CASIO

MSR ライブラリマニュアル

このマニュアルは、MSR ライブラリの仕様について記載します。

Ver. 1.00

ご注意

- このソフトウェアおよびマニュアルの、一部または全部を無断で使用、複製することはできません。
- このソフトウェアおよびマニュアルは、本製品の使用許諾契約書のもとでのみ使用することができます。
- このソフトウェアおよびマニュアルを運用した結果の影響については、一切の責任を負いかねますのでご了承ください。
- このソフトウェアの仕様、およびマニュアルに記載されている事柄は、将来予告なしに変更することがあります。
- このマニュアルの著作権はカシオ計算機株式会社に帰属します。
- 本書中に含まれている画面表示は、実際の画面とは若干異なる場合があります。予めご了承ください。

© 2010 カシオ計算機株式会社

Microsoft, MS, ActiveSync, Active Desktop, Outlook, Windows, Windows NT, および Windows ロゴは、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。Microsoft 社の製品は、OEM 各社に、Microsoft Corporation の 100%出資子会社である Microsoft Licensing, Inc.によりライセンス供与されています。

変更履歴

バージョン	変更日付	ページ	内容
0.90	2010.07	―	開発推進版
0.91	2010.07	5	DeviceEmulatorにおけるMSRRead関数の処理を変更
		7	DeviceEmulatorにおけるMSRSetEventNotification関数の処理を変更
		10	DeviceEmulatorにおけるMSRSetPowerStatus関数の処理を変更
1.00	2010.11	10	MSRSetPowerStatus関数の戻り値を追加

目次

1. 概要……1
2. 動作環境……1
3. 機能一覧……2
3.1 関数一覧……2
3.1.1 MSROpen……3
3.1.2 MSRClose……4
3.1.3 MSRRead……5
3.1.4 MSRSetEventNotification……7
3.1.5 MSRGetEventNotification……8
3.1.6 MSRGetLastEventStatus……9
3.1.7 MSRSetPowerStatus……10
3.1.8 MSRGetPowerStatus……11
4. プログラミング上の注意点……12
5. Device Emulator……15
5.1 MSRLib.ini……15

1. 概要

MSR（Magnetic Stripe Reader）ライブラリは、通帳の磁気ストライプからデータを読み取る機能を提供します。

2. 動作環境

MSR ライブラリの動作環境を以下に示します。

対象機種
- DT-5300L30SW-MSR

対象 OS
- Microsoft WindowsMobile 6.5

開発環境
- Microsoft Visual Studio 2005 + SP1
- Microsoft Visual Studio 2008 + SP1

提供ファイル
- MSRLib.h
- MSRLib.lib
- MSRLib.dll
- MSRLibNet.dll

使用方法
- プログラムソース内に MSRLib.h をインクルードし、MSRLib.lib を使用するライブラリとして指定してください
- MSRLib.dll は本体に内蔵されています
- MSRLibNet.dll を実行モジュールと同じフォルダにコピーしてください

3. 機能一覧

3.1 関数一覧

MSR ライブラリの提供する関数は、以下のとおりです。

関数名	機能	DT-5200	DT-X7	DT-9800	DT-5300※
MSROpen	MSR ライブラリの初期化	-	-	-	○
MSRClose	MSR ライブラリの終了処理	-	-	-	○
MSRRead	読取データの取得	-	-	-	○
MSRSetEventNotification	データ取得通知方法の設定	-	-	-	○
MSRGetEventNotification	データ取得通知方法の取得	-	-	-	○
MSRGetLastEventStatus	読取完了ステータスの取得	-	-	-	○
MSRSetPowerStatus	MSR の電源状態の設定	-	-	-	○
MSRGetPowerStatus	MSR の電源状態の取得	-	-	-	○

○ 関数サポート
－ 関数未サポート

※ DT-5300L30SW-MSR のみ

3.1.1 MSROpen

MSR ライブラリを初期化します。

```
[C++]
DWORD MSROpen(
    HWND  hWnd
)
```

```
[Visual Basic]
Public Shared Function MSROpen(
    ByVal hWnd As IntPtr _
) As Int32
```

```
[C#]
public static Int32 MSROpen(
    IntPtr hWnd
)
```

解説

本関数は、MSR ライブラリを初期化します。
本ライブラリを使用する場合は、必ず最初に実行してください。

パラメータ

hWnd
アプリケーションのウィンドウハンドルを指定します。
イベント通知方法がメッセージの場合、指定したウィンドウハンドルに対して、メッセージを送信します。
NULL を指定した場合は、BROADCAST に対してメッセージを送信します。

戻り値

以下の値を返します。

MSR_SUCCESS	: 正常終了
MSR_ERR_OPENED	: オープン済み
MSR_ERR_NOT_ENOUGH_MEMORY	: メモリ不足 DeviceEmulator では発生しません
FUNCTION_UNSUPPORT	: 未サポートエラー

3.1.2 MSRClose

MSR ライブラリの終了処理を実行します。

```
[C++]
DWORD MSRClose()
```

```
[Visual Basic]
Public Shared Function MSRClose() As Int32
```

```
[C#]
public static Int32 MSRClose()
```

解説

本関数は、MSR ライブラリの終了処理を実行します。

パラメータ

なし

戻り値

以下の値を返します。

MSR_SUCCESS	: 正常終了
MSR_ERR_NOTOPEN	: 未オープンエラー
FUNCTION_UNSUPPORT	: 未サポートエラー

3.1.3 MSRRead

MSR で読み取ったデータを取得します。

```
[C++]
DWORD MSRRead(
    BYTE   *pbData,
    DWORD  *pdwLength,
    DWORD  *pdwReserved1,
    DWORD  *pdwReserved2
)
```

```
[Visual Basic]
Public Shared Function MSRRead( _
    ByVal pbData        As Byte(), _
    ByRef pdwLength     As Int32, _
    ByRef pdwReserved1  As Int32, _
    ByRef pdwReserved2  As Int32 _
) As Int32
```

```
[C#]
public static Int32 MSRRead(
    Byte[]     pbData,
    ref Int32  pdwLength,
    ref Int32  pdwReserved1,
    ref Int32  pdwReserved2
)
```

解説

本関数は、MSR で読み取ったデータを取得します。
読取完了時にメッセージ"WM_MSR_READING(WM_USER+0x528)"、またはイベント"MSREventReading"を発行します。
メッセージまたはイベント発行後は、MSRGetLastEventStatus関数を実行して、読取完了ステータスを取得してください。読取完了ステータスが"MSR_DATASUCCESS"の場合、本関数を実行すると読取データを取得することができます。
DeviceEmulatorでは、MSRLib.iniファイルで指定したデータを取得します。iniファイルの詳細については、MSRLib.iniを参照してください。

パラメータ

pbData
磁気ストライプから読み取ったデータ(JIS7 単位符号)を取得します。

pdwLength
磁気ストライプから読み取ったデータの長さを取得します。

pdwReserved1
現在のバージョンではこの引数を使用しません。NULL を指定してください。

pdwReserved2

現在のバージョンではこの引数を使用しません。NULL を指定してください。

戻り値

以下の値を返します。

MSR_SUCCESS	：正常終了
MSR_ERR_NODATA	：取得データなし
MSR_ERR_NOTOPEN	：未オープンエラー
MSR_ERR_PARAM	：パラメータエラー
FUNCTION_UNSUPPORT	：未サポートエラー

補足

MSR使用時の消費電力を抑えるために、読取前にMSRSetPowerStatus関数で電源ONにしてから通帳を走査してください。また、読取後にMSRSetPowerStatus関数で電源OFFにしてください。

3.1.4 MSRSetEventNotification

データ取得通知方法を設定します。

```
[C++]
DWORD MSRSetEventNotification(
    DWORD dwMode
)
```

```
[Visual Basic]
Public Shared Function MSRSetEventNotification(
    ByVal dwMode As Int32 _
) As Int32
```

```
[C#]
public static Int32 MSRSetEventNotification(
    Int32 dwMode
)
```

解説

本関数は、データ取得通知方法を設定します。
データ取得通知方法を変更する場合は、MSRRead関数を実行する前に本関数を使用してください。

■ ウィンドウメッセージ通知
WM_MSR_READING（WM_USER+0x528）のウィンドウメッセージを指定したウィンドウハンドルに対して送信します。

■ イベント通知
名前付きイベント"MSREventReading"を発行します。

DeviceEmulatorでは、設定にかかわらずウィンドウメッセージを送信します。詳細は、MSRSetPowerStatus関数を参照してください。

パラメータ

dwMode
データ取得通知方法を指定します。
MSR_NOTIFICATION_MSG ：ウィンドウメッセージ通知（デフォルト）
MSR_NOTIFICATION_EVENT ：イベント通知

戻り値

以下の値を返します。
MSR_SUCCESS ：正常終了
MSR_ERR_NOTOPEN ：未オープンエラー
MSR_ERR_PARAM ：パラメータエラー
FUNCTION_UNSUPPORT ：未サポートエラー

3.1.5 MSRGetEventNotification

データ取得通知方法を取得します。

```
[C++]
DWORD MSRGetEventNotification(
    DWORD *pdwMode
)
```

```
[Visual Basic]
Public Shared Function MSRGetEventNotification(
    ByRef pdwMode As Int32 _
) As Int32
```

```
[C#]
public static Int32 MSRGetEventNotification(
    ref Int32 pdwMode
)
```

解説

本関数は、データ取得通知方法を取得します。

パラメータ

pdwMode

データ取得通知方法を取得します。取得する値の詳細については、MSRSetEventNotification 関数を参照してください。

戻り値

以下の値を返します。

MSR_SUCCESS	: 正常終了
MSR_ERR_NOTOPEN	: 未オープンエラー
MSR_ERR_PARAM	: パラメータエラー
FUNCTION_UNSUPPORT	: 未サポートエラー

3.1.6 MSRGetLastEventStatus

読取完了ステータスを取得します。

```
[C++]
DWORD MSRGetLastEventStatus(
    DWORD *pMode
)
```

```
[Visual Basic]
Public Shared Function MSRGetLastEventStatus(
    ByRef pMode As Int32 _
) As Int32
```

```
[C#]
public static Int32 MSRGetLastEventStatus(
    ref Int32 pMode
)
```

解説

本関数は、読取完了ステータスを取得します。
DeviceEmulatorでは、MSRLib.iniファイルの有無によりMSRの状態を取得します。iniファイルが存在する場合はMSR_DATASUCCESSを、存在しない場合はMSR_NODATAを返します。

パラメータ

pMode
読取完了ステータスを取得します。

MSR_DATASUCCESS	: 取得データあり
MSR_NODATA	: 取得データなし
MSR_READERROR	: リードエラー
MSR_BUFFOVER	: バッファ許容量オーバーエラー
MSR_PARITYERROR	: パリティエラー

戻り値

以下の値を返します。

MSR_SUCCESS	: 正常終了
MSR_ERR_NOTOPEN	: 未オープンエラー
MSR_ERR_PARAM	: パラメータエラー
FUNCTION_UNSUPPORT	: 未サポートエラー

3.1.7　MSRSetPowerStatus

MSR の電源状態を設定します。

```
[C++]
DWORD MSRSetPowerStatus(
    DWORD dwPowerStatus
)
```

```
[Visual Basic]
Public Shared Function MSRSetPowerStatus(
    ByVal dwPowerStatus As Int32 _
) As Int32
```

```
[C#]
public static Int32 MSRSetPowerStatus(
    Int32 dwPowerStatus
)
```

解説

本関数は、MSR の電源状態を設定します。
DeviceEmulator では、電源 ON にした 1 秒後に WM_MSR_READING メッセージを送信します。

パラメータ

dwPowerStatus
電源状態を指定します。

MSR_POWERON	: 電源 ON
MSR_POWEROFF	: 電源 OFF

戻り値

以下の値を返します。

MSR_SUCCESS	: 正常終了
MSR_ERR_NOTOPEN	: 未オープンエラー
MSR_ERR_PARAM	: パラメータエラー
MSR_ERR_DEVICE	: デバイスが正しく動作していません DeviceEmulator では発生しません
MSR_ERR_POWERSUPPLIED	: MSR に電源が既に供給されています DeviceEmulator では発生しません
FUNCTION_UNSUPPORT	: 未サポートエラー

補足

MSR 使用時の消費電力を抑えるために、読取前に本関数で電源 ON にしてから通帳を走査してください。また、読取後に本関数で電源 OFF にしてください。

3.1.8 MSRGetPowerStatus

MSR の電源状態を取得します。

```
[C++]
DWORD MSRGetPowerStatus(
    DWORD *pdwPowerStatus
)
```

```
[Visual Basic]
Public Shared Function MSRGetPowerStatus(
    ByRef pdwPowerStatus As Int32 _
) As Int32
```

```
[C#]
public static Int32 MSRGetPowerStatus(
    ref Int32 pdwPowerStatus
)
```

解説

本関数は、MSR の電源状態を取得します。

パラメータ

pdwPowerStatus

電源状態を取得します。取得する値の詳細については、MSRSetPowerStatus関数を参照してください。

戻り値

以下の値を返します。

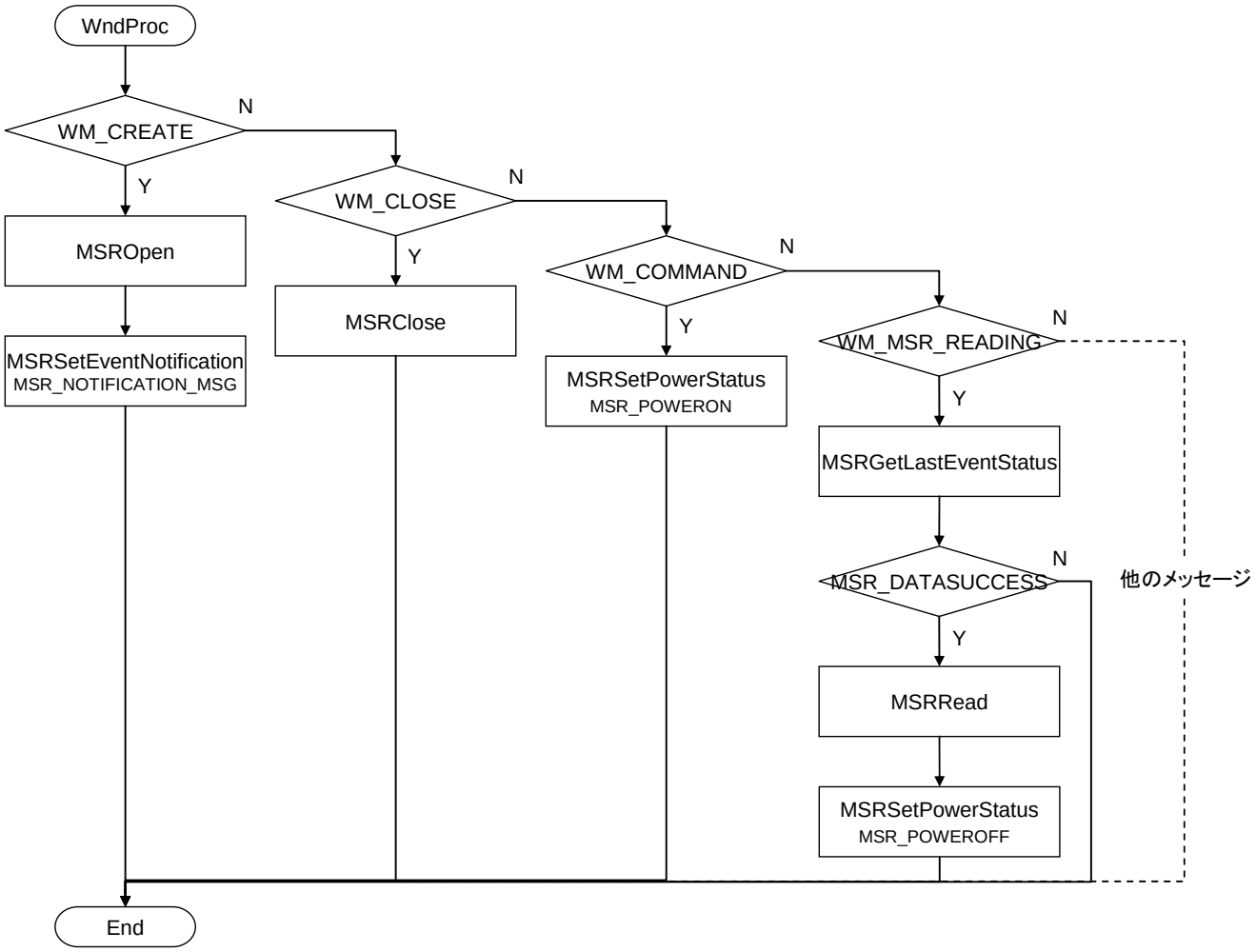
MSR_SUCCESS	: 正常終了
MSR_ERR_NOTOPEN	: 未オープンエラー
MSR_ERR_PARAM	: パラメータエラー
FUNCTION_UNSUPPORT	: 未サポートエラー

4. プログラミング上の注意点

本ライブラリを使用して磁気ストライプを読み取る手順を以下に示します。

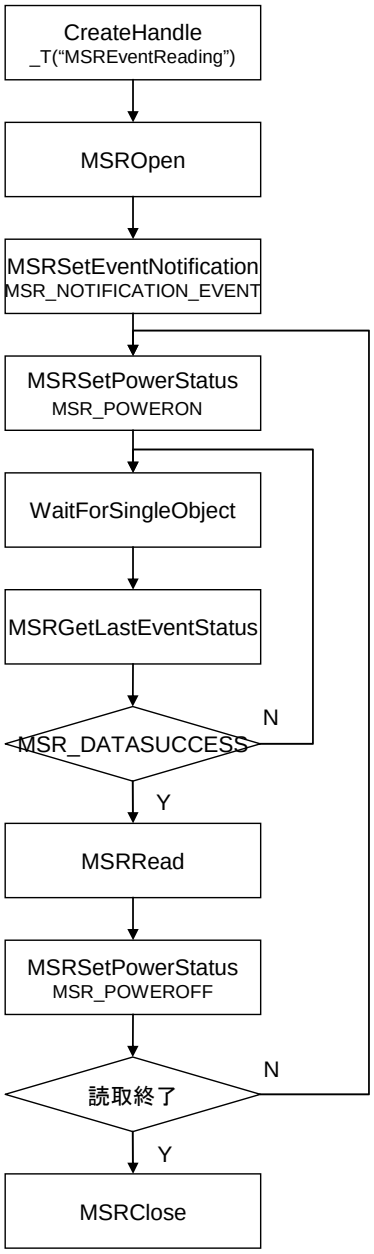
1. MSROpen関数により、本ライブラリを初期化します。
2. MSRSetEventNotification関数により、データ取得通知方法を設定します。
3. MSRSetPowerStatus関数により、MSRの電源をONにします。これにより、MSRは磁気ストライプの読取待ちの状態になります。
4. 磁気ストライプをスキャンします。
5. 2.で設定したウィンドウメッセージ（WM_MSR_READING）またはイベント（MSREventReading）を待機します。
6. MSRGetLastEventStatus関数により、MSRの読取完了ステータスを取得します。読取完了ステータスが取得データあり（MSR_DATASUCCESS）以外の場合は 4.からやり直してください。
7. MSRRead関数により、MSRのデータを取得します。
8. MSRSetPowerStatus関数により、MSRの電源をOFFにします。
9. MSRClose関数により、本ライブラリの終了処理を実行します。

ウィンドウメッセージ通知を使用する場合



※ 上記の手順 3.は任意の位置で実行してください。本フローでは WM_COMMAND メッセージの処理で実行しています。同様に、上記の手順 8.についても任意の位置で実行してください。

イベント通知を使用する場合



※ 上記の手順 3.は任意の位置で実行してください。同様に、上記の手順 8.についても任意の位置で実行してください。

5. Device Emulator

5.1 MSRLib.ini

MSR の読取データを格納します。
MSRLib.ini のサンプルを以下に示します。

```
[Read]
DATA="0123456789"
```

[Read]		読取データセクション
	DATA	読取データを文字列で指定します。(参照:MSRRead関数)

カシオ計算機お問い合わせ窓口

製品に関する最新情報

●製品サポートサイト（カシオペア・ハンディターミナル）

http://casio.jp/support/pa/

製品の取扱い方法のお問い合わせ

●情報機器コールセンター

0570-022066
市内通話料金でご利用いただけます。
携帯電話・PHS 等をご利用の場合、**048-233-7241**

カシオ計算機株式会社
〒151-8543　東京都渋谷区本町 1-6-2
TEL 03-5334-4638（代）