



Ethernet/IP ドライバ

API リファレンスマニュアル

株式会社マイクロネット

<http://www.mnc.co.jp>

TEL: +81(0)299-90-1733

FAX: +81(0)299-90-8557

目次

1. 概要	1
1.1. システム構成	1
1.2. 動作要件	1
1.3. ユーティリティソフト	1
1.4. 開発環境	2
1.5. 通信ボードの取り扱い方法	2
1.5.1. ロータリースイッチ設定	2
1.5.2. Ethernet/IP 回線への接続	2
2. cifX ドライバインストール	3
2.1. Windows 用ドライバ	3
2.2. INtime 用ドライバ	3
2.2.1. ドライバの起動設定	3
3. 通信ボード設定手順	4
3.1. ファームウェアダウンロード	4
3.2. SYCON コンフィグレーションツールによるハードウェア設定	5
3.3. INtime へのパス	8
4. Ethernet/IP 高レベル API について	9
1.1. 関数リスト	9
4.1.1. API 初期化・終了	9
4.1.2. サイクリック関数	9
4.2. API 関数	10
■ Init	10
■ Open	11
■ Close	13
■ Read	14
■ Write	16
4.3. Config 設定内容	18
4.4. 仮想 I/O スペース	19
■ 仮想 I/O スペース構成	19
■ 統計情報	21
■ 情報領域	21
4.5. エラーコードリスト	23

1. 概要

本ドキュメントは、Ethernet/IP プロトコルによる通信を実現するための API ライブラリについて説明しています。本 API を利用することで Ethernet/IP のスキャナ機能を実装可能です。

1.1. システム構成

Ethernet/IP 高レベル API を利用した通信環境構成は以下の通りです。

- 1) 通信ボード： Hilscher 製 CIFS 50-RE
- 2) 制御ドライバ： Hilscher 製 CIFS INtime Driver
- 3) 補助ツール： Hilscher 製 SYCON コンフィグレーションツール

1.2. 動作要件

1)	INtime v6.4.22130.3 以降	INtime アプリケーション実行に Runtime が必要です
2)	Hilscher cifXDriver for Windows 2.6.1.0 以降	Windows 上で cifX ボードの設定を行うために使用します
3)	Hilscher cifXDriver for INtime 2.3.0.0 以降	INtime から cifX ボードへアクセスするために使用します
4)	Ethernet/IP スキャナファームウェア Cifxeim.nxf (v2.11.0.3)	cifX 50-RE ボードを Ethernet/IP のスキャナとして利用するために使用します

1.3. ユーティリティソフト

1)	cifX Driver Setup Utility	cifX ボードへのファームウェアダウンロードに使用します
2)	SYCON コンフィグレーションツール	cifX ボードのコンフィグレーションに使用します

1.4. 開発環境

Ethernet/IP 高レベル API を利用したアプリケーション開発に必要なものは以下の通りです。

	項目		説明
1.	lib	HiETHIP.rsl	Ethernet/IP 高レベル API の実行時ライブラリ
2.		HiETHIP.lib	Ethernet /IP 高レベル API の開発時ライブラリ
3.	header	HiETHIP.h	Ethernet /IP 高レベル API のヘッダファイル
4.	Sample	HiETHIP_sample.rta	Ethernet/IP 高レベル API を利用するサンプルプログラムです
5.	Source	HiETHIP_sample.sln	EtherCAT スレーブ I/F 本体ソースコードです(VisualStudio2012)

Hilscher から取得したファイルは以下に保存してあります。

① Windows 用ドライバやファームウェア、ユーティリティツール類

file:///192.168.1.240/Jobs/JD0010(INtime 周辺ソフトウェア)/Hilscher/CIFX/HiAPI/trunk
/0060_Tools/01_Windows 関連/Communication_Solutions_2022-03-1.zip

② INtime 用ドライバ

file:///192.168.1.240/Jobs/JD0010(INtime 周辺ソフトウェア)/Hilscher/CIFX/HiAPI/trunk
/0060_Tools/ NXDRV-INtime_2021-05-1_V2.3.0.0.zip

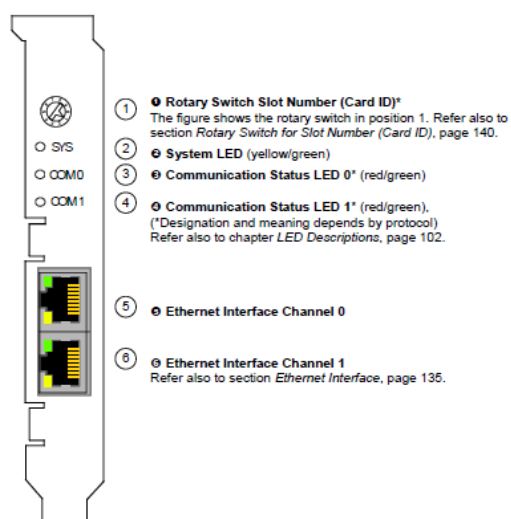
1.5. 通信ボードの取り扱い方法

1.5.1. ロータリースイッチ設定

ロータリースイッチは"0"を設定してください。

1.5.2. Ethernet/IP 回線への接続

Ethernet Interface Channel 0 ポートをネットワークハブに接続します。



2. cifX ドライバインストール

2.1. Windows 用ドライバ

1.4 節 ①で挙げたファイル内の以下に存在するインストーラを起動してインストールしてください。

¥Driver_¥Toolkit¥Device Driver (NXDRV-WIN)¥Installation¥cifx device driver setup.exe

2.2. INtime 用ドライバ

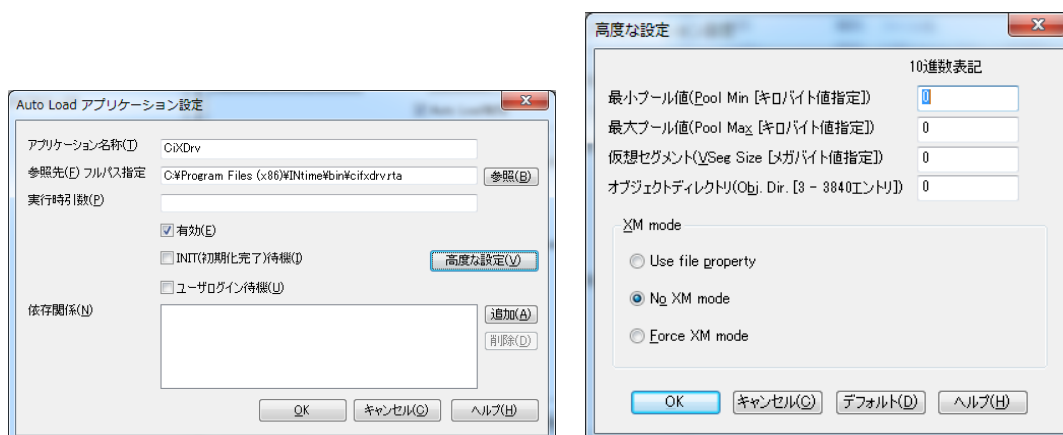
1.4 節②で挙げたファイル内の以下に存在するインストーラを起動してインストールしてください。

¥CD V2.3.0.0¥Installation¥cifx INtime driver setup.exe

2.2.1. ドライバの起動設定

INtime 用 cifX ドライバ利用時は以下の通り Autoload の設定を行います。

No.	項目名	値
1	アプリケーション名称	CifXDrv
2	参照先	C:¥Program Files (x86)¥INtime¥bin¥cifxdrv.rta
3	自動起動	有効
4	高度な設定 - XM mode	NO XM mode



3. 通信ボード設定手順

cifX 50-RE ボードを Ethernet/IP 通信のスカナ(マスタ)として設定するための手順を示します。設定を行う際はボードを Windows 割り当てにしてください。

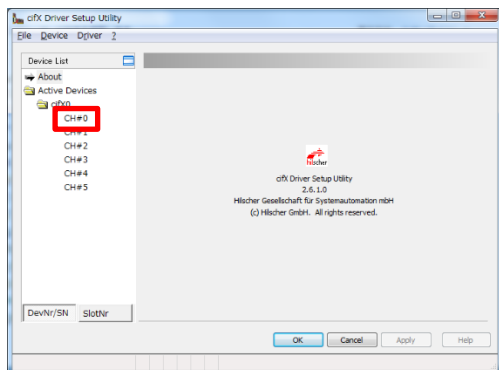
3.1. ファームウェアダウンロード

cifX 50E-RE ボード上の NetX チップに EtherCAT スレーブのファームウェアをダウンロードする手順を以下に示します。

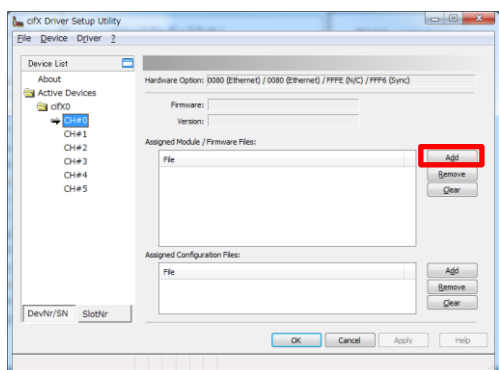
- ① コントロールパネルを開き、一覧の中から cifX Setup を選択します。



- ② cifX Driver Setup utility ダイアログが開くので、Device List から CH#0 を選択します。

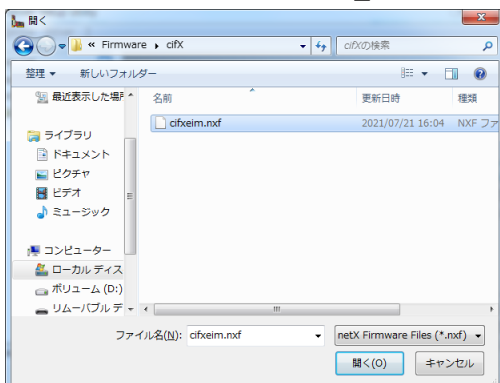


- ③ 右ペインに割り当てるファームウェアの設定が表示されるので、上部の Add ボタンをクリックします。

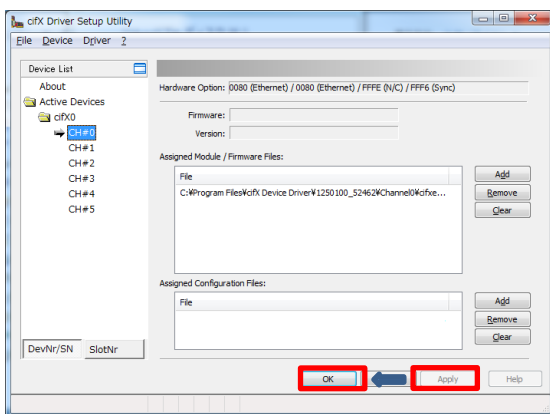


- ④ .nxf ファイルを選択するダイアログが開くので、cifX ドライバ(Windows)のインストールで利用した Communication Solution DVD フォルダ内の以下ファイルを選択し、開くボタンをクリックします。

firmware&EDS¥COMSOL_EIM V2.11.0.3¥Firmware¥cifX¥cifxeim.nxf



- ⑤ 選択したファームウェアが右ペインのリスト内に表示されるので、Apply ボタン→OK ボタンの順にクリックします。

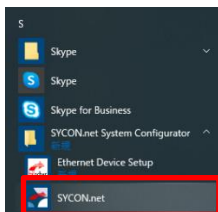


以上で、Ethernet/IP スキャナのファームウェアダウンロードの手順は完了です。

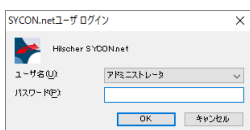
3.2. SYCON コンフィグレーションツールによるハードウェア設定

SYCON コンフィグレーションツールを利用して、Ethernet/IP スキャナの詳細設定を行う手順を以下に示します。

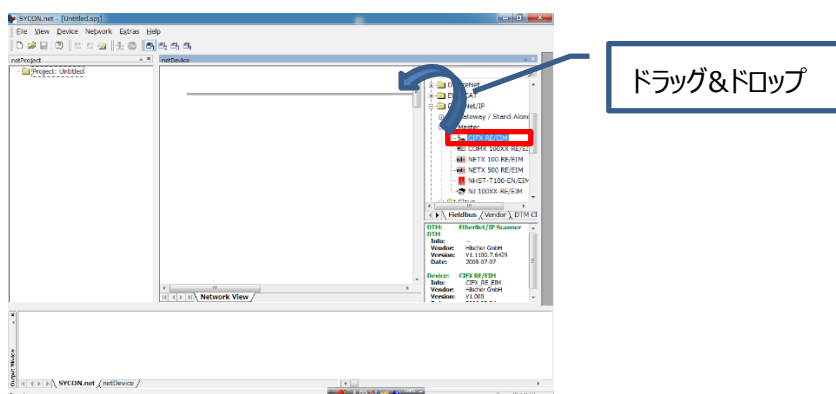
- ① Windows メニューから SYCON.net system configuration > SYCON.net をクリックします。



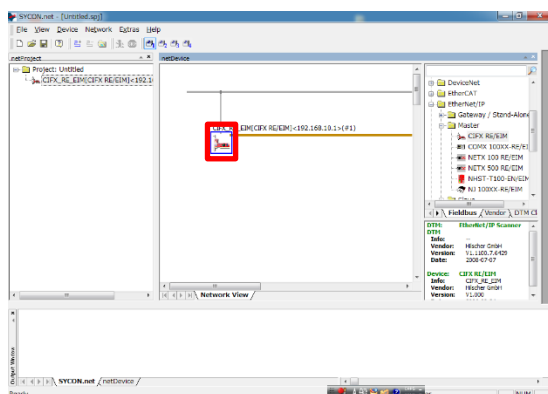
- ② 初回はログイン用のユーザ名とパスワード登録ダイアログが表示されますので、任意のユーザ名とパスワードを設定します。2 回目以降はログインユーザ名とパスワードの入力が求められます。



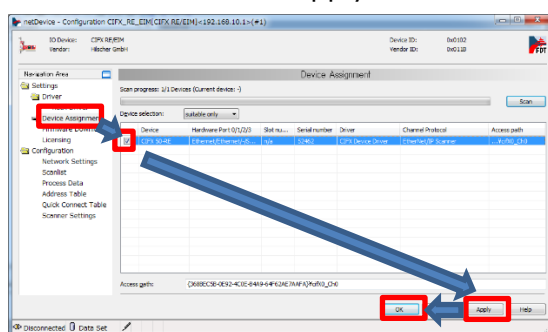
- ③ ログイン後、SYCON.net コンフィグレーションツールのウィンドウが開きますので、右上ペインのプロトコルフォルダの中から[Ethernet/IP] > [Master] > [CIFX RE/EIM]をクリックし、中央ペインのベースライン上にドラッグアンドドロップします。



- ④ 中央ペイン ベースラインの下に CIFX RE/EIM アイコンが表示されるのでダブルクリックします。

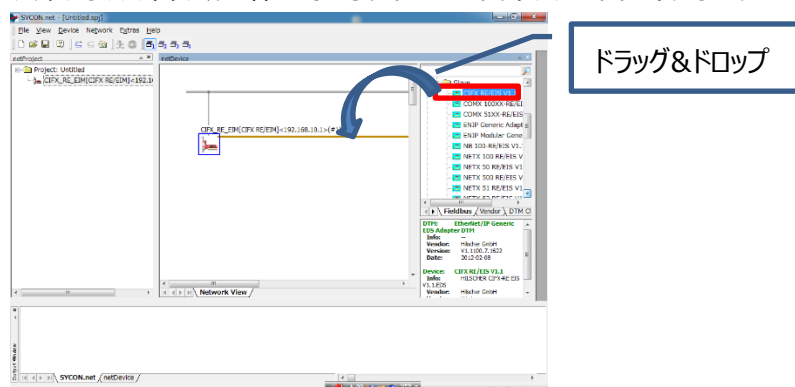


- ⑤ コンフィグレーションダイアログが表示されますので、Navigation Aria から[Settings] > [Device Assignment]を選択します。続いて右ペインのリスト内に表示される CIFX 50-RE のチェックボックスをクリックします。最後に Apply ボタン→OK ボタンの順にクリックします。

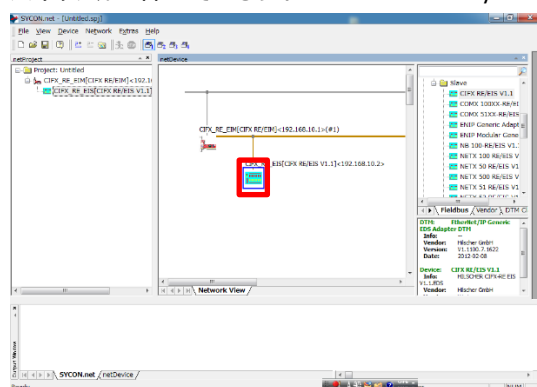


- ⑥ 続いて、通信対象の Ethernet/IP アダプタの設定を行います。

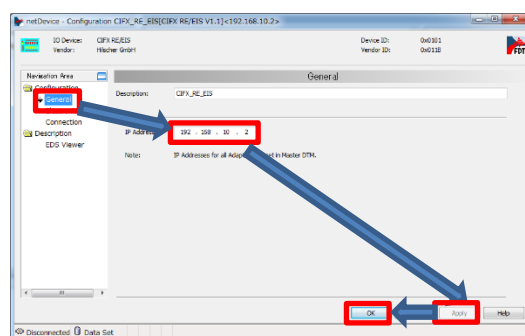
右上ペインのプロトコルフォルダの中から[Ethernet/IP] > [Slave] > [CIFX RE/EIS v1.1]をクリックし、スキャナから伸びているライン上にドラッグアンドドロップします。



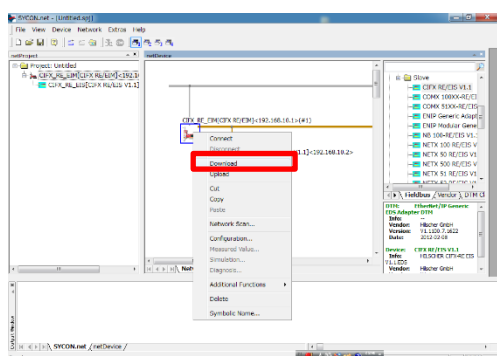
- ⑦ スキャナから伸びているライン上に CIFX RE/EIS v1.1 が表示されるのでダブルクリックします。



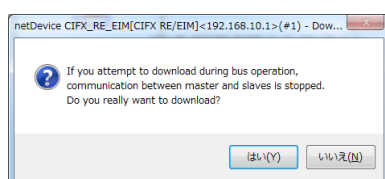
- ⑧ コンフィグレーションダイアログが表示されますので、Navigation Aria から[Configuration] > [General]を選択します。続いて右ペインの IP Settings 内に表示される IP Address に接続されるアダプタの IP アドレスを設定します。最後に Apply ボタン→OK ボタンの順にクリックします。



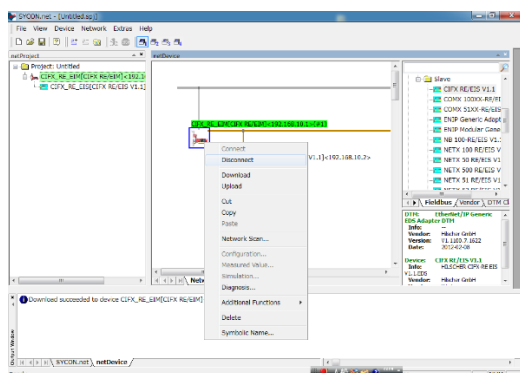
- ⑨ Ethernet/IP スキャナと接続されるアダプタの設定が完了しました。最後に通信ボードへ設定をダウンロードするため、中央ペインの CIFS RE/EIM アイコンを右クリックして Download を選択します。



- ⑩ 確認ダイアログが表示されるので、「はい」を選択します。



- ⑪ CIFS RE/EIM アイコン上で右クリックし、[ディスコネクト]を選択します。



- ⑫ 必要に応じて設定用のプロジェクトを保存し、SYCON コンフィグレーションツールを終了します。

以上で、SYCON コンフィグレーションツールによる Ethernet/IP スキャナの詳細設定手順は完了です。

3.3. INtime へのパス

INtime へのデバイスのパスは INtime Device Manager にて、「INtime に移す—IRQ なし(ポーリング)」を指定してください。

パスの際に INtime cifX PCI/PCIe Device ドライバのインストール確認が表示されるので、「はい」を選択し、その後 PC の再起動を実施してください。

4. Ethernet/IP 高レベル API について

基本的な Ethernet/IP 通信制御を 5 つの API 関数で実現します。

1.1. 関数リスト

以下に提供される API のリストを提供します：

4.1.1. API 初期化・終了

API 関数	説明
<u>Init()</u>	API 初期化、ドライバ開始

4.1.2. サイクリック関数

ドライバ・通信チャンネル動作制御

API 関数	説明
<u>Open()</u>	通信チャンネル オープン
<u>Close()</u>	通信チャンネル クローズ

仮想メモリ設定・参照

API 関数	説明
<u>Read()</u>	入力領域・情報領域・統計情報 読み込み
<u>Write()</u>	出力領域 書き込み

4.2. API 関数

■ Init

API ライブラリの初期化をします。引数に指定された自ノード設定情報をもとに、設定変更を行います。また、ドライバをオープンします。

シンタックス

```
DWORD Init (char *szIFName, char *Config, IFHANDLE *hIF,
            DWORD dwTimeout );
```

引 数

*szIFName	現在未使用。NULL を指定してください。
*Config	自ノード設定情報を指定します (Config 設定内容参照) デフォルト値を利用する場合は、NULL を指定してください。
*hIF	インターフェースハンドル返却用 (Open 時に必要です)
dwTimeout	現在未使用。0 を指定してください。

戻り値

DWORD [Ethernet/IP HI API ステータスコード](#)を返却します。

エラー

ER_HI_OK	正常終了
ER_HI_BAD_IFHANDLE	インターフェースハンドル不正
ER_HI_DRV_NOT_INIT	ドライバ未初期化
ER_HIEP_INV_BOARD	通信ボード未検出
ER_HIEP_FUNC_FAILED	API 呼び出し失敗

注釈

- この関数はアプリケーションが本ドライバの API を使用する場合はじめに一度コールされる必要があります。二回以上コールした場合は、自ノード設定変更のみ行います。

コールサンプル

Init コール例

```
DWORD status;
IFHANDLE hIF;
// 起動時に一度だけコールします:
status= Init (NULL, "rwapi_timeout=500|stateapi_timeout=500", &hIF, 5000 );
if (status != ER_HI_OK ){
    printf("Ethernet/IP Init error [%08x]¥n", status);
}
```

■ Open

Ethernet/IP 通信チャンネルをオープンします。

シンタックス

```
DWORD Open (IFHANDLE hIF, char *Target, RSIHANDLE *hRsi,  
            DWORD dwTimeout);
```

引 数

hIF	Init()で取得したインターフェースハンドルを指定します。
*Target	通信に利用するチャンネルを指定します。 (2022/12 現在、“ch0”のみ有効です。) デフォルトの設定を利用する場合は NULL を指定してください。
*hRsi	接続インスタンスハンドル格納用 (Close,Read,Write に必要です)
dwTimeout	現在未使用。

戻り値

DWORD	Ethernet/IP HI API ステータスコード を返却します。
--------------	---

エラー

ER_HI_OK	正常終了
ER_HI_BAD_IFHANDLE	インターフェースハンドル不正
ER_HI_BAD_RSIHANDLE	接続インスタンスハンドル不正
ER_HIEP_INV_HANDLE	ドライバ/チャンネルハンドル不正
ER_HIEP_FUNC_FAILED	API 呼び出し失敗
ER_HIEP_READ_STATE	チャンネルステート読み込み失敗
ER_HIEP_NOT_READY	通信チャンネル未動作
ER_HIEP_NOT_RUNNIG	通信チャンネル未設定
ER_HIEP_STATE_ON_TIMEOUT	通信チャンネル ステート ON タイムアウト
ER_HIEP_BUS_ON_TIMEOUT	通信チャンネル バスステート ON タイムアウト
ER_HI_OPEN_FAILED	API オープン失敗

注釈

- オープンに繰り返し失敗する場合は、設定ツールにてスキャナ / アダプタの設定を見直してください。

コールサンプル

Open コール例

```
DWORD status;  
RSIHANDLE hRsi;  
  
// 通信チャンネルをオープンします  
status = Open ( hIF, "ch0", &hRsi, 5000 );  
if (status != ER_HI_OK ){  
    printf("Ethernet/IP Open error [%08x]¥n", status);  
}
```

■ Close

Ethernet/IP 通信チャンネルをクローズします。

シンタックス

```
DWORD Close ( RSIHANDLE hRsi );
```

引 数

hRsi Open()で取得した接続インスタンスハンドルを指定します。

戻り値

DWORD [Ethernet/IP HI API ステータスコード](#)を返却します。

エラー

ER_HI_OK	正常終了
ER_HI_BAD_RSIHANDLE	接続インスタンスハンドル不正
ER_HIEP_INV_HANDLE	通信チャンネルハンドル不正
ER_HIEP_FUNC_FAILED	API 呼び出し失敗
ER_HIEP_STATE_OFF_TIMEOUT	通信チャンネル ステート OFF タイムアウト
ER_HIEP_BUS_OFF_TIMEOUT	通信チャンネル バスステート OFF タイムアウト
ER_HI_CLOSE_FAILED	API クローズ失敗

コールサンプル

Close コール例

```
WORD status;  
  
// 通信チャンネルをクローズします  
status = Close ( hRsi );  
if (status != ER_HI_OK){  
    printf("Ethernet/IP Close error [%08x]¥n", status);  
}
```

■ Read

仮想 I/O スペース(入力領域・情報領域・統計情報)から指定バイトサイズのデータを読み取ります。

シンタックス

```
DWORD Read (RSIHANDLE hRsi, DWORD dwAddr, void *Buf,  
            DWORD dwSize, DWORD dwBufSize, DWORD dwTimeout );
```

引 数

hRsi	Open()で取得した接続インスタンスハンドルを指定します。
dwAddr	出力領域・情報領域・統計情報の仮想 I/O スペースアドレスを指定します。 (単位はバイトです。) <div> 入力領域 B 範囲 - 0x1700 ~ 0x2D7F 情報領域 範囲 - 0x3000 ~ 0x30A4 統計情報 範囲 - 0x4000 ~ 0x4027 </div>
*Buf	読み取ったデータを格納するメモリ領域へのポインタを指定します。 dwSize 指定値以上のサイズが確保されている必要があります。
dwSize	読み取りデータサイズを指定します。単位はバイトです。
dwBufSize	読み取ったデータを格納するメモリ領域のサイズを指定します。単位はバイトです。
dwTimeout	現在未使用。0 を指定してください。

戻り値

DWORD	Ethernet/IP HI API ステータスコード を返却します。
--------------	---

エラー

ER_HI_OK	正常終了
ER_HI_BAD_RSIHANDLE	接続インスタンスハンドル不正
ER_HI_NOT_OPENED	未オープンエラー
ER_HI_SIZE_ZERO	読み込みバッファサイズが 0
ER_HI_BUFFER_NULL	読み込み先バッファが NULL
ER_HI_BUFFERSIZE_ZERO	読み込み先バッファサイズが 0
ER_HI_BUFFERSIZE_SMALL	読み込み先バッファのバイトサイズが読み込みバイトサイズより小さい
ER_HI_SIZE_OVER	指定アドレス・サイズが領域範囲外
ER_HIEP_RESERVED_AREA	予約済領域(入力領域 A)へのアクセス
ER_HIEP_INV_HANDLE	通信チャンネルハンドル不正
ER_HIEP_FUNC_FAILED	API 呼び出し失敗
ER_HI_READ_FAILED	読み込み失敗

注釈

- 入力領域 A アドレス指定(0x16C0~0x16FF)はエラーとなります。

- 通信チャンネルオープン状態でのみ参照が可能です。
- [情報領域](#)からは通信チャンネル情報を参照することが出来ます。
- ケーブル断線などで通信異常が発生した場合、読み込まれたデータが全て 0 となります。
通信異常時も戻り値としては ER_HI_OK が返りますので、必要に応じて情報領域内の通信チャンネルエラーコードの値を利用した異常判定を実装してください。(こちらの値は通信異常時に 0 以外の値を返します。)

コールサンプル

Read コール例

```
DWORD status;

BYTE buf_inB[5760];           // 入力領域 B データ格納
BYTE buf_mgr[28];             // 統計情報 格納
BYTE buf_info[164];           // 情報領域 格納
// 入力領域 B の内容を参照します(先頭から 10byte)

status = Read ( hRsi, INPUT_B_START_ADDR, buf_inA, 10, sizeof(buf_inB), 0);
if (status != ER_HI_OK ){
    printf("Ethernet/IP Read error [%08x]¥n", status);
}

// 情報領域の内容を参照します(統計情報データを参照)
status = Read ( hRsi, STATISTICS_START_ADDR, buf_mgr, 28, sizeof(buf_mgr),
0);
if (status != ER_HI_OK ){
    printf("Ethernet/IP Read error [%08x]¥n", status);
}

// 情報領域の内容を参照します(NodeID=1 のデータを参照)
status = Read ( hRsi, INFOMATION_START_ADDR, &buf_info, sizeof(buf_info),
0);
if (status != ER_HI_OK ){
    printf("Ethernet/IP Read error [%08x]¥n", status);
}
```

■ Write

出力領域へ指定バイトサイズのデータを書き込みます。

シンタックス

```
DWORD Write (RSIHANDLE hRsi, DWORD dwAddr, void *Buf,  
             DWORD dwSize, DWORD dwTimeout);
```

引 数

hRsi	Open()で取得した接続インスタンスハンドルを指定します。
dwAddr	出力領域の 仮想 I/O スペース アドレスを指定します。 単位はバイトです。 出力領域 B 範囲 - 0x0040 ~ 0x16BF
*Buf	書き込むデータが格納されたメモリ領域へのポインタを指定します。 dwSize 指定値以上のサイズが確保されている必要があります。
dwSize	書き込みデータサイズを指定します。単位はバイトです。
dwTimeout	現在未使用。0 を指定してください。

戻り値

DWORD	Ethernet/IP HI API ステータスコード を返却します。
--------------	---

エラー

ER_HI_OK	正常終了
ER_HI_BAD_RSIHANDLE	接続インスタンスハンドル不正
ER_HI_NOT_OPENED	未オープンエラー
ER_HI_SIZE_ZERO	書き込みバッファサイズが 0
ER_HI_BUFFER_NULL	書き込み元バッファが NULL
ER_HI_SIZE_OVER	指定アドレス・サイズが領域範囲外
ER_HI_DISCONNECT	通信エラー
ER_HIEP_RESERVED_AREA	予約済領域(出力領域 A)へのアクセス
ER_HIEP_INV_HANDLE	通信チャンネルハンドル不正
ER_HIEP_FUNC_FAILED	API 呼び出し失敗
ER_HI_READ_FAILED	書き込み失敗

注釈

- 出力領域 A 範囲(0x0000~0x003F)への書き込みはエラーとなります。
- 出力領域 B の範囲を超えるアドレス・サイズ指定はできません。
- 通信チャンネルオープン状態でのみ書き込みが可能です。
- 書き込み時に ER_HI_DISCONNECT が発生すると、統計情報内通信ステータスが 0 になります。



書込データのバッファサイズよりも大きなサイズを指定しないでください。
ページフォールトが発生し、プログラムがクラッシュする可能性があります。

コールサンプル

Write コール例

```
DWORD status;

BYTE byte_buf[4] = {0x00,0x01,0x02,0x03};    // 書き込みデータ
QWORD outB_address = 0x0040                  // 出力領域 B 先頭アドレス
WORD size = 4;                                // 書き込みサイズ

// 出力領域 B に書き込みを行います
status = Write ( hRsi, outB_address, byte_buf, size, 0);
if (status != ER_HI_OK ){
    printf("Ethernet/IP Write error [%08x]¥n", status);
}
```

4.3. Config 設定内容

Init に渡す初期値設定 Config の指定内容についてご説明します。

- 設定値を変更する場合は「項目名=設定値」の形で、指定します。
例: "rwapi_timeout =10"
- 複数の設定を変更する場合、「|」で区切って指定します。
例: "rwapi_timeout=500|stateapi_timeout=500"
- 指定がない項目については、デフォルト値で初期化されます。

設定可能項目

項目名	内容	デフォルト値
rwapi_timeout	データ読み書き タイムアウト時間[ms]	100
stateapi_timeout	通信チャンネルオープン タイムアウト時間[ms]	100

設定変更例

```
DWORD status;  
IFHANDLE hIF;  
  
// 初期化处理(自ノード ID=1,トークン監視時間=5ms,最小許容フレーム間隔=10 に設定変更 )  
status = Init (NULL,"rwapi_timeout=500|stateapi_timeout=500",&hIF,5000 );  
if (status != ER_HI_OK ){  
    printf("Ethernet/IP Init error [%08x]¥n", status);  
}
```

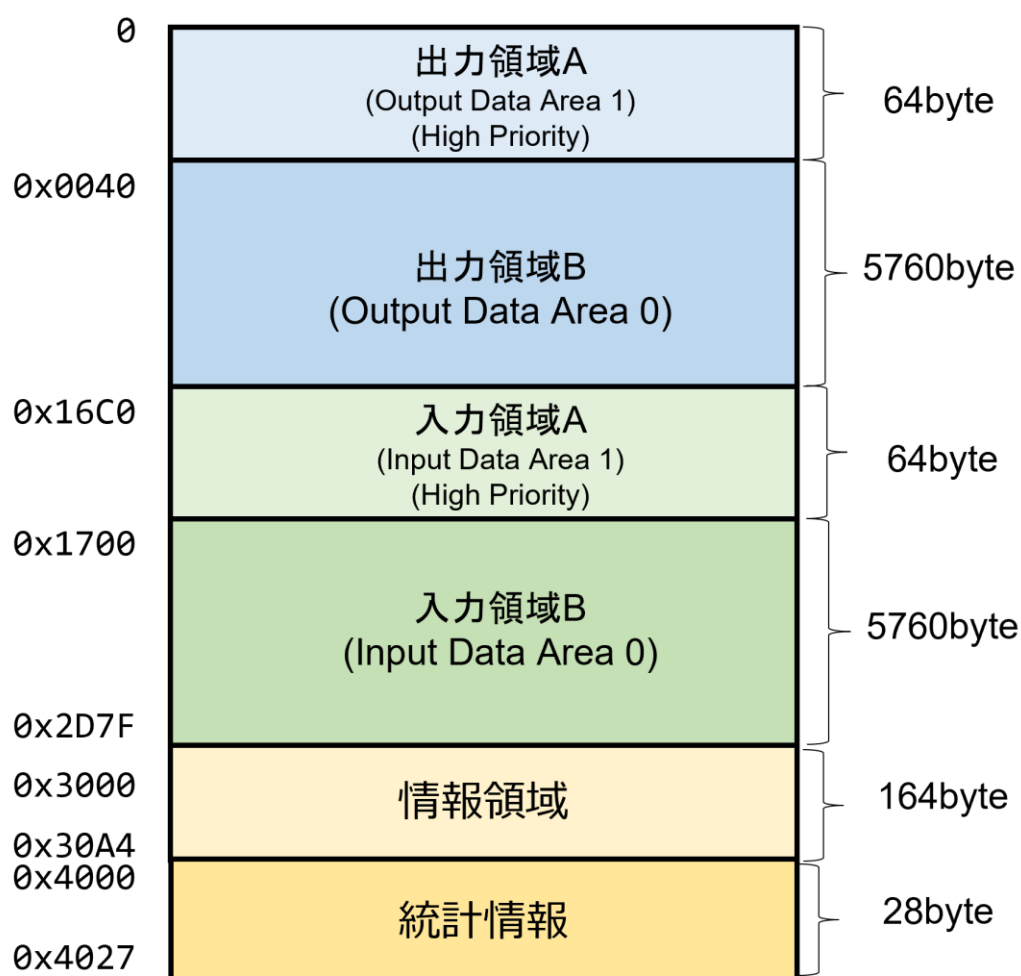
4.4. 仮想 I/O スペース

RSI HI API では、データの読み書きは仮想 I/O スペースアドレスでアクセスします。

■ 仮想 I/O スペース構成

仮想 I/O スペースは下記の領域で構成されます。

- 出力領域 A - 範囲: 0x0000 ~ 0x003F
- 出力領域 B - 範囲: 0x0040 ~ 0x16BF
- 入力領域 A - 範囲: 0x16C0 ~ 0x16FF
- 入力領域 B - 範囲: 0x1700 ~ 0x2D7F
- 情報領域 - 範囲: 0x3000 ~ 0x30A4
- 統計情報 - 範囲: 0x4000 ~ 0x4027



出力領域 A および入力領域 A は Explicit 通信用の予約領域となっています。
2022/12 現在、通信ボードのファームウェアが Explicit 通信に非対応のため、
本領域への読み込み/書き込みはできません。

各領域の先頭アドレス・終了アドレスは HiETHIP.h に定義しています。

#define OUTPUT_A_START_ADD	0x0000	// 出力領域 A 先頭アドレス
#define OUTPUT_A_END_ADD	0x003F	// 出力領域 A 終了アドレス
#define OUTPUT_B_START_ADD	0x0040	// 出力領域 B 先頭アドレス
#define OUTPUT_B_END_ADD	0x16BF	// 出力領域 B 終了アドレス
#define INPUT_A_START_ADD	0x16c0	// 出力領域 A 先頭アドレス
#define INPUT_A_END_ADD	0x16FF	// 出力領域 A 終了アドレス
#define INPUT_B_START_ADD	0x1700	// 出力領域 B 先頭アドレス
#define INPUT_B_END_ADD	0x247F	// 出力領域 B 終了アドレス
#define INFOMATION_START_ADD	0x3000	// 情報領域 先頭アドレス
#define INFOMATION_END_ADD	0x30A4	// 情報領域 終了アドレス
#define STATISTICS_START_ADD	0x4000	// 統計情報 先頭アドレス
#define STATISTICS_END_ADD	0x401B	// 統計情報終了アドレス

■ 統計情報

【範囲：0x4000 ～ 0x4027】

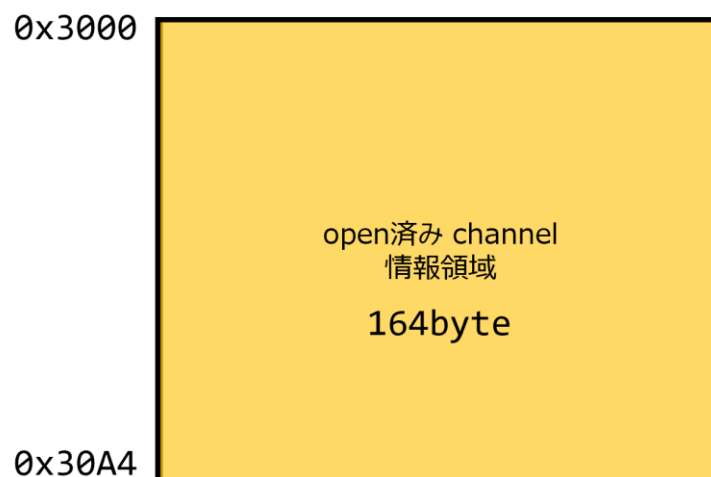
この情報領域には、接続ステータスや API のコール回数、最終エラーコードデータ等が格納されています。

アドレス	説明	サイズ(byte)
0x4000	通信ステータス (0:未通信/通信異常 1:通信正常)	4
0x4004	Init コール回数	4
0x4008	Open コール回数	4
0x400C	Close コール回数	4
0x4010	Read コール回数	4
0x4014	Write コール回数	4
0x4018	最終エラーコード情報	4

■ 情報領域

【範囲：0x3000～0x30A4】

この情報領域にはオープン済みの接続チャンネル情報が格納されています。



情報領域のデータ内容は以下の通りです。

アドレス	説明	サイズ(byte)
0x00	通信ボード名	16
0x10	通信ボード エイリアス番号	16
0x20	デバイス番号	4
0x24	シリアルナンバー	4
0x28	ファームウェア メジャー-Ver.	2
0x2A	ファームウェア マイナー-Ver.	2
0x2C	ファームウェア ビルド番号	2
0x2E	ファームウェア リビジョン番号	2
0x30	ファームウェア名 長さ	1
0x31	ファームウェア名	63
0x70	ファームウェア ビルド年	2
0x72	ファームウェア ビルド月	1
0x73	ファームウェア ビルド日	1
0x74	通信チャンネルエラーコード	4
0x78	オープン回数	4
0x7C	通信パケット 送信成功回数	4
0x80	通信パケット 受信成功回数	4
0x84	システム予約 使用不可	4
0x88	入力領域番号	4
0x8C	出力領域番号	4
0x90	システム予約 使用不可	4
0x94	システム予約 使用不可	4
0x98	システム予約 使用不可	4
0x9C	システム予約 使用不可	4
0xA0	システム予約 使用不可	4

4.5. エラーコードリスト

Ethernet/IP HI API 関数コールに失敗した場合、以下に示すエラーコードを返却します：

二一モニック	エラー コード値	説明
ER_HI_OK	0x00000000	関数は正常に終了しました。
ER_HI_TIMEOUT	0xC0000003	ドライバ起動待機タイムアウトエラー
ER_HI_NOT_OPENED	0xC0000004	未オープンエラー
ER_HI_OPEN_FAILED	0xC0000005	オープン失敗
ER_HI_BAD_IFHANDLE	0x80000000	インターフェースハンドル不正
ER_HI_BAD_RSIHANDLE	0x80000001	接続インスタンスハンドル不正
ER_HI_DRV_NOT_INIT	0x80000002	ドライバの初期化に失敗しました。 ドライバが起動しているか確認してください。
ER_HI_SIZE_ZERO	0x80000004	読み込み・書き込みバッファサイズに 0 が指定されました。
ER_HI_SIZE_OVER	0x80000005	指定アドレス・サイズが、仮想 I/O 領域の範囲を超えています。
ER_HI_BUFFER_NULL	0x80000006	読み込み・書き込みバッファアドレスに NULL が指定されました。
ER_HI_BUFFERSIZE_ZERO	0x80000007	読み込み先バッファサイズに 0 が指定されました。
ER_HI_BUFFERSIZE_SMALL	0x80000008	読み込み先バッファのバイトサイズが読み込みバイトサイズより小さい値で指定されました。
ER_HI_CLOSE_FAILED	0x80000009	クローズ失敗
ER_HI_READ_FAILED	0x8000000A	メモリ読込失敗
ER_HI_WRITE_FAILED	0x8000000B	メモリ書込失敗
ER_HI_DISCONNECT	0x8000000C	書き込み時通信エラー
ER_HIEP_FUNC_FAILED	0x80001F00	API 呼び出し失敗
ER_HIEP_INV_BOARD	0x80001F01	通信ボード未検出
ER_HIEP_INV_CHANNEL	0x80001F02	オープンで指定したチャンネル番号不正
ER_HIEP_INV_HANDLE	0x80001F03	ドライバ/通信チャンネルハンドル不正
ER_HIEP_READ_STATE	0x80001F04	通信チャンネル ステータス取得失敗
ER_HIEP_INIT_FAILED	0x80001F05	ドライバの初期化に失敗しました。 ドライバが起動しているか確認してください。
ER_HIEP_NOT_READY	0x80001F06	通信チャンネル未動作
ER_HIEP_NOT_RUNNIG	0x80001F07	通信チャンネル未設定
ER_HIEP_STATE_ON_TIMEOUT	0x80001F08	通信チャンネル ステート ON タイムアウト
ER_HIEP_STATE_OFF_TIMEOUT	0x80001F09	通信チャンネル ステート OFF タイムアウト
ER_HIEP_BUS_ON_TIMEOUT	0x80001F0A	通信チャンネル バスステート ON タイムアウト
ER_HIEP_BUS_OFF_TIMEOUT	0x80001F0B	通信チャンネル バスステート OFF タイムアウト
ER_HIEP_BUS_ON_FAILED	0x80001F0C	通信チャンネル バスステート ON 失敗
ER_HIEP_RESERVED_AREA	0x80001F0D	予約領域(入力領域 A / 出力領域 A)へのアクセス

更新履歴

版	日付	備考
1.0	2022.12.19	初版リリース

REALTIME SERVICE for Windows

Ethernet/IP ドライバ API リファレンスマニュアル

発行元：株式会社マイクロネット

TEL: +81(0)299-90-1733

FAX: +81(0)299-90-8557

- ・本書の著作権は、マイクロネットに帰属します。
- ・本書の内容、及び付属のソフトウェアの全部または一部を無断で転載することは禁止しております。
- ・本製品の内容については、将来予告なしに変更することがあります。
- ・本製品の内容について万が一ご不審な点や記載もれなどお気づきの点がございましたら、お手数ですが、当社までご連絡ください。
- ・Windows XP、Windows 7、Windows 8、Windows 10 等、Windows は、米国 Microsoft Corporation における登録商標です。
- ・Visual Studio、Visual C++等は、米国、およびその他の国における Microsoft Corporation の登録商標です。
- ・Intime は米国 TenAsys における登録商標です。
- ・その他、記載されている会社名、製品名は、各社の商標又は登録商標です