FL-net PCI カード（100Mbps 版）DLL 仕様書

変更履歴

REV	日付	内容
1.00	06.06.25	新規作成
1.01	07.04.17	P23、P49 「0：通信無効検知あり、1：なし」→ 「0：なし、1：通信無効検知あり」へ修正しました。
1.02	08.05.20	P21、P47 ReadMessage の FL-PCI カード側の受信メッセージキューの件数を 32→16 に修正しました。
1.03	10.12.07	P6、13、17、38、42 WriteCyclicAll→WriteCyclicALL（末尾の l2 つを大文字の L に修正しました。） WriteCyclic_All→WriteCyclic_ALL（〃） ReadCyclicAll→ReadCyclicALL（〃） ReadCyclic_All→ReadCyclic_ALL（〃） P13、P14 WriteCyclicALL の注意事項を注意事項(1) (2)に変更しました。 WriteCyclicRelative の注意事項を注意事項(1)(2)に変更しました。 P38、P39 WriteCyclic_ALL の注意事項を注意事項(1) (2)に変更しました。 WriteCyclic_Rel の注意事項を注意事項(1)(2)に変更しました。 P15、P16、P17 ReadCyclicAbsolute の注意事項を注意事項(1) (2)として追記しました。 ReadCyclicRelative の注意事項を注意事項(1) (2)として追記しました。 ReadCyclicALL の注意事項を注意事項(1) (2)として追記しました。 P40、P41、P42 ReadCyclic_Abs の注意事項を注意事項(1) (2)として追記しました。 ReadCyclic_Rel の注意事項を注意事項(1) (2)として追記しました。 ReadCyclic_ALL の注意事項を注意事項(1) (2)として追記しました。
1.04	11.03.31	P9 2.4 インストールの方法 を追記しました。

REV	日付	内容
1.05	11.09.01	P3 関数名の誤記を訂正しました。(小文字→大文字) ×WriteCyclic_All ○WriteCyclic_ALL P9 「#pragma comment の行は、#include “stdafx.h”の後に置いてください。」を追記しました。 P14, P39 ×comnon ○common P19～20 WriteMessage 関数 返り値の説明を詳細にしました。
	15.01.23	P10 DLL(flpci_mul.dll)の配置場所を、WOW64 システムが実装されている場合と、 実装されていない場合に別けて記述。
1.06	15.06.04	P11 WOW64 システムが実装されている場合とは、Windows7(64bit OS) / Windows8 の場合であることを明示。 Visual C++ ランタイムライブラリが必要なことを記述。 P12 2.5 Visual C++ ランタイムについて を追記。
1.07	15.11.24	P27, 52 wait_flg を 1(受信完了を待つ)として ReadMessage(C 言語および VB の I/F)が呼ばれた場合、受信完了するまで無限に待つわけではなく、2 秒のタイムアウトがあることを明記。 P10-14 64bit DLL について記述。

REV	日付	内容
1.08	16.08.30	P38　3.20 Firmware_Version P64　4.20 Firmware_Version バイナリデータの誤記を訂正しました。 Bit8～Bit15:　×10h = FL-PCI/V2 (FL-net Ver2) ○　0Ah = FL-PCI/V2 (FL-net Ver2) ×20h= FL-PCI/V2-100 または 100L (FL-net Ver2) ○14h= FL-PCI/V2-100 または 100L (FL-net Ver2)
1.09	2022.03.03	P16　FL_initialize で設定する最小許容フレーム間隔の値の範囲に関し 注意事項２を追記しました。
1.10	2022.06.01	P38　FirmwareVersion によって得られるバイナリデータに下記を追記 Bit 8 ～ Bit15　1Eh= FL-PCIe または FL-PCI/V2 100L(FL-net Ver3) P64　FirmwareVersion によって得られるバイナリデータに下記を追記 Bit 8 ～ Bit15　1Eh= FL-PCIe または FL-PCI/V2 100L(FL-net Ver3)

目次

1. 概要

本書は、FL-PCI カードを装着した Windows システムにおいて、DLL（ダイナミック・リンク・ライブラリ）を用いてアプリケーションから FL-PCI カードを制御するためのインターフェースについて記載します。

本書は、マルチプロセス対応版（flpci_mul.dll）について記載します。

2. DLL

2.1. 関数一覧

C（C++）と Visual Basic は、同じ DLL をコールしますが、一部別仕様になっています。

関数名	機能	C	VB
FL_Initialize	FL-PCI カードを初期化します。	○	○
WriteCyclicALL	自ノードの全領域のデータを一括書き込みします。	○	
WriteCyclic_ALL	自ノードの全領域のデータを一括書き込みします。		○
WriteCyclicRelative	自ノードのコモンメモリ 1 または 2 の指定オフセット位置へ指定データ数書き込みます。	○	
WriteCyclic_Rel	自ノードのコモンメモリ 1 または 2 の指定オフセット位置へ指定データ数ライトします。		○
ReadCyclicAbsolute	コモンメモリ 1 または 2 の先頭からのオフセットとサイズを指定して、指定バッファへ読み込みます。	○	
ReadCyclic_Abs	コモンメモリ 1 または 2 の先頭からのオフセットとサイズを指定して、指定バッファへ読み込みます。		○
ReadCyclicALL	指定ノードのコモン 1､コモン 2 領域の全データを読みます。	○	
ReadCyclic_ALL	指定ノードのコモン 1､コモン 2 領域の全データを読みます。		○
ReadCyclicRelative	指定ノードのコモンメモリ 1 または 2 の領域先頭からのオフセットとサイズを指定して指定バッファへ読込みます。	○	
ReadCyclic_Rel	指定ノードのコモンメモリ 1 または 2 の領域先頭からのオフセットとサイズを指定して指定バッファへ読込みます		○
WriteMessage	引数で指定されたメッセージを FL-PCI カードへ渡します。送信終了を待つかどうかの指定が可能です。	○	○
CheckMessageEnd	WriteMessage で送信終了を待たなかった場合、処理完了の状態を返します。	○	○
ReadMessage	メッセージの受信情報を返します。 受信終了を待つかどうかの指定が可能です。	○	○
GetNodeInfo	指定されたノード情報を返します。	○	○
GetRingInfo	現在の FL-net に参加しているノードＮｏを返します。	○	○
GetNetInfo	現在のリフレッシュサイクル測定値を返します。	○	○

関数名	機能	C	VB
GetLogInfo	現在の FL-net ログ情報を返します。	○	○
SetUplStatus	上位層ステータスの設定を行います。	○	○
GetErrInfo	重大エラーが発生した時の詳細情報を返します。	○	○
DeviceOpen	デバイスをオープンします。	○	○
DeviceClose	デバイスをクローズして、FL-PCI カードをリセット状態にします。	○	○
ReadUls_All	ノード No.1～254 の ULS（上位層の状態）と LKS（リンクの状態）を返します。	○	○
Dll_Version	DLL のバージョン No.を返します。	○	○
Firmware_Version	FL-PCI ファームウェアのバージョン No.を返します。	○	○
Set_Speed	イーサネットの通信速度を指定します。	○	○
Get_Speed	イーサネットの通信速度を取得します。	○	○

2.2. エラーコード

DLL の戻り値とその対応方法は以下のとおりです。

Code	エラー種別	処理
2	パラメータエラー	引数に誤りがあります。再確認して下さい。
3	バッファフル	FL-PCI カード側のバッファがフルなのでリトライして下さい。
4	データなし	指定の他ノードからサイクリックデータを受信していないため、再度読み出しして下さい。
5	ノード離脱	指定ノードは離脱しています。
6	送信失敗	メッセージ送信を失敗しました。 自ノードのメッセージが正しいか確認して下さい。 相手ノードが離脱していないか確認して下さい。
7	送信未完	メッセージの送信は未完のため再度送信完了を確認してください。
8	タイムアウト	PCI カードとの通信タイムアウトです。 ドライバーのインストールが未完の場合に発生します。
9	重大エラー	処理の続行は不可能です。弊社へご連絡ください。
10	デバイスオープンエラー1	ドライバーのインストールが未完の場合に発生します。
11	デバイスオープンエラー2	ドライバーのインストールが未完の場合に発生します。
12	デバイスオープンエラー3	ドライバーのインストールが未完の場合に発生します。
14	デバイス未オープン	FL-PCI カードの初期化が未完です。 FL_Initialize()を実行して下さい。
15	デバイスクローズエラー	デバイスクローズに失敗しました。 アプリケーションをいったん終了して再起動をしてください。
16	デバイス 2 重オープン	すでにオープンしているデバイスに対して、再度オープンを要求しました。
17	デバイス未初期化	デバイスオープン関数が、未初期化のデバイスに対して実行されました。
18	デバイス BUSY	デバイスオープン中の FL-PCI へ初期化を行おうとしました。 1 カードの 2 デバイスをクローズさせて初期化を行ってください。
19	mutex エラー	DLL 内部で mutex の取得、または排他制御時にエラーが発生しました。

2.3. マルチプロセス／マルチスレッド

（1）マルチプロセス、マルチスレッド環境をサポートします。

（2）ご使用できるデバイス No.は次の通りです。

カード 1 枚目は、No.0 と 1

カード 2 枚目は、No.2 と 3

デバイス No.は、アプリケーション・プロセスが FL-PCI をアクセスする時に必要なオブジェクトハンドルを取得するために使用します。

FL-PCI 側のファームウェアでは、アプリケーションからの要求に対して、デバイス No.の管理・判別を行うことはありません。

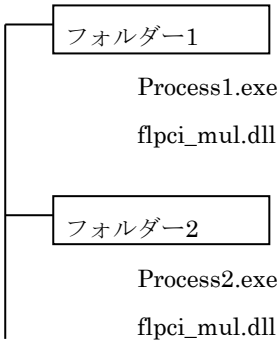
（3）DLL 起動環境の注意

DLL 関数は FL-PCI カードとの通信にデュアルポート RAM を使用するため、DLL 内部では Mutex による排他処理を行います。

以下の環境の場合には、排他制御が有効に機能しないためご注意をお願いします。

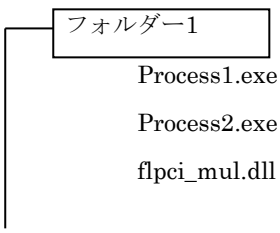
＜正常に動作しない構成＞

以下の場合、各 DLL が個別に起動され、固有の Mutex を取得するため、排他制御が有効に機能しません。



＜正常に動作する構成＞

flpci_mul.dll を 1 個所に配置します。（実行ファイルと異なるフォルダーに置くことも可能です）



flpci_mul.dll は、Windows のシステムフォルダに置くことができます。この場合、各アプリケーションのディレクトリに別途 flpci_mul.dll を置く必要はありません。

使用する OS とアプリケーション	システムフォルダ	インストールする DLL
32bit OS の場合	C:¥Windows¥System32	32bit DLL
64bit OS で、32 ビットアプリケーションを使用する場合	C:¥Windows¥SysWOW64	32bit DLL
64bit OS で、64 ビットアプリケーションを使用する場合	C:¥Windows¥System32	64bit DLL

64bit OS で、32bit DLL と 64bit DLL の両方をインストールして使用することも可能です。ただし、双方の間での排他制御は行えません。

(4) 初期設定

FL-PCI カード 1 枚へ 1 度に割り当てることのできる初期設定値は 1 種類のみです。

FL-PCI カード 1 枚につき 2 デバイス割り当て可能ですが、デバイスごとに初期設定値を指定することはできません。

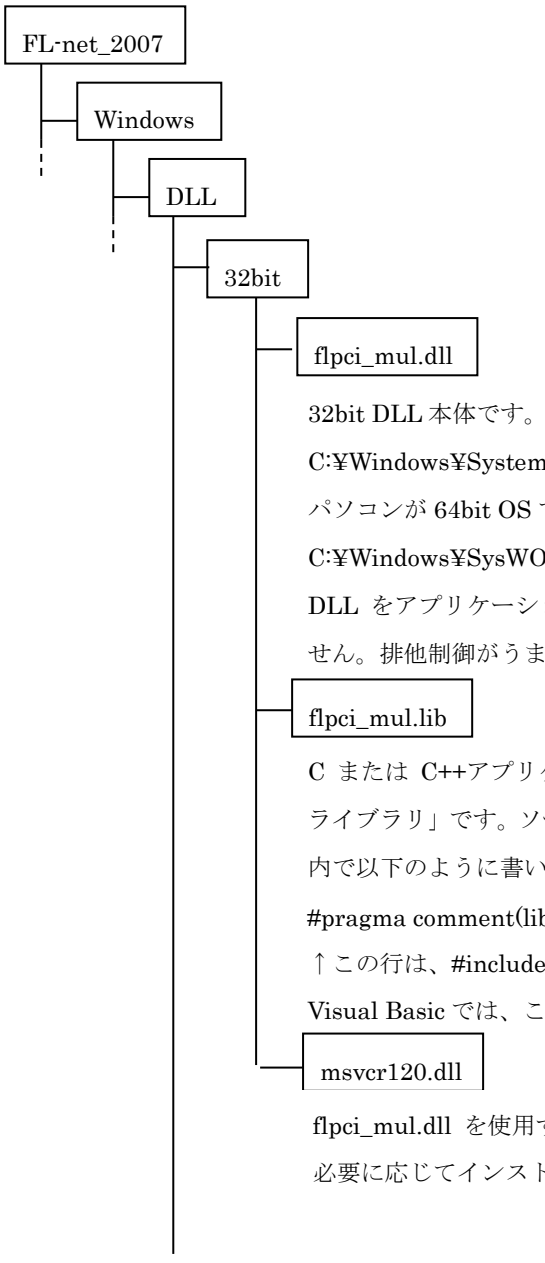
(5) プロセス終了時の注意

FL_Initialize()、DeviceOpen()の後、使用しているデバイス No.に対する DeviceClose()を行わないでプロセス終了をした場合（異常終了、強制終了を含む）、その後に実行する FL_Initialize()、DeviceOpen()の正常処理が保証されない場合があります。

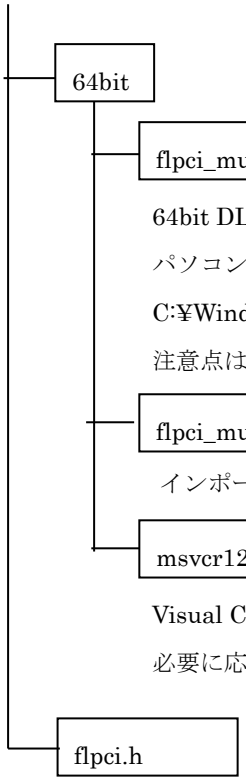
2.4. インストール方法

本製品に付属している CD-ROM の内容は、以下の通りです。

本製品を使用する際には、Visual C++ ランタイムライブラリが必要です。ランタイムライブラリの確認方法・インストール方法については、2.5 章を参照してください。



(全ページから続く)



64bit DLL 本体です。
パソコンが 64bit OS で、64bit アプリケーションを使用する場合、このファイルを C:¥Windows¥System32 ディレクトリにコピーしてください。
注意点は 32bitDLL の場合と同様です。

インポートライブラリです。使用方法は 32bit のインポートライブラリと同様です。

Visual C++ ランタイムライブラリです。
必要に応じてインストールしてください。(2.5 章参照)

C または C++アプリケーションをお客様が作成するときに#include すると便利な「ヘッダファイル」です。(必須ではありません)
Visual Basic では、このファイルは使いません。

2.5. Visual C++ ランタイムライブラリについて

本製品の DLL を使用する際には、Visual C++ ランタイムライブラリのうち、 msvcr120.dll が必要です。

この DLL は、Windows の初期状態にはインストールされていませんが、パソコンによってはインストールされた状態で出荷されている場合もあります。

インストールされているかどうかは、以下のディレクトリを参照して確認してください。

32bit OS の場合	C:¥Windows¥System32
64bit OS で、32 ビットアプリケーションを使用する場合	C:¥Windows¥SysWOW64
64bit OS で、64 ビットアプリケーションを使用する場合	C:¥Windows¥System32

インストールされていない場合、以下のいずれかの方法でインストールしてください。

1) Microsoft 社のサイトから、Visual C++ 再頒布可能パッケージを取得し、インストールしてください。

   https://www.microsoft.com/ja-jp/download/details.aspx?id=40784

   この際、以下の表を参考にしてファイルを取得してください。

32bit OS の場合	vcredist_x86.exe
64bit OS で、32 ビットアプリケーションを使用する場合	vcredist_x86.exe
64bit OS で、64 ビットアプリケーションを使用する場合	vcredist_x64.exe

2) ソフトウェアパッケージの flpci_mul.dll と同じ場所にある msvcr120.dll を、前述のディレクトリにコピーしてください。

3. C 言語 I/F 仕様

3.1. FL_Initialize

機能：デバイスをオープンし、FL-PCI カードを初期化します。

記述：unsigned short rc = FL_Initialize(&init_prm);

struct InitInfo {

unsigned long	ip_adr;	： 自ノード IP アドレス。下 1 バイトは、自ノード No.とする。※
unsigned short	dev_no;	： デバイス No.　ボード 1=0 or 1　ボード 2=2 or 3
unsigned short	c1_adr;	： コモンメモリ 領域 1 アドレス（0～511）
unsigned short	c1_size;	： コモンメモリ 領域 1 サイズ（0～512）
unsigned short	c2_adr;	： コモンメモリ 領域 2 アドレス（0～8191）
unsigned short	c2_size;	： コモンメモリ 領域 2 サイズ（0～8192）
unsigned char	tw;	： トークン監視タイムアウト値(1～255)
unsigned char	mft;	： 最小許容フレーム間隔（0～50）
unsigned char	vender[11];	： ベンダ名。”CENTURYSYS” 固定 (11Byte 目は NULL)
unsigned char	maker[11];	： メーカー型式。”S-0616___” 固定 (11Byte 目は NULL)
unsigned char	name[11];	： ノード名。(10 文字以内で任意　11Byte 目は NULL)

} init_prm;

unsigned short rc;　:

0=正常終了

2=パラメータエラー

旧 FL-PCI /V2 へ 100Mbps、オートネゴシエーションを指定した場合にも通知されます。

8=タイムアウト（最大 4 秒）

9=重大エラー（詳細は P34を参照）

10=デバイスオープンエラー（ファイルクリエート）

11=デバイスオープンエラー（デバイス取得）

12=デバイスオープンエラー（ポインタ取得）

16=デバイスの 2 重オープン

18=一方のデバイスがオープン中に初期化を行った

19=mutex 取得エラー

※ リトル・エンディアンで渡してください。つまり、inet_addr(“192.168.250.1”)は誤りです。0xc0a8fa01 などとしてください。

注意事項１：初期化の後再初期化を行う場合は、一旦 関数 DeviceClose()を使って 2 デバイスをクローズしてください。
デバイスクローズを行わないで再初期化を行うと、デバイスオープンエラーが発生します。

注意事項２：FA リンクプロトコル仕様書には最小許容フレーム間隔は、１００マイクロ秒単位で 0 から５０とすると書かれているので 0 から５０の範囲で設定することに仕様上の問題はありません。
しかし、実装規定書には下記の目標性能が明記されていますので、0 から１２の範囲で、設定するようにお願いいたします。
目標性能
ノードのコモンメモリのアサインを「領域１：４ワード」と「領域２：６４ワード」にしたとき、ノードがトークンを受けてから、次のノードにトークンを送信するまでの時間は概算で 1.5ms 以内。

3.2. DeviceOpen

機能： 指定デバイスをオープンします。
FL-PCI カードは 1 カードで 2 デバイスが使用可能です。
片側のデバイスが FL_Initialize で FL-PCI カードを初期化した後に、この関数を呼び出してもう 1 方のデバイスをオープンして下さい。
初期設定前にこの関数を呼び出した場合には、エラー終了します。

記述： rc = DeviceOpen(dev_no);
unsigned short dev_no; ： デバイス No.（0～3）

unsigned short rc; ：
0=正常終了
2=パラメータエラー
8=タイムアウト（最大 4 秒）
9=重大エラー（詳細は P34を参照）
10=デバイスオープンエラー（ファイルクリエート）
11=デバイスオープンエラー（デバイス取得）
12=デバイスオープンエラー（ポインタ取得）
16=デバイスの 2 重オープン
17=デバイスは未初期化
19=mutex 取得エラー

注意事項：

3.3. WriteCyclicALL

機能： 初期設定時に設定した自ノードコモンメモリー領域へ、自ノードのサイクリックデータを書き込みます。
引数のコモン領域 1、2 のデータバッファから、デュアルポート RAM へ一括コピーします。
ノード No.、領域情報は、初期化関数の引数から自動認識します。

記述： rc = WriteCyclicALL(&buf1, &buf2, dev_no);

```
unsigned short      *buf1;     : コモンメモリーバッファ 1 アドレス
unsigned short      *buf2;     : コモンメモリーバッファ 2 アドレス
unsigned short      dev_no;    : デバイス No.（0～3）

unsigned short      rc;  :
  0=正常終了
  2=パラメータエラー
  3=バッファフル
  14=デバイスはオープンされていない
  19=mutex 排他制御エラー
```

注意事項:

(1) 本関数によってデュアルポート RAM にサイクリックデータが書き込まれると、FL-PCI カードにおいて、自ノードサイクリックデータ送信バッファの更新処理が発生します。この処理は FL-PCI カードの CPU に負荷をかけることになり、タイムロスになりますので、サイクリックデータ変更が発生したときにのみ本関数をコールして下さい。

(2) 自ノードのサイクリックデータは、FL-net プロトコルの規定で、トークンが自分にまわってきたときにネットワークに送出されます。このため、本関数をリフレッシュサイクル（トークン 1 周）より短い周期でコールした場合や、離脱時にコールした場合、戻り値 3（バッファフル）が返されることがあります。これはワーニングエラーですので、少し待った後リトライを行うと正常終了します。

3.4. WriteCyclicRelative

機能： 自ノードが占有する領域先頭をオフセット 0 とする相対アドレスに対して、書き込みを行います。
領域内のワードオフセットからワードサイズのサイクリックデータを、指定バッファの先頭から指定ワード分書き込みます。
ノード No.、領域情報は、初期化関数の引数から自動認識します。

記述： rc = WriteCyclicRelative(common, offset, size, &buff, dev_no);

unsigned short common; : 1=コモン領域 1、2=コモン領域 2
unsigned short offset; : ワードオフセット（0～511 または 0～8191）
unsigned short size; : ワードサイズ（1～512 または 1～8192）
unsigned short *buff; : バッファアドレス
unsigned short dev_no; : デバイス No.（0～3）

unsigned short rc; :
0=正常終了
2=パラメータエラー
3=バッファフル
14=デバイスはオープンされていない
19=mutex 排他制御エラー

注意事項:

(1) 本関数によってデュアルポート RAM にサイクリックデータが書き込まれると、FL-PCI カードにおいて、自ノードサイクリックデータ送信バッファの更新処理が発生します。この処理は FL-PCI カードの CPU に負荷をかけることになり、タイムロスになりますので、サイクリックデータ変更が発生したときにのみ本関数をコールして下さい。

(2) 自ノードのサイクリックデータは、FL-net プロトコルの規定で、トークンが自分にまわってきたときにネットワークに送出されます。 このため、本関数をリフレッシュサイクル（トークン 1 周）より短い周期でコールした場合や、離脱時にコールした場合、戻り値 3（バッファフル）が返されることがあります。これはワーニングエラーですので、少し待った後リトライを行うと正常終了します。

3.5. ReadCyclicAbsolute

機能： コモンメモリ 1 または 2 の先頭からのワードオフセット、ワードサイズを指定して、サイクリックデータを指定バッファへ読み出します。
この関数では、該当するコモン領域に割り当てられているノードがあれば、そのノードの最新のデータを読み出します。
読み出し領域が複数のノードの領域にまたがる場合、それらのデータを合成してデータを読み出します。
どのノードの送信領域にもなっていない領域には、ゼロが書き込まれます。

記述： rc = ReadCyclicAbsolute(common, offset, size, &buff, dev_no);

unsigned short	common;	: 1=コモン領域 1、2=コモン領域 2
unsigned short	offset;	: ワードオフセット（0～511 または 0～8191）
unsigned short	size;	: ワードサイズ（1～512 または 1～8192）
unsigned short	*buff;	: バッファアドレス
unsigned short	dev_no;	: デバイス No.（0～3）

unsigned short rc; :

0=正常終了

2=パラメータエラー

14=デバイスはオープンされていない

19=mutex 排他制御エラー

注意事項：

(1) 本関数は指定されたコモン領域の最新のサイクリックデータを読み出すものです。
ノードのサイクリックデータはリフレッシュサイクル（トークン一周）で更新されますので、サイクリックデータの変化を完全に把握したい場合は、本関数をリフレッシュサイクルより短い周期でコールして下さい。

(2) 本関数は、指定されたコモン領域に離脱中のノードのコモン領域が含まれる場合、そのサイクリックデータはゼロをセットします。

3.6. ReadCyclicRelative

機能： コモンメモリで指定ノードが占有する領域先頭をオフセット 0 とする相対
アドレスを用います。
領域内のワードオフセットからワードサイズのサイクリックデータをバッファ
へ読み込みます。
指定されたノードが FL-net に参加していなければ、エラー（ノード離脱）を返します。

記述： rc = ReadCyclicRelative(node, common, offset, size, &buff, dev_no);

unsigned short node ： ノード No.（1～254）
unsigned short common; ： 1=コモン領域 1、2=コモン領域 2
unsigned short offset; ： ワードオフセット（0～511 または 0～8191）
unsigned short size; ： ワードサイズ（1～512 または 1～8192）
unsigned short *buff; ： バッファアドレス
unsigned short dev_no; ： デバイス No.（0～3）

unsigned short rc; ：
0=正常終了
2=パラメータエラー
4=データなし
5=ノード離脱
14=デバイスはオープンされていない
19=mutex 排他制御エラー

注意事項：

(1) 本関数は指定ノードの指定領域の最新のサイクリックデータを読み出すものです。
ノードのサイクリックデータはリフレッシュサイクル（トークン一周）で更新されますので、サイクリックデータの変化を完全に把握したい場合は、本関数をリフレッシュサイクルより短い周期でコールして下さい。

(2) 本関数はコモン領域を持たないノードには使用できません。

3.7. ReadCyclicALL

機能： 指定ノードのコモンメモリ領域から指定ノードのコモン 1、2 の全サイクリックデータを読み出します。
引数のコモン領域 1、2 のデータバッファへデュアルポート RAM から一括コピーします。
指定されたノードが FL-net に参加していなければ、エラー（ノード離脱）を返します。

記述： rc = ReadCyclicALL(node, &buf1, &buf2 , dev_no);

```
unsigned short      node;     : ノード No.（1～254）
unsigned short      *buf1;    : コモンメモリーバッファ 1 アドレス
unsigned short      *buf2;    : コモンメモリーバッファ 2 アドレス
unsigned short      dev_no;   : デバイス No.（0～3）

unsigned short      rc;  :
            0=正常終了
             2=パラメータエラー
             4=データなし
             5=ノード離脱
            14=デバイスはオープンされていない
            19=mutex 排他制御エラー
```

注意事項：

(1) 本関数は指定されたノードの最新のサイクリックデータを読み出すものです。
ノードのサイクリックデータはリフレッシュサイクル（トークン一周）で更新されますので、サイクリックデータの変化を完全に把握したい場合は、本関数をリフレッシュサイクルより短い周期でコールして下さい。

3.8. WriteMessage

機能： 構造体で設定されたメッセージを送ります。

FL-net ではメッセージ送信の条件が揃うまで送信を待ちます。

また、1：1 メッセージ送信の場合、相手ノードからの正常終了のＡＣＫを受信するまで 3 回のリトライが行われることがあります。

そのため送信の登録から送信完了まで時間がかかることがあり、引数の wait_flg で、送信完了を待って関数を return するか、それとも完了を待たずに return するかを選択します。

完了を待たない選択をした場合、CheckMessageEnd 関数を使ってメッセージ送信が正常終了したかをチェックする必要があります。

記述： rc = WriteMessage(&msnd_prm, &buff, wait_flg, dev_no);

```
struct MessageInfo {
        unsigned char     msg_node        ：相手先ノード No.
                                          （1～255、 255 はブロードキャスト指定）
        unsigned char     msg_result;     ： 結果コード（0～2）
        unsigned long     msg_tcd;        ： トランザクションコード
        unsigned short    msg_bc;         ： バイトサイズ（0～1024）
        unsigned long     msg_adr;        ： 仮想アドレス空間 アドレス
        unsigned short    msg_lng;        ： 仮想アドレス空間 データ長
} msnd_prm;
unsigned char     *buff;    ： バッファアドレス
unsigned short    wait_flg  ： 0= 送信完了を待たない、1=送信完了を待つ
unsigned short    dev_no;   ： デバイス No.（0～3）

unsigned short  rc; :
        0=正常終了
        2=パラメータエラー
        3=バッファフル
        5=自ノード離脱
            WriteMessage 関数が呼び出された時点で、自ノードが離脱している場合、リターン値 5 が返されます。
        6=送信失敗
            WriteMessage 関数が wait_flg=1 で呼び出された時点で、自ノードは FL-net に参加しているが、メッセージ送信先ノードが離脱している場合、あるいはメッセージ送信先ノードから ACK が返されなかった場合、リターン値 6 が返されます。（メッセージ送信先ノードが離脱している場合には、即座にリターン値 6 が返されますが、そうでない場合には 3 回リトラ
```

イ後リターン値 6 が返されます。）

8=タイムアウト

WriteMessage 関数が wait_flg = 1 で呼び出された時点で、自ノードは FL-net に参加していたが、送信先ノードから ACK が返る前に自ノードが離脱した場合、3 秒後にリターン値 8 が返されます。

9=重大エラー（詳細は P34を参照）

14=デバイスはオープンされていない

19=mutex 排他制御エラー

注意事項：ｒｃ=バッファフルの時は、アプリケーション側で再送信を行ってください。

＜メッセージ デバイスプロファイル応答 のデータ形式＞

FL-net プロトコルでは、メッセージのデバイスプロファイルが必須となっております。

以下は、その応答メッセージで用いるデータフォーマットです。

MSTC 認証済みの情報を用いる必要があり、必ずこのデータをお使いください。

ストリングの最後に、NULL は含みません。

xxh は、16 進数を意味します。

データは、全 116Byte です。

30h 72h

30h 70h

13h 06h “COMVER”

02h 01h 01h

13h 02h “ID”

13h 07h “SYSPARA”

13h 03h “REV”

02h 01h 14h

13h 07h “REVDATE”

30h 0Ah

02h 02h 07h D6h

02h 01h 08h

02h 01h 01h

13h 0Ah “DVCATEGORY”

13h 08h “COMPUTER”

13h 06h “VENDOR”

13h 0Ah “CENTURYSYS”

13h 07h “DVMODEL”

13h 06h “S-0616”

3.9. CheckMessageEnd

機能： メッセージ送信の関数 WriteMessage で、メッセージ送信完了を待たずに return をする選択をした時に、メッセージ送信が正常終了かを後で確認する必要があります。
この関数で、メッセージ送信の終了ステータスを返します。
複数のメッセージが送信未完の時には、最古の情報から通知します。

記述： rc = CheckMessageEnd(&node, &tcd, dev_no);
unsigned short *node; ：送信終了したメッセージのノード No.
unsigned short *tcd ：送信終了したメッセージのＴＣＤ
unsigned short dev_no; ： デバイス No.（0～3）

unsigned short rc; ：
0=正常終了
2=パラメータエラー
6=送信失敗
7=送信未完
9=重大エラー（詳細は P34を参照）
14=デバイスはオープンされていない
19=mutex 排他制御エラー

注意事項： rc=送信失敗の場合、ＦＡリンクプロトコル仕様では再送信の必要はないとされています。

3.10. ReadMessage

機能： メッセージ受信の有無をチェックして、受信がある場合その情報を構造体とバッファへセットします。
受信データは 16 件までキューイングを行い、最古のものから渡します。
wait_flg で、受信完了を待って関数を return するか完了を待たずに return するかを選択します。
wait_flg として０（受信完了を待たない）が指定された場合、受信データが存在するとき、その情報を構造体とバッファにセット後リターン値０を返し、受信データが存在しないときリターン値４（データなし）を返します。 wait_flg として１（受信完了を待つ）が指定された場合、２秒以内に受信データが存在すれば、その情報を構造体と バッファにセット後リターン値０を返し、２秒待っても受信データが存在しない場合、リターン値８（タイムアウト）を返します。

記述： rc = ReadMessage(&mrcv_prm, &buff, wait_flg, dev_no);

```
struct MessageInfo {
unsigned char     node      : 相手先ノード No.（1～254）
unsigned char     result;   : 結果コード
                              bit4：1=ブロードキャスト受信、0=1：1 受信
                              bit0～2：ＦＡリンクヘッダーの M_RLT
unsigned long     tcd;      : トランザクションコード
unsigned short    bc;       : バイトサイズ
unsigned long     addr;     : 仮想アドレス空間 アドレス
unsigned short    length;   : 仮想アドレス空間 データ長
} mrcv_prm;

unsigned char     *buff;    : バッファアドレス
unsigned short    wait_flg  : 0= 受信完了を待たない、1=受信完了を待つ
unsigned short    dev_no;   : デバイス No.（0～3）

unsigned short  rc; :
          0=正常終了
          2=パラメータエラー
          4=データ無し（wait_flg=0 の時のみ通知）
          8=タイムアウト（2 秒）
          9=重大エラー（詳細は P34を参照）
         14=デバイスはオープンされていない
         19=mutex 排他制御エラー
```

注意事項：

3.11. GetNodeInfo

機能： 指定されたノードの情報を指定の構造体へ返します。

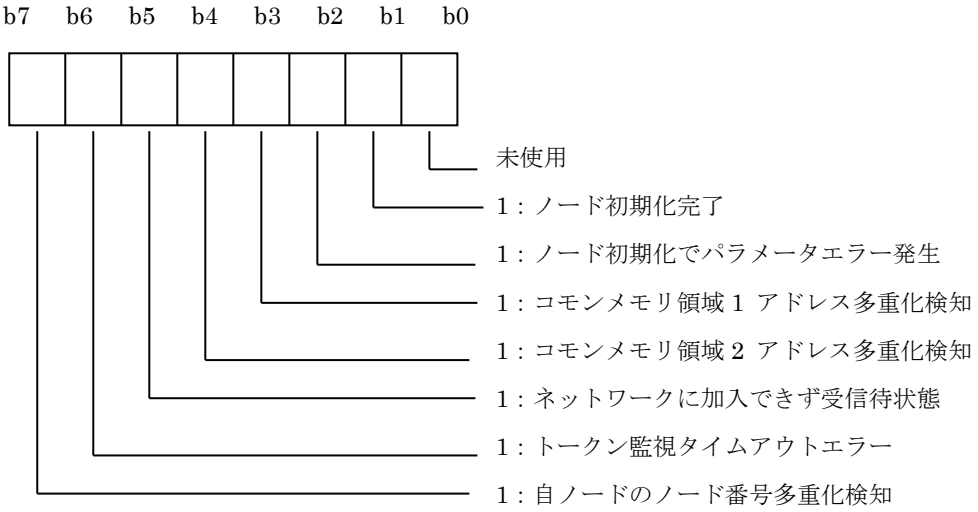
記述： rc = GetNodeInfo(node, &node_inf, dev_no);

```
unsigned short      node;                 : ノード No.（1～254）
struct NodeInfo {
        char              sanka;          : 0=離脱、1=参加
        unsigned short    uls;            : 上位層の状態
        unsigned short    cmn1_addr;      : コモンメモリ 1 アドレス
        unsigned short    cmn1_size;      : コモンメモリ 1 サイズ
        unsigned short    cmn2_addr;      : コモンメモリ 2 アドレス
        unsigned short    cmn2_size;      : コモンメモリ 2 サイズ
        unsigned short    rct;            : リフレッシュサイクル許容時間
        unsigned char     tw;             : トークン監視時間
        unsigned char     mft;            : 最小フレーム間隔
        unsigned char     link_status;    : リンクの状態（注）
        unsigned char     my_status;      : 自ノードの状態（注）
        unsigned char     vender[11];     : ベンダ名(11Byte 目は NULL)
        unsigned char     maker[11];      : メーカー名(11Byte 目は NULL)
        unsigned char     node[11];       : ノード名(11Byte 目は NULL)
} node_inf;
unsigned short      dev_no;               : デバイス No.（0～3）

unsigned short    rc;  :
        0=正常終了
        2=パラメータエラー
        8=タイムアウト（1 秒）
        9=重大エラー（詳細は P34を参照）
       14=デバイスはオープンされていない
       19=mutex 排他制御エラー
```

注意事項： 自ノードの状態とリンクの状態の詳細は、次頁を参照してください。
自局が途中参加時には、ベンダ名、メーカー名、ノード名に有効データがセットされない場合があります。

自ノードの状態

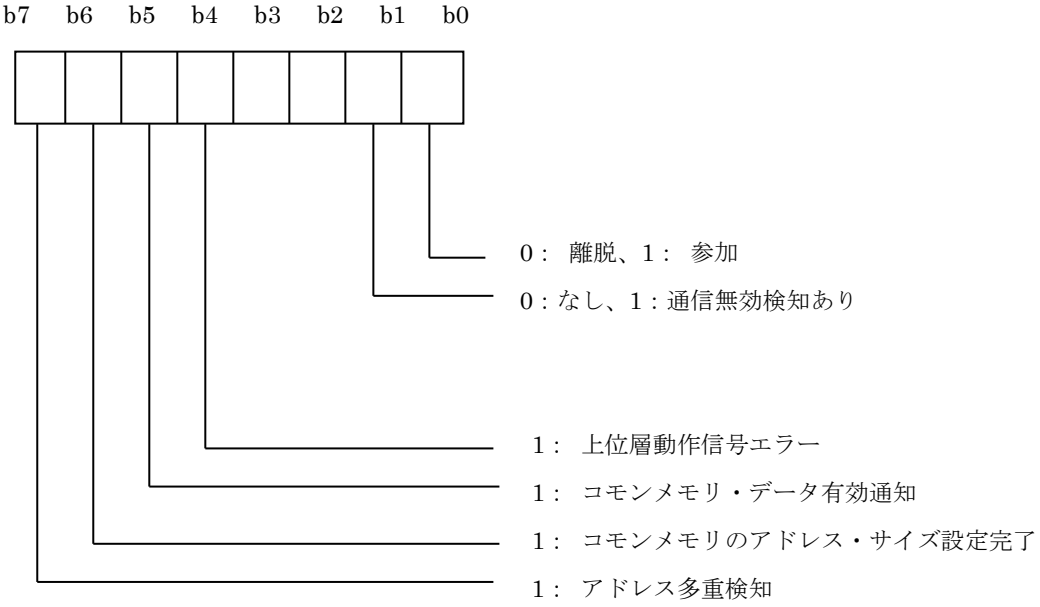
FL-PCI カードが把握している 自ノードの状態を、次の 1 バイトデータで
通知する。



(注) b2 または b7 が 1 の場合、初期化データのフェイタルエラーを示します。

リンクの状態

ヘッダー情報 LKS に参加／離脱情報を付加する。



3.12. GetRingInfo

機能： 現在の FL-net に参加中の他ノードの No.を渡します。
ノード数と参加中の全のノード No.（自ノードを除く）をバッファにセット
します。

記述： rc = GetRingInfo(&ring_inf, dev_no);

```
struct SankaReqTable {
        unsigned char node_cnt;      : ノード数（1～254）
        unsigned char node_no[254];  : 参加ノード No.
} ring_inf;
unsigned short    dev_no;            : デバイス No.（0～3）

unsigned short  rc;  :
        0=正常終了
        2=パラメータエラー
        8=タイムアウト(1 秒)
        9=重大エラー（詳細は P34を参照）
       14=デバイスはオープンされていない
       19=mutex 排他制御エラー
```

注意事項：

3.13. GetNetInfo

機能： 現在の FL-net のリフレッシュサイクル計測値を返します。

記述： rc = GetNetInfo(&net_inf, dev_no);

```
struct    NetInfo {
            unsigned short      rct;            : リフレッシュサイクル設定値
            unsigned short      rcm;            : リフレッシュサイクル測定値（現在）
            unsigned short      rcm_max;        : リフレッシュサイクル測定値（最大）
            unsigned short      rcm_min;        : リフレッシュサイクル測定値（最小）
            unsigned char       link_status;    : リンクの状態（注）
 } net_inf;
 unsigned short      dev_no;   : デバイス No.（0～3）

unsigned short  rc;  :
            0=正常終了
             2=パラメータエラー
             8=タイムアウト（1 秒）
             9=重大エラー（詳細は P34を参照）
            14=デバイスはオープンされていない
            19=mutex 排他制御エラー
```

注意事項： リンクの状態（注）はＰ23 を参照。

3.14. GetLogInfo

機能： 自ノードの現在の FL-net ログ情報を返します。
FL-net プロトコルで定義されるログ情報の「必要」「任意」の全情報をサポートします。
詳細は JEMA 規格の『実装ガイドライン』JEM-TR213 を参照願います。

記述： rc = GetLogInfo(&log_buf, dev_no);

unsigned long log_buf[128]; ： ログデータ情報のバッファ
unsigned short dev_no; ： デバイス No.（0～3）

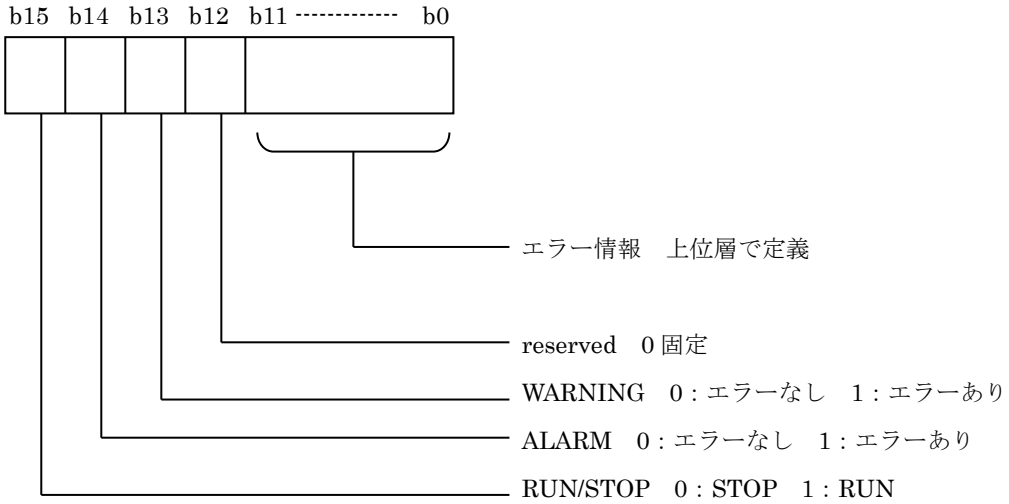
unsigned short rc; ：
0=正常終了
2=パラメータエラー
8=タイムアウト（1 秒）
9=重大エラー（詳細は P34を参照）
14=デバイスはオープンされていない
19=mutex 排他制御エラー

注意事項：

3.15. SetUplStatus

機能： この関数で指定した上位層ステータスは、ＦＡリンクヘッダーの ULS にセット
されます。
FL-PCI カードは、初期化コマンドを受け取った時に RUN/STOP Bit=0 をセットします。
（ULS ディフォルト値=0000h）
上位層の状態が変化した時にこの関数を用いてＦＡリンクヘッダーの ULS の値を書き換える必要があります。

ULS（上位層の状態）



記述： rc = SetUplStatus(uls, dev_no);

unsigned short　uls;　：上位層ステータス

unsigned short　dev_no;　：デバイス No.（0～3）

unsigned short　rc;　:

0=正常終了

2=パラメータエラー

8=タイムアウト（1 秒）

9=重大エラー（詳細は P34を参照）

14=デバイスはオープンされていない

19=mutex 排他制御エラー

注意事項：

3.16. GetErrInfo

機能：　重大エラーが発生した時に、その詳細情報を取り込みます。

記述：　rc = GetErrInfo(&err_inf, dev_no);

```
struct ErrInfo {
        unsigned short      err_code;      : エラーコード（注）
        unsigned short      length;        : メッセージ長
        unsigned char       err_msg[128];  : メッセージ（ASCII code）
} err_inf;

unsigned short  rc;  :
        0=正常終了
        2=パラメータエラー
       14=デバイスはオープンされていない
```

注意事項：エラーコードの内容

7F00h：システムエラー。詳細はエラーメッセージにセットされます。

7F01h：フラッシュメモリーエラー。（初期化時のみ発生）

7F02h：RAM テストエラー。（初期化時のみ発生）

7F03h：Ｅｔｈｅｒｎｅｔテストエラー。（初期化時のみ発生）

7F04h：ＥＥＰＲＯＭサムチェックエラー。（初期化時のみ発生）

7F05h：ＣＰＵ　ＢＵＳエラーが発生した。

7F06h：イリーガルインストラクションが発生した。

7F07h：ボードタイプとファームウェアが一致しない。

3.17. DeviceClose

機能： 初期化でオープンしたデバイスをクローズします。
アプリケーション終了時と FL-PCI カード再初期化の時には、この関数
を呼んで下さい。
デバイスクローズ時の FL-PCI カードを RESET 状態にするかどうかを引数で指定します。

記述： rc = DeviceClose(dev_no, reset_f);
unsigned short dev_no; ： デバイス No.（0～3）
unsigned short reset_f ： FL-PCI カードの RESET 指示
（0：RESET なし、1：RESET あり）
unsigned short rc; ：
0=正常終了
2=パラメータエラー
15=デバイスクローズエラー
19=mutex クローズエラー

注意事項：

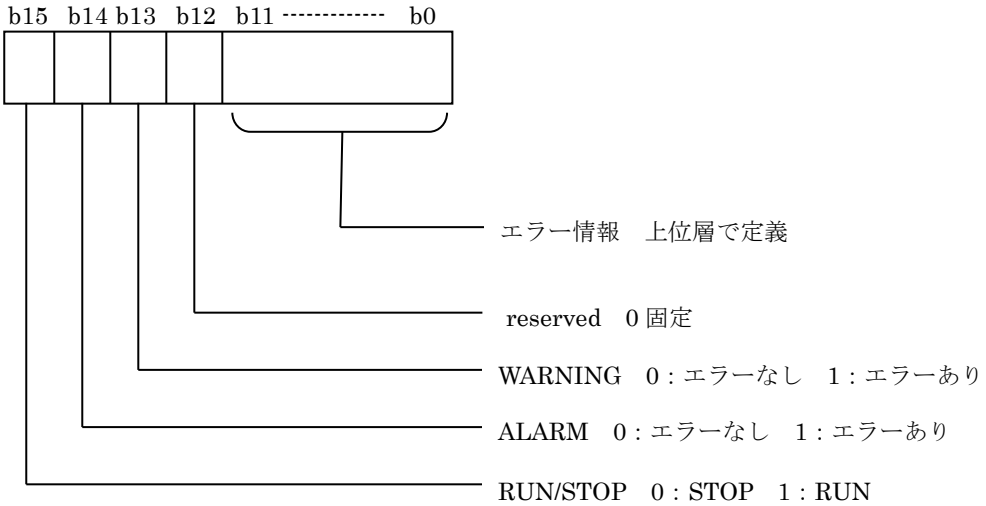
3.18. ReadUls_All

機能： ノード No.1～254 の ULS（上位層の状態）と LKS（リンクの状態）を得ます。
離脱中のノードの ULS,LKS は 0 でセットされます。

記述： rc = ReadUls_All (&UlsInfo, dev_no);

```
struct UlsInfo {
    unsigned short uls[254];    : uls[0]にはノード 1 の ULS がセットされます。
    unsigned char  lks[254];    : lks[0]にはノード 1 の LKS がセットされます
};
unsigned short dev_no;          : デバイス No.（0～3）
unsigned short  rc;  :
                0=正常終了
                2=パラメータエラー
                8=タイムアウト（1 秒）
                9=重大エラー（詳細は P34を参照）
               14=デバイスはオープンされていない
               19=mutex 排他制御エラー
```

ULS（上位層の状態）



LKS（リンクの状態）

P23 を参照ください。

3.19. Dll_Version

機能： DLL の現在のバージョン No.を返します。

記述： dll_ver = Dll_Version (void);

unsigned short dll_ver; : バイナリデータ（01～ ）

注意事項：

デバイスオープンなしで実行ができます。

この機能は、マルチプロセス対応版の DLL 以降のサポートとなります。

3.20. Firmware_Version

機能： FL-PCI カードの現在のファームウェアバージョン No.を返します。

記述： rc =Firmware_Version (&version, dev_no);

unsigned short version; : バイナリデータ

Bit0～7： ファームウェアバージョン No. (01～255)

Bit8～15： 00h=FL-PCI/V2 (FL-net Ver1)

0Ah= FL-PCI/V2 (FL-net Ver2)

14h= FL-PCI/V2-100 または 100L (FL-net Ver2)

1Eh= FL-PCIe または FL-PCI/V2 100L(FL-net Ver3)

unsigned short dev_no : デバイス No.（0～3）

unsigned short rc; :

0=正常終了

2=パラメータエラー

8=タイムアウト（1 秒）

9=重大エラー（詳細は P を参照）

14=デバイスはオープンされていない

19=mutex 排他制御エラー

注意事項：

この機能は、初期化が正常終了した後に行ってください。

旧製品 FL-PCI/V2 の Ver.47(FL-net Ver1 対応品)と Ver.1004(FL-net Ver2 対応品)以前のバージョンは本機能をサポートしていません。

3.21. Set_Speed

機能： イーサネットの通信速度を指定します。

記述： rc = Set_Speed (speed, dev_no);

unsigned long speed; : 0：10Mbps 1：100Mbps 2：Auto Negotiation

（注）

0：10Mbps のとき、半 2 重固定になります。

1：100Mbps のとき、半 2 重固定になります。

2：Auto Negotiation の場合は、

10Mbps／100Mbps、、半 2 重／全 2 重は相手装置によって

自動選択となります。

unsigned short dev_no : デバイス No.（0～3）

unsigned short rc; :

0=正常終了

2=パラメータエラー

注意事項：

Set_Speed()は、関数 FL_Initialize()を Call する前に行ってください。

Set_Speed()を行わない場合は、ディフォルト設定である 10Mbps の設定になります。

1 枚の FL-PCI カードへデバイス No.別に 2 種類の動作モードを指定することはできません。

この機能は、新 FL-PCI/V2-100 でのみ有効です。旧 FL-PCI /V2 では 10Mbps 固定となります。

3.22. Get_Speed

機能： 現在設定されているイーサネットの通信速度を取得します。
Set_Speed()で Auto Negotiation を選択した場合は、通信モードは相手先に合わせて自動設定されます。
Get_Speed()にて現在設定されている通信モードを知ることができます。

記述： rc = Set_Speed (&speed, &duplex, dev_no);

unsigned long *speed; : 0：10Mbps 1：100Mbps
unsigned long *duplex; : 0：半２重 1：全２重
unsigned short dev_no : デバイス No.（0～3）
unsigned short rc; :
0=正常終了
2=パラメータエラー
8=タイムアウト（1 秒）
9=重大エラー（詳細は P34を参照）
14=デバイスはオープンされていない
19=mutex 排他制御エラー
20=ケーブル外れのため設定が未完

注意事項：
Get_Speed()は、関数 FL_Initialize() の後に行ってください。
この機能は、新 FL-PCI/V2-100 および 100L でのみ有効です。旧 FL-PCI /V2 ではタイムアウトエラーとなります。

4. Visual Basic I/F 仕様

本仕様内でのデータ型表記は.Net 対応になります。

・16 ビットは「Short」、32 ビットは「Integer」になります。

・ユーザ定義型は「Structure」になります。

4.1. FL_Initialize

機能： デバイスをオープンし、FL-PCI カード を初期化します。

記述： rc = FL_Initialize(init_prm)

Byref init_prm As InitInfo ： 初期化引数のユーザ定義型

Structure InitInfo

ip_adr As Short ： 自ノード IP アドレス。下 1 バイトは、自ノード No.とする

dev_no As Short ： デバイス No.

ボード 1=0 or 1 ボード 2=2 or 3

cmn1_adr As Short ： コモンメモリ 領域 1 アドレス（0～511）

cmn1_size As Short ： コモンメモリ 領域 1 サイズ（0～512）

cmn2_adr As Short ： コモンメモリ 領域 2 アドレス(0～8191)

cmn2_size As Short ： コモンメモリ 領域 2 サイズ（0～8192）

tw As Byte ： トークン監視タイムアウト値（1～255）

mft As Byte ： 最小フレーム間隔（0～50）

vender(10) As Byte ： ベンダ名 ”CENTURYSYS” 固定

maker(10) As Byte ： メーカー型式 ”S-0616___” 固定

name(10) As Byte ： ノード名（10 文字以内の任意の文字列）

End Structure

rc As Short ：

0=正常終了

2=パラメータエラー

旧 FL-PCI /V2 へ 100Mbps、オートネゴシエーションを指定した場合にも通知されます。

8=タイムアウト（最大 4 秒）

9=重大エラー（詳細は P54 を参照）

10=デバイスオープンエラー（ファイルクリエート）

11=デバイスオープンエラー（デバイス取得）

12=デバイスオープンエラー（ポインタ取得）

18=一方のデバイスがオープン中に初期化を行った

19=mutex 取得エラー

注意事項： 初期化の後再初期化を行う場合は、一旦 DeviceClose()を使って 2 デバイスをクローズしてください。デバイスクローズを行わないで再初期化を行うと、デバイスオープンエラーが発生します。

4.2. DeviceOpen

機能： 指定デバイスをオープンします。
FL-PCI カードは 1 カードで 2 デバイスが使用可能です。
片側のデバイスが FL_Initialize で FL-PCI カードを初期化した後に、
この関数を呼び出してもう 1 方のデバイスをオープンして下さい。
初期設定前にこの関数を呼び出した場合には、エラー終了します。

記述： rc = DeviceOpen(dev_no);
ByVal dev_no As Short : デバイス No.（0～3）

rc As Short :
0=正常終了
2=パラメータエラー
8=タイムアウト（最大 4 秒）
9=重大エラー（詳細は P54 を参照）
10=デバイスオープンエラー（ファイルクリエート）
11=デバイスオープンエラー（デバイス取得）
12=デバイスオープンエラー（ポインタ取得）
16=デバイスの 2 重オープン
17=デバイスは未初期化
19=mutex 取得エラー

注意事項：

4.3. WriteCyclic_ALL

機能： 初期設定時に設定した自ノードコモンメモリー領域へ、自ノードのサイクリックデータを書き込みます。
引数のコモン領域 1、2 のバッファからデュアルポート RAM へ一括コピーします。
ノード No.、領域情報は、初期化関数の引数から自動認識します。

記述： rc = WriteCyclic_ALL（ flbuf, dev_no ）

```
ByRef  flbuf  As  CmnStruct      : ユーザーコモン バッファ 1、2
Structure CmnStruct
    common1_buf（511）  As  Short
    common2_buf (512 * 16 – 1) As Short
End Structure
ByVal  dev_no  As  Short        : デバイス No.（0～3）

rc  As  Short  :
                0=正常終了
                2=パラメータエラー
                3=バッファフル
               14=デバイスはオープンされていない
               19=mutex 排他制御エラー
```

注意事項:

(1) 本関数によってデュアルポート RAM にサイクリックデータが書き込まれると、FL-PCI カードにおいて、自ノードサイクリックデータ送信バッファの更新処理が発生します。この処理は FL-PCI カードの CPU に負荷をかけることになり、タイムロスになりますので、サイクリックデータ変更が発生したときにのみ本関数をコールして下さい。

(2) 自ノードのサイクリックデータは、FL-net プロトコルの規定で、トークンが自分にまわってきたときにネットワークに送出されます。このため、本関数をリフレッシュサイクル（トークン 1 周）より短い周期でコールした場合や、離脱時にコールした場合、戻り値 3（バッファフル）が返されることがあります。これはワーニングエラーですので、少し待った後リトライを行うと正常終了します。

4.4. WriteCyclic_Rel

機能： コモンメモリで指定ノードが占有する領域先頭をオフセット 0 とする相対アドレスを用います。
ユーザーコモンバッファの自ノード領域のワードオフセット位置からワードサイズ分のサイクリックデータを最新データ上に更新します。
ノード No.、領域情報は、初期化関数の引数から自動認識します。

記述： rc = WriteCyclic_Rel(common, offset, size, flbuf, dev_no)

ByVal common As Short : 1=コモン領域 1、2=コモン領域 2
ByVal offset As Short : ワードオフセット（0～511 または 0～8191）
ByVal size As Short : ワードサイズ（1～512 または 1～8192）

ByRef flbuf As CmnStruct : ユーザーコモン バッファ 1、2

```
Structure CmnStruct
    common1_buf (511) As Short
    common2_buf (512 * 16 – 1) As Short
End Structure
```

ByVal dev_no As Short : デバイス No.（0～3）

rc As Short :
0=正常終了
2=パラメータエラー
3=バッファフル
14=デバイスはオープンされていない
19=mutex 排他制御エラー

注意事項:

(1) 本関数によってデュアルポート RAM にサイクリックデータが書き込まれると、FL-PCI カードにおいて、自ノードサイクリックデータ送信バッファの更新処理が発生します。この処理は FL-PCI カードの CPU に負荷をかけることになり、タイムロスになりますので、サイクリックデータ変更が発生したときにのみ本関数をコールして下さい。

(2) 自ノードのサイクリックデータは、FL-net プロトコルの規定で、トークンが自分にまわってきたときにネットワークに送出されます。 このため、本関数をリフレッシュサイクル（トークン 1 周）より短い周期でコールした場合や、離脱時にコールした場合、戻り値 3（バッファフル）が返されることがあります。これはワーニングエラーですので、少し待った後リトライを行うと正常終了します。

4.5. ReadCyclic_Abs

機能： コモンメモリ 1 または 2 の先頭からのワードオフセット、ワードサイズを指定して、サイクリックデータをユーザーコモンバッファへ読み出します。
この関数では、該当するコモン領域に割り当てられているノードがあれば、そのノードの最新のデータを読み出します。
読み出し領域が複数のノードの領域にまたがる場合、それらのデータを合成して、データを読み出します。
どのノードの送信領域にもなっていない領域には‘0’が書き込まれます。

記述： rc = ReadCyclic_Abs(common, offset, size, flbuf ,dev_no)

ByVal common As Short ： 1=コモン領域 1、2=コモン領域 2
ByVal offset As Short ： ワードオフセット（0～511 または 0～8191）
ByVal size As Short ： ワードサイズ（1～512 または 1～8192）

ByRef flbuf As CmnStruct ： ユーザーコモン バッファ 1、2

```
Structure CmnStruct
    common1_buf (511) As Short
    common2_buf (512 * 16 – 1) As Short
End Structure
```

ByVal dev_no As Short ： デバイス No.（0～3）

rc As Short ：
0=正常終了
2=パラメータエラー
14=デバイスはオープンされていない
19=mutex 排他制御エラー

注意事項：

(1) 本関数は指定されたコモン領域の最新のサイクリックデータを読み出すものです。
ノードのサイクリックデータはリフレッシュサイクル（トークン一周）で更新されますので、サイクリックデータの変化を完全に把握したい場合は、本関数をリフレッシュサイクルより短い周期でコールして下さい。

(2) 本関数は、指定されたコモン領域に離脱中のノードのコモン領域が含まれる場合、そのサイクリックデータはゼロをセットします。

4.6. ReadCyclic_Rel

機能： コモンメモリで指定ノードが占有する領域先頭をオフセット 0 とする相対
アドレスを用います。
領域内のワードオフセットからワードサイズのサイクリックデータをユーザ
コモンバッファへ読み込みます。
指定されたノードがトークンリングに参加していなければ、エラーを返します。

記述： rc = ReadCyclic_Rel(node, common, offset, size, flbuf, dev_no)

ByVal node As Short : ノード No.（1～254）
ByVal common As Short : 1=コモン領域 1、2=コモン領域 2
ByVal offset As Short : ワードオフセット（0～511 または 0～8191）
ByVal size As Short : ワードサイズ（1～512 または 1～8192）

ByRef flbuf As CmnStruct : ユーザーコモン バッファ 1、2

```
Structure CmnStruct
    common1_buf (511) As Short
    common2_buf (512 * 16 – 1) As Short
End Structure
```

ByVal dev_no As Short : デバイス No.（0～3）

rc As Short :
0=正常終了
2=パラメータエラー
4=データなし
5=ノード離脱
14=デバイスはオープンされていない
19=mutex 排他制御エラー

注意事項：

(1) 本関数は指定ノードの指定領域の最新のサイクリックデータを読み出すものです。
ノードのサイクリックデータはリフレッシュサイクル（トークン一周）で更新されますので、サイクリックデータの変化を完全に把握したい場合は、本関数をリフレッシュサイクルより短い周期でコールして下さい。

(2) 本関数はコモン領域を持たないノードには使用できません。

4.7. ReadCyclic_ALL

機能： 指定ノードのコモンメモリ領域から指定ノードのコモン 1、2 の全サイクリックデータを
読み出します。
引数のコモン領域 1、2 のデータバッファへデュアルポート RAM から一括コピーします。

記述： rc = ReadCyclic_ALL(node, flbuf, dev_no);

```
ByVal  node  As  Short        : ノード No.（1～254）
ByRef  flbuf  As  CmnStruct    : ユーザーコモン バッファ 1、2
Structure CmnStruct
    common1_buf (511) As Short
    common2_buf (512 * 16 – 1) As Short
End Structure
ByVal  dev_no  As  Short      : デバイス No.（0～3）

rc  As  Short  :
        0=正常終了
        2=パラメータエラー
        4=データなし
        5=ノード離脱
       14=デバイスはオープンされていない
       19=mutex 排他制御エラー
```

注意事項：

(1) 本関数は指定されたノードの最新のサイクリックデータを読み出すものです。
ノードのサイクリックデータはリフレッシュサイクル（トークン一周）で更新されますので、サイクリックデータの変化を完全に把握したい場合は、本関数をリフレッシュサイクルより短い周期でコールして下さい。

4.8. WriteMessage

機能： 構造体で設定されたメッセージを送ります。
FL-net ではメッセージ送信の条件がそろうまで送信を待ちます。
また、1：1 メッセージ送信の場合、相手ノードからの正常終了の ACK を受信するまで 3 回のリトライが行われることがあります。
そのため送信の登録から送信完了まで時間がかかることがあり、引数の wait_flg で、送信完了を待って関数を return するか完了を待たずに return するかを選択します。
完了を待たない選択をした場合、CheckMessageEnd 関数を使ってメッセージ送信が正常終了をしたかをチェックする必要があります。

記述： rc = WriteMessage(msnd_prm, buff, wait_flg, dev_no)

```
ByRef  msnd_prm  As  MessageInfo  ： メッセージ送信 引数 ユーザ定義型
Structure MessageInfo
    node  As  Byte     ： 相手先ノード No.（1～255、255 はブロードキャスト指定）
    result  As  Byte   ： 結果コード（0～2）
    tcd  As  Integer   ： トランザクションコード
    bc  As  Short      ： バイトサイズ（0～1024）
    addr  As  Integer  ： 仮想アドレス空間 アドレス
    length  As  Short  ： 仮想アドレス空間 データ長
  End Structure

ByRef     buff  As  MsgData     ： バッファアドレス
Structure MsgData
   msg_buf (1023) As Byte
End Structure

ByVal  wait_flg  As  Short  ：0=送信完了を待たない、1=送信完了を待つ
ByVal  dev_no  As  Short  ：デバイス No.（0～3）
 rc  As  Short :
          0=正常終了
           2=パラメータエラー
           3=バッファフル
           5=ノード離脱
           6=送信失敗（wait_flg=1 の時のみ通知）
           8=タイムアウト(最大 2 秒)
```

9=重大エラー（詳細は P54 を参照）

14=デバイスはオープンされていない

19=mutex 排他制御エラー

注意事項： ｒｃ=バッファフルの時は、アプリケーション側で再送信を行ってください。

ｒｃ=送信失敗の場合、ＦＡリンクプロトコル仕様では再送信の必要はないとされています。

＜ メッセージ デバイスプロファイル応答 のデータ形式＞

FL-net プロトコルでは、メッセージのデバイスプロファイルが必須となっております。

以下は、その応答メッセージで用いるデータ フォーマットです。

MSTC 認証済みの情報を用いる必要があり、必ずこのデータをお使いください。

ストリングの最後に、NULL は含みません。

xxh は、HEX 表記データを意味します。

データは、全 116Byte です。

30h 72h

30h 70h

13h 06h “COMVER”

02h 01h 01h

13h 02h “ID”

13h 07h “SYSPARA”

13h 03h “REV”

02h 01h 14h

13h 07h “REVDATE”

30h 0Ah

02h 02h 07h D6h

02h 01h 08h

02h 01h 01h

13h 0Ah “DVCATEGORY”

13h 08h “COMPUTER”

13h 06h “VENDOR”

13h 0Ah “CENTURYSYS”

13h 07h “DVMODEL”

13h 06h “S-0616”

4.9. CheckMessageEnd

機能： メッセージ送信の関数 WriteMessage で、メッセージ送信完了を
待たずに return をする選択をした時に、メッセージ送信が正常終了かを
後で確認する必要があります。
この関数で、メッセージ送信の終了ステータスを返します。
複数のメッセージが送信未完の時には、最古の情報から通知します。

記述： rc = CheckMessageEnd(node, tcd, dev_no)

ByRef node As Short ：送信終了したメッセージのノード No.
ByRef tcd As Short ：送信終了したメッセージの TCD
ByVal dev_no As Short ： デバイス No.（0～3）

rc As Short ：
0=正常終了
2=パラメータエラー
6=送信失敗
7=送信未完
9=重大エラー（詳細は P54 を参照）
14=デバイスはオープンされていない
19=mutex 排他制御エラー

注意事項： rc=送信失敗の場合、ＦＡリンクプロトコル仕様では再送信の必要はないとされています。

4.10. ReadMessage

機能： メッセージ受信の有無をチェックして、受信がある場合その情報を構造体とバッファへセットします。
受信データは 16 件までキューイングを行い、最古のものから渡します。
wait_flg で、受信完了を待って関数を return するか完了を待たずに return するかを選択します。
wait_flg として０（受信完了を待たない）が指定された場合、受信データが存在するとき、その情報を構造体とバッファにセットしてリターン値０を返し、受信データが存在しないときリターン値４（データなし）を返します。
wait_flg として１（受信完了を待つ）が指定された場合、２秒以内に受信データが存在すれば、その情報を構造体とバッファにセット後リターン値０を返し、２秒待っても受信データが存在しないときは、リターン値８（タイムアウト）を返します。

記述： rc = ReadMessage(mrcv_prm, buff, wait_flg, dev_no)

```
ByRef   mrcv_prm   As   MessageInfo    :  メッセージ送信 引数 ユーザ定義型
Structure MessageInfo
    node   As   Byte      :  相手先ノード No.（1～254）
    result   As   Byte    :  結果コード
                             bit4：1=ブロードキャスト受信、0=1：1 受信
                             bit0～3：ＦＡリンクヘッダーの M_RLT
    tcd   As    Integer   :  トランザクションコード
    bc   As   Short       :  バイトサイズ
    addr   As   Integer   :  仮想アドレス空間 アドレス
    length   As   Short   :  仮想アドレス空間 データ長
End Structure
ByRef     buff   As   MsgData       :  バッファアドレス
Structure MsgData
    msg_buf (1023) As Byte
End Structure
ByVal   dev_no   As   Short       :  デバイス No.（0～3）
ByVal   wait_flg   As   Short     : 0=受信完了を待たない、
                                    1=受信完了を待つ
rc   As   Short   :
          0=正常終了
          2=パラメータエラー
          4=データ無し（wait_flg=0 の時のみ通知）
```

8=タイムアウト（2 秒）

9=重大エラー（詳細は P54 を参照）

14=デバイスはオープンされていない

19=mutex 排他制御エラー

4.11. GetNodeInfo

機能： 指定されたノードの情報を指定の構造体へ返します。

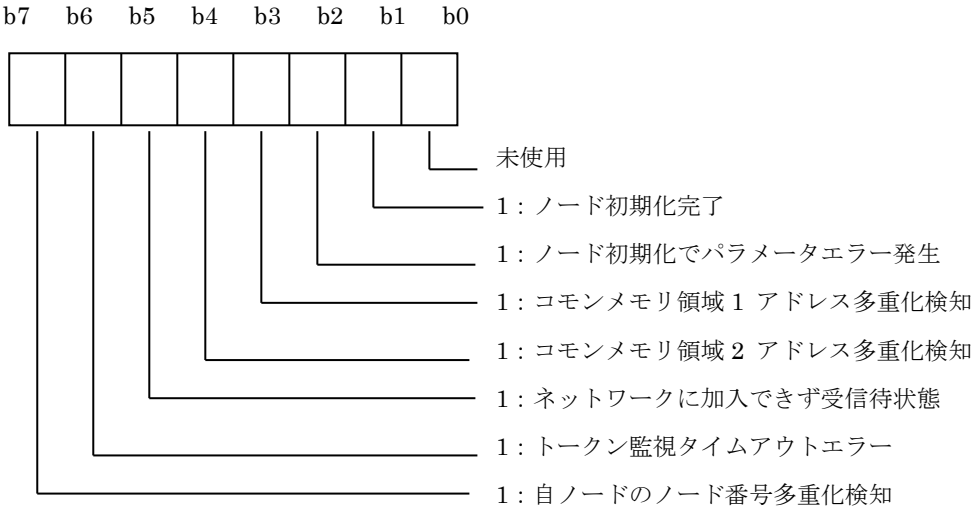
記述： rc = GetNodeInfo(node, node_inf, dev_no)

```
ByVal  node  As  Short                 ： ノード No.（1～254）
ByRef  node_inf  As  NodeInfo          ： 戻り値のユーザ定義型
Structure NodeInfo
        sanka  As  Byte                ： 0=離脱、1=参加
        uls  As  Short                 ： 上位層の状態
        cmn1_addr  As  Short           ： コモンメモリ 1 アドレス
        cmn1_size  As  Short           ： コモンメモリ 1 サイズ
        cmn2_addr  As  Short           ： コモンメモリ 2 アドレス
        cmn2_size  As  Short           ： コモンメモリ 2 サイズ
        rct  As  Short                 ： リフレッシュサイクル許容時間
        tw  As  Byte                   ： トークン監視時間
        mft  As  Byte                  ： 最小フレーム間隔
        link_status  As  Byte          ： リンクの状態（注）
        my_status  As  Byte            ： 自ノードの状態（注）
        vender(10)  As  Byte           ： ベンダ名(11Byte 目は NULL)
        maker(10)  As  Byte            ： メーカー名(11Byte 目は NULL)
        node(10)  As  Byte             ： ノード名(11Byte 目は NULL)
End Structure
ByVal  dev_no  As  Short               ： デバイス No.（0～3）

rc  As  Short  ：
        0=正常終了
        2=パラメータエラー
        8=タイムアウト（1 秒）
        9=重大エラー（詳細は P54 を参照）
       14=デバイスはオープンされていない
       19=mutex 排他制御エラー
```

注意事項： 自ノードの状態とリンクの状態の詳細は、次頁を参照してください。
自局が途中参加時には、ベンダ名、メーカー名、ノード名に有効データがセットされない場合があります。

自ノードの状態

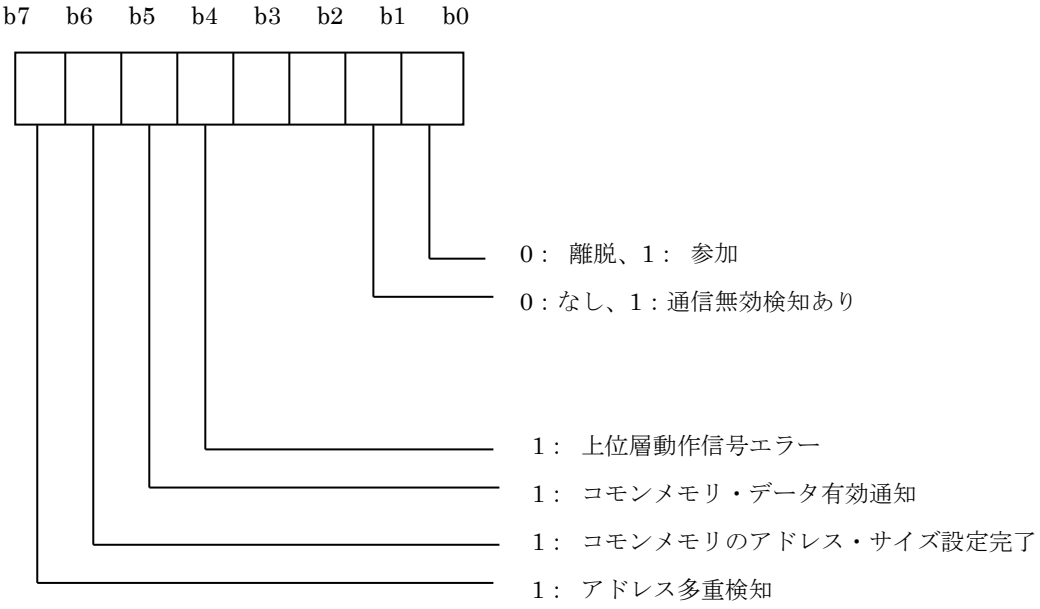
FL-PCI カードが把握している 自ノードの状態を、次の 1 バイトデータで通知する。



(注) b2 または b7 が 1 の場合、初期化データのフェイタルエラーを示します。

リンクの状態

ヘッダー情報 LKS に参加／離脱情報を付加する。



4.12. GetRingInfo

機能： 現在の FL-net に参加中の他ノードの No.を渡します。
ノード数と参加中の全ノード No.（自ノードを除く）をバッファにセット
します。

記述： rc = GetRingInfo(ring_inf, dev_no)

```
ByRef  ring_inf  As  SankaReqTable    :  参加情報のユーザ定義型
Structure SankaReqTable
        node_cnt   As  Byte          :  ノード数（1～254）
        node_no(253)  As  Byte       :  参加ノード No.
End Structure
ByVal  dev_no  As  Short      :  デバイス No.（0～3）
```

rc As Short :

0=正常終了
2=パラメータエラー
8=タイムアウト(1 秒)
9=重大エラー（詳細は P54 を参照）
14=デバイスはオープンされていない
19=mutex 排他制御エラー

注意事項：

4.13. GetNetInfo

機能： 現在の FL-net のリフレッシュサイクル計測値を返します。

記述： rc = GetNetInfo(net_inf ,dev_no)

```
        ByRef   net_inf   As   NetInfo          :  ネット情報 ユーザ定義型
Structure   NetInfo
                rct   As   Short            :  リフレッシュサイクル設定値
                rcm   As   Short            :  リフレッシュサイクル測定値（現在）
                rcm_max   As   Short        :  リフレッシュサイクル測定値（最大）
                rcm_min   As   Short        :  リフレッシュサイクル測定値（最小）
                link_status   As   Byte     :  リンクの状態（注）
          End Structure
          ByVal   dev_no   As   Short       :  デバイス No.（0～3）

        rc   As   Short   :
                0=正常終了
                2=パラメータエラー
                8=タイムアウト（1 秒）
                9=重大エラー（詳細は P54 を参照）
               14=デバイスはオープンされていない
               19=mutex 排他制御エラー
```

注意事項： リンクの状態（注）は P49 を参照。

4.14. GetLogInfo

機能： 自ノードの現在の FL-net ログ情報を返します。

FL-net プロトコルで定義されるログ情報の「必要」「任意」の全情報をサポートします。

詳細は JEMA 規格の『実装ガイドライン』JEM-TR213 を参照願います。

記述： rc = GetLogInfo(logbuf , dev_no)

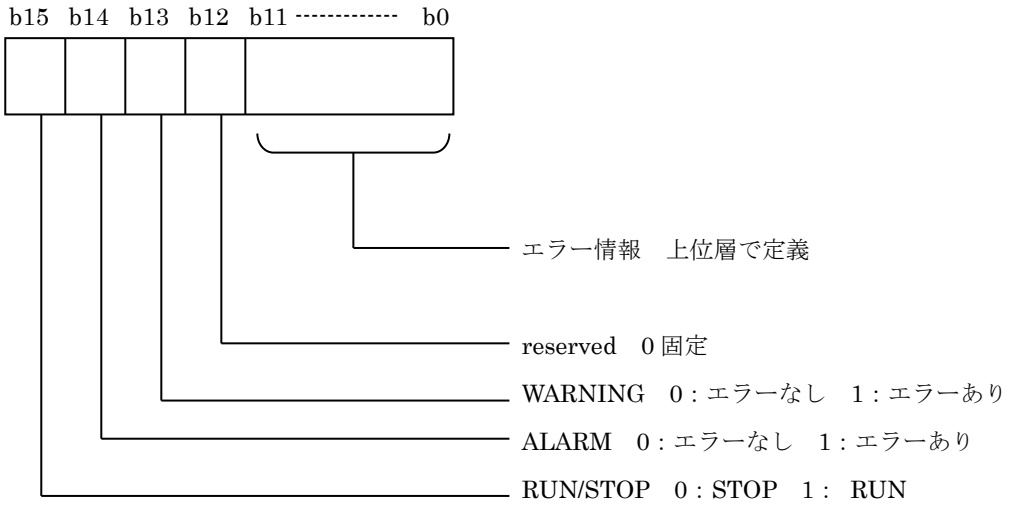
```
ByRef  logbuf  As  LogInfo      ： ログデータ情報のユーザ定義型
Structure LogInfo
    Log_buf (127) As Integer
End Structure
ByVal  dev_no  As  Short        ： デバイス No.（0～3）

  rc  As  Short  ：
        0=正常終了
        2=パラメータエラー
        8=タイムアウト（1 秒）
        9=重大エラー（詳細は P54 を参照）
       14=デバイスはオープンされていない
       19=mutex 排他制御エラー
```

注意事項：

4.15. SetUplStatus

機能： この関数で指定した上位層ステータスは、ＦＡリンクヘッダーの ULS にセット
されます。
FL-PCI カードは、初期化コマンドを受け取った時に RUM／STOP Bit=0 をセットします。
（ULS ディフォルト値=0000h）
上位層の状態が変化した時にこの関数を用いてＦＡリンクヘッダーの
ULS の値を書き換える必要があります。

ULS（上位層の状態）



記述： rc = SetUplStatus(uls, dev_no)
ByVal uls As Short ： 上位層ステータス
ByVal dev_no As Short ： デバイス No.（0～3）

rc As Short ：
0=正常終了
2=パラメータエラー
8=タイムアウト（1 秒）
9=重大エラー（詳細は P54 を参照）
14=デバイスはオープンされていない
19=mutex 排他制御エラー

注意事項：

4.16. GetErrInfo

機能：　重大エラーが発生した時に、その詳細情報を取り込みます。

記述：　rc = GetErrInfo(err_inf, dev_no)

ByRef　err_inf　As　ErrInfo　　：エラー情報ユーザ定義型

Structure ErrInfo

err_code　As　Short　　：エラーコード（注）

length　As　Short　　：メッセージ長

err_msg(127)　As　Byte　　：メッセージ（ASCII code）

End Structure

ByVal　dev_no　As　Short　　：デバイス No.（0～3）

rc　As　Short　：

0=正常終了

2=パラメータエラー

14=デバイスはオープンされていない

19=mutex 排他制御エラー

注意事項：エラーコードの内容

7F00h：システムエラー。詳細はエラーメッセージにセットされます。

7F01h：フラッシュメモリーエラー。（初期化時のみ発生）

7F02h：RAM テストエラー。（初期化時のみ発生）

7F03h：Ｅｔｈｅｒｎｅｔテストエラー。（初期化時のみ発生）

7F04h：ＥＥＰＲＯＭサムチェックエラー。（初期化時のみ発生）

7F05h：ＣＰＵ　ＢＵＳエラーが発生した。

7F06h：イリーガルインストラクションが発生した。

7F07h：ボードタイプとファームウェアが一致しない。

4.17. DeviceClose

機能： 初期化でオープンしたデバイスをクローズします。
アプリケーション終了時と FL-PCI カード再初期化の時には、この関数を
呼んで下さい。
デバイスクローズ時の FL-PCI カードを RESET 状態にするかどうかを引数で指定します。

記述： rc = DeviceClose(dev_no, reset_f)

ByVal dev_no As Short : デバイス No.（0～3）
ByVal reset_f As Short : FL-PCI カードの RESET 指示
（0：RESET なし、1：RESET あり）
rc As Short :
0=正常終了
15=デバイスクローズエラー
19=mutex クローズエラー

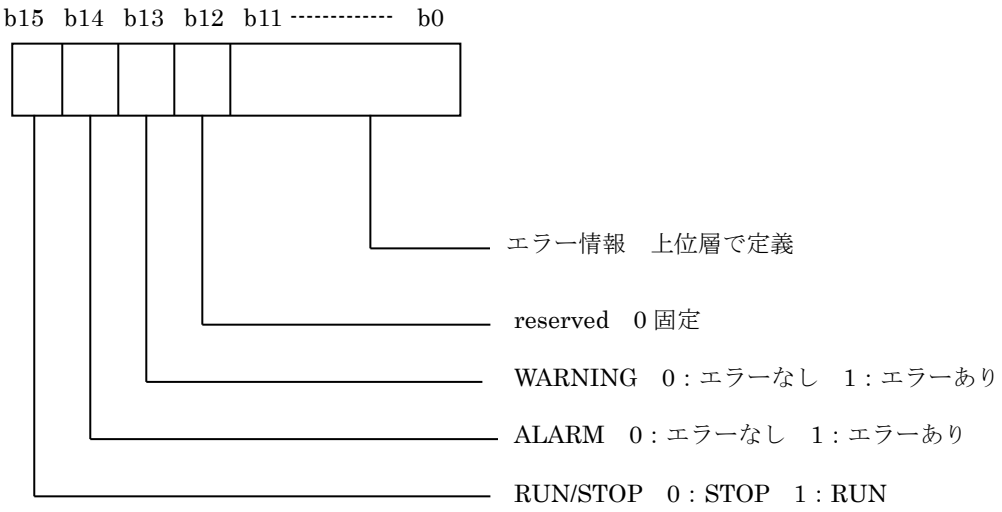
注意事項：

4.18. ReadUls_All

機能：　ノード No.1～254 の ULS（上位層の状態）と LKS（リンクの状態）を得ます。
　　離脱中のノードの ULS,LKS は 0 でセットされます。

記述：　rc = ReadUls_All (ulsInfo, dev_no)
　　ByRef　ulsinfo　 As　 UlsInfo
Structure UlsInfo
　　uls(253) As　Short　 :　uls(0)にはノード 1 の ULS がセットされます。
　　lks(253) As　Byte　 :　lks(0)にはノード 1 の LKS がセットされます
End Structure
ByVal　dev_no　As　Short　 :　デバイス No.（0～3）
rc　As　Short　:
　　0=正常終了
　　2=パラメータエラー
　　8=タイムアウト（1 秒）
　　9=重大エラー（詳細は P54 を参照）
　　14=デバイスはオープンされていない
　　19=mutex 排他制御エラー

ULS（上位層の状態）



LKS（リンクの状態）
　P49 を参照ください。

4.19. Dll_Version

機能： DLL の現在のバージョン No.を返します。

記述： dll_ver = Dll_Version()

dll_ver As Short : バイナリデータ（01～ ）

注意事項：

デバイスオープンなしで実行ができます。

4.20. Firmware_Version

機能： FL-PCI カードの現在のファームウェアバージョン No.を返します。

記述： rc =Firmware_Version (version, dev_no)

ByRef version As Short ：バイナリデータ

Bit0～7： ファームウェアバージョン No. (01～255)

Bit8～15： 00h=FL-PCI/V2 （FL-net Ver1）

0Ah= FL-PCI/V2 (FL-net Ver2)

14h= FL-PCI/V2-100 または 100L （FL-net Ver2)

1Eh= FL-PCIe または FL-PCI/V2 100L(FL-net Ver3)

ByVal dev_no As Short ： デバイス No.（0～3）

rc As Short ：

0=正常終了

2=パラメータエラー

8=タイムアウト（1 秒）

9=重大エラー（詳細は P54 を参照）

14=デバイスはオープンされていない

19=mutex 排他制御エラー

注意事項：

この機能は、初期化が正常終了した後に行ってください。

旧製品 FL-PCI/V2 の Ver.47(FL-net Ver1 対応品)と Ver.1004(FL-net Ver2 対応品)以前のバージョンは本機能をサポートしていません。

4.21. Set_Speed

機能：　イーサネットの通信速度を指定します。

記述：　rc = Set_Speed (speed, dev_no)

ByVal　speed　As　Integer　：　0：10Mbps　1：100Mbps　2：Auto Negotiation

（注）

0：10Mbps のとき、半 2 重固定になります。

1：100Mbps のとき、半 2 重固定になります。

2：Auto Negotiation の場合は、

10Mbps／100Mbps、、半 2 重／全 2 重は相手装置によって

自動選択となります。

ByVal　dev_no　As　Short　：デバイス No.（0～3）

rc　As　Short　：

0=正常終了

2=パラメータエラー

注意事項：

Set_Speed()は、関数 FL_Initialize()を Call する前に行ってください。

Set_Speed()を行わない場合は、ディフォルト設定である 10Mbps の設定になります。

1 枚の FL-PCI カードへデバイス No.別に 2 種類の動作モードを指定することはできません。

この機能は、新 FL-PCI/V2-100 でのみ有効です。旧 FL-PCI /V2 では 10Mbps 固定となります。

4.22. Get_Speed

機能： 現在設定されているイーサネットの通信速度を取得します。
Set_Speed()で Auto Negotiation を選択した場合は、通信モードは相手先に合わせて自動設定されます。
Get_Speed()にて現在設定されている通信モードを知ることができます。

記述： rc = Get_Speed (speed, duplex, dev_no)

ByRef speed As Integer : 0：10Mbps 1：100Mbps

ByRef duplex As Integer : 0：半 2 重 1：全 2 重

ByVal dev_no As Short : デバイス No.（0～3）

rc As Short :

0=正常終了
2=パラメータエラー
8=タイムアウト（1 秒）
9=重大エラー（詳細は P54 を参照）
14=デバイスはオープンされていない
19=mutex 排他制御エラー
20=ケーブル外れのため設定が未完

注意事項：

Get_Speed()は、関数 FL_Initialize() の後に行ってください。

この機能は、新 FL-PCI/V2-100 および 100L でのみ有効です。旧 FL-PCI /V2 ではタイムアウトエラーとなります。