

USB-PIO 8/16-**-FT シリーズ USB-RLSW-2C***-FT シリーズ USB-RLSW-*RSMB-FT シリーズ USB-TIM-AC10A*-FT シリーズ

ライブラリ関数&コマンド リファレンス

重要事項

本マニュアルはシステムサコム工業株式会社製の USB I/O 機器(USB-PIO 8/16-**-FT シリーズ, USB-RLSW-2C***-FT シリーズ, USB-RLSW-*RSMB-FT シリーズ, USB-TIM-AC10A*-FT シリーズ)をお使いいただくために必要なソフトウェアの開発および使用方法に関して解説します。

本マニュアルに記載されたソフトウェアをお使いになる際は、すでに製品付属のデバイスドライバソフトウェアのインストールが完了していることを前提にしています。デバイスドライバソフトウェアのインストールについては別冊の USB デバイスドライバインストールマニュアルをご参照ください。またインストール手順はお使いの PC にインストールされた Windows の詳細なバージョン、マイクロソフト社による修正パッチ、サービスパックの違いや、すでにインストールされているソフトウェアなどにより、表示される内容が異なる場合があります。お使いの PC の個別の環境による違いについて説明は再現も難しく非常に困難ですで行っておりません。その際にご必要に応じてマイクロソフト等へお問い合わせ願います。当社製品に直接関するご質問は当社へお願いいたします。

当社ではデバイスドライバのインストールおよび附属ソフトウェアによる機器運用を理由とする損失、逸失利益等の請求につきましては、理由如何に関わらずいかなる責任も負いかねますので、予めご了承ください。

Ver.1.02

2019年11月12日

システムサコム工業株式会社

免責事項

- 1) 本インストールマニュアルの内容は予告なく変更することがあります。
- 2) 本インストールマニュアルの一部または全部を無断転載することは禁じられています。
- 3) 本インストールマニュアルの内容は万全を期して作成いたしました。万が一不審な事やお気づきの事がありましたら、システムサコム工業株式会社までご連絡下さい。
- 4) 当社では本製品の運用を理由とする損失、逸失利益等の請求につきましては、上記に関わらずいかなる責任も負いかねますので、予めご了承下さい。
- 5) 本製品は、人命に関わる設備や機器、高度な信頼性を必要とする設備や機器などへの組込や制御などへの使用は意図されておりません。これら設備や機器などに本装置を使用され人身事故、財産損害などが生じても、当社はいかなる責任も負いかねます。
- 6) 本製品およびソフトウェアが外国為替及び外国貿易管理法の規定により戦略物資（又は役務）に該当する場合には日本国外へ輸出する際に日本国政府の輸出許可が必要です。

Copyright © 2013, 2018 System Sacom Industry Corporaion. All rights reserved.
システムサコム工業株式会社の許可なく、本書の内容の複製、改変などを行うことはできません。

FTDIは英国Future Technology Devices International Limitedの登録商標です。
Microsoft, Windows, Windows XP, Windows Vista, Windows 7, Windows 8, Windows 8.1, Windows 10, Windows Update, Microsoft Update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。その他、記載されている会社名、製品名は、各社の商標および登録商標です。

目次

1. 概要	4
2. サンプルアプリケーション.....	5
2.1 プロジェクトの設定	6
3. ライブラリー関数リファレンス.....	7
3.1 sacomUsbIoOpen	7
3.2 sacomUsbIoClose	8
3.3 sacomUsbIoDirL	9
3.4 sacomUsbIoDirH	10
3.5 sacomUsbIoDir	11
3.6 sacomUsbIoWriteL.....	12
3.7 sacomUsbIoWriteH	13
3.8 sacomUsbIoWrite	14
3.9 sacomUsbIoReadL	15
3.10 sacomUsbIoReadH.....	16
3.11 sacomUsbIoRead	17
3.12 sacomUsbIoBit	18
3.13 sacomUsbIoReadback	19
3.14 sacomUsbIoUnitID	20
4. コマンドリファレンス	21
4.1 U(get Unit number)	22
4.2 D(get Direction)	23
4.3 I(Input data)	24
4.4 O(read Output data)	25
4.5 F(write Flash memory).....	26
4.6 E(set Echo mode).....	27
4.7 S(set Silent mode).....	28
4.8 T(get Title string)	29
4.9 V(get Version number).....	30
4.10 P(blink Power led).....	31
4.11 DL(set Direction Low byte)	32
4.12 DH(set Direction High byte)	33
4.13 OL(Output data Low byte)	34
4.14 OH(Output data High byte)	35
4.15 D(set Direction).....	36
4.16 O(Output data)	37
4.17 T(set Title string).....	38

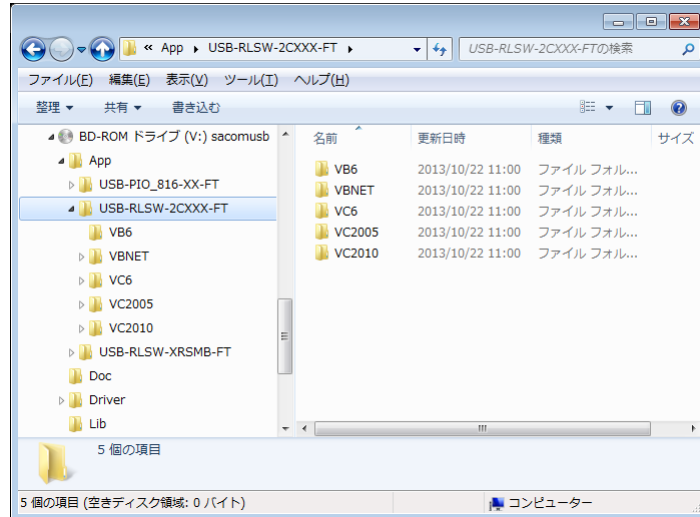
1. 概要

製品をご使用になる際には以下の2種類の操作方法があります。いずれの方法も事前にデバイスドライバーソフトウェアのインストールを行っておく必要があります。インストールの方法は別冊の「デバイスドライバーソフトウェア インストールマニュアル」をご参照ください。

- ・ Visual C++または Visual Basic でアプリケーションプログラムを作成する。
後述の 2. サンプルアプリケーションで説明するソースコード付サンプルアプリケーションを改造または新規にプロジェクトを作成して、アプリケーションプログラムを開発し、このアプリケーションを利用して操作します。
- ・ 仮想 COM ポートを利用し、ターミナルソフト上からコマンド操作する。
PC にインストールしたデバイスドライバーソフトウェアにより、仮想 COM ポートとして RS-232C 対応通信ソフト経由でデバイスにアクセスして操作します。

2. サンプルアプリケーション

製品附属の CD-ROM には製品を使用するためのサンプルアプリケーションソフトウェアがソースコードを含め収録されていますので、適切な場所にコピーしてお使いください。CD-ROM には製品シリーズごとにフォルダー分けされて収録されています。各シリーズ用とも同じフォルダー構成になっています。



VB6	Visual Basic 6 用
VBNET	Visual Basic.NET 用(Visual Basic 2003 以降)
VC6	Visual C++ 6 用
VC2005	Visual C++ 2005 用
VC2010	Visual C++ 2010 用

- Visual C++用のサンプルソフトと Visual Basic 用のサンプルソフトの動作は同一です。
- Visual C++のプロジェクトは MFC を前提に作成されているため、Express エディションでは開発することができません。実行ファイル自体は MFC スタティックライブラリーをリンクしているため、開発環境がインストールされていない場合でも実行が可能です。
- Visual C++ 2005, Visual C++ 2010 用は Visual C++ 6 用のプロジェクトを変換したもので、ソースファイルの内容は全て同一です。その他のバージョンの Visual C++でもプロジェクトファイルを変換することに使用できます。
- プロジェクトフォルダーの中には構築後の実行ファイルが入っており、そのまま実行することができます。
- 各サンプルソフトで設定できるユニット番号は 0 に固定されています。

詳しくは各プロジェクトのソースコードなどをご覧ください。

2.1 プロジェクトの設定

2.1.1 Visual C++ 6, Visual C++ 2005, Visual C++ 2010

- ・ 附属 CD-ROM もしくは弊社ウェブサイトからダウンロードしたファイル中の Lib フォルダより以下のファイルをプロジェクト中の任意のフォルダ(通常はソースファイルと同一フォルダで可)にコピーします。

node,

```
sacom_usb_uo_ft.lib      // プロジェクトに追加します。  
sacom_usb_uo_ft.h // ソースファイルからインクルードします。
```

- ・ 以下の DLL ファイルを実行ファイル(生成された.exe ファイル)と同じフォルダにコピーします。

```
sacom_usb_uo_ft.dll      // 実行時に必要です。
```

2.1.2 Visual Basic 6

- ・ 附属 CD-ROM もしくは弊社ウェブサイトからダウンロードしたファイル中の Lib フォルダより以下のファイルをプロジェクト中の任意のフォルダ(通常はソースファイルと同一フォルダで可)にコピーします。

```
sacom_usb_uo_ft.bas      ' プロジェクトに追加します。  
sacom_usb_uo_ft.dll      ' 実行時に必要です。
```

2.1.3 Visual Basic.NET

- ・ 附属 CD-ROM もしくは弊社ウェブサイトからダウンロードしたファイル中の Lib フォルダより以下のファイルをプロジェクト中の任意のフォルダ(通常は.vb はソースファイルと同一フォルダ、.dll は bin フォルダで可)にコピーします。

```
sacom_usb_uo_ft.vb      ' プロジェクトに追加します。  
sacom_usb_uo_ft.dll      ' 実行時に必要です。
```

3. ライブラリー関数リファレンス

3.1 sacomUsbIoOpen

C, C++宣言:

```
BOOL _stdcall sacomUsbIoOpen(DWORD dwId)
```

VB6 定義:

```
Declare Function sacomUsbIoOpen Lib "sacom_usb_io_ft" (ByVal id As Long) As Boolean
```

VB.NET 定義:

```
Declare Function sacomUsbIoOpen Lib "sacom_usb_io_ft.dll" (ByVal id As Integer) As Boolean
```

解説:

指定 ID に一致するユニット番号のデバイスをオープンします。ユニット番号とはデバイス本体上にある 2 個のロータリースイッチにより設定された 16 進数 2 桁の値です。1 台の PC に複数台のデバイスを接続した際の識別用として使用するため、重複しない値を設定する必要があります。設定できる値は 00h ~ FEh の 255 種類です。FFh はドライバー関数がデバイスにユニット番号を問い合わせるために使用するために、ユーザープログラムから使用することはできません。このデバイスに関する各種関数を使用する前に必ず呼び出す必要があります。

引数:

dwID または id

オープンするデバイスのユニット番号。指定可能な値は 16 進数の 00h から FEh です。

戻値:

TRUE / True

成功

FALSE / False

失敗

3.2 sacomUsbIoClose

C, C++宣言:

```
BOOL _stdcall sacomUsbIoClose(DWORD dwId)
```

VB6 定義:

```
Declare Function sacomUsbIoClose Lib "sacom_usb_io_ft" (ByVal id As Long) As Boolean
```

VB.NET 定義:

```
Declare Function sacomUsbIoClose Lib "sacom_usb_io_ft" (ByVal id As Integer) As Boolean
```

解説:

指定 ID に一致するユニット番号のデバイスをクローズします。

引数:

dwID または id

クローズするデバイスのユニット番号。指定可能な値は 16 進数の 00h から FEh です。また指定するユニット番号のデバイスはオープンされている必要があります。

戻値:

TRUE / True

成功

FALSE / False

失敗

3.3 sacomUsbIoDirL

C, C++宣言:

```
BOOL _stdcall sacomUsbIoDirL(DWORD dwId, DWORD dwDir)
```

VB6 定義:

```
Declare Function sacomUsbIoDirL  
    Lib "sacom_usb_io_ft" (ByVal id As Long, ByVal dir As Long) As Boolean
```

VB.NET 定義:

```
Declare Function sacomUsbIoDirL  
    Lib "sacom_usb_io_ft" (ByVal id As Integer, ByVal dir As Integer) As Boolean
```

解説:

16ビットのデジタル入出力のうち、下位8ビット(IO0~IO7)の入出力方向を指定します。

引数:

dwID または id

入出力方向を指定するデバイスのユニット番号。指定可能な値は16進数の00hからFEhです。また指定するユニット番号のデバイスはオープンされている必要があります。

dwDir または dir

入出力方向を指定するビットパターン。指定値のビット0がIO0、ビット7がIO7をあらわします。各ビットは1で出力設定、0で入力設定です。

戻値:

TRUE / True

成功

FALSE / False

失敗

3.4 sacomUsbIoDirH

C, C++宣言:

```
BOOL _stdcall sacomUsbIoDirH(DWORD dwId, DWORD dwDir)
```

VB6 定義:

```
Declare Function sacomUsbIoDirH  
    Lib "sacom_usb_io_ft" (ByVal id As Long, ByVal dir As Long) As Boolean
```

VB.NET 定義:

```
Declare Function sacomUsbIoDirH  
    Lib "sacom_usb_io_ft" (ByVal id As Integer, ByVal dir As Integer) As Boolean
```

解説:

16ビットのデジタル入出力のうち、上位8ビット(IO8~IO15)の入出力方向を指定します。

引数:

dwID または id

入出力方向を指定するデバイスのユニット番号。指定可能な値は16進数の00hからFEhです。また指定するユニット番号のデバイスはオープンされている必要があります。

dwDir または dir

入出力方向を指定するビットパターン。指定値のビット0がIO8、ビット7がIO15をあらわします。各ビットは1で出力設定、0で入力設定です。

戻値:

TRUE / True

成功

FALSE / False

失敗

3.5 sacomUsbIoDir

C, C++宣言:

```
BOOL _stdcall sacomUsbIoDir(DWORD dwId, DWORD dwDir)
```

VB6 定義:

```
Declare Function sacomUsbIoDir  
    Lib "sacom_usb_io_ft" (ByVal id As Long, ByVal dir As Long) As Boolean
```

VB.NET 定義:

```
Declare Function sacomUsbIoDir  
    Lib "sacom_usb_io_ft" (ByVal id As Integer, ByVal dir As Integer) As Boolean
```

解説:

16 ビットのデジタル入出力の全ビット(IO0~IO15)の入出力方向を指定します。

引数:

dwID または id

入出力方向を指定するデバイスのユニット番号。指定可能な値は 16 進数の 00h から FEh です。また指定するユニット番号のデバイスはオープンされている必要があります。

dwDir または dir

入出力方向を指定するビットパターン。指定値のビット 0 が IO0、ビット 15 が IO15 をあらわします。各ビットは 1 で出力設定、0 で入力設定です。

戻値:

TRUE / True

成功

FALSE / False

失敗

3.6 sacomUsbIoWriteL

C, C++宣言:

```
BOOL _stdcall sacomUsbIoWriteL(DWORD dwId, DWORD dwData)
```

VB6 定義:

```
Declare Function sacomUsbIoWriteL  
    Lib "sacom_usb_io_ft" (ByVal id As Long, ByVal data As Long) As Boolean
```

VB.NET 定義:

```
Declare Function sacomUsbIoWriteL  
    Lib "sacom_usb_io_ft" (ByVal id As Integer, ByVal data As Integer) As Boolean
```

解説:

16 ビットのデジタル入出力のうち、下位 8 ビット(IO0~IO7)にデータを出力します。入出力方向で出力設定されたビットにのみ、指定値を出力します。

引数:

dwID または id

データ出力を行うデバイスのユニット番号。指定可能な値は 16 進数の 00h から FEh です。また指定するユニット番号のデバイスはオープンされている必要があります。

dwData または data

出力データをあらわすビットパターン。指定値のビット 0 が IO0、ビット 7 が IO7 をあらわします。各ビットは 1 で H レベル、0 で L レベルをそれぞれ出力します。

戻値:

TRUE / True

成功

FALSE / False

失敗

3.7 sacomUsbIoWriteH

C, C++宣言:

```
BOOL _stdcall sacomUsbIoWriteH(DWORD dwId, DWORD dwData)
```

VB6 定義:

```
Declare Function sacomUsbIoWriteH  
    Lib "sacom_usb_io_ft" (ByVal id As Long, ByVal data As Long) As Boolean
```

VB.NET 定義:

```
Declare Function sacomUsbIoWriteH  
    Lib "sacom_usb_io_ft" (ByVal id As Integer, ByVal data As Integer) As Boolean
```

解説:

16ビットのデジタル入出力のうち、上位8ビット(IO8~IO15)にデータを出力します。入出力方向で出力設定されたビットにのみ、指定値を出力します。

引数:

dwID または id

データ出力を行うデバイスのユニット番号。指定可能な値は16進数の00hからFEhです。また指定するユニット番号のデバイスはオープンされている必要があります。

dwData または data

出力データをあらわすビットパターン。指定値のビット0がIO8、ビット7がIO15をあらわします。各ビットは1でHレベル、0でLレベルをそれぞれ出力します。

戻値:

TRUE / True

成功

FALSE / False

失敗

3.8 sacomUsbIoWrite

C, C++宣言:

```
BOOL _stdcall sacomUsbIoWrite(DWORD dwId, DWORD dwData)
```

VB6 定義:

```
Declare Function sacomUsbIoWrite  
    Lib "sacom_usb_io_ft" (ByVal id As Long, ByVal data As Long) As Boolean
```

VB.NET 定義:

```
Declare Function sacomUsbIoWrite  
    Lib "sacom_usb_io_ft" (ByVal id As Integer, ByVal data As Integer) As Boolean
```

解説:

16 ビットのデジタル入出力の全ビット (IO0~IO15)にデータを出力します。入出力方向で出力設定されたビットにのみ、指定値を出力します。

引数:

dwID または id

データ出力を行うデバイスのユニット番号。指定可能な値は 16 進数の 00h から FEh です。また指定するユニット番号のデバイスはオープンされている必要があります。

dwData または data

出力データをあらわすビットパターン。指定値のビット 0 が IO0、ビット 15 が IO15 をあらわします。各ビットは 1 で H レベル、0 で L レベルをそれぞれ出力します。

戻値:

TRUE / True

成功

FALSE / False

失敗

3.9 sacomUsbIoReadL

C, C++宣言:

```
BOOL _stdcall sacomUsbIoReadL(DWORD dwId, LPDWORD lpdwData)
```

VB6 定義:

```
Declare Function sacomUsbIoReadL  
    Lib "sacom_usb_io_ft" (ByVal id As Long, ByRef data As Long) As Boolean
```

VB.NET 定義:

```
Declare Function sacomUsbIoReadL  
    Lib "sacom_usb_io_ft" (ByVal id As Integer, ByRef data As Integer) As Boolean
```

解説:

16 ビットのデジタル入出力のうち、下位 8 ビット(I00~I07)からデータを入力します。入出力方向で入力設定されたビットからのみ入力し、出力設定されたビットは常に 0 が返ります。

引数:

dwID または id

データ入力を行うデバイスのユニット番号。指定可能な値は 16 進数の 00h から FEh です。また指定するユニット番号のデバイスはオープンされている必要があります。

lpdwData または data

入力データをあらわすビットパターンを格納します。入力値のビット 0 が I00、ビット 7 が I07 をあらわします。各ビットは 1 で H レベル、0 で L レベルをそれぞれあらわします。

戻値:

TRUE / True

成功

FALSE / False

失敗

3.10 sacomUsbIoReadH

C, C++宣言:

```
BOOL _stdcall sacomUsbIoReadH(DWORD dwId, LPDWORD lpdwData)
```

VB6 定義:

```
Declare Function sacomUsbIoReadH
```

```
    Lib "sacom_usb_io_ft" (ByVal id As Long, ByVal data As Long) As Boolean
```

VB.NET 定義:

```
Declare Function sacomUsbIoReadH
```

```
    Lib "sacom_usb_io_ft" (ByVal id As Integer, ByVal data As Integer) As Boolean
```

解説:

16ビットのデジタル入出力のうち、上位8ビット(IO8~IO15)からデータを入力します。入出力方向で入力設定されたビットからのみ入力し、出力設定されたビットは常に0が返ります。

引数:

dwID または id

データ入力を行うデバイスのユニット番号。指定可能な値は16進数の00hからFEhです。また指定するユニット番号のデバイスはオープンされている必要があります。

lpdwData または data

入力データをあらわすビットパターンを格納します。入力値のビット0がIO8、ビット7がIO15をあらわします。各ビットは1でHレベル、0でLレベルをそれぞれあらわします。

戻値:

TRUE / True

成功

FALSE / False

失敗

3.11 sacomUsbIoRead

C, C++宣言:

```
BOOL _stdcall sacomUsbIoRead(DWORD dwId, LPDWORD lpdwData)
```

VB6 定義:

```
Declare Function sacomUsbIoRead  
    Lib "sacom_usb_io_ft" (ByVal id As Long, ByRef data As Long) As Boolean
```

VB.NET 定義:

```
Declare Function sacomUsbIoRead  
    Lib "sacom_usb_io_ft" (ByVal id As Integer, ByRef data As Integer) As Boolean
```

解説:

16ビットのデジタル入出力の全ビット(I00~I015)からデータを入力します。入出力方向で入力設定されたビットからのみ入力し、出力設定されたビットは常に0が返ります。

引数:

dwID または id

データ入力を行うデバイスのユニット番号。指定可能な値は16進数の00hからFEhです。また指定するユニット番号のデバイスはオープンされている必要があります。

lpdwData または data

入力データをあらわすビットパターンを格納します。入力値のビット0がI00、ビット15がI015をあらわします。各ビットは1でHレベル、0でLレベルをそれぞれあらわします。

戻値:

TRUE / True

成功

FALSE / False

失敗

3.12 sacomUsbIoBit

C, C++宣言:

```
BOOL _stdcall sacomUsbIoBit(DWORD dwId, DWORD dwBit, BOOL bData)
```

VB6 定義:

```
Declare Function sacomUsbIoBit Lib "sacom_usb_io_ft"  
    (ByVal id As Long, ByVal bit As Long, ByVal data As Boolean) As Boolean
```

VB.NET 定義:

```
Declare Function sacomUsbIoBit Lib "sacom_usb_io_ft"  
    (ByVal id As Integer, ByVal bit As Integer, ByVal data As Boolean) As Boolean
```

解説:

デジタル入出力の特定のビットにデータを出力します。入出力方向で出力設定されたビットにのみ、指定値を出力します。

引数:

dwID または id

ビットデータ出力を行うデバイスのユニット番号。指定可能な値は 16 進数の 00h から FEh です。また指定するユニット番号のデバイスはオープンされている必要があります。

dwBit または bit

出力するビット位置。デジタル入出力のビットに対応し、0 が IO0、15 が IO15 をあらわします。

bData または data

出力するビットデータ。TRUE または True で H レベルまたはリレーを ON、FALSE または False で L レベルまたはリレーを OFF に設定します。

戻値:

TRUE / True

成功

FALSE / False

失敗

3.13 sacomUsbIoReadback

C, C++宣言:

```
BOOL _stdcall sacomUsbIoReadback(DWORD dwId, LPDWORD lpdwData)
```

VB6 定義:

```
Declare Function sacomUsbIoReadback  
    Lib "sacom_usb_io_ft" (ByVal id As Long, ByRef data As Long) As Boolean
```

VB.NET 定義:

```
Declare Function sacomUsbIoReadback  
    Lib "sacom_usb_io_ft" (ByVal id As Integer, ByRef data As Integer) As Boolean
```

解説:

16ビットのデジタル入出力の全ビット(I00~I015)へ出力中のデータの現在の状態を取得します。入出力方向で出力設定されたビットからのみ取得し、入力設定されたビットは常に0が返ります。

引数:

dwID または id

データ取得を行うデバイスのユニット番号。指定可能な値は16進数の00hからFEhです。また指定するユニット番号のデバイスはオープンされている必要があります。

lpdwData または data

取得データをあらわすビットパターンを格納します。取得値のビット0がI00、ビット15がI015をあらわします。各ビットは1でHレベル、0でLレベルをそれぞれあらわします。

戻値:

TRUE / True

成功

FALSE / False

失敗

3.14 sacomUsbIoUnitID

C, C++宣言:

```
BOOL _stdcall sacomUsbIoUnitID(DWORD dwId, LPDWORD lpdwData)
```

VB6 定義:

```
Declare Function sacomUsbIoUnitID  
    Lib "sacom_usb_io_ft" (ByVal id As Long, ByRef data As Long) As Boolean
```

VB.NET 定義:

```
Declare Function sacomUsbIoUnitID  
    Lib "sacom_usb_io_ft" (ByVal id As Integer, ByRef data As Integer) As Boolean
```

解説:

接続されたデバイスのユニット番号を取得します。

引数:

dwID または id

ユニット番号取得を行うデバイスのユニット番号。指定可能な値は 16 進数の 00h から FEh です。
また指定するユニット番号のデバイスはオープンされている必要があります。

lpdwData または data

取得したユニット番号を格納します。dwID または id で指定したユニット番号と同じ内容が格納されます。

戻値:

TRUE / True

成功

FALSE / False

失敗

4. コマンドリファレンス

概要:

デバイスは接続された PC にインストールされているデバイスドライバーソフトウェアにより仮想 COM として Windows アプリケーションからシリアルポートとして認識されます。次項の通信仕様に従いポートをオープンすることにより、シリアルターミナルソフトなどからコマンドを使用してデバイスを操作することができます。操作は ASCII 文字列からなるコマンドを PC からデバイスに送信し、デバイスから送信されるレスポンスを PC で受信することで行われます。コマンドの終端として後述のデリミター文字を送信し、レスポンスの終端は送信したコマンドのデリミター文字がそのまま返送されます。

通信仕様:

通信方式	仮想 COM による調歩式同期(非同期)
通信速度	115200bps
データ長	8 ビット
パリティ	なし
ストップビット	1 ビット
フロー制御	なし

基本書式:

コマンドは以下の書式で構成されています。

1. 16 進数 2 桁からなるユニット番号文字列。
2. アルファベット 1 あるいは 2 文字からなるコマンド文字列。
3. コマンドに対応する 0~63 文字からなる指定文字列。
4. '/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), '|'(7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれか 1 文字のデリミター文字。

レスポンスは以下の書式で構成されています。

1. コマンドに対応する 0~63 文字からなる応答文字列。
2. '/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), '|'(7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれか 1 文字のデリミター文字。

4.1 U(get Unit number)

動作機種:

全機種で動作します。

解説:

現在接続されているデバイスのユニット番号(デバイス本体のロータリースイッチに設定)を16進数2桁で取得します。

書式:

0	1	2	3
ID		CMD	DE

ID 2バイト 16進数2桁のユニット番号の文字列。通常は全ユニット共通番号'FF'(46h, 46h)を指定します。

CMD 1バイト コマンド文字'U'(55h)または'u'(75h)。

DE 1バイト デリミター文字。使用できる文字は'/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), '|'(7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれかです。

応答:

0	1	2
ID		DE

ID 2バイト 16進数2桁のユニット番号のASCII文字列。デバイス本体のロータリースイッチに設定された内容が返ります。

DE 1バイト デリミター文字。コマンドで使用したデリミター文字と同じデータが返ります。'/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), '|'(7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれかです。

例:

FFU 接続されているID不明のデバイスに対してユニット番号取得を要求します。

12 接続されているデバイスのユニット番号は12hです。

4.2 D(get Direction)

動作機種:

USB-PIO 8/16-BX-FT, USB-PIO 8/16-PR-FT, USB-PIO 8/16-BXLV-FT, USB-PIO 8/16-PRLV-FT

解説:

デバイスに設定されている IO 方向を 16 進数 4 桁で取得します。ビット 0 がデジタル入出力 0(IO0)、ビット 15 がデジタル入出力 15(IO15)です。各ビットは 0 が入力、1 が出力をあらわしています。

書式:

0	1	2	3
ID	CMD	DE	

ID 2 バイト 16 進数 2 桁のユニット番号の ASCII 文字列。

CMD 1 バイト コマンド文字'D'(44h)または'd'(64h)。

DE 1 バイト デリミター文字。使用できる文字は'/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), '|'(7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれかです。

応答:

0	1	2	3	4
DIR				DE

DIR 4 バイト 16 進数 4 桁の IO 方向のビットパターンをあらわす ASCII 文字列。各ビットは 0 が入力、1 が出力をあらわしています。

DE 1 バイト デリミター文字。コマンドで使用したデリミター文字と同じデータが返ります。'/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), '|'(7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれかです。

例:

12D ユニット番号 12h のデバイスに対して IO 方向取得を要求します。

FF00 このデバイスに設定されている IO 方向は FF00h です。ビット 15~8 が出力、ビット 7~0 が入力に設定されています。

4.3 I(Input data)

動作機種:

USB-PIO 8/16-BX-FT, USB-PIO 8/16-PT-FT, USB-PIO 8/16-BXLV-FT, USB-PIO 8/16-PRLV-FT

解説:

デバイスのデジタル入出力端子からデータを入力し 16 進数 4 桁で取得します。ビット 0 がデジタル入出力 0(IO0)、ビット 15 がデジタル入出力 15(IO15)です。各ビットは 0 が L、1 が H をあらわしています。IO 方向が出力に設定されているビットは 0(L)固定となります。

書式:

0	1	2	3
ID		CMD	DE

ID 2 バイト 16 進数 2 桁のユニット番号の ASCII 文字列。
CMD 1 バイト コマンド文字'I'(49h)または'i'(69h)。
DE 1 バイト デリミター文字。使用できる文字は '/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), '|'(7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれかです。

応答:

0	1	2	3	4
DATA				DE

DATA 4 バイト 16 進数 4 桁の入力値をあらわす ASCII 文字列。各ビットは 0 が L、1 が H をあらわしています。
DE 1 バイト デリミター文字。コマンドで使用したデリミター文字と同じデータが返ります。 '/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), '|'(7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれかです。

例:

12I ユニット番号 12h のデバイスに対してデジタル入出力端子からの入力を要求します。
00AA 入力データは 00AAh です。

4.4 O(read Output data)

動作機種:

全機種で動作します。

解説:

現在デバイスのデジタル入出力端子から出力されている出力データの状態を16進数4桁でリードバックし取得します。ビット0がデジタル入出力0(IO0)、ビット15がデジタル入出力15(IO15)です。各ビットは0がL、1がHをあらわしています。IO方向が入力に設定されているビットは0(L)固定となります。

書式:

0	1	2	3
ID	CMD	DE	

ID 2バイト 16進数2桁のユニット番号のASCII文字列。

CMD 1バイト コマンド文字'O'(4Fh)または'o'(6Fh)。

DE 1バイト デリミター文字。使用できる文字は'/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), '|'(7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれかです。

応答:

0	1	2	3	4
DATA				DE

DATA 4バイト 16進数4桁のリードバックされた出力値をあらわすASCII文字列。各ビットは0がL、1がHをあらわしています。

DE 1バイト デリミター文字。コマンドで使用したデリミター文字と同じデータが返ります。'/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), '|'(7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれかです。

例:

120 ユニット番号 12h のデバイスに対してデジタル入出力端子の出力値のリードバックを要求します。

5500 現在出力中のデータは5500hです。

4.5 F(write Flash memory)

動作機種:

全機種で動作します。

解説:

現在接続されているデバイス内に保存可能な情報をフラッシュメモリに書込みます。保存される内容は IO 方向設定です。

書式:

0	1	2	3
ID		CMD	DE

ID 2 バイト 16 進数 2 桁のユニット番号の文字列。

CMD 1 バイト コマンド文字'F'(46h)または'f'(66h)。

DE 1 バイト デリミター文字。使用できる文字は'/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), '|'(7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれかです。

応答:

0
DE

DE 1 バイト デリミター文字。コマンドで使用したデリミター文字と同じデータが返ります。'/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), '|'(7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれかです。

例:

12F ユニット番号 12h のデバイスに対してフラッシュメモリの書込みを要求します。

なし デリミター文字が返るだけで、情報はありません。

4.6 E(set Echo mode)

動作機種:

全機種で動作します。

解説:

コマンドのエコーバックを有効に設定します。エコーバックを有効にすると、デバイスを接続しているPCからコマンドデータを送信する際に、1バイトずつエコーバックとしてデバイスから同じデータが返ってきます。デバイスは電源投入時にはエコーバックは無効に設定されています。

書式:

0	1	2	3
ID		CMD	DE

ID 2バイト 16進数2桁のユニット番号の文字列。
CMD 1バイト コマンド文字'E'(45h)または'e'(65h)。
DE 1バイト デリミター文字。使用できる文字は'/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), '|'(7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれかです。

応答:

0
DE

DE 1バイト デリミター文字。コマンドで使用したデリミター文字と同じデータが返ります。
'/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), '|'(7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれかです。

例:

12E ユニット番号 12h のデバイスに対してエコーバック有効を要求します。
なし デリミター文字が返るだけで、情報はありません。

4.7 S(set Silent mode)

動作機種:

全機種で動作します。

解説:

コマンドのエコーバックを無効に設定します。エコーバックを無効にすると、デバイスを接続しているPCからコマンドデータを送信する際に、応答が返るまでの間、エコーバックが返らなくなります。デバイスは電源投入時にはエコーバックは無効に設定されています。

書式:

0	1	2	3
ID		CMD	DE

ID 2 バイト 16 進数 2 桁のユニット番号の文字列。
CMD 1 バイト コマンド文字'S'(53h)または's'(73h)。
DE 1 バイト デリミター文字。使用できる文字は'/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), '|'(7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれかです。

応答:

0
DE

DE 1 バイト デリミター文字。コマンドで使用したデリミター文字と同じデータが返ります。
'/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), '|'(7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれかです。

例:

12S ユニット番号 12h のデバイスに対してエコーバック無効を要求します。
なし デリミター文字が返るだけで、情報はありません。

4.8 T(get Title string)

動作機種:

全機種で動作します。

解説:

デバイス内フラッシュメモリに書き込まれたユニットのタイトル文字列を取得します。製品出荷時にはタイトルが書き込まれていないため、空の文字列が返ります。タイトル文字列はデリミター文字として有効な '/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), '|'(7Ch), 改行(0Dh), ラインフィード(0Ah)を除く ASCII 文字で構成され、最大 63 文字を格納することが可能です。

書式:

0	1	2	3
ID		CMD	DE

ID 2 バイト 16 進数 2 桁のユニット番号の文字列。

CMD 1 バイト コマンド文字 'T'(54h)または 't'(74h)。

DE 1 バイト デリミター文字。使用できる文字は '/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), '|'(7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれかです。

応答:

0	n-1	n
TI		DE

TI 可変長 0~63 バイトの長さを持つ可変長の文字列です。

DE 1 バイト デリミター文字。コマンドで使用したデリミター文字と同じデータが返ります。 '/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), '|'(7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれかです。

例:

12T ユニット番号 12h のデバイスに対してタイトル文字列取得を要求します。

I/O unit #12 デバイス内フラッシュメモリに記録されたタイトル文字列です。

4.9 V(get Version number)

動作機種:

全機種で動作します。

解説:

デバイス内 ROM に書き込まれたファームウェアのバージョン情報文字列を取得します。機種名、バージョン番号、ファームウェアをビルドした日時などが記録されています。

書式:

0	1	2	3
ID	CMD	DE	

ID 2バイト 16進数2桁のユニット番号の文字列。

CMD 1バイト コマンド文字'V'(56h)または'v'(76h)。

DE 1バイト デリミター文字。使用できる文字は'/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), '|'(7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれかです。

応答:

0	n-1	n
TI	DE	

TI 可変長 可変長の文字列です。データ長に制限はありません。

DE 1バイト デリミター文字。コマンドで使用したデリミター文字と同じデータが返ります。'/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), '|'(7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれかです。

例:

12V

ユニット番号 12h のデバイスに対してバージョン情報文字列取得を要求します。

USB-PIO 8/16-BX-FT 2.0.0

2013-09-06 17:31:14

デバイス内 ROM に記録されたバージョン情報文字列です。

4.10 P(blink Power led)

動作機種:

USB-RLSW-2CXX-FT シリーズを除く全機種で動作します。

解説:

デバイス本体上部の POWER LED を約 1 秒間点滅させます。複数台のユニットを接続し運用する場合、目視で特定のユニットを探す際に使用します。

書式:

0	1	2	3
ID		CMD	DE

ID 2 バイト 16 進数 2 桁のユニット番号の文字列。

CMD 1 バイト コマンド文字'P'(50h)または'p'(70h)。

DE 1 バイト デリミター文字。使用できる文字は'/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), '|'(7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれかです。

応答:

0
DE

DE 1 バイト デリミター文字。コマンドで使用したデリミター文字と同じデータが返ります。'/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), '|'(7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれかです。

例:

12P ユニット番号 12h のデバイスに対して POWER LED 点滅を要求します。

なし デリミター文字が返るだけで、情報はありません。

4.11 DL(set Direction Low byte)

動作機種:

USB-PIO 8/16-BX-FT, USB-PIO 8/16-PR-FT, USB-PIO 8/16-BXLV-FT, USB-PIO 8/16-PRLV-FT

解説:

デバイスのデジタル入出力端子のうち、下位 8 ビットの IO 方向を 16 進数 2 桁で設定します。ビット 0 がデジタル入出力 0 (IO0)、ビット 7 がデジタル入出力 7 (IO7) です。各ビットは 0 が入力、1 が出力をあらわしています。

書式:

0	1	2	3	4	5	6
ID		CMD		DIR		DE

- ID 2 バイト 16 進数 2 桁のユニット番号の ASCII 文字列。
- CMD 2 バイト コマンド文字 "DL"(44h, 4Ch) または "dl"(64h, 6Ch)。
- DIR 2 バイト 16 進数 2 桁の IO 方向のビットパターンをあらわす ASCII 文字列。各ビットは 0 が入力、1 が出力をあらわしています。
- DE 1 バイト デリミター文字。使用できる文字は '/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), '|'(7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれかです。

応答:

0
DE

- DE 1 バイト デリミター文字。コマンドで使用したデリミター文字と同じデータが返ります。 '/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), '|'(7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれかです。

例:

- 12DLAA ユニット番号 12h のデバイスに対して下位 8 ビットの IO 方向を設定します。ビット 7, 5, 3, 1 を出力、ビット 6, 4, 2, 0 を入力に設定します。
- なし デリミター文字が返るだけで、情報はありません。

4.12 DH(set Direction High byte)

動作機種:

USB-PIO 8/16-BX-FT, USB-PIO 8/16-PR-FT, USB-PIO 8/16-BXLV-FT, USB-PIO 8/16-PRLV-FT

解説:

デバイスのデジタル入出力端子のうち、上位 8 ビットの IO 方向を 16 進数 2 桁で設定します。ビット 0 がデジタル入出力 8(IO8)、ビット 7 がデジタル入出力 15(IO15)です。各ビットは 0 が入力、1 が出力をあらわしています。

書式:

0	1	2	3	4	5	6
ID		CMD		DIR		DE

ID	2 バイト	16 進数 2 桁のユニット番号の ASCII 文字列。
CMD	2 バイト	コマンド文字"DH"(44h, 48h)または"dh"(64h, 68h)。
DIR	2 バイト	16 進数 2 桁の IO 方向のビットパターンをあらわす ASCII 文字列。 各ビットは 0 が入力、1 が出力をあらわしています。
DE	1 バイト	デリミター文字。使用できる文字は'/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), ' '(7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれかです。

応答:

0
DE

DE	1 バイト	デリミター文字。コマンドで使用したデリミター文字と同じデータが返ります。 '/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), ' '(7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれかです。
----	-------	---

例:

12DH55	ユニット番号 12h のデバイスに対して上位 8 ビットの IO 方向を設定します。 ビット 15, 13, 11, 9 を入力、ビット 14, 12, 10, 8 を出力に設定します。
なし	デリミター文字が返るだけで、情報はありません。

4.13 OL(Output data Low byte)

動作機種:

全機種で動作します。

解説:

デバイスのデジタル入出力端子のうち、下位 8 ビットに対してデータを出力します。ビット 0 がデジタル入出力 0 (IO0)、ビット 7 がデジタル入出力 7 (IO7) です。各ビットは 0 が L 状態、1 が H 状態をあらわしています。

書式:

0	1	2	3	4	5	6
ID		CMD		DATA		DE

ID	2 バイト	16 進数 2 桁のユニット番号の ASCII 文字列。
CMD	2 バイト	コマンド文字"OL"(4Fh, 4Ch)または"ol"(6Fh, 6Ch)。
DATA	2 バイト	16 進数 2 桁の出力値のビットパターンをあらわす ASCII 文字列。各ビットは 0 が L、1 が H をあらわしています。
DE	1 バイト	デリミター文字。使用できる文字は '/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), ' '(7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれかです。

応答:

0
DE

DE	1 バイト	デリミター文字。コマンドで使用したデリミター文字と同じデータが返ります。 '/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), ' '(7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれかです。
----	-------	---

例:

120LAA	ユニット番号 12h のデバイスに対してビット 7, 5, 3, 1 に H、ビット 6, 4, 2, 0 に L を出力します。
なし	デリミター文字が返るだけで、情報はありません。

4.14 OH(Output data High byte)

動作機種:

全機種で動作します。

解説:

デバイスのデジタル入出力端子のうち、上位 8 ビットに対してデータを出力します。ビット 0 がデジタル入出力 8(IO8)、ビット 7 がデジタル入出力 15(IO15)です。各ビットは 0 が L 状態、1 が H 状態をあらわしています。

書式:

0	1	2	3	4	5	6
ID		CMD		DATA		DE

ID	2 バイト	16 進数 2 桁のユニット番号の ASCII 文字列。
CMD	2 バイト	コマンド文字"OH"(4Fh, 48h)または"oh"(6Fh, 68h)。
DATA	2 バイト	16 進数 2 桁の出力値のビットパターンをあらわす ASCII 文字列。各ビットは 0 が L、1 が H をあらわしています。
DE	1 バイト	デリミター文字。使用できる文字は '/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), ' '(7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれかです。

応答:

0
DE

DE	1 バイト	デリミター文字。コマンドで使用したデリミター文字と同じデータが返ります。 '/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), ' '(7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれかです。
----	-------	---

例:

120H55	ユニット番号 12h のデバイスに対してビット 15, 13, 11, 9 に L、ビット 14, 12, 10, 8 に H を出力します。
なし	デリミター文字が返るだけで、情報はありません。

4.15 D(set Direction)

動作機種:

USB-PIO 8/16-BX-FT, USB-PIO 8/16-PR-FT, USB-PIO 8/16-BXLV-FT, USB-PIO 8/16-PRLV-FT

解説:

デバイスのデジタル入出力端子の IO 方向を 16 進数 4 桁で設定します。ビット 0 がデジタル入出力 0(IO0)、ビット 15 がデジタル入出力 15(IO15)です。各ビットは 0 が入力、1 が出力をあらわしています。

書式:

0	1	2	3	4	5	6	7
ID	CMD	DIR				DE	

- ID 2 バイト 16 進数 2 桁のユニット番号の ASCII 文字列。
- CMD 1 バイト コマンド文字'D'(44h)または'd'(64h)。
- DIR 4 バイト 16 進数 4 桁の IO 方向のビットパターンをあらわす ASCII 文字列。各ビットは 0 が入力、1 が出力をあらわしています。
- DE 1 バイト デリミター文字。使用できる文字は'/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), '|'(7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれかです。

応答:

0
DE

- DE 1 バイト デリミター文字。コマンドで使用したデリミター文字と同じデータが返ります。'/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), '|'(7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれかです。

例:

- 12DAA55 ユニット番号 12h のデバイスに対して IO 方向を設定します。ビット 14, 12, 10, 8, 7, 5, 3, 1 を入力、ビット 15, 13, 11, 9, 6, 4, 2, 0 を出力に設定します。
- なし デリミター文字が返るだけで、情報はありません。

4.16 O(Output data)

動作機種:

全機種で動作します。

解説:

デバイスのデジタル入出力端子に対してデータを出力します。ビット 0 がデジタル入出力 0(IO0)、ビット 15 がデジタル入出力 15(IO15)です。各ビットは 0 が L 状態、1 が H 状態をあらわしています。

書式:

0	1	2	3	4	5	6	7
ID	CMD	DATA				DE	

ID	2 バイト	16 進数 2 桁のユニット番号の ASCII 文字列。
CMD	2 バイト	コマンド文字"OH"(4Fh, 48h)または"oh"(6Fh, 68h)。
DATA	4 バイト	16 進数 4 桁の出力値のビットパターンをあらわす ASCII 文字列。各ビットは 0 が L、1 が H をあらわしています。
DE	1 バイト	デリミター文字。使用できる文字は '/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), ' '(7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれかです。

応答:

0
DE

DE	1 バイト	デリミター文字。コマンドで使用したデリミター文字と同じデータが返ります。 '/'(2Fh), '%'(25h), '\$'(24h), ':'(3Ah), ' '(7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれかです。
----	-------	---

例:

1201234	ユニット番号 12h のデバイスに対してビット 12, 9, 5, 4, 2 に H、それ以外のビットに L を出力します。
なし	デリミター文字が返るだけで、情報はありません。

4.17 T(set Title string)

動作機種:

全機種で動作します。

解説:

デバイス内フラッシュメモリにタイトル文字列を書き込みます。製品出荷時にはタイトルが書き込まれておらず、空の文字列が設定されています。タイトル文字列はデリミター文字として有効な '/'(2Fh), '%' (25h), '\$'(24h), ':'(3Ah), '|' (7Ch), 改行(0Dh), ラインフィード(0Ah)を除く ASCII 文字で構成され、最大 63 文字を格納することが可能です。

書式:

0	1	2	3	n-1	n
ID	CMD	TI			DE

ID	2 バイト	16 進数 2 桁のユニット番号の文字列。
CMD	1 バイト	コマンド文字'T'(54h)または't'(74h)。
TI	可変長	0~63 バイトの長さを持つ可変長の文字列です。
DE	1 バイト	デリミター文字。使用できる文字は '/'(2Fh), '%' (25h), '\$'(24h), ':'(3Ah), ' ' (7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれかです。

応答:

0
DE

DE	1 バイト	デリミター文字。コマンドで使用したデリミター文字と同じデータが返ります。 '/'(2Fh), '%' (25h), '\$'(24h), ':'(3Ah), ' ' (7Ch), 改行(0Dh), ラインフィード(0Ah)のいずれかです。
----	-------	---

例:

12TI/O unit #12 ユニット番号 12h のデバイスに対してタイトル文字列書き込みを要求します。
なし デリミター文字が返るだけで、情報はありません。

改版履歴

2019年11月12日	1.02	住所変更
2018年06月07日	1.01	記述の誤り、レイアウトの乱れを訂正
2016年12月19日	1.00	初版

システムサコム工業株式会社 <http://www.sacom.co.jp/>

〒130-0021 東京都墨田区緑 1-22-5 州ビル 4F

TEL:03-6659-9261 FAX:03-6659-9264 E-Mail:info@sacom.co.jp