	標準モジュールとして追加します。
	M3069Lcd.bas	標準モジュールとして追加します。
Visual Basic .NET (2005)	USBM3069.vb	プロジェクトに追加します。
	M3069Lcd.vb	プロジェクトに追加します。

---

□ ATF ファイルの情報

図 2 は使用する ATF ファイルの情報です。本ライブラリ使用中は、ユーザーメモリの「プログラムの先頭アドレス」から、「プログラムの最終アドレス」の範囲が使用できなくなりますので、ご注意ください

The image shows a Windows-style dialog box titled "ATFファイルの情報" (ATF File Information). It contains several fields for file and program details:

ファイル形式のバージョン	00020000
プログラムの説明	M3069 LCD Control ATF
作成者	Technowave Ltd.
プログラムのバージョン	00010001
要求するファームバージョン	00030000
プログラムの先頭アドレス	00FFDF20
プログラムの最終アドレス	00FFE68A
コマンドハンドラのアドレス	00FFE4DC
メイン関数のアドレス	00000000

An "OK" button is located at the bottom right of the dialog box.

図 2 「M3069Lcd.atf」の情報

### 3. 使用方法

#### □ ハードウェア接続

通常、キャラクタタイプの LCD では、4 ビットのデータバスでの制御と、8 ビットのデータバスでの制御が可能になっていますが、本ライブラリでは 4 ビット制御のみサポートします。

制御には P40-P47、または、PA0-PA7 を使用します。P40-P47 を使用して接続する場合の例を示します。

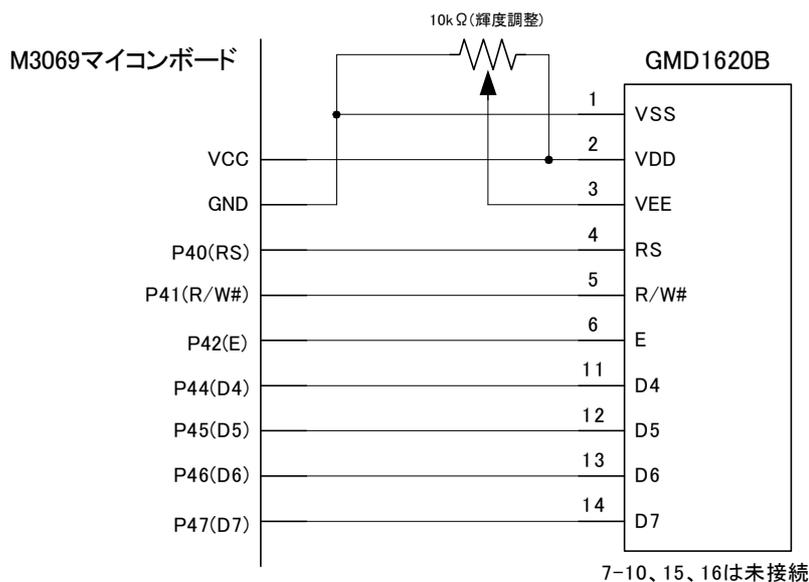


図 3 GMD1620B(スタンレー電気)の接続例

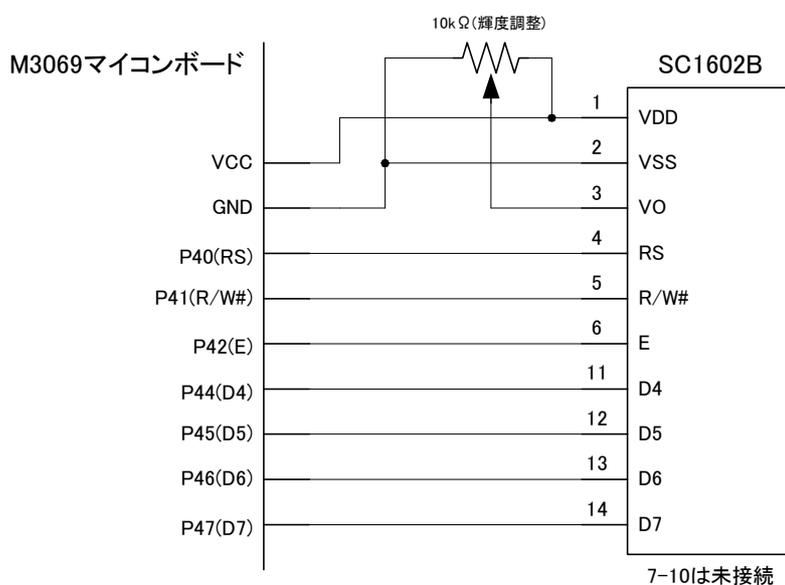


図 4 SC1602B(Sunlike Display Tech.)の接続例

## □ プログラミング

### 初期化作業

リスト 1 は C 言語での初期化例です。

リスト 1

```
1 //接続
2 m_hDev = USBM_Open();
3
4 if(m_hDev) {
5     //初期化
6     USBM_Initialize(m_hDev);
7
8     //ATF ファイルのダウンロード
9     if(ATF_LCDAttach(m_hDev)) {
10        //ATF ファイルのダウンロードに失敗した場合
11        USBM_Close(m_hDev);
12        return;
13    }
14    ATF_LCDInit(m_hDev); //LCD の初期化
15    ATF_LCDWriteReg(m_hDev, ATF_LCD_CLEAR); //LCD をクリア
16    ATF_LCDWriteReg(m_hDev, ATF_LCD_ENABLE); //表示を許可する
17 }
```

2～6 行目は、ボードへの接続作業です。「USBM ライブラリ」を使用する場合と同様です。

9 行目で「M3069Lcd.atf」をボードへダウンロードしています。ファイルは DLL ファイルと同じ順序で検索されますので、通常は呼び出し元の「.exe」ファイルと同じフォルダにおいてください。

14 行目は、LCD モジュールの初期化を行う関数を呼び出しています。

15、16 行目は、LCD モジュールのインストラクション・レジスタにコマンドを送り、表示の準備を行っています。LCD モジュールに対して命令を送る場合には、この例のように `ATF_LCDWriteReg()` 関数を使用し、インストラクション・レジスタに対して書き込みを行います。使用できる制御コマンドについては関数リファレンスを参照してください。

### 表示方法

キャラクタタイプの LCD では、表示位置と対応する DDRAM アドレスにキャラクタコードを書き込むことで文字を表示します。16 文字×2 行と、20 文字×4 行の LCD モジュールの、表示位置とアドレスの関係を図 5、図 6 に示します。

0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
0x40	0x41	0x42	0x43	0x44	0x45	0x46	0x47	0x48	0x49	0x4A	0x4B	0x4C	0x4D	0x4E	0x4F

図 5 16 文字×2 行の LCD の表示位置とアドレスの関係

0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F	0x10	0x11	0x12	0x13
0x40	0x41	0x42	0x43	0x44	0x45	0x46	0x47	0x48	0x49	0x4A	0x4B	0x4C	0x4D	0x4E	0x4F	0x50	0x51	0x52	0x53
0x14	0x15	0x16	0x17	0x18	0x19	0x1A	0x1B	0x1C	0x1D	0x1E	0x1F	0x20	0x21	0x22	0x23	0x24	0x25	0x26	0x27
0x54	0x55	0x56	0x57	0x58	0x59	0x5A	0x5B	0x5C	0x5D	0x5E	0x5F	0x60	0x61	0x62	0x63	0x64	0x65	0x66	0x67

図 6 20 文字×4 行の LCD の表示位置とアドレスの関係

リスト 2 に C 言語による表示プログラム例、図 7 に表示結果を示します。

リスト 2

```

1 //2 行目の先頭にカーソルを移動
2 ATF_LCDWriteReg(m_hDev, ATF_LCD_SET_DDRAM | 0x40);
3 //文字列の表示
4 ATF_LCDPuts(m_hDev, "M3069 Series");

```

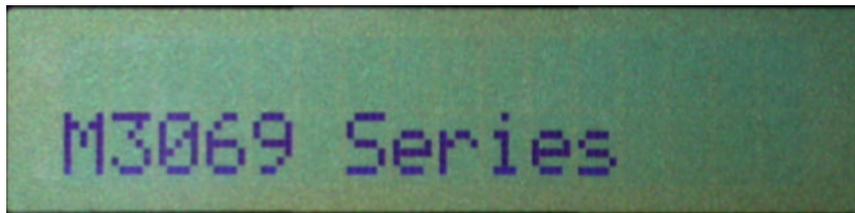


図 7 表示結果

## 4. 関数リファレンス

各関数の説明は、C 言語、Visual Basic® (6.0 以前)、Visual Basic.NET (Visual Basic 2005)それぞれにおけるプロトタイプ、変数の説明、動作説明の順になっています。

ほとんどの関数の戻り値は 32 ビットの整数で関数の実行結果を表します(以下参照)。関数がそれ以外の特別な戻り値を返す場合は、各関数の動作説明の欄で内容を示します。

### □ 関数の戻り値

表 6 は、「USBM ライブラリ」の関数の戻り値です。戻り値を示す各定数は、「USBM ライブラリ」の各言語用の定義ファイル(拡張子が「.h」、「.bas」、「.vb」のファイル)中で定義されています。

表 6 関数の戻り値

定数	値	意味
TW_OK	0x00000000	正常終了
TW_INVALID_HANDLE	0x00000001	デバイスのハンドルが無効
TW_DEVICE_NOT_FOUND	0x00000002	デバイスが見つからない
TW_IO_ERROR	0x00000004	送受信中にエラーが発生した
TW_INSUFFICIENT_RESOURCES	0x00000005	リソースエラー(デバイスの最大接続数を超えた場合など)
TW_INVALID_ARGS	0x00000010	関数に渡された引数が無効
TW_NOT_SUPPORTED	0x00000011	サポートされない機能
TW_OTHER_ERROR	0x00000012	
TW_TIMEOUT	0xffff0001	送信または受信処理がタイムアウトした
TW_FILE_ERROR	0xffff0002	ファイル操作に関するエラーが発生した
TW_MEMORY_ERROR	0xffff0003	メモリの確保に失敗した
TW_DATA_NOT_FOUND	0xffff0004	有効なデータが見つからなかった
TW_SOCKET_ERROR	0xffff0005	Winsock のエラー(多くの場合 WSAGetLastError() を呼び出すとさらに詳しい情報を得ることができます)
TW_ACCESS_DENIED	0xffff0006	デバイスとの認証作業に失敗した
TW_ATF_ERR_FILE_VERSION	0xffff0101	対応するバージョンより新しい ATF ファイルフォーマット
TW_ATF_ERR_ILLEGAL_FILE	0xffff0102	ATF ファイルの内容が不正
TW_ATF_ERR_SERVICE_VERSION	0xffff0103	ファームのバージョンが ATF ファイルの要求より古い

---

□ 初期化／終了用関数

### ATF\_LCDAttach()

```
TW_STATUS ATF_LCDAttach(TW_HANDLE hDev)
Function ATF_LCDAttach(ByVal hDev As Long) As Long
Function ATF_LCDAttach(ByVal hDev As Integer) As Integer
```

hDev : デバイスのハンドル

キャラクタ LCD 制御用のアタッチメントファーム (M3069Lcd.atf) をデバイスにダウンロードします。他のライブラリ関数の使用に先立って必ず呼び出す必要があります。

### ATF\_LCDDetach()

```
TW_STATUS ATF_LCDDetach(TW_HANDLE hDev)
Function ATF_LCDDetach(ByVal hDev As Long) As Long
Function ATF_LCDDetach(ByVal hDev As Integer) As Integer
```

hDev : デバイスのハンドル

キャラクタ LCD 制御用のアタッチメントファームの使用を終了します。

### ATF\_LCDSetPort()

```
TW_STATUS ATF_LCDSetPort(TW_HANDLE hDev, DWORD Port)
Function ATF_LCDSetPort(ByVal hDev As Long, ByVal Port As Long) As Long
Function ATF_LCDSetPort(ByVal hDev As Integer, ByVal Port As Integer) As Integer
```

hDev : デバイスのハンドル  
Port : LCD と接続するポートを指定します  
    USBM\_P4 : P40-P47 を使用します  
    USBM\_PA : PA0-PA7 を使用します

LCD 制御に使用するポートを指定します。デフォルトでは P40-P47 を使用します。

### ATF\_LCDInit()

```
TW_STATUS ATF_LCDInit(TW_HANDLE hDev)
Function ATF_LCDInit(ByVal hDev As Long) As Long
Function ATF_LCDInit(ByVal hDev As Integer) As Integer
```

hDev : デバイスのハンドル

LCD を初期化します。LCD の電源投入後に必ず実行します。

---

## □ 制御用関数

### ATF\_LCDReadReg ()

TW\_STATUS ATF\_LCDReadReg(TW\_HANDLE hDev, BYTE \*pReg)

Function ATF\_LCDReadReg(ByVal hDev As Long, ByRef pReg As Byte) As Long

Function ATF\_LCDReadReg (ByVal hDev As Integer, ByRef pReg As Byte) As Integer

hDev : デバイスのハンドル

pReg : [out]読み出したレジスタの値

ビット0-6 : アドレスカウンタの値

ビット7 : ビジーフラグ

LCD の制御レジスタ (インストラクション・レジスタ) から読み出しを行います。

通常ビット0-6に現在のアドレスカウンタの値、ビット7にビジーフラグが格納されます。

詳しくはLCDのデータシートを参照してください。

### ATF\_LCDWriteReg ()

TW\_STATUS ATF\_LCDWriteReg(TW\_HANDLE hDev, BYTE Reg)

Function ATF\_LCDWriteReg(ByVal hDev As Long, ByVal Reg As Byte) As Long

Function ATF\_LCDWriteReg(ByVal hDev As Integer, ByVal Reg As Byte) As Integer

hDev : デバイスのハンドル

Reg : 書き込む値

ATF\_LCD\_CLEAR (0x01) : 画面をクリアしてカーソルをホームに戻す

ATF\_LCD\_HOME (0x02) : カーソルをホームに戻す

ATF\_LCD\_ENABLE (0x0c) : 表示を許可する

ATF\_LCD\_CRCSR\_ENABLE (0x0a) : カーソルを表示する

ATF\_LCD\_CRCSR\_BLINK (0x09) : カーソル位置の文字をブリンクする

ATF\_LCD\_DISABLE (0x08) : 表示をオフ

ATF\_LCD\_CRCSR\_LEFT (0x10) : カーソルを左に移動

ATF\_LCD\_CRCSR\_RIGHT (0x14) : カーソルを右に移動

ATF\_LCD\_SCRL\_LEFT (0x18) : 画面を左にスクロール

ATF\_LCD\_SCRL\_RIGHT (0x1c) : 画面を右にスクロール

ATF\_LCD\_SET\_DDRAM (0x80) : DDRAM アドレスをセット

ATF\_LCD\_SET\_CGRAM (0x40) : CGRAM アドレスをセット

LCD の制御レジスタ (インストラクション・レジスタ) に書き込みを行います。

ATF\_LCD\_ENABLE、ATF\_LCD\_CRCSR\_ENABLE、ATF\_LCD\_CRCSR\_BLINK はORで結合して書き込むことができます。

ATF\_LCDWriteData()などでアクセスするRAMアドレスを指定する場合は、ATF\_LCD\_SET\_DDRAM や、ATF\_LCD\_SET\_CGRAM とRAMアドレスをORで結合して書き込みを行います。

インストラクションに関する詳細はLCDのデータシートを参照してください。

---

## ATF\_LCDReadData ()

TW\_STATUS ATF\_LCDReadData(TW\_HANDLE hDev, BYTE \*pData)

Function ATF\_LCDReadData(ByVal hDev As Long, ByRef pData As Byte) As Long

Function ATF\_LCDReadData(ByVal hDev As Integer, ByRef pData As Byte) As Integer

hDev : デバイスのハンドル

pData : [out]読み出した値

表示領域と対応した DDRAM や、キャラクタ定義用の CGRAM から読み出しを行います。

ATF\_LCDWriteReg() で DDRAM アドレスをセットした後のアクセスは DDRAM、CGRAM アドレスを指定した後のアクセスは CGRAM を対象に行われます。

## ATF\_LCDWriteData ()

TW\_STATUS ATF\_LCDWriteData(TW\_HANDLE hDev, BYTE Data)

Function ATF\_LCDWriteData(ByVal hDev As Long, ByVal Data As Byte) As Long

Function ATF\_LCDWriteData(ByVal hDev As Integer, ByVal Data As Byte) As Integer

hDev : デバイスのハンドル

Data : 書き込む値

表示領域と対応した DDRAM や、キャラクタ定義用の CGRAM に書き込みを行います。

ATF\_LCDWriteReg() で DDRAM アドレスをセットした後のアクセスは DDRAM、CGRAM アドレスを指定した後のアクセスは CGRAM を対象に行われます。

## ATF\_LCDPuts ()

TW\_STATUS ATF\_LCDPuts(TW\_HANDLE hDev, LPCTSTR szData)

Function ATF\_LCDPuts(ByVal hDev As Long, ByVal szData As String) As Long

Function ATF\_LCDPuts(ByVal hDev As Integer, ByVal szData As String) As Integer

hDev : デバイスのハンドル

szData : 表示する文字列

LCD に文字列を表示します。

---

## サポート情報

製品に関する情報、最新のファームウェア、ユーティリティなどは弊社ホームページにてご案内しております。また、お問い合わせ、ご質問などは下記までご連絡ください。

**テクノウェーブ(株)**

**URL : <http://www.techw.co.jp>**

**E-mail : [support@techw.co.jp](mailto:support@techw.co.jp)**

改訂記録

年月	版	改訂内容
2007年12月	初	