

IOB30RTA ActiveX Control

Visual Basic による KCXH-IOB30RTA 入門

使用説明書(V1.00)

株式会社 エスアイ創房

IOB30RTA

改定履歴

第 0.98 版	2001/11/21	
第 0.99 版	2002/01/22	(アナログ7、8CH 入力部バグ FIX)
第 0.99a 版	2002/02/28	(BI_Value,BO_Value VC 使用時のバグ FIX)
第 1.00 版	2007/09/01	(VC 版 FIX)

Microsoft, Windows, Visual Basic, ActiveX は米国 Microsoft Corporation の登録商標です。
KARACRIX は株式会社エスアイ創房の登録商標です。
その他、本文中に記載されている社名および商品名は、一般に開発メーカーの登録商標です。

IOB30RTA 使用説明書 第 1.00 版 © S.I.Soubou Inc.

目次

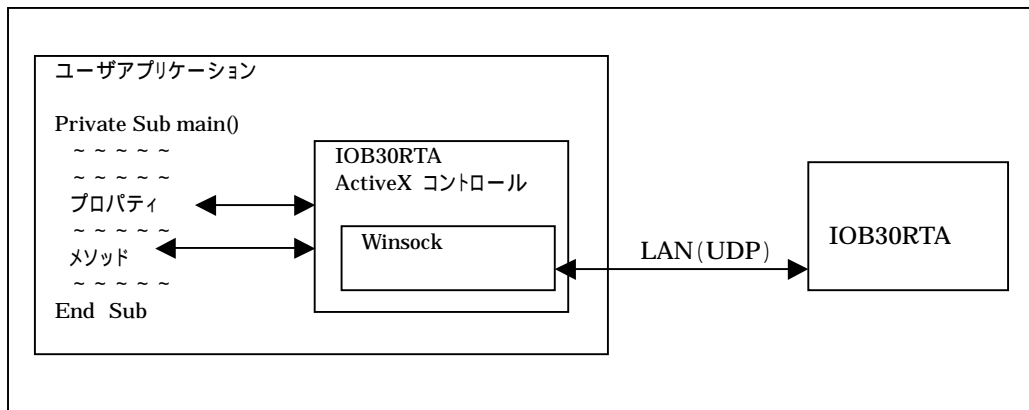
1. 概要.....	4
2. インストール	5
2.1 ダウンロードとセットアップ.....	5
2.2 プロジェクトへ IOB30RTA ActiveX Control の追加.....	5
3. IOB30RTA ActiveX Control プログラミング	6
3.1 IOB30RTA の装置クロック情報を取得してみる	6
3.2 IOB30RTA のデジタル&アナログ値を定期的に計測表示してみる.....	8
3.3 IOB30RTA のリレーを操作してみる.....	11
4. IOB30RTA ActiveX Control リファレンス	14
4.1 プロパティ	14
4.1.1 プロパティ一覧.....	14
4.1.2 AI_Value プロパティ.....	14
4.1.3 BI_Value プロパティ.....	15
4.1.4 BO_Value プロパティ	15
4.1.5 FirmVer プロパティ.....	15
4.1.6 IP プロパティ.....	16
4.1.7 KernelTime プロパティ	16
4.1.8 LocalPort プロパティ.....	17
4.1.9 MacAddress プロパティ.....	17
4.1.10 Port プロパティ	17
4.1.11 Timeout プロパティ.....	18
4.1.12 MachineName プロパティ.....	18
4.1.13 ErrWindowStop プロパティ	19
4.2 メソッド.....	21
4.2.1 メソッド一覧.....	21
4.2.2 GetAin メソッド.....	21
4.2.3 GetBin メソッド.....	22
4.2.4 Calibrate メソッド	23
4.2.5 GetMachineInfo メソッド	24
4.2.6 SetBout メソッド	25
4.2.7 SetRange メソッド.....	26
4.3 イベント.....	27
4.4 エラーコード一覧.....	27

IOB30RTA

付録. サンプルプログラムの使用法.....	28
1. 環境設定ダイアログの設定	28
2. メイン情報表示画面の使い方.....	29

1. 概要

IOB30RTA 用の ActiveX コントロールを使う事により、VisualBasic から IOB30RTA を容易に扱えるようになります。また、ユーザアプリケーションからは複雑なプログラミングを行うことなくプロパティ、メソッドを通し、直感的に IOB30RTA の操作が行なえるようになります。



IOB30RTA ActiveX コントロール の概要図

< IOB30RTA 2台と PC 1台の LAN 接続例 >



IOB30RTA

2. インストール

IOB30RTA用のActiveXコントロールをダウンロードし、インストールする手順を説明します。

2.1 ダウンロードとセットアップ

1. 古いバージョンのActiveX-OCXを既にご使用の場合には、コントロールパネルの「アプリケーションの追加と削除」でこれに相当する“IOB30RTA ActiveX Control”を予め削除しておいてください。
2. ActiveX-OCXファイル(KCX_OCX_IOB30RTA_v???.lzh)を弊社サイトからダウンロードします。(???はバージョン番号を示します)
3. ActiveX-OCXファイルを解凍します。解凍されたフォルダの中には、KCX_OCX_IOB30RTA_v???.EXE(OCXインストーラ)が含まれます。
4. 解凍後、KCX_OCX_IOB30RTA_v???.EXE を実行します。後は画面の指示に従ってください。
OCXは、ウィンドウズのシステムディレクトリにインストールされます。

2.2 プロジェクトへ IOB30RTA ActiveX Control の追加

1. [VB/メニュー]->[プロジェクト(P)]->[コンポーネント(O)]で、コンポーネント・ダイアログを開きます。
コントロール・タグ上で、“IOB30RTA ActiveXコントロール モジュール”のチェックを選択(チェック)しOKボタンを押して確定します。
ツールボックス上に歯車のアイコン(IOB30RTA)が追加されるのを確認します。

3. IOB30RTA ActiveX Control プログラミング

VisualBasic から、IO30RTA 用コントロールを使ったいくつかのプログラム例を紹介します。なお下記説明の IOB30RTA 装置に関する設定は、工場出荷時の設定値を使用しています。

IOB30RTA 工場出荷時の設定は、以下の通りです。

IP アドレス = 192.168.0.200

コントロールポート = 20000

3.1 IOB30RTA の装置クロック情報を取得してみる

IOB30RTA には、装置内の時間を示すカーネルタイマ・カウンタが組み込まれており、装置の時間を簡単に外部から取得することが出来ます。その時間を読み取ってみましょう。

1. フォームを1つ作成します。

2. IOB30RTA の装置時間を表示する為の、TextBox コントロールを1つ配置します。
オブジェクト名は、"Text1"としました。

3. ツールボックスの IOB30RTA 用コントロール(歯車アイコン)を選択し、IOB30RTA コントロールを1つ配置します。

オブジェクト名を、"IOB30RTA1"としました。

プロパティウィンドウ内の IP と Port プロパティが IOB30RTA のデフォルト状態と同じであることを確認します。

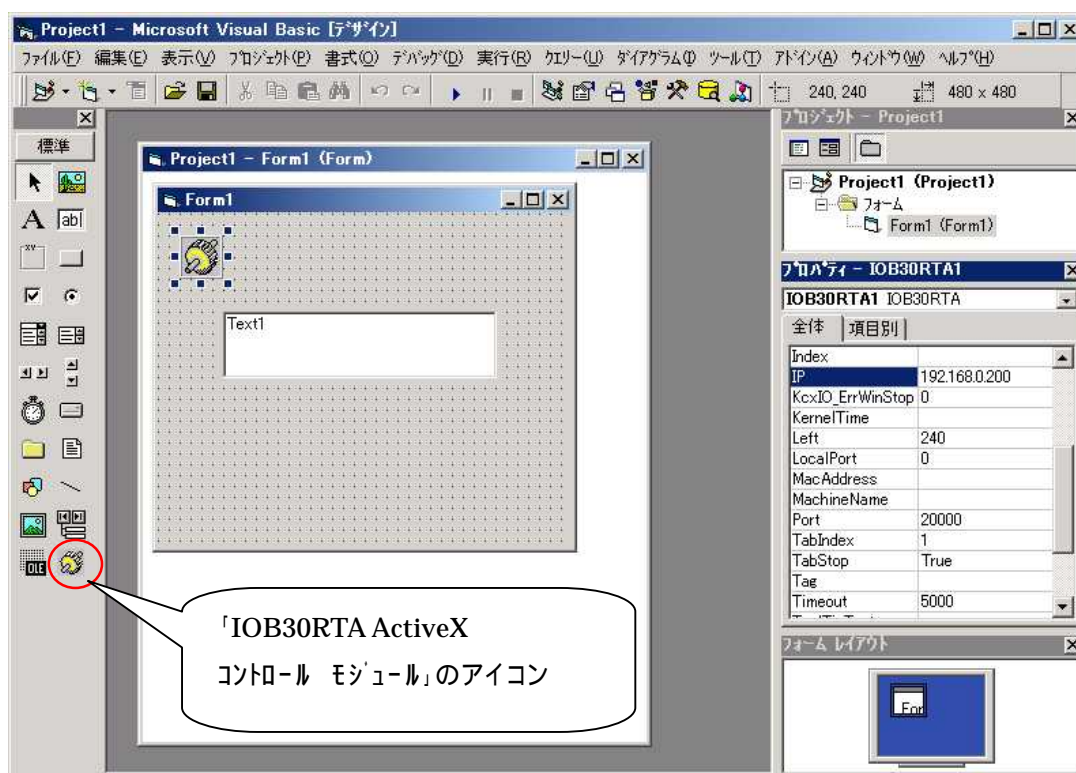
IP = 192.168.0.200

Prot = 20000

IOB30RTA との通信エラータイムアウトは構内 LAN で使用するため、5秒ということにしておきます。

Timeout = 5000

IOB30RTA



4. プロシージャの記述

プログラムは、以下の通りです。

プログラム実行時、つまり、Form_Load() 実行時に、IOB30RTA の GetMachineInfo メソッドを実行し、IOB30RTA の装置情報を取得します。通信エラーが発生しなければ、KernelTime プロパティにカーネルタイマカウンタ値が格納されていますので、これをテキストボックスに表示してみます。

```
Private Sub Form_Load()  
    IOB30RTA1.GetMachineInfo  
    Text1.Text = IOB30RTA1.KernelTime  
End Sub
```

5. 実行

プログラムを何度か実行してみてください。

テキストボックスに表示される装置時間(カーネルタイマカウンタ値)が、1/76.29 秒毎に1ずつ増えていけば、装置及びプログラムは正常に動作しています。

3.2 IOB30RTA のデジタル&アナログ値を定期的に計測表示してみる

IOB30RTA には、4つの接点と8つのアナログの状態を入力することができます。その状態を「定期的」に取り取ってみましょう。

ここでは、説明簡略化のため、接点1チャンネルとレンジの異なるアナログ2チャンネル分を対象としました。

アナログ計測レンジの設定は、1ch 目にユニポーラ(0 ~ +2.5V)計測を、2ch 目にバイポーラ(-2.5V ~ +2.5V)計測を指定し、計測レートは共にデフォルトの 15Hz(約 14Bit 精度)としました。

1. フォームを1つ作成します。
2. 接点1つ、アナログ2つ分の状態を表示する為の、TextBox コントロールを都合3つ配置します。
オブジェクト名は、それぞれ"Text1", "Text2", "Text3"としました。

3. IOB30RTA を定期的に計測させるには、インターバル周期を発生させる必要があります。これにタイマーコントロールを使い、オブジェクトを1つ配置します。オブジェクト名は、"Timer1"としました。この時、プロパティウィンドウ内の Interval プロパティには、IOB30RTA をインターバル計測させたい間隔時間をミリ秒単位で入れます。また、タイマーの Enabled プロパティの初期値には、False を選択しておきましょう。
(False にしておく理由: プログラム内でタイマーを有効にしたいタイミングで起動するため)

4. ツールボックスの IOB30RTA 用コントロールを選択し、IOB30RTA コントロールを1つ配置します。オブジェクト名を、"IOB30RTA1"としました。

プロパティウィンドウ内の IP と Port プロパティが IOB30RTA のデフォルト状態と同じであることを確認します。

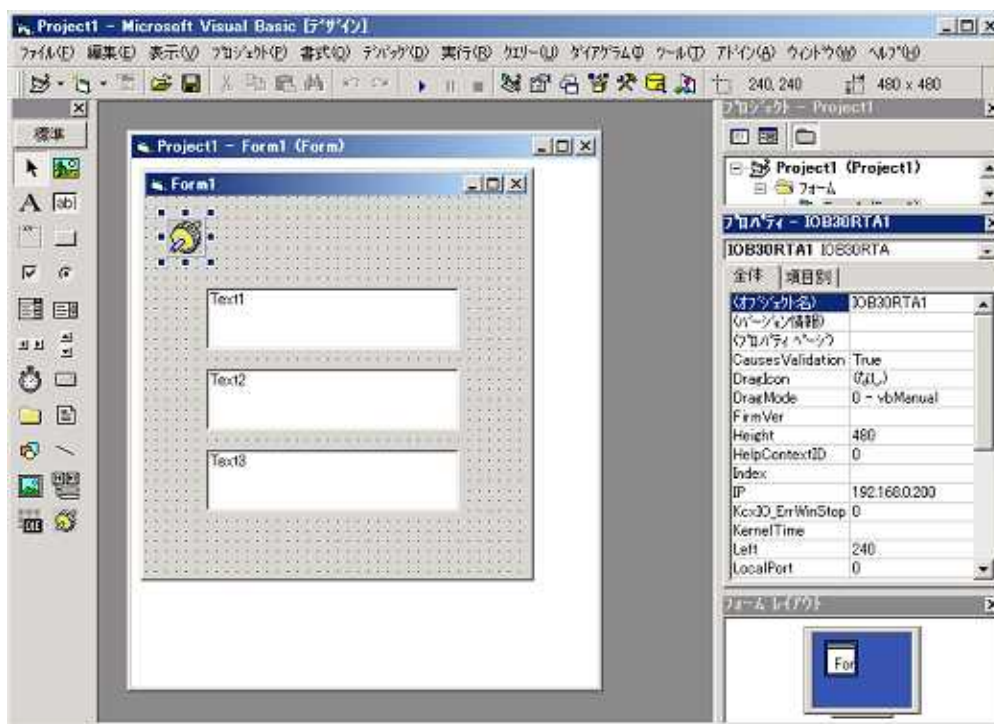
IP = 192.168.0.200

Prot = 20000

IOB30RTAとの通信エラータイムアウトは構内 LAN で使用するため、5秒ということにしておきます。

Timeout = 5000

IOB30RTA



5. プロシージャの記述

プログラムは、以下の通りです。

プログラムでは、実行時(Form_Load())に、IOB30RTA の SetRange メソッドを実行して、アナログレンジの設定を行っています。この設定は IOB30RTA がリセットされるか、あるいは再度 SetRange メソッドを実行するまで、レンジは変化することはありません。

なお、定期計測までの準備期間中は、タイマーの実行を停止させておきます。

アナログレンジ設定の詳細に関しては、後述のリファレンスを参照してください。

例えば、バイポーラ(-2.5V ~ +2.5V)で、有効精度約 12Bit の計測を行いたい場合は
rt = AI_RANGE_BIPOLAR + AI_RATE_616HZ と記述します。

```
Private Sub Form_Load()
```

```
Dim rt1, rt2, rt3, rt4, rt5, rt6, rt7, rt8
```

```
Timer1.Enabled = False ' 準備終了までタイマーは無効にしておく
```

```
rt1 = AI_RANGE_UNIPOLAR ' 1ch=ユニポーラ(0 ~ +2.5V)計測
```

```
rt2 = AI_RANGE_BIPOLAR ' 2ch=バイポーラ(-2.5V ~ +2.5V)計測
```

```
rt3 = AI_NO_MEASURE ' 3ch=非計測
```

```
rt4 = AI_NO_MEASURE ' 4ch=非計測
```

```
rt5 = AI_NO_MEASURE ' 5ch=非計測
```

```
rt6 = AI_NO_MEASURE      ' 6ch=非計測
rt7 = AI_NO_MEASURE      ' 7ch=非計測
rt8 = AI_NO_MEASURE      ' 8ch=非計測
IOB30RTA1.SetRange rt1, rt2, rt3, rt4, rt5, rt6, rt7, rt8
Timer1.Enabled = True    ' 準備終了の為タイマーを有効にする
End Sub
```

アナログレンジの設定が正常に終了すれば、タイマーイベントが発生し、Timer1_Timer()が実行されます。Timer1_Timer() には、On Error GoTo 文が挿入されています。これは、IOB30RTA との通信エラーが発生した場合、実行時エラー・ダイアログの表示を抑止させ、引き続きインターバル計測させるためのものです。IOB30RTA の GetBin メソッドを実行すると、接点状態が、GetAin メソッドを実行すると、アナログ値が、それぞれ、BI_Value と Ai_Value のプロパティに格納されます。これをテキストボックスに表示します。

```
Private Sub Timer1_Timer()
    On Error GoTo ErrNetCom ' 通信エラー発生時計測を停止させない為の処置
    IOB30RTA1.GetBin        ' 接点計測(通信)
    Text1.Text = IOB30RTA1.BI_Value(0) ' 1ch 目の接点状態表示
    IOB30RTA1.GetAin       ' アナログ計測(通信)
    Text2.Text = IOB30RTA1.Ai_Value(0) ' 1ch 目のアナログ値表示
    Text3.Text = IOB30RTA1.Ai_Value(1) ' 2ch 目のアナログ値表示
ErrNetCom:
End Sub
```

6. 実行

プログラムを実行してみてください。

テキストボックスに IOB30RTA の状態が表示されるはずです。

IOB30RTA

3.3 IOB30RTA のリレーを操作してみる

IOB30RTA には、4つの(フォトモス)リレーがありこれを操作することができます。

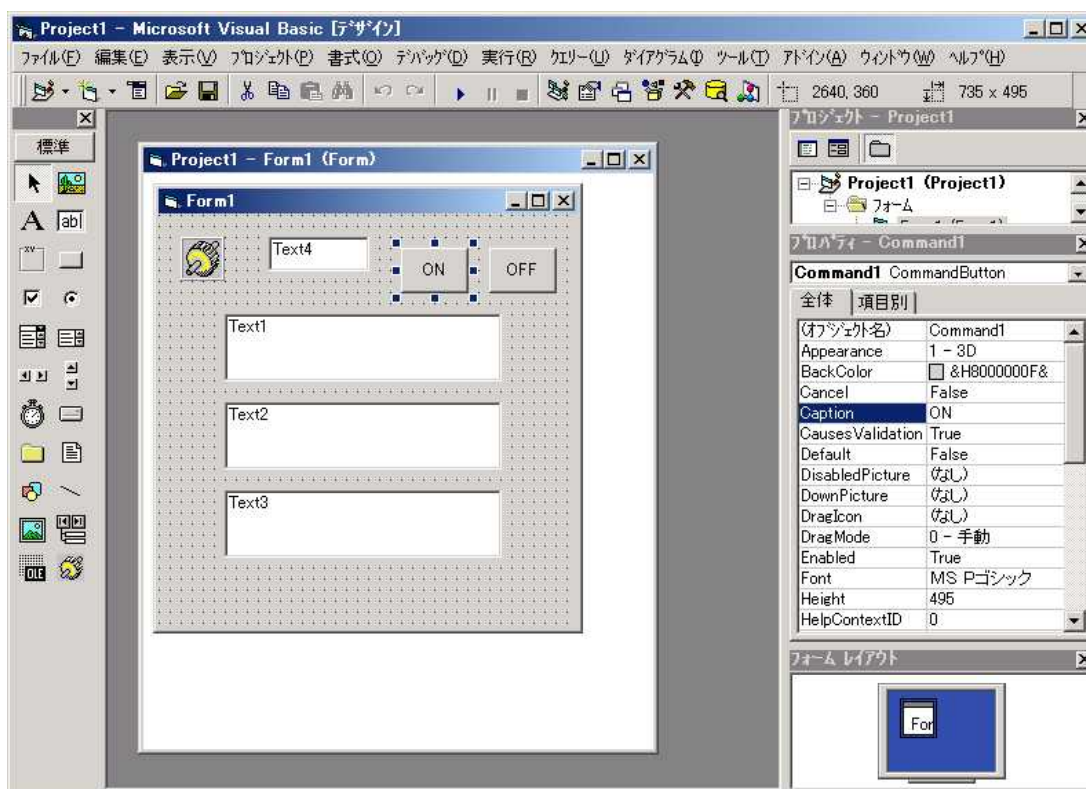
上記、「3-2 IOB30RTA のデジタル&アナログ値を定期的に計測表示してみる」で作ったものに、リレーを操作するボタンを追加してみます。

1. IOB30RTA のリレー状態を表示させる為の、TextBox コントロールを1つ配置します。

オブジェクト名を、"Text4"としました。

また、リレーを ON/OFF 操作させる為の、Button コントロールを 2つ配置します。

このオブジェクト名は、それぞれ、ON-> "Command1", OFF-> "Command2" としました。



2. プロシージャの記述

プログラムは、以下の通りです。

```
Private Sub Form_Load()
    Dim rt1, rt2, rt3, rt4, rt5, rt6, rt7, rt8
    Timer1.Enabled = False      ' 準備終了までタイマーは無効にしておく
    rt1 = AI_RANGE_UNIPOLAR     ' 1ch=ユニポーラ(0 ~ +2.5V)計測
    rt2 = AI_RANGE_BIIPOLAR     ' 2ch=バイポーラ(-2.5V ~ +2.5V)計測
    rt3 = AI_NO_MEASURE         ' 3ch=非計測
    rt4 = AI_NO_MEASURE         ' 4ch=非計測
    rt5 = AI_NO_MEASURE         ' 5ch=非計測
    rt6 = AI_NO_MEASURE         ' 6ch=非計測
    rt7 = AI_NO_MEASURE         ' 7ch=非計測
    rt8 = AI_NO_MEASURE         ' 8ch=非計測
    IOB30RTA1.SetRange rt1, rt2, rt3, rt4, rt5, rt6, rt7, rt8
    Timer1.Enabled = True      ' 準備終了の為タイマーを有効にする
End Sub
```

IOB30RTA の GetBin メソッドを実行すると、接点状態の他に現在のリレー状態も同時に取得できる仕様になっています。リレーの状態は、BO_Value プロパティに格納されていますので、これをテキストボックスに表示します。

```
Private Sub Timer1_Timer()
    On Error GoTo ErrNetCom     ' 通信エラー発生時計測を停止させない為の処置
    IOB30RTA1.GetBin            ' 接点計測(通信)
    Text1.Text = IOB30RTA1.BI_Value(0) ' 1ch 目の接点状態表示
    Text4.Text = IOB30RTA1.BO_Value(0) ' 1ch 目のリレー状態表示
    IOB30RTA1.GetAin           ' アナログ計測(通信)
    Text2.Text = IOB30RTA1.Ai_Value(0) ' 1ch 目のアナログ値表示
    Text3.Text = IOB30RTA1.Ai_Value(1) ' 2ch 目のアナログ値表示
ErrNetCom:
End Sub
```

IOB30RTA

フォームに配置した、ONボタンを押すと、Command1_Click() が実行され、IOB30RTA オブジェクトの SetBout メソッドを実行し 1ch 目の "リレーON メッセージ" が送信されます。Command2_Click() の場合は同様に 1ch 目の "リレーOFF メッセージ" が送信されます。SetBout メソッド の引数に BO_ON を指定すると、リレーはオン(接点短絡)、BO_OFF を指定すると、オフ(接点開放)を要求することになります。操作要求しない、つまり現状のままよいという場合には、BO_NOCHANGE を指定します。

```
Private Sub Command1_Click()  
    ' 1ch 目のリレーON を送信  
    IOB30RTA1.SetBout BO_ON, BO_NOCHANGE, BO_NOCHANGE, BO_NOCHANGE  
End Sub
```

```
Private Sub Command2_Click()  
    ' 1ch 目のリレーOFF を送信  
    IOB30RTA1.SetBout BO_ON, BO_NOCHANGE, BO_NOCHANGE, BO_NOCHANGE  
End Sub
```

3. 実行

プログラムを実行してみてください。
テキストボックスに IOB30RTA の諸状態を表示しながら、リレーの ON/OFF 操作が出来るはずで

4. IOB30RTA ActiveX Control リファレンス

4.1 プロパティ

4.1.1 プロパティ一覧

項番	名称	データ型	属性	説明	初期値
1	AI_Value	Long	R	アナログ入力(AI)各チャンネルの状態	
2	BI_Value	Integer	R	接点入力(BI)各チャンネルの状態	
3	BO_Value	Integer	R	リレー出力(BO)各チャンネルの状態	
4	FirmVer	String	R	IOB30RTA のファームウェアバージョン	
5	IP	String	RW	IOB30RTA の IP アドレス(ホスト名)	192.168.0.200
6	KernelTime	Long	R	IOB30RTA のカーネルタイムカウンタ	
7	LocalPort	Long	RW	ローカルコンピュータのポート番号	0
8	MacAddress	String	R	IOB30RTA の MAC アドレス	
9	Port	Long	RW	IOB30RTA のポート番号	20000
10	Timeout	Long	RW	各メソッド実行時のタイムアウト時間(ms)	5000
11	MachineName	String	RW	機械名称	
12	ErrWindowStop	Long	RW	デバッグダイアログ表示フラグ	

4.1.2 AI_Value プロパティ

アナログ入力各チャンネルの状態を返します。

構文

object.AI_Value(*ch*)

データ型

長整数型(Long)

構文の指定項目は次のようになります。

指定項目	データ型	内容
<i>object</i>		オブジェクトを指すオブジェクト式です。
<i>ch</i>	Integer	チャンネル番号(0~7)

IOB30RTA

4.1.3 BI_Value プロパティ

接点入力 (BI) 各チャンネルの状態を返します。

構文

object.BI_Value(ch)

データ型

バイト型 (Byte)

構文の指定項目は次のようになります。

指定項目	データ型	内 容
<i>object</i>		オブジェクトを指すオブジェクト式です。
ch	Integer	チャンネル番号 (0 ~ 3)

4.1.4 BO_Value プロパティ

リレー出力 (BO) 各チャンネルの状態を返します。

構文

object.BO_Value(ch)

データ型

バイト型 (Byte)

構文の指定項目は次のようになります。

指定項目	データ型	内 容
<i>object</i>		オブジェクトを指すオブジェクト式です。
ch	Integer	チャンネル番号 (0 ~ 3)

4.1.5 FirmVer プロパティ

IOB30RTA のファームウェアバージョンを返します。

構文

object.FirmVer

データ型

文字列型 (String)

構文の指定項目は次のようになります。

指定項目	データ型	内 容
<i>object</i>		オブジェクトを指すオブジェクト式です。

4.1.6 IP プロパティ

通信するIOB30RTAのIPアドレスを設定します。値の取得も可能です。デザイン時に使用できます。

構文

object.IP [=value]

データ型

文字列型 (String)

構文の指定項目は次のようになります。

指定項目	データ型	内 容
<i>object</i>		オブジェクトを指すオブジェクト式です。
<i>value</i>	String	IOB30RTA のIPアドレスを指定します。

4.1.7 KernelTime プロパティ

IOB30RTAのカーネルタイムカウンタを返します。

構文

object.KernelTime

データ型

長整数型 (Long)

構文の指定項目は次のようになります。

指定項目	データ型	内 容
<i>object</i>		オブジェクトを指すオブジェクト式です。

IOB30RTA

4.1.8 LocalPort プロパティ

使用するローカル ポートを設定します。値の取得も可能です。デザイン時に使用できません。

構文

object.LocalPort [= value]

データ型

長整数型 (Long)

構文の指定項目は次のようになります。

指定項目	データ型	内 容
<i>object</i>		オブジェクトを指すオブジェクト式です。
<i>value</i>	Long	データの送信に使うローカル ポートを指定します。特定のポートを必要とするアプリケーションでなければ、0 を指定してください。0 を指定すると、任意のポートが選択されます。

4.1.9 MacAddress プロパティ

IOB30RTAのMACアドレスを返します。

構文

object.MacAddress

データ型

文字列型 (String)

構文の指定項目は次のようになります。

指定項目	データ型	内 容
<i>object</i>		オブジェクトを指すオブジェクト式です。

4.1.10 Port プロパティ

接続するIOB30RTAのポート番号を設定します。値の取得も可能です。デザイン時に使用できます。

構文

object.Port [= value]

データ型

長整数型 (Long)

構文の指定項目は次のようになります。

指定項目	データ型	内 容
<i>object</i>		オブジェクトを指すオブジェクト式です。
<i>value</i>	Long	接続する IOB30RTA のポート番号。既定値は 20000 です。

4.1.1.1 Timeout プロパティ

各メソッド実行時のタイムアウト時間を設定します。値の取得も可能です。デザイン時に使用できます。

構文*object*.Timeout [= value]**データ型**

長整数型 (Long)

構文の指定項目は次のようになります。

指定項目	データ型	内 容
<i>object</i>		オブジェクトを指すオブジェクト式です。
<i>value</i>	Long	タイムアウト時間 (ms) を指定する。

4.1.1.2 MachineName プロパティ

IOB30RTA の名称を設定します。値の取得も可能です。デザイン時に使用できます。

構文*object*.MachineName [= value]**データ型**

文字列型 (String)

構文の指定項目は次のようになります。

指定項目	データ型	内 容
<i>object</i>		オブジェクトを指すオブジェクト式です。
<i>value</i>	String	IOB30RTA の名称を指定する。

IOB30RTA

4.1.1.3 ErrWindowStop プロパティ

プログラム実行時に、エラーが発生したときにデバッグダイアログ画面を表示するフラグです。

構文

object. **ErrWindowStop** [= value]

構文の指定項目は次のようになります。

指定項目	データ型	内 容
<i>object</i>		オブジェクトを指すオブジェクト式です。
value	Long	1 を指定すると、デバッグダイアログ画面を表示するモードに設定されます。 デフォルトは 0(無効)です。0 に設定されている場合は、メソッドの戻り値を使用して、エラーの有無を判定することができます。

データ型

長整数型(Long)

解説

(取得例) 0

(設定例) IOB30RTA1.ErrWindowStop = 1

ErrWindowStop プロパティは、プログラム実行時のデバッグの方法を選択するフラグとして機能します。デフォルト値は 0(無効)になっており、この場合、メソッド実行時にエラーが発生した場合には、メソッドの戻り値にエラーコード(負値)が返ります。以下にコーディング例を示します。

```
Private Sub func1()  
    Dim status As Long  
    status = IOB30RTA1.GetBin      ' メソッドの実行  
    If ( status > 0 ) Then  
        Text1.Text = IOB30RTA1.BI_Value(0)  ' 1ch 目のデジタル入力状態表示  
        Text2.Text = IOB30RTA1.BI_Value(1)  ' 2ch 目のデジタル入力状態表示  
    Else  
        Text3.Text = status      ' エラーコードを表示  
    End If  
End Sub
```

本プロパティを 1(有効)に設定すると、メソッド実行時にエラーが発生した場合には以下のようにエラーダイアログにエラーコードとエラーメッセージが表示され処理が中断されま

す。



エラーダイアログで、終了ボタンを押すと、プログラムが終了します。また、デバッグボタンを押すとVBのデバッグ機能でエラーの発生したコード行が表示されます。

(参考)

本プロパティを有効にしている場合には、On Error GoTo ラベル構文を使用することでエラーダイアログの表示を抑止しながらエラー処理を行うことが可能です。以下にコーディング例を示します。

```
Private Sub func1()
    Dim msg$
    On Error GoTo ErrNetCom ' エラー発生時にラベル(ErrNetCom)へジャンプ
    IOB30RTA1.GetAin ' メソッドの実行
    Text1.Text = IOB30RTA1.AI_Value(0) ' 1ch 目のアナログ入力値表示
    Text2.Text = IOB30RTA1.AI_Value(1) ' 2ch 目のアナログ入力値表示
ErrNetCom:
    msg$ = "エラー:" & _
        "" & Err.Description & "" & _
        "(" & Format(Err.Number) & ")"
    sbrMain.Panels(1).Text = msg$ ' ステータスバーにエラーを表示する
End Sub
```

IOB30RTA

4.2 メソッド

4.2.1 メソッド一覧

項番	名称	説明
1	GetAin	アナログ入力(AI)全チャンネルの状態を取得
2	GetBin	接点入力(BI)とリレー出力(BO)全チャンネルの状態を取得
3	Calibrate	A/Dコンバータの校正を行う
4	GetMachineInfo	装置情報の取得
5	SetBout	リレー出力(BO)各チャンネルに状態を設定
6	SetRange	アナログ入力の入力電圧レンジ、及び取込周波数の設定

4.2.2 GetAin メソッド

アナログ入力全チャンネルの状態を取得します。

構文

object. **GetAin**

データ型

ブール型 (Boolean)

構文の指定項目は次のようになります。

指定項目	データ型	内 容
<i>object</i>		オブジェクトを指すオブジェクト式です。

戻り値

値	内容
True	正常終了
False	現在A/D変換中(状態取得失敗)

解説

各チャンネルの状態は本メソッド実行後AI_Valueプロパティより取得します。
アナログ各チャンネルの値は、計測レンジの最小最大に対し、ユニポーラ(正極)で計測している場合、0～65535の長整数値が、バイポーラ(正負極)の場合は、-32768～32767の長整数値として取得することができます。

なお、このメソッドに対するIOB30RTAからの応答時間は、アナログレンジ(取込周波数[入力レート])の設定に大きく影響し一定ではありません。詳しくは、アナログ・デジタル I/O 装置(KCXH-IOB30RTA)取扱説明書をご参照ください。

4.2.3 GetBin メソッド

接点入力(BI)とリレー出力(BO)全チャンネルの状態を取得します。

構文

object. GetBin

構文の指定項目は次のようになります。

指定項目	データ型	内 容
<i>object</i>		オブジェクトを指すオブジェクト式です。

戻り値 (v1.00 より追加)

値	内容
0	正常終了
負値(<0)	エラー(エラーコード一覧参照)

解説

各チャンネルの状態は本メソッド実行後BI_Valueプロパティ、及びBO_Valueプロパティより取得します。

計測値が、0 の場合は、OFF(端子オープン)、1 の場合は、ON(端子ショート)を示します。

IOB30RTA

4.2.4 Calibrate メソッド

A/Dコンバータの校正を行います。

構文

object. Calibrate

データ型

ブール型 (Boolean)

構文の指定項目は次のようになります。

指定項目	データ型	内 容
<i>object</i>		オブジェクトを指すオブジェクト式です。

戻り値

値	内容
True	正常終了
False	現在A/D変換中(校正失敗)

解説

アナログ入力チャネルの入力レンジや入力レートを変更した場合、または計測装置の設置場所の温度環境等が大きく変化した場合に実行する必要があります。

なお、このメソッドに対するIOB30RTAからの応答時間は、数秒から10数秒かかります。詳しくは、アナログ・デジタルI/O装置(KCXH-IOB30RTA)取扱説明書をご参照ください。

4.2.5 GetMachineInfo メソッド

IOB30RTA のシステム情報を取得します。

構文

object. **GetMachineInfo**

構文の指定項目は次のようになります。

指定項目	データ型	内 容
<i>object</i>		オブジェクトを指すオブジェクト式です。

戻り値 (v1.00 より追加)

値	内容
0	正常終了
負値(<0)	エラー(エラーコード一覧参照)

解説

MacAddressプロパティ、FirmVerプロパティ、及びKernelTimeプロパティは本メソッド実行後に有効となります。

IOB30RTA

4.2.6 SetBout メソッド

リレー出力(BO)各チャンネルに状態を設定します。

構文

object. **SetBout** b1, b2, b3, b4

構文の指定項目は次のようになります。

指定項目	データ型	内 容
<i>object</i>		オブジェクトを指すオブジェクト式です。
b1 ~ 4	Integer	リレー出力各チャンネルの設定値

設定値

引数 b1,b2,b3,b4 の設定値

定 数	値	説 明
BO_OFF	0	OFF(リレーオープン)
BO_ON	1	ON(リレーショート)
BO_NOCHANGE	- 1	無変更

戻り値 (v1.00 より追加)

値	内容
0	正常終了
負値(<0)	エラー(エラーコード一覧参照)

解説

設定した状態はBO_Valueプロパティにも反映します。

設定した値が上記設定値以外の場合、そのチャンネルの状態は変更されません。

4.2.7 SetRange メソッド

アナログ入力各チャンネルの入力電圧レンジ、及び取込周波数(入力レート)を設定します。

構文

object. **SetRange** rt1, rt2, rt3, rt4, rt5, rt6, rt7, rt8

データ型

ブール型 (Boolean)

構文の指定項目は次のようになります。

指定項目	データ型	内容
<i>object</i>		オブジェクトを指すオブジェクト式です。
rt1 ~ 8	Integer	指定するレンジ、レートを示す値の合計値を指定します。

設定値

引数 rt1 ~ 8 の設定値

定数	値	説明
AI_NO_MEASURE	-1	計測なし
AI_RANGE_UNIPOLAR	0	0 ~ +2.5V
AI_RANGE_BIPOLAR	1	-2.5 ~ +2.5V
AI_RATE_188HZ	&H100	1.88Hz(有効精度 16bit 以上)
AI_RATE_376HZ	&H200	3.76Hz(有効精度約 16bit)
AI_RATE_751HZ	&H300	7.51Hz(有効精度約 15bit)
AI_RATE_150HZ	0	15.0Hz(有効精度約 14bit) [デフォルト]
AI_RATE_300HZ	&H400	30.0Hz(有効精度約 13bit)
AI_RATE_616HZ	&H500	61.6Hz(有効精度約 12bit)
AI_RATE_845HZ	&H600	84.5Hz(有効精度約 10bit 以下)
AI_RATE_1011HZ	&H700	101.1Hz(有効精度約 16bit 以下)

戻り値

値	内容
True	正常終了
False	現在A/D変換中(設定失敗)

解説

入力レートを上げると取込データの有効精度が下がりますので注意して下さい。
計測する必要の無いチャンネルが存在する場合は、計測なし(AI_NO_MEASURE)を指定してください。この場合、該当チャンネルのA/D変換が抑止され計測時間(GetA

IOB30RTA

inメソッドのレスポンス)の短縮に役立ちます。抑止されたチャンネルの計測値は0となります。

設定した値が上記設定値以外の場合、そのチャンネルは計測なしになります。

4.3 イベント

イベントはありません。

4.4 エラーコード一覧

コード*	エラー内容
-100	IOB30RTA との通信タイムアウトが発生しました。
-200	ソケット作成時にエラーが発生しました。
-201	ソケットバインド時にエラーが発生しました。
-202	ソケットクローズ時にエラーが発生しました。
-203	送信 SELECT でエラーが発生しました。
-204	受信 RECVFROM でエラーが発生しました。
-300	パケットID取得時にエラーが発生しました。
-301	パケットコマンド取得時にエラーが発生しました。
-302	パケット不正データ取得エラーが発生しました。
-400	不正なパラメータが指定されています。
-500	AD 変換中です。
-501	IOB30RTA でエラーが発生しました。

*メソッドの返り値で取得する場合には負号(-)が付きます。

付録. サンプルプログラムの使用法

Visual Basicサンプルプログラムファイル(KCX_VB_IOB30RTA_v???.lzh)を弊社サイトからダウンロードします。(???はバージョン番号を示します)

ダウンロードしたファイルを解凍すると本OCXを使用したVBサンプルインストーラとソースコードが以下のファイル名で同梱されています。

KCX_VB_IOB30RTA_v???.EXE (VBサンプルインストーラ)

(???はバージョン番号を示します)

IOB30RTA_Vbsample01src (VBサンプルソース)

VB サンプルインストーラを実行して指示に従ってインストールして下さい。

インストールフォルダに、IOB30RTA_Vbsample???.EXE(VB サンプル実行ファイル)が作成されます。

ここでは、サンプルプログラムの使用法を説明します。(??はバージョン番号を示します)

1. 環境設定ダイアログの設定

インストールしたフォルダのサンプルプログラムをクリックして起動すると、以下の環境設定ダイアログが表示されます。



各入力項目は以下のようになります。

リモートホスト: 通信する IOB30RTA 装置に設定されている IP アドレスです。

リモートポート: 通信する IOB30RTA 装置に設定されているコントロールポート番号です。

データ取得間隔: タイマーコントロールで定期的に IOB30RTA 装置からデータ取得するインターバル時間(秒)です。

タイムアウト: IOB30RTA 装置に通信コマンドを送信時の応答タイムアウト時間(ミリ秒)です。

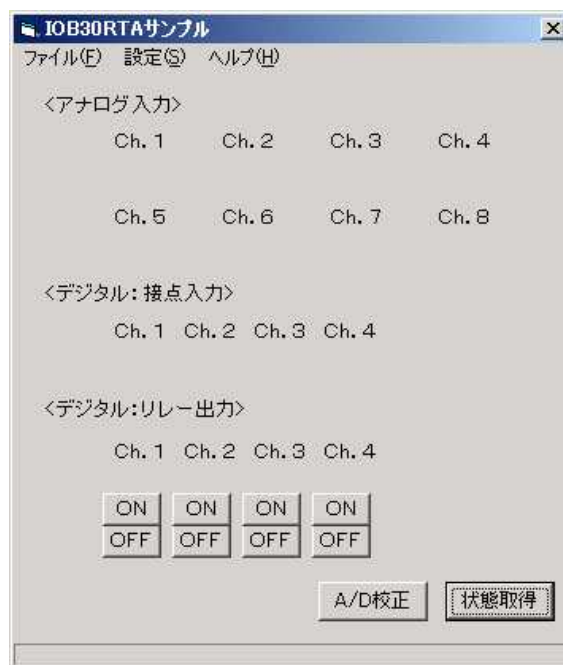
上記のうち、リモートホスト、リモートポートは、IOB30RTA の環境設定と同じ値でないと通信することが出来ませんので、よく確認して設定して下さい。

環境設定値が正しく設定出来たら、「OK」ボタンをクリックして下さい。

IOB30RTA

2. メイン情報表示画面の使い方

環境設定ダイアログで正しく接続情報を入力すると、以下のようにメイン情報表示画面が表示されます。



各項目の説明を以下に示します。

- <アナログ入力> : IOB30RTA 装置から取得したアナログ入力値をチャンネル別に表示します。
- <デジタル: 接点入力> : IOB30RTA 装置から取得した接点入力状態をチャンネル別に表示します。
- <デジタル: リレー出力> : IOB30RTA 装置のリレー出力(デジタル出力)をON / OFF操作します。また、IOB30RTA 装置から取得したデジタル出力の現在値をチャンネル別に表示します。
- 状態取得ボタン: IOB30RTA 装置にデータ取得コマンド、操作コマンドを送信して、応答ステータスを上記の各項目に表示します。
- A/D 校正ボタン: IOB30RTA 装置に搭載されているA/Dコンバータを校正します。

次に、メニューバーの各項目の説明を以下に示します。

[ファイル]メニュー：

終了(X)：本サンプルプログラムを終了します。

[設定]メニュー：

環境(F)：環境設定ダイアログを表示します。接続する装置アドレス等を変更する場合に使用します。

レンジ(R)：レンジ設定ダイアログを表示します。アナログ入力各チャンネルの入力電圧レンジ、サンプルレートを変更する場合に使用します。



タイマー(T)：タイマーコントロールを有効にします。定期的にデータ取得を行う場合は選択してチェックを入れます。

[ヘルプ]メニュー：

バージョン情報(A)：本サンプルプログラムのバージョン情報を表示します。

IOB30RTA 装置と通信して現在の入力状態値を取得するには、「状態取得」ボタンをクリックして下さい。正常に通信が成功すれば以下の画面のように取得状態値が表示されます。

IOB30RTA

