



# MCR ライブラリマニュアル

このマニュアルは、MCR ライブラリの仕様について記載します。

## 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。

© 2011 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。



# 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 MCROpen .....	3
3.1.2 MCRClose.....	4
3.1.3 MCRRead.....	5
3.1.4 MCRReadRaw .....	8
3.1.5 MCRSelectTrack.....	11
3.1.6 MCRSetEventNotification .....	12
3.1.7 MCRGetEventNotification .....	13
3.1.8 MCRGetStatus .....	14
3.1.9 MCRSetAutoPowerOff .....	15
3.1.10 MCRGetAutoPowerOff.....	16
3.1.11 MCRResume .....	17
3.1.12 MCRGetModelInfo .....	18
3.1.13 MCRGetDataLength.....	19
4. プログラミング上の注意点.....	21
5. Device Emulator.....	22
5.1 MCRLib.ini .....	22

## 1. 概要

MCR ライブラリは、磁気カードリーダーを使用するための機能を提供します。

## 2. 動作環境

MCR ライブラリの動作環境を以下に示します。

### 対象機種

- IT-9000

### 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

- MCRReadLib.h
- MCRReadLib.lib
- MCRReadLib.dll
- MCRReadLibNet.dll (クラスライブラリ)

### 使用方法

- プログラムソース内に MCRReadLib.h をインクルードし、MCRReadLib.lib を使用するライブラリとして指定してください
- MCRReadLib.dll は本体に内蔵されています。
- MCRReadLibNet.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

MCR 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	IT-9000
MCROpen	MCR を使用可能にします	
MCRClose	MCR の使用を終了します	
MCRRead	MCR で読み取ったデータを取得します	
MCRReadRaw	MCR で読み取った生データを取得します	
MCRSelectTrack	データ取得を行うトラックを選択します	
MCRSetEventNotification	データ取得通知方法を設定します	
MCRGetEventNotification	データ取得通知方法を取得します	
MCRGetStatus	MCR の状態を取得します	
MCRSetAutoPowerOff	一定時間後に MCR の電源を自動的に OFF にします	
MCRGetAutoPowerOff	設定している自動電源 OFF までの時間を取得します	
MCRResume	自動電源 OFF から ON に復帰します 自動電源 OFF タイマーのカウントをクリアします	
MCRGetModelInfo	MCR のモデル情報を取得します	
MCRGetDataLength	各トラックのデータ長を取得します	

関数サポート

- 関数未サポート

### 3.1.1 MCROpen

MCR を使用可能にします。

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

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

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

#### 説明

本関数は MCR の初期化および電源投入を行います。  
MCR ライブラリを使用する場合は、はじめに必ず本関数を実行してください。  
DeviceEmulator では、常に TRUE を返します。

#### 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。  
データ通知方法をメッセージに設定している場合、この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、メッセージは BROADCAST に送信します。

#### 戻り値

正常終了時は TRUE を、異常終了時は FALSE を返します。

### 3.1.2 MCRClose

MCR の使用を終了します。

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

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

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

#### 説明

本関数は MCR で使用していたリソースの開放や電源 OFF 等の終了処理を行います。  
MCR の使用を終了する場合は、必ず本関数を実行してください。  
DeviceEmulator では、常に TRUE を返します。

#### パラメータ

なし

#### 戻り値

正常終了時は TRUE を、異常終了時は FALSE を返します。



### 3.1.3 MCRRead

MCR で読み取ったデータを取得します。

```
[C++]
DWORD MCRRead (
    UCHAR    DataJIS2[MCR_JIS2SIZE],
    UCHAR    DataISO1[MCR_ISO1SIZE],
    UCHAR    DataISO2[MCR_ISO2SIZE],
    UCHAR    DataISO3[MCR_ISO3SIZE],
    int      *StatusJIS2,
    int      *StatusISO1,
    int      *StatusISO2,
    int      *StatusISO3
)
```

```
[Visual Basic]
Public Shared Function MCRRead (
    ByVal    DataJIS2 As Byte(), _
    ByVal    DataISO1 As Byte(), _
    ByVal    DataISO2 As Byte(), _
    ByVal    DataISO3 As Byte(), _
    ByRef    StatusJIS2 As Int32, _
    ByRef    StatusISO1 As Int32, _
    ByRef    StatusISO2 As Int32, _
    ByRef    StatusISO3 As Int32 _
) As Int32
```

```
[C#]
public static Boolean MCRRead (
    out byte[] DataJIS2,
    out byte[] DataISO1,
    out byte[] DataISO2,
    out byte[] DataISO3,
    out Int32 StatusJIS2,
    out Int32 StatusISO1,
    out Int32 StatusISO2,
    out Int32 StatusISO3
)
```

#### 説明

本関数は MCR で読み取ったデータを取得します。

MCR はデータ読み取り完了時にメッセージ、またはイベントを発行するので、この通知を受けた後、本関数を実行すると読み取りデータを取得することができます。

DeviceEmulator では、MCRLib.iniファイルで指定したデータを取得し、ステータスを MCR\_DATA\_SUCCESS にします。また、iniファイルが存在しない場合は、FALSEを返し、ステータスを MCR\_NO\_DATA にします。iniファイルの詳細については、MCRLib.iniを参照してください。

#### パラメータ

#### *DataJIS2*

69 バイト分の RAM 領域をアプリケーションで確保し、その先頭アドレスを指定します。  
JIS2 データの取得が不要な場合は、NULL を指定してください (RAM の確保不要)。

#### *DataISO1*

76 バイト分の RAM 領域をアプリケーションで確保し、その先頭アドレスを指定します。  
ISO1 データの取得が不要な場合は、NULL を指定してください (RAM の確保不要)。

#### *DataISO2*

37 バイト分の RAM 領域をアプリケーションで確保し、その先頭アドレスを指定します。  
ISO2 データの取得が不要な場合は、NULL を指定してください (RAM の確保不要)。

#### *DataISO3*

104 バイト分の RAM 領域をアプリケーションで確保し、その先頭アドレスを指定します。  
ISO3 データの取得が不要な場合は、NULL を指定してください (RAM の確保不要)。

#### *StatusJIS2*

JIS2 トラックのステータスを取得するポインタを指定します。  
JIS2 トラックのステータスが不要な場合は、NULL を指定してください。

#### *StatusISO1*

ISO1 トラックのステータスを取得するポインタを指定します。  
ISO1 トラックのステータスが不要な場合は、NULL を指定してください。

#### *StatusISO2*

ISO2 トラックのステータスを取得するポインタを指定します。  
ISO2 トラックのステータスが不要な場合は、NULL を指定してください。

#### *StatusISO3*

ISO3 トラックのステータスを取得するポインタを指定します。  
ISO3 トラックのステータスが不要な場合は、NULL を指定してください。

### 戻り値

正常終了時は TRUE を、異常終了時は FALSE を返します。

### 補足

読み取ったデータは、DataJIS2[0] ~ [68]、DataISO1[0] ~ [75]、DataISO2[0] ~ [36]、  
DataISO3[0] ~ [103]に格納します。

エラーが発生した場合、エラーが発生したバッファはデータを格納せず、ステータス取得変数に  
エラー情報を格納します (ステータス情報は下記参照)。

読み取ったカードデータは MCR ドライバ/ライブラリが保持していますが、本関数を実行するとクリアします。

データが入っていないトラックのステータスには“MCR\_STX\_NOTFOUND”が入ります。

読み取り可能なトラック (全 3 トラック) は MCR モデル毎に異なります。

1. CA/MA/GE モデル (JIS2 対応モデル)  
読み取り可能トラック : JIS2 / ISO1 / ISO2 (ISO3 未使用)
2. HE/PE/PU モデル (ISO3 対応モデル)  
読み取り可能トラック : ISO1 / ISO2 / ISO3 (JIS2 未使用)

## ステータス詳細

名称	説明	詳細
MCR_NO_DATA	データ未取得	MCR がデータを取得していない状態
MCR_DATA_SUCCESS	有効データ取得済み	MCR が有効データを取得した状態
MCR_BUFFER_FULL_ERR	バッファフルエラー	規定量以上のデータ取得
MCR_DATASHORT_ERR	Min.データエラー	必要最低限のデータが無い場合
MCR_DATA_ERR1	プリアンプルエラー	"1"連続によりデータ取り込み中止
MCR_DATA_ERR2	データ異常	"1"連続によりデータ取り込み中止
MCR_PULSE_ERR1	Min.パルスエラー	Min.値以下のパルス検出
MCR_PULSE_ERR2	Max.パルスエラー	Max.値以上のパルス検出
MCR_TIMEOUT_ERR	タイムアウトエラー	カード走行時間がタイムアウト
MCR_CARDSPEED_ERR1	加速エラー	カード走査中の異常加速
MCR_CARDSPEED_ERR2	減速エラー	カード走査中の異常減速
MCR_PARITY_ERR	キャラクタパリティエラー	取得データのパリティエラー
MCR_STX_NOTFOUND	STX 検出エラー	STX が検出できない
MCR_ETX_NOTFOUND	ETX 検出エラー	ETX が検出できない
MCR_PACKET_LRC_ERR	パケット LRC エラー	パケットの LRC エラー

DeviceEmulator では、MCR\_NO\_DATA および MCR\_DATA\_SUCCESS のみを取得します。

### 3.1.4 MCRReadRaw

MCR で読み取った生データを取得します。

```
[C++]
DWORD MCRReadRaw (
    UCHAR    DataJIS2[MCR_RAWSIZE],
    UCHAR    DataISO1[MCR_RAWSIZE],
    UCHAR    DataISO2[MCR_RAWSIZE],
    UCHAR    DataISO3[MCR_RAWSIZE],
    int      *StatusJIS2,
    int      *StatusISO1,
    int      *StatusISO2,
    int      *StatusISO3
)
```

```
[Visual Basic]
Public Shared Function MCRReadRaw (
    ByVal    DataJIS2 As Byte(), _
    ByVal    DataISO1 As Byte(), _
    ByVal    DataISO2 As Byte(), _
    ByVal    DataISO3 As Byte(), _
    ByRef    StatusJIS2 As Int32, _
    ByRef    StatusISO1 As Int32, _
    ByRef    StatusISO2 As Int32, _
    ByRef    StatusISO3 As Int32 _
) As Int32
```

```
[C#]
public static Boolean MCRReadRaw (
    out byte[] DataJIS2,
    out byte[] DataISO1,
    out byte[] DataISO2,
    out byte[] DataISO3,
    out Int32 StatusJIS2,
    out Int32 StatusISO1,
    out Int32 StatusISO2,
    out Int32 StatusISO3
)
```

#### 説明

本関数は MCR で読み取った生データを取得します。

MCR はデータ読み取り完了時にメッセージ、またはイベントを発行するので、この通知を受けた後、本関数を実行すると読み取りデータを取得することができます。

DeviceEmulator では、MCRLib.iniファイルで指定したデータを取得し、ステータスを MCR\_DATA\_SUCCESS にします。また、ini ファイルが存在しない場合は、FALSE を返し、ステータスを MCR\_NO\_DATA にします。ini ファイルの詳細については、MCRLib.iniを参照してください。

#### パラメータ

#### *DataJIS2*

92 バイト分の RAM 領域をアプリケーションで確保し、その先頭アドレスを指定します。  
JIS2 生データの取得が不要な場合は、NULL を指定してください (RAM の確保不要)。

#### *DataISO1*

92 バイト分の RAM 領域をアプリケーションで確保し、その先頭アドレスを指定します。  
ISO1 生データの取得が不要な場合は、NULL を指定してください (RAM の確保不要)。

#### *DataISO2*

92 バイト分の RAM 領域をアプリケーションで確保し、その先頭アドレスを指定します。  
ISO2 生データの取得が不要な場合は、NULL を指定してください (RAM の確保不要)。

#### *DataISO3*

92 バイト分の RAM 領域をアプリケーションで確保し、その先頭アドレスを指定します。  
ISO3 生データの取得が不要な場合は、NULL を指定してください (RAM の確保不要)。

#### *StatusJIS2*

JIS2 トラックのステータスを取得するポインタを指定します。  
JIS2 トラックのステータスが不要な場合は、NULL を指定してください。

#### *StatusISO1*

ISO1 トラックのステータスを取得するポインタを指定します。  
ISO1 トラックのステータスが不要な場合は、NULL を指定してください。

#### *StatusISO2*

ISO2 トラックのステータスを取得するポインタを指定します。  
ISO2 トラックのステータスが不要な場合は、NULL を指定してください。

#### *StatusISO3*

ISO3 トラックのステータスを取得するポインタを指定します。  
ISO3 トラックのステータスが不要な場合は、NULL を指定してください。

### 戻り値

正常終了時は TRUE を、異常終了時は FALSE を返します。

### 補足

読み取ったデータは、DataJIS2[0] ~ [91]、DataISO1[0] ~ [91]、DataISO2[0] ~ [91]、DataISO3[0] ~ [91]に格納します。

エラーが発生した場合、エラーが発生したバッファはデータを格納せず、ステータス取得変数にエラー情報を格納します (ステータス情報は下記参照)。

読み取ったカードデータは MCR ドライバ/ライブラリが保持していますが、本関数を実行するとクリアします。

データが入っていないトラックのステータスには“MCR\_STX\_NOTFOUND”が入ります。

読み取り可能なトラック (全 3 トラック) は MCR モデル毎に異なります。

1. CA/MA/GE モデル (JIS2 対応モデル)  
読み取り可能トラック : JIS2 / ISO1 / ISO2 (ISO3 未使用)
2. HE/PE/PU モデル (ISO3 対応モデル)  
読み取り可能トラック : ISO1 / ISO2 / ISO3 (JIS2 未使用)

## ステータス詳細

名称	説明	詳細
MCR_NO_DATA	データ未取得	MCR がデータを取得していない状態
MCR_DATA_SUCCESS	有効データ取得済み	MCR が有効データを取得した状態
MCR_BUFFER_FULL_ERR	バッファフルエラー	規定量以上のデータ取得
MCR_DATASHORT_ERR	Min.データエラー	必要最低限のデータが無い場合
MCR_DATA_ERR1	プリアンプルエラー	"1"連続によりデータ取り込み中止
MCR_DATA_ERR2	データ異常	"1"連続によりデータ取り込み中止
MCR_PULSE_ERR1	Min.パルスエラー	Min.値以下のパルス検出
MCR_PULSE_ERR2	Max.パルスエラー	Max.値以上のパルス検出
MCR_TIMEOUT_ERR	タイムアウトエラー	カード走行時間がタイムアウト
MCR_CARDSPEED_ERR1	加速エラー	カード走査中の異常加速
MCR_CARDSPEED_ERR2	減速エラー	カード走査中の異常減速

### 3.1.5 MCRSelectTrack

データ取得を行うトラックを選択します。

```
[C++]
DWORD MCRSelectTrack (
    DWORD dwTargetTrack
)
```

```
[Visual Basic]
Public Shared Function MCRSelectTrack ( _
    ByVal dwTargetTrack As Int32 _
) As Int32
```

```
[C#]
public static Int32 MCRSelectTrack (
    Int32 dwTargetTrack
)
```

#### 説明

本関数はデータ取得を行うトラックを選択します。

本関数を使用してトラックを限定する場合は、MCRRead関数を実行する前に本関数を使用してください。

本関数を使用しない場合や本関数で ALL\_TRACK を指定した場合は、3トラックすべてのデータを取得します。

#### パラメータ

*dwTargetTrack*

データ取得を行うトラックを指定します。

ISO1_TRACK	: ISO1 フォーマットのトラックのみ読む場合
ISO2_TRACK	: ISO2 フォーマットのトラックのみ読む場合
ISO3_TRACK	: ISO3 フォーマットのトラックのみ読む場合
JIS2_TRACK	: JIS-II フォーマットのトラックのみ読む場合
ISO1_ISO2_TRACK	: ISO1 と ISO2 フォーマットのトラックを読む場合
ISO1_JIS2_TRACK	: ISO1 と JIS-II フォーマットのトラックを読む場合
ISO2_JIS2_TRACK	: ISO2 と JIS-II フォーマットのトラックを読む場合
ISO1_ISO3_TRACK	: ISO1 と ISO3 フォーマットのトラックを読む場合
ISO2_ISO3_TRACK	: ISO2 と ISO3 フォーマットのトラックを読む場合
ALL_TRACK	: 全トラックを読む場合

#### 戻り値

正常終了時は TRUE を、異常終了時は FALSE を返します。

### 3.1.6 MCRSetEventNotification

データ取得通知方法を設定します。

```
[C++]
DWORD MCRSetEventNotification (
    BOOL bEnable
)
```

```
[Visual Basic]
Public Shared Function MCRSetEventNotification ( _
    ByVal bEnable As Boolean _
) As Int32
```

```
[C#]
public static Int32 MCRSetEventNotification (
    Boolean bEnable
)
```

#### 説明

本関数はデータ取得通知方法を設定します。

本関数を使用してデータ取得通知方法を変更する場合は、MCRRead関数を実行する前に本関数を使用してください。

通知方法をイベントに設定した場合、名前付きイベント“MCREventReading”でデータ取得を通知します。

また、自動電源 OFF 機能を使用している場合は“MCREventAPO”で OFF 状態に遷移したことを通知します。

通知方法をメッセージに設定した場合、メッセージを発行します(メッセージは MCRLIB.h で定義しています)。

WM\_MCR\_READING : MCR 読み取り完了通知  
WM\_MCR\_APO : MCR 自動電源 OFF 通知

DeviceEmulator では、設定値を内部変数として格納するため、何も動作しませんが、MCRGetEventNotification関数を実行することにより、設定値を確認することができます。

#### パラメータ

*bEnable*

データ通知方法を指定します。

TRUE : 名前付きイベントに設定  
FALSE : メッセージに設定

#### 戻り値

正常終了時は TRUE を、異常終了時は FALSE を返します。



### 3.1.7 MCRGetEventNotification

データ取得通知方法を取得します。

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

```
[Visual Basic]  
Public Shared Function MCRGetEventNotification ( ) As In32
```

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

#### 説明

本関数はデータ取得通知方法を取得します。

#### パラメータ

なし

#### 戻り値

名前付きイベント設定時は TRUE を、メッセージ設定時は FALSE を返します。

### 3.1.8 MCRGetStatus

MCR の状態を取得します。

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

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

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

#### 説明

本関数は MCR の状態を取得します。

DeviceEmulator では、MCRLib.ini ファイルの有無により MCR の状態を取得します。ini ファイルが存在する場合は MCR\_DATA\_SUCCESS を、存在しない場合は MCR\_NO\_DATA を返します。

#### パラメータ

なし

#### 戻り値

MCR の状態を返します。

MCR_NO_DATA	: 取得データなし
MCR_DATA_SUCCESS	: 取得データあり
MCR_AUTO_POWER_OFF	: オートパワーOFF 状態 DeviceEmulator では発生しません
MCR_DATA_ERROR	: データ取得エラー DeviceEmulator では発生しません

#### 注意

イベント/メッセージによる読み取り/解析終了通知を受け取れない場合は、本関数を使用して MCR の状態をポーリングしてください。

### 3.1.9 MCRSetAutoPowerOff

一定時間後に MCR の電源を自動的に OFF にします。

```
[C++]
DWORD MCRSetAutoPowerOff (
    DWORD dwTime
)
```

```
[Visual Basic]
Public Shared Function MCRSetAutoPowerOff (
    ByVal dwTime As Int32 _
) As Int32
```

```
[C#]
public static Int32 MCRSetAutoPowerOff (
    Int32 dwTime
)
```

#### 説明

本関数は一定時間後に MCR の電源を自動的に OFF にします。

自動電源 OFF 状態になった後、再度 MCR を使用する場合は MCRResume 関数を使用して、電源を ON にしてください。

DeviceEmulator では、設定値を内部変数として格納するため、何も動作しませんが、MCRGetAutoPowerOff 関数を実行することにより、設定値を確認することができます。

#### パラメータ

*dwTime*

自動電源 OFF までの時間 (秒) を 0 ~ 300 の範囲で指定します。

0 を指定すると、自動電源 OFF 機能を解除します。

#### 戻り値

正常終了時は TRUE を、異常終了時は FALSE を返します。

#### 注意

本関数を使用すると、MCR の電源投入 (MCROpen 関数) や MCR の読み取り終了 (イベントまたはメッセージ発行) から自動電源 OFF のカウントダウンを開始し、設定時間経過後に MCR の電源を自動的に OFF にします。MCRResume 関数を実行すると、MCR の電源を再投入し、設定時間から再びカウントダウンを開始します。

電源自動 OFF 状態になるとドライバはメッセージまたはイベントによる通知を行います。初期設定ではメッセージ通知になっており、“WM\_MCR\_APO” を発行します。イベント通知の設定を変更した場合、名前付きイベント “MCREventAPO” で通知します。通知方法の切り替えは MCRSetEventNotification 関数で行ってください。

### 3.1.10 MCRGetAutoPowerOff

MCRSetAutoPowerOff関数で設定した自動電源 OFF までの時間を取得します。

```
[C++]  
DWORD MCRGetAutoPowerOff (  
    DWORD *dwTime  
)
```

```
[Visual Basic]  
Public Shared Function MCRGetAutoPowerOff ( _  
    ByRef dwTime As Int32 _  
) As Int32
```

```
[C#]  
public static Int32 MCRGetAutoPowerOff (  
    ref Int32 dwTime  
)
```

#### 説明

本関数はMCRSetAutoPowerOff関数で設定した自動電源 OFF までの時間を取得します。

#### パラメータ

*dwTime*

自動電源 OFF までの時間(秒)を取得する変数のポインタを指定します。

#### 戻り値

終了時は TRUE を返します。

### 3.1.11 MCRResume

自動電源 OFF から ON に復帰します。また、自動電源 OFF タイマーのカウントをクリアします。

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

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

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

#### 説明

本関数は自動電源 OFF している MCR の電源を再投入します。また、MCRSetAutoPowerOff関数で設定した値までカウントダウンを戻します。

DeviceEmulator では、何も動作しません。

#### パラメータ

なし

#### 戻り値

終了時は TRUE を返します。

### 3.1.12 MCRGetModelInfo

MCR のモデル情報を取得します。

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

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

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

#### 説明

本関数は MCR のモデル情報を取得します。

読み取りデータを取得できない場合は、本関数を実行して、MCR のモデル情報を確認してください。

DeviceEmulator では、MCRLib.ini ファイルの情報を読み込みます。ini ファイルが存在しない場合は、MCR\_MODEL\_INVALID を返します。詳細はMCRLib.iniを参照してください。

#### パラメータ

なし

#### 戻り値

MCR のモデル情報を返します。

MCR_MODEL_JIS2	: ISO1/ISO2/JIS2 対応モデル
MCR_MODEL_ISO3	: ISO1/ISO2/ISO3 対応モデル
MCR_MODEL_INVALID	: モデル種不明

### 3.1.13 MCRGetDataLength

各トラックのデータ長を取得します。

```
[C++]
DWORD MCRGetDataLength (
    BYTE *DataLenJIS2,
    BYTE *DataLenISO1,
    BYTE *DataLenISO2,
    BYTE *DataLenISO3
)
```

```
[Visual Basic]
Public Shared Function MCRGetDataLength ( _
    ByRef DataLenJIS2 As Byte, _
    ByRef DataLenISO1 As Byte, _
    ByRef DataLenISO2 As Byte, _
    ByRef DataLenISO3 As Byte _
) As Int32
```

```
[C#]
public static Int32 MCRGetDataLength (
    out byte DataLenJIS2,
    out byte DataLenISO1,
    out byte DataLenISO2,
    out byte DataLenISO3
)
```

#### 説明

本関数は各トラックの有効データ長を取得をします。

DeviceEmulator では、MCRLib.ini ファイルのスキャンデータセクション内の各文字列長を取得します。ini ファイルが存在しない場合は、0 を取得します。詳細はMCRLib.iniを参照してください。

#### パラメータ

*DataLenJIS2*

JIS2 トラックの有効データ長を取得する変数のポインタを指定します。

*DataLenISO1*

ISO1 トラックの有効データ長を取得する変数のポインタを指定します。

*DataLenISO2*

ISO2 トラックの有効データ長を取得する変数のポインタを指定します。

*DataLenISO3*

ISO3 トラックの有効データ長を取得する変数のポインタを指定します。

#### 戻り値

終了時は TRUE を返します。

## 注意

読み取ったデータ長は MCR ドライバ/ライブラリが保持していますが、本関数が実行するとクリアします。

下記の条件では、データ長は 0 になります。

1. MCR モデルごとに使用しないトラック
2. MCRSelectTrack関数で指定していないトラック
3. エラーが発生したトラック
4. 本関数でデータ長を取得し、カードを通さずに再度本関数を実行した場合  
本注意事項は、DeviceEmulator では発生しません。



## 4. プログラミング上の注意点

### データ取得までの流れ

MCROpen関数を実行し、MCR を使用可能状態にします。

磁気カードの読み取りが完了すると、本体からアプリケーションに対してメッセージ、またはイベントで通知します。

アプリケーションがデータを取得する場合は、上記のメッセージ、またはイベントによる通知を受けてから、MCRRead関数を実行します。

MCRRead関数実行により、データ、またはエラー情報を取得します。

MCR の使用を終了する場合は、MCRClose関数を実行します。

磁気カードを通す前は、データが存在していないため、MCRRead関数を実行してもデータを取得することはできません。

### データ取得通知方法

MCR ドライバはデータの取得、および自動電源 OFF した際に、メッセージまたはイベントを発行します(初期設定はメッセージ通知になっています)。

通知方法をメッセージに設定している場合、データ取得時に“WM\_MCR\_READING (WM\_USER+0x520) ”、自動電源 OFF 時に“WM\_MCR\_APO (WM\_USER+0x521) ”のメッセージを発行します。

MCRSetEventNotification関数を使用して通知方法をイベントに設定すると、名前付きイベントによる通知を行います。名前付きイベントの名称はデータ取得時が“MCREventReading”、自動電源 OFF 時が“MCREventAPO”になっています。

名前付きイベントのイベント名称は端末のレジストリで設定されており、このレジストリを変更することにより、イベント名を変更することができます。

名前付きイベントの名称が設定されているレジストリは以下の場所です。

```
[HKEY_LOCAL_MACHINE¥Drivers¥BuiltIn¥MCRDRV]
"EventName"="MCREventReading" : データ取得時のイベント
"OffEventName"="MCREventAPO" : 自動電源 OFF 時のイベント
```

## 5. Device Emulator

Device Emulator 上で本ライブラリを正しく動作させるためには、以下の設定ファイルが必要です。

MCRLib.ini

設定ファイルはあらかじめ以下のフォルダにインストールされています。

¥Storage Card¥MCR

### 5.1 MCRLib.ini

MCR のモデル情報およびスキャンデータを格納します。

MCRLib.ini のサンプルを以下に示します。

```
[Model]
Info=0
[Scan]
JIS2="0123456789ABCDEF"
ISO1="0123456789ABCDEF"
ISO2="0123456789ABCDEF"
ISO3="0123456789ABCDEF"
```

[Model]      モデル情報セクション	
Info	MCR のモデル情報を指定します。(参照:MCRGetModelInfo関数) 0 : ISO1/ISO2/JIS2 対応モデル 1 : ISO1/ISO2/ISO3 対応モデル 2 : モデル種不明
[Scan]      スキャンデータセクション	
JIS2	JIS2 スキャンデータを文字列で指定します。(参照:MCRRead関数)
ISO1	ISO1 スキャンデータを文字列で指定します。(参照:MCRRead関数)
ISO2	ISO2 スキャンデータを文字列で指定します。(参照:MCRRead関数)
ISO3	ISO3 スキャンデータを文字列で指定します。(参照:MCRRead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/ht/>

### 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)