

ユーザーズ ガイド

Rev. 1.0.0.0

はじめに

- 本ドキュメントについて P.1

第 1 章 ドライバの設定

- 1.1. IO・FWL 起動設定ツールの起動 P.2
- 1.2. ドライバのロード設定の編集 P.3
- 1.3. 設定の適用 P.3

第 2 章 プログラミング

- 2.1. ファームウェアライブラリ P.4
- 2.2. 基本的な使用手順 P.14

付 録

- ドライバの制限 P.17
- 配列や構造体の宣言方法 P.17


はじめに

■ 本ドキュメントについて

本書は、INplcからINtimeの機能を利用するためのINplc-Driver「INFnCtrl」の取扱説明書です。
なお、利用にあたっては、INtimeのオブジェクト（セマフォ、セグメントなど）について理解されていることが前提となります。

関連するドキュメントは、以下のとおりです。

- INFnCtrl セットアップガイド 【INFnCtrl_Setup.pdf】
- INplc ユーザーズマニュアル 【INplc ユーザーズマニュアル.pdf】
- INplc クイックスタートガイド 【MULTIPROG クイックスタートガイド.pdf】

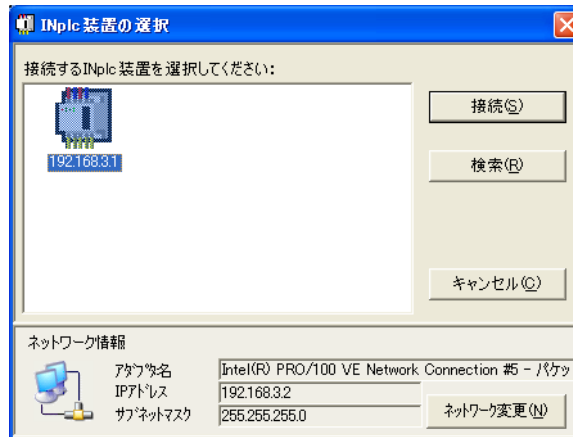
 本書では原則として、Windows XP、INplc-SDK(Pro+)の操作手順および画像を使用しています。
お使いのOSやINplc-SDKのエディション等により、記載されている操作手順や画像などが異なる場合がありますので、
適時読み替えてご利用ください。

第1章 ドライバの設定

開発 PC (INplc-SDK) から、INplc-Controller に接続し、INFnCtrl の起動および動作設定を行います。

1.1. IO・FWL 起動設定ツールの起動

1. 開発 PC 側スタートメニューのすべてのプログラムから [INplc] ▶ [INplcTool] を選択します。
2. INplc-Controller 選択ダイアログが表示されるので、一覧から INplc-Driver を適用する INplc-Controller を選択して [接続] ボタンをクリックします。

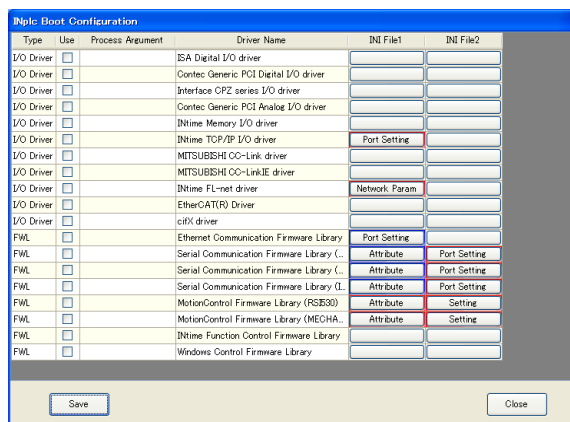


3. INplc Configuration Panel が表示されるので、[IO・FWL 起動設定] アイコンをダブルクリックします。



- ※ **INplc-Controller が稼働中 (PLC プログラムの実行中) の場合は、IO・FWL 起動設定ツールを起動できません。稼働中の場合は、MULTIPROG から PLC プログラムを停止してください。**

以上で、IO・FWL 起動設定ツールの起動は完了です。
表示される画面から、各ドライバ設定の編集ができます。



1.2. ドライバのロード設定の編集

ドライバのロード時の設定を編集します。

画面の [Driver Name] 列に **[INtime Function Control Driver]** と表記されている行の各項目を編集してください。

1) INplc でドライバをロードする設定

[Use] 列のチェックボックスを ON にすることで、INFnCtrl が INplc にロードされるようになります。

1.3. 設定の適用

変更した設定内容を、INplc-Controller に適用します。

注意) 設定の適用は、他の I/O ドライバおよびファームウェアライブラリの設定も適用されます。

1. IO・FWL 起動設定ツールの画面左下の [Save] ボタンをクリックします。
2. INplc の再起動メッセージが表示されるので、[はい] を選択して再起動を行うことで、設定が適用されます。

※ この再起動はソフトウェアの再起動です。INplc-Controller 本体の再起動ではありません。

第2章 プログラミング

2.1. ファームウェアライブラリ

ファームウェアライブラリには、以下のファンクションブロックがあります：

No.	名称	機能概要
1	mn_IMEMCPY	データコピー FU
2	LookUp	オブジェクトハンドル取得 FB
3	SendSem	セマフォ：ユニットリリース FB
4	WaitSem	セマフォ：ユニット受信 FB
5	ReadMem	セグメント：データ読み込み FB
6	WriteMem	セグメント：データ書き込み FB
7	InByte	I/Oポート：データ読み込み（1バイト）FB
8	InWord	I/Oポート：データ読み込み（2バイト）FB
9	InDWord	I/Oポート：データ読み込み（4バイト）FB
10	OutByte	I/Oポート：データ書き込み（1バイト）FB
11	OutWord	I/Oポート：データ書き込み（2バイト）FB
12	OutDWord	I/Oポート：データ書き込み（4バイト）FB

1) mn_IEMCPY

機能		このファンクションをコールすることで、SRC（コピー元データ領域）から DST（コピー先データ領域）にデータを CNT に指定したバイト数分コピーします。SRC と DST が重なっているときの動作は未定義です。 ※ SRC または DST の領域を超えるデータコピーを行うと、メモリ破壊（バッファオーバーラン/バッファオーバーフロー）を引き起こす可能性があります。十分注意してください。		
No.	入出力	パラメータ	データ型	説明
1	IN	CNT	DINT	コピー データ数を指定します。（バイト単位）
2	IN	SRC	ANY	コピー元データ領域の先頭に位置する変数（配列・構造体など）を指定します。
3	IN	SRC_OFF	DINT	SRC に設定した位置からのオフセットを指定します。（バイト単位）
4	IN	DST	ANY	コピー先データ領域の先頭に位置する変数（配列・構造体など）を指定します。
5	IN	DST_OFF	DINT	DST に設定した位置からのオフセットを指定します。（バイト単位）
6	OUT	mn_IEMCPY	INT	処理結果を示します。 0：エラーなし -1：不正なデータ領域 SRC または DST の値が不正（NULL）。 14：不正な範囲設定 データ数、オフセットがデータ領域を超過。

使用例	
[recv_msg] 配列の [0] バイト目（先頭）から [40] バイト分のデータを、[save_data] 配列の [save_posi] 変数に指定された位置にコピーする例です。	

☞ 配列・構造体の定義方法は、P.17「**配列や構造体の宣言方法**」をご覧ください。

2) LookUp

No.	入出力	パラメータ	データ型	説明
<p>機能</p> <p>このファンクションブロックをコールすることで、カタログされたINtime オブジェクト（セマフォ、セグメント）のハンドル値を取得します。</p> <p>INtime オブジェクトを管理するプロセスが削除された場合、ハンドル値は不定となります。</p> <p>ハンドル値が不定の場合は、INtime オブジェクトとアクセスが出来ませんので、再度本ファンクションブロックをコールしてハンドル値を再取得してください。</p>				
1	IN	EXE	BOOL	立ち上がりエッジ検出時に実行します。
2	IN	PROCESS	STRING	オブジェクトがカタログされているプロセスのカタログ名を指定します。オブジェクトがルートプロセスにカタログされている場合は、空文字 [""] を指定します。 ※ プロセスは、ルートプロセスにカタログされている必要があります。
3	IN	NAME	STRING	オブジェクトのカタログ名を指定します。
4	IN	DONE	BOOL	ハンドル値の取得完了を示します。
5	OUT	ERROR	BOOL	ハンドル値の取得失敗を示します。
6	OUT	ERRORID	UDINT	取得失敗時、エラーコードを示します。 0 : エラーなし 1 : ハンドル値不正 ハンドル値の取得に失敗しました。 他 : INtime システムコールエラー INtime のヘルプを参照してください。
7	OUT	HANDLE	UINT	取得したハンドル値が格納されます。
<p>使用例</p> <p>[TEST_AP] プロセスにカタログされている [SHARE_MEM] のハンドル値を取得する例です。</p>				

3) SendSem

機能		このファンクションブロックをコールすることで、指定したセマフォに対し、ユニットをリリースします。		
No.	入出力	パラメータ	データ型	説明
1	IN	EXE	BOOL	立ち上がりエッジ検出時に実行します。
2	IN	HANDLE	UINT	FB「LookUp」で取得したセマフォのハンドル値を指定します。
3	IN	UNIT	UINT	セマフォにリリースするユニット数を指定します。
4	OUT	DONE	BOOL	リリース完了を示します。
5	OUT	ERROR	BOOL	リリース失敗を示します。
6	OUT	ERRORID	UDINT	リリース失敗時、エラーコードを示します。 0 : エラーなし 4 : ユニット数超過 指定したユニット数が多すぎます。 32770 : ハンドル値不正 指定したハンドル値がセマフォ以外。 他 : INtime システムコールエラー INtime のヘルプを参照してください。
使用例				
<p>[hdl_sem] 変数に指定されたハンドル値のセマフォに [1] つのユニットをリリースする例です。</p>				

4) WaitSem

機能		このファンクションブロックをコールすることで、指定したセマフォから、ユニットを受け取ります。			
No.	入出力	パラメータ	データ型	説明	
1	IN	EXE	BOOL	立ち上がりエッジ検出時に実行します。	
2	IN	HANDLE	UINT	FB「LookUp」で取得したセマフォのハンドル値を指定します。	
3	IN	UNIT	UINT	取得するユニット数を指定します。	
4	OUT	DONE	BOOL	取得完了を示します。	
5	OUT	ERROR	BOOL	取得失敗を示します。	
6	OUT	ERRORID	UDINT	取得失敗時、エラーコードを示します。 0 : エラーなし 1 : ハンドル値不正 指定したハンドル値がセマフォ以外。 他 : INtime システムコールエラー INtime のヘルプを参照してください。	
使用例					
[hdl_sem] 変数に指定されたハンドル値のセマフォから [1] つのユニットを取得する例です。					

5) ReadMem

機能		このファンクションブロックをコールすることで、指定したセグメント（共有メモリ）の内容を取得します。		
No.	入出力	パラメータ	データ型	説明
1	IN	EXE	BOOL	立ち上がりエッジ検出時に実行します。
2	IN	HANDLE	UINT	FB「LookUp」で取得したセグメントのハンドル値を指定します。
3	IN	OFFSET	UDINT	セグメントの先頭アドレスからの読み込み位置を指定します。（バイト単位）
4	IN	SIZE	UDINT	読み込むデータ数を指定します。（バイト単位）
5	IN/OUT	DATA	ANY	読み込んだデータを格納する変数（配列・構造体など）を指定します。
6	OUT	DONE	BOOL	データ取得完了を示します。
7	OUT	ERROR	BOOL	データ取得失敗を示します。
8	OUT	ERRORID	UDINT	データ取得失敗時、エラーコードを示します。 0：エラーなし 32770：ハンドル値不正 指定したハンドル値がセグメント以外。 32783：マップ失敗 セグメントのマップに失敗。 226：領域不正 データ数、オフセットがDATAの領域を超過。 他：INtime システムコールエラー INtime のヘルプを参照してください。

使用例	
<p>[hdl_mem] 変数に指定されたハンドル値のセグメントの [256] バイト目から [128] バイト分のデータを、[recv_msg] 配列に取得する例です。</p>	

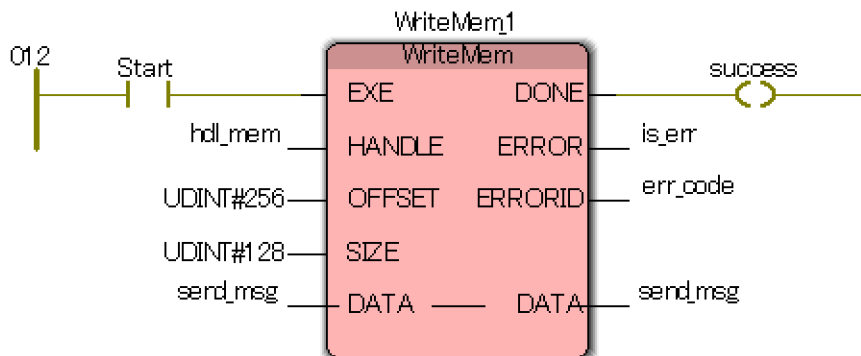
☞ 配列・構造体の定義方法は、P.17「**配列や構造体の宣言方法**」をご覧ください。

6) WriteMem

機能		このファンクションブロックをコールすることで、指定したセグメント（共有メモリ）にデータを書き込みます。		
No.	入出力	パラメータ	データ型	説明
1	IN	EXE	BOOL	立ち上がりエッジ検出時に実行します。
2	IN	HANDLE	UINT	FB「LookUp」で取得したセグメントのハンドル値を指定します。
3	IN	OFFSET	UDINT	セグメントの先頭アドレスからの書き込み位置を指定します。（バイト単位）
4	IN	SIZE	UDINT	書き込むデータ数を指定します。（バイト単位）
5	IN/OUT	DATA	ANY	書き込むデータが格納された変数（配列・構造体など）を指定します。
6	OUT	DONE	BOOL	データ書き込み完了を示します。
7	OUT	ERROR	BOOL	データ書き込み失敗を示します。
8	OUT	ERRORID	UDINT	データ書き込み失敗時、エラーコードを示します。 0：エラーなし 32770：ハンドル値不正 指定したハンドル値がセグメント以外。 32783：マップ失敗 セグメントのマップに失敗。 226：領域不正 データ数、オフセットがDATAの領域を超過。 32772：指定サイズ不正 データ数がDATAの領域を超過。 他：INtime システムコールエラー INtime のヘルプを参照してください。

使用例

[hdl_mem] 変数に指定されたハンドル値のセグメントの [256] バイト目へ、[send_msg] 配列のデータを [128] バイト分書き込む例です。



☞ 配列・構造体の定義方法は、P.17「**■ 配列や構造体の宣言方法**」をご覧ください。

7) InByte

機能		このファンクションブロックをコールすることで、指定 I/O ポートアドレスの値（1 バイト）を取得します。		
No.	入出力	パラメータ	データ型	説明
1	IN	EXE	BOOL	TRUE の時に常に実行します。
2	IN	PORT	UINT	取得する I/O ポートアドレスを指定します。
3	OUT	DONE	BOOL	取得完了を示します。
4	OUT	DATA	BYTE	取得した値が格納されます。

使用例

8) InWord

機能		このファンクションブロックをコールすることで、指定 I/O ポートアドレスの値（2 バイト）を取得します。		
No.	入出力	パラメータ	データ型	説明
1	IN	EXE	BOOL	TRUE の時に常に実行します。
2	IN	PORT	UINT	取得する I/O ポートアドレスを指定します。
3	OUT	DONE	BOOL	取得完了を示します。
4	OUT	DATA	WORD	取得した値が格納されます。

使用例

9) InDWord

機能		このファンクションブロックをコールすることで、指定 I/O ポートアドレスの値（4 バイト）を取得します。		
No.	入出力	パラメータ	データ型	説明
1	IN	EXE	BOOL	TRUE の時に常に実行します。
2	IN	PORT	UINT	取得する I/O ポートアドレスを指定します。
3	OUT	DONE	BOOL	取得完了を示します。
4	OUT	DATA	DWORD	取得した値が格納されます。

使用例

10) OutByte

機能		このファンクションブロックをコールすることで、指定 I/O ポートアドレスに指定値（1 バイト）を出力します。 ※ 指定する I/O ポートアドレスおよび出力するデータには十分注意してください。 誤った I/O ポートへの書き込みを行うと、INplc-Controller の動作が不安定になる可能性があります。		
No.	入出力	パラメータ	データ型	説明
1	IN	EXE	BOOL	立ち上がりエッジ検出時に実行します。
2	IN	PORT	UINT	出力する I/O ポートアドレスを指定します。
3	IN	DATA	BYTE	出力する値を指定します。
4	OUT	DONE	BOOL	出力完了を示します。

使用例

11) OutWord

機能	このファンクションブロックをコールすることで、指定I/Oポートアドレスに指定値（2バイト）を出力します。 ※ 指定するI/Oポートアドレスおよび出力するデータには十分注意してください。 誤ったI/Oポートへの書き込みを行うと、INplc-Controllerの動作が不安定になる可能性があります。			
No.	入出力	パラメータ	データ型	説明
1	IN	EXE	BOOL	立ち上がりエッジ検出時に実行します。
2	IN	PORT	UINT	出力するI/Oポートアドレスを指定します。
3	IN	DATA	WORD	出力する値を指定します。
4	OUT	DONE	BOOL	出力完了を示します。
使用例				

12) OutDWord


機能	このファンクションブロックをコールすることで、指定I/Oポートアドレスに指定値（4バイト）を出力します。 ※ 指定するI/Oポートアドレスおよび出力するデータには十分注意してください。 誤ったI/Oポートへの書き込みを行うと、INplc-Controllerの動作が不安定になる可能性があります。			
No.	入出力	パラメータ	データ型	説明
1	IN	EXE	BOOL	立ち上がりエッジ検出時に実行します。
2	IN	PORT	UINT	出力するI/Oポートアドレスを指定します。
3	IN	DATA	DWORD	出力する値を指定します。
4	OUT	DONE	BOOL	出力完了を示します。
使用例				

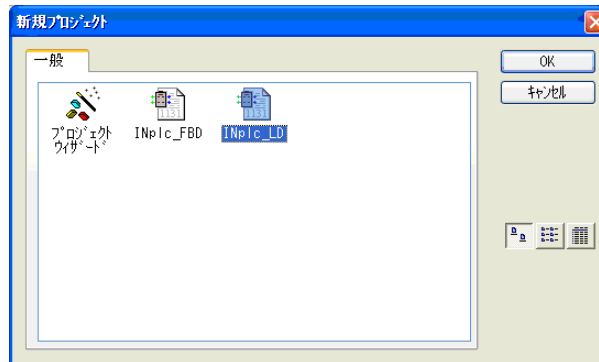
2.2. 基本的な使用手順

ラダーダイアグラム (LD) のプログラムで使用する手順を説明します。

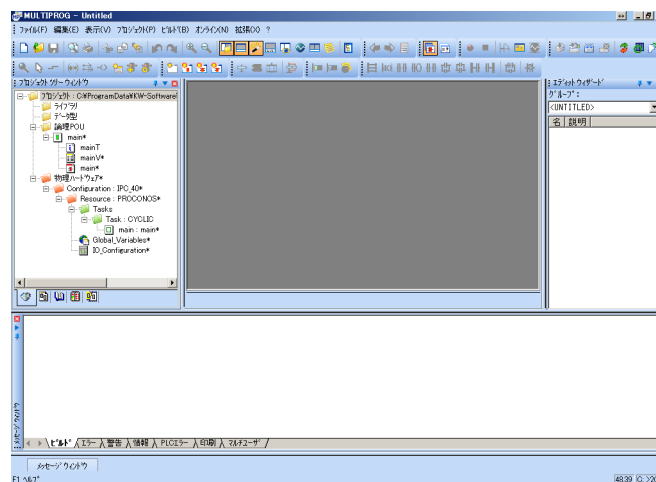
1) PLCプロジェクト作成

MULTIPROG を起動して、新規に PLC プロジェクトを作成します。

1. 開発 PC 側スタートメニューのすべてのプログラムにある、KW-Software グループ内 MULTIPROG グループの中から [MULTIPROG] を選択します。
2. MULTIPROG が起動するので、ツールバーの [新規プロジェクト] アイコン  をクリックします。
3. テンプレート一覧から [INplc_LD] を選択します。



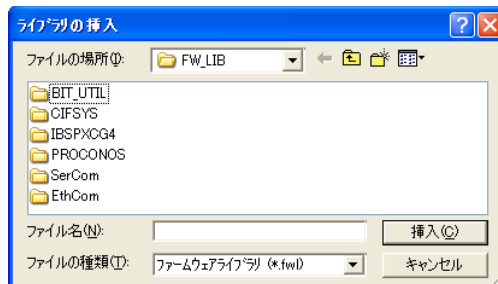
新しいプロジェクトが作成され、画面左側のプロジェクトツリーウィンドウにプロジェクトツリーが表示されます。



2) ファームウェアライブラリの挿入

PLCプログラムにファームウェアライブラリを挿入します。

1. プロジェクトツリーの [ライブラリ] を右クリックして、表示されるメニューの [挿入] > [ファームウェアライブラリ] を選択します。
2. ライブラリの挿入ダイアログが表示されます。



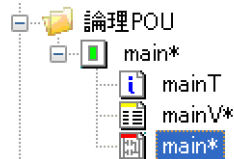
3. 一覧の「INFnCtrl」フォルダ内の『INFnCtrl.FWL』を選択して [挿入] ボタンをクリックします。


プロジェクトツリーの [ライブラリ] の中に [INFnCtrl] が追加されます。

3) プログラムコードの作成

取得した入力データを、そのまま出力データとして設定するプログラムを作成します。

1. プロジェクトツリーから [main] POU のコードワークシートを開きます。
開いたら、コードワークシート上をクリックして [+] カーソルを置きます。



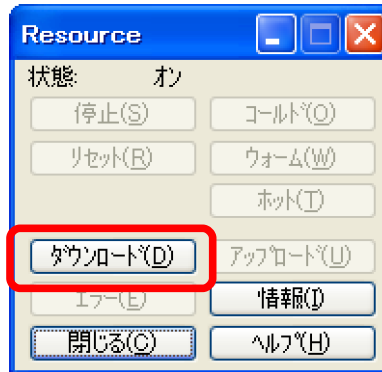
2. 画面右端の「エディット ウィザード」のグループから『<INFnCtrl>』を選択します。
3. ファンクション・ファンクションブロックの一覧が表示されるので、使用したい項目をダブルクリックします。
4. 各パラメータに、変数または接点/コイルを接続しています。
5. ツールバーの [メイク] アイコン  をクリックして、プロジェクトのコンパイルを行います。

コンパイルが正常に完了したら、プログラムコードの作成は完了です。

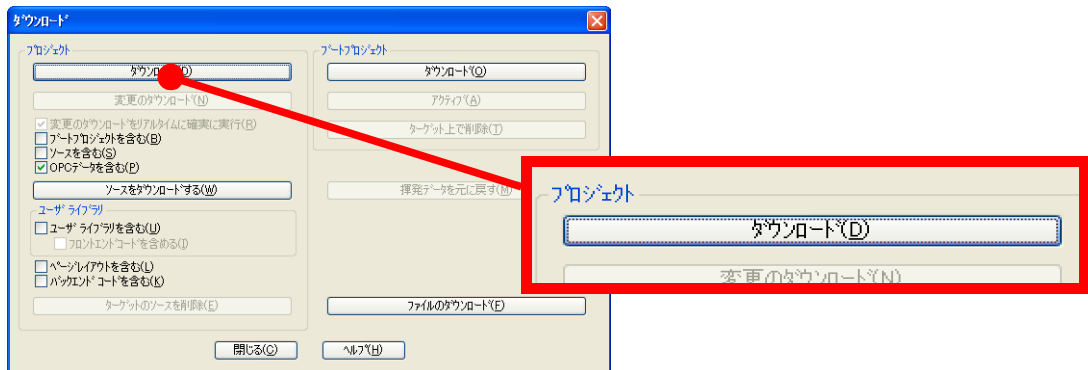
4) プロジェクトの実行・モニタリング

プロジェクトをINplc-Controllerにダウンロードして実行し、モニタリングを行います。

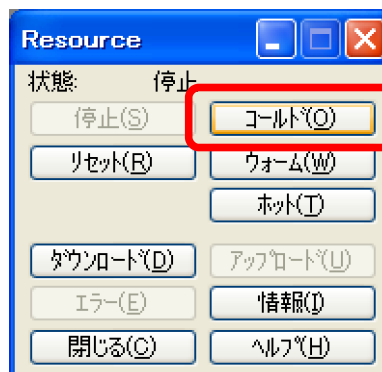
1. オンラインメニューの [プロジェクト コントロール] を選択してリソースダイアログを開き、[ダウンロード] ボタンをクリックします。



2. ダウンロードダイアログが表示されるので、[プロジェクト] の [ダウンロード] ボタンをクリックします。



3. ダウンロードが完了したら、リソースダイアログの [コールド] ボタンをクリックして、PLCプログラムを稼動状態にします。



4. ツールバーの [デバッグのオン/オフ] アイコン  をクリックしてオンライン モードに切り替えます。

オンライン値を見たり、デバッグ ダイアログから値を変更することができます。

付 録

■ ドライバの制限

- データのコピーの際に誤った設定で実行すると、メモリ破壊（バッファオーバーラン/バッファオーバーフロー）を引き起こす可能性があります。十分注意してご利用ください。
- I/O ポートアクセス機能は、I/O ポートに直接アクセスすることの危険性について理解した上で、正しくお使いください。本機能を利用して発生したいかなる損害に対しても補償いたしかねます。

■ 配列や構造体の宣言方法

配列や構造体などのユーザー定義データ型は、データ型ワークシート に定義します。



【プロジェクト】メニューの【データ型の追加】を選択すると、挿入ダイアログが表示されます。

必要な項目を入力して【OK】ボタンを押すと、プロジェクトツリーの【データ型】の中にデータ型ワークシートが追加されます。

配列 [ARRAY]、構造体 [STRUCT] を定義してワークシートをコンパイルすると、各 POU から定義したデータ型を使用できるようになります。

```
1 TYPE
2   ByteBuff : ARRAY [0..1023] OF BYTE
3 END_TYPE
4 |
```



INplc-Driver [INFnCtrl] ユーザーズガイド

株式会社 **マイクロネット**

- ▷ Windows は、米国 Microsoft Corporation の米国およびその他の国における商標または登録商標です。
- ▷ MULTIPROG と ProConOS は、KW-Software GmbH, Langenbruch 6, 32657 Lemgo, Germany の登録商標です。
- ▷ その他、本書に記載されている会社名、商品名は、各社の商標または登録商標です。
- ▷ 本書の内容を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、予告なしに変更することがあります。あらかじめご了承ください。