

外部 PC 制御機能付 USB 挿抜試験装置

DTC2U3

取扱説明書

Rev.2.04

目次

1	概要	4
1.1	特徴	4
1.2	仕様	4
1.3	セット内容	5
1.4	外観	6
2	操作方法	9
2.1	手動 USB 挿抜モード操作方法	9
2.2	自動 USB 挿抜モード操作方法	10
2.3	手動 USB エラー誘発モード操作方法	11
2.4	自動 USB エラー誘発モード操作方法	12
2.5	外部 PC 制御時の操作方法	13
3	Windows GUI 制御アプリケーション “DTC2U3App.exe” 説明	16
3.1	説明	16
3.2	使用方法	16
3.3	起動画面	16
4	Windows コマンドライン制御アプリケーション “MLEVGEN.exe” 説明	17
4.1	説明	17
4.2	使用方法	17
4.3	用例 1 ～プッシュボタン押下をエミュレーション～	18
4.4	用例 2 ～外部イベント#0を送信～	18
4.5	用例 3 ～制御対象とする DTC2U3 を選択して、外部イベント#0を送信～	18
5	各種エミュレーション動作設定	19
5.1	USB ケーブル接続エミュレーション動作設定	19
5.2	USB ケーブル切断エミュレーション動作設定	19
5.3	USB エラー誘発設定	19
6	処理スクリプトによる各種カスタマイズ	20
6.1	処理スクリプト概要	20
7	処理スクリプト書き込みツール “FWSWriter.exe” 使用方法	21
8	処理スクリプト基本書式	22
8.1	命令書式	22
8.2	ラベル書式	22
8.3	コメント書式	22
8.4	引数書式	23
9	制御レジスタ説明	24
9.1	制御レジスタ一覧	24
9.2	USB 制御レジスタ USBCTL	25
9.3	LED 制御レジスタ LEDCTL	26
9.4	ステータスレジスタ STATUS	27
9.5	イベントレジスタ EVENT	28
9.6	タイマーレジスタ #0 TIM0	29
9.7	タイマーレジスタ #1/#2/#3 TIM1/TIM2/TIM3	29
9.8	汎用レジスタ A/B/C/D/E/F/G/H	30
10	処理スクリプト命令説明	31
10.1	命令一覧	31
10.2	SET	32
10.3	MOVE	33
10.4	WAIT	34
10.5	NOP	35
10.6	JUMP	36
10.7	CALL	37
10.8	RETURN	38

10.9	END	39
10.10	ADD	40
10.11	SUB	40
10.12	AND	41
10.13	OR	41
10.14	XOR	41
10.15	NOT	42
10.16	CMP	43
10.17	JUMPEQ	44
10.18	JUMPNE	44
10.19	JUMPGT	45
10.20	JUMPGE	45
10.21	JUMPLT	46
10.22	JUMPLE	46
10.23	CHKBIT	47
10.24	JUMPNZ	48
10.25	JUMPZ	48
11	制御アプリケーション作成用ライブラリ MLGPLib 説明	49
11.1	ファイル構成	49
11.2	API 一覧	49
11.3	各 API 説明	50
11.4	MLGPLib ヘッダファイル	56
11.5	MLGPLib アプリケーションサンプル	57
12	安全上の注意	58
13	その他事項	59
14	改版履歴	59

1 概要

本外部 PC 制御機能付 USB 挿抜試験装置 (DTC2U3)は、USB2.0/USB3.0 のホスト-デバイス間に設置し、USB ケーブルの接続、切断の電氣的なエミュレーション、ならびに CRC エラーの誘発を行う装置です。USB ケーブル挿抜時、または CRC エラー発生時の不具合摘出、再現を可能としますので、USB 機器の開発や生産工程の効率化にご利用いただけます。

1.1 特徴

- USB ケーブルの電氣的な接続・切断エミュレーション、または CRC エラーの誘発を行います。
- 各種 USB 処理(接続・切断、エラー誘発)を自動実行する各種動作モードを搭載しています。
- 各種 USB 処理(接続・切断、エラー誘発)は、本体搭載のプッシュボタン、または USB ケーブルで接続された PC から制御可能です。
- 各種 USB 処理(接続・切断、エラー誘発)の設定や、自動処理手順はスクリプトを作成いただくことでカスタマイズできます。

1.2 仕様

ハードウェア仕様

対応 USB 制御機能は下記の通りです。

USB3.0 信号の接続/切断制御機能

- VBUS の接続/切断
- USB2.0 D+/D-の接続/切断
- USB3.0 SSTX+/SSTX-の接続/切断
- USB3.0 SSRX+/SSRX-の接続/切断

USB3.0 信号の瞬断による CRC エラー誘発機能

- USB2.0 D+/D-の瞬断
- USB3.0 SSTX+/SSTX-の瞬断
- USB3.0 SSRX+/SSRX-の瞬断

外形寸法及び重量

- 外形寸法 : W120mm X L70mm X H28mm
- 質量 : 50 g

電氣的仕様

- 電源電圧 : DC+5.0V±10%以内
- 装置消費電流 : 100mA (Typ値)
- USB ターゲット機器消費電流(VBUS 電流) : 1A 以下

環境仕様

- 使用温度範囲 : 5 ~ +40°C
- 使用湿度範囲 : 35 ~ 85%RH (但し結露しないこと)
- 有害物質の使用制限 : REACH、RoHS

ソフトウェア仕様

各種 USB エミュレーション動作の設定変更、外部 PC からの制御、ならびに自動処理のカスタマイズ用に、下記ソフトウェアを準備しています。各ソフトウェアは、USB ケーブルで接続した制御用 PC 上で実行します。

ソフトウェア一覧

- ・処理スクリプト書き込みツール for Windows “FWSWriter.exe”
- ・外部 PC 制御用 Windows GUI アプリケーション “DTC2U3App.exe”
- ・外部 PC 制御用 Windows コマンドラインアプリケーション “MLEVGEN.exe”

制御用 PC 動作環境

対応 PC : USB ホスト搭載の Windows PC
対応 OS : Windows XP/Vista/7/8/8.1/10 (32bit/64bit)

1.3 セット内容

ハードウェア構成物

- ・USB ケーブル挿抜試験装置本体

ソフトウェア構成物

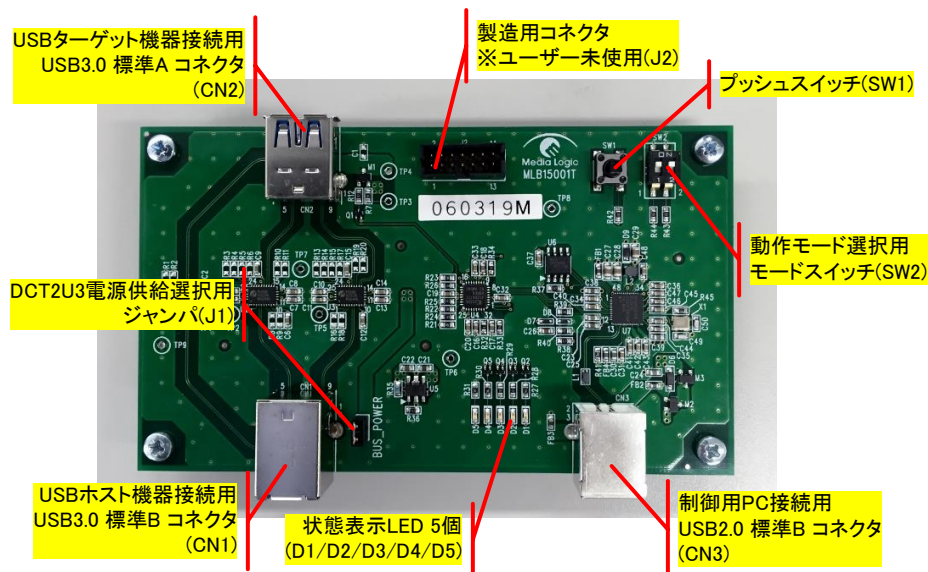
- ・取扱説明書(本書)
- ・処理スクリプト書き込みツール for Windows “FWSWriter.exe”
- ・外部 PC 制御用 Windows GUI アプリケーション “DTC2U3App.exe”
- ・外部 PC 制御用 Windows コマンドラインアプリケーション “MLEVGEN.exe”
- ・制御アプリケーション用 Windows ライブラリ “MLGPLib” *1
- ・制御アプリケーションサンプルソースコード *1
 - “DTC2U3App.exe” ソースコード
 - “MLEVGen.exe” ソースコード

*1: Microsoft Visual Studio 2008/2010/2012 用です。その他の開発環境では使用できません。

※上記ソフトウェアは下記 URL よりダウンロードしてください。

http://www.medialogic.co.jp/Download/Manual_SupportSoft.html

1.4 外観



USB ホスト機器接続用 USB3.0 標準 B コネクタ (CN1)

各種 USB 処理(接続・切断、エラー誘発)を行う対象となる USB ホスト機器を接続します。

USB ターゲット機器接続用 USB3.0 標準 A コネクタ (CN2)

各種 USB 処理(接続・切断、エラー誘発)を行う対象となる USB ターゲット機器を接続します。

プッシュスイッチ (SW1)

各種 USB 処理(接続・切断、エラー誘発)を開始(または停止)するときに使用します。詳細は後述の「操作方法」を参照してください。

動作モード選択用モードスイッチ(SW2)

動作モードを選択します。DTC2U3 の電源を Off した状態で操作してください。

SW2-1	SW2-2	動作モード	説明
OFF	OFF	手動 USB 挿抜モード	プッシュスイッチを押したとき、USB ケーブルの切断(または接続)を行います。
OFF	ON	自動 USB 挿抜モード	10 秒間隔で、USB ケーブルの接続、切断を繰り返します。
ON	OFF	手動 USB エラー誘発モード	プッシュスイッチを押したとき、USB エラーを誘発します。
ON	ON	自動 USB エラー誘発モード	10 秒間隔で、USB エラーを誘発します。

※このモードスイッチ設定は「標準スクリプト」実行時のものです。カスタマイズした処理スクリプトを使用時のモードスイッチ仕様は、処理スクリプトの内容に依存します。

状態表示 LED (D1/D2/D3/D4/D5)

DTC2U3 の各種動作状態を表示します。

LED 名称	LED 色	説明
Power LED (D5)	青	電源 On 時:点灯 電源 Off 時:消灯
LED (D4)	緑	未使用
USB 接続状態 LED (D3)	黄	USB 接続状態時:点灯 USB 切断状態時:消灯
USB エラーLED (D2)	橙	USB エラー誘発時:点灯 通常時:消灯
自動処理状態 LED (D1)	赤	自動処理実行中:点灯 自動処理停止中:消灯

※この LED 仕様は「標準スクリプト」実行時のものです。カスタマイズした処理スクリプトを使用時の LED 仕様は、処理スクリプトの内容に依存します。

制御用 PC 接続用 USB2.0 標準 B コネクタ (CN3)

外部 PC より各種 USB 処理(接続・切断、エラー誘発)を制御する場合、またはカスタマイズした処理スクリプトの本装置への書き込みを行い場合に使用します。制御用 PC を USB ケーブルで接続してください。

DTC2U3 電源供給設定用ジャンパ(J1)

DTC2U3 の電源供給方法を選択します。

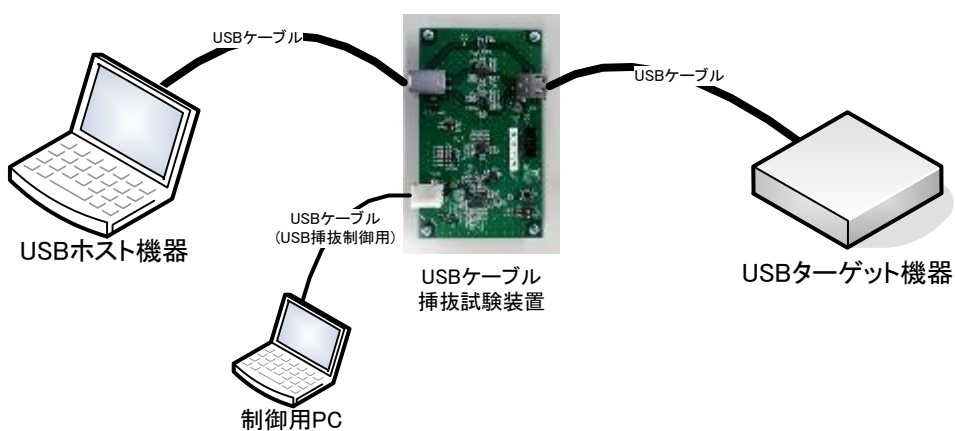
ジャンパ状態	説明
Short	電源供給を CN1 と CN3 で自動切り替え (出荷時設定)
Open	電源供給を CN3 のみとする

※ここでの電源供給とは DTC2U3 制御回路への電源供給を指します。CN2 の VBUS へは CN1 からのみ行われます

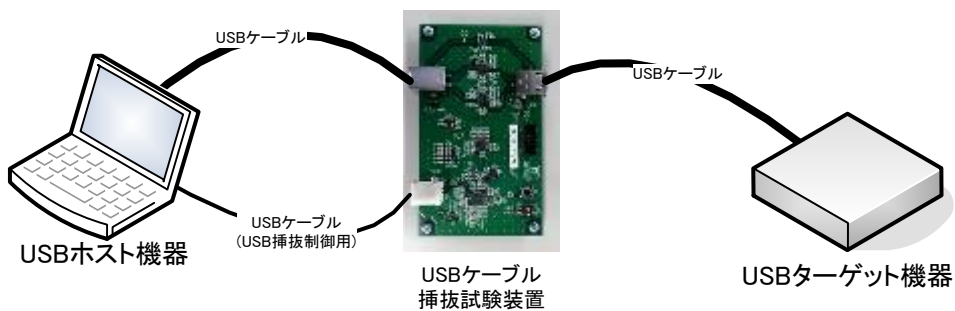
接続例 1 ～スタンドアロン動作～



接続例 2 ～外部制御用 PC より USB 抜き差し制御～



接続例 3 ～USB ホスト機器と外部制御用 PC を兼用～

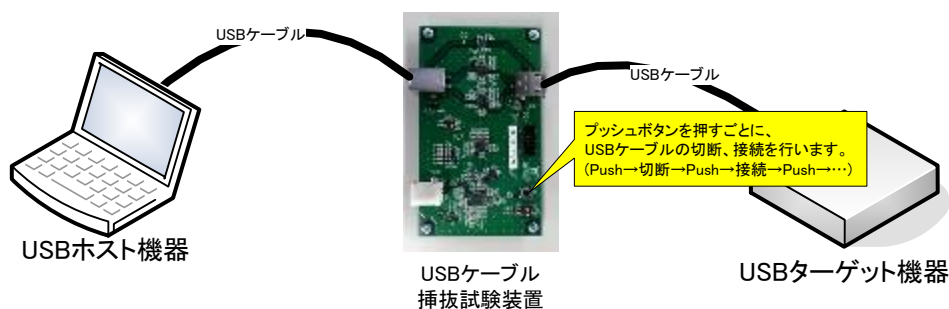


2 操作方法

以下、各動作モードの操作方法を説明します。

2.1 手動 USB 挿抜モード操作方法

接続図



準備手順

[手順 1] 動作モード選択

本装置から全てのケーブルを取り外し、モードスイッチを「SW2-1=OFF, SW2-2=OFF」の状態にしてください。



[手順 2] USB ホスト機器への接続

本装置を USB ホスト機器に USB ケーブルで接続してください。

[手順 3] USB ターゲット機器の接続

USB ターゲット機器を本装置に接続してください。

USB ターゲット機器は USB ホスト機器に USB 接続された状態となります。

以上で準備完了です。下記操作手順にて USB ケーブルの切断、接続を行ってください。

操作手順

[手順 1] USB ケーブルの切断

本装置のプッシュボタン(SW1)を押してください。

- ・本装置は USB ケーブルの切断操作をエミュレートします。
- ・USB ターゲット機器は USB ホスト機器から切断された状態となります。

[手順 2] USB ケーブルの接続

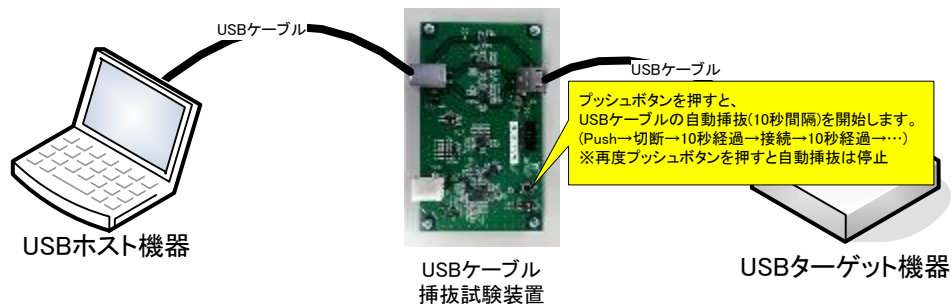
再度、本装置のプッシュボタン(SW1)を押してください。

- ・本装置は USB ケーブルの接続操作をエミュレートします。
- ・USB ターゲット機器は USB ホスト機器に USB 接続された状態となります。

以降、[手順 1]、[手順 2]を繰り返すことで、USB ケーブルの切断、接続を繰り返し行うことができます。

2.2 自動 USB 挿抜モード操作方法

接続図



準備手順

[手順 1] 動作モード選択

本装置から全てのケーブルを取り外し、モードスイッチを「SW2-1=OFF, SW2-2=ON」の状態にしてください。



[手順 2] USB ホスト機器への接続

本装置を USB ホスト機器に USB ケーブルで接続してください。

[手順 3] USB ターゲット機器の接続

USB ターゲット機器を本装置に接続してください。

USB ターゲット機器は USB ホスト機器に USB 接続された状態となります。

以上で準備完了です。下記操作手順にて USB ケーブルの自動挿抜処理を開始してください。

操作手順

[手順 1] USB ケーブル自動挿抜処理の開始

本装置のプッシュボタン(SW1)を押してください。

- 本装置は USB ケーブルの自動挿抜処理を開始します。

- 以降、10 秒間隔で、USB ケーブルの切断操作のエミュレート、USB ケーブルの接続操作のエミュレートを繰り返し行います。

[手順 2] USB ケーブル自動挿抜処理の停止

本装置のプッシュボタン(SW1)を押してください。

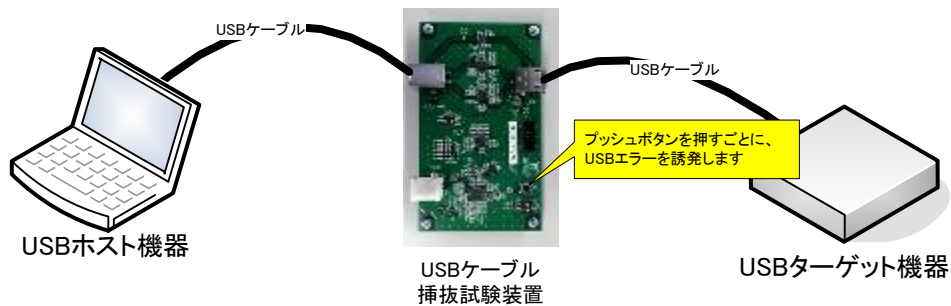
- 本装置は USB ケーブルの自動挿抜処理を停止します。

- プッシュボタン押下時、USB ケーブルが接続状態にあった場合は、USB ターゲット機器は USB ホスト機器に USB 接続された状態となります。プッシュボタン押下時、USB ケーブルが切断状態にあった場合は、USB ターゲット機器は USB ホスト機器から切断された状態となります。

以降、[手順 1]、[手順 2]を繰り返すことで、USB ケーブル自動挿抜処理の開始・停止を繰り返し行うことができます。

2.3 手動 USB エラー誘発モード操作方法

接続図



準備手順

[手順 1] 動作モード選択

本装置から全てのケーブルを取り外し、モードスイッチを「SW2-1=ON, SW2-2=OFF」の状態にしてください。



[手順 2] USB ホスト機器への接続

本装置を USB ホスト機器に USB ケーブルで接続してください。

[手順 3] USB ターゲット機器の接続

USB ターゲット機器を本装置に接続してください。

USB ターゲット機器は USB ホスト機器に USB 接続された状態となります。

以上で準備完了です。下記操作手順にて USB エラー誘発操作を行ってください。

操作手順

[手順 1] USB エラー誘発

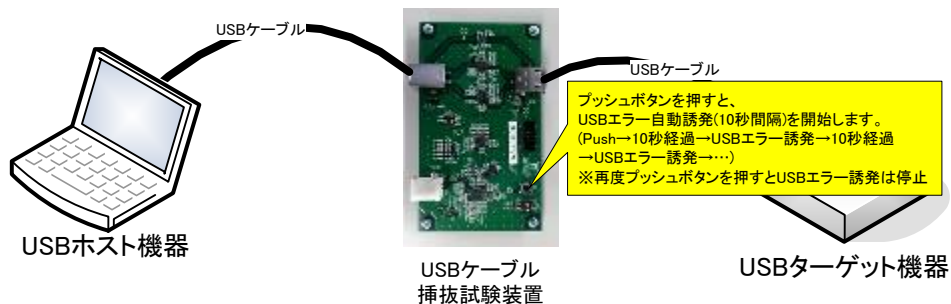
本装置のプッシュボタン(SW1)を押してください。

・本装置は USB エラーを誘発します。

以降、[手順 1]を繰り返すことで、USB エラー誘発を繰り返し行うことができます。

2.4 自動 USB エラー誘発モード操作方法

接続図



準備手順

[手順 1] 動作モード選択

本装置から全てのケーブルを取り外し、モードスイッチを「SW2-1=ON, SW2-2=ON」の状態にしてください。



[手順 2] USB ホスト機器への接続

本装置を USB ホスト機器に USB ケーブルで接続してください。

[手順 3] USB ターゲット機器の接続

USB ターゲット機器を本装置に接続してください。

USB ターゲット機器は USB ホスト機器に USB 接続された状態となります。

以上で準備完了です。下記操作手順にて USB エラー自動誘発処理を開始してください。

操作手順

[手順 1] USB エラー自動誘発処理の開始

本装置のプッシュボタン(SW1)を押してください。

- ・本装置は USB エラー自動誘発処理を開始します。
- ・以降、10 秒間隔で、USB エラーの誘発を繰り返し行います。

[手順 2] USB ケーブル自動挿抜処理の停止

本装置のプッシュボタン(SW1)を押してください。

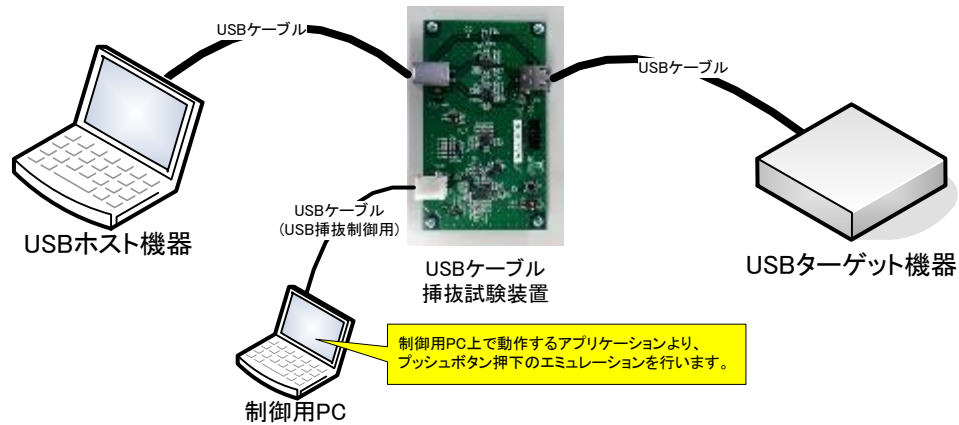
- ・本装置は USB エラー自動誘発処理を停止します。

以降、[手順 1]、[手順 2]を繰り返すことで、USB エラー自動誘発処理の開始・停止を繰り返し行うことができます。

2.5 外部 PC 制御時の操作方法

外部 PC より、各種 USB 制御を行う場合の操作方法を説明します。

接続図



準備手順

[手順 1] 動作モード選択

本装置から全てのケーブルを取り外し、動作モードを選択してください(手動 USB 挿抜モード、自動 USB 挿抜モード、手動 USB エラー誘発モード、自動 USB エラー誘発モード)。

[手順 2] 制御用 PC の接続

本装置に制御用 PC に USB ケーブルで接続してください (CN3 に接続)。

[手順 3] USB ホスト機器への接続

本装置を USB ホスト機器に USB ケーブルで接続してください (CN1 に接続)。

[手順 4] USB ターゲット機器の接続

USB ターゲット機器を本装置に接続してください (CN2 に接続)。

以上で準備完了です。下記操作手順にて、各種 USB 制御を行ってください。

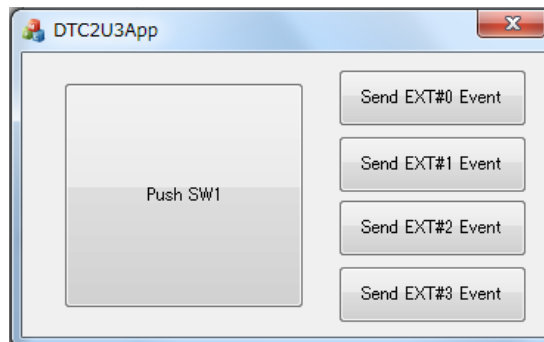
操作手順

制御用 PC 上で起動した制御アプリケーションより、本装置の制御を行います。制御アプリケーションは、GUI アプリケーションとコマンドラインアプリケーションの2種類を準備しております。それぞれの操作方法を説明します。

Windows GUI アプリケーション

[手順 1] 制御アプリケーションの起動

制御用 PC 上で、“DTC2U3App.exe” をダブルクリックしてアプリケーションを起動してください。



[手順 2] プッシュボタン押下のエミュレーション

「Push SW1」ボタンをクリックすることで、本装置のプッシュボタン(SW1)を押した状態をエミュレーションします。プッシュボタン押下時の動作は、選択された動作モードにより異なります(USB 挿抜や USB エラー誘発など)。

Windows コマンドラインアプリケーション

[手順 1] MS-DOS プロンプトの起動

制御用 PC 上の Windows スタートメニューから、「コマンド プロンプト」を起動してください。そして、制御用コマンドラインアプリケーション“MLEVGEN.EXE”の格納されたフォルダーに、カレントフォルダーを移動してください。



[手順 2] プッシュボタン押下のエミュレーション

コマンドプロンプトより、下記のコマンドを入力して実行してください。本装置のプッシュボタン(SW1)を押した状態をエミュレーションします。

プッシュボタン押下時の動作は、選択された動作モードにより異なります(USB 挿抜や USB エラー誘発など)。

<コマンド入力> プッシュボタン押下のエミュレーションを行うコマンド

mlevgen button

<実行例>

```
E:\>mlevgen button

Send Event #4 to DTC2U3.
-> Success.

completed!
```

3 Windows GUI 制御アプリケーション "DTC2U3App.exe" 説明

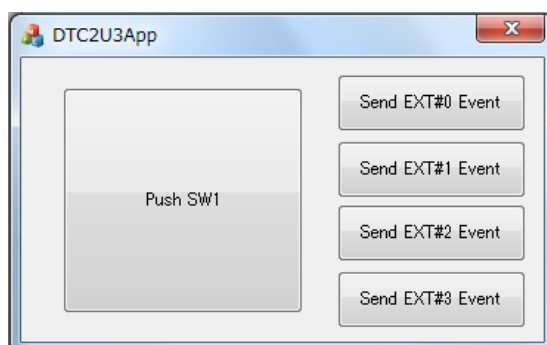
3.1 説明

- DTC2U3App.exe は、DTC2U3 の制御を行うための Windows GUI アプリケーションです。
 - 本プログラムを使用することで、DTC2U3 に対して各「イベント」を送信することができます。イベントを受信した DTC2U3 は、イベントに内容に応じて各種動作を行います。(各イベント送信時の DTC2U3 の動作内容は、DTC2U3 に書き込みされたスクリプト内容に依存します。)
 - 「プッシュボタンイベント」を送信することで、DTC2U3 本体のプッシュボタンが押された状態をエミュレーションすることができます。
 - 本アプリケーションのソースコードは製品に同梱されています。自由にカスタマイズして使用できます。 *1
- *1 : Microsoft Visual Studio 2008/2010/2012 (Win32) 専用です。その他の開発環境では使用できません。

3.2 使用方法

DTC2U3 を USB ケーブルで 制御用 PC に接続した状態で、"DTC2UApp.exe"をダブルクリックして、アプリケーションを起動してください。

3.3 起動画面



各ボタン説明

- [Push SW1] : DTC2U3 のプッシュボタンが押された状態をエミュレーションします。
- [Send EXT#0 Event] : DTC2U3 に外部イベント#0 を送信します。
- [Send EXT#1 Event] : DTC2U3 に外部イベント#1 を送信します。
- [Send EXT#2 Event] : DTC2U3 に外部イベント#2 を送信します。
- [Send EXT#3 Event] : DTC2U3 に外部イベント#3 を送信します。

4 Windows コマンドライン制御アプリケーション “MLEVGEN.exe” 説明

4.1 説明

- ・DTC2U3App.exe は、DTC2U3 の制御を行うための Windows コマンドラインアプリケーションです。
 - ・本プログラムを使用することで、DTC2U3 に対して各「イベント」を送信することができます。イベントを受信した DTC2U3 は、イベントに内容に応じて各種動作を行います。(各イベント送信時の DTC2U3 の動作内容は、DTC2U3 に書き込みされたスクリプト内容に依存します。)
 - ・「プッシュボタンイベント」を送信することで、DTC2U3 本体のプッシュボタンが押された状態をエミュレーションすることができます。
 - ・本アプリケーションのソースコードは製品に同梱されています。自由にカスタマイズして使用できます。 *1
- *1 : Microsoft Visual Studio 2008/2010/2012 (Win32) 専用です。その他の開発環境では使用できません。

4.2 使用方法

- ・DTC2U3 を USB ケーブルで 制御用 PC に接続してください。
- ・Windows のスタートメニューより、「コマンド プロンプト」を起動してください。
- ・コマンドプロンプトのカレントフォルダーを、“MLEVGEN.EXE”の格納されたフォルダーに移動してください。
- ・コマンドプロンプトより、下記の書式でコマンド実行してください。
- ・制御用 PC に複数の DTC2U3 が接続されている場合は、/SN オプションを指定して、制御対象とする DTC2U3 を選択してください。/SN オプション指定を省略した場合は、最初に見つかった DTC2U3 を制御対象とします。

コマンド書式

MLEVGEN (オプション) [イベント名] ※オプションは省略可能

イベント名

button : DTC2U3 のプッシュボタンが押された状態をエミュレーションします。

ext0 : DTC2U3 に外部イベント#0 を送信します。

ext1 : DTC2U3 に外部イベント#1 を送信します。

ext2 : DTC2U3 に外部イベント#2 を送信します。

ext3 : DTC2U3 に外部イベント#3 を送信します。

オプション

/LIST : 現在接続されている DTC2U3 デバイスの一覧を表示します。 ※Ver.1.02 以降のみ対応

/SN=XXXXXXXX : 制御対象とする DTC2U3 の USB シリアル番号を指定します。 ※Ver.1.02 以降のみ対応

(XXXXXXXX : USB シリアル番号文字列 ※最大 16 文字)

4.3 用例 1 ～プッシュボタン押下をエミュレーション～

MLEVGEN button

4.4 用例 2 ～外部イベント#0 を送信～

MLEVGEN ext0

4.5 用例 3 ～制御対象とする DTC2U3 を選択して、外部イベント#0 を送信～

MLEVGEN /SN=0000201507280721 ext0

(USB シリアル番号="0000201507280721"の DTC2U3 を制御対象とする場合)

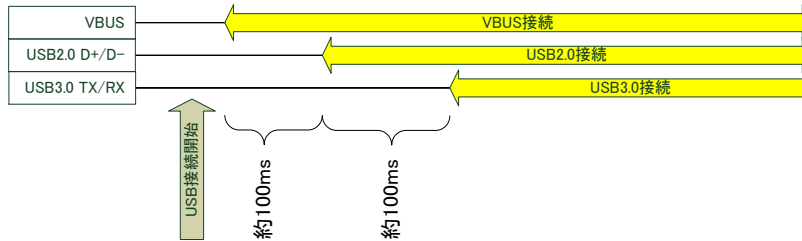
※MLEVGEN.EXE Ver.1.02 以降のみ対応です。

5 各種エミュレーション動作設定

5.1 USB ケーブル接続エミュレーション動作設定

「VBus ピン接続」から「USB2.0 D+/D-ピン接続」までの時間間隔 : 100 ms

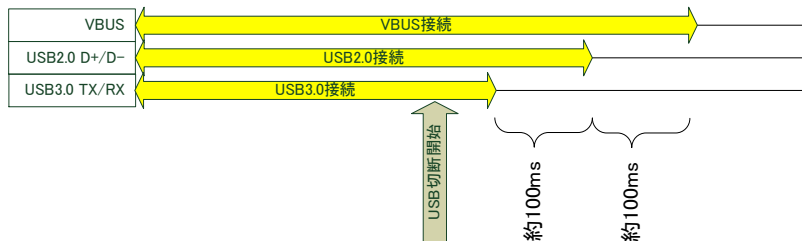
「USB2.0D+/D-ピン接続」から「USB3.0 TX/RX ピン接続」までの時間間隔 : 100 ms



5.2 USB ケーブル切断エミュレーション動作設定

「USB3.0 TX/RX ピン切断」から「USB2.0 D+/D-ピン切断」までの時間間隔 : 100 ms

「USB2.0 D+/D-ピン切断」から「VBus ピン切断」からまでの時間間隔 : 100 ms

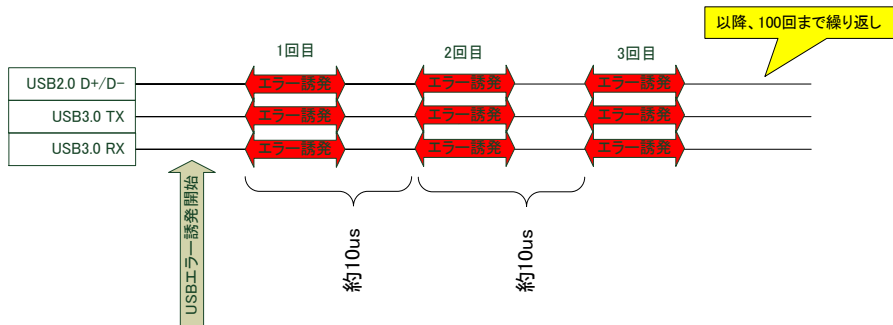


5.3 USB エラー誘発設定

USB エラー種別 : USB2.0 エラー、USB3.0 TX エラー、USB3.0 RX エラー同時誘発

USB エラー誘発回数 : 100 回

USB エラー誘発間隔 : 約 10 us



6 処理スクリプトによる各種カスタマイズ

本装置は、本装置内部に格納された「処理スクリプト」を実行することで、各種 USB 挿抜制御、ならびに USB エラー誘発制御の自動処理を行っています。処理スクリプトの内容を変更することで、各エミュレート処理の動作設定変更や、自動処理手順をカスタマイズすることが可能です。

カスタマイズの例

- USB ケーブルをゆっくり接続する動作、または早く接続行う動作をエミュレートする。
 - USB ケーブル接続&切断を高速に 100 回行う動作をエミュレートする。
 - 「USB2.0 ケーブル接続&切断」、「USB3.0 ケーブル接続&切断」を交互に行う。
 - USB3.0 TX エラーのみ誘発させる。
 - USB ケーブルの挿抜を1時間毎に発生させる。
 - USB ケーブルの瞬断を発生させる。
- など

6.1 処理スクリプト概要

- 本装置は USB 制御用レジスタ(USBCTL)と、各種タイミング制御用のレジスタを持ちます。処理スクリプトにて、これらのレジスタ値を読み書きする処理手順を記述することで、各種自動制御処理を実現します。
- 処理スクリプトはテキストファイルで記述します。PC 上で動作するテキストエディタなどを使用して、処理スクリプトを編集することができます。
- 編集した処理スクリプトは、製品添付の処理スクリプト書き込みツール "FWSWriter.exe" を使用して、本装置の不揮発領域に保存します。

保存された処理スクリプトは、本装置の電源投入時に自動的に処理開始されます。(制御 PC 不要。スタンドアロン動作)

処理スクリプトファイル仕様

- ファイル形式 : テキストファイル (Shift-JIS 形式のみ対応)
- ファイル拡張子 : "FWS" (例 "usb_connect.fws")
- 最大命令数 : 496 個

処理スクリプトファイルの例

```

; USB2.0 接続した後、50ms 待った後、USB エラーを誘発させる。
CALL USB2CONNECT      ; USB 接続
WAIT 50000             ; 50ms ウェイト
CALL USB2ERR          ; USB エラー誘発
END                   ; 処理スクリプト終了

; USB2.0 接続サブルーチン
:USB2CONNECT
SET USBCTL 0b00001000 ; VBUS 接続
WAIT 1000             ; 1000us ウェイト
SET USBCTL 0b00001100 ; USB2.0 接続
RETURN                ; CALL 元へ復帰

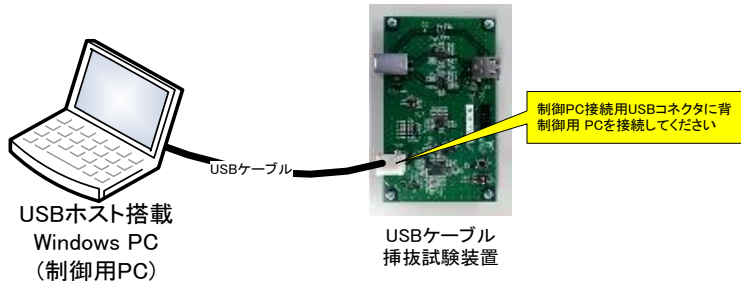
```

7 処理スクリプト書き込みツール “FWSWriter.exe” 使用方法

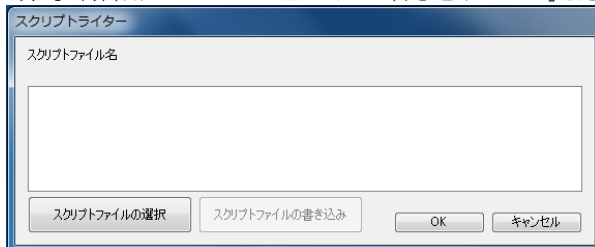
手順

[手順 1] 制御用 PC と接続

本装置の制御 PC 接続用 USB ポートに、制御用 PC を接続してください。

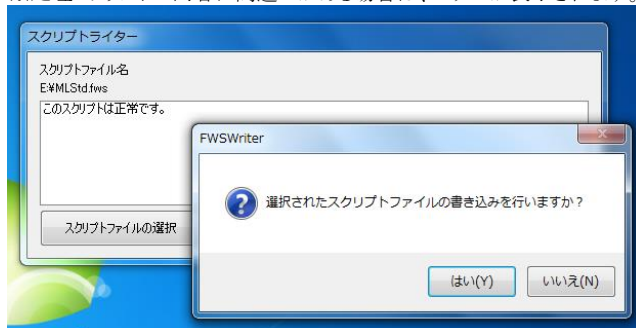


[手順 2] 制御用 PC 上で「処理スクリプト書き込みツール」を起動してください。



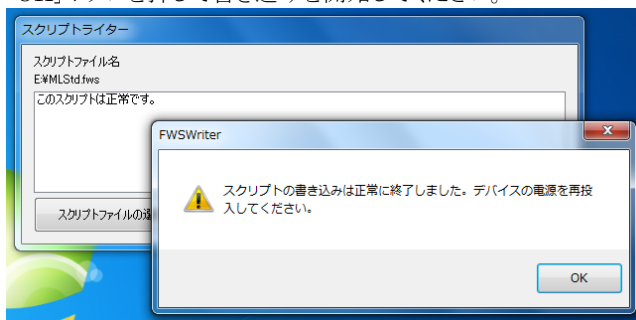
[手順 3] 書き込みツールの「スクリプトファイル選択」ボタンを押して、書き込みを行いたい処理スクリプトファイル(FWS ファイル)を選択してください。

※処理スクリプトの内容に間違いがある場合は、エラーが表示されます。処理スクリプトの内容を正しく修正してください。



[手順 4] 本装置が正しく制御用 PC に接続されている場合、書き込み確認のダイアログが表示されます。

「OK」ボタンを押して書き込みを開始してください。



[手順 5] 以上で処理スクリプトの書き込みは終了です。

本装置の USB ケーブルを全て切断して、本装置の電源を一旦 Off にしてください。電源再投入後、書き込みを行った処理スクリプトは自動的に実行開始されます。

8 処理スクリプト基本書式

処理スクリプトは下記書式で記述してください。

8.1 命令書式

引数がない命令の場合

[命令] [改行文字]

引数がある命令の場合

[命令] [引数 1] [改行文字]

引数がある命令の場合

[命令] [引数 1] [引数 2] [改行文字]

- 必要な引数の個数は、命令毎に異なります。
- [命令]と[引数 1]の間、または[引数 1]と[引数 2]の間は、一つ以上の区切り文字(空白またはタブ文字)を挿入してください。※空白文字は半角文字を使用してください。
- 引数は、命令の種類によって、[レジスタ名]、[数値]、[ラベル名]のいずれかを指定します。
- 命令行の最後尾は改行してください。一行に複数の命令を続けて記述することはできません。
- 各行先頭の区切り文字(空白またはタブ文字)は無視されます。

8.2 ラベル書式

分岐命令による分岐先を「ラベル」と定義します。ラベルは下記の書式で記述してください。

ラベル書式

:[ラベル名称] [改行文字]

- 先頭文字が“:”(半角コロン)の場合、その行はラベルとして扱います。コロンの後の文字列をラベル名称とします。
- ラベル名称は、ユーザーが自由に指定できます。(ただし、半角英数文字を使用してください)
- ラベル名称は大文字・小文字を区別しません。

8.3 コメント書式

行の先頭が“;”(半角セミコロン)の場合、その行は無視されます。処理スクリプトのコメント、注釈の記述に使用してください。コメント分は全角文字(日本語など)も使用可能です。

コメント書式

:[コメント行] [改行文字]

8.4 引数書式

引数がレジスタの場合

レジスタ名称を指定してください。

例)

USBCTL
EVENT
STATUS
TIM0
A
など

引数が数値の場合

- 引数が数値の場合、2 進数、10 進数、16 進数のいずれかの書式で指定してください。
- マイナス(負)の数値は使用できません。
- 指定可能な数値の最大ビット長は 16bit です。
- 16bit 長に満たない数値を引数に指定した場合、指定の無い上位 bit は自動的に 0 となり、結果 16bit 長の数値に拡張されます。(例「0b0111 を指定」 → 「0b0000000000000111 を指定」に相当する)

[2 進数の場合] 数値の先頭に"0b"または"0B"をつける。

例)

0b0
0b101
0B00101001
0b1111111111111111

[10 進数の場合] 数値の先頭に何もつけない。または数値の先頭に"0d"または"0D"をつける

例)

0
128
0d123
0D65535

[16 進数の場合] 数値の先頭に"0x"または"0X"をつける。

例)

0x0
0xF
0X01A9
0xFFFF

引数がラベルの場合

ラベル名称を指定してください。(先頭のコロンは除く)

9 制御レジスタ説明

本装置は USB 制御用レジスタ1つと、その他タイミングや手順制御用の各種レジスタを持ちます。処理スクリプトにて、各レジスタを読み書きして、各種制御を行ってください。

9.1 制御レジスタ一覧

レジスタ名称	略称	説明
USB 制御レジスタ	USBCTL	各種 USB の制御を行う
LED 制御レジスタ	LEDCTL	各 LED の制御を行う
ステータスレジスタ	STATUS	モードスイッチ、プッシュボタンの状態を取得する
イベントレジスタ	EVENT	各種イベントの発生状況を取得する
タイマーレジスタ#0	TIM0	時間計測用レジスタ#0 (100ms 毎に-1 される)
タイマーレジスタ#1	TIM1	時間計測用レジスタ#1 (100ms 毎に-1 される)
タイマーレジスタ#2	TIM2	時間計測用レジスタ#2 (100ms 毎に-1 される)
タイマーレジスタ#3	TIM3	時間計測用レジスタ#3 (100ms 毎に-1 される)
汎用レジスタ A	A	処理スクリプトで使用可能な汎用レジスタ A
汎用レジスタ B	B	処理スクリプトで使用可能な汎用レジスタ B
汎用レジスタ C	C	処理スクリプトで使用可能な汎用レジスタ C
汎用レジスタ D	D	処理スクリプトで使用可能な汎用レジスタ D
汎用レジスタ E	E	処理スクリプトで使用可能な汎用レジスタ E
汎用レジスタ F	F	処理スクリプトで使用可能な汎用レジスタ F
汎用レジスタ G	G	処理スクリプトで使用可能な汎用レジスタ G
汎用レジスタ H	H	処理スクリプトで使用可能な汎用レジスタ H

※全てのレジスタは 16bit 長です。

9.2 USB 制御レジスタ USBCTL

ビット定義

Bit	bit 15-8	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
USBCTL	USBERR REPEAT	Rsv	U2 ERR	U3RX ERR	U3TX ERR	VBUS CONN	U2 CONN	U3RX CONN	U3TX CONN

※Reserved(Rsv)フィールドには 0 を設定すること

各フィールド説明

U3TX CONN

USB ホスト機器と USB ターゲット機器間の USB3.0 TX 信号の接続状態を制御します。

- 1 : USB3.0 TX 信号を接続。
- 0 : USB3.0 TX 信号を切断。

U3RX CONN

USB ホスト機器と USB ターゲット機器間の USB3.0 RX 信号の接続状態を制御します。

- 1 : USB3.0 RX 信号を接続。
- 0 : USB3.0 RX 信号を切断。

U2 CONN

USB ホスト機器と USB ターゲット機器間の USB2.0 D+/D-信号の接続状態を制御します。

- 1 : USB2.0 D+/D-信号を接続。
- 0 : USB2.0 D+/D-信号を切断。

VBUS CONN

USB ホスト機器と USB ターゲット機器間の VBUS 信号の接続状態を制御します。

- 1 : VBUS 信号を接続。
- 0 : VBUS 信号を切断。

U3TX ERR

USB ホスト機器と USB ターゲット機器間の USB3.0 TX 信号上にエラー誘発します。

- 1 : USB3.0 TX 信号上にエラーを誘発。 ※本ビットは自動的に 0 にクリアされます
- 0 : USB3.0 TX 信号上にエラーを誘発させない。

U3RX ERR

USB ホスト機器と USB ターゲット機器間の USB3.0 RX 信号上にエラーを誘発します。

- 1 : USB3.0 RX 信号上にエラーを誘発。 ※本ビットは自動的に 0 にクリアされます
- 0 : USB3.0 RX 信号上にエラーを誘発させない。

U2 ERR

USB ホスト機器と USB ターゲット機器間の USB2.0 D+/D-信号上にエラーを誘発します。

- 1 : USB2.0 D+/D-信号上にエラーを誘発。 ※本ビットは自動的に 0 にクリアされます
- 0 : USB2.0 D+/D-信号上にエラーを誘発させない。

USBERR REPEAT

USB エラー(U2ERR/U3TXERR/U3RXERR)を繰り返し誘発する回数を指定します。 ※本フィールドは自動的に 0 にクリアされます。

- 0 : USB エラーを 1 回誘発します。
- N (0 以外) : USB エラーを [N+1]回連続して誘発します。(連続エラー誘発の間隔 : 約 10us)

9.3 LED 制御レジスタ LEDCTL

ビット定義

Bit	bit 15-4	bit 3	bit 2	bit 1	bit 0
LEDCTL	Reserved (0)	LED4 (緑)	LED3 (黄)	LED2 (橙)	LED1 (赤)

※Reserved フィールドには 0 を設定すること

各フィールド説明

LED1 (赤)

本装置上の赤 LED (D1)の点灯状態を制御します。

- 1：赤 LED (D1)を点灯。
- 0：赤 LED (D1)を消灯。

LED2 (橙)

本装置上の橙 LED (D2)の点灯状態を制御します。

- 1：橙 LED (D2)を点灯。
- 0：橙 LED (D2)を消灯。

LED3 (黄)

本装置上の黄 LED (D3)の点灯状態を制御します。

- 1：黄 LED (D3)を点灯。
- 0：黄 LED (D3)を消灯。

LED4 (緑)

本装置上の緑 LED (D4)の点灯状態を制御します。

- 1：緑 LED (D4)を点灯。
- 0：緑 LED (D4)を消灯。

9.4 ステータスレジスタ STATUS

ビット定義

Bit	bit 15-3	bit 2	bit 1	bit 0
STATUS	Reserved (0)	PUSHSW	MODESW2	MODESW1

※本レジスタは読み取り専用です

各フィールド説明

MODESW1

本装置上のモードスイッチ(SW2)の SW2-1 の設定状態を取得します。

- 1 : SW2-1 は ON 状態
- 0 : SW2-1 は OFF 状態

MODESW2

本装置上のモードスイッチ(SW2)の SW2-2 の設定状態を取得します。

- 1 : SW2-2 は ON 状態
- 0 : SW2-2 は OFF 状態

PUSHSW

本装置上のプッシュボタン(SW1)の押下状態を取得します。

- 1 : プッシュボタンは押されている状態
- 0 : プッシュボタンは押されていない状態

9.5 イベントレジスタ EVENT

ビット定義

Bit	bit 15-12	bit 11	bit 10	bit 9	bit 8	bit 7-5	bit 4	bit 3	bit 2	bit 1	bit 0
EVENT	Reserved	EXT3	EXT2	EXT1	EXT0	Reserved	SW PUSH	TIM3 EXP	TIM2 EXP	TIM1 EXP	TIM0 EXP

各イベント発生時、それぞれ対応するビットが1にセットされます。

各ビットのクリアは自動的に0にクリアされません。処理スクリプト側でクリアしてください。

各ビットのクリアする場合、クリアしたいビットに1をセットしてください。※0をライトしてもクリアされません。

各フィールド説明

TIM0 EXP

時間計測用 TIM0 レジスタ値が 0 になったことを通知します。

1：TIM0 レジスタ値が 0 になった。

0：TIM0 レジスタ値が 0 以外。または TIM0 レジスタ値が 0 のまま変化無し。

TIM1/2/3 EXP

(TIM1 EXP, TIM2 EXP, TIM3 EXP の説明は TIM0 EXP と同様のため省略します)

SW PUSH

プッシュスイッチが押されたことを通知します。

1：プッシュスイッチが押された。

0：プッシュスイッチが押されていない。

EXT0

外部イベント#0 が発生した通知します。

1：外部イベント#0 が発生した。

0：外部イベント#0 が発生していない。

※外部イベントは、制御用 PC 上で動作するユーザーアプリケーションより、USB 経由で本装置に対して通知されます。

EXT1/2/3 EXP

(EXT1, EXT2, EXT3 の説明は EXT0 と同様のため省略します)

9.6 タイマーレジスタ #0 TIM0

ビット定義

Bit	bit 15-0
TIM0	TIMER0

各フィールド説明

TIMER0

時間計測用タイマー値です。※秒オーダーの時間計測用です。
処理スクリプトにて、本フィールド値に計測したい時間(単位 100msec)を設定してください。
本フィールド値は 100msec 毎に 1 減算されます。※0 になった場合、それ以降、値は 0 のまま維持されます。
本フィールド値が 0 になった時点で、イベントレジスタの TIM0 EXP ビットに 1 がセットされます。
※本フィールドに設定可能な最大値は FFFFh(65535)です。最大約 109 分までの時間計測が可能です。
※本フィールド設定後、初回の減算は 100ms 以内に行われる場合があります。

9.7 タイマーレジスタ #1/#2/#3 TIM1/TIM2/TIM3

(TIM1, TIM2, TIM3 レジスタの説明は、TIM0 レジスタの説明と同様のため省略します)

9.8 汎用レジスタ A/B/C/D/E/F/G/H

ビット定義

Bit	bit 15-0
A	A
B	B
C	C
D	D
E	E
F	F
G	G
H	H

各フィールド説明

A/B/C/D/E/F/G/H

処理スクリプトで使用可能な汎用レジスタです。
各種状態保存、回数カウント用変数など、自由に使用してください。

10 処理スクリプト命令説明

10.1 命令一覧

本装置の対応する処理スクリプト命令は下記の通りです。これら命令を使用して、レジスタ制御ならびに処理手順の制御を行います。

基本命令一覧

命令	引数1	引数2	説明
SET	レジスタ	値	指定レジスタに指定値をセットする
MOVE	レジスタ1	レジスタ2	指定レジスタ2の値を、指定レジスタ1にセットする
WAIT	値	—	指定時間(単位:usec)待つ
NOP	—	—	何もしない
JUMP	ラベル	—	指定ラベルに分岐する
CALL	ラベル	—	指定ラベルに分岐する(分岐戻り位置を保存)
RETURN	—	—	CALL命令で保存された分岐戻り位置に戻る
END	—	—	処理スクリプトを終了する

レジスタ演算命令一覧

命令	引数1	引数2	説明
ADD	レジスタ	値	指定レジスタに指定値を「加算」する
SUB	レジスタ	値	指定レジスタから指定値を「減算」する
AND	レジスタ	値	指定レジスタに指定値を「AND 演算」する
OR	レジスタ	値	指定レジスタに指定値を「OR 演算」する
XOR	レジスタ	値	指定レジスタに指定値を「XOR 演算」する
NOT	レジスタ	—	指定レジスタを「NOT 演算」する (ビット反転)

条件分岐命令一覧

命令	引数1	引数2	説明
CMP	レジスタ	値	指定レジスタ値と指定値を比較して、その結果を保存する。
JUMPEQ	ラベル	—	直前の CMP 命令の比較結果が「等しい」場合、指定ラベルに分岐する。
JUMPNE	ラベル	—	直前の CMP 命令で比較結果が「等しくない」場合、指定ラベルに分岐する。
JUMPGT	ラベル	—	直前の CMP 命令の比較結果が「指定レジスタ > 指定値」の場合、指定ラベルに分岐する。
JUMPGE	ラベル	—	直前の CMP 命令の比較結果が「指定レジスタ ≥ 指定値」の場合、指定ラベルに分岐する。
JUMPLT	ラベル	—	直前の CMP 命令の比較結果が「指定レジスタ < 指定値」の場合、指定ラベルに分岐する。
JUMPLE	ラベル	—	直前の CMP 命令の比較結果が「指定レジスタ ≤ 指定値」の場合、指定ラベルに分岐する。

命令	引数1	引数2	説明
CHKBIT	レジスタ	ビット値	指定レジスタ値と指定ビット値をビット比較して、その結果を保存する。
JUMPNZ	ラベル	—	直前の CHKBIT 命令で「指定ビットがセットされている」場合、指定ラベルに分岐する。
JUMPZ	ラベル	—	直前の CHKBIT 命令で「指定ビットがセットされていない」場合、指定ラベルに分岐する。

10.2 SET

書式

SET [レジスタ] [値]

引数

引数 1：値をセットしたいレジスタ名

引数 2：レジスタにセットしたい値

説明

引数 1 に指定したレジスタに、引数 2 に指定した値をセットします。

用例

; USBCTL レジスタに 00001111(2 進数)をセット

```
SET USBCTL 0b00001111
```

; A レジスタに 100 (10 進数)をセット

```
SET A 100
```


10.3 MOVE

書式

MOVE [レジスタ 1] [レジスタ 2]

引数

引数 1：値をセットしたいレジスタ名

引数 2：レジスタ 1 にセットしたい値を保持しているレジスタ名

説明

引数 2 に指定したレジスタ 2 が保持している値を、引数1に指定したレジスタ 1 にセットします。

用例

; レジスタ A の値を、レジスタ B にセット

MOVE B A

; EVENT レジスタの値を、A レジスタにセット

MOVE A EVENT

10.4 WAIT

書式

WAIT [値]

引数

引数 1: ウェイト時間値 (単位:usec)

説明

引数 1 に指定した時間間隔の待ち時間を挿入します。

用例

```
; USB 接続処理。
; 「VBUS 接続→USB2.0 接続→USB3.0 接続」のそれぞれの間に 1000us の待ち時間を挿入
SET USBCTL 0b00001000 ; VBUS 接続
WAIT 1000 ; 1000us 待つ
SET USBCTL 0b00001100 ; USB2.0 接続
WAIT 1000 ; 1000us 待つ
SET USBCTL 0b00001111 ; USB3.0 接続
```

10.5 NOP

書式

NOP

引数

引数無し

説明

何もしない

10.6 JUMP

書式

JUMP [ラベル名]

引数

引数 1: 分岐先ラベル名

説明

引数 1 に指定したラベルに分岐します。

用例

```
; LED 点滅を繰り返す
:LEDBLINK
SET LEDCTL 0b00000001 ; LED 赤点灯
WAIT 50000 ; 50ms 空ける
SET LEDCTL 0b00000000 ; LED 赤消灯
WAIT 50000 ; 50ms 空ける
JUMP LEDBLINK ; LED 点滅を繰り返し行う
```

10.7 CALL

書式

CALL [ラベル名]

引数

引数 1: 分岐先のラベル名

説明

引数 1 に指定したラベルに分岐します。本 CALL 命令の次の命令の位置を、復帰先として保存します。本 CALL 命令により分岐した後、後述の RETURN 命令を実行することで、本 CALL 命令の次の命令位置に戻ることができます。

最大 16 回まで、多重に CALL 命令を発行することが可能です。※17 回以上の多重 CALL は未対応です。

用例

```

; USB2.0 接続した後、50ms 待った後、USB エラーを誘発させる。
CALL USB2CONNECT      ; USB 接続
WAIT 50000             ; 50ms ウェイト
CALL USB2ERR          ; USB エラー誘発
END                   ; 処理スクリプト終了(これ以降の命令は実行されません)

; USB2.0 接続サブルーチン
:USB2CONNECT
SET USBCTL 0b00001000 ; VBUS 接続
WAIT 1000
SET USBCTL 0b00001100 ; VBUS & USB2.0 接続
RETURN                ; CALL 元へ復帰

; USB2.0 エラー誘発サブルーチン
:USB2ERR
SET USBCTL 0b01001111 ; USB エラー誘発
CALL REDLED           ; 赤 LED 点灯(多重 CALL)
RETURN                ; CALL 元へ復帰

; 赤 LED 点灯サブルーチン
:REDLED
SET LEDCTL 0b00000001 ; 赤 LED 点灯
RETURN                ; CALL 元へ復帰

```

10.8 RETURN

書式

RETURN

引数

引数無し

説明

CALL 命令により保存された復帰先に分岐します。(保存された復帰先情報は削除されます。)
多重に CALL 命令が発行されている場合、最後に発行された CALL 命令により保存された復帰先に分岐します。
※CALL 命令が発行されていない状況で、RETURN 命令は発行しないでください。

用例

(省略) CALL 命令の用例を参照してください。

10.9 END

書式

END

引数

引数無し

説明

スクリプト処理を終了します。

10.10 ADD

書式

ADD [レジスタ名] [値]

引数

引数 1: 値を加算したいレジスタ名

引数 2: レジスタに加算したい値

説明

引数 1 に指定したレジスタに、引数2に指定した値を加算します。

用例

; A レジスタに 0 をセットして、その値に1を加算し、さらに3を加算します。

SET A 0 ; A = 0

ADD A 1 ; A = A(0) + 1 → A の値は 1 になります

ADD A 3 ; A = A(1) + 3 → A の値は 4 になります

10.11 SUB

書式

SUB [レジスタ名] [値]

引数

引数 1: 値を減算したいレジスタ名

引数 2: レジスタより減算したい値

説明

引数 1 に指定したレジスタより、引数2に指定した値を減算します。

用例

; A レジスタに 100 をセットして、その値から1を減算し、さらに3を減算します。

SET A 100 ; A = 100

SUB A 1 ; A = A(100) - 1 → A の値は 99 になります

SUB A 3 ; A = A(99) - 3 → A の値は 96 になります

10.12 AND

書式

AND [レジスタ名] [値]

引数

引数 1: 値を AND 演算したいレジスタ名

引数 2: レジスタに AND 演算したい値

説明

引数 1 に指定したレジスタに、引数2に指定した値を AND 演算します。

用例

```
; A レジスタに 1100(2 進数)をセットして、その値に 0111(2 進数)を AND します
SET A 0b1100      ; A = 1100(2 進数)
AND A 0b0111     ; A = A(1100) AND 0111 → A の値は 0100(2 進数)になります
```

10.13 OR

書式

OR [レジスタ名] [値]

引数

引数 1: 値を OR 演算したいレジスタ名

引数 2: レジスタに OR 演算したい値

説明

引数 1 に指定したレジスタに、引数2に指定した値を OR 演算します。

10.14 XOR

書式

SET [レジスタ名] [値]

引数

引数 1: 値を XOR 演算したいレジスタ名

引数 2: レジスタに XOR 演算セットしたい値

説明

引数 1 に指定したレジスタに、引数2に指定した値を XOR 演算します。

10.15 NOT

書式

NOT [レジスタ名]

引数

引数 1: 値を NOT 演算したいレジスタ名

説明

引数 1 に指定したレジスタを NOT 演算します。

用例

```
; Aレジスタに 1100(2進数)をセットして、その値を NOT 演算します
SET A 0b1100      ; A = 000000001100(2進数)
NOT A             ; Aの値は 111111110011(2進数)になります
```

10.16 CMP

書式

CMP [レジスタ名] [値]

引数

引数 1：値を比較したいレジスタ名

引数 2：レジスタと比較したい値

説明

引数 1 に指定したレジスタと、引数 2 に指定した値を比較して、比較結果を保存します。後述の条件分岐命令は、本比較結果に従って分岐するかどうかの判断を行います。

※比較結果の保存領域は、後述の CHKBIT 命令のビット確認結果の保存領域と共通です。CHKBIT 命令を実行すると、CMP 命令の比較結果は破棄されますのでご注意ください。

用例

```
; USB エラーを 10 回連続で誘発させる (回数カウント用に A レジスタを使用)
SET A 0          ; A レジスタをゼロで初期化する

:LOOP           ; USB エラー繰り返し用ラベル
CALL USB2ERR    ; USB エラー誘発
ADD A 1         ; A レジスタの値を+1
CMP A 10      ; A レジスタの値を 10 と比較
JUMPEQ COMPLETE ; 一致した場合、終了処理へ分岐する
JUMP LOOP       ; 一致しなかった場合、USB エラー誘発を繰り返す

; スクリプト終了
:COMPLETE
END
```

10.17 JUMPEQ

書式

JUMPEQ [ラベル名]

引数

引数 1：比較結果が等しい場合に分岐するラベル名

説明

直前に発行された CMP 命令の比較結果が等しい場合、引数 1 に指定したラベルに分岐します。比較結果が等しくない場合は、次の命令へ進みます。

用例

(省略) CMP 命令の用例を参照してください。

10.18 JUMPNE

書式

JUMPNE [ラベル名]

引数

引数 1：比較結果が等しくない場合に分岐するラベル名

説明

直前に発行された CMP 命令の比較結果が等しくない場合、引数 1 に指定したラベルに分岐します。比較結果が等しい場合は、次の命令へ進みます。

10.19 JUMPGT

書式

JUMPGT [ラベル名]

引数

引数 1: 比較結果が大きい場合 (レジスタ > 値) に分岐するラベル名

説明

直前に発行された CMP 命令の比較結果が大きい場合 (レジスタ > 値)、引数 1 に指定したラベルに分岐します。比較結果が大きくない場合 (レジスタ ≤ 値) は、次の命令へ進みます。

10.20 JUMPGE

書式

JUMPGE [ラベル名]

引数

引数 1: 比較結果が大きいまたは等しい場合 (レジスタ ≥ 値) に分岐するラベル名

説明

直前に発行された CMP 命令の比較結果が大きいまたは等しい場合 (レジスタ ≥ 値)、引数 1 に指定したラベルに分岐します。比較結果が小さい場合 (レジスタ < 値) は、次の命令へ進みます。

10.21 JUMPLT

書式

JUMPEQ [ラベル名]

引数

引数 1：比較結果が小さい場合 (レジスタ < 値) に分岐するラベル名

説明

直前に発行された CMP 命令の比較結果が小さい場合 (レジスタ < 値)、引数 1 に指定したラベルに分岐します。比較結果が小さくない場合 (レジスタ \geq 値) は、次の命令へ進みます。

10.22 JUMPLE

書式

JUMPEQ [ラベル名]

引数

引数 1：比較結果が小さいかまたは等しい場合 (レジスタ \leq 値) に分岐するラベル名

説明

直前に発行された CMP 命令の比較結果が小さいかまたは等しい場合 (レジスタ \leq 値)、引数 1 に指定したラベルに分岐します。比較結果が大きい場合 (レジスタ > 値) は、次の命令へ進みます。

10.23 CHKBIT

書式

CHKBIT [レジスタ名] [値]

引数

引数 1：ビット値を比較したいレジスタ名

引数 2：レジスタと比較したいビット値

説明

引数 1 に指定したレジスタの、引数2に指定したビット値がセットされているかどうか確認して、確認結果を保存します。後述の条件分岐命令(JUMPNZ, JUMPZ)は、本確認結果に従って分岐するかどうかの判断を行います。

※確認結果の保存領域は、前述の CMP 命令の比較結果の保存領域と共通です。CMP 命令を実行すると CHKBIT 命令の確認結果は破棄されますのでご注意ください。

用例

```

; 30 秒経過した後、USB 接続を開始する
; EVENT レジスタの TIM0 ビットがセットされているかどうか確認します

SET EVENT 0x0001      ; TIM0 イベントビット(bit0)をクリア
SET TIM0 300          ; 30 秒計測のため、TIM0 レジスタに用に 300 をセット(300×100ms)

:CHKEVENT              ; イベント発生待ちループ用ラベル
CHKBIT EVENT 0x0001   ; TIM0 ビットがセットされているかどうかチェック
JUMPNZ TIM0EXPIRED    ; TIM0 ビットがセットされた(30 秒経過した)ので分岐
JUMP CHKEVENT         ; イベントチェックを繰り返す

; 30 秒経過
:TIM0EXPIRED
CALL USBCONNECT       ; USB 接続
END

```

10.24 JUMPNZ

書式

JUMPNZ [ラベル名]

引数

引数 1：CHKBIT 命令のビット確認結果、ビットがセットされていた場合に分岐するラベル名

説明

直前に発行された CHKBIT 命令のビット確認の結果、ビットがセットされていた場合に、引数 1 に指定したラベルに分岐します。ビットがセットされていない場合は、次の命令へ進みます。

用例

(省略) CHKBIT の用例を参照

10.25 JUMPZ

書式

JUMPZ [ラベル名]

引数

引数 1：CHKBIT 命令のビット確認結果、ビットがセットされていない場合に分岐するラベル名

説明

直前に発行された CHKBIT 命令のビット確認の結果、ビットがセットされていない場合に、引数 1 に指定したラベルに分岐します。ビットがセットされている場合は、次の命令へ進みます。

11 制御アプリケーション作成用ライブラリ MLGPLib 説明

- MLGPLib は、制御アプリケーション開発用の Windows スタティックライブラリです。
- 本ライブラリを使用して、外部 PC 制御用アプリケーションを自由に作成できます。
- 本ライブラリは Microsoft Visual Studio 2008/2010/2012 (Win32, 開発言語 C++) 専用です。その他の開発環境では使用できません。

11.1 ファイル構成

- ”MLGPLib.lib” MLGPLib ライブラリ本体(スタティックライブラリ)
 “MLGP.h” MLGPLib 用ヘッダファイル

11.2 API 一覧

API 名	引数	説明
MLGP_GetVersion	無し	本ライブラリのバージョンを取得します
MLGP_OpenHandle	無し	DTC2U3 のデバイスハンドルを取得します
MLGP_CloseHandle	デバイスハンドル	取得したデバイスハンドルをクローズします
MLGP_SetEvent	デバイスハンドル イベント番号	DTC2U3 にイベントを送信します。

DTC2U3 複数台接続対応用 API (ライブラリバージョン Ver.2.01 以降対応)

API 名	引数	説明
MLDU03_EnumMLDU03Devices	無し	現在接続中の DTC2U3 デバイスを検索します。
MLDU03_OpenHandle	デバイス番号	指定デバイス番号の DTC2U3 のデバイスハンドルを取得します。
MLDU03_GetIndexedDeviceInfo	デバイス番号 デバイス情報構造体ポインタ	指定デバイス番号の DTC2U3 デバイス情報を取得します。
MLDU03_GetDeviceInfo	デバイスハンドル デバイス情報構造体ポインタ	指定デバイスハンドルの DTC2U3 デバイス情報を取得します。

11.3 各 API 説明

MLGP_GetVersion

関数定義

```
USHORT MLGP_GetVersion(void);
```

説明

- ・本ライブラリのバージョンを取得します。

引数

無し

返り値

バージョン番号 (ex. Ver.1.00 → 0100h)

MLGP_OpenHandle

関数定義

```
MLDU03_HANDLE MLGP_OpenHandle (void);
```

説明

- ・PC に接続された DTC2U3 デバイスのデバイスハンドルを取得します。
- ・PC に複数台の DTC2U3 デバイスが同時に接続されている場合は、最初に見つかった DTC2U3 デバイスのみを制御対象とします。
- ・複数の DTC2U3 デバイスを接続して使用する場合は、後述の MLDU03_EnumMLDU03Devices API ならびに MLDU03_OpenHandle API を使用してください。

引数

無し

返り値

NULL : デバイス未接続
NULL 以外の値 : DTC2U3 デバイスハンドル

MLGP_CloseHandle

関数定義

```
void MLGP_CloseHandle (MLDU03_HANDLE hDevice);
```

説明

・MLGP_OpenHandle(または MLDU03_OpenHandle)で取得したデバイスハンドルをクローズします。

引数

hDevice : クローズする DTC2U3 デバイスハンドル

返回值

無し

MLGP_SetEvent

関数定義

```
BOOL MLGP_SetEvent (MLDU03_HANDLE hDevice, int nEventNum);
```

説明

- ・DTC2U3 に対して外部イベントを送信します。
外部イベントを送信すると、DTC2U3 の EVENT レジスタの該当ビットがセットされます。

引数

hDevice : GPIO 制御を行う MLDU03 デバイスハンドル
nEventNum : 送信するイベント番号 (0-15)

イベント番号 (nEventNum)

各イベント送信時の動作は下記の通りです。

イベント 番号	セットされる EVENT レジスタフラグ	説明
4	SW PUSH	DTC2U3 本体のプッシュボタンが押された時と同等の動作をします。
8	EXT0	DTC2U3 に外部イベント#0 を発生させます。
9	EXT1	DTC2U3 に外部イベント#1 を発生させます。
10	EXT2	DTC2U3 に外部イベント#2 を発生させます。
11	EXT3	DTC2U3 に外部イベント#3 を発生させます。

返り値

TRUE : イベント送信正常終了
FALSE : イベント送信異常終了

MLDU03_EnumMLDU03Devices

関数定義

```
int MLDU03_EnumMLDU03Devices(void);
```

説明

- ・現在接続中の DTC2U3 デバイスを検索し、接続デバイス数を取得します。
 - ・PC に複数の DTC2U3 デバイスが接続されている場合は、本 API をコールした後、後述の MLDU03_OpenHandle API を使用して、制御対象の DTC2U3 デバイスのハンドルを取得してください。
 - ・DTC2U3 デバイスの USB 接続状態が変化した場合は、再度本 API をコールした後、MLDU03_OpenHandle API を使用してデバイスハンドルを取得しなおしてください。
- ※本 API は、ライブラリバージョン Ver.2.01 以降のみ対応です。

引数

無し

返り値

現在接続中の DTC2U3 のデバイス数

MLDU03_OpenHandle

関数定義

```
MLDU03_HANDLE MLDU03_OpenHandle(int nDeviceIndex);
```

説明

- ・指定されたデバイス番号の DTC2U3 のデバイスハンドルを取得します。
 - ・取得したハンドルのクローズは、MLGP_CloseHandle API を使用して行ってください。
- ※本 API は、ライブラリバージョン Ver.2.01 以降のみ対応です。

引数

デバイスハンドルを取得する DTC2U3 のデバイス番号 (0 より開始されるデバイス番号)

※MLDU03_EnumMLDU03Devices API にて取得したデバイス数よりも小さい値を指定してください。

返り値

NULL : デバイス未接続
NULL 以外の値 : DTC2U3 デバイスハンドル

MLDU03_GetIndexedDeviceInfo**関数定義**

```

BOOL MLDU03_GetIndexedDeviceInfo(
    int nDeviceIndex, PMLDU03_DEVICE_INFO pDeviceInfo);

```

説明

- ・指定されたデバイス番号の DTC2U3 のデバイス情報を取得します。
- ※本 API は、ライブラリバージョン Ver.2.01 以降のみ対応です。

引数

nDeviceIndex : デバイス情報を取得する DTC2U3 のデバイス番号 (0 より開始されるデバイス番号)

pDeviceInfo : デバイス情報を格納する MLDU03_DEVICE_INFO 構造体のポインタ

返り値

TRUE : デバイス情報取得成功

FALSE : デバイス情報取得失敗

MLDU03_DEVICE_INFO 構造体定義

Offset	サイズ	名称	説明
+00h	2 bytes	wUSBVID	DTC2U3 の USB VID
+02h	2 bytes	wUSBPID	DTC2U3 の USB PID
+04h	2 bytes	wUSBBCD	DTC2U3 のファームウェアバージョン
+06h	32 bytes	szUSBSN	DTC2U3 の USB シリアル番号文字列 (ASCII)
+26h	4 bytes	nConnIndex	DTC2U3 の接続された USB Hub ポート番号
+2Ah	260 bytes	szHubName	DTC2U3 の接続された USB Hub の名称 (ASCII)

MLDU03_GetDeviceInfo

関数定義

```
BOOL MLDU03_GetDeviceInfo(  
    MLDU03_HANDLE hDevice, PMLDU03_DEVICE_INFO pDeviceInfo);
```

説明

- ・指定されたデバイスハンドルの DTC2U3 のデバイス情報を取得します。
- ※本 API は、ライブラリバージョン Ver.2.01 以降のみ対応です。

引数

hDevice : デバイス情報を取得する DTC2U3 のデバイスハンドル
pDeviceInfo : デバイス情報を格納する MLDU03_DEVICE_INFO 構造体のポインタ

返り値

TRUE : デバイス情報取得成功
FALSE : デバイス情報取得失敗

MLDU03_DEVICE_INFO 構造体定義

MLDU03_GetIndexedDeviceInfo の説明を参照してください。

11.4 MLGPLib ヘッダファイル

MLGPLib 用ヘッダファイル “MLGP.h” の内容は下記の通りです。

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Medialogic DTC2U3 制御ライブラリ MLGPLib Rev.2.01 ヘッダファイル
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

#include <windows.h>
#include <basetyps.h>

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// 各種定義
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

// MLDU03 デバイスハンドル
typedef PVOID          MLDU03_HANDLE;

// 挿抜装置 EVENT 番号定義
#define DTC2U3_EVENT_TIM0_EXP          0
#define DTC2U3_EVENT_TIM1_EXP          1
#define DTC2U3_EVENT_TIM2_EXP          2
#define DTC2U3_EVENT_TIM3_EXP          3
#define DTC2U3_EVENT_BUTTON            4
#define DTC2U3_EVENT_EXT_BASE          8
#define DTC2U3_EVENT_EXT0              (DTC2U3_EVENT_EXT_BASE+0)
#define DTC2U3_EVENT_EXT1              (DTC2U3_EVENT_EXT_BASE+1)
#define DTC2U3_EVENT_EXT2              (DTC2U3_EVENT_EXT_BASE+2)
#define DTC2U3_EVENT_EXT3              (DTC2U3_EVENT_EXT_BASE+3)
#define DTC2U3_EVENT_MAX                15

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// DTC2U3 ライブラリ関数
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
extern USHORT MLGP_GetVersion(void);
extern MLDU03_HANDLE MLGP_OpenHandle(void);
extern void MLGP_CloseHandle(MLDU03_HANDLE hMLDU03Device);
extern BOOL MLGP_SetEvent(MLDU03_HANDLE hMLDU03Device, int nEventNum);

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// 複数デバイス対応用ライブラリ関数
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

// DTC2U3 デバイス情報構造体
#pragma pack(1)
typedef struct _m1du03_dev_info{

    USHORT    wUSBVID;                // DTC2U3 の USB VID
    USHORT    wUSBPID;                // DTC2U3 の USB PID
    USHORT    wUSBBCD;                // DTC2U3 の FW バージョン
    char      szUSBSN[32];            // DTC2U3 の USB シリアル番号
    int       nConnIndex;             // デバイスの接続されたハブポート番号
    char      szHubName[MAX_PATH];    // デバイスの接続されたハブ名称

} MLDU03_DEVICE_INFO, *PMLDU03_DEVICE_INFO;
#pragma pack()

extern int MLDU03_EnumMLDU03Devices(void);
extern MLDU03_HANDLE MLDU03_OpenHandle(int nDeviceIndex);
extern BOOL MLDU03_GetIndexedDeviceInfo(
    int nDeviceIndex, PMLDU03_DEVICE_INFO pDeviceInfo);
extern BOOL MLDU03_GetDeviceInfo(
    MLDU03_HANDLE hMLDU03Device, PMLDU03_DEVICE_INFO pDeviceInfo);

```


11.5 MLGPLib アプリケーションサンプル

MLGPLib を使用したアプリケーションの例です。DTC2U3 本体のプッシュボタンが押された状態のエミュレーションを、外部 PC から行います。

```
#include "stdafx.h"
#pragma comment(lib, "mlgplib.lib")
#include "mlgp.h"

int _tmain(int argc, _TCHAR* argv[])
{
    MLDU03_HANDLE h;
    int nResult = -1;

    // MLDU03 デバイスハンドル取得
    h = MLGP_OpenHandle();
    if(h == NULL){
        return nResult;
    }

    // プッシュボタンイベント送信
    printf("¥n Send Button Event to DTC2U3.¥n");
    if(MLGP_SetEvent(h, DTC2U3_EVENT_BUTTON)){
        printf("    -> Success.¥n");
        nResult = 0;
    }else{
        printf("    -> Failed.¥n");
        nResult = -2;
    }

    // MLDU03 デバイスハンドルクローズ
    MLGP_CloseHandle(h);

    printf("¥ncompleted!¥n");
    return nResult;
}
```

12 安全上の注意

- ・ご使用前に、安全上の注意をよくお読みの上、正しくご使用ください。
- ・この項に記載している警告及び注意には、使用者や第三者への肉体的危害や財産への損害を未然に防ぐ内容を含んでおりますので、必ずご理解の上、守って頂くようお願いいたします。



警告及び注意

- ★電氣的許容範囲を守ってください。
機器指定の電圧、電流許容範囲を必ず守ってください。
定格を越えた範囲での使用は、火災や感電、故障の原因となります。
- ★コネクタの向きを確認して機器を接続してください。
コネクタの向きを間違えての無理な挿入などは、故障や火災の原因となります。
- ★煙が出る、異臭がする。
万一、煙が出る、異臭がするときはすぐに機器の電源を切断し、弊社サポートまでご送付ください。
- ★機器の分解、改造をしない。
機器の分解、改造をすることは、火災や感電の原因となります。修理は弊社サポートまでご送付ください。
- ★機器の内部に異物や水を入れない。
筐体のすきまから内部に異物や水が入った場合は、すぐに機器の電源を切断し、弊社サポートまでご送付ください。
- ★不安定な場所で機器を使用しないでください。
ぐらついた台の上や傾いた場所、大きな振動が発生する場所で機器を使用しないでください。
落ちたり、倒れたりしてけがの原因になることがあります。そのまま使用されると故障の原因となる可能性があります。
- ★取り扱いについて
機器接続ケーブル、機器本体に重い物を置いたり、引っ張ったり、無理にねじったり、加熱したりすると、火災や感電、故障の原因となります。
- ★雷が鳴り出したら機器接続ケーブル、金属部分に触れないでください。感電や火災の原因となります。
- ★ぬれたままの手で機器に触れないでください。感電や故障の原因となります。
- ★高温や湿気、水気のある場所では使用しないでください。
直射日光のあたる場所、暖房器具の近くなど高温になる場所、5℃未満の低温になる場所、油煙や湯気のあたる調理台、ほこりや湿気の多い場所で使用しないでください。誤動作や故障、火災の原因となります。
- ★足を引っかける恐れのある所で使用しないでください。
- ★長期間使用しない場合は、機器接続ケーブルを外し保管してください。
- ★機器を移動するときは機器接続ケーブル類をすべて外してください。
接続したままの移動は、コネクタの変形や故障の原因となります。
- ★小さいお子様を近づけないでください。
小さいお子様が誤って食べたり、飲み込むことがないように、手の届かない所で使用、保管してください。
- ★殺虫剤を使って害虫駆除する際には、機器の電源を切断し、ビニールなどで保護してください。故障の原因となります。
- ★本製品は精密電子機器ですので静電気を与えると、誤動作や故障の原因となる可能性があります。

13 その他事項

- ★本製品を使用するによって生じた直接、間接の損害について、弊社は一切の責任を負いません。
- ★本製品は、医療機器、原子力機器、航空宇宙機器、など人命に関わる設備や機器、設備、機器での使用は意図されておりません。このような環境下での使用に関しては、一切の責任を負いません。
- ★本書の内容等に関しましては、将来予告なしに変更することがあります。

販売・サポート 株式会社 メディアロジック

住所: 〒939-1119

富山県高岡市オフィスパーク 5 番 富山県産業高度化センター内 O-C

E-Mail: support@medialogic.co.jp

14 改版履歴

版数	発行日	内容
2.01	2015/09/15	初版作成
2.02	2015/12/21	制御ライブラリの DTC2U3 複数台接続対応
2.03	2016/01/04	コマンドライン制御アプリケーションのデバイス指定オプション対応 (/SN=, /LIST)
2.04	2017/01/17	外形寸法及び重量の記載追加 電氣的仕様の記載追加 環境仕様の記載追加 安全上の注意記載追加 その他の事項記載追加 同梱物よりソフトウェア CD の記述を削除。ソフトウェアダウンロード先 URL の記述を追加。