

InTouch I/O Server

Ver 1.1

取扱説明書

株式会社 **マイクロネット**

目次

1.	はじめに	- 4 -
2.	動作環境	- 5 -
3.	セットアップ	- 5 -
4.	システム構成	- 6 -
5.	動作設定	- 7 -
5. 1	I/O サーバ	- 7 -
5. 2	I/O サーバ設定画面	- 8 -
5. 3	I/O サーバ詳細設定画面	- 9 -
6.	アプリケーションの起動方法	- 11 -
6. 1	I/Fドライバの起動	- 11 -
6. 2	INtime 側ユーザーアプリケーション	- 11 -
6. 3	I/Oサーバ及び InTouch アプリケーション (Viewer) の起動	- 11 -
7.	アプリケーション作成手順	- 12 -
7. 1	InTouchとINtimeの関係	- 12 -
7. 2	InTouch側	- 12 -
7. 3	INtime側ユーザーアプリケーション	- 13 -
8.	INtime I/Fライブラリ	- 14 -
9.	その他	- 18 -
9. 1	InTouchタグ	- 18 -
9. 2	I/Oサーバについて	- 18 -
9. 3	INtime 側アプリケーションについて	- 18 -
9. 4	INtime側 API テスト AP について	- 18 -
10.	サンプルプログラム	- 19 -
10. 1	起動方法	- 19 -
10. 2	サンプル画面	- 19 -
11.	付録	- 22 -
11. 1	Visual Studio.NET2003 以上でのサンプルプログラムの実行方法	- 22 -

ご確認ください

本製品の構成並びに内容物に関して全てが揃っていることをご確認願います。
万一、不足品や破損している場合には、販売店または弊社までご連絡下さい。

製品構成

- 取扱説明書 1部
- セットアップCD 1枚

セットアップCDの内容

setup.exe

取扱説明書.PDF

1. はじめに

この度は、本製品をご購入いただきまして、有難うございます。

本製品は、同一 PC 上の Windows 上で動作する監視制御ツール Wonderware 社製 InTouch(※1)と Windows にハードリアルタイム機能を拡張する TenAsys 社製 INtime(※2)上で動作するユーザーアプリケーションとのインターフェースを可能とするソフトウェアです。

本製品は、大きく分けて下記の3つに分類されます。

- ①I/Fドライバ INtime 上で動作し、InTouch アプリケーションと INtime アプリケーションのインターフェースを行います。
- ②I/Fライブラリ INtime 上で動作させるユーザーアプリケーションにリンクすることにより、I/Fドライバが管理するデータアクセス並びにI/Oサーバからのイベント待ち等が可能となるライブラリです。
- ③I/Oサーバ Windows 上で動作し、InTouch 上で定義した外部入出力データ(I/O 型)を元に INtime 上で動作するI/Fドライバの管理するデータに対して、アクセスを行います。

(※1)InTouch とは？

Windows 上で動作する、産業用オートメーション、プロセス制御、監視用等のグラフィカル HMI(ヒューマン・マシン・インターフェース)です。

(※2)INtime とは？

Windows にハードリアルタイム機能を拡張することができ、Windows のみでは実現が難しかったハードリアルタイム処理が可能となります。

※ご注意

- ・本書の内容に関して、無断で転載することは禁止されています。
- ・本書の内容に関しては、将来予告なしに変更することがあります。
- ・本製品の運用を理由とする損失、逸失利益等の請求につきましては、いかなる責任も負いかねますので、あらかじめご了承下さい。
- ・MS、Microsoft は、米国 Microsoft Corporation の登録商標です。

Windows は、米国 Microsoft Corporation の登録商標です。

その他、本書中に使用している会社名及び製品名は、一般に各メーカーの商標または登録商標です。

2. 動作環境

動作環境につきましては、以下の製品の動作環境を御参照下さい。

同一の PC で以下の製品が動作する PC であれば、本製品も動作可能です。

Wonderwear 社 InTouch Ver9.5 以降

TenAsys 社 INtime Ver3.0 以降

3. セットアップ

セットアップには、本製品のセットアップCDを使用します。

セットアップするターゲットPCにセットアップCDを挿入し、CD 内の「setup.exe」をクリックすることで、セットアップが開始されます。開始後は、ガイダンスに従いセットアップを完了させて下さい。

※本製品のセットアップを行う際には、必ず「INtime」ならびに「InTouch」がセットアップされた状態で行ってください。「INtime」または「InTouch」がセットアップしていない場合には、本製品のセットアップを中止します。

本製品は、ターゲットPCにセットアップされている「INtime」のライセンスに対応した製品をセットアップします。

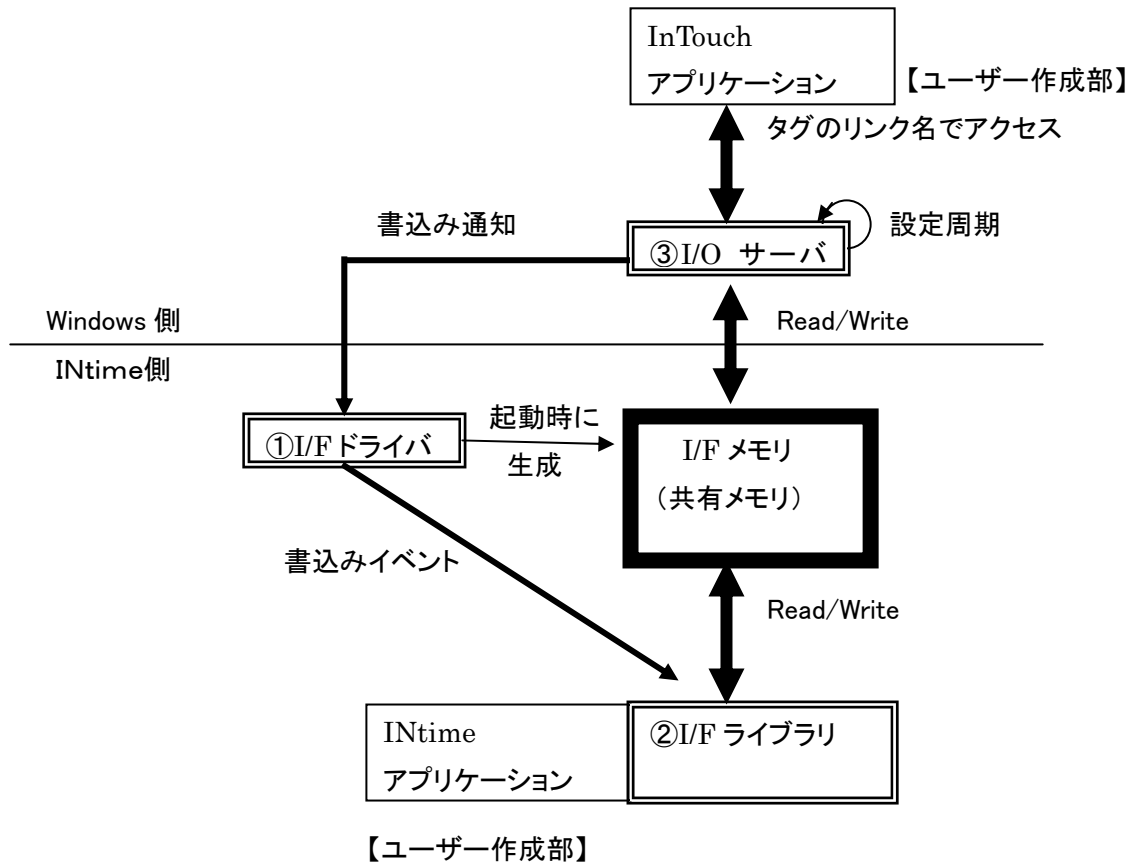
製品名称	ライセンス		デフォルトセットアップパス	名称
	DK	RT		
I/F ドライバ	○	○	C:\Program Files\InTouchIOSV	InDriver.rta
I/F ライブラリ【ライブラリ】	○		C:\Program Files\INtime\rt\lib	InTouch.lib
I/F ライブラリ【ヘッダー】	○		C:\Program Files\INtime\rt\include	InTouch.if.h
I/O サーバ	○	○	C:\Program Files\Wonderware\InTouch	InServer.exe
INtime側サンプルプログラム	○		C:\Program Files\InTouchIOSV\Rt	AP_SAMPLE.rta
InTouch 側サンプルプログラム	○		C:\Program Files\InTouchIOSV\InTouch	Sample
INtime側 API テスト AP	○	○	C:\Program Files\InTouchIOSV	IF_TEST.rta

※ライセンス項目内 DK=INtime開発キットライセンス RT=INtime実行ライセンス

ライセンスに関するご質問は、販売店または弊社までお尋ね下さい。

4. システム構成

本製品の構成は、InTouch が動作する Windows 側とリアルタイム処理を行うINtime側の二つに分かれます。



製品名称	動作側	内容
①I/Fドライバ	INtime	INtime Kernel 起動時に実行させます。 I/Fドライバを起動することで、Windows 側で動作する③I/O サーバが起動可能となります。
②I/Fライブラリ	INtime	INtime 側で動作するアプリケーションにリンクします。 これにより、InTouch 側が認識するデータの変更並びに InTouch 側から変更されたデータ、変更イベントをユーザーが作成する INtime アプリケーションで認識することが可能となります。
③I/O サーバ	Windows	InTouch で作成したアプリケーションと外部 I/O をリンクするサーバです。 指定した周期で①I/Fドライバが管理している I/F メモリに対して Read/Write を行います。また、①I/Fドライバが起動されていない場合は、エラーメッセージを表示し、終了します。

5. 動作設定

InTouch 側と INtime 側 (制御側) のインターフェースを行う為に「I/Oサーバ」が存在します。「I/Oサーバ」は、InTouch 側との I/F を可能とする為、トピック名、サンプリング周期の設定を行う必要があります。設定完了後、InTouch 実行画面 (InTouch WindowViewer) の前に起動することにより、INtime 側 (制御側) とのインターフェースが可能となります。

また、「I/Oサーバ」起動前に InTouch 実行画面を起動した場合でも、「I/Oサーバ」の起動確認画面が表示されますので、その時点で起動可能です。

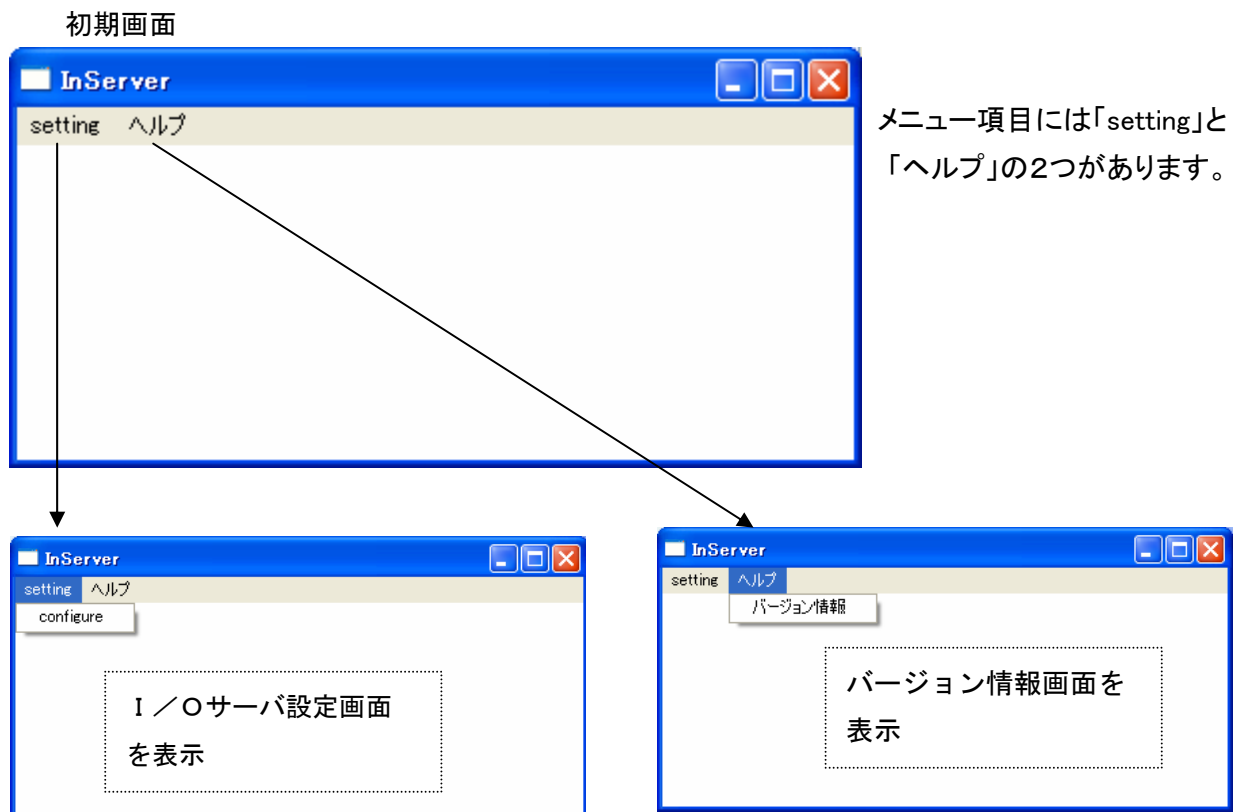
※「I/Oサーバ」起動前には必ず INtime 側 (制御側) の「I/F ドライバ」を起動して下さい。

詳細は、「6. アプリケーション起動方法」を御参照下さい。

5.1 I/O サーバ

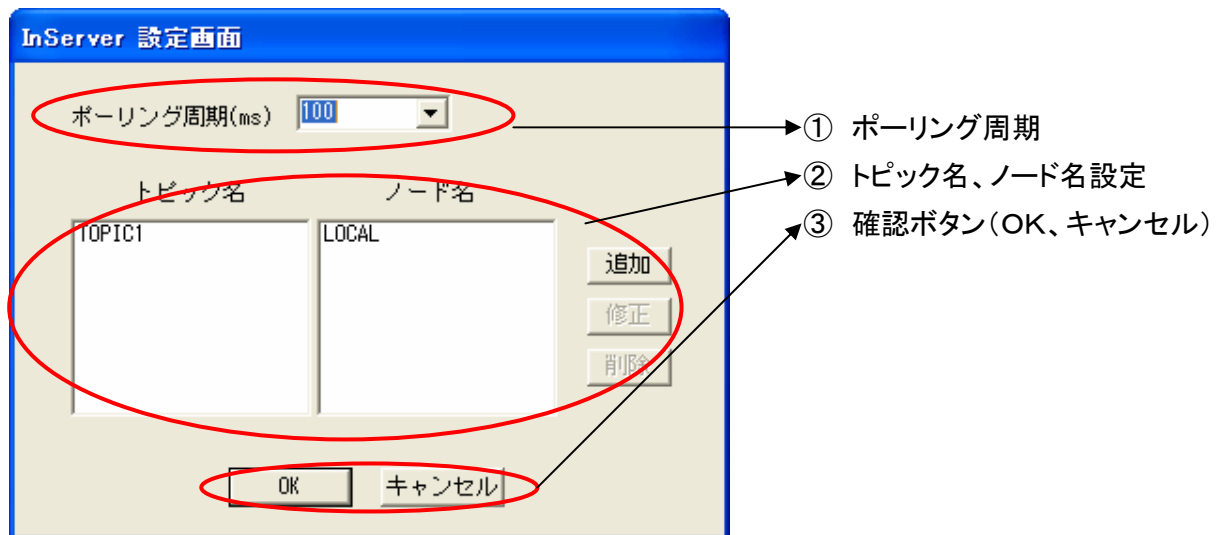
I/Oサーバは、「I/Oサーバ設定画面」にて指定された情報 (ポーリング周期並びにトピック名とノード名) を元に動作します。未指定の場合には、何も動作しません。

I/Oサーバを起動すると、以下の初期画面が表示されます。(実行名 InServer.exe)



5.2 I/O サーバ設定画面

「I/Oサーバ」のメニュー上にある「setting」内の「configure」をクリックすると「I/Oサーバ設定画面」が表示されます。



①ポーリング周期

INtime 側(制御側)に対して、入力情報を取得する時間です。入力単位はミリ秒で行います。周期はコンボボックス又は直接入力に変更可能です。
 ※ポーリング周期の値を小さく(高速)するとCPUの負荷率が高くなります。
 使用するPCの性能等に依存しますが、100m秒を目安にしてください。

②トピック名、ノード名設定

InTouch が外部へのアクセスを行うI/O型(論理、整数、実数、文字列)登録時に設定するアクセス名の中にあるトピックス名を設定します。ノード名は「LOCAL」固定となります。
 ※I/O サーバは、ノード名が重複した指定はできません。
 従って、ノード名「LOCAL」に対応するトピック名は1つしかできません。

③確認ボタン(OK、キャンセル)

OKボタン	設定内容の変更を更新し、再立上げ要求を行います。
キャンセルボタン	設定内容の変更をキャンセルし、通常画面に戻ります。

5.3 I/O サーバ詳細設定画面

INtime 側とインターフェースを行う為には、InTouch 側の「アクセス名の変更」で設定するトピック名と同じ定義を I/O サーバにもしておく必要があります。

InTouch 側は I/O 型タグ変数登録時に本トピック名の登録を行い、このトピック名をキーにして I/O サーバに対してアクセスします。

InTouch側設定例

ノード名 : 未使用
 アクセス名 : 任意の名称
 アプリケーション名 : InServer (固定)
 トピックス名 : 任意の名称
 プロトコル名 : SuiteLink (固定)
 サーバへの通知 : 有効アイテムのみ要求 (固定)

※(固定)と記述されている箇所は
 設定画面例通りに設定して下さい。

InTouch側アクセス名設定画面例

①追加時

I/Oサーバ設定画面から追加ボタンをクリックすると、詳細画面が表示されます。

INtime 側(制御側)への接続先は自PC(ローカル)のみとなります。

トピック名は、InTouch 側で定義されたトピック名を定義します。

トピック名の定義が異なると正常に動作しません。

例として、InTouch 側のトピック名が「TOPIC1」ならば、右のトピック名に定義する内容も「TOPIC1」となります。

②修正時

I/Oサーバ設定画面上で表示しているトピック名又はノード名を選択し、修正ボタンをクリックすると、選択した項目の詳細画面が表示されます。表示した内容で修正が可能となります。

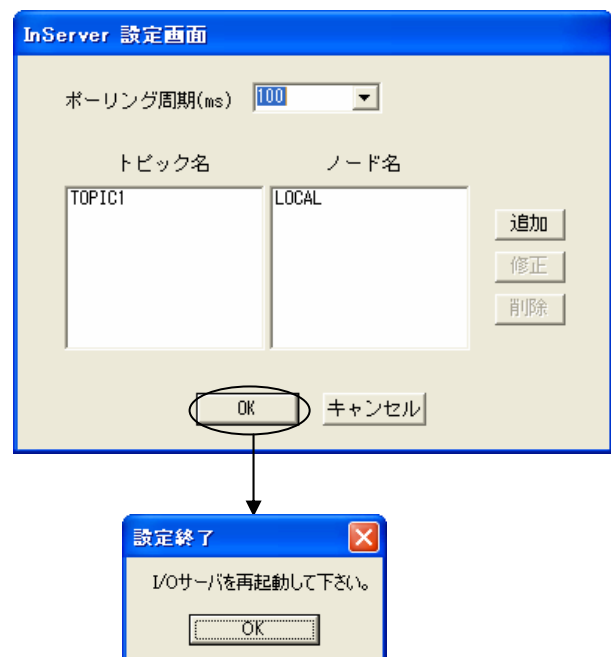
③削除時

I/Oサーバ設定画面上で表示しているトピック名又はノード名を選択し、削除ボタンをクリックすると、選択した項目を削除します。

※注意

- ①ローカル設定時はトピックス名のみの登録となります。
- ②接続先にローカルが登録できるのは、1つのみです。

I/Oサーバ設定画面にて、各種設定又は修正完了後、OKボタンをクリックすると設定内容を保存後、再起動要求のメッセージを表示後、終了します。アプリケーションの再起動を行うことにより設定された内容で動作を開始します。



6. アプリケーションの起動方法

6. 1 I/Fドライバの起動

I/Fドライバは、サービスの一覧中に「InTouch IO Server」の名称で登録されています。

INtime Kernel のサービス起動時に一緒に開始してください。本ドライバを起動させずに INtime 側アプリケーション、I/Oサーバは起動できません。

6. 2 INtime 側ユーザーアプリケーション

通常の INtime アプリケーションの起動と同じです。但し、I/Fドライバが起動された後に実行して下さい。

6. 3 I/Oサーバ及び InTouch アプリケーション (Viewer) の起動

I/Oサーバは、InTouch 実行画面 (Viewer) を立ち上げる前に起動してください。

また、I/Oサーバ起動前に、InTouch 実行画面 (Viewer) を起動した場合でも、「I/Oサーバ」の起動確認画面が表示されますので、その時点で起動可能です。

但し、INtime 側の I/Fドライバ起動後に実行して下さい。

※InTouch 実行画面 (Viewer) の起動前に I/Oサーバを起動しておけば、「I/Oサーバ」の起動確認画面は表示されません。

7. アプリケーション作成手順

7.1 InTouchとINtimeの関係

InTouch 側で定義するアイテム名(タグ名)と INtime 側のユーザーアプリケーションとの関係は以下の表です。

I/Fドライバが、InTouch 側と INtime 側の共通データエリアを管理しています。InTouch 側でアクセスする各型に対応する INtime 側の関数が存在します。使用する型に対応する関数を INtime 側のユーザーアプリケーションで使用するにより InTouch 側とのデータ交換が可能となります。

InTouch側	I/Fドライバ	INtime側	
		対応関数	アドレス
B0~B32767	論理型	InTouch_BitRead InTouch_BitWrite InTouch_ByteRead InTouch_ByteWrite InTouch_WordRead InTouch_WordWrite	0~32767
D0~D32767	整数型	InTouch_IntRead InTouch_IntWrite	0~32767
F0~F32767	実数型	InTouch_FloatRead InTouch_FloatWrite	0~32767
S0~S127	文字列型	InTouch_StrRead InTouch_StrWrite	0~127

※InTouch 側と INtime 側の両方から同じデータに対して Read/Write が可能です。

同じデータに対して InTouch 側、INtime 側ともに書込みを行うと、思わぬ不具合が発生する可能性がありますので、開発時には InTouch 側→INtime 側への書込みエリア、INtime 側→InTouch 側への書込みエリアとわけて使用してください。

7.2 InTouch側

作成自体は通常の InTouch アプリケーションと同一です。異なる点は、外部とのインターフェースを行う I/O型のアイテム名定義が異なる点です。

定義の名称、範囲は上記の InTouch 側の欄を参照下さい。

7.3 INtime側ユーザーアプリケーション

InTouch 側とインターフェースを行う、INtime 側のユーザーアプリケーションを構築するには以下の手順で行います。

- ①Microsoft Visual C++にて、INtime のアプリケーションを作成します。
- ②生成したアプリケーションに、InTouch との I/F を可能とするヘッダーファイル `intc_if.h` とライブラリ `InTouch.lib` を追加します。各々の格納場所は、「3. セットアップ」に記述されているパスです。セットアップ先のパスを変更している場合には、変更先パスにしてください。
- ③アプリケーションの実行開始時に一回だけ InTouch ライブラリの初期化 API を行います。
ライブラリ初期化 API `InTouch_if_init`
- ④以降、InTouch と INtime 側で共有するメモリの読み込み、書き込み、InTouch からの書き込みイベント待ち、InTouch からの書き込みチャンネル付きイベント待ち API を使用してユーザーアプリケーションを構築します。

※サンプルアプリケーションを参照してください。

8. INtime I/Fライブラリ

INtime側のユーザーアプリケーションが利用可能なAPI群です。

①初期化関数

関数名 **WORD InTouch_if_init(void)**
 内容 ライブラリの初期化
 実行時1回のみコールして下さい
 引数 なし
 戻り値 0:正常 1:共有メモリ異常(I/Fスレッド未起動)
 4:セマフォ異常 6:システム異常

②論理型対応関数

関数名 **WORD InTouch_BitRead(long l_Add, BYTE *bDt)**
 内容 指定論理型アドレス位置のデータを取得
 引数 long l_Add 論理型データ格納位置
 BYTE *bDt 読み込みデータ格納変数(Oor1)
 戻り値 0:正常 1:共有メモリ異常(I/Fスレッド未起動)
 2:指定アドレス異常

関数名 **WORD InTouch_BitWrite(long l_Add, BYTE bDt)**
 内容 指定論理型アドレス位置にデータ書込み
 引数 long l_Add 論理型データ格納位置
 BYTE bDt 書込みデータ(Oor1)
 戻り値 0:正常 1:共有メモリ異常(I/Fスレッド未起動)
 2:指定アドレス異常 3:指定データ異常
 4:セマフォ異常 5:タイムアウト

関数名 **WORD InTouch_ByteRead(long l_Add, BYTE *bDt)**
 内容 指定論理型アドレス位置から8つのデータをバイトイメージで取得
 引数 long l_Add 論理型先頭データ格納位置
 BYTE *bDt 読み込みデータ格納変数(8つのデータがビット毎にOor1を格納)
 戻り値 0:正常 1:共有メモリ異常(I/Fスレッド未起動)
 2:指定アドレス異常

関数名 **WORD InTouch_ByteWrite(long l_Add, BYTE bDt)**
 内容 指定論理型アドレス位置から8つのデータをバイトイメージで書込み
 引数 long l_Add 論理型先頭データ格納位置
 BYTE bDt 書込みデータ格納変数(8つのデータがビット毎に0or1を格納)
 戻り値 0:正常 1:共有メモリ異常(I/Fスレッド未起動)
 2:指定アドレス異常 4:セマフォ異常
 5:タイムアウト

関数名 **WORD InTouch_WordRead(long l_Add, WORD *wDt)**
 内容 指定論理型アドレス位置から16のデータをワードイメージで取得
 引数 long l_Add 論理型先頭データ格納位置
 WORD *wDt 読み込みデータ格納変数(16のデータがビット毎に0or1を格納)
 戻り値 0:正常 1:共有メモリ異常(I/Fスレッド未起動)
 2:指定アドレス異常

関数名 **WORD InTouch_WordWrite(long l_Add, WORD wDt)**
 内容 指定論理型アドレス位置から16のデータをワードイメージで書込み
 引数 long l_Add 論理型先頭データ格納位置
 WORD wDt 書込みデータ格納変数(16のデータがビット毎に0or1を格納)
 戻り値 0:正常 1:共有メモリ異常(I/Fスレッド未起動)
 2:指定アドレス異常 4:セマフォ異常
 5:タイムアウト

③整数型対応関数

関数名 **WORD InTouch_IntRead(long l_Add, int *iDt)**
 内容 指定整数型アドレス位置のデータを取得
 引数 long l_Add 整数型データ格納位置
 Int *iDt 読み込みデータ格納変数
 戻り値 0:正常 1:共有メモリ異常(I/Fスレッド未起動)
 2:指定アドレス異常

関数名 **WORD InTouch_IntWrite(long l_Add, int iDt)**
 内容 指定整数型アドレス位置にデータ書込み
 引数 long l_Add 整数型データ格納位置
 Int iDt 書込みデータ
 戻り値 0:正常 1:共有メモリ異常(I/Fスレッド未起動)
 2:指定アドレス異常 4:セマフォ異常
 5:タイムアウト

④実数型対応関数

関数名 **WORD InTouch_FloatRead(long l_Add, float *fDt)**

内容 指定実数型アドレス位置のデータを取得

引数 long l_Add 実数型データ格納位置

Float *fDt 読み込みデータ格納変数

戻り値 0:正常 1:共有メモリ異常(I/Fスレッド未起動)

2:指定アドレス異常

関数名 **WORD InTouch_FloatWrite(long l_Add, float fDt)**

内容 指定実数型アドレス位置にデータ書込み

引数 long l_Add 実数型データ格納位置

Float fDt 書込みデータ

戻り値 0:正常 1:共有メモリ異常(I/Fスレッド未起動)

2:指定アドレス異常 4:セマフォ異常

5:タイムアウト

⑤文字列型対応関数

関数名 **WORD InTouch_StrRead(long l_Add, char *cDt)**

内容 指定文字列型アドレス位置の文字列データを取得

引数 long l_Add 文字列型データ格納位置

Char *cDt 読み込みデータ格納変数先頭アドレス

戻り値 0:正常 1:共有メモリ異常(I/Fスレッド未起動)

2:指定アドレス異常

関数名 **WORD InTouch_StrWrite(long l_Add, char *cDt)**

内容 指定文字列型アドレス位置に文字列データ書込み

引数 long l_Add 文字列型データ格納位置

Char *cDt 書込みデータ先頭アドレス

戻り値 0:正常 1:共有メモリ異常(I/Fスレッド未起動)

2:指定アドレス異常 4:セマフォ異常

5:タイムアウト

9. その他

9.1 InTouchタグ

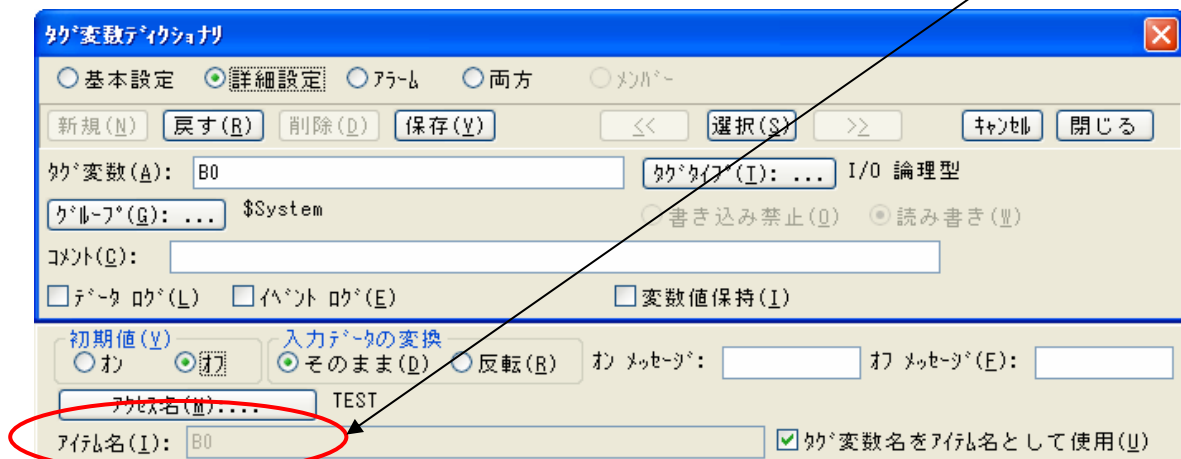
InTouchから外部へのインターフェースを行う際に定義するI/O型の範囲は以下の表です。

(アイテム名に登録します)

InTouch I/O 型	リンク名称	範囲
I/O 論理型	B	B0~B32767
I/O 整数型	D	D0~D32767
I/O 実数型	F	F0~F32767
I/O 文字列型	S	S0~S127

表の範囲に示されている範囲の値を InTouch 側のアイテム名に定義します。範囲外を指定した場合は、無効扱いとなります。

InTouchのタグ変数の定義例



9.2 I/Oサーバについて

I/O サーバは、1台の PC で InTouch 及び INtime の動作を行います。

I/O サーバのポーリング周期を高速にすることは可能ですが、InTouch 側の動作に負荷がかかりますので調整が必要となります。(デフォルト100msec)

9.3 INtime 側アプリケーションについて

INtime 側で実行するアプリケーションで本ライブラリをリンクしたプロセスは1プロセスとなります。複数のプロセスで起動した場合、不定な動作となります。

9.4 INtime側 API テスト AP について

INtime 側で提供しているライブラリをテストするアプリケーションとなります。(IF_TEST.rta)

INtime 側から InTouch のアプリケーションに対して I/O サーバ経由で動作確認する際に利用できます。

INtime 側で実行し、InTouch側が参照、更新するI/O型のデータに対してアクセスできます。

10. サンプルプログラム

本サンプルプログラムは、INtime 側のサンプルプログラム、InTouch のサンプルプログラムの2種用意されています。

10.1 起動方法

サンプルプログラムを起動する前には、INtime 側の「I/F ドライバ」の起動を確認してください。

未起動であれば、起動してください。その後、Windows 側の「I/O サーバ」を起動して下さい。

「I/O サーバ」の設定がされていなければ、トピック名を【TOPIC1】、ポーリング周期を【100】msec とし再起動してください。

「I/F ドライバ」及び「I/O サーバ」が起動しましたら、INtime 側のサンプルプログラム「AP_SAMPLE.rta」、続いて InTouch のサンプルプログラムの順に起動してください。

10.2 サンプル画面

InTouch 実行画面(InTouch WindowViewer)にて InTouch のサンプルプログラムを実行すると、以下の画面が表示されます。

画面は、大きく分けて3エリアあり、「①論理、整数、実数型入出力エリア」、「②文字列型エリア」、「③アナログ波形エリア」となります。

①論理、整数、実数型入出力エリア

論理型、整数型、実数型の現在値を表示します。値の変更も可能です。本エリアの上端に表示されている値は、ベースアドレス(0、1000、5000、10000、30000)となります。本エリアの左端に表示されている値はオフセットアドレス(0~9)となり、この2つのアドレスを加算した値がアドレスとなります。

例:ベースアドレス0、オフセットアドレス5の論理部は I/O 論理型の5(B5)を示します。

ベースアドレス 30000、オフセットアドレス9の実数部は I/O 実数型の 30009(F30009)を示します。

INtime 側から情報が変更されると、画面上の表示も更新されます。

また、画面上から変更したい情報をクリックすると入力エリアが表示されます。入力値を変更することも可能です。変更時には、INtime 側サンプルプログラムに変更イベント通知が発行されます。

②文字列型エリア

文字列型の現在値を表示します。値の変更も可能です。本エリアの左端に表示されている値がアドレス(0、1、5、10、20、40、60、80、100、127)となります。

INtime 側から情報が変更されると、画面上の表示も更新されます。

また、画面上から変更したい文字列情報をクリックすると入力エリアが表示されます。入力文字を変更することも可能です。変更時には、INtime 側サンプルプログラムに変更イベント通知が発行されます。

③アナログ波形エリア

ラジオボタンで選択可能な「正弦波」、「パルス波」、「ラジオ波」を選択することで、INtime 側のサンプルプログラムに通知され、指定された波形データを作り、その波形データを下に画面右側のグラフに波形を表示します。

ラジオボタンの操作を行うと I/O 整数型の 32766 (D32766) に対して値を書き込みます。

「正弦波」=1、「パルス波」=2、「ラジオ波」=3

INtime 側サンプルプログラムは、値の変更イベント通知を受信し、要求された波形を認識、波形データを作成し、定周期に波形データを I/O 整数型の 32767 (D32767) に書き込みます。

画面右側のグラフには、I/O 整数型の 32767 (D32767) に格納された値を取り込み、グラフ表示を行います。また、ラジオボタンの下に表示されている値は、I/O 整数型の 32767 (D32767) の内容です。

サンプル画面

	0		1000		5000		10000		30000	
	論理 整数	実数	論理 整数	実数	論理 整数	実数	論理 整数	実数	論理 整数	実数
0	Off 0	0.00	Off 0	0.00	Off 0	0.00	Off 0	0.00	Off 0	0.00
1	Off 0	0.00	Off 0	0.00	Off 0	0.00	Off 0	0.00	Off 0	0.00
2	Off 0	0.00	Off 0	0.00	Off 0	0.00	Off 0	0.00	Off 0	0.00
3	Off 0	0.00	Off 0	0.00	Off 0	0.00	Off 0	0.00	Off 0	0.00
4	Off 0	0.00	Off 0	0.00	Off 0	0.00	Off 0	0.00	Off 0	0.00
5	Off 0	0.00	Off 0	0.00	Off 0	0.00	Off 0	0.00	Off 0	0.00
6	Off 0	0.00	Off 0	0.00	Off 0	0.00	Off 0	0.00	Off 0	0.00
7	Off 0	0.00	Off 0	0.00	Off 0	0.00	Off 0	0.00	Off 0	0.00
8	Off 0	0.00	Off 0	0.00	Off 0	0.00	Off 0	0.00	Off 0	0.00
9	Off 0	0.00	Off 0	0.00	Off 0	0.00	Off 0	0.00	Off 0	0.00

文字列

0	
1	
5	
10	
20	
40	
60	
80	
100	
127	

波形タイプ

正弦波

パルス波

ランプ波

正弦波

2968

①論理、整数、実数型入出力エリア

②文字列型入出力エリア

③アナログ波形エリア

※INtime側の開発環境パスをデフォルト以外で使用されている場合には、そちらのパスに合わせる必要があります。

INtime側サンプルプログラム実行画面(AP_SAMPLE.rta)

InTouch側から操作されたデータに対応したイベントが表示されます。

```

CA RT I/O Console
Write Event Recive
Discrete Type : Ch = 0
Write Event Recive
Integer Type : Ch = 1
Write Event Recive
Real Type : Ch = 2
Write Event Recive
String Type : Ch = 0
    
```

- ①Write Event Recive
- ②Discrete Type : Ch = ?
- ③Integer Type : Ch = ?
- ④Real Type : Ch = ?
- ⑤String Type : Ch = ?

①Write Event Recive

→関数【InTouch_WriteEvt】でデータ更新イベントを待っているスレッドが表示
InTouch側から何らかのデータ更新があった。

②Discrete Type : Ch = ?

→関数【InTouch_ChWriteEvt】で論理データ更新イベントを待っているスレッドが表示
InTouch側から論理データのデータ更新があった。(Ch = ? : 変更のあったNoを通知)

③Integer Type : Ch = ?

→関数【InTouch_ChWriteEvt】で整数データ更新イベントを待っているスレッドが表示
InTouch側から整数データのデータ更新があった。(Ch = ? : 変更のあったNoを通知)

④Real Type : Ch = ?

→関数【InTouch_ChWriteEvt】で実数データ更新イベントを待っているスレッドが表示
InTouch側から実数データのデータ更新があった。(Ch = ? : 変更のあったNoを通知)

⑤String Type : Ch = ?

→関数【InTouch_ChWriteEvt】で文字列データ更新イベントを待っているスレッドが表示
InTouch側から文字列データのデータ更新があった。(Ch = ? : 変更のあったNoを通知)

1 1. 付録

1 1. 1 Visual Studio.NET2003 以上でのサンプルプログラムの実行方法

下記は Visual Studio.NET2003 以上の環境でサンプルプログラムを実行する際の手順です。

1. Visual Studio.NET2003 を起動します。
2. 「ファイル」→「新規作成」→「プロジェクト」と進み、「プロジェクトの種類」は「INtime Projects」、「テンプレート」は「Application Wizard」を選択します。そしてファイル名を入力し「OK」をクリックします。



3. 「生成するアプリケーションの種類」は「空のプロジェクト(E)」を選択し、「OK」をクリックします。



4. 空のプロジェクトの作成後ソースコードを追加します。「ファイル」→「既存項目の追加」と進み、<「3. セットアップ」に記述されているパス>¥AP_SAMPLE.c を選択し、「開く」をクリックします。
同じく Poll1.c も選択し、「開く」をクリックします。
5. そして InTouch.lib を追加します。「ファイル」→「既存項目の追加」と進み、<INtime インストールパス>¥rt¥lib¥InTouch.lib を選択し、「開く」をクリックします。
6. これで Visual Studio.NET2003 の環境でサンプルプログラムをコンパイルおよび実行することが可能です。