P10990-100

CUnet Family

CU-49USB

REFERENCE MANUAL

https://www.steptechnica.com/

CUnet (MKY49) 搭載 USB ユニット

CU-49USB リファレンスマニュアル

CU-49USB は、CUnet IC である MKY49 を搭載し、USB 通信で CUnet を制御するユニットです。

このマニュアルでは、Windows で使用できるドライバーやライブラリーについて説明します。
これを利用することで PC から CUnet を制御することが可能となります。

©2026- StepTechnica Co., Ltd.

はじめに

本マニュアルは、CUnet IC である MKY49 を搭載した CU-49USB のソフトウェアについて記述します。
本マニュアルをお読みになる前に、「**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、弊社の各種 IC を初めてご利用になる方
- CU-49USB を使用して CUnet の評価を行う方
- CU-43USB で CUnet を使用した経験のある方

■ 読者の必要とする知識

- 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ーおよびメモリー）に関する標準的な知識

■ 関連マニュアル

- CU-49USB データシート
- CUnet 導入ガイド
- CUnet テクニカルガイド
- MKY49 データシート
- MKY49 リファレンスマニュアル

【注意事項】

本マニュアル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 ファミリーに関する専門知識は、技術ドキュメント（マニュアル等）を元にご理解ください。

目　次

1　概要

1.1　動作環境 1-2
1.2　著作権 1-2
1.3　機能説明 1-2

2　ドライバー

2.1　ファイル構成 2-1
2.2　インストール 2-2

3　ライブラリー（DLL）

3.1　ファイル構成 3-1
3.2　アドレス空間 3-2
3.2.1　MKY リセット（Adrs=0x3F00） 3-2
3.2.2　ユニット ID 読み出し（Adrs=0x3F02） 3-2
3.3　AutoTrans 機能 3-3
3.4　API リファレンス 3-4
3.4.1　CubGetVersion 3-5
3.4.2　CubGetLastError 3-6
3.4.3　CubCountDevice 3-7
3.4.4　CubSearchBoard 3-8
3.4.5　CubOpenHandle 3-10
3.4.6　CubCloseHandle 3-11
3.4.7　CubBoardID 3-12
3.4.8　CubResetMKY 3-13
3.4.9　CubReadWord 3-14
3.4.10　CubWriteWord 3-15
3.4.11　CubReadData 3-16
3.4.12　CubWriteData 3-17
3.4.13　CubReadProtect 3-18
3.4.14　CubWriteProtect 3-19
3.4.15　CubGetFirmwareVersion 3-20
3.4.16　CubStartAutoTrans 3-21
3.4.17　CubStopAutoTrans 3-22
3.4.18　CubReadGM 3-23
3.4.19　CubReadMFR 3-24
3.5　制限事項 3-25

4　MKY43 互換モードでの動作

4.1　MKY43 のレジスタ 4-2

1　概要

CU-49USB と PC を USB ケーブルで接続し、本マニュアルで説明するドライバー、ライブラリーを使用することでユーザーアプリケーションから CUnet を制御することが可能となります。

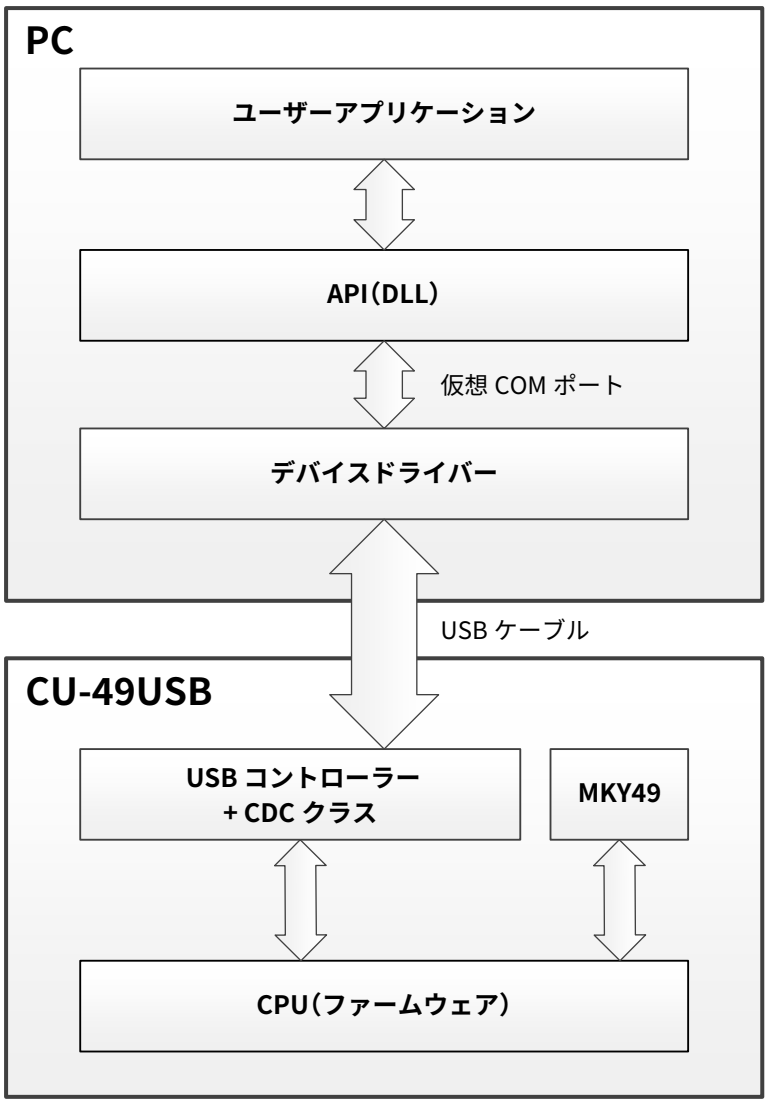


図 1.1　ソフトウェア構成

1.1　動作環境

・CU-49USB と接続するための空き USB ポートを持つ PC
（接続ケーブルは同梱しておりません。お客様にてご準備ください。）

・対応 OS：Windows 11

なお、PC の省電力設定には対応しておりません。

1.2　著作権

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 CU-49USB 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なく、CU-49USB 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 ライブラリー API は、説明に従って正しくお使いください。
③ 仕様・内容は、将来予告な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、一切保証いたしません。
④ OS や開発環境等に関するお問い合わせについてはサポートできません。
⑤ 不具合などを発見された場合は、弊社 Web ページの「お問い合わせフォーム」よりご連絡ください。

1.3　機能説明

本製品は、PC を CUnet ネットワークに接続するための製品です。
別途ご提供する Windows 用のドライバー、ライブラリーを利用することで MKY49 の機能を簡単に利用することができ、CUnet システムを構築する際の評価、およびサブ・ステーション側を開発する時にご利用いただくことができます。
また、MKY43 互換モードをご用意しておりますので、これまで CU-43USB をご使用いただいていた環境をそのまま本製品へ移行することが可能となります。

注意事項

一部未サポートの機能があります。詳しくは“4 MKY43 互換モードでの動作”をご参照ください。

2　ドライバー

CU-49USB 用として、Windows 64bit ドライバーを提供しています。
弊社ホームページよりダウンロードしてお使いください。

URL：https://www.steptechnica.com/jp/products/03_board/cu-49usb/cu-49usb.html

このドライバーでは、Windows 標準の USB シリアル通信用ドライバー（Usbser.sys）を用いて、仮想シリアル通信を行っています。

2.1　ファイル構成

ダウンロードファイルは、zip 形式となっており、次のデータが含まれています。

CU-49USB_Driver_Vx.x.zip	提供ファイル
┣ cu49usb.cat	カタログファイル
┗ cu49usb.inf	インストール用ファイル

2.2　インストール

以下の手順でドライバーをインストールし、PC が CU-49USB を認識できるようにします。

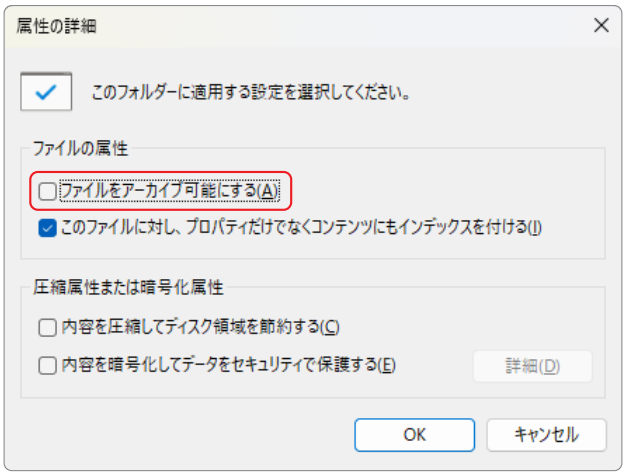
① ダウンロードした zip ファイルを解凍します。

② OS の制限から解凍したファイルのプロパティが変更されている場合があります。
このため、２つのファイルのプロパティを確認します。

(1) セキュリティの項目が表示されている場合、“許可する”のチェックボックスをクリックし、解除します。



(2) ［詳細設定］ボタンを押し、“ファイルをアーカイブ可能にする”のチェックボックスがオンになっていない場合、ここにチェックを入れ［OK］ボタンを押します。

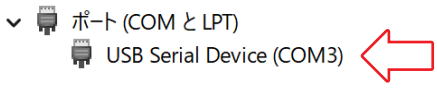


③ Windows 11 のタスクバーにある検索ボックスに“Device Manager”と入力し、結果表示された“Device Manager”をクリックして起動します。

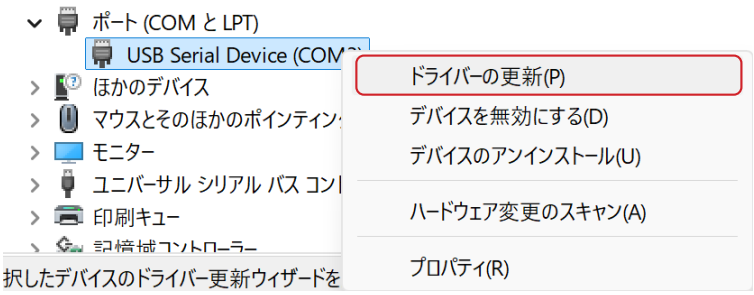


④ CU-49USB を PC に接続します。

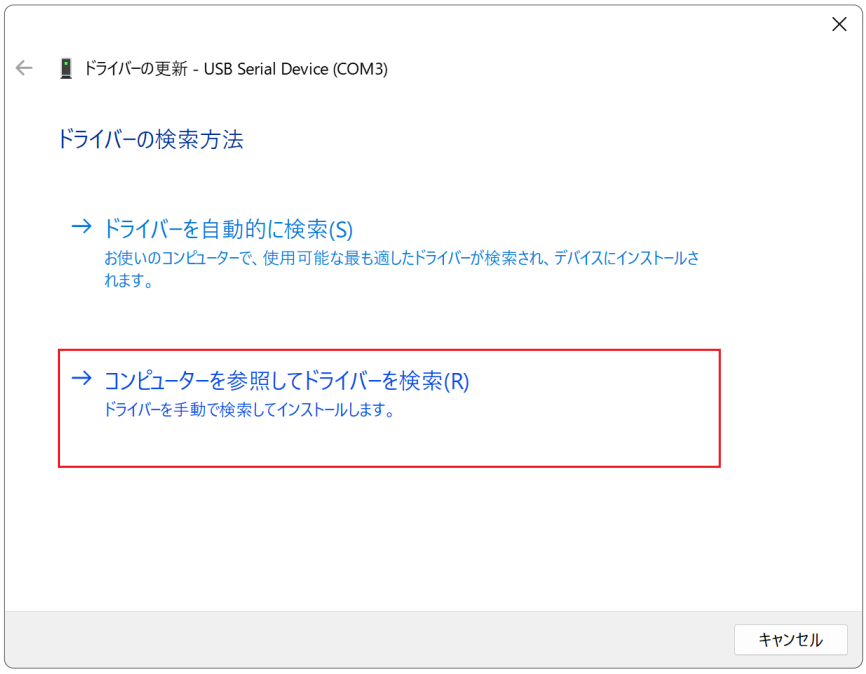
⑤ “ポート (COM と LPT)”に“USB Serial Device(COMn)”が表示されることを確認します。(COMn の‘n’は数字になっています)



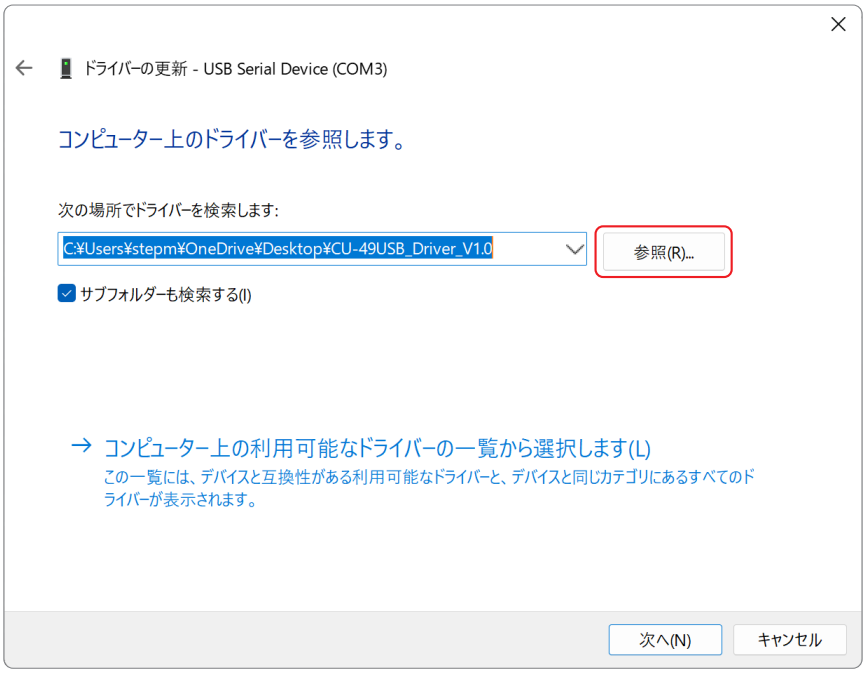
⑥ “USB Serial Device(COMn)”を選択し、マウスの右ボタンをクリックし、“ドライバーの更新”を選択します。



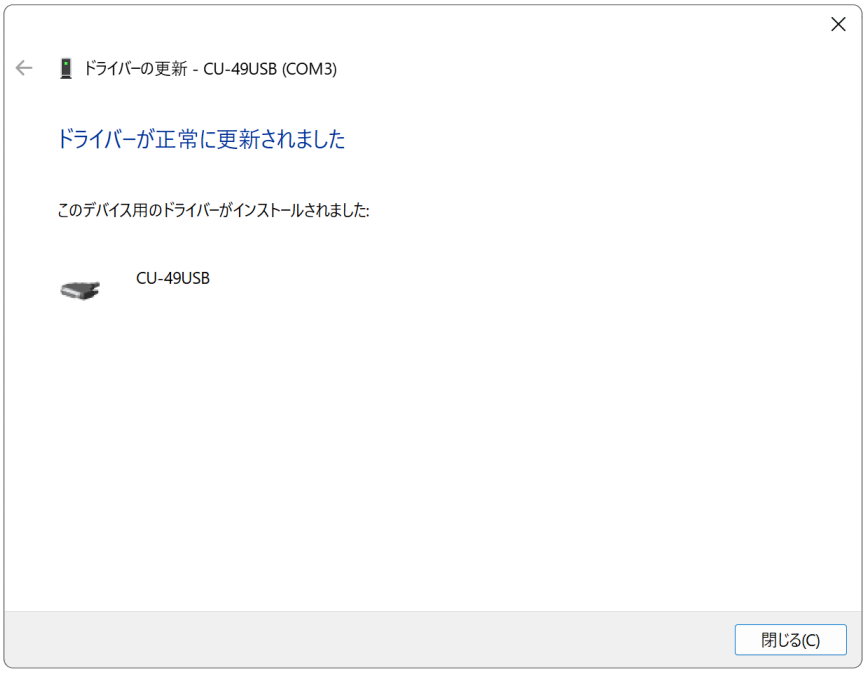
⑦ “コンピューターを参照してドライバーを検索”をクリックします。



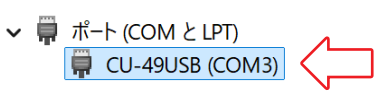
⑧［参照］ボタンをクリックし、解凍したドライバーが格納されているフォルダーを選択し、［次へ］ボタンを押します。



⑨ インストールが実施され、結果が表示されます。



⑩ “Device Manager”での表示が“CU-49USB(COMn)”となっていることを確認します。



3　ライブラリー（DLL）

CU-49USB を制御する際に使用するライブラリーを提供しています。
弊社ホームページよりダウンロードしてお使いください。

URL：https://www.steptechnica.com/jp/products/03_board/cu-49usb/cu-49usb.html

3.1　ファイル構成

ダウンロードファイルは、zip 形式となっており、次のデータが含まれています。

CU-49USB_DLL_Vx.x.zip	提供ファイル
├ cu49usb.dll	ダイナミックライブラリー
├ cu49usb.lib	スタティックライブラリー
├ cu49usb.cs	C# 用のテンプレートファイル
└ cu49usb.h	ライブラリー関数のヘッダファイル

3.2　アドレス空間

MKY49 は 0x0000 ～ 0x23FF までのアドレス空間を持っています。
もちろん、このすべての空間にアクセスできますが、これ以外に特殊レジスタを 2 つ持っています。

表 3.1　CU-49USB 専用レジスタ

アドレス	領域名	Bit 数	R/W	対象機能
0x3F00	MKYReset	16	W	CUnet デバイス（MKY49）のリセット
0x3F02	ReadID	16	R	ユニットの設定スイッチにある ID の読み出し

3.2.1　MKY リセット（Adrs=0x3F00）

CU-49USB に搭載されている CUnet IC（MKY49）をリセットします。
書き込む値は意味を持たず、書き込み動作を行うと CUnet IC のリセット端子を Low にしてリセットします。

MKYReset（Adrs=0x3F00）

ビット：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
初期値：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1

その後、CCR（Chip Code Register）を読み出し、動作していることを確認してから、新たに制御を開始してください。

3.2.2　ユニット ID 読み出し（Adrs=0x3F02）

ユニットに搭載している設定スイッチのうち、1 番・2 番のポジションに割り当てられている ID の状態を返答します。
この値は、電源投入時のものであり、その後に操作されたとしても電源投入時の値を返します。

ReadID（Adrs=0x3F02）

ビット：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	ID1	ID0
初期値：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

3.3　AutoTrans 機能

このライブラリーでも、CU-43USB と同様に AutoTrans 機能を組み込んでいます。
この機能は、設定された周期で CU-49USB からグローバルメモリーの全領域のデータと MFR（Member Flag Register)の値を読み込み､内部に格納します｡その値を専用の API を使って読み出せるものです。

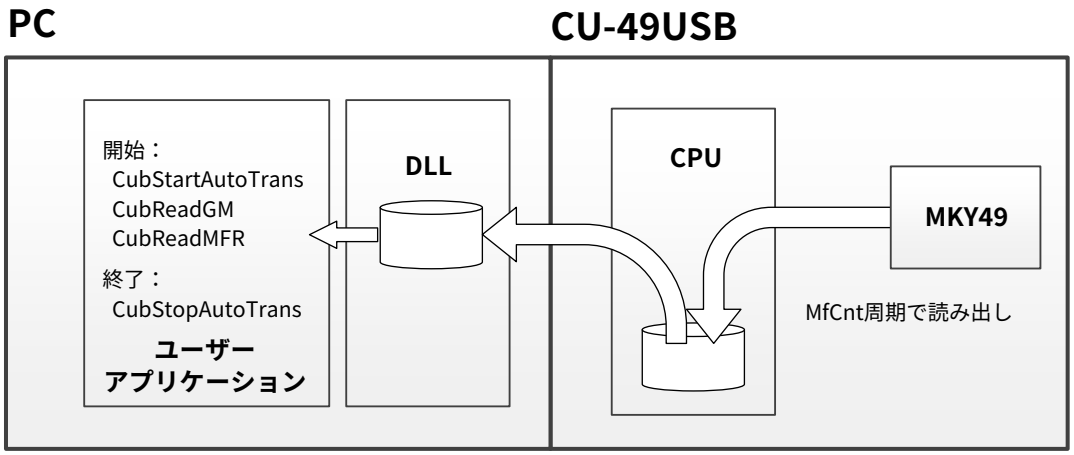


図 3.1　AutoTrans 機能

読み出し周期（MfCnt）は 1msec ～ 100msec の間で 0.125msec 単位で指定することが可能です。

注意事項　周期は、PC のスペックや他に動作しているアプリケーションにより一定にならない可能性があります。

3.4　API リファレンス

表 3.2　API 関数一覧表

API 関数名	機　能
CubGetVersion	API のバージョン番号取得
CubGetLastError	API 関数の終了状態を取得
CubCountDevice	PC に接続されている CU-49USB の台数を取得
CubSearchBoard	PC に接続されている CU-49USB の台数とそのユニット ID を取得
CubOpenHandle	CU-49USB のハンドルをオープン
CubCloseHandle	CubOpenHandle で取得したハンドルをクローズ
CubBoardID	ユニット ID を取得
CubResetMKY	MKY49 へリセットを指示
CubReadWord	指定したアドレスから 2 バイトのデータ読み出し
CubWriteWord	指定したアドレスへ 2 バイトのデータ書き込み
CubReadData	指定したアドレスから指定ワード長のデータ読み出し
CubWriteData	指定したアドレスへ指定ワード長のデータ書き込み
CubReadProtect	グローバルメモリーのデータ読み込み
CubWriteProtect	グローバルメモリーへデータ書き込み
CubGetFirmwareVersion	CU-49USB のファームウェアバージョン番号を取得
CubStartAutoTrans	定期更新の開始
CubStopAutoTrans	定期更新の停止
CubReadGM	定期更新による全グローバルメモリーの最新データを取得
CubReadMFR	定期更新による MFR の最新データを取得

3.4.1　CubGetVersion

書　式

```
WORD CubGetVersion(void);
```

機　能

ライブラリー API のバージョン番号を取得します。

引　数

なし

返り値

バージョン番号　Bit15 ～ 8 ：メジャー番号
Bit7 ～ 4 ：マイナー番号
Bit3 ～ 0 ：ビルド番号
例）0x0102
→メジャー番号＝ 1
マイナー番号＝ 0
ビルド番号＝ 2

エラーコード（実行後に CubGetLastError() で取得）

CUB_SUCCESS　正常終了

補　足

バージョン番号が更新される要因は次のとおりです。
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
マイナー番号：API 関数の追加など、互換性を保ったままの変更がなされた時
ビルド番号：不具合修正など、仕様に影響がなく修正された時

3.4.2　CubGetLastError

書　式

```
WORD 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。

引　数

なし

返り値

定義されているエラーコードを返します。

補　足

エラーが発生した後、別の API を呼び出すとその結果が上書きされます。
エラーの保持機能はありません。

表 3.3　エラーコード表

値	cu49usb.h 内での文字定義	内　容
0	CUB_SUCCESS	正常終了
1	CUB_ERR_DEVICENOTEXIST	デバイスが存在しない
2	CUB_ERR_ALREADYOPENED	すでにオープンされている
3	CUB_ERR_CLOSED	CubOpenHandle が一度もコールされていない
4	CUB_ERR_INVALIDPARAM	無効なパラメータでコールされた
5	CUB_ERR_NORESOURCE	実行に必要なリソースが足りない
6	CUB_ERR_FAILED	原因不明により処理が遂行されなかった
7	CUB_ERR_AUTO_TRANS_ALREADY_START	定期更新がすでに開始されている
8	CUB_ERR_AUTO_TRANS_STOP	定期更新が開始されていない
9	CUB_ERR_USB_TIMEOUT_SUCCESS_STOP_CUNET	USB 通信中にタイムアウトが発生、CUnet 通信の停止に成功
10	CUB_ERR_USB_TIMEOUT_FAILED_STOP_CUNET	USB 通信中にタイムアウトが発生、CUnet 通信の停止に失敗
11	CUB_ERR_REACQUISITION_OF_HANDLE	ハンドルの再取得が行われていない
12	CUB_ERR_NOT_SUPPORT_FIRM_VERSION	サポートされてないファームウェアバージョンを確認
13	CUB_ERR_INVALID_SEQUENCE_NUMBER	シーケンス番号が無効
99	CUB_NOTCALLYET	CUBAPI が一度もコールされていない

3.4.3　CubCountDevice

書　式

```
short CubCountDevice(void);
```

機　能

パソコンに接続されている CU-49USB の台数を返します。

引　数

なし

返り値

CU-49USB の台数　　-1：5 台以上
　　0：1 台も接続されていない
　　1 ～ 4：1 台～ 4 台

エラーコード（CubGetLastError() で取得）

CUB_SUCCESS　　正常終了

補　足

PC に 5 台以上接続することはできません。

3.4.4　CubSearchBoard

書　式

```
BOOL CubSearchBoard(BYTE *unit_num, BYTE *unit_id_list);
```

機　能

PC に接続されている CU-49USB の台数とそのユニット ID リストを返します。

引　数

*unit_num	ユニット台数がセットされるバイト型変数へのアドレス
*unit_id_list	ユニット ID を受け取るためのバイト型を 4 要素持つ配列のポインタ

返り値

正常終了時は TRUE(1)、失敗時は FALSE(0) を返します。
引数で渡したアドレス、ポインタには次の値が格納されます。

*unit_num	ユニットの台数がセットされます。 -1：5 台以上 0：1 台も接続されていない 1 ～ 4：1 台～ 4 台
*unit_id_list	検出したユニットのユニット ID を最大 4 台分まで格納します。 0x00 ～ 0x03：ユニット ID 0x80：既に CubOpenHandle でハンドル値を取得されている 0xFF：存在しない、認識できない

エラーコード（CubGetLastError() で取得）

CUB_SUCCESS	正常終了
CUB_ERR_INVALIDPARAM	*unit_num に NULL が指定された
CUB_ERR_USB_TIMEOUT_SUCCESS_STOP_CUNET	USB 通信中にタイムアウトが発生、 CUnet 通信の停止に成功
CUB_ERR_USB_TIMEOUT_FAILED_STOP_CUNET	USB 通信中にタイムアウトが発生、 CUnet 通信の停止に失敗
CUB_ERR_INVALID_SEQUENCE_NUMBER	シーケンス番号が無効
CUB_ERR_FAILED	原因不明により処理が遂行されなかった

補　足

ユニット ID は、ユニットの設定スイッチの状態により変更されます。
CU-49USB が複数台接続されている場合は、ユニット ID により識別することができます。
本 API 関数では、最大 4 台までの CU-49USB 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。

```
__int8 unit_num;
__int8 unit_id_list[4];
CubSearchBoard(&unit_num, &unit_id_list[0]);
```

*unit_id_list には NULL を指定することが可能です。
この場合、ユニット ID は返さず、ユニットの台数のみを返します。

例）PC に 3 台の CU-49USB が接続されており、それぞれのユニット ID が
1 台目：ユニット ID ＝ 0
2 台目：ユニット ID ＝ 2
3 台目：ユニット ID ＝ 1　と設定されていたとします。

この時に CubSearchBoard を呼び出すと次のように返答されます。

```
unit_num = 3;
unit_id_list[0] = 0、
unit_id_list[1] = 2、
unit_id_list[2] = 1、
unit_id_list[3] = 0xFF
```

3.4.5　CubOpenHandle

書　式

```
HANDLE CubOpenHandle(short index_no);
```

機　能

CU-49USB へのハンドルをオープンします。

引　数

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。

返り値

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（CubGetLastError() で取得）

CUB_SUCCESS	正常終了
CUB_ERR_ALREADYOPENED	すでにオープンされている
CUB_ERR_DEVICENOTEXIST	デバイスが存在しない
CUB_ERR_NOT_SUPPORT_FIRM_VERSION	サポートされてないファームウェアバージョンを確認
CUB_ERR_REACQUISITION_OF_HANDLE	ハンドルの再取得が行われていない
CUB_ERR_FAILED	原因不明により処理が遂行されなかった

補　足

CU-49USB が 1 台の場合は、CubSearchBoard を実行する必要がありません。
CU-49USB が複数台接続されている場合は、“CubSearchBoard” を先に実行し、CubOpenHandle で指定する index_no を決める必要があります。
index_no はユニット ID ではなく、CubSearchBoard の *unit_id_list で返答される配列の index です。

例）CubSearchBoard を呼び出した結果が次の状態である時

```
unit_num = 3;
unit_id_list[0] = 0、
unit_id_list[1] = 2、
unit_id_list[2] = 1、
unit_id_list[3] = 0xFF
```

ユニット ID = 2 の CU-49USB に対してアクセスする場合は、index_no として「1」を指定します。

3.4.6　CubCloseHandle

書　式

```
BOOL CubCloseHandle(HANDLE CUHandle);
```

機　能

CubOpenHandle によって取得したハンドルを閉じます。

引　数

CUHandle	CU-49USB のハンドル値

返り値

正常終了時は TRUE(1)、失敗時は FALSE(0) を返します。

エラーコード（CubGetLastError() で取得）

CUB_SUCCESS	正常終了
CUB_ERR_INVALIDPARAM	無効な CUHandle が指定された
CUB_ERR_INVALID_SEQUENCE_NUMBER	シーケンス番号が無効
CUB_ERR_FAILED	原因不明により処理が遂行されなかった

補　足

定期更新を行っている場合は、定期更新も終了します。

3.4.7　CubBoardID

書　式

```
short CubBoardID(HANDLE CUHandle);
```

機　能

CU-49USB のユニット ID を取得します。

引　数

CUHandle　　CU-49USB のハンドル値

返り値

正常終了時はユニット ID（0 ～ 3）を返します。失敗時は -1 を返します。

エラーコード（CubGetLastError() で取得）

CUB_SUCCESS	正常終了
CUB_ERR_INVALIDPARAM	無効な CUHandle が指定された
CUB_ERR_USB_TIMEOUT_SUCCESS_STOP_CUNET	USB 通信中にタイムアウトが発生、CUnet 通信の停止に成功
CUB_ERR_USB_TIMEOUT_FAILED_STOP_CUNET	USB 通信中にタイムアウトが発生、CUnet 通信の停止に失敗
CUB_ERR_REACQUISITION_OF_HANDLE	ハンドルの再取得が行われていない
CUB_ERR_INVALID_SEQUENCE_NUMBER	シーケンス番号が無効
CUB_ERR_FAILED	原因不明により処理が遂行されなかった

3.4.8　CubResetMKY

書　式

```
BOOL CubResetMKY(HANDLE CUHandle);
```

機　能

ユニットに実装されている CUnet IC（MKY49）をリセットします。

引　数

CUHandle	CU-49USB のハンドル値

返り値

正常終了時は TRUE(1)、失敗時は FALSE(0) を返します。

エラーコード（CubGetLastError() で取得）

CUB_SUCCESS	正常終了
CUB_ERR_INVALIDPARAM	無効な CUHandle が指定された
CUB_ERR_USB_TIMEOUT_SUCCESS_STOP_CUNET	USB 通信中にタイムアウトが発生、CUnet 通信の停止に成功
CUB_ERR_USB_TIMEOUT_FAILED_STOP_CUNET	USB 通信中にタイムアウトが発生、CUnet 通信の停止に失敗
CUB_ERR_REACQUISITION_OF_HANDLE	ハンドルの再取得が行われていない
CUB_ERR_INVALID_SEQUENCE_NUMBER	シーケンス番号が無効
CUB_ERR_FAILED	原因不明により処理が遂行されなかった

3.4.9　CubReadWord

書　式

```
BOOL CubReadWord(HANDLE CUHandle, WORD Adrs, WORD *Data);
```

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。

引　数

CUHandle	CU-49USB のハンドル値
Adrs	読み出し対象アドレス値
*Data	読み込みデータ格納先のアドレス

返り値

正常終了時は TRUE(1)、失敗時は FALSE(0) を返します。

エラーコード（CubGetLastError() で取得）

CUB_SUCCESS	正常終了
CUB_ERR_INVALIDPARAM	無効な CUHandle が指定された Adrs が 2 の倍数でない Adrs が範囲外 *Data に NULL が指定された
CUB_ERR_USB_TIMEOUT_SUCCESS_STOP_CUNET	USB 通信中にタイムアウトが発生、 CUnet 通信の停止に成功
CUB_ERR_USB_TIMEOUT_FAILED_STOP_CUNET	USB 通信中にタイムアウトが発生、 CUnet 通信の停止に失敗
CUB_ERR_REACQUISITION_OF_HANDLE	ハンドルの再取得が行われていない
CUB_ERR_INVALID_SEQUENCE_NUMBER	シーケンス番号が無効
CUB_ERR_FAILED	原因不明により処理が遂行されなかった

補　足

2 バイトデータのアクセスのため、アドレス指定は 2 の倍数でないとエラーとなります。
Adrs 指定範囲は 0x0000 ～ 0x3F02 です。
（CU-43USB のライブラリーは 0x0000 ～ 0x0F02 でした ）

3.4.10 CubWriteWord

書　式

```
BOOL CubWriteWord(HANDLE CUHandle, WORD Adrs, WORD Data);
```

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。

引　数

CUHandle	CU-49USB のハンドル値
Adrs	書き込み対象アドレス値
Data	書き込みデータ

返り値

正常終了時は TRUE(1)、失敗時は FALSE(0) を返します。

エラーコード（CubGetLastError() で取得）

CUB_SUCCESS	正常終了
CUB_ERR_INVALIDPARAM	無効な CUHandle が指定された Adrs が 2 の倍数でない Adrs が範囲外
CUB_ERR_USB_TIMEOUT_SUCCESS_STOP_CUNET	USB 通信中にタイムアウトが発生、 CUnet 通信の停止に成功
CUB_ERR_USB_TIMEOUT_FAILED_STOP_CUNET	USB 通信中にタイムアウトが発生、 CUnet 通信の停止に失敗
CUB_ERR_REACQUISITION_OF_HANDLE	ハンドルの再取得が行われていない
CUB_ERR_INVALID_SEQUENCE_NUMBER	シーケンス番号が無効
CUB_ERR_FAILED	原因不明により処理が遂行されなかった

補　足

2 バイトデータのアクセスのため、アドレス指定は 2 の倍数でないとエラーとなります。
Adrs 指定範囲は 0x0000 ～ 0x3F02 です。
（CU-43USB のライブラリーは 0x0000 ～ 0x0F02 でした ）

3.4.11 CubReadData

書　式

```
BOOL CubReadData(HANDLE CUHandle, WORD Adrs, WORD WordLen, void *Data);
```

機　能

指定したアドレスから指定ワード長のデータを読み込みます。

引　数

CUHandle	CU-49USB のハンドル値
Adrs	読み出し対象アドレス値
WordLen	読み出しワード長
*Data	読み込みデータ格納先のアドレス

返り値

正常終了時は TRUE(1)、失敗時は FALSE(0) を返します。

エラーコード（CubGetLastError() で取得）

CUB_SUCCESS	正常終了
CUB_ERR_INVALIDPARAM	無効な CUHandle が指定された Adrs が 2 の倍数でない Adrs が範囲外 WordLen が範囲外 *Data に NULL が指定された
CUB_ERR_USB_TIMEOUT_SUCCESS_STOP_CUNET	USB 通信中にタイムアウトが発生、 CUnet 通信の停止に成功
CUB_ERR_USB_TIMEOUT_FAILED_STOP_CUNET	USB 通信中にタイムアウトが発生、 CUnet 通信の停止に失敗
CUB_ERR_REACQUISITION_OF_HANDLE	ハンドルの再取得が行われていない
CUB_ERR_INVALID_SEQUENCE_NUMBER	シーケンス番号が無効
CUB_ERR_FAILED	原因不明により処理が遂行されなかった

補　足

ワード（16bit）単位でアクセスのため、アドレス指定は 2 の倍数でないとエラーとなります。

Adrs 指定範囲は 0x0000 ～ 0x3F02 です。

（CU-43USB のライブラリーは 0x0000 ～ 0x0F02 でした ）

WordLen の最大サイズは 0x0180（0x0300 バイト）です。

（CU-43USB のライブラリーは 0x0400（0x0800 バイト）でした ）

3.4.12 CubWriteData

書　式

```
BOOL CubWriteData(HANDLE CUHandle, WORD Adrs, WORD WordLen, void *Data);
```

機　能

指定したアドレスから指定ワード長のデータを書き込みます。

引　数

CUHandle	CU-49USB のハンドル値
Adrs	書き込み対象アドレス値
WordLen	書き込みワード長
*Data	書き込みデータ格納先のアドレス

返り値

正常終了時は TRUE(1)、失敗時は FALSE(0) を返します。

エラーコード（CubGetLastError() で取得）

CUB_SUCCESS	正常終了
CUB_ERR_INVALIDPARAM	無効な CUHandle が指定された Adrs が 2 の倍数でない Adrs が範囲外 WordLen が範囲外 *Data に NULL が指定された
CUB_ERR_USB_TIMEOUT_SUCCESS_STOP_CUNET	USB 通信中にタイムアウトが発生、 CUnet 通信の停止に成功
CUB_ERR_USB_TIMEOUT_FAILED_STOP_CUNET	USB 通信中にタイムアウトが発生、 CUnet 通信の停止に失敗
CUB_ERR_REACQUISITION_OF_HANDLE	ハンドルの再取得が行われていない
CUB_ERR_INVALID_SEQUENCE_NUMBER	シーケンス番号が無効
CUB_ERR_FAILED	原因不明により処理が遂行されなかった

補　足

ワード（16bit）単位でアクセスのため、アドレス指定は 2 の倍数でないとエラーとなります。

Adrs 指定範囲は 0x0000 ～ 0x3F02 です。

（CU-43USB のライブラリーは 0x0000 ～ 0x0F02 でした ）

WordLen の最大サイズは 0x0180（0x0300 バイト）です。

（CU-43USB のライブラリーは 0x0400（0x0800 バイト）でした ）

3.4.13 CubReadProtect

書　式

```
BOOL CubReadProtect(HANDLE CUHandle, WORD BlockNo, void *Data);
```

機　能

ハザード防止機能を使用してグローバルメモリーのデータを読み込みます。

引　数

CUHandle	CU-49USB のハンドル値
BlockNo	読み出し対象メモリーブロック番号
*Data	読み込みデータ格納先（8 バイト分）のアドレス

返り値

正常終了時は TRUE(1)、失敗時は FALSE(0) を返します。

エラーコード（CubGetLastError() で取得）

CUB_SUCCESS	正常終了
CUB_ERR_INVALIDPARAM	無効な CUHandle が指定された BlockNo が範囲外 *Data に NULL が指定された
CUB_ERR_USB_TIMEOUT_SUCCESS_STOP_CUNET	USB 通信中にタイムアウトが発生、 CUnet 通信の停止に成功
CUB_ERR_USB_TIMEOUT_FAILED_STOP_CUNET	USB 通信中にタイムアウトが発生、 CUnet 通信の停止に失敗
CUB_ERR_REACQUISITION_OF_HANDLE	ハンドルの再取得が行われていない
CUB_ERR_INVALID_SEQUENCE_NUMBER	シーケンス番号が無効
CUB_ERR_FAILED	原因不明により処理が遂行されなかった

補　足

BlockNo 指定範囲は 0 ～ 63 です。
MKY49 では、一つのグローバルメモリー内でのハザードは発生しません。
このため CubReadData を用いてもグローバルメモリーの 1 ブロックであれば同じです。
この API は CU-43USB との互換を保つために用意しています。

3.4.14 CubWriteProtect

書　式

```
BOOL CubWriteProtect(HANDLE CUHandle, WORD BlockNo, void *Data);
```

機　能

ハザード防止機能を使用してグローバルメモリーへデータを書き込みます。

引　数

CUHandle	CU-49USB のハンドル値
BlockNo	書き込み対象メモリーブロック番号
*Data	書き込みデータ格納先（8 バイト分）のアドレス

返り値

正常終了時は TRUE(1)、失敗時は FALSE(0) を返します。

エラーコード（CubGetLastError() で取得）

CUB_SUCCESS	正常終了
CUB_ERR_INVALIDPARAM	無効な CUHandle が指定された BlockNo が範囲外 *Data に NULL が指定された
CUB_ERR_USB_TIMEOUT_SUCCESS_STOP_CUNET	USB 通信中にタイムアウトが発生、 CUnet 通信の停止に成功
CUB_ERR_USB_TIMEOUT_FAILED_STOP_CUNET	USB 通信中にタイムアウトが発生、 CUnet 通信の停止に失敗
CUB_ERR_REACQUISITION_OF_HANDLE	ハンドルの再取得が行われていない
CUB_ERR_INVALID_SEQUENCE_NUMBER	シーケンス番号が無効
CUB_ERR_FAILED	原因不明により処理が遂行されなかった

補　足

BlockNo 指定範囲は 0 ～ 63 です。

MKY49 では、一つのグローバルメモリー内でのハザードは発生しません。

このため CubWriteData を用いてもグローバルメモリーの 1 ブロックであれば同じです。

この API は CU-43USB との互換を保つために用意しています。

3.4.15 CubGetFirmwareVersion

書　式

```
short CubGetFirmwareVersion(HANDLE CUHandle);
```

機　能

CU-49USB のファームウェアバージョン番号を取得します。

引　数

CUHandle	CU-49USB のハンドル値

返り値

バージョン番号	Bit15 ～ 8 ：メジャー番号
	Bit7 ～ 4 ：マイナー番号
	Bit3 ～ 0 ：ビルド番号
例）	0x0104
	→メジャー番号＝ 1
	マイナー番号＝ 0
	ビルド番号＝ 4

エラーコード（CubGetLastError() で取得）

CUB_SUCCESS	正常終了
CUB_ERR_INVALIDPARAM	無効な CUHandle が指定された
CUB_ERR_USB_TIMEOUT_SUCCESS_STOP_CUNET	USB 通信中にタイムアウトが発生、CUnet 通信の停止に成功
CUB_ERR_USB_TIMEOUT_FAILED_STOP_CUNET	USB 通信中にタイムアウトが発生、CUnet 通信の停止に失敗
CUB_ERR_REACQUISITION_OF_HANDLE	ハンドルの再取得が行われていない
CUB_ERR_INVALID_SEQUENCE_NUMBER	シーケンス番号が無効
CUB_ERR_FAILED	原因不明により処理が遂行されなかった

補　足

バージョン番号が更新される要因は次のとおりです。

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
マイナー番号：API 関数の追加など、互換性を保ったままの変更がなされた時
ビルド番号：不具合修正など、仕様に影響がなく修正された時

3.4.16 CubStartAutoTrans

書　式

```
BOOL CubStartAutoTrans(HANDLE CUHandle, WORD MfCnt);
```

機　能

CU-49USB の全グローバルメモリーと MFR の定期更新を開始します。

引　数

CUHandle	CU-49USB のハンドル値
MfCnt	更新周期

返り値

正常終了時は TRUE(1)、失敗時は FALSE(0) を返します。

エラーコード（CubGetLastError() で取得）

CUB_SUCCESS	正常終了
CUB_ERR_INVALIDPARAM	無効な CUHandle が指定された MfCnt が範囲外
CUB_ERR_AUTO_TRANS_ALREADY_START	定期更新がすでに開始されている
CUB_ERR_USB_TIMEOUT_SUCCESS_STOP_CUNET	USB 通信中にタイムアウトが発生、 CUnet 通信の停止に成功
CUB_ERR_USB_TIMEOUT_FAILED_STOP_CUNET	USB 通信中にタイムアウトが発生、 CUnet 通信の停止に失敗
CUB_ERR_REACQUISITION_OF_HANDLE	ハンドルの再取得が行われていない
CUB_ERR_INVALID_SEQUENCE_NUMBER	シーケンス番号が無効
CUB_ERR_FAILED	原因不明により処理が遂行されなかった

補　足

MfCnt は 125μsec 単位で 1msec から 100msec を指定することができます。
すなわち、数値としては 8 ～ 800 までの範囲となります。
PC のスペックや他に動作しているアプリケーションにより、指定した周期で定期更新ができない可能性がありますので注意してください。
本 API を使用後、CubReadGM、CubReadMFR を使って定期更新データを取得します。

3.4.17 CubStopAutoTrans

書　式

```
BOOL CubStopAutoTrans(HANDLE CUHandle);
```

機　能

CU-49USB の全グローバルメモリーと MFR の定期更新を停止します。

引　数

CUHandle	CU-49USB のハンドル値

返り値

正常終了時は TRUE(1)、失敗時は FALSE(0) を返します。

エラーコード（CubGetLastError() で取得）

CUB_SUCCESS	正常終了
CUB_ERR_INVALIDPARAM	無効な CUHandle が指定された
CUB_ERR_AUTO_TRANS_STOP	定期更新が開始されていない
CUB_ERR_USB_TIMEOUT_SUCCESS_STOP_CUNET	USB 通信中にタイムアウトが発生、CUnet 通信の停止に成功
CUB_ERR_USB_TIMEOUT_FAILED_STOP_CUNET	USB 通信中にタイムアウトが発生、CUnet 通信の停止に失敗
CUB_ERR_REACQUISITION_OF_HANDLE	ハンドルの再取得が行われていない
CUB_ERR_INVALID_SEQUENCE_NUMBER	シーケンス番号が無効
CUB_ERR_FAILED	原因不明により処理が遂行されなかった

補　足

定期更新を行っている時に、CubCloseHandle を呼び出すと、この API を呼ばずとも定期更新は停止します。

3.4.18 CubReadGM

書　式

```
BOOL CubReadGM(HANDLE CUHandle, void *Data);
```

機　能

定期更新による全グローバルメモリーの最新データを取得します。

引　数

CUHandle	CU-49USB のハンドル値
*Data	グローバルメモリーデータ格納先（512 バイト分）のアドレス

返り値

正常終了時は TRUE(1)、失敗時は FALSE(0) を返します。

エラーコード（CubGetLastError() で取得）

CUB_SUCCESS	正常終了
CUB_ERR_INVALIDPARAM	無効な CUHandle が指定された *Data に NULL が指定された
CUB_ERR_AUTO_TRANS_STOP	定期更新停止中
CUB_ERR_REACQUISITION_OF_HANDLE	ハンドルの再取得が行われていない
CUB_ERR_FAILED	原因不明により処理が遂行されなかった

補　足

CubStartAutoTrans を実行すると内部で定期的にグローバルメモリーのデータと MFR を読み出しています。この API でそのグローバルメモリーのデータを取得できます。
定期更新停止中に呼び出した場合はエラーとなります。
MKY49 に直接アクセスしているわけではありません。そのため、最新のデータが必要な場合などは、CubReadWord、CubReadData を使用してください。

3.4.19　CubReadMFR

書　式

```
BOOL CubReadMFR(HANDLE CUHandle, void *Data);
```

機　能

定期更新による MFR の最新データを取得します。

引　数

CUHandle	CU-49USB のハンドル値
*Data	MFR 格納先（8 バイト分）のアドレス

返り値

正常終了時は TRUE(1)、失敗時は FALSE(0) を返します。

エラーコード（CubGetLastError() で取得）

CUB_SUCCESS	正常終了
CUB_ERR_INVALIDPARAM	無効な CUHandle が指定された *Data に NULL が指定された
CUB_ERR_AUTO_TRANS_STOP	定期更新停止中
CUB_ERR_REACQUISITION_OF_HANDLE	ハンドルの再取得が行われていない
CUB_ERR_FAILED	原因不明により処理が遂行されなかった

補　足

CubStartAutoTrans を実行すると内部で定期的にグローバルメモリーのデータと MFR を読み出しています。この API でその MFR のデータを取得できます。
定期更新停止中に呼び出した場合はエラーとなります。
MKY49 に直接アクセスしているわけではありません。そのため、最新のデータが必要な場合などは、CubReadWord、CubReadData を使用してください。

3.5　制限事項

このライブラリーは、CU-43USB と同様に以下の制限事項がありますので、留意してご使用ください。

■ マルチスレッド

本 API 関数は、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は、同時呼び出しが起こらないようご配慮ください。

■ USB 通信でのタイムアウト

本 API では、CU-49USB とのデータ送受信時の最大待機時間（タイムアウト時間）を 1 秒としています。
ご使用環境によりタイムアウト時間が経過しても送受信が完了しない（タイムアウト）場合があります。
タイムアウトが発生した API 関数の返り値は、エラーを返します。
この後、CubGetLastError 関数でエラーコードを取得することができますが、2 つの状態が発生します。

- CUB_ERR_USB_TIMEOUT_SUCCESS_STOP_CUNET（9）
  タイムアウトが発生すると API 内部にて CUnet ネットワークの停止と定期更新機能の停止を行います。この時、正常に CUnet ネットワークの停止と定期更新機能の停止に成功した場合のエラーコードです。
- CUB_ERR_USB_TIMEOUT_FAILED_STOP_CUNET（10）
  ネットワークの停止と定期更新機能の停止に失敗した場合のエラーコードです。

タイムアウト発生後は、CubCloseHandle 関数にて使用しているハンドルをクローズし、
CubOpenHandle 関数にてハンドルを再取得してください。
ハンドルの再取得が行われるまで CubGetVersion、CubCountDevice、CubSearchBoard、
CubGetLastError、CubOpenHandle、CubCloseHandle 以外の API 関数の返り値にはエラーを返します。
その際に CubGetLastError 関数は、CUB_ERR_REACQUISITION_OF_HANDLE（11）を返します。

4　MKY43 互換モードでの動作

CU-49USB は CU-43USB をシミュレートする MKY43 互換モードを持っています。
設定スイッチの 4 番目のポジションを ON にして電源を投入すると、CU-43USB として動作します。
この時は、CU-43USB のドライバーが必要ですので、CU-43USB のマニュアルを参照の上、ドライバーをインストールしてお使いください。

CU-43USB に対して準備されたユーザーアプリケーションの多くは、この MKY43 互換モードで使用できる場合があります。
しかし、MKY49 と MKY43 では IC 単体で同じでない部分がいくつかあります。
この章ではこの相違点について説明します。

4.1　MKY43 のレジスタ

CU-49USB は制御 IC として MKY49 を搭載しています。CU-43USB は MKY43 を搭載していました。この 2 つのデバイスは完全互換ではありません。
この違いについては、「**MKY49 リファレンスマニュアル**」の “付録 2:MKY43 との相違点” をご覧ください。

CU-49USB の互換モードでアクセスできるレジスター覧を表 4.1 に示します。

ここでは、

- 単にレジスタアドレスを変換してハード互換の機能を使うもの
- 機能的には互換になっていないが、CU-49USB のファームウェアで吸収するもの
- 一部機能を互換にできないものが残るもの
- 機能がなく実現できないもの

に分かれます。

機能面から説明すると、次の点が異なります。

◆ DataRenewal 機能
MKY49 には、DataRenewal の機能がありません。
このため、これに関連するレジスタ、フラグはサポートしていません。

◆ INT1CR ／ IT1CR ／ INT1SR
MKY49 には、割込み端子が一つしかありません。
このため、これに関連するレジスタはサポートしていません。

◆ INT0CR ／ INT0SR ／ SSR の MGNC・MGNE
MFR に関する動作が MKY49 では変更になっています。
MKY43 の MGNC（bit5）は MGR ＞ MFR の状態を示し、MGNE（bit4）は MGR ≠ MFR の状態を示します。
一方 MKY49 の bit5 は MNG で MGR ＞ MFR の状態を示し、bit4 は MOK で MGR ≦ MFR の状態を示します。
このため互換モードでは、MNG は MGNC に割り当てられ、MGNE はサポートせず常に “0” となります。

◆ハザード防止機能
MKY49 は SPI インターフェイスであり、一つのメモリーブロック内ではハザードが発生しない仕組みになっています。
このため、ソフトウェアの互換性を保つため、シミュレートを行っています。
ただし、ライトプロテクト機能はないため、GMM モード中や自己占有領域以外であっても MKY49 へ書き込み動作を行います。

表 4.1　MKY43 レジスタ互換表

アドレス	領域名	Bit 数	対象機能	互　換	説　明
0x0000	GM	64*64	共有メモリー	◎	
0x0200	MSB	64*32	メール送信バッファー	◎	
0x0300	RFR	64	リンク検出	◎	
0x0308	LFR	64	リンク検出	◎	
0x0310	MFR	64	メンバー検出	◎	
0x0318	DRFR	64	データ遷移検出	×	
0x0320	LGR	64	リンク検出	◎	
0x0328	MGR	64	メンバー検出	◎	
0x0330	DRCR	64	データ遷移検出	×	
0x0338	RHCR0	16	リードハザード防止	○	
0x033A	RHCR1	16	リードハザード防止	○	
0x033C	WHCR0	16	ライトハザード防止	○	
0x033E	WHCR1	16	ライトハザード防止	○	
0x0340	MSLR	16	メール送信	◎	
0x0342	MSRR	16	メール送信	◎	
0x0344	MESR	16	メール送信	◎	
0x0346	MSCR	16	メール送信	◎	
0x0348	MR0CR	16	メール受信	○	
0x034A	MR1CR	16	メール受信	○	
0x034C	CCTR	16	システム補助	○	
0x034E	UTCR	16	システム補助	×	
0x0350	QCR	16	システム補助	◎	Type=0x05 追加
0x0352	NFSR	16	システム	◎	
0x0354	FSR	16	システム	◎	
0x0356	BCR	16	システム	○	
0x0358	INT0CR	16	割込み制御	△	MGNC ≦ MNG、MGNE/DR なし
0x035A	INT1CR	16	割込み制御	×	
0x035C	IT0CR	16	割込み制御	△	DR なし
0x035E	IT1CR	16	割込み制御	×	
0x0360	INT0SR	16	割込み制御	△	MGNC ≦ MNG、MGNE/DR なし
0x0362	INT1SR	16	割込み制御	×	
0x0364	SSR	16	システム	○	
0x0366	SCR	16	システム	△	DR なし
0x0368	CCR	64	システム	◎	
0x0370	RHPB0	64	リードハザード防止	○	
0x0378	RHPB1	64	リードハザード防止	○	
0x0380	WHPB0	64	ライトハザード防止	○	
0x0388	WHPB1	64	ライトハザード防止	○	
0x0400	MRB0	64*32	メール受信バッファー 0	◎	
0x0500	MRB1	64*32	メール受信バッファー 1	◎	

◎：ハード互換
○：ソフト互換
△：一部機能違い
×：サポートなし

改訂履歴

バージョン No.	更新年月日	ページ	改訂内容
100	2026 年 9 月	ー	新規作成

ご注意

1. 本資料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、本資料が最新の版であるかをご確認ください。
2. 本資料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いただくための参考資料です。
3. 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。
4. 本資料に記載された情報、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。
5. 本製品および本資料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いたしません。
6. 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。
7. 本資料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。

CUnet（MKY49）搭載 USB ユニット

CU-49USB リファレンスマニュアル

ドキュメント No.：P10990-100
発行年月日：2026 年 9 月

■開発・製造

株式会社ステップテクニカ

〒207-0021　東京都東大和市立野1−1−15
https://www.steptechnica.com/
info@steptechnica.com

©2026- StepTechnica Co., Ltd.