

リファレンスマニュアル

C U n e t D L L

CN43eMst. DLL

G8CNMst. DLL

G8CNMst64. DLL

目 次

第 1 章 関数一覧

第 2 章 関数仕様

2-1 Utility	2-1
2-2 Open/Close	2-5
2-3 メール	2-11
2-4 レジスタアクセス	2-18
2-5 GMアクセス	2-24

第 1 章 関数一覧

1) Utility

関 数	機 能
CN_boardname()	ボード名称を取得します
CN_dllname()	ライブラリ名称を取得します
CN_version()	ライブラリの Version を取得します
CN_maxline()	最大 Line 数を取得します

2) Open/Close

関 数	機 能
CN_open()	ボードをオープンします
CN_close()	ボードをクローズします
CN_start()	通信を開始します
CN_stop()	通信を停止します
CN_reset()	自己の占有エリアの GM をクリアします

3) 位置決めユニット

関 数	機 能
CN_MailOpen()	メールをオープンします
CN_MailClose()	メールをクローズします
CN_MailSend()	メールを送信します
CN_MailReceive()	メールの受信データを取得します
CN_MailStatus()	メールのステータスを取得します
CN_MailWaitObject()	メール送信完了、メール受信、メールクローズのいずれかが発生したときに制御を戻します

4) レジスタアクセス

関 数	機 能
CN_GetReg64()	64 ビットアクセスでレジスタを読みみます
CN_GetReg32()	32 ビットアクセスでレジスタを読みみます
CN_GetReg16()	16 ビットアクセスでレジスタを読みみます
CN_SetReg64()	64 ビットアクセスでレジスタを書込みます
CN_SetReg32()	32 ビットアクセスでレジスタを書込みます
CN_SetReg16()	16 ビットアクセスでレジスタを書込みます

5) GM アクセス

関 数	機 能
CN_GetMemShort()	16 ビットアクセスで GM からデータを読みみます
CN_GetMemLong()	32 ビットアクセスで GM からデータを読みみます
CN_GetMemDLong()	64 ビットアクセスで GM からデータを読みみます
CN_SetMemShort()	16 ビットアクセスで GM にデータを書込みます
CN_SetMemLong()	32 ビットアクセスで GM にデータを書込みます
CN_SetMemDLong()	64 ビットアクセスで GM にデータを書込みます

- * CN43eMst.dll では 8bit アクセスをサポートしていません。
- * G8CNMst.dll では 8bit アクセスをサポートしていません。
- * G8CNMst64.dll では 8bit アクセスをサポートしていません。

第2章 関数仕様

2-1 Utility

CN_boardname 関数

機能	ボード名称を取得します
書式	void CN_boardname(char *BrdName)
引数	BrdName : ボード名称
戻り値	なし
説明	ボード名称を取得します

CN_dllname 関数

機能	ライブラリ名称を取得します
書式	<code>void CN_dllname(char *DllName)</code>
引数	<code>DllName</code> : DLL 名称
戻り値	なし
説明	ライブラリ名称を取得します。

CN_version 関数

機能 ライブラリの Version を取得します

書式 void CN_version(char *Version)

引数 Version : Version 名称

戻り値 なし

説明 ライブラリの Version を取得します。

CN_maxline 関数

機能 最大 Line 数を取得します

書式 `int CN_maxline(void)`

引数 なし

戻り値 最大 Line 数

説明 最大 Line 数を取得します。

2-2 Open/Close

CN_open 関数

機能 ボードをオープンします

書式 `int CN_open(int Board)`

引数 `Board` : ボード番号 (0 のみ)

戻り値

<code>MKY43_ER_OK</code>	: 正常
<code>MKY43_ER_INVALIDPARAM</code>	: 無効な引数です
<code>MKY43_ER_ALREADYOPEN</code>	: 既にオープンしています
<code>MKY43_ER_CREATESEMAPH</code>	: セマフォ作成に失敗しました
<code>MKY43_ER_CREATETHREAD</code>	: スレッド作成に失敗しました
<code>MKY43_ER_CREATEEVENT</code>	: イベント作成に失敗しました
<code>MKY43_ER_OPENDEVICE</code>	: デバイスドライバ起動に失敗しました

説明

`Board` で指定された CUnet マスタボードをオープンします。
`Board` に存在しないボード番号を指定した場合、`[MKY43_ER_INVALIDPARAM]` を返します。
ボード番号は CUnet 拡張基板の場合、0 に固定されます。
複数のアプリケーションでこの関数をコールしても正常であれば `[MKY43_ER_OK]` を返し正常に動作します。

CN_close 関数

機能 ボードをクローズします

書式 int CN_close(int Board)

引数 Board : ボード番号 (0 のみ)

戻り値 MKY43_ER_OK : 正常
 MKY43_ER_INVALIDPARAM : 無効な引数です
 MKY43_ER_NOTOPEN : オープンされていません

説明 Board で指定された CUnet マスタボードをクローズします。
 Board に存在しないボード番号を指定した場合、[MKY43_ER_INVALIDPARAM] を返します。

CN_start 関数

機能

通信を開始します

書式`int CN_start(int Board, int Line, int StationAdr, int StationSize, int ByteBusSize)`**引数**

Board : ボード番号 (0 のみ)
Line : ライン番号 (0 のみ)
StationAdr : ステーションアドレス (0~63)
StationSize : ステーションサイズ (1~64)
ByteBusSize : 予約 (無視されます)

戻り値

MKY43_ER_OK : 正常
MKY43_ER_INVALIDPARAM : 無効な引数です
MKY43_ER_NOTOPEN : オープンされていません
MKY43_ER_ALREADYCONNECT : コネクト済みです
MKY43_ER_CONNECT : CUnet にコネクトできません (していません)

説明

CUnet と通信を開始します。
StationAdr で指定されたステーションアドレスから、StationSize で指定されたサイズ分、自己の GM エリアを占有します。
ライン番号はボード上に CUnet 通信 IC が複数ある場合の番号です。
PC カードの場合は CUnet 通信 IC が一個ですので、0 : 固定になります。
複数のアプリケーションでこの関数をコールした場合、現状態での CUnet 通信が開始、停止いずれの状態でも再度関数で指定されたパラメータで通信を開始します。
他のアプリケーションですでに通信開始されている場合は注意が必要です。

CN_stop 関数

機能	通信を停止します		
書式	int CN_stop(int Board, int Line)		
引数	Board	:	ボード番号 (0 のみ)
	Line	:	ライン番号 (0 のみ)
戻り値	MKY43_ER_OK	:	正常
	MKY43_ER_INVALIDPARAM	:	無効な引数です
	MKY43_ER_NOTOPEN	:	オープンされていません
説明	CUnet との通信を停止します。		

CN_reset 関数

機能	自己の占有エリアの GM をクリアします		
書式	int CN_reset(int Board, int Line, WORD Data)		
引数	Board	:	ボード番号 (0 のみ)
	Line	:	ライン番号 (0 のみ)
	Data	:	データ
戻り値	MKY43_ER_OK	:	正常
	MKY43_ER_INVALIDPARAM	:	無効な引数です
	MKY43_ER_NOTOPEN	:	オープンされていません
説明	Data で指定された値で、自己の占有エリアの GM をクリアします。		

CN_GetCnectStat 関数

機能 接続ステータスを取得します

書式 int CN_GetCnectStat(int Board, int Line, int *StationAdr, int *StationSize)

引数

Board	: ボード番号 (0 のみ)
Line	: ライン番号 (0 のみ)
StationAdr	: ステーションアドレス
StationSize	: ステーションサイズ

戻り値

MKY43_ER_OK	: 正常
MKY43_ER_INVALIDPARAM	: 無効な引数です
MKY43_ER_NOTOPEN	: オープンされていません

説明 自己のステーションアドレスとステーションサイズを取得します。

2-3 メール

CN_MailOpen 関数

機能 メールをオープンします

書式 `int CN_MailOpen(int Board, int Line)`

引数

Board	: ボード番号 (0 のみ)
Line	: ライン番号 (0 のみ)

戻り値

MKY43_ER_OK	: 正常
MKY43_ER_INVALIDPARAM	: 無効な引数です
MKY43_ER_NOTOPEN	: オープンされていません
MKY43_ER_UNCONNECT	: CUnet にコネクしていません
MKY43_ER_ALREADYMAILOPEN	: メールが既にオープンしています

説明

メールをオープンします。
この関数コール後、その他メール機能が使用可能となります。
メール機能を使用の際には、必ずコールする必要があります。
メール機能は複数アプリケーションでの使用はできません。
別のアプリケーションで既にオープンがコールされている場合、関数は
[MKY43_ER_ALREADYMAILOPEN] を返します。

CN_MailClose 関数

機能 メールをクローズします

書式 int CN_MailClose(int Board, int Line)

引数 Board : ボード番号 (0 のみ)
 Line : ライン番号 (0 のみ)

戻り値 MKY43_ER_OK : 正常
 MKY43_ER_INVALIDPARAM : 無効な引数です
 MKY43_ER_NOTOPEN : オープンされていません
 MKY43_ER_UNCONNECT : CUnet にコネクしていません
 MKY43_ER_MAILNOTOPEN : メールがオープンされていません

説明 メールをクローズします。
 この関数コール後、その他メール機能が使用不可になります。
 アプリケーションの終了時に必ずコールする必要があります。

CN_MailSend 関数

機能

メールを送信します (256 バイト以下のデータを送信します)

書式

```
int CN_MailSend(int Board, int Line, int StationAdr, void *pMail, int Len)
```

引数

Board : ボード番号 (0 のみ)
Line : ライン番号 (0 のみ)
StationAdr : 送信先のステーションアドレス (0~63)
pMail : 送信するデータのポインタ
Len : 送信するデータのバイトサイズ (1~256)

戻り値

MKY43_ER_OK : 正常
MKY43_ER_INVALIDPARAM : 無効な引数です
MKY43_ER_NOTOPEN : オープンされていません
MKY43_ER_UNCONNECT : CUnet にコネクしていません
MKY43_ER_MAILSIZE : サイズエラーです
MKY43_ER_MAILSENDING : 送信中の為送信できません

説明

指定したステーションアドレスに指定したサイズのデータを送信します。
Len に 256 を超える値を指定した場合、および奇数値を指定した場合、関数はメール送信せずに[MKY43_ER_MAILSIZE]を返します。

送信後の結果は[CN_MailWaitObject]関数、[CN_MailStatus]関数より取得できます。

CN_MailReceive 関数

機能

メールの受信データを取得します

書式

```
int CN_MailReceive(int Board, int Line, void *pMail, int Len, int *pStationAdr)
```

引数

Board : ボード番号 (0 のみ)
Line : ライン番号 (0 のみ)
pMail : 受信したデータを格納するポインタ
Len : 受信したデータを取得するバイトサイズ (1~256)
pStationAdr : 送信元のステーションアドレスを格納するポインタ

戻り値

MKY43_ER_OK : 正常
MKY43_ER_INVALIDPARAM : 無効な引数
MKY43_ER_NOTOPEN : オープンされていません
MKY43_ER_UNCONNECT : CUnet にコネクしていません
MKY43_ER_MAILSIZE : サイズエラー
MKY43_ER_MAILNOTRECEIVE : メールを受信していません
上記以外 : エラー

説明

指定したサイズのデータを受信します。
Len に 256 を超える値を指定した場合、および奇数値を指定した場合、関数はメール受信せずに[MKY43_ER_MAILSIZE]を返します。
メール着信のシグナルは[CN_MailWaitObject]関数、[CN_MailStatus]関数より取得できます。

CN_MailStatus 関数

機能

メールのステータスを取得します

書式

int CN_MailStatus(int Board, int Line, unsigned short *Status)

引数

Board

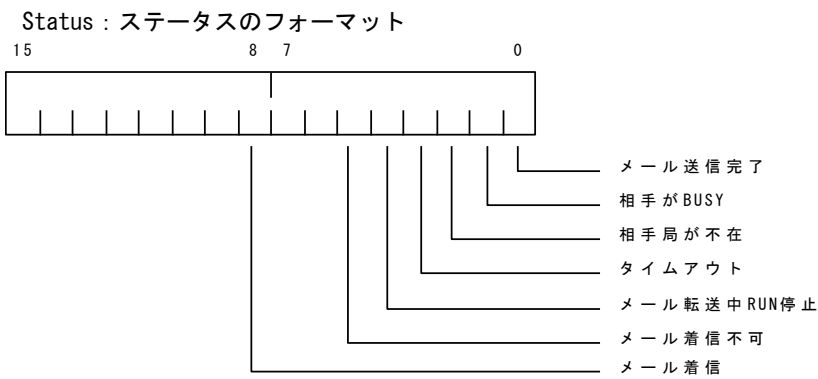
: ボード番号 (0 のみ)

Line

: ライン番号 (0 のみ)

Status

: ステータスを格納するポインタ



* メール着信不可 Bit が ON した時は、CUnet ラインが一度切断された可能性がある為、メールをクローズし、再度メールをオープンする必要があります。

Status	値	内容
MKY43_MAIL_STATUS_SENDE	0x0001	メール送信完了
MKY43_MAIL_STATUS_BUSY	0x0002	相手が BUSY
MKY43_MAIL_STATUS_NOEXIST	0x0004	相手局が不在
MKY43_MAIL_STATUS_TIMEOUT	0x0008	タイムアウト
MKY43_MAIL_STATUS_TRANSSTOP	0x0010	メール転送中 RUN 停止
MKY43_MAIL_STATUS_NOTRECEIVE	0x0020	メール着信不可
MKY43_MAIL_STATUS_RECEIVE	0x0100	メール着信

戻り値	MKY43_ER_OK	: 正常
	MKY43_ER_INVALIDPARAM	: 無効な引数です
	MKY43_ER_NOTOPEN	: オープンされていません
	MKY43_ER_UNCONNECT	: CUnet にコネクしていません

説明	メールのステータスを取得します。
	メール送信をすると、送信関連のステータスはクリアされます。
	メール送信完了 [MKY43_MAIL_STATUS_SENDEND] は送信完了すると 1 が読込めます。
	エラーによる送信中断の場合、値は不定です。
	メールを着信すると、[MKY43_MAIL_STATUS_RECEIVE] は 1 が読込めます。

CN_MailWaitObject 関数

機能 メール送信完了、メール受信、メールクローズのいずれかが発生したときに制御を戻します

書式 int CN_MailWaitObject(int Board, int Line, unsigned short *Status)

引数

Board	: ボード番号 (0 のみ)
Line	: ライン番号 (0 のみ)
Status	: 送信結果ステータスを格納するポインタ

* Status の詳細は、CN_MailStatus 関数の項を参照して下さい。

戻り値

MKY43_ER_OK	: 正常
MKY43_ER_INVALIDPARAM	: 無効な引数です
MKY43_ER_NOTOPEN	: オープンされていません
MKY43_ER_UNCONNECT	: CUnet にコネクしていません
MKY43_ER_MAILSENDRESULT	: メール送信に結果を取得しました
MKY43_ER_MAILRECEIVE	: メール受信しました
MKY43_ER_MAILCLOSE	: メールのクローズが発行されました
上記以外	: エラー

説明

メール送信完了、メール受信をしたときに制御を戻します。
メールをクローズした場合も制御を戻します。
メール送信結果取得の時のみ Status にデータが入ります。

シグナル状態になるまでの間は待機状態に入りますので、この関数はスレッド内部でコールして下さい。

2-4 レジスタアクセス

CN_GetReg64 関数

機能	64 ビットアクセスでレジスタを読みみます		
書式	<code>int CN_GetReg64(int Board, int Line, unsigned short OffsetAdr, ULONGLONG *pReg)</code>		
引数	Board	: ボード番号 (0 のみ)	
	Line	: ライン番号 (0 のみ)	
	OffsetAdr	: レジスタのアドレス	
	pReg	: レジスタを格納するポインタ	
戻り値	MKY43_ER_OK	: 正常	
	MKY43_ER_INVALIDPARAM	: 無効な引数です	
	MKY43_ER_NOTOPEN	: オープンされていません	
	MKY43_ER_UNCONNECT	: CUnet にコネクしていません	
説明	OffsetAdr で指定したレジスタを 64 ビットアクセスで読みみます。		

CN_GetReg32 関数

機能 32 ビットアクセスでレジスタを読み込みます

書式 int CN_GetReg32(int Board, int Line, unsigned short OffsetAdr, unsigned long *pReg)

引数

Board	: ボード番号 (0 のみ)
Line	: ライン番号 (0 のみ)
OffsetAdr	: レジスタのアドレス
pReg	: レジスタを格納するポインタ

戻り値

MKY43_ER_OK	: 正常
MKY43_ER_INVALIDPARAM	: 無効な引数です
MKY43_ER_NOTOPEN	: オープンされていません
MKY43_ER_UNCONNECT	: CUnet にコネクしていません

説明 OffsetAdr で指定したレジスタを 32 ビットアクセスで読み込みます。

CN_GetReg16 関数

機能 16 ビットアクセスでレジスタを読みみます

書式 int CN_GetReg16(int Board, int Line, unsigned short OffsetAdr, unsigned short *pReg)

引数

Board	: ボード番号 (0 のみ)
Line	: ライン番号 (0 のみ)
OffsetAdr	: レジスタのアドレス
pReg	: レジスタを格納するポインタ

戻り値

MKY43_ER_OK	: 正常
MKY43_ER_INVALIDPARAM	: 無効な引数です
MKY43_ER_NOTOPEN	: オープンされていません
MKY43_ER_UNCONNECT	: CUnet にコネクしていません

説明 OffsetAdr で指定したレジスタを 16 ビットアクセスで読みみます。

CN_SetReg64 関数

機能 64 ビットアクセスでレジスタを書込みます

書式 int CN_SetReg64(int Board, int Line, unsigned short OffsetAdr, ULONGLONG Data)

引数

Board	: ボード番号 (0 のみ)
Line	: ライン番号 (0 のみ)
OffsetAdr	: レジスタのアドレス
Data	: レジスタ

戻り値

MKY43_ER_OK	: 正常
MKY43_ER_INVALIDPARAM	: 無効な引数です
MKY43_ER_NOTOPEN	: オープンされていません
MKY43_ER_UNCONNECT	: CUnet にコネクしていません

説明 OffsetAdr で指定したレジスタに 64 ビットアクセスで書込みます。

CN_SetReg32 関数

機能 32 ビットアクセスでレジスタを書込みます

書式 int CN_SetReg32(int Board, int Line, unsigned short OffsetAdr, unsigned long Data)

引数

Board	: ボード番号 (0 のみ)
Line	: ライン番号 (0 のみ)
OffsetAdr	: レジスタのアドレス
Data	: レジスタ

戻り値

MKY43_ER_OK	: 正常
MKY43_ER_INVALIDPARAM	: 無効な引数です
MKY43_ER_NOTOPEN	: オープンされていません
MKY43_ER_UNCONNECT	: CUnet にコネクしていません
CN_ER_COMMANDTIMEOUT	: タイムアウトで終了しました

説明 OffsetAdr で指定したレジスタに 32 ビットアクセスで書込みます。

CN_SetReg16 関数

機能 16 ビットアクセスでレジスタを書込みます

書式 int CN_SetReg16(int Board, int Line, unsigned short OffsetAdr, unsigned short Data)

引数

Board	: ボード番号 (0 のみ)
Line	: ライン番号 (0 のみ)
OffsetAdr	: レジスタのアドレス
Data	: レジスタ

戻り値

MKY43_ER_OK	: 正常
MKY43_ER_INVALIDPARAM	: 無効な引数です
MKY43_ER_NOTOPEN	: オープンされていません
MKY43_ER_UNCONNECT	: CUnet にコネクしていません

説明 OffsetAdr で指定したレジスタに 16 ビットアクセスで書込みます。

2-5 GM アクセス

CN_GetMemShort 関数

機能 16ビットアクセスでGMからデータを読み込みます。

書式 `int CN_GetMemShort(int Board, int Line, unsigned short *pBuf, short StationAdr, short StationOffsetAdr, unsigned short ShortSize)`

引数

Board	: ボード番号 (0のみ)
Line	: ライン番号 (0のみ)
pBuf	: 16ビットデータを格納するポインタ
StationAdr	: ステーションアドレス
StationOffsetAdr	: オフセットアドレス
ShortSize	: 取得するデータのサイズ (2バイト単位)

戻り値

MKY43_ER_OK	: 正常
MKY43_ER_INVALIDPARAM	: 無効な引数です
MKY43_ER_NOTOPEN	: オープンされていません
MKY43_ER_OTHER	: その他のエラーです

説明 StationAdr で指定したステーションアドレスのGMの StationOffsetAdr で指定したオフセットから、ShortSize で指定したサイズ (2バイト単位) のデータを 16ビットアクセスで読み込みます。

CN_GetMemLong 関数

機能

32 ビットアクセスで GM からデータを読み込みます

書式

```
int CN_GetMemLong(int Board, int Line, unsigned long *pBuf, short StationAdr,  
                  short StationOffsetAdr, unsigned short LongSize)
```

引数

Board : ボード番号 (0 のみ)
Line : ライン番号 (0 のみ)
pBuf : 32 ビットデータを格納するポインタ
StationAdr : ステーションアドレス
StationOffsetAdr : オフセットアドレス
LongSize : 取得するデータのサイズ (4 バイト単位)

戻り値

MKY43_ER_OK : 正常
MKY43_ER_INVALIDPARAM : 無効な引数です
MKY43_ER_NOTOPEN : オープンされていません
MKY43_ER_OTHER : その他のエラーです

説明

StationAdr で指定したステーションアドレスの GM の StationOffsetAdr で指定したオフセットから、LongSize で指定したサイズ (4 バイト単位) のデータを 32 ビットアクセスで読み込みます。

CN_GetMemDLong 関数

機能

64 ビットアクセスで GM からデータを読み込みます

書式

```
int CN_GetMemDLong(int Board, int Line, ULONGLONG *pBuf, short StationAdr,  
                   short StationOffsetAdr, unsigned short DLongSize)
```

引数

Board : ボード番号 (0 のみ)
Line : ライン番号 (0 のみ)
pBuf : 64 ビットデータを格納するポインタ
StationAdr : ステーションアドレス
StationOffsetAdr : オフセットアドレス
DLongSize : 取得するデータのサイズ (8 バイト単位)

戻り値

MKY43_ER_OK : 正常
MKY43_ER_INVALIDPARAM : 無効な引数
MKY43_ER_NOTOPEN : オープンされていない
MKY43_ER_OTHER : その他のエラー

説明

StationAdr で指定したステーションアドレスの GM の StationOffsetAdr で指定したオフセットから、DLongSize で指定したサイズ (8 バイト単位) のデータを 64 ビットアクセスで読み込みます。

CN_SetMemShort 関数

機能 16 ビットアクセスで GM にデータを書込みます

書式 `int CN_SetMemShort(int Board, int Line, unsigned short *pBuf, short StationAdr, short StationOffsetAdr, unsigned short ShortSize)`

引数

Board	: ボード番号 (0 のみ)
Line	: ライン番号 (0 のみ)
pBuf	: 16 ビットデータのポインタ
StationAdr	: ステーションアドレス
StationOffsetAdr	: オフセットアドレス
ShortSize	: 書き込むデータのサイズ (2 バイト単位)

戻り値

MKY43_ER_OK	: 正常
MKY43_ER_INVALIDPARAM	: 無効な引数です
MKY43_ER_NOTOPEN	: オープンされていません
MKY43_ER_OTHER	: その他のエラーです

説明 StationAdr で指定したステーションアドレスの GM の StationOffsetAdr で指定したオフセットから、ShortSize で指定したサイズ (2 バイト単位) のデータを 16 ビットアクセスで書き込みます。

CN_SetMemLong 関数

機能

32 ビットアクセスで GM にデータを書込みます

書式

```
int CN_SetMemLong(int Board, int Line, unsigned long *pBuf, short StationAdr,  
                  short StationOffsetAdr, unsigned short LongSize)
```

引数

Board : ボード番号 (0 のみ)
Line : ライン番号 (0 のみ)
pBuf : 32 ビットデータのポインタ
StationAdr : ステーションアドレス
StationOffsetAdr : オフセットアドレス
LongSize : 書き込むデータのサイズ (4 バイト単位)

戻り値

MKY43_ER_OK : 正常
MKY43_ER_INVALIDPARAM : 無効な引数です
MKY43_ER_NOTOPEN : オープンされていません
MKY43_ER_OTHER : その他のエラーです

説明

StationAdr で指定したステーションアドレスの GM の StationOffsetAdr で指定したオフセットから、LongSize で指定したサイズ (4 バイト単位) のデータを 32 ビットアクセスで書き込みます。

CN_SetMemDLong 関数

機能

64 ビットアクセスで GM にデータを書込みます

書式

```
int CN_SetMemDLong(int Board, int Line, ULONGLONG *pBuf, short StationAdr,  
                   short StationOffsetAdr, unsigned short DLongSize)
```

引数

Board : ボード番号 (0 のみ)
Line : ライン番号 (0 のみ)
pBuf : 64 ビットデータのポインタ
StationAdr : ステーションアドレス
StationOffsetAdr : オフセットアドレス
DLongSize : 書き込むデータのサイズ (8 バイト単位)

戻り値

MKY43_ER_OK : 正常
MKY43_ER_INVALIDPARAM : 無効な引数です
MKY43_ER_NOTOPEN : オープンされていません
MKY43_ER_OTHER : その他のエラーです

説明

StationAdr で指定したステーションアドレスの GM の StationOffsetAdr で指定したオフセットから、DLongSize で指定したサイズ (8 バイト単位) のデータを 64 ビットアクセスで書き込みます。

このリファレンスマニュアルについて

- (1) 本書の内容の一部または全部を当社からの事前の承諾を得ることなく、無断で複写、複製、掲載することは固くお断りします。
- (2) 本書の内容に関しては、製品改良のためお断りなく、仕様などを変更することがありますのでご了承下さい。
- (3) 本書の内容に関しては万全を期しておりますが、万一ご不審な点や誤りなどお気づきのことがございましたらお手数ですが巻末記載の弊社もしくは、営業所までご連絡下さい。その際、巻末記載の書籍番号も併せてお知らせ下さい。