



CUnet (MKY43) 搭載 PCI Express ボード

CUB-43PCIEXP

ユーザーズマニュアル

ご注意

1. 本書に記載された内容は、将来予告なしに変更する場合があります。本製品をご使用になる際には、本書が最新の版であるかをご確認ください。
2. 本書において記載されている説明や回路例などの技術情報は、お客様が用途に応じて本製品を適切にご利用をいただくための参考資料です。実際に本製品をご使用になる際には、基板上における本製品の周辺回路条件や環境を考慮の上、お客様の責任においてシステム全体を十分に評価し、お客様の目的に適合するようシステムを設計してください。当社は、お客様のシステムと本製品との適合可否に対する責任を負いません。
3. 本書に記載された情報、製品および回路等の使用に起因する損害または特許権その他権利の侵害に関して、当社は一切その責任を負いません。
4. 本製品および本書の情報や回路などをご使用になる際、当社は第三者の工業所有権、知的所有権およびその他権利に対する保証または実施権を許諾致しません。
5. 本製品は、人命に関わる装置用としては開発されておりません。人命に関わる用途への採用をご検討の際は、当社までご相談ください。
6. 本書の一部または全部を、当社に無断で転載および複製することを禁じます。

はじめに

本マニュアルは、CUnet 専用 IC である MKY43 を搭載した PCI Express ボードの CUB-43PCIEXP について記述します。

CUB-43PCIEXP の利用および本マニュアルの理解に先駆けて、“CUnet 導入ガイド”を必ずお読みください。

●対象読者

- ・ CUnet を初めて構築する方
- ・ CUnet を構築するために、弊社の CUB-43PCIEXP を初めてご利用になる方

●読者が必要とする知識

- ・ ネットワーク技術に関する標準的な知識
- ・ 半導体製品（特にマイクロコントローラおよびメモリ）に関する標準的な知識

●関連マニュアル

- ・ CUnet 導入ガイド
- ・ CUnet テクニカルガイド
- ・ MKY43 ユーザーズマニュアル

【注意事項】

本書において記載されている一部の用語は、弊社の Web および営業用ツール（総合カタログ等）において記載されている用語とは異なっています。営業用ツールにおいては、様々な業界において弊社製品をご理解いただけるよう、一般的用語を用いています。

CUnet ファミリに関する専門知識は、技術ドキュメント（マニュアル等）を基にご理解ください。

改訂履歴

Ver	日付	改訂内容	
		ページ	説明
Ver1.0J	2015年9月	--	初版発行
Ver2.0J	2017年3月	--	Windows8/8.1 に対応
		3-2	「3.4 制限事項」を追加
		3-4	API 関数に MKY36 からの割り込み発生回数、割り込み要因情報を取得・クリア関数を追加
Ver3.0J	2018年11月	--	Windows10 に対応
		1-1	「表 1-1 ボード仕様」の構成を変更
Ver4.0J	2022年3月	2-1	Rev.A2 基板と Rev.A3 基板の見分け方
		2-2	図 2-2 パネル外観 パネル説明 Rev.A3 の場合の説明追記
		2-3	図 2-3 コネクタ周辺回路図 Rev.A3 の図を追記
		2-4	図 2-4 CUB-43PCIEXP ボードの設定 Rev.A3 の図を追記
		2-7	図 2-5 寸法図 Rev.A3 の図に修正
		3-3	3.4.3 割り込み処理 割り込み処理内容について説明文訂正
		3-4	3.4.3 割表 3-1 API 関数 下記関数を削除 CubGetInterrupt0Count / CubGetInterrupt1Count CubClearInterrupt0Count / CubClearInterrupt1Count CubGetInterrupt0StatusInfo / CubGetInterrupt1StatusInfo CubClearInterrupt0StatusInfo / CubClearInterrupt1StatusInfo
		3-12	下記章を削除 3.5.13 CubGetInterrupt0Count、CubGetInterrupt1Count 3.5.14 CubClearInterrupt0Count、CubClearInterrupt1Count 3.5.15 CubGetInterrupt0StatusInfo、CubGetInterrupt1StatusInfo 3.5.16 CubClearInterrupt0StatusInfo、CubClearInterrupt1StatusInfo
3-14	3.6.2 割り込み処理サンプルを削除		

目次

第1章 製品概要

1.1 特徴	1-1
1.2 仕様	1-1

第2章 ハードウェア

2.1 コネクタ仕様	2-2
2.2 ディップスイッチ	2-4
2.2.1 ボード ID スイッチ (SW5)	2-4
2.2.2 ターミネーション設定スイッチ (SW4)	2-4
2.2.3 メーカー設定スイッチ (SW6)	2-4
2.3 メモリマップ	2-5
2.3.1 MKY43	2-5
2.3.2 CUB-43PCIEXP 独自のレジスタ	2-6
2.4 寸法図	2-7

第3章 ソフトウェア

3.1 概要	3-1
3.2 著作権・免責	3-1
3.3 ファイル構成	3-2
3.4 制限事項	3-2
3.4.1 マルチスレッド	3-2
3.4.2 省電力モードについて	3-2
3.4.3 割込み処理	3-3
3.4.4 ドライバを使用しない場合のアクセス方法	3-3
3.5 API 仕様	3-4
3.5.1 CubGetVersion	3-5
3.5.2 CubGetLastError	3-6
3.5.3 CubCountDevice	3-6
3.5.4 CubBoardID	3-7
3.5.5 CubResetBoard	3-7
3.5.6 CubSearchBoard	3-8
3.5.7 CubOpenHandle	3-9
3.5.8 CubCloseHandle	3-10
3.5.9 CubReadByte	3-10
3.5.10 CubWriteByte	3-11
3.5.11 CubReadWord	3-11
3.5.12 CubWriteWord	3-12
3.6 サンプルプログラム	3-13

図 目 次

図 2-1	基板リビジョン表示.....	2-1
図 2-2	パネル外観	2-2
図 2-3	コネクタ周辺回路.....	2-3
図 2-4	CUB-43PCIEXP ボードの設定.....	2-4
図 2-5	寸法図	2-7

表 目 次

表 1-1	仕様.....	1-1
表 2-1	メモリマップ.....	2-5
表 3-1	API 関数.....	3-4
表 3-2	バージョン番号の構成	3-5
表 3-3	エラーコードリスト.....	3-6

第 1 章 製品概要

本章は、CUB-43PCIEXP の製品概要について記述します。

1.1 特徴

CUB-43PCIEXP は、MKY43 を搭載した PCI Express 拡張バス対応の CUnet 通信ボードです。ステップテクニカ提供の Windows 用の API と併せて利用することにより、MKY43 の機能を簡単に利用することが可能です。

1.2 仕様

CUB-43PCIEXP の仕様を、表 1-1 に示します。

表 1-1 仕様

CUnet デバイス	MKY43 1 個
CUnet 通信方式	半二重通信
CUnet 通信速度	12M/6M/3Mbps (MKY43 レジスタにて設定)
CUnet 通信コネクタ	RJ45 タイプ (8pin モジュラー) × 2 個
対応バス	PCI Express x1 Gen1 準拠した拡張バス
占有リソース	16KB の連続したメモリエリア (PnP にて自動割当)
割込み	1 ライン使用 (PnP にて自動割当)
対応 OS	Windows10 (64bit/32bit) Windows8.1 (64bit/32bit) Windows8 (64bit/32bit) Windows7 (64bit/32bit)
電源	DC +3.3V
消費電流	500mA 以下
動作環境	温度 0 ~ 50℃ 湿度 20 ~ 90% (非結露)
保存環境	温度 0 ~ 80℃ 湿度 0 ~ 90% (非結露)
外形寸法	119.9mm(W) × 68.9mm(H) ※パネル部含まず (Low Profile 対応)
付属品	Low Profile 用ブラケット

第2章 ハードウェア

本書は、CUB-43PCUEXP Rev.A2 と Rev.A3 のハードウェアについて記述します。

Rev.A2 と Rev.A3 は、図 2-1 にとおり製品のシルク表示により見分けることができます。

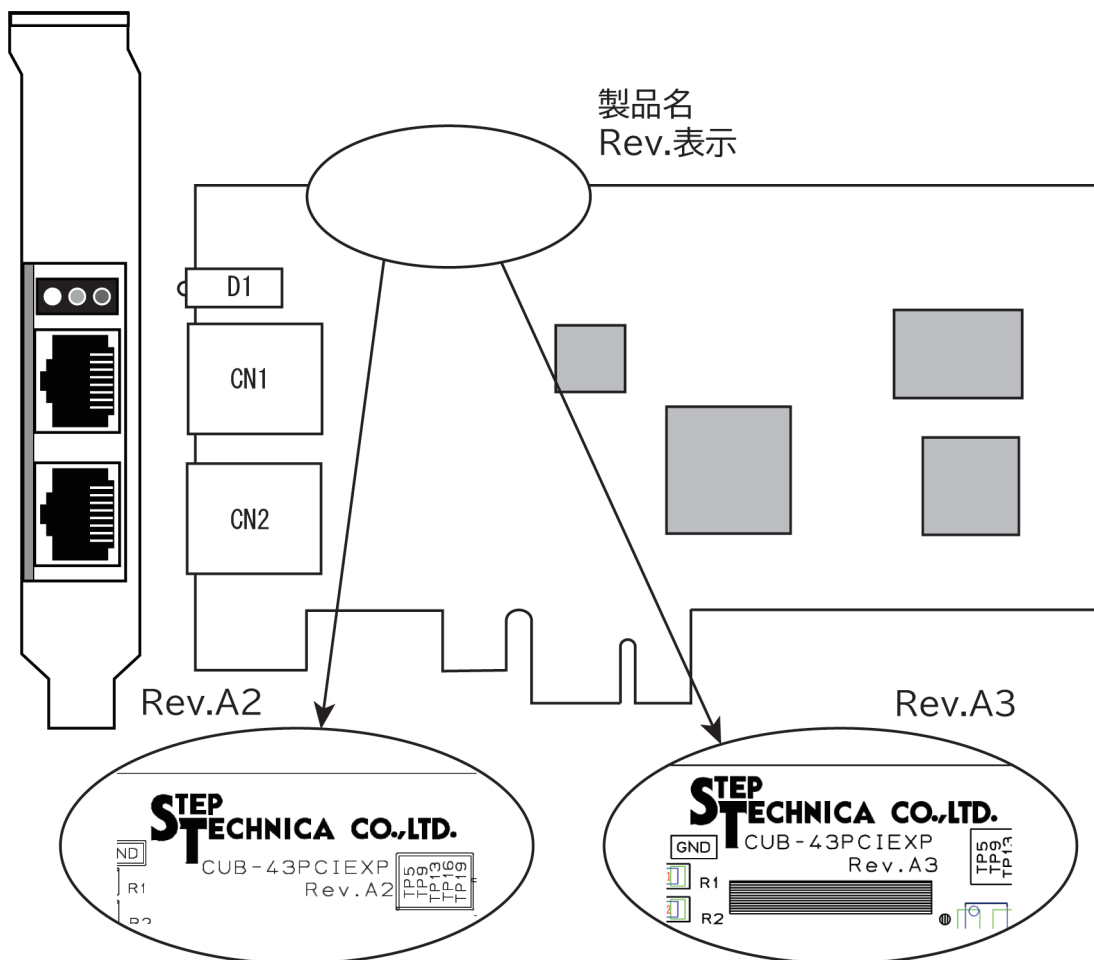


図 2-1 基板リビジョン表示

2.1 コネクタ仕様

CUB-43PCIEXP のパネル面とその詳細を図 2-2 に示します。

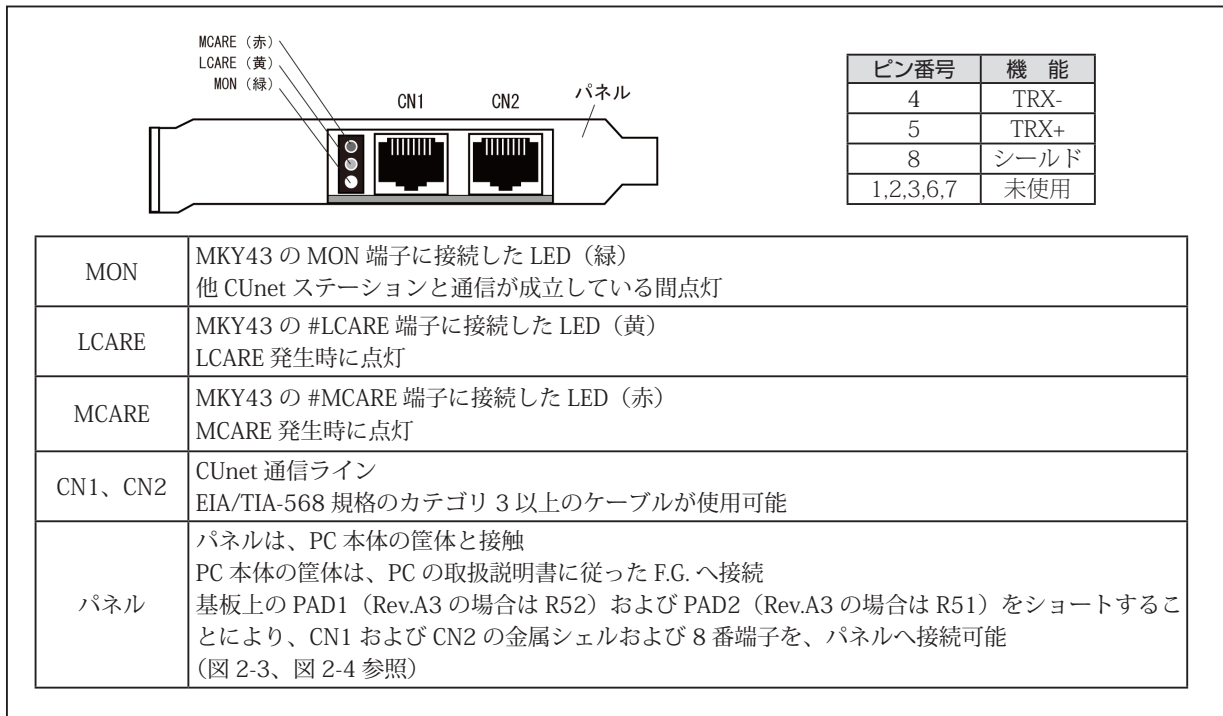
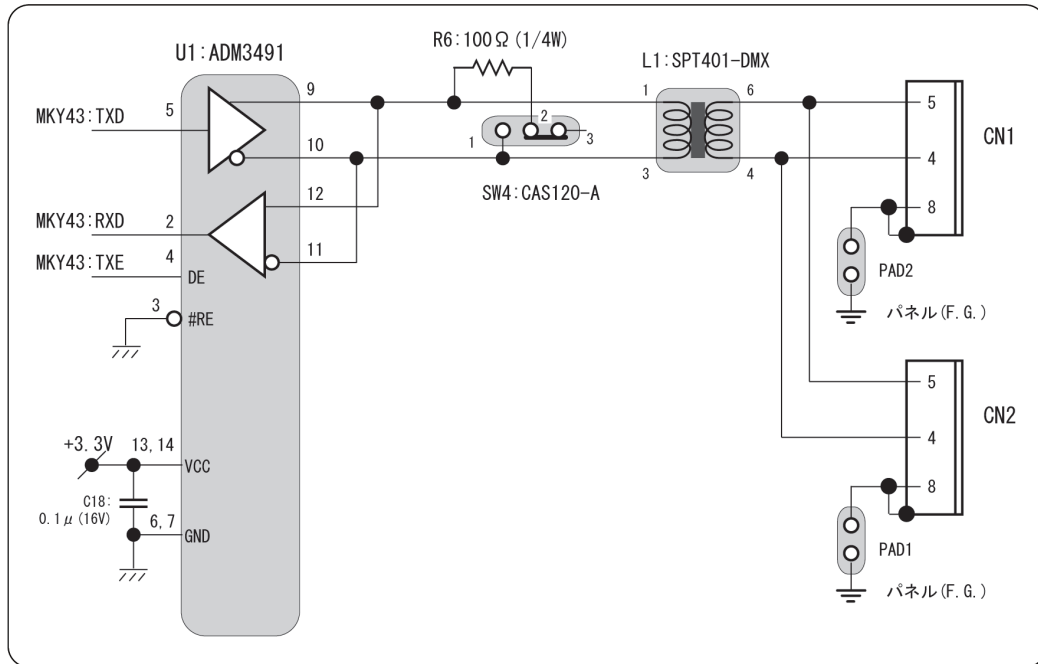


図 2-2 パネル外観

CN1、CN2 コネクタ周辺回路を図 2-3 に示します。

Rev.A2 基板



Rev.A3 基板

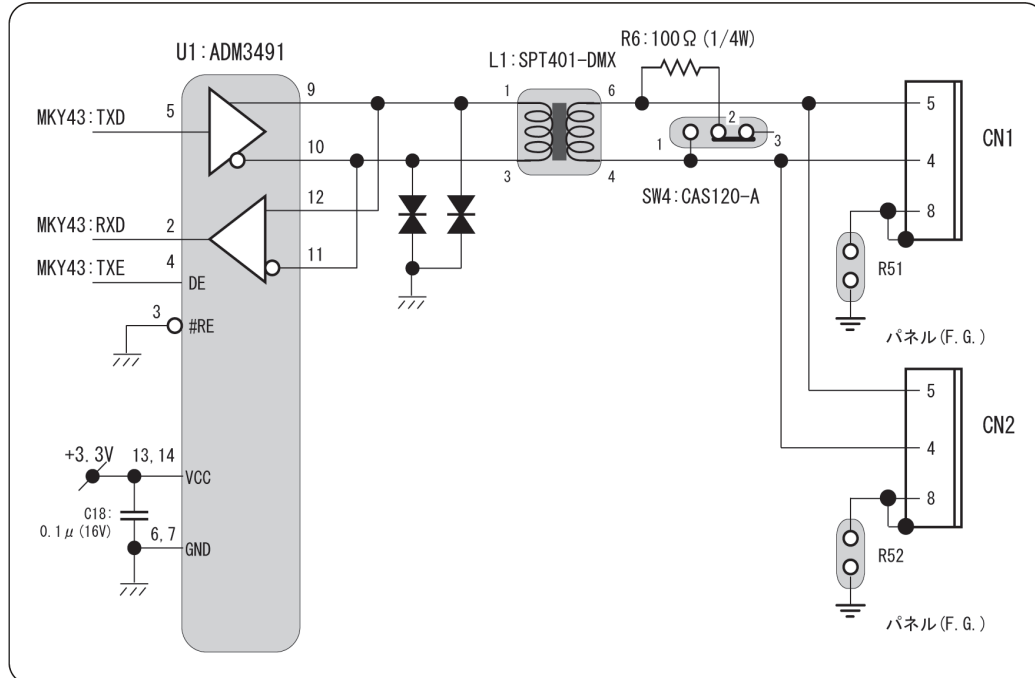
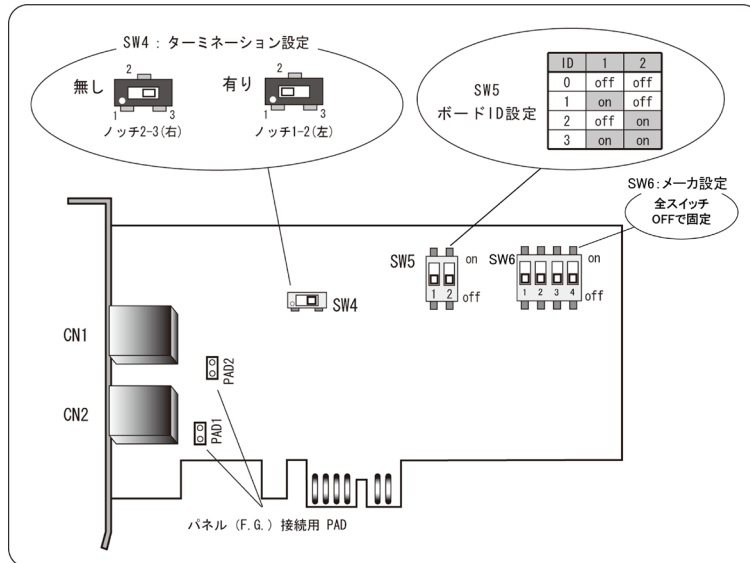


図 2-3 コネクタ周辺回路

2.2 ディップスイッチ

CUB-43PCIEXP のディップスイッチの設定を図 2-4 に示します。

Rev.A2 基板



Rev.A3 基板

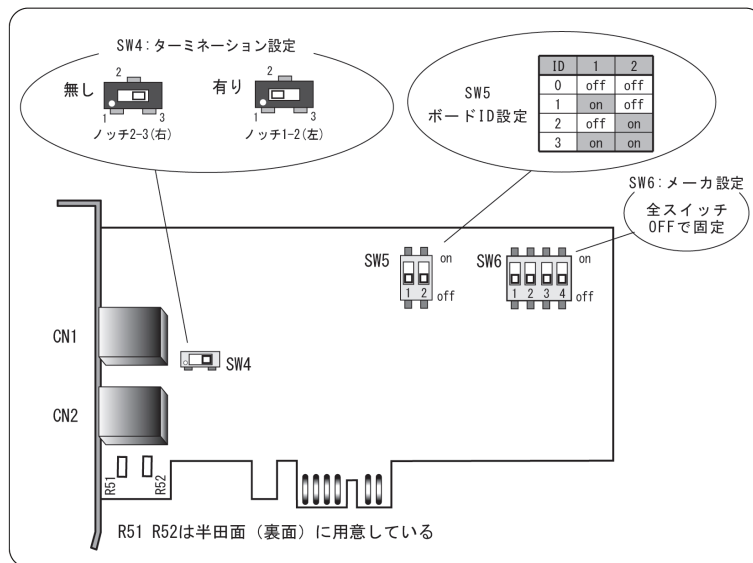


図 2-4 CUB-43PCIEXP ボードの設定

2.2.1 ボード ID スイッチ (SW5)

複数 CUB-43PCIEXP を同一の機器に搭載する場合には、SW5 のボード ID を各ボード個別の番号に設定してください。
(工場出荷時設定 ボード ID : 0)

2.2.2 ターミネーション設定スイッチ (SW4)

CUB-43PCIEXP が CUnet 通信ラインの末端になる場合は、ターミネーションを ON (有効) にしてください。
ターミネーションを ON に設定すると通信経路が 100 Ω で終端されます。
それ以外の場合は、OFF (無効) にしてください。OFF に設定すると終端されません。
(工場出荷時設定 ターミネーション : 無効)

2.2.3 メーカー設定スイッチ (SW6)

メーカー用設定スイッチです。すべて OFF の状態で使用してください。

2.3 メモリマップ

CUB-43PCIEXP のメモリマップを表 2-1 に示します。

メモリマップ中のアドレスは CUB-43PCIEXP の先頭アドレスからの相対値であり、実際のアドレスは PCI BIOS から自動的に割り振られたボードの先頭アドレス値を加算したアドレスになります。

表 2-1 メモリマップ

アドレス	概要
000H ~ 5FFH	MKY43
600H ~ BFFH	未使用
C00H	Chip Reset Register
C02H ~ DFFH	未使用
E00H	Board ID Register
E02H ~ FFFH	未使用

2.3.1 MKY43

MKY43 のメモリマップについては「MKY43 ユーザーズマニュアル」の「第 4 章 MKY43 のソフトウェア」、
「4.1.1 メモリマップ」をご参照ください。

2.3.2 CUB-43PCIEXP 独自のレジスタ

表 2-1 のメモリマップに示された C00H および E00H のレジスタは、CUB-43PCIEXP 独自のレジスタです。以下に、そのレジスタの詳細を記載します。

Chip Reset Register アドレス : C00H

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W
機能	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	CRST

[機能説明] CRST (Chip ReSeT) へ“1”をライトすることにより、MKY43 の RST 端子へリセット信号を印加します。RST 端子へのリセット期間は、280ns です。また、本レジスタは書き込み専用レジスタの為、読み込みを行った場合のデータは不定です。

Board ID Register アドレス : E00H

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
機能	--	--	--	--	--	--	--	--	--	--	--	--	--	--	BID1	BID0

[機能説明] BID0,1 (Board ID) ビットをリードすることにより、SW5 によって設定されたボード ID の値を取得することができます。詳細については、“2.2 ディップスイッチ”を参照ください。



表 2-1 のメモリマップに示されている未使用領域 (“600H ~ BFFH、C02H ~ DFFH、E02H ~ FFFH”) はアクセスしないでください。システムを不安定にする可能性があります。

2.4 寸法図

寸法図を図 2-5 に示します。外形寸法は Rev.A2、Rev.A3 に相違はありません。(図は、Rev.A3 です。)

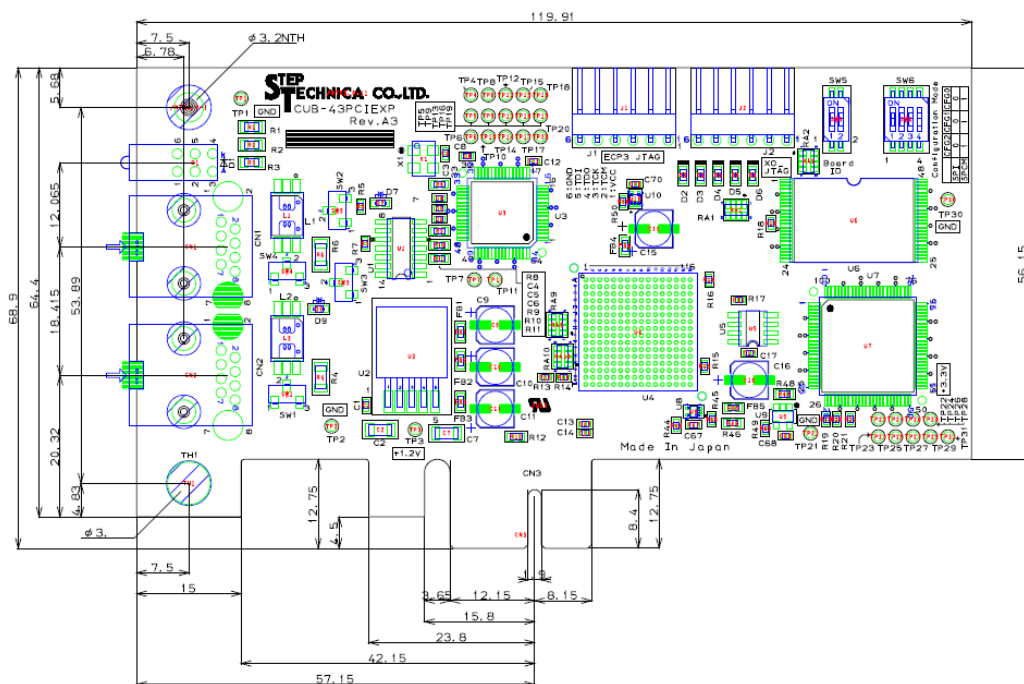


図 2-5 寸法図

第3章 ソフトウェア

本章は、ステップテクニカ社提供の API について記述します。

本マニュアルは、API バージョンのメジャー番号「2」以上に基づいて記述しています。
製品のご使用にあたっては、弊社ホームページにより最新の情報をご確認してください。

3.1 概要

Windows 上のユーザアプリケーションからの CUB-43PCIEXP へのアクセスするための API を用意しています。
下記ステップテクニカ社のダウンロードページより API をダウンロードできます。

URL : <https://www.steptecnica.com/jp/download/index.html>

対応 OS は

- Windows10 (64bit/32bit)
- Windows8.1 (64bit/32bit)
- Windows 8 (64bit/32bit)
- Windows 7 (64bit/32bit)

です。

提供している API は、Microsoft Visual Studio などから呼び出すことが可能です。

3.2 著作権・免責

全てのドキュメント・プログラム・プログラムソースの著作権は、株式会社ステップテクニカが所有しています。
株式会社ステップテクニカは、以下の注意事項を了承された個人・法人、その他の団体が CUB-43PCIEXP を利用する場合に限り、これら著作物の複製・利用をする権利をライセンスするものです。株式会社ステップテクニカに断り無く、CUB-43PCIEXP 以外でこれら著作物の一部または全部を改訂・再配布や複製・利用することはできません。



注意事項

- ① 弊社 web ページより入手した全てのソフトウェアの使用による、いかなる結果に対しても弊社は一切責任を負いません。
- ② API は、説明に従って正しくお使いください。
- ③ 仕様・内容は、将来予告無く変更になる場合があります。弊社は、将来への互換性について、一切保証致しません。
- ④ OS や開発環境等に関するお問い合わせはサポート致しかねます。
- ⑤ バグ・不具合などを発見された方は、弊社システム開発部までご連絡ください。

3.3 ファイル構成

“DLL” フォルダに収められたファイルは以下のとおりです。

【cub43pciexp.dll】

DLL 本体です。Windows のシステムフォルダか、本 DLL を使用するユーザプログラムと同じフォルダにコピーしてお使いください。

【cub43pciexp.lib】

Microsoft Visual C/C++ 用のインポートライブラリです。

【cub43pciexp.h】

API のヘッダファイルです。ご使用の際は、Windows.h より後ろにインクルードしてください。

3.4 制限事項

ここでは、本 API を使用してアプリケーションを作成する際の制限事項について記します。

3.4.1 マルチスレッド

DLL 内の API は複数スレッドから同時に使用することはできません。

アプリケーションをマルチスレッド構成にする場合、同時呼び出しが起らないように配慮して下さい。

3.4.2 省電力モードについて

CUB-43PCIEXP は、省電力モード機能に対応しておりません。

OS のスリープ機能を停止したうえでご使用ください。スリープに入った場合には、搭載されている MKY43 への電源供給が遮断され、通信が停止します。また、省電力モードからの復帰時にはリセットがかかる為、各レジスタは初期化され、グローバルメモリ、メール送信バッファ、メール受信バッファ 0、1 領域は不定の状態になりますのでご注意ください。

3.4.3 割込み処理

MKY43 では、割込みを発生させることが可能であり、INTOCR または INT1CR により、各種割込みの要因の有効・無効を設定することができ、INTOSR または INT1SR から発生した割込み要因を確認することができます。

CUB-43PCIEXP では、MKY43 からの割込みをドライバ側で認識することができる仕様になっています。

しかしながら、下記の理由により弊社から提供しているドライバ及び API を用いて MKY43 の割込み処理を作成できない仕様となっております。ユーザアプリケーションにおいて、MKY43 の割込み機能をご使用される場合は、別途ドライバを用意していただく必要がございます。

- ドライバ内の割込み処理にて割込み発生要因を示す、INTOSR/INT1SR を自動的にクリアしているため、ユーザアプリケーション側から INTOSR/INT1SR を読み出しても正しい情報を取得できない
- ドライバ内の割込み処理は、INTOSR/INT1SR のクリアだけを行っており、コールバック関数の登録するなど、ユーザアプリケーション側に処理を渡す機能がない

3.4.4 ドライバを使用しない場合のアクセス方法

ステップテクニカ社製ドライバを使用せずに CUB-43PCIEXP へ直接アクセスする場合には、以下の点について注意が必要です。

CUB-43PCIEXP には、常に 32bit アクセスを行ってください。その時、下位 16bit データが有効となり、上位 16bit は使用されません。その為、アクセスするアドレスは上記メモリマップの 2 倍を指定する必要があります。

例えば、MKY43 のアドレス 200H を Read する場合、CUB-43PCIEXP の 400H を 32bit Read することで、その下位 16bit に MKY43 の 200H の 2 バイトデータが取得できます。CUB-43PCIEXP 独自レジスタに関しても同様のアクセスが必要です。

3.5 API 仕様

表 3-1 にサポートしている API 関数の一覧を示します。

以降に説明する API 関数は、cub43pciexp.h に収録されている関数です。

表 3-1 API 関数

関 数	機能概要
CubGetVersion	API のバージョン番号を取得
CubGetLastError	API 関数の終了状態を取得
CubCountDevice	CUB-43PCIEXP ボードの枚数を取得
CubBoardID	ボード ID を取得
CubResetBoard	MKY43 のリセットを指示
CubSearchBoard	CUB-43PCIEXP ボードの枚数とそのボード ID を取得
CubOpenHandle	CUB-43PCIEXP のハンドル値を取得
CubCloseHandle	CubOpenHandle で取得したハンドルをクローズ
CubReadByte	指定したアドレスから 1 バイトのデータ読み込み
CubWriteByte	指定したアドレスへ 1 バイトのデータ書き込み
CubReadWord	指定したアドレスから 2 バイトのデータ読み込み
CubWriteWord	指定したアドレスへ 2 バイトのデータ書き込み

3.5.1 CubGetVersion

書式

UINT CubGetVersion(void);

機能

API のバージョン番号を取得します。

パラメータ

なし

戻り値API のバージョン番号 (BCD コード 16 進数)
(メジャー番号 + マイナー番号 + アップデート番号)**エラーコード**

本関数実行後に CubGetLastError が返すエラーコードとエラー発生要因は以下の通りです。

CUB_SUCCESS 正常終了

注釈表 3-2 に、バージョン番号の構成を示します。
バージョン番号が変更される原因は、下記のとおりです。

メジャー番号 : API の仕様変更など、互換性を保てなかった変更がなされた時に変わります。

マイナー番号 : API 関数の追加など、互換性を保ったまま変更がなされた時に変わります。

アップデート番号 : 不具合修正など、仕様に影響が無く修正された時に変わります。

表 3-2 バージョン番号の構成

戻り値 (例)	メジャー番号 (ビット 15 ~ 8)	マイナー番号 (ビット 7 ~ 4)	アップデート番号 (ビット 3 ~ 0)
0x0102	1	0	2
0x1398	13	9	8

3.5.2 CubGetLastError

書式

UINT CubGetLastError(void);

機能

最後に呼び出された API 関数の終了状態を取得します。

パラメータ

なし

戻り値

cub43pciexp.h で定義しているエラーコードを返します。

注釈

表 3-3 に cub43pciexp.h で定義しているエラーコードを記します。

表 3-3 エラーコードリスト

文字定数	値	内容
CUB_SUCCESS	0	正常終了
CUB_ERR_DEVICENOTEXIST	1	デバイスが存在しない
CUB_ERR_ALREADYOPENED	2	すでにオープンされている
CUB_ERR_CLOSED	3	CubOpenHandle が一度もコールされていない
CUB_ERR_INVALIDPARAM	4	ハンドルが無効
CUB_ERR_NORESOUCE	5	実行に必要なリソースが足りない
CUB_ERR_FAILED	6	原因不明により処理が実行できなかった
CUB_NOTCALLYET	99	API 関数が一度もコールされていない

3.5.3 CubCountDevice

書式

INT CubCountDevice(void);

機能

パソコンに装着している CUB-43PCIEXP ボードの枚数を返します。

パラメータ

なし

戻り値

CUB-43PCIEXP ボードの枚数を返します。

-1 : 5 枚以上
 0 : 1 枚も装着されていない
 1 ~ 4 : 1 枚 ~ 4 枚

エラーコード

本関数実行後に CubGetLastError が返すエラーコードとエラー発生要因は以下の通りです。

CUB_SUCCESS 正常終了

注釈

パソコンに 5 枚以上装着することはできません。

3.5.4 CubBoardID

書式

INT CubBoardID(HANDLE CUBHandle);

機能

CUB-43PCIEXP ボードのボード ID を取得します。

パラメータ

HANDLE CUBHandle CUB-43PCIEXP のハンドル

戻り値

正常終了時はボード ID (0 ~ 3) を返します。失敗時は -1 を返します。

エラーコード

本関数実行後に CubGetLastError が返すエラーコードとエラー発生要因は以下の通りです。

CUB_SUCCESS	正常終了
CUB_ERR_INVALIDPARAM	ハンドルが無効
CUB_ERR_FAILED	原因不明により処理が実行できなかった

3.5.5 CubResetBoard

書式

BOOL CubResetBoard (HANDLE CUBHandle);

機能

指定された CUB-43PCIEXP ボードの MKY43 をリセットします。

パラメータ

HANDLE CUBHandle CUB-43PCIEXP のハンドル

戻り値

正常終了時は TRUE、失敗時は FALSE を返します。

エラーコード

本関数実行後に CubGetLastError が返すエラーコードとエラー発生要因は以下の通りです。

CUB_SUCCESS	正常終了
CUB_ERR_INVALIDPARAM	ハンドルが無効
CUB_ERR_FAILED	原因不明により処理が実行できなかった

3.5.6 CubSearchBoard

書式

```
BOOL CubSearchBoard(BYTE *board_num , BYTE *board_id_list);
```

機能

パソコンに装着されている CUB-43PCIEXP ボードの枚数とそのボード ID リストを返します。

パラメータ

*board_num	ボード枚数がセットされるバイト型変数へのアドレスを指定します。 セットされた値の意味は以下の通りです。 <ul style="list-style-type: none">• -1 : 5 枚以上確認された• 0 : 1 枚もない• 1 ~ 4 : 認識したボード枚数
*board_id_list	ボード ID を受け取る為に、バイト型を 4 要素持つ配列のアドレスを指定します。または NULL を指定することも可能です。 NULL が指定された場合は、ボード枚数のみを返します。 セットされた値の意味は以下の通りです。 <ul style="list-style-type: none">• 0x00 ~ 0x03 : ボード ID• 0xFF : 認識できなかった

戻り値

正常終了時は TRUE、失敗時は FALSE を返します。

エラーコード

本関数実行後に CubGetLastError が返すエラーコードとエラー発生要因は以下の通りです。

CUB_SUCCESS	正常終了
CUB_ERR_INVALIDPARAM	*board_num に NULL が指定された
CUB_ERR_FAILED	原因不明により処理が実行できなかった

注釈

ボード ID は、SW5 により設定します。
CUB-43PCIEXP ボードが複数枚装着している場合は、ボード ID により識別することができます。
本 API 関数では、最大 4 枚までの CUB-43PCIEXP ボードを識別できます。
下記のようにバイト型配列をパラメータとして指定してください。

```
BYTE board_num;  
BYTE board_id_list[4];  
CubSearchBoard(&board_num, &board_id_list[0]);
```

動作例として、パソコンに 3 枚の CUB-43PCIEXP ボードが装着されており、それぞれのボード ID が 1 枚目：ボード ID=0、2 枚目：ボード ID=1、3 枚目：ボード ID=2 と設定されています。
パソコンが認識した順番が 1 枚目、3 枚目、2 枚目となっている状態で CubSearchBoard が実行された場合

```
board_num = 3;  
board_id_list[0] = 0、board_id_list[1] = 2、board_id_list[2] = 1、board_id_list[3] = 0xFF
```

と返します。

3.5.7 CubOpenHandle

書式

```
HANDLE CubOpenHandle(int index_no);
```

機能

CUB-43PCIEXP のハンドルを取得します。

パラメータ

int index_no	インデックス番号 インデックス番号には、0～3が指定できます。 CUB-43PCIEXP ボードが1枚の場合は、0をセットしてください。 詳しくは、“注釈”を参照してください。
--------------	---

戻り値

正常終了時は、1以上の値を返します。失敗時は -1 (INVALID_HANDLE_VALUE) を返します。

エラーコード

本関数実行後に CubGetLastError が返すエラーコードとエラー発生要因は以下の通りです。

CUB_SUCCESS	正常終了
CUB_ERR_DEVICENOTEXIST	デバイスが存在しない
CUB_ERR_FAILED	原因不明により処理が実行できなかった

注釈

プログラム終了時には、CubCloseHandle によりハンドルをクローズしてください。

CUB-43PCIEXP ボードが1枚の場合は、CubSearchBoard を実行する必要がありません。
CUB-43PCIEXP ボードが複数枚装着している場合は、CubSearchBoard を先に実行し、操作を行う対象の CUB-43PCIEXP を確認しておく必要があります。

例として、パソコンに3枚の CUB-43PCIEXP ボードが装着されており、それぞれのボード ID が
1枚目：ボード ID=0、2枚目：ボード ID=1、3枚目：ボード ID=2
と設定されています。ここでボード ID=2 のハンドルを取得する為には

```
BYTE board_num;  
BYTE board_id_list[4];  
CubSearchBoard(&board_num, &board_id_list[0]);
```

を実行した結果、

board_id_list [0] = 0、board_id_list [1] = 2、board_id_list [2] = 1、board_id_list [3] = 0xFF
になったと仮定します。

この場合、インデックス番号 1 がボード ID=2 であることが確認できます。

つまり CubOpenHandle のパラメータであるインデックス番号は、1 になります。

3.5.8 CubCloseHandle

書式

```
BOOL CubCloseHandle(HANDLE CUBHandle);
```

機能

CubOpenHandle によって取得したハンドルを閉じます。

パラメータ

HANDLE CUBHandle CUB-43PCIEXP のハンドル

戻り値

正常終了時は TRUE、失敗時は FALSE を返します。

エラーコード

本関数実行後に CubGetLastError が返すエラーコードとエラー発生要因は以下の通りです。

CUB_SUCCESS	正常終了
CUB_ERR_INVALIDPARAM	ハンドルが無効
CUB_ERR_FAILED	原因不明により処理が実行できなかった

3.5.9 CubReadByte

書式

```
BOOL CubReadByte(HANDLE CUBHandle,const ULONG Adr,BYTE *Dat);
```

機能

指定したアドレスから 1 バイトのデータを読み込みます。

パラメータ

HANDLE CUBHandle	CUB-43PCIEXP のハンドル
const ULONG Adr	アドレス値 入力条件は以下の通り ・入力範囲：0x0000 ~ 0x0FFF
BYTE *Dat	読み込みデータ格納先のアドレス

戻り値

正常終了時は TRUE、失敗時は FALSE を返します。

エラーコード

本関数実行後に CubGetLastError が返すエラーコードとエラー発生要因は以下の通りです。

CUB_SUCCESS	正常終了
CUB_ERR_INVALIDPARAM	ハンドルが無効 Adr が範囲外 *Dat に NULL が指定された
CUB_ERR_FAILED	原因不明により処理が実行できなかった

3.5.10 CubWriteByte

書式

```
BOOL CubWriteByte(HANDLE CUBHandle, const ULONG Adr, const BYTE Dat);
```

機能

指定したアドレスへ 1 バイトのデータを書き込みます。

パラメータ

HANDLE CUBHandle	CUB-43PCIEXP のハンドル
const ULONG Adr	アドレス値 入力条件は以下の通り ・入力範囲：0x0000 ~ 0x0FFF
const WORD Dat	書き込みデータ

戻り値

正常終了時は TRUE、失敗時は FALSE を返します。

エラーコード

本関数実行後に CubGetLastError が返すエラーコードとエラー発生要因は以下の通りです。

CUB_SUCCESS	正常終了
CUB_ERR_INVALIDPARAM	ハンドルが無効 Adr が範囲外
CUB_ERR_FAILED	原因不明により処理が実行できなかった

3.5.11 CubReadWord

書式

```
BOOL CubReadWord(HANDLE CUBHandle, const ULONG Adr, WORD *Dat);
```

機能

指定したアドレスから 2 バイトのデータを読み込みます。

パラメータ

HANDLE CUBHandle	CUB-43PCIEXP のハンドル
const ULONG Adr	アドレス値 入力条件は以下の通り ・2 の倍数 ・入力範囲：0x0000 ~ 0x0FFE
WORD *Dat	読み込みデータ格納先のアドレス

戻り値

正常終了時は TRUE、失敗時は FALSE を返します。

エラーコード

本関数実行後に CubGetLastError が返すエラーコードとエラー発生要因は以下の通りです。

CUB_SUCCESS	正常終了
CUB_ERR_INVALIDPARAM	ハンドルが無効 Adr が範囲外 Adr が 2 の倍数でない *Dat に NULL が指定された
CUB_ERR_FAILED	原因不明により処理が実行できなかった

3.5.12 CubWriteWord

書式

BOOL CubWriteWord(HANDLE CUBHandle, const ULONG Adr, const WORD Dat);

機能

指定したアドレスへ 2 バイトのデータを書き込みます。

パラメータ

HANDLE CUBHandle	CUB-43PCIEXP のハンドル
const ULONG Adr	アドレス値 入力条件は以下の通り ・ 2 の倍数 ・ 入力範囲：0x0000 ~ 0x0FFE
const WORD Dat	書き込みデータ

戻り値

正常終了時は TRUE、失敗時は FALSE を返します。

エラーコード

本関数実行後に CubGetLastError が返すエラーコードとエラー発生要因は以下の通りです。

CUB_SUCCESS	正常終了
CUB_ERR_INVALIDPARAM	ハンドルが無効 Adr が範囲外 Adr が 2 の倍数でない
CUB_ERR_FAILED	原因不明により処理が実行できなかった

3.6 サンプルプログラム

本 API を使用した MKY43 の初期化、CUNet 通信設定、グローバルメモリの値を取得するまでのサンプルプログラムを記します。

```
int main(int argc, char *argv[])
{
    HANDLE CUBHandle;
    WORD mky43_scr;
    WORD sa1_gm[4];
    WORD sa63_gm[4];
    int i;
    UINT api_version;

    /** API のバージョンチェック */
    api_version = CubGetVersion();
    if (api_version < 0x200 || api_version > 0x299) {
        printf(" 互換性の無いバージョンの cub43pciexp.dll です。 \n");
        exit(1);
    }

    /** ハンドル生成 */
    CUBHandle = CubOpenHandle(0);
    if (CUBHandle == INVALID_HANDLE_VALUE) {
        exit(1);
    }

    /** MKY43 を初期化 */
    // (1) メモリマップ内の 0x000 ~ 0x2FF(GM + MSB) を 0x00 でライト
    for (i=0;i<0x300;i+=2) {
        CubWriteWord(CUBHandle, i, 0);
    }

    // (2) メモリマップ内の 0x400 ~ 0x5FF(MRB0 + MRB1) を 0x00 でライト
    for (i=0x400;i<0x600;i+=2) {
        CubWriteWord(CUBHandle, i, 0);
    }

    // (3) 通信設定を行う
    // (3-1) BCR ヘライトするために GMM 機能を ON
    CubWriteWord(CUBHandle, 0x366, 0x8000);
    // (3-2) BCR へ通信条件を設定
    // サンプルでは、SA=0,OWN=1,BPS=6Mbps と BCR に設定
    CubWriteWord(CUBHandle, 0x356, 0x0180);
    // (3-3) GMM 機能を OFF
    CubWriteWord(CUBHandle, 0x366, 0x0000);
}
```

```
/** CUnet の起動 */
CubWriteWord(CUBHandle, 0x366, 0x0100);

/** 本サンプルプログラムでは、CUB-43PCIEXP とは別に 2 つの CUnet ステーション (SA1 と SA63) と
 * リンクが成立していると仮定し、SA1 と SA63 のグローバルメモリのデータ読みを行っています。
 */
while(1) {
    /** CUnet のネットワーク状態を確認 */
    CubReadWord(CUBHandle, 0x366, &mky43_scr);
    if ((mky43_scr&0x0100)==0) {
        CubWriteWord(CUBHandle, 0x366, 0x0100); // ネットワークが停止中なら再起動
    }
    // SA1 のグローバルメモリ読み込み
    CubReadWord(CUBHandle, 0x0008, &sa1_gm[0]);
    CubReadWord(CUBHandle, 0x000A, &sa1_gm[1]);
    CubReadWord(CUBHandle, 0x000C, &sa1_gm[2]);
    CubReadWord(CUBHandle, 0x000E, &sa1_gm[3]);
    // SA63 のグローバルメモリ読み込み
    CubReadWord(CUBHandle, 0x01f8, &sa63_gm[0]);
    CubReadWord(CUBHandle, 0x01fA, &sa63_gm[1]);
    CubReadWord(CUBHandle, 0x01fC, &sa63_gm[2]);
    CubReadWord(CUBHandle, 0x01fE, &sa63_gm[3]);
}
/** 生成したハンドルを閉じる */
CubCloseHandle(CUBHandle);
return 0;
}
```


■開発・製造

株式会社ステップテクニカ

〒 358-0011 埼玉県入間市下藤沢 2-32-6

TEL: 04-2964-8804

<https://www.steptecnica.com/>

info@steptecnica.com

**CUnet (MKY43) 搭載 PCI Express ボード
CUB-43PCIEXP
ユーザーズマニュアル**

ドキュメント No. : STD_CUB43PCIEXP_V4.0J

発行年月日 : 2022 年 3 月