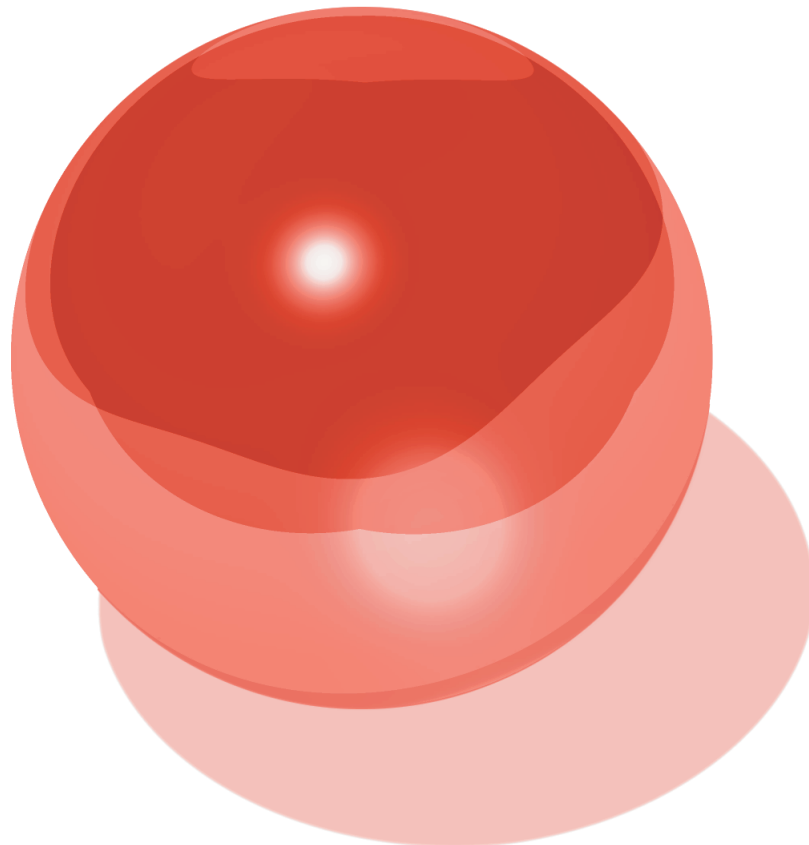


μGPCSX series

プログラミングマニュアル
オペレーション編



このたびは、TOYO FAデジタルコントローラ μ GPCsxをお買い上げいただきまことにありがとうございます。

このプログラミングマニュアル オペレーション編は、TDsxEditor のメニュー、アイコンなどの説明およびオペレーションについて解説したものです。 μ GPCsx を正しくお使いいただくために、このプログラミングマニュアルをよくお読みください。

また、下表に示す関連マニュアルも併せてお読みくださるようお願いいたします。

名称	マニュアル番号	記載内容
μ GPCsx シリーズ プログラミング マニュアル (命令語編)	IGJ057A	μ GPCsx シリーズのメモリ、言語、システム定義の内容などを解説
μ GPCsx シリーズ プログラミング マニュアル (技術編)	IGJ059A	プログラムの組み方、作成方法について解説
μ GPCsx シリーズ ユーザーズ マニュアル (ハードウェア編)	IGJ060A	GPCsx シリーズのシステム構成、各モジュールのハードウェア仕様などを解説

ご注意


- (1) 本書の内容の一部または全部を無断で転載、複製することは禁止されております。
- (2) 本書の内容に関しては、改良のため予告なしに仕様などを変更することがありますのでご了承ください。
- (3) 本書の内容に関しては万全を期しておりますが、万一ご不審な点や誤りなどお気付きのことがありましたら、お手数ですが巻末記載の弊社営業所までご連絡ください。その際、表紙記載のマニュアル番号も併せてお知らせください。

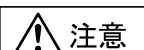


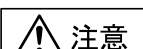


本製品をご使用の前に「安全上のご注意」をよくお読みの上、正しくご使用ください。

ここでは、安全上の注意事項のレベルを「危険」および「注意」として区分しており、意味は下記のとおりです。

 **危険** : 取り扱いを誤った場合に、死亡または重傷を受ける可能性があります。

 **注意** : 取り扱いを誤った場合に、中程度の障害や軽傷を受ける可能性、あるいは物的損傷が発生する可能性があります。


なお、 **注意** に記載した事項でも、状況によっては重大な結果に結びつく可能性があります。

いずれも重要な内容を記載しておりますので、必ず守ってください。

特に注意していただきたい点を以下に示しますが、マニュアルの本文中にも上記記号で示します。

 **危険**

非常停止回路・インタロック回路などは、PCの外部で構成してください。
PCの故障により、機械の破損や事故のおそれがあります。

 **注意**

運転中のプログラム変更、強制出力、起動、停止などの操作は十分安全を確認してから行ってください。
操作ミスにより機械が動作し、機械の破損や事故のおそれがあります。





.....

マニュアル番号は、このマニュアルの表紙の右下に記載しております。

印刷日付	マニュアル番号	改訂内容
2001年5月	IGJ058A	初版印刷（暫定版）

.....



はじめに

安全上のご注意

改定履歴

目次

第 1 章 システムの準備と立ち上げ	1 - 1
1 - 1 TDsxEditor のシステム構成	1 - 1
1 - 1 - 1 TDsxEditor のシステム構成	1 - 1
1 - 2 システムの必要事項	1 - 2
1 - 2 - 1 ハードウェアの必要事項	1 - 2
1 - 2 - 2 ソフトウェアの必要事項	1 - 2
1 - 3 インストールとアンインストール	1 - 3
1 - 3 - 1 インストールの方法	1 - 3
1 - 3 - 2 アンインストール	1 - 6
第 2 章 TDsxEditor ユーザーインターフェイス	2 - 1
2 - 1 TDsxEditor の画面構成と機能	2 - 1
2 - 1 - 1 TDsxEditor の画面構成	2 - 1
2 - 1 - 2 メニューバー	2 - 2
2 - 1 - 3 ツールバー	2 - 2
第 3 章 プロジェクト作成	3 - 1
3 - 1 プロジェクトについて	3 - 1
3 - 1 - 1 TDsxEditor のウインドウ	3 - 1
3 - 1 - 2 プロジェクトとは	3 - 2
3 - 1 - 3 タスク 1、タスク 2 とは	3 - 3
3 - 1 - 4 μ GPCsx システムでのプロジェクト	3 - 4
3 - 1 - 5 CPU タイプ	3 - 5
3 - 2 プロジェクト編集	3 - 6
3 - 2 - 1 プロジェクト関係処理メニュー操作	3 - 6
3 - 2 - 2 ツリーノードによるポップアップメニュー	3 - 7
3 - 2 - 3 サブプログラム関係編集操作	3 - 8
第 4 章 回路編集	4 - 1
4 - 1 サブプログラム、サブルーチンの回路編集	4 - 1
4 - 1 - 1 回路ウインドウのモード	4 - 1
4 - 2 書き込みモード	4 - 2
4 - 2 - 1 書き込みモードの基本操作	4 - 2
4 - 2 - 2 A 接点の挿入の仕方	4 - 5

4 - 2 - 3	コイルの入力の仕方	4 - 6
4 - 2 - 4	タイマの指定	4 - 7
4 - 2 - 5	データフロー記述の仕方	4 - 8
4 - 2 - 6	定数表現について	4 - 9
4 - 2 - 7	関数シンボル	4 - 10
4 - 2 - 8	ポップアップメニュー	4 - 11
4 - 3	サブルーチンの設計方法	4 - 12
4 - 3 - 1	サブルーチン追加	4 - 12
4 - 3 - 2	サブルーチン編集	4 - 12
4 - 4	回路一覧	4 - 13
4 - 4 - 1	回路一覧での操作	4 - 13
4 - 5	メニューによる操作	4 - 14
4 - 5 - 1	「ファイル」メニュー	4 - 14
4 - 5 - 2	「編集」メニュー	4 - 14
4 - 5 - 3	「表示」	4 - 17
4 - 6	オンラインでの回路	4 - 18
4 - 6 - 1	モニタ	4 - 18
4 - 6 - 2	サブルーチン内のモニタ	4 - 19
4 - 6 - 3	デバッグ	4 - 20
第5章	その他の項目の編集	5 - 1
5 - 1	リレー、レジスタ使用点数割付	5 - 1
5 - 2	定数データ	5 - 2
5 - 2 - 1	整数データ (ki)、実数データ (kr)	5 - 2
5 - 2 - 2	オンタイマ (TS)、オフタイマ (TR)	5 - 3
5 - 3	パターンデータ	5 - 4
5 - 4	メモリ転送定義	5 - 5
5 - 5	トレースバック	5 - 6
5 - 5 - 1	トレースバック	5 - 6
5 - 5 - 2	トレースバック設定ウインドウ	5 - 6
5 - 5 - 3	トレースバック設定ウインドウ リレー設定部	5 - 7
第6章	システム定義情報	6 - 1
6 - 1	システム定義情報	6 - 1
6 - 2	システム定義情報の種類	6 - 1
6 - 3	システム構成定義の編集	6 - 2
6 - 3 - 1	ツールバーボタン	6 - 2
6 - 3 - 2	システム構成定義例	6 - 2
6 - 3 - 3	リモート I/O 使用時の定義例	6 - 2
6 - 3 - 4	モジュール挿入、プロパティ	6 - 3

6 - 3 - 5	IOレジスタの割付	6 - 5
6 - 3 - 6	CPU モジュールパラメータメモリ境界定義	6 - 6
6 - 3 - 7	μGPCsx システムでのデータメモリの考え方	6 - 7
6 - 3 - 8	メモリ種別	6 - 8
第7章	オンライン機能	7 - 1
7 - 1	リレー表示	7 - 1
7 - 2	レジスタ表示	7 - 2
7 - 3	トレンドグラフ	7 - 3
7 - 3 - 1	トレンドグラフ	7 - 3
7 - 3 - 2	トレンドメニュー	7 - 3
7 - 3 - 3	レジスタ表示項目編集	7 - 5
7 - 3 - 4	リレー表示項目編集	7 - 6
第8章	印刷	8 - 1
8 - 1	印刷の概要	8 - 1
8 - 1 - 1	画面操作方法	8 - 1
8 - 1 - 2	プリンタ設定の確認と変更	8 - 2
8 - 1 - 3	図枠印字設定	8 - 2
8 - 2	個別印刷	8 - 3
8 - 2 - 1	[回路リスト]印刷	8 - 3
8 - 2 - 2	[システム定義]印刷	8 - 4
8 - 2 - 3	[パラメータ]印刷	8 - 4
8 - 2 - 4	[クロスリファレンス]印刷	8 - 5
8 - 2 - 5	[接点コメント]印刷	8 - 5
8 - 2 - 6	[その他]印刷	8 - 5
9章	環境設定、書き込みモードカスタマイズ	9 - 1
9 - 1	環境設定	9 - 1
9 - 1 - 1	色の設定	9 - 1
9 - 1 - 2	ツール設定	9 - 2
9 - 1 - 3	接続先通信設定	9 - 3
9 - 1 - 4	USB ドライバのセットアップ	9 - 4
9 - 2	書き込みモードカスタマイズ	9 - 5



第1章 システムの準備と立ち上げ

1 - 1	TDsxEditor のシステム構成	1 - 1
1 - 1 - 1	TDsxEditor のシステム構成	1 - 1
1 - 2	システムの必要事項	1 - 2
1 - 2 - 1	ハードウェアの必要事項	1 - 2
1 - 2 - 2	ソフトウェアの必要事項	1 - 2
1 - 3	インストールとアンインストール	1 - 3
1 - 3 - 1	インストールの方法	1 - 3
1 - 3 - 2	アンインストール	1 - 6

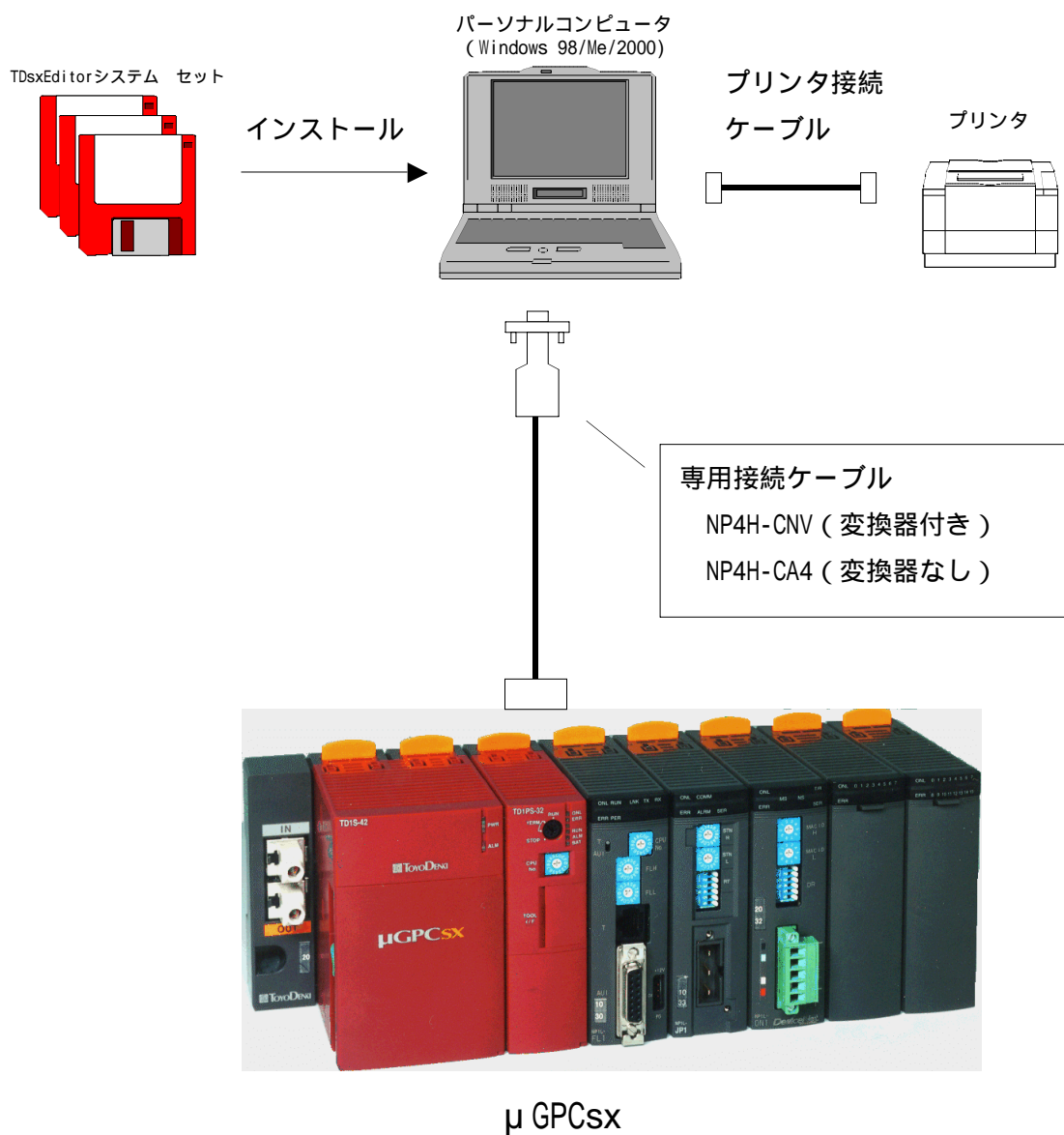


第 1 章 システムの準備と立ち上げ

1 - 1 μGPCsx プログラミングツールシステムの構成

1 - 1 - 1 μGPCsx プログラミングツールシステムの構成

下図のようにパーソナルコンピュータ上に TDsxEditor (システムソフトウェア) をインストールすることにより、μGPCsx のプログラミングツールとして使用できます。



1 - 2 システムの必要事項

1 - 2 - 1 ハードウェアの必要事項

TDsxEditor を動作させるには、必ず次のハードウェアの必要事項を満足してください。

- ・ Intel Pentium (300MHz 以上推奨) の IBM コンパチブルパソコン、DOS/V パソコン。
- ・ WindowsVGA 解像度 800×600 ドット以上 (SVGA 解像度 1024×768 ドット推奨)
- ・ 100MB 以上のハードディスクの空き容量。
- ・ 32MB 以上のメモリ。
- ・ 3.5 インチフロッピーディスクドライブ (1.44MB フォーマットのフロッピーが読み出し可能であること。) もしくは CD-ROM ドライブ。

1 - 2 - 2 ソフトウェアの必要事項

TDsxEditor を動作させるには、いずれかのオペレーティングシステムが必要です。

Microsoft Windows98/Me (日本語版・英語版)

Microsoft Windows2000 (日本語版・英語版)

Pentium、Windows は登録商標です。

1 - 3 インストールとアンインストール

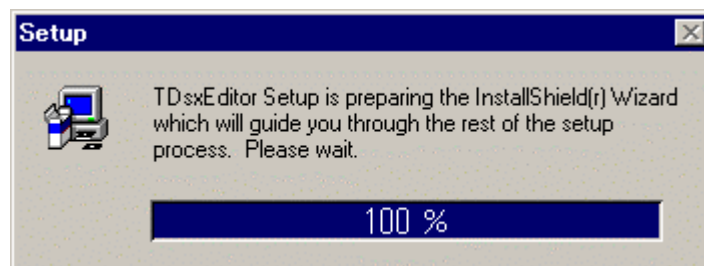
1 - 3 - 1 インストールの方法

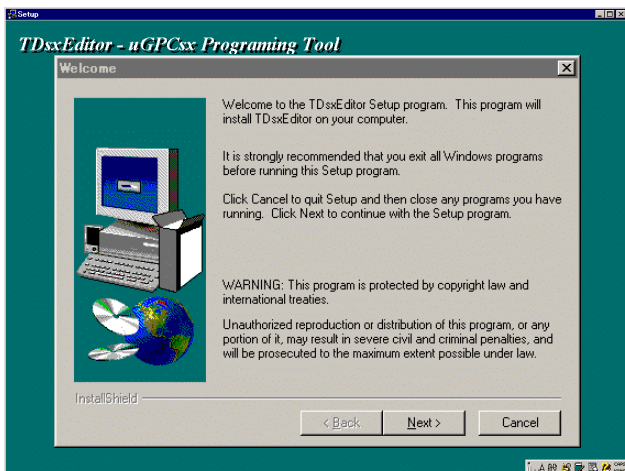
TDsxEditor ソフトウェアパッケージは、複数のフロッピーディスクで出荷されます。インストールディスクには、インストールに必要な動作およびアイコン登録などを自動実行するインストールプログラムが付属されています。

ネットワークを使用してインストールする場合、ネットワークの環境および使用環境により、コピーやインストールが正常に行われない場合があります。

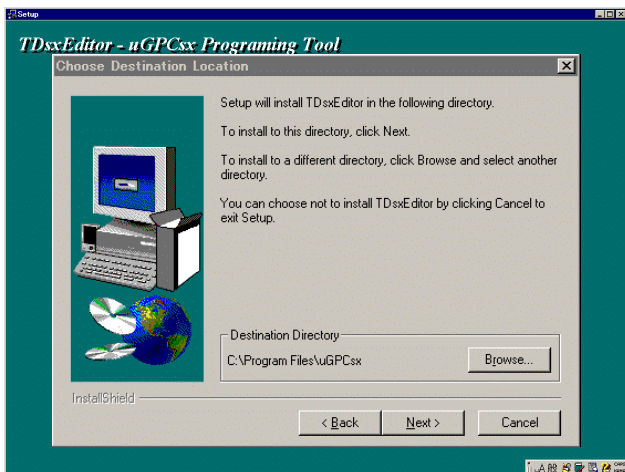
インストール

- (1) ウイルス検出ソフトウェアおよびスクリーンセーバなどを動作しない状態にします。
 - (2) Windows98/Me の「スタート」メニューの「設定(S)」サブメニューからの「コントロールパネル(C)」を選択します。
 - (3) 「コントロールパネル」ダイアログ内の「アプリケーションの追加と削除」アイコンを左クリックします。
 - (4) 「インストール(I)」ボタンを左クリックします。
 - (5) インストールプログラムの入っている(Disk1)をフロッピーディスクドライブに挿入します。
 - (6) 「次へ(N)>」ボタンを左クリックします。
 - (7) インストールプログラムのコマンドライン(C)」のテキストボックスに<A: /Setup.exe>が表示されていることを確認します。もし表示されていない場合は、「参照(R)」ボタンを左クリックし、フロッピーディスクのドライブ を選択し、ファイル名「Setup.exe」を選択します。
 - (8) 「Install Shield Wizard」準備中ワーキングボックス表示されます。
- 注) インストールプログラムは英語版で作成されていますが、TDsxEditor が起動すると自動的にOSの言語を読み出しますので、日本語版OSでは日本語版TDsxEditorとして起動します。

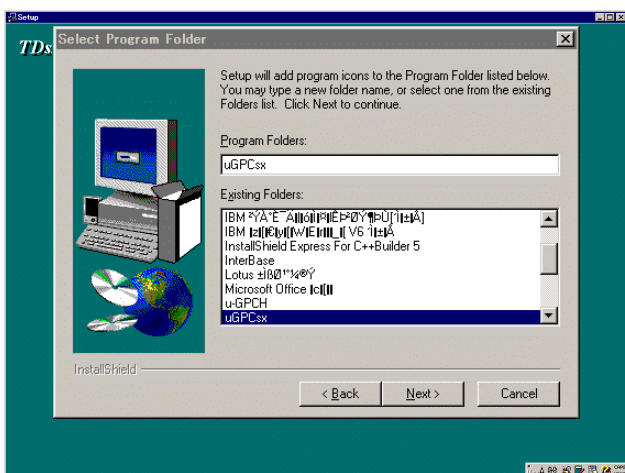




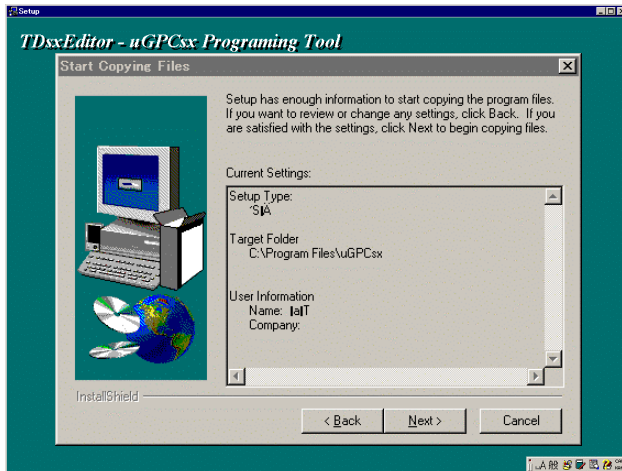
左記ダイアログが表示されますので
「Next >」でインストールを始めます。
「Cancel」でインストールを中止します。



インストールするフォルダ名を指定します。
フォルダ名を変更しない場合は、
「Next >」を左クリックします。
フォルダ名を変更する場合は、「Browse」に
てフォルダ名を指定して下さい。

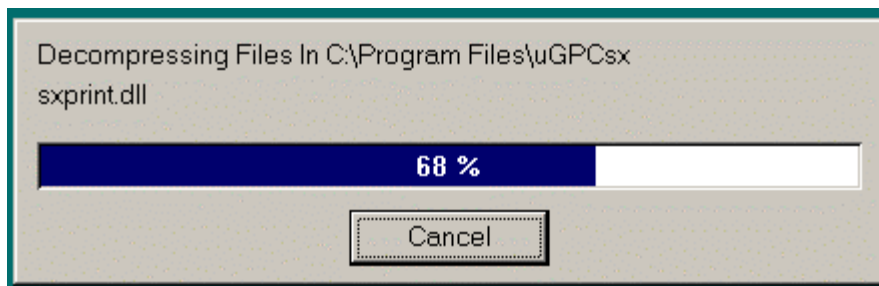


インストールするグループを選択します。
通常は変更しないで下さい。
注) フォントの関係により既存日本語アプリ
ケーショングループ名が文字化けしてしま
すが、動作には影響ありません。

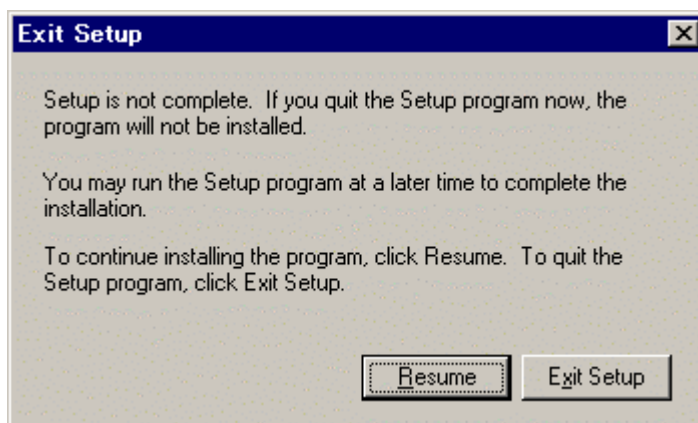


インストール確認ダイアログが表示されます。
表示されている内容でよろしければ
「Next >」でコピーを始めます。

コピー中のメッセージ中は「Cancel」ボタンを左クリックするとコピーを中断できます。



「Cancel」ボタンを左クリックすると下記のダイアログが表示されますので



「Resume」でインストールを続行します。

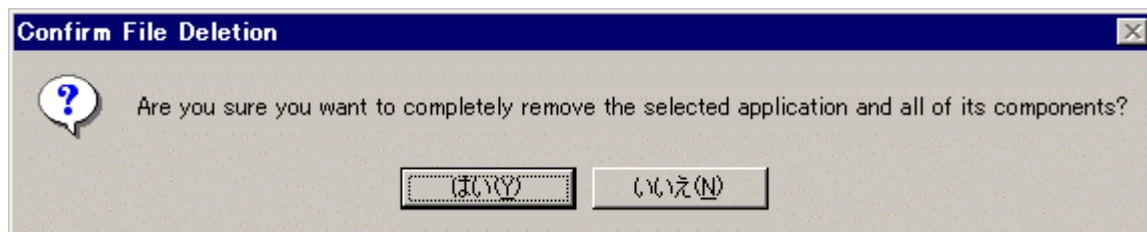
「Exit Setup」でインストールを終了します。

1 - 3 - 2 アンインストール

- (1) Windows98/Me の「スタート」メニューの「設定(S)」サブメニューからの「コントロールパネル(C)」を選択します。
- (2) 「コントロールパネル」ダイアログ内の「アプリケーションの追加と削除」アイコンを左クリックします。
- (3) TDsxEditor を選択して、「追加と削除(R)」を左クリックします。



メッセージボックスにて、「あなたは完全に選ばれたアプリケーション、およびすべてのそのコンポーネントを取り去りたいことは確かですか？」と問い合わせがありますので、



「はい(Y)」でアンインストールが実行されます。

「いいえ(N)」でアンインストールがキャンセルされます。



第2章 TdsxEditor ユーザーインターフェイス

第2章

2 - 1	TdsxEditor の画面構成と機能	2 - 1
2 - 1 - 1	TdsxEditor の画面構成	2 - 1
2 - 1 - 2	メニューバー	2 - 2
2 - 1 - 3	ツールバー	2 - 2



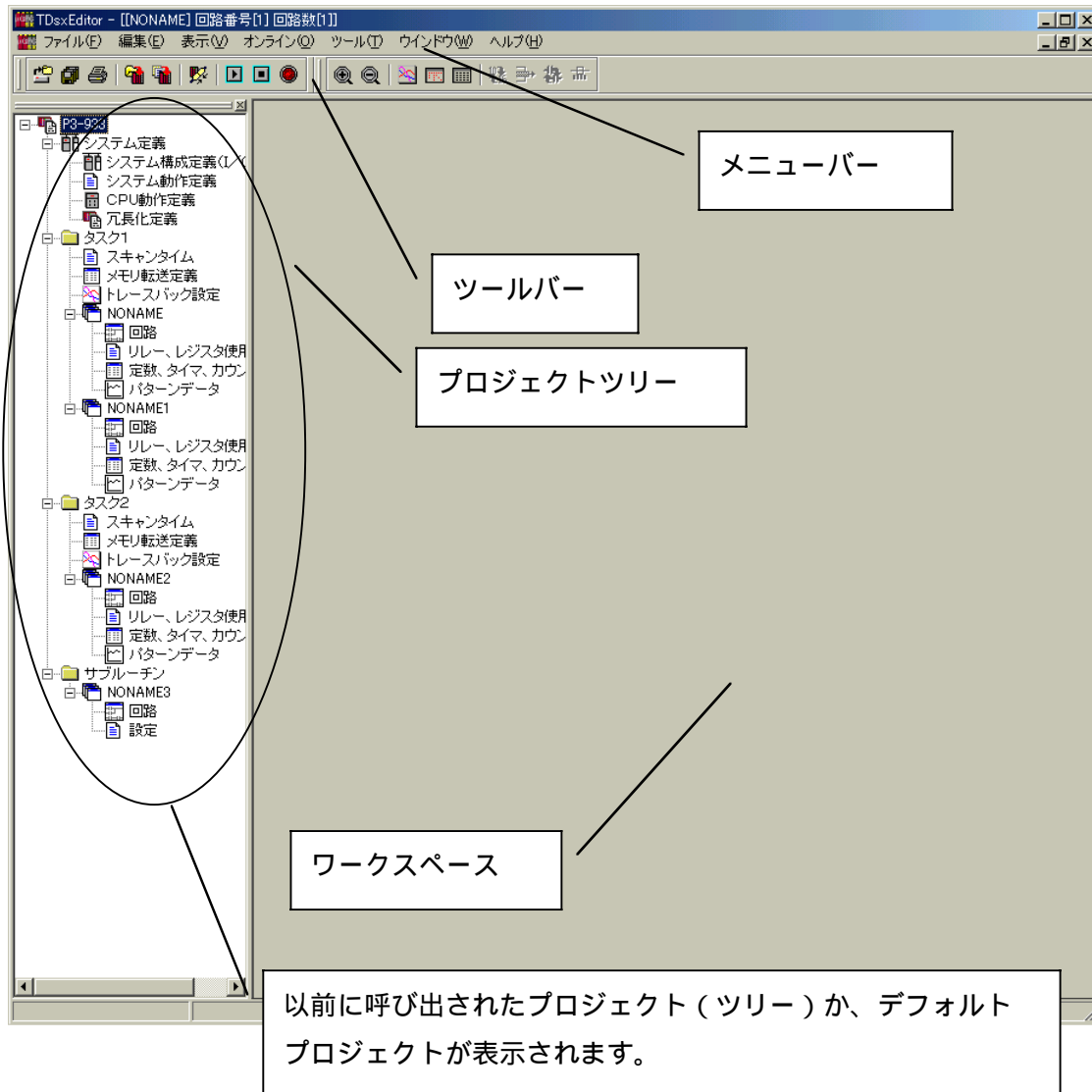


第2章 TDsxEditor ユーザーインターフェイス

2 - 1 TDsxEditor の画面構成と機能

2 - 1 - 1 TDsxEditor の画面構成

TDsxEditor を起動すると下図のような画面が表示されます。



2 - 1 - 2 メニューバー

メニューバーは各種機能を実行します。

「ファイル」

プロジェクトの構成、保存、設計、および印刷に使用するコマンドが含まれています。

「編集」

プロジェクト内のサブプログラムの構成の編集に使用するコマンドが含まれています。

「表示」

プロジェクト内のクロスリファレンスの表示、ツールバーの表示非表示に関するコマンドが含まれています。

「オンライン」

プロジェクトのダウンロードアップロード、 μ GPCsx 制御、各種 μ GPCsx 状態表示に関するコマンドが含まれています。

「ツール」

ツールの環境設定、トレースバック設定に関するコマンドが含まれています。ツールの環境設定とは各ウィンドウの色、TDsxEditor の設定、 μ GPCsx との通信設定が含まれています。

「ウィンドウ」

ウィンドウ表示の変更に使用するコマンドが含まれています。

「ヘルプ」

ヘルプの呼び出しに使用するコマンドが含まれています。

「ファイル」「編集」「表示」メニューに関しては、作業をする項目に応じて変化します。

2 - 1 - 3 ツールバー

ツールバーには複数のボタンが含まれており、これらのボタンを使用すれば使用頻度の高い機能を快適に実行できます。

第3章 プロジェクト作成

3 - 1	プロジェクトについて	3 - 1
3 - 1 - 1	TDsxEditor のウインドウ	3 - 1
3 - 1 - 2	プロジェクトとは	3 - 2
3 - 1 - 3	タスク 1、タスク 2 とは	3 - 3
3 - 1 - 4	μGPCsx システムでのプロジェクト	3 - 4
3 - 1 - 5	CPU タイプ	3 - 5
3 - 2	プロジェクト編集	3 - 6
3 - 2 - 1	プロジェクト関係処理メニュー操作	3 - 6
3 - 2 - 2	ツリーノードによるポップアップメニュー	3 - 7
3 - 2 - 3	サブプログラム関係編集操作	3 - 8

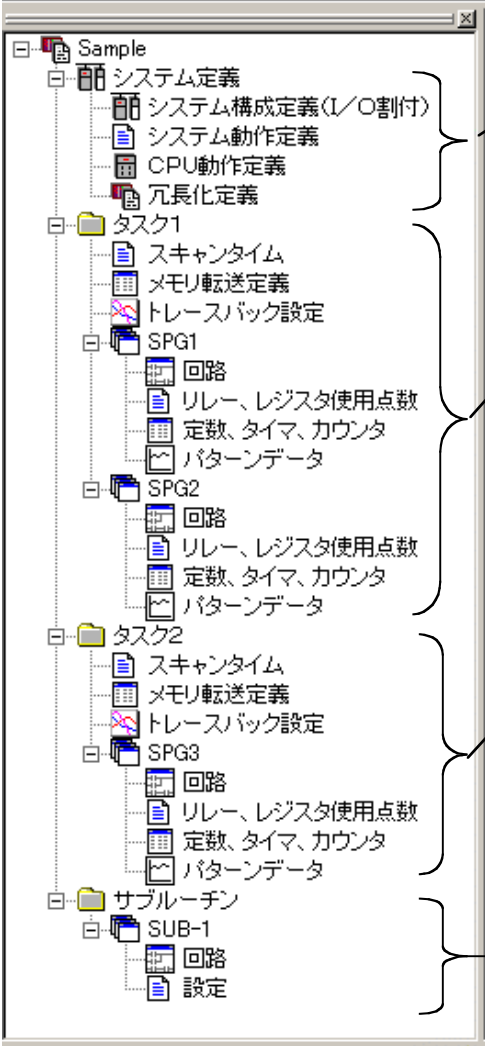


第3章 プロジェクト作成

3 - 1 プロジェクトについて

3 - 1 - 1 TDsxEditor のウインドウ

プロジェクトツリーは、編集に必要な項目が表示されるウインドウです。このツリーは「システム定義」「タスク 1」「タスク 2」「サブルーチン」というグループで構成されています。



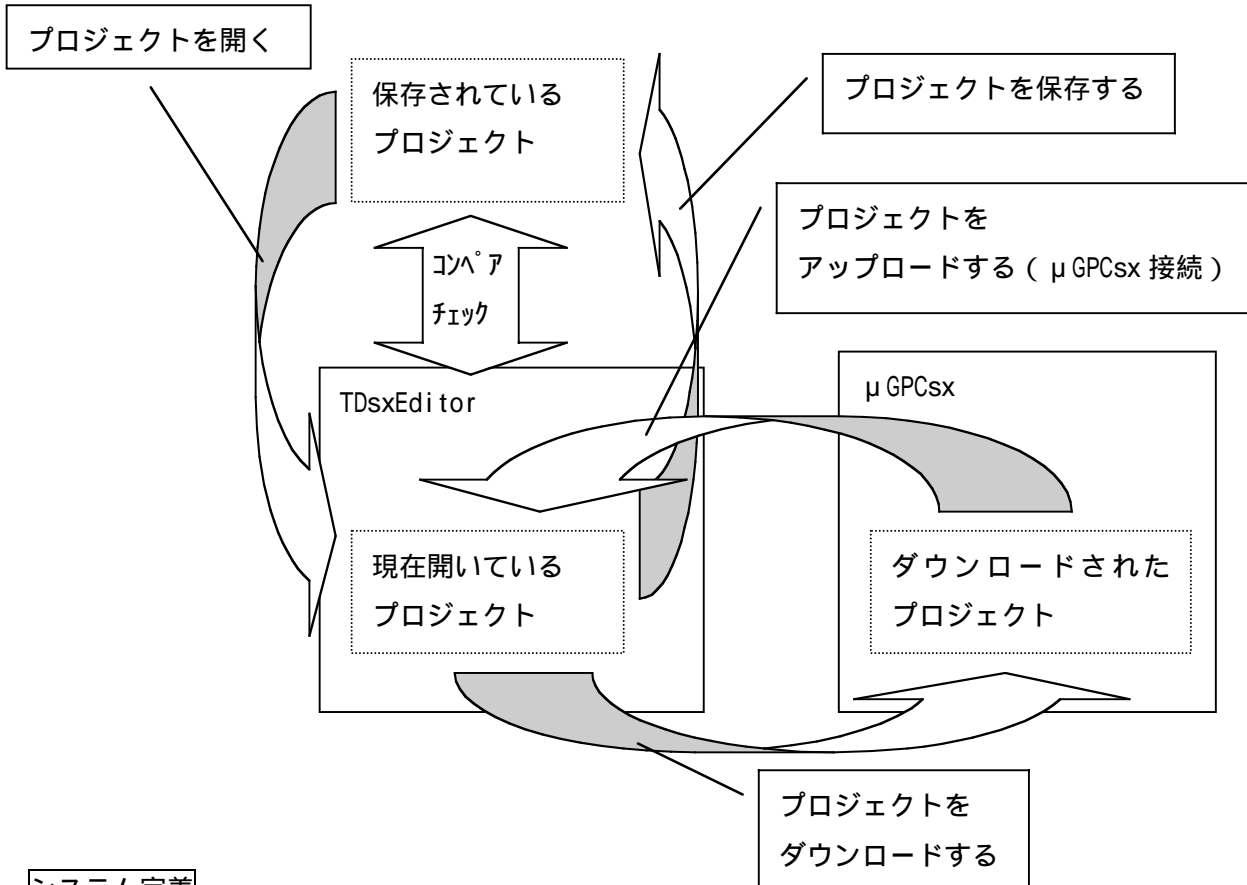
The screenshot shows a project tree window titled 'Sample'. The tree is organized into four main categories, each with a corresponding explanatory text box:

- システム定義 (System Definition):** This category includes 'システム構成定義(I/O割付)', 'システム動作定義', 'CPU動作定義', and '冗長化定義'. The text box explains that this is composed of settings for the μ GPCsx system.
- タスク 1 (Task 1):** This category includes 'スキャンタイム', 'メモリ転送定義', 'トレースバック設定', and subprograms 'SPG1' and 'SPG2'. Each subprogram has sub-items like '回路', 'リレー、レジスタ使用点数', '定数、タイマ、カウンタ', and 'パターンデータ'. The text box explains that tasks are composed of one or more subprograms executed in order, and SPG1 and SPG2 are the names of these subprograms.
- タスク 2 (Task 2):** This category includes 'スキャンタイム', 'メモリ転送定義', 'トレースバック設定', and subprogram 'SPG3'. Similar to Task 1, it has sub-items for '回路', 'リレー、レジスタ使用点数', '定数、タイマ、カウンタ', and 'パターンデータ'. The text box explains that tasks are composed of one or more subprograms, and SPG3 is the name of the subprogram in this case.
- サブルーチン (Subroutine):** This category includes 'SUB-1' with sub-items '回路' and '設定'. The text box explains that subroutines are called by subprograms and are composed of subroutines themselves, with settings including arguments and stack register usage range.

3 - 1 - 2 プロジェクトとは

μGPCsx アプリケーションプログラムはシステム定義とタスクが存在し、1 CPU 内にて存在するこれらを含めてプロジェクトと称します。

TDsxEditor と μGPCsx 内プロジェクトの関係



第3章

システム定義

μGPCsx システムの定義が含まれています。システム定義には「システム構成定義(I/O 割付)」、
「システム動作定義」、「CPU 動作定義」、「冗長化定義」が含まれます。

タスク 1、タスク 2

タスクとは POU のプログラム処理(実行タイムスケジュール)を決定します。タスクは 1 度に 2 つまで実行できます。ただしタスク 1 の方が 2 より優先度は高いです。

タスク 1、タスク 2 には複数のサブプログラムが存在し、それには回路、リレーレジスタ使用点数、定数データ、パターンデータが存在します。タスク 1、タスク 2 はそれぞれ別のスキャン時間を定義し μGPCsx を実行させることができます。

サブルーチン

サブルーチンとはサブプログラムから呼び出せる回路です。タスク 1、タスク 2 に存在する複数のサブプログラムから呼び出すことができます。

3 - 1 - 3 タスク1、タスク2とは

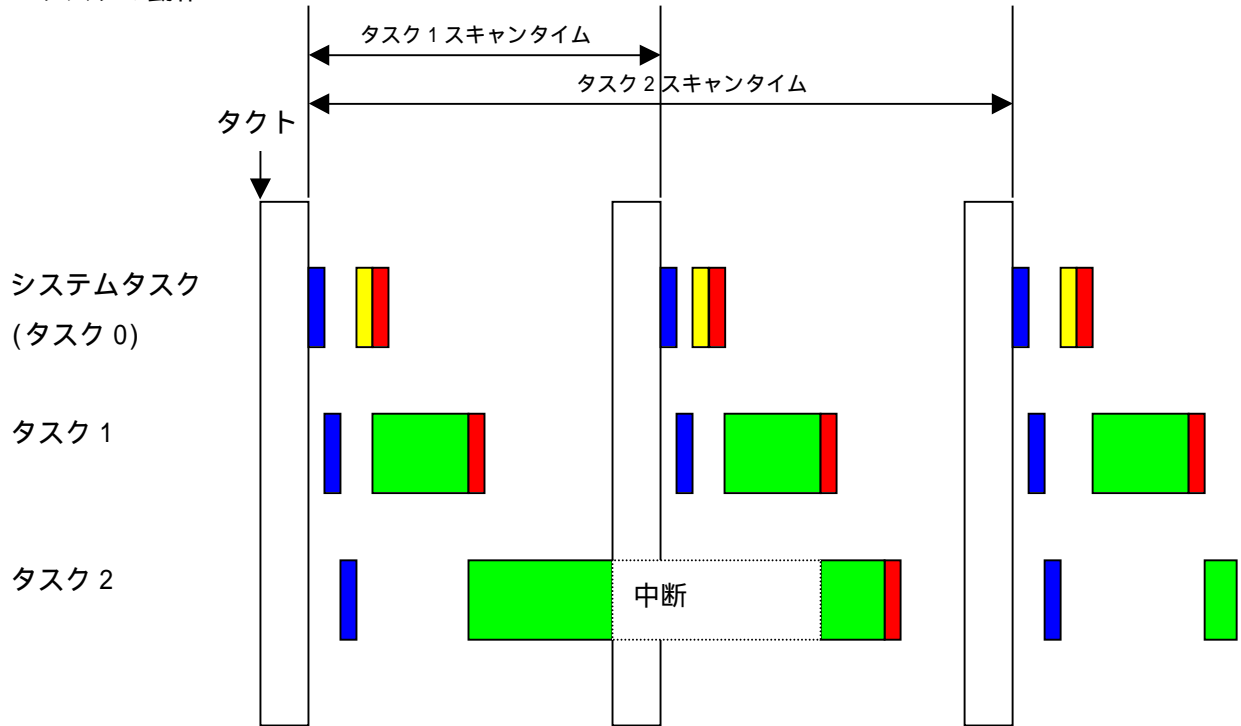
タスクはスキャン時間により、サブプログラムの処理（実行スケジュール）を決定します。

μGPCsx シリーズではタスク1、タスク2の2つのサブプログラム処理を持っています。

1、2はタスクの優先度で1が優先されます。

スキャン時間は SX バスタクト時間(システム定義-システム動作定義の中)で設定した値の整数倍としてください。

< タスクの動作 >

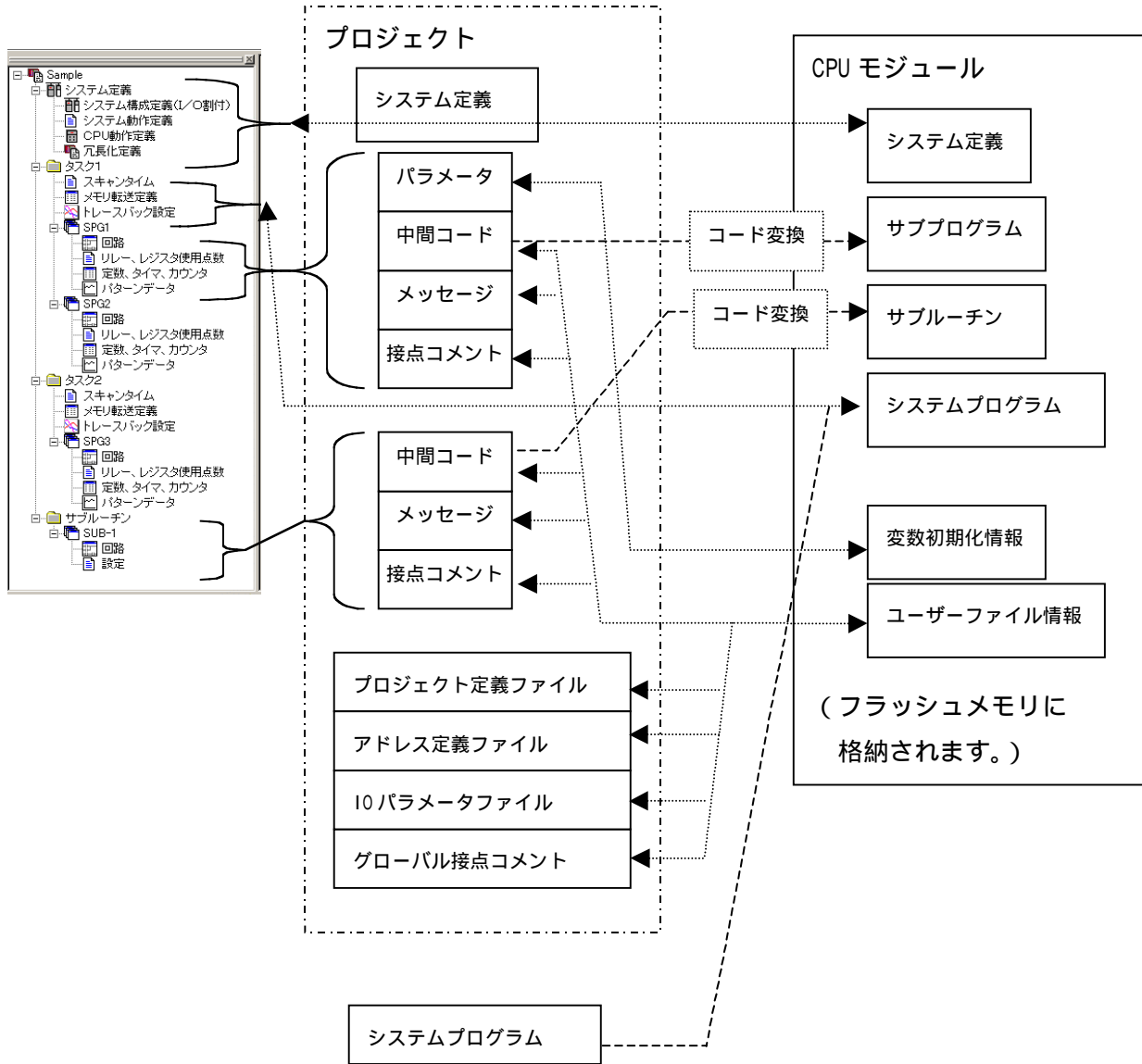


- : 入力リレー、レジスタリフレッシュ
- : 出力リレー、レジスタリフレッシュ
- : システムタスクの演算
- : ユーザープログラムの演算

第3章

3 - 1 - 4 μGPCsx でのプロジェクト

μGPCsx と TDsxEditor プロジェクトの対応



-----▶ : ダウンロードのみ行う処理です。

◀-----▶ : ダウンロード、アップロードを行う処理です。

(フラッシュメモリに格納されます。)

3 - 1 - 5 CPU タイプ

CPU モジュールのタイプにより、データメモリ、プログラムメモリ容量が異なりますので、下記数値を目安にして選定して下さい。

TD1PS-32

サブプログラム数 8本 (1サブプログラムあたりのローカルメモリ 512ワードの時)

サブプログラム総ページ数 200ページ以下

TD1PS-74

サブプログラム数 32本 (1サブプログラムあたりのローカルメモリ 512ワードの時)

サブプログラム総ページ数 400ページ以下

TD1PS-117R

サブプログラム数 64本 (1サブプログラムあたりのローカルメモリ 512ワードの時)

サブプログラム総ページ数 800ページ以下

1サブプログラムあたり 10ページを目安に設計してください。

回路を開いたときに表示される 1画面を 1ページとして換算します。

3 - 2 プロジェクト編集

3 - 2 - 1 プロジェクト関係処理メニュー操作

「ファイル」メニュー

「新規プロジェクト」

新規にプロジェクトを作成します。初回起動時は自動的に新規プロジェクトが作成されます。一旦プロジェクトを編集した後は、次回起動時よりそのプロジェクトが自動的に開かれます。

「プロジェクトを開く/圧縮プロジェクトを開く」

既存のプロジェクトまたは圧縮プロジェクト(「プロジェクトの圧縮保存」で保存されたもの)を開きます。

「プロジェクトを上書き保存」

現在のプロジェクトを保存します。新規プロジェクト時(プロジェクトツリーの最上位項目が「プロジェクト」と表示されている時)は上書き保存時にプロジェクト名称を指定して下さい。

「プロジェクトに名前を付けて保存」

プロジェクトに別の名前を付けて保存します。

「プロジェクトの圧縮保存」

プロジェクト内のファイルを1つのファイルに圧縮して保存します。

「印刷図枠設定」

印刷時の図枠定義を編集します。

「印刷」

プロジェクト内容を印字します。

「プロジェクトの圧縮保存」はFDなどにコピーするときや、バックアップをするときなどに最適です。

「コンペアチェック」

現在開かれているプロジェクトと保存されたプロジェクトとを比較します。オンライン（μGPCsx にダウンロードされているプロジェクトと対象）で「コンペアチェック」を行いたい場合は「オンライン」「μGPCsx 接続」にてアップロードを行った後、「コンペアチェック」を選択します。違いが見つければその部分を画面に表示します。

<コンペアチェックを行う項目>

- ・ システム定義(システム構成定義、システム動作定義、CPU 動作定義)
- ・ スキャンタイム
- ・ メモリ転送定義
- ・ トレースバック設定
- ・ 回路
- ・ リレー、レジスタ使用点数
- ・ 定数、タイマ、カウンタ
- ・ パターンデータ

変更された項目	変更された箇所	変更された値	保存されている値
		コンペアチェック	コンペアチェック[比較先]
システム動作定義	SXバスタク時間	4.0ms	1.0ms
CPU動作定義	バッテリーレス運転設定	する	しない
タスク1	起動周期	10	6
NONAME			
	回路番号1	B00002	B00002
		3: —R—	3: —H—

例えば回路の場合、変更された回路番号、ラベル名、シンボル、行数が表示されます。

3 - 2 - 2 ツリーノードによるポップアップメニュー

サブプログラム名称以外の項目（ツリーノード）が選択された場合。

選択された項目を開きます。

トレースバックデータがサンプリングされている場合、トレースバック表示ウィンドウを開きます。

サブプログラムの新規作成を行います。

サブプログラムのコピーを行います。

選択されている項目のタスクのスキャンタイム設定を開きます。

サブプログラム名称の項目（ツリーノード）が選択された場合。

無効サブプログラムである事を示します。
(オンライン時のみ)

サブプログラム有効無効設定
一時的に動作させないサブプログラムを定義できます。

プログラム有効
プログラム無効

を左クリックすると無効にしたサブプログラムを動作させます。

プログラム有効
プログラム無効

を左クリックするとサブプログラムを無効に（スキップ）します。

選択されたサブプログラムの回路を開きます。

選択されたサブプログラムの各種オンライン機能を実行します。(オンライン時のみ)

サブプログラム編集機能を実行します。

各サブプログラムの各種編集ウィンドウを表示します。

3 - 2 - 3 サブプログラム関係編集操作

「編集」メニュー

「演算順序を上げる」

プロジェクトツリーで選択されたサブプログラムの演算順序を上げます。

「演算順序を下げる」

プロジェクトツリーで選択されたサブプログラムの演算順序を下げます。

「プログラム名称の変更」

サブプログラムの名称を変更します。プロジェクトツリー上でサブプログラム名称を選択し、このコマンドを実行すると、サブプログラム名称が変更可能状態となりますので任意のサブプログラム名称を入力して下さい。

「プログラム新規作成」

サブプログラムを新規に作成します。

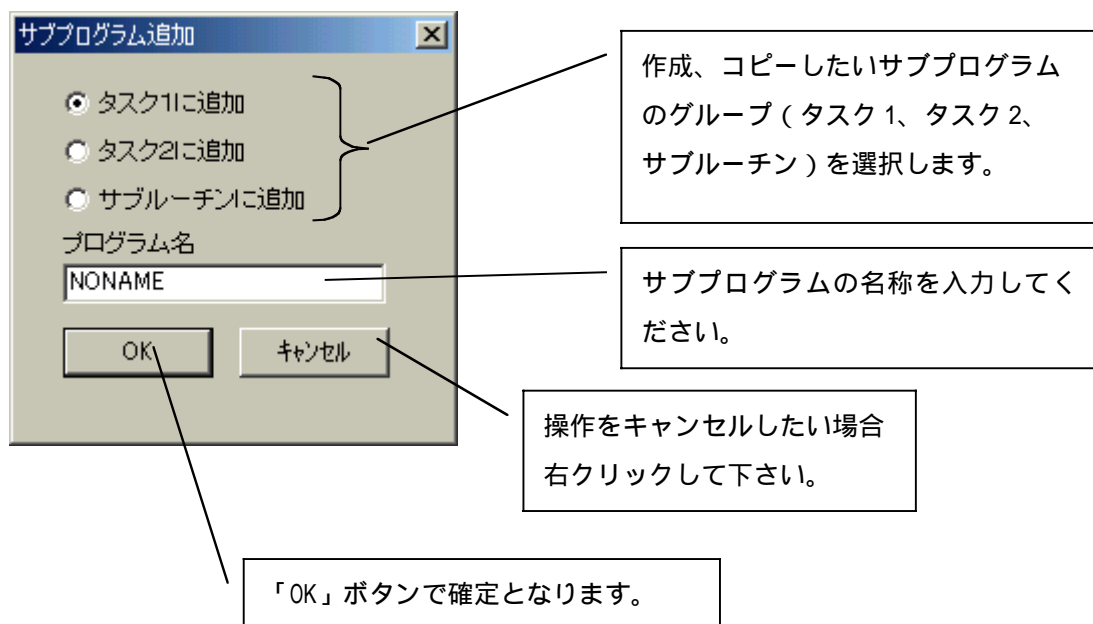
「プログラム削除」

選択されたサブプログラムを削除します。

「プログラムコピー」

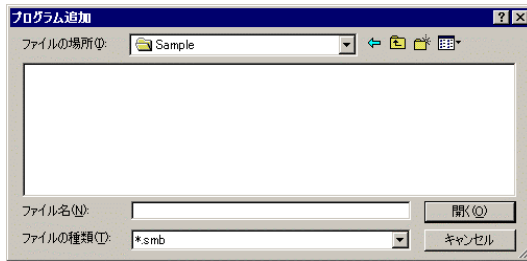
選択されたサブプログラムをコピーし、別のサブプログラムとして生成します。

サブプログラム新規作成、コピー時には下記のようなダイアログボックスが表示されます。



「プログラム追加」

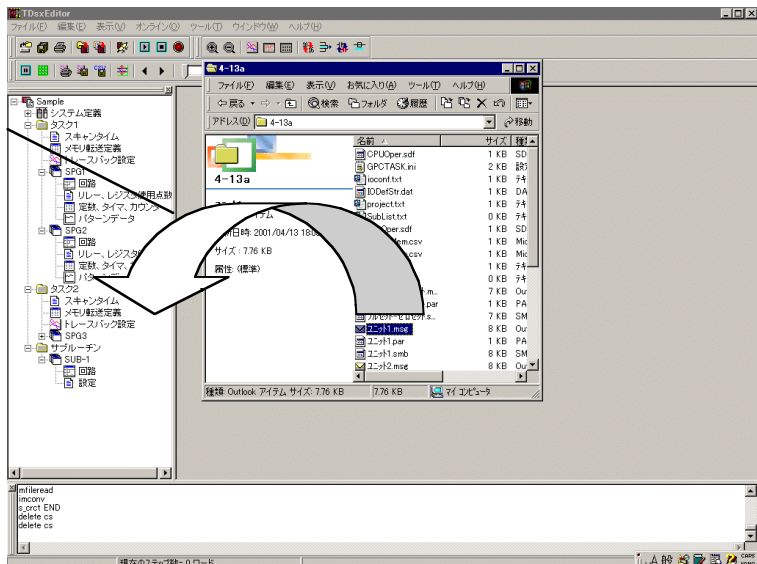
サブプログラムを追加します。プログラム追加ダイアログが表示されますので



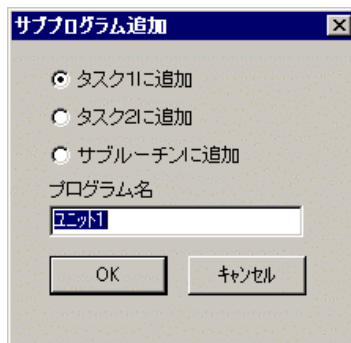
μ GPCH、μ GPCsx で作成されたファイル（拡張子 .par/.smb/.msg/.cmt）を選択します。

また、Windows のエクスプローラからプロジェクトツリーにドラッグドロップすることによってもサブプログラムを追加できます。

.par、.smb、.msg、.cmt
.prm、.sym、.mes、.ccm
いずれかのファイルを
ドラッグドロップします。



サブプログラム追加ダイアログボックスが表示されますので、サブプログラム名を入力して、「OK」ボタンを左クリックしてください。



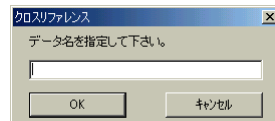
「表示」

「全プログラムクロスリファレンス」

プロジェクト内の全サブプログラム、サブルーチンを対象にクロスリファレンスを検索します。

選択方法は

「G0」、「mi」などのデータ属性のみ指定と「G00000」、「mi00000」などのようにオフセット部を含めた指定ができます。



クロスリファレンス表示例

データ名を示します。

データの存在するサブプログラム名称を示します

クロスリファレンス情報を示します。ウィンドウサイズを広げると、1行あたりのクロスリファレンス情報数も増えます。

B00002	押出機オシレート	0001-10(S)	0001-14(L)		
B00002	庫み計単独	0001-10(S)	0001-14(L)		
B00002	マイクドライブ	0001-04(S)	0001-07(L)		
B00002	マテハン	0010-14(S)	0010-15(L)		
B00002	巻取機共通サーカス	0013-02(S)	0013-05(L)		
B00002	巻取機共通サーカス2	0002-04(L)	0002-05(S)		
B00002	巻取り補機	0001-04(S)	0001-06(L)	0001-07(L)	
		0001-10(L)	0003-08(L)		
B00002	巻取り補機2	0001-04(S)	0001-07(L)		
B00002	A軸巻取機-1	0001-11(L)			

クロスリファレンスウィンドウを閉じます

現在のウィンドウの内容をCSVファイルに保存します。

ダブルクリックすることにより、1行1データ名表示になります。ウィンドウサイズを変更することにより解除されます。

表示例

B00003	マイクドライブ	0001-05(S)	0001-10(L)		
B00003	マテハン	0011-09(L)	0015-02(L)	0015-05(L)	0015-08(S)
B00003	巻取機共通サーカス	0004-03(S)	0004-18(L)		
B00003	巻取機共通サーカス2	0001-03(S)	0002-05(L)	0002-08(L)	0002-09(L)
B00003	巻取り補機	0001-04(L)	0001-07(S)	0001-09(L)	0001-10(L)
B00003	巻取り補機2	0001-08(S)	0001-10(L)		
B00003	A軸巻取機-1	0001-12(L)			
B00003	B軸巻取機-1	0001-12(L)			
B00003	計数取込	0005-19(S)	0006-03(L)		

クロスリファレンス情報

XXXX-YY(Z)

XXXX : 回路番号

YY : 行番号

Z L : ロード (接点) S : ストア (コイル)

I : 関数、サブルーチン引数 (入力)

O : 関数、サブルーチン引数 (出力)

b : 無条件実行サブルーチン

f : 無条件実行関数

B : 条件付き実行サブルーチン

F : 条件付き実行関数

S、I、O の色は 「ツール」 「環境設定」 で変更することができます。

「ツールバー」

各ツールバーの表示非表示を選択します。

標準ツールバー	
共通ツールバー	
ラダーツールバー (回路編集でのみ有効)	
演算ツールバー (回路編集でのみ有効)	
関数 1 ツールバー (回路編集でのみ有効)	
関数 2 ツールバー (回路編集でのみ有効)	
トレンドツールバー (トレンドグラフでのみ有効)	

ボタンの図柄を見て何をするためのツールボタンが分からない場合は、マウスポインタをボタンの上に置いて下さい。しばらくしてマウスポインタの下にツールの名称(コマンド名称)が表示されます。

「オンライン」メニュー

(ダウンロードアップロード関係)

「ダウンロード」

現在開かれているプロジェクトすべてを μ GPCsx にダウンロードします。ダウンロード時には μ GPCsx を停止し、ダウンロード終了時はリセットを行います。

「パラメータ/プログラムダウンロード」

現在開かれているプロジェクトのタスク 1、タスク 2 内のサブプログラムとサブルーチンをダウンロードします。 μ GPCsx 運転中にも可能です。(システム定義、タスク 1 スキャンタイム、タスク 2 スキャンタイムはダウンロードしません。)この時は、変更されたサブプログラム、サブルーチンのみをダウンロードしますのでダウンロード時間の短縮ができます。

「システム定義ダウンロード」

システム定義、スキャンタイムのみをダウンロードします。ダウンロード時には μ GPCsx を停止し、ダウンロード終了時はリセットを行います。

「 μ GPCsx 接続 (アップロード)」

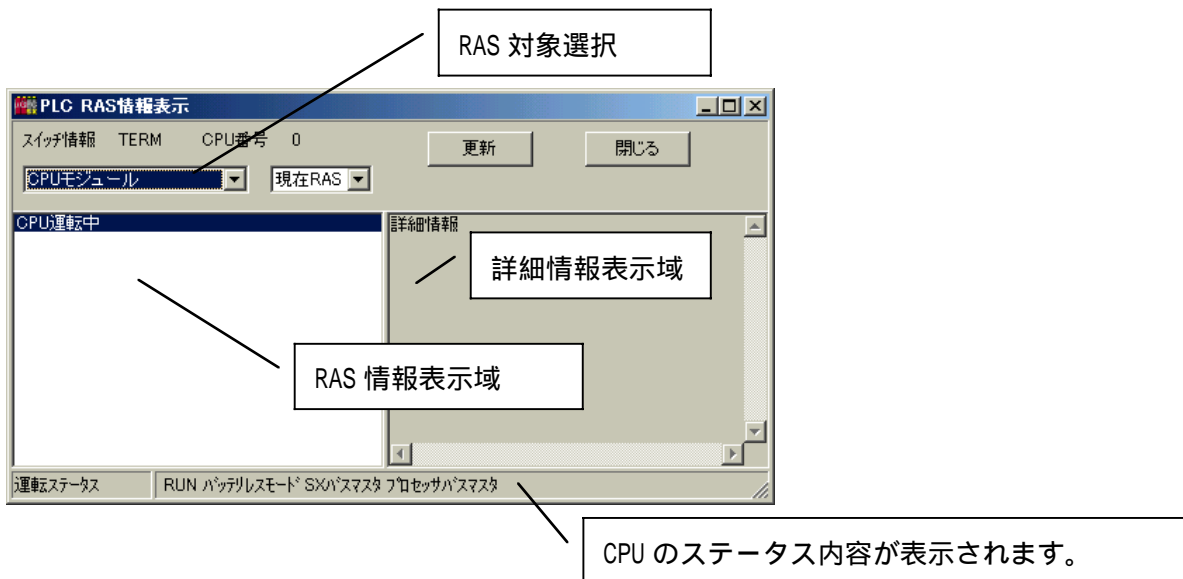
μ GPCsx からプロジェクトをアップロードします。

注)

- ・「パラメータ/プログラムダウンロード」はシステム初期化後の μ GPCsx に対してはできません。
- ・回路モニタ、デバッガ、リレー表示、レジスタ表示、トレンドグラフは上記 3 つの中のどれか 1 つを行わないとできません。
(これらが可能なモードをオンラインモードと呼びます。)

(μ GPCsx 状態表示関係)

「 μ GPCsx RAS 情報表示」



スイッチ情報

CPU モジュールのキー-SW の状態、CPU 番号が表示されます。

RAS 対象選択

RAS 情報を表示する対象を選択します。

更新

最新の RAS 情報に更新します。

閉じる

RAS 情報表示を閉じます。

現在 RAS、一世代前、二世代前、三世代前

過去に発生した来歴のある RAS 情報の場合には、過去の RAS も表示します。

RAS 情報表示域

現在選択されている対象の RAS が表示されます。

詳細情報表示域

詳細項目のある RAS 情報では、詳細情報も表示します。

「μ GPCsx 時計設定」

PLC時計設定

PLC
2001/08/14 17:53:56

パソコン
2001/08/14 17:53:49

設定域
01/08/14 17:53:46

パソコン→PLC 設定域→PLC 閉じる

μ GPCsx の時計を表示します。

パソコンの時計を表示します。

設定域

設定域の値を μ GPCsx に送信します。

パソコンの時計の値を μ GPCsx に送信します。

「リソース情報」

リソース情報

タスクレベル 1

起動周期現在値(μs) 5991

起動周期最大値(μs) 6006

起動周期最小値(μs) 5988

実行時間現在値(μs) 733

実行時間最大値(μs) 737

実行時間最小値(μs) 731

POUメモリ使用量 2465 ワード

閉じる

タスクを指定します。
タスク 1=1
タスク 2=2

タスクの起動周期を
μ s 単位で表示します。

タスクの実行時間を
μ s 単位で表示します。

μ GPCsx 内のプログラム
メモリの使用量を表示します。

.....

「ツール」メニュー

「環境設定」

「ヘルプ」メニュー

プロジェクトツリー

プロジェクトをツリー上に表示させ、ダブルクリックにより各編集項目の編集ウィンドウを表示させます。

ワークスペース

各編集項目の編集ウィンドウが配置されます。

第4章 回路編集

4 - 1	サブプログラム、サブルーチンの回路編集	4 - 1
4 - 1 - 1	回路ウインドウのモード	4 - 1
4 - 2	書き込みモード	4 - 2
4 - 2 - 1	書き込みモードの基本操作	4 - 2
4 - 2 - 2	A 接点の挿入の仕方	4 - 5
4 - 2 - 3	コイルの入力の仕方	4 - 6
4 - 2 - 4	タイマの指定	4 - 7
4 - 2 - 5	データフロー記述の仕方	4 - 8
4 - 2 - 6	定数表現について	4 - 9
4 - 2 - 7	関数シンボル	4 - 10
4 - 2 - 8	ポップアップメニュー	4 - 11
4 - 3	サブルーチンの設計方法	4 - 12
4 - 3 - 1	サブルーチン追加	4 - 12
4 - 3 - 2	サブルーチン編集	4 - 12
4 - 4	回路一覧	4 - 13
4 - 4 - 1	回路一覧での操作	4 - 13
4 - 5	メニューによる操作	4 - 14
4 - 5 - 1	「ファイル」メニュー	4 - 14
4 - 5 - 2	「編集」メニュー	4 - 14
4 - 5 - 3	「表示」	4 - 17
4 - 6	オンラインでの回路	4 - 18
4 - 6 - 1	モニタ	4 - 18
4 - 6 - 2	サブルーチン内のモニタ	4 - 19
4 - 6 - 3	デバッグ	4 - 20



第4章 回路編集

4-1 サブプログラム、サブルーチンの回路編集。

4-1-1 回路ウィンドウの各モード

回路ウィンドウには4つのモードがあります。

- ・読み出しモード

既存回路を開いた時の初期状態のモードです。回路内容の参照のみできます。各モードへ移行できます。

- ・書き込みモード

回路の編集ができるモードです。

- ・モニタ

μGPCsxの回路の演算状態がモニタできます。

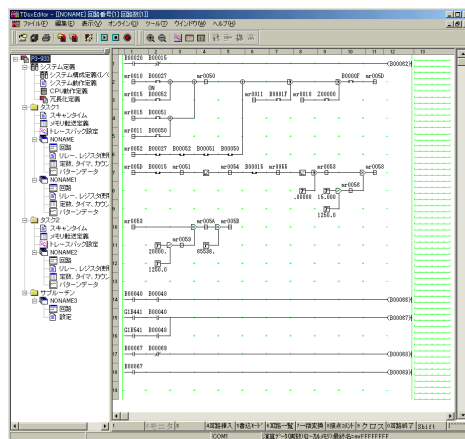
- ・デバッグ

モニタモードに加え、各種デバッグ機能が行えます。

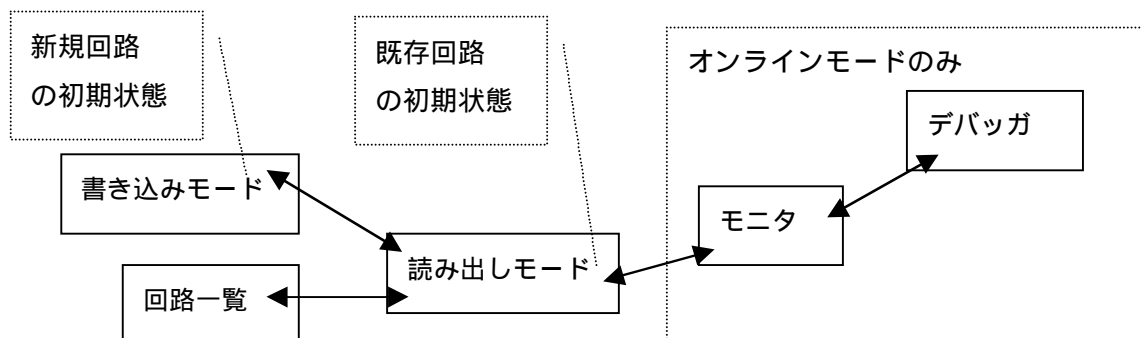
- ・回路一覧

回路をページ単位のスライド形式にて表示します。

読み出しモードウィンドウ



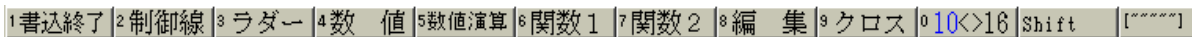
各モードの状態遷移は下図のようになっています。



4 - 2 書き込みモード

4 - 2 - 1 書き込みモードの基本操作

ファンクションキー配列（書き込みモードメインメニュー）



「書込終了」

書き込みモードを終了し、読み出しモードに移行します。

「制御線」「ラダー」「数値」「数値演算」「関数1」「関数2」

下記のような各シンボル挿入メニューに切り替えます。

「制御線」	1メインメニュー 2 ① 3 →① 4 ⊥ 5 ⊥ 6 ⊥ 7 ⊥ 8 ⊥ 9 ⊥ 0 ⊥ Shift [*****]
「ラダー」	1メインメニュー 2 ⊥ 3 ⊥ 4 () 5 ⊥ 6 ⊥ 7 ⊥ 8 ⊥ 9 ⊥ 0 ⊥ Shift [*****]
Shift	1メインメニュー 2 ① 3 →① 4 ⊥ 5 ⊥ 6 ⊥ 7 ⊥ 8 ⊥ 9 ⊥ 0 ⊥ Shift [*****]
「数値」	1メインメニュー 2 ⊥ 3 ⊥ 4 ⊥ 5 ⊥ 6 ⊥ 7 ⊥ 8 ⊥ 9 ⊥ 0 ⊥ Shift [*****]
Shift	1メインメニュー 2 ① 3 →① 4 ⊥ 5 ⊥ 6 ⊥ 7 ⊥ 8 ⊥ 9 ⊥ 0 ⊥ Shift [*****]
「数値演算」	1メインメニュー 2 ⊕ 3 ⊕ 4 ⊗ 5 ⊗ 6 ⊗ 7 ⊗ 8 ⊗ 9 ⊗ 0 ⊥ Shift [*****]
Shift	1メインメニュー 2 ⊕ 3 ⊕ 4 ⊗ 5 ⊗ 6 ⊗ 7 ⊗ 8 ⊗ 9 ⊗ 0 ⊥ Shift [*****]
「関数1」	1メインメニュー 2 ⊥ 3 ⊥ 4 ⊥ 5 ⊥ 6 ⊥ 7 ⊥ 8 ⊥ 9 ⊥ 0 ⊥ Shift [*****]
Shift	1メインメニュー 2 ⊥ 3 ⊥ 4 ⊥ 5 ⊥ 6 ⊥ 7 ⊥ 8 ⊥ 9 ⊥ 0 ⊥ Shift [*****]
「関数2」	1メインメニュー 2 ⊥ 3 ⊥ 4 ⊥ 5 ⊥ 6 ⊥ 7 ⊥ 8 ⊥ 9 ⊥ 0 ⊥ Shift [*****]
Shift	1メインメニュー 2 ① 3 →① 4 ⊥ 5 ⊥ 6 ⊥ 7 ⊥ 8 ⊥ 9 ⊥ 0 ⊥ Shift [*****]

「Shift」

各ファンクションメニューを切り替えます。



ファンクションメニュー位置をウィンドウ上部/下部へと移動できます。

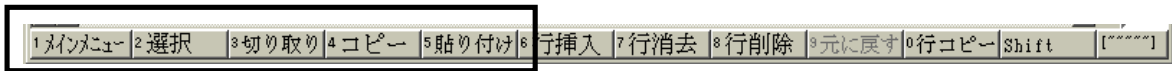
現在下部です。上部に移動します。

現在上部です。下部に移動します。

この設定は次回起動時にも反映されます。

「編集」

編集メニューに切り替えます。



「メインメニュー」 書き込みモードメインメニューに戻ります。

「選択」 カーソル位置を、切り取り、コピー範囲の始点とします。

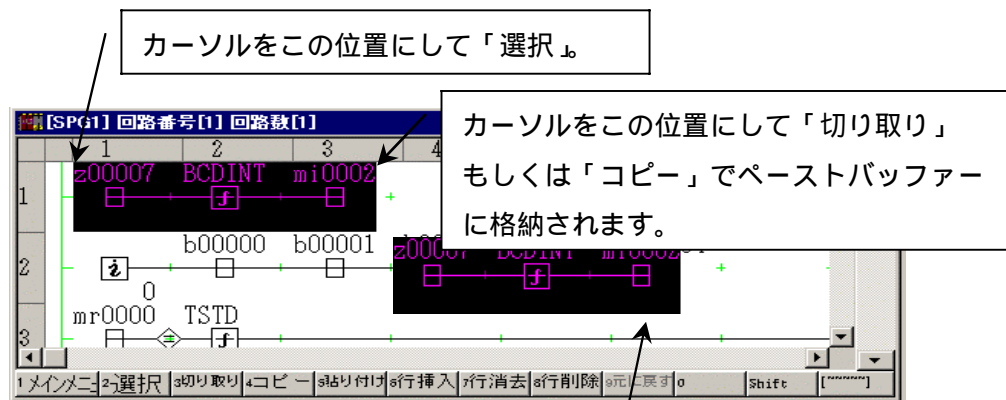
「切り取り」 ボックスで囲まれた範囲を切り取ります。

「コピー」 ボックスで囲まれた範囲をコピーします。

「貼り付け」 切り取り、コピーした内容を貼り付けます。

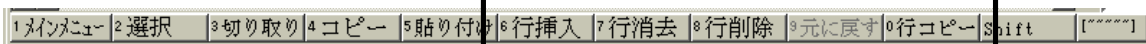
「キャンセル」 カーソル位置へのボックス描画がキャンセルされます。（「選択」操作のキャンセル）

ボックス選択したウィンドウ例



「切り取り」「コピー」後は、カーソル位置に表示されますので、貼り付けたい位置にて「貼り付け」を行ってください。
一度「貼り付け」を行うとカーソル位置にペーストバッファの内容は表示されなくなりますが、「貼り付け」にてペーストバッファの内容は貼り付けされます。

尚、「選択」「切り取り」「コピー」「貼り付け」はメニューバー、右クリックのポップアップメニューからも実行できます。



- 「行挿入」 カーソル位置に1行挿入します。
- 「行消去」 カーソル位置の行を消去します。
- 「行削除」 カーソル位置の行を削除し、1行上に詰めます。
- 「元に戻す」 「行消去」、「行削除」の操作を1回だけキャンセルできます。
- 「行コピー」 カーソル位置の行をカーソル行より下方向にある1番最初のスペース行にコピーします。

「クロス」

クロスリファレンスを表示します。

[b0][PG1]クロスリファレンス[5]	
b00000	0001-02(S)
b00001	0001-02(S)
b00002	0001-02(S)
b00003	0001-02(S)
b00004	0001-02(S)

読み出しモード、モニタ、デバッガ時
クロスリファレンス情報をクリックすること
により該当箇所へジャンプします。

OK CSVファイル保存

CSV ファイル
に保存します。

クロスリファレンス
ウィンドウを閉じます。

ダブルクリックすることにより、
1行1データ名表示になります。
ウィンドウサイズを変更すること
により解除されます。

表示例

[A1][オシロスコープ]クロスリファレンス[386]					
B00010	0002-04(L)	0002-12(L)	0003-04(L)	0004-04(L)	0004-12(L)
B00011	0001-05(S)	0001-06(L)	0002-04(L)	0002-12(L)	0003-04(L)
B00015	0012-15(S)	0012-15(L)	0012-16(L)		
B00016	0012-17(S)	0012-18(L)			
B00018	0008-03(S)	0008-08(L)			
B00019	0008-06(S)	0008-08(L)			

OK CSVファイル保存

「10<>16」

回路内で使用されている整数データの数値表示の10進16進を切り替えます。

現在のモードは「10」、「16」どちらかの数字が青で表示されます。

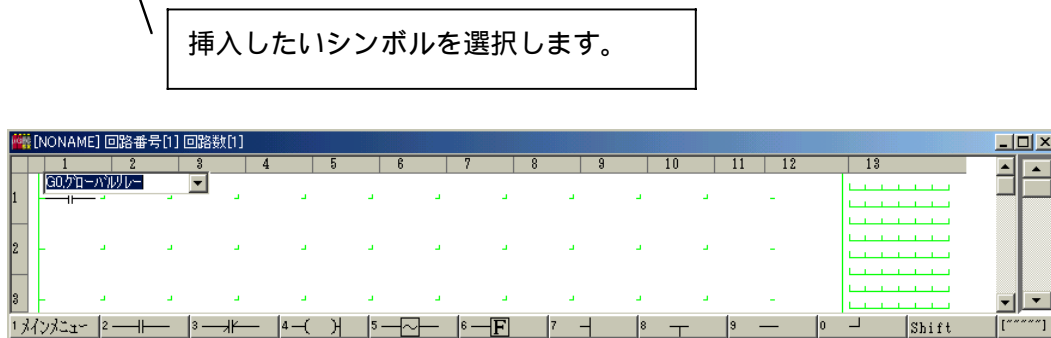
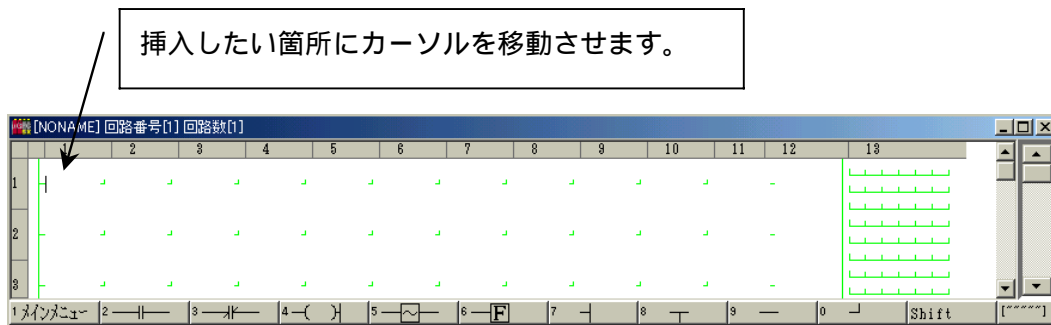
10進表示時

010<>16

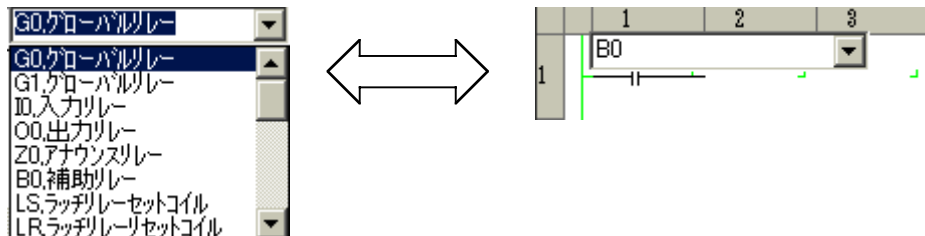
16進表示時

010<>16

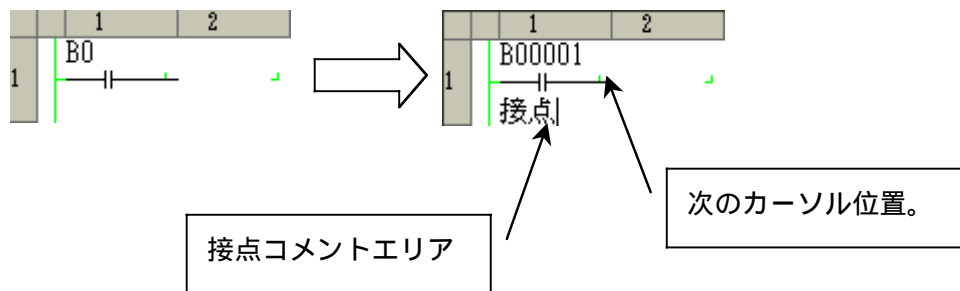
4 - 2 - 2 A接点の挿入の仕方



リストボックスによりリレー名を選択します。またリストボックスに直接入力することもできます。

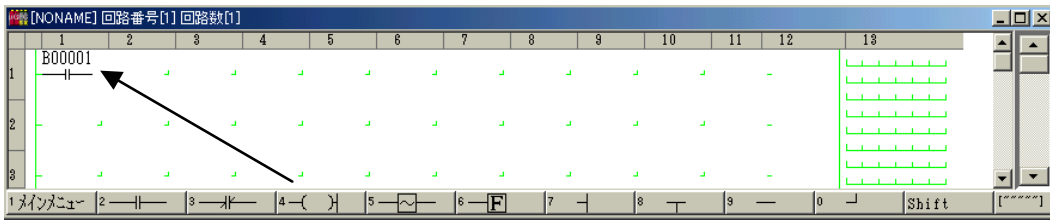


そしてリレー番号を直接入力します。必要に応じて接点コメントを入力し[Enter]キーで次の入力位置へカーソルが移動します。

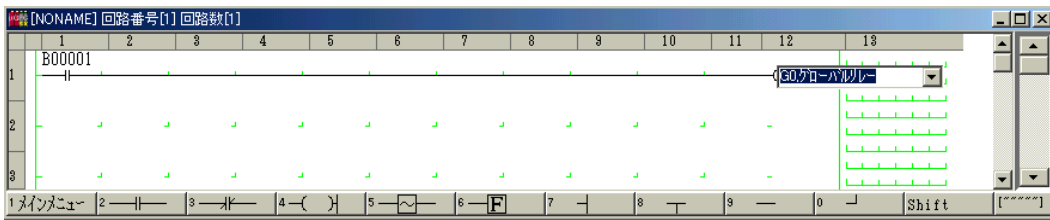


4 - 2 - 3 コイルの入力の仕方

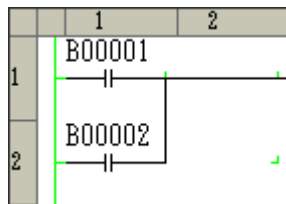
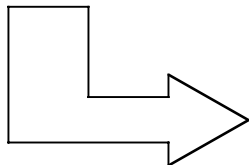
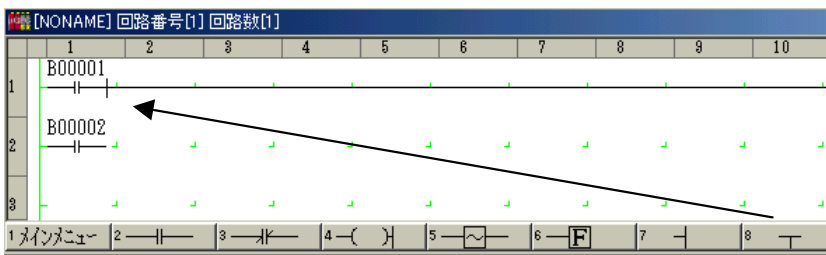
接点シンボルの後ろにカーソルを持っていき、



コイルシンボル選択すると下記のようなラダーとなります。



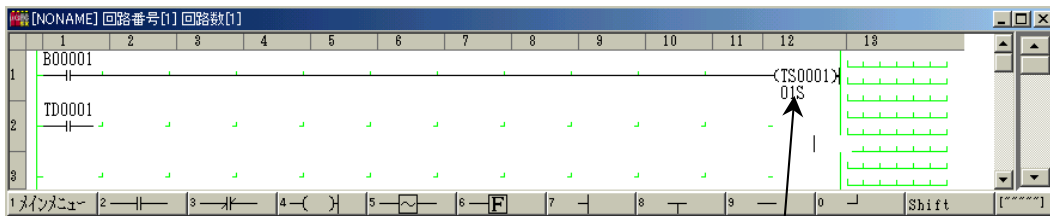
アンド回路入力例 下記状態にて 制御線シンボルを挿入すると



不足分は自動的に補完します。

タイマーコイルの設定例

コイルの下にタイマ、カウンタ値を入力します。



タイマ、カウンタ値入力エリア 01S = 1秒

4 - 2 - 4 タイマの指定



タイマ名を入力し、その下の行にタイマ値を入力します。

タイマ値がゼロの場合、00.00S が表示されます。

タイマ値が未入力の場合は、現在の設定値に変化はありません。

タイマ値入力形式

00H00M ... 60進 H:時

00M00S ... 60進 M:分

00.00S ... 10進 S:秒

タイマ値の設定は定数、タイマ、カウンタウィンドウでも行なうことができます。

同じタイマ名を使用し、異なった値を指定した時は、ラインの大きい方又は、回路が異なる時は、新しく編集した回路の値が有効となります。

カウンタの指定



カウンタ名を入力し、その下の行にカウンタ値を入力します。

カウンタ値がゼロの場合、000000 が表示されます。

カウンタ値が未入力の場合は、現在の設定値に変化はありません。

カウンタ値設定範囲

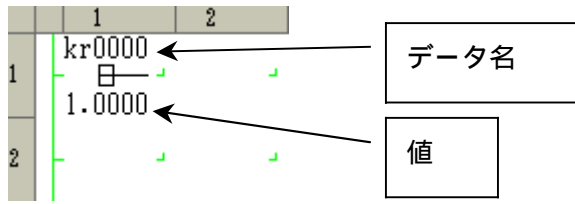
0 ~ 65535

カウンタ値の設定はパラメータウィンドウでも行なうことができます。

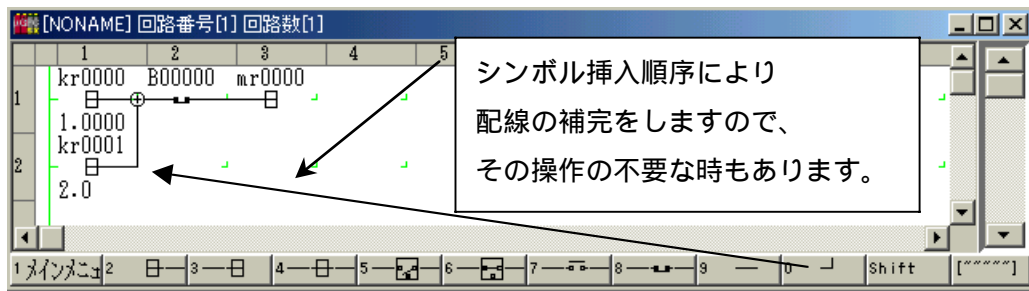
同じカウンタ名を使用し、異なった値を指定した時は、ラインの大きい方又は、回路が異なる時は、新しく編集した回路の値が有効となります。

4 - 2 - 5 データフロー記述の仕方

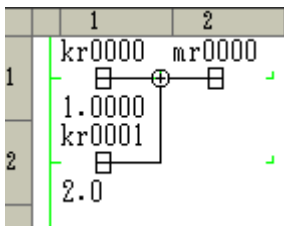
定数データはロード命令を挿入し、下部に値を入力します。



加算シンボルはロード命令の右のクロスポイント (+) に挿入します。

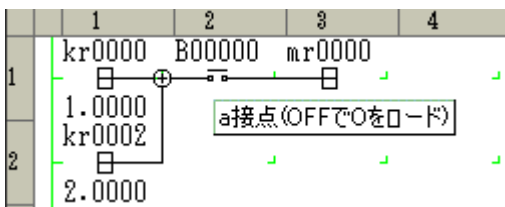


下にロード命令を挿入した後、結線します。



データフローは必ずストア命令で終端します。

データ名の付随するシンボル入力例はクロスポイントの間に挿入します。



4 - 2 - 6 定数表現について

数値は、シンボルの1行下に入力します。

定数値がゼロの時、

整数型 (kiXXXX) : 000000

実数型 (krXXXX) : .00000

と表示されます。

定数入力例

整数	123	(10進)	80H	(16進)
	-123	(10進)	8005H	(16進)
実数	123.4	.12345		
	-123.4	-.2345		

1未満の実数では1の桁を省略します。

定数値の設定はパラメータウィンドウでも行なうことができます。

同じ定数名の定数に異なった数値を入力した場合は、ラインの大きい方又は、回路が異なる時は、新しく編集した回路の値が有効となります。

4 - 2 - 7 関数シンボル

関数シンボルはシンボル挿入後、引数設定ウィンドウが表示されます。

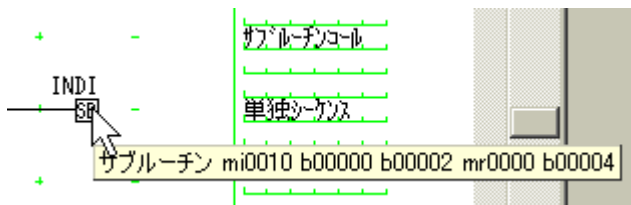


引数設定ウィンドウを閉じた後は、関数シンボルをダブルクリックすることにより表示されます。

各関数のパラメータについてはプログラミングマニュアルを参照してください。

シンボルは右クリックによっても入力できます。

関数シンボルやサブルーチンシンボルの上にマウスポインタを置くと、その関数や、サブルーチンのパラメータが表示されます。



4 - 2 - 8 ポップアップメニュー

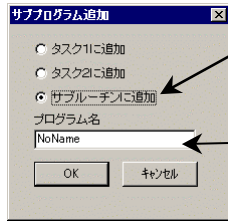
ポップアップメニューでは各シンボル挿入機能、編集機能がマウスの右クリックによりダイレクトに実行できます。種類は下記となります。

メイン	制御命令	論理	数値	数値演算	数値関数
<ul style="list-style-type: none"> 配線(M) 選択(N) 切り取り(O) コピー(C) 貼り付け(P) キャンセル(R) 行挿入(S) 行消去(D) 行削除(U) 制御命令(V) ▶ 論理(W) ▶ 数値(X) ▶ 数値演算(Y) ▶ 関数(Z) ▶ 	<ul style="list-style-type: none"> ①▶ 結合子ロード *① 結合子ストア L ラベル命令 └ 終端線 	<ul style="list-style-type: none"> ≡ A接点 ≡ B接点 ◁ 論理反転 └ アンド線上 └ アンド線下 ○ コイル F 条件付きシステム関数 SB 条件付きサブルーチン └ 終端線 	<ul style="list-style-type: none"> □ ロード □ ストア/ロード □ ストア → a接点 → b接点 ◻ c接点(aタイプ) ◻ c接点(bタイプ) ◇ コンペアハイ ◇ コンペアロウ ◇ コンペアイコール Ⓜ 局所定数:整数 Ⓜ 局所定数:実数 └ 終端線 	<ul style="list-style-type: none"> ⊕ 加算 ⊖ 減算 ⊗ 乗算 ⊘ 除算 ⊙ 剰余 ⊞ 上位優先(下限リミット) ⊟ 下位優先(上限リミット) Ⓜ 数値積 Ⓜ 数値和 Ⓜ 数値排他和 └ 終端線 	<ul style="list-style-type: none"> △ 負数変換 1' 補数 ▭ 絶対値変換 ⊕ 1/2カット ⊖ 3/4カット 2分の1 2倍 2乗 指数関数 2乗根 ビットカウント グレイコード/バイナリ変換 不感帯 パルス 微分補償 位相補償 PID補償 ARC S-ARC 算術平均 フィルタ PID補償 一時遅れ(移動平均) 遅れ(時間遅れ) 定周期パルス 変数設定パルス 上下限リミット ヒステリシス システム関数 サブルーチン

4 - 3 サブルーチンの設計方法

4 - 3 - 1 サブルーチン追加

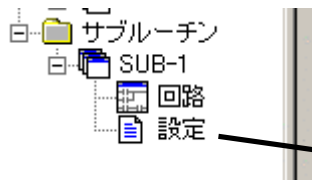
プロジェクトツリーにてプログラム新規作成を行ないサブルーチンを追加します。



「サブルーチンに追加」を選択します。

サブルーチン名を指定します。

サブルーチンの設定を行います。



スタックレジスタの使用数を定義します。最大使用可能数は最終名に表示されます。



パラメータ数を設定します。出力点数を決定すると、自動的に入力点数も決定されます。

あらかじめ作成されたサブルーチンで使用数がわからない場合使用数を検索します。

4 - 3 - 2 サブルーチン編集

サブルーチンとのデータのやり取りは引数を使用して行います。

引数とは呼び出し元の回路からサブルーチンプログラムに渡すパラメータ、または受け取る演算結果のことを指します。

前ページのサブルーチン設定画面において設定されたパラメータの数(入力・出力点数)が引数設定画面に反映、入力と出力の部分が色で分けられます。

ここで入力にはサブルーチンに渡すラベル名を、出力にはサブルーチンから受け取るラベル名を左側に入力します。

引数には数値レジスタ名以外にリレーシンボル名も設定することが出来ます。

入力 → SI/si/sr 0008



入力 → SIO080 B00000

ラベル名を入力すると、引数の型が自動的に選択されます。

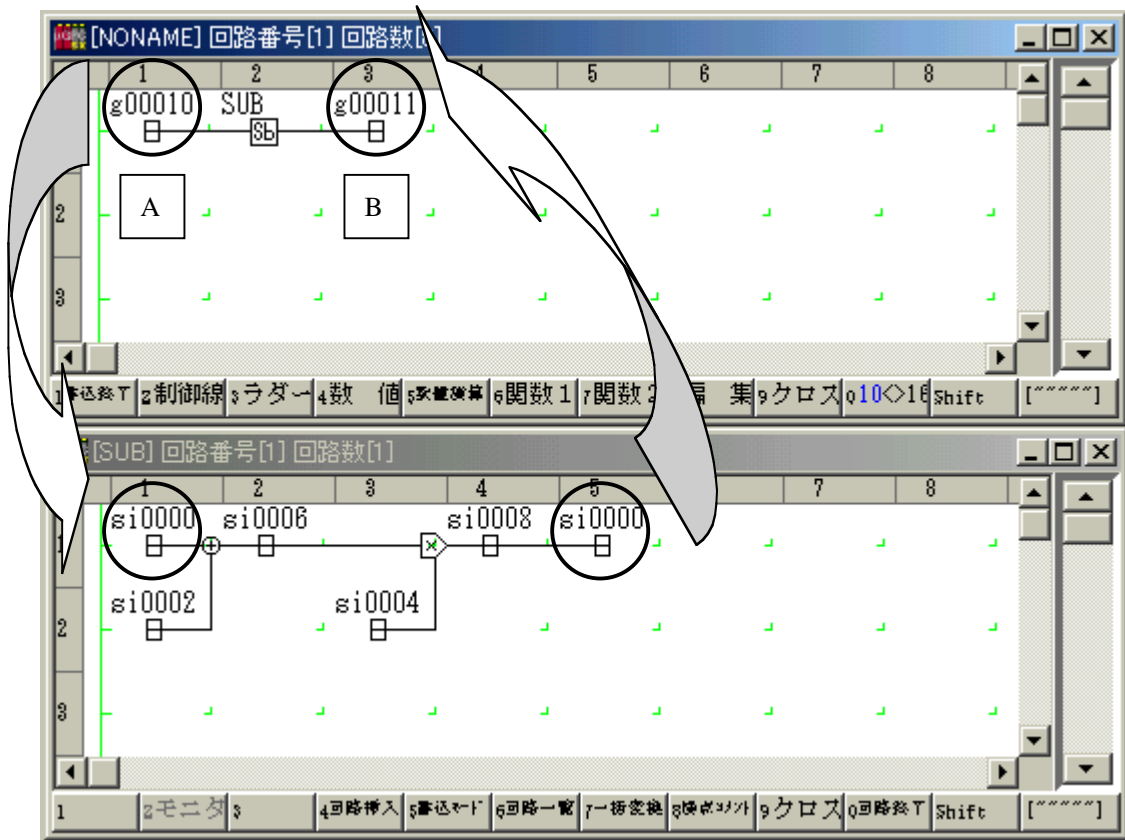
f サブルーチン		
入力 → si0002	ki0000	10
入力 → si0004	mi0001	
出力 si0006 →	mi0010	
出力 si0008 →	g00000	

OK キャンセル 適用

定数 (ki、kr) を使用する場合は左側にラベル名、右側に値を入力します。

入力と出力で色分けされます。

引数定義例



入力 → si0002	ki0000	10
入力 → si0004	mi0001	
出力 si0006 →	mi0010	
出力 si0008 →	g00000	

OK キャンセル 適用

上図のサブルーチンのようにスタックレジスタに si0000(sr0000、SI0000 も同様)を使用すると、矢印のようにデータが渡され、呼び出し元の “A” が入力、“B” が出力となります。

上図のサブルーチンの流れ

si0000 には呼び出し元の g00010 の値がロードされ、si0002 には引数で設定した ki0000(=10) の値がロードされます。サブルーチン内で両者は加算されます。

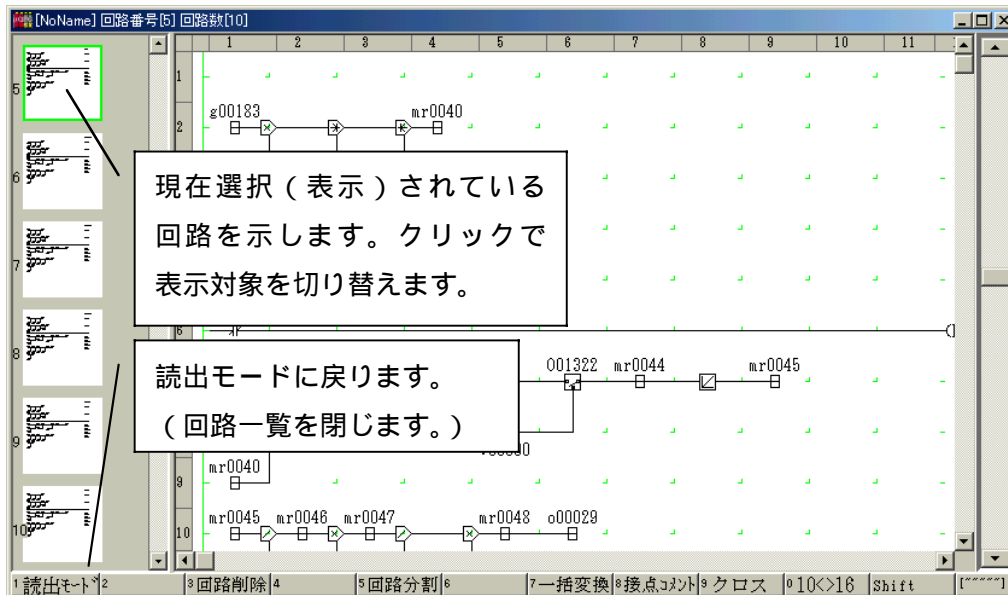
si0006 には で計算された値がストアされ、引数で設定した mi0010 にストアされます。引数で設定した mi0001 の値は si0004 にロードされ、 の結果に乗算されます。



si0008 には で計算された値がストアされ、引数で設定した g00000 にストアされます。最後に の値が si0000 にロードされ、呼び出し元の g00011 にストアされます。

4 - 4 回路一覧

4 - 4 - 1 回路一覧での操作

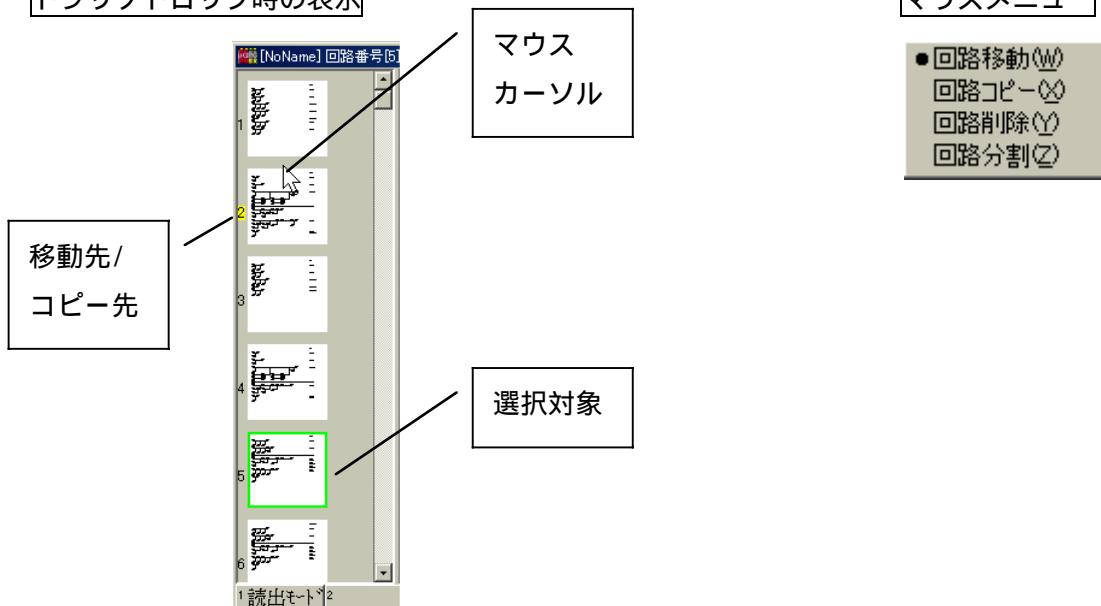
回路一覧はページ単位の回路の削除、移動、コピーを行う時に使用します。



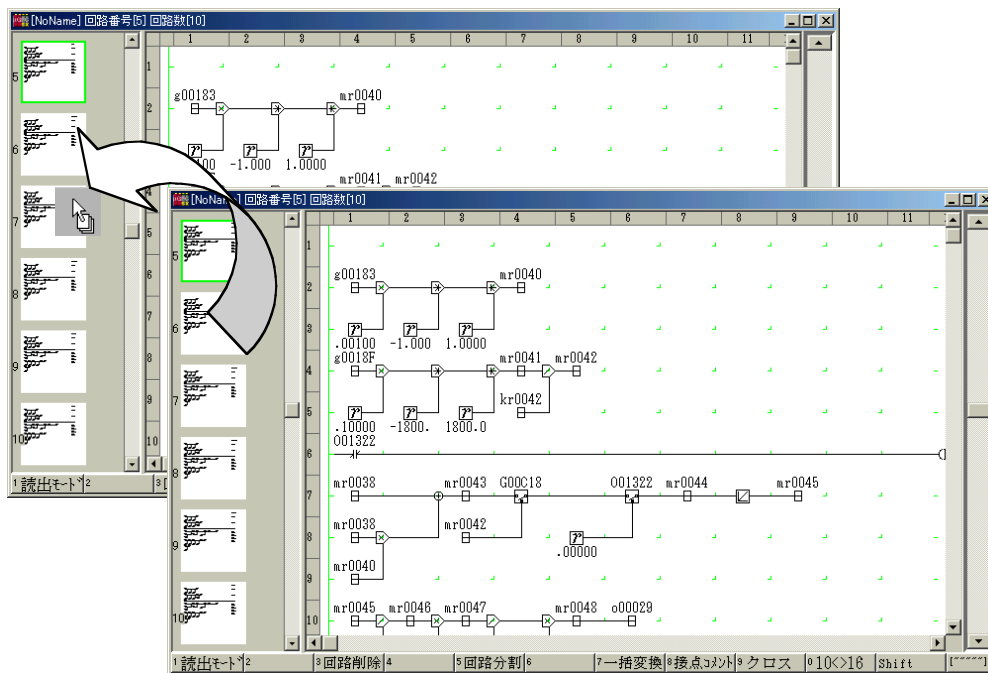
- 回路移動 ドラッグドロップにて回路を移動できます。
ドラッグすると右のようなマウスポインタが表示されます。 
- 回路コピー ドラッグドロップにて回路をコピーできます。
ドラッグすると右のようなマウスポインタが表示されます。 
- 回路削除 選択されている回路を削除します。
[Delete]キーでも回路を削除することができます。
- 回路分割 選択されている回路から新しいサブプログラムとして分割します。

ドラッグドロップ時の表示

マウスメニュー

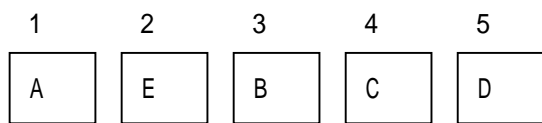
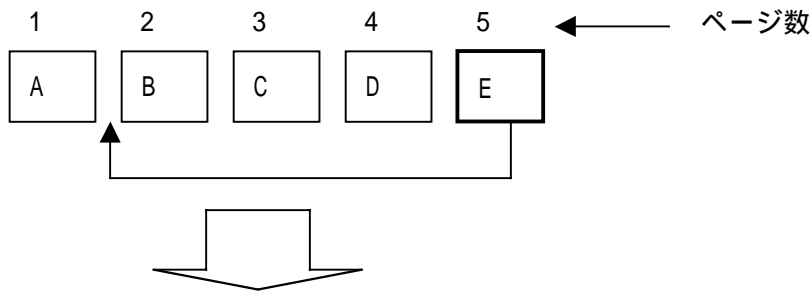


別のサブプログラムからのコピーもできます。



回路コピーにしたのち、別のサブプログラムからドラッグドロップにて回路をコピーできます。

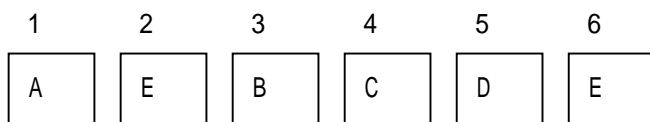
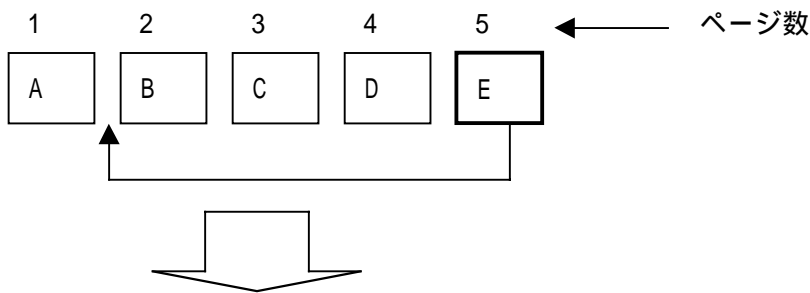
・移動時(5 ページ目にある “E” を 2 ページ目に移動)



移動とは

1 つのサブプログラムの中で複数ページがある場合に選択したページをドラッグドロップにより指定したページに移すことです。上図では 5 ページ目にある “E” を 2 ページ目に移動することにより、“A” “B” “C” “D” “E” という順番だったサブプログラムが、“A” “E” “B” “C” “D” という順番になります。

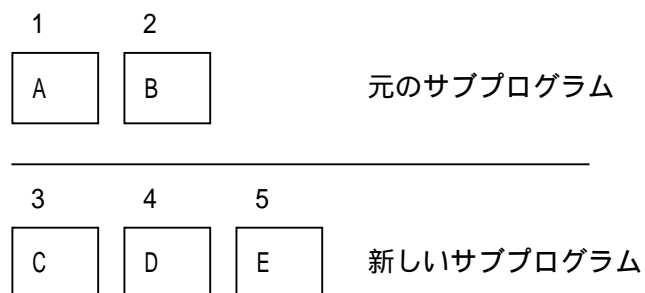
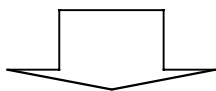
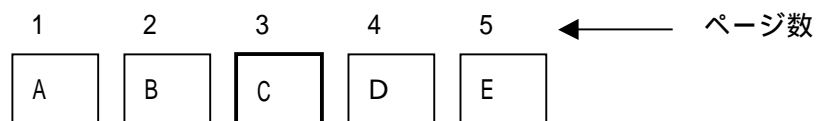
・コピー時(5 ページ目にある “E” を 2 ページ目にコピーし挿入)



コピーとは

1 つのサブプログラムの中で選択したページを、ドラッグドロップにより指定したページにコピーし挿入することです。上図では 5 ページ目にある “E” を 2 ページ目にコピーし挿入することにより “A” “B” “C” “D” “E” という順番だったサブプログラムが、“A” “E” “B” “C” “D” “E” という順番になります。移動と違ってコピーしたページはそのまま残りページ数はコピーし挿入したページから 1 ページずれます。

・分割時(3ページ目の“C”から分割)



分割とは

1つのサブプログラムの中で複数ページがある場合に、選択したページから残り全部を切り取り新たなサブプログラムに追加することです。上図では3ページ目の“C”を選択した場合、元のプログラムは“A”“B”の2ページに、新たに作られたプログラムは“C”“D”“E”の3ページになります。

4 - 5 メニューによる操作

4 - 5 - 1 「ファイル」メニュー

各モードへ移行します。現在のモードにはチェックマークが付きます。
 移行できないモードは濃淡表示となります。

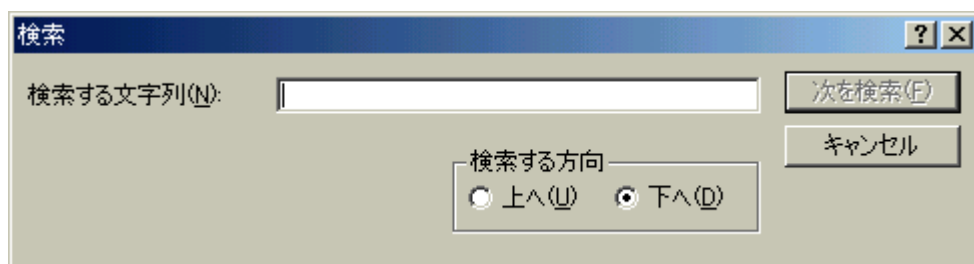
<input checked="" type="checkbox"/> 読出モード(R) Ctrl+R	<p>現在のモードを示します。</p> <p>回路ウィンドウを閉じます。</p>
<input type="checkbox"/> 書込モード(W) Ctrl+E	
<input type="checkbox"/> モニタ(M) Ctrl+M	
<input type="checkbox"/> デバッガ(D) Ctrl+D	
<input type="checkbox"/> 回路一覧(L)	
<input type="checkbox"/> 回路を閉じる(O)	

4 - 5 - 2 「編集」メニュー

選択	<p>回路内をブロック単位で選択し、 切り取り、コピー、貼り付けを行います。</p>
切り取り Ctrl+X	
コピー Ctrl+C	
貼り付け Ctrl+V	
キャンセル	
行挿入	カーソル行に1行挿入します。
行消去	カーソル行を消去します。
行削除	カーソル行に削除し1行ずつ詰めます。
検索	
置換	
接点コメント	書込みモードでのみ有効です。

「検索」

指定されたデータ名を検索します。



検索する文字列 データ名を入力して下さい。

検索する方法 上へ：回路番号 - 1・・・ にて検索します。

下へ：回路番号 + 1・・・ にて検索します。

「置換」

指定されたデータ名を置換します

**閉じる**

一括変換ウィンドウを閉じます。

自動割付

ローカルメモリのアドレスを自動的に割付ます。

変換実行

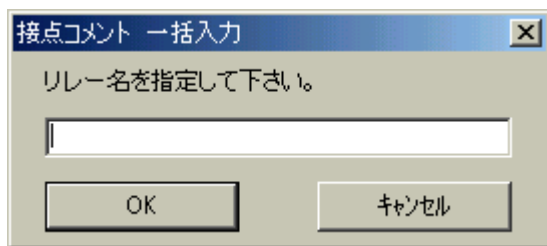
置換後の文字列 変換後のデータ名を入力して下さい。

変換回路範囲指定

変換を実行する回路の範囲が指定できます。

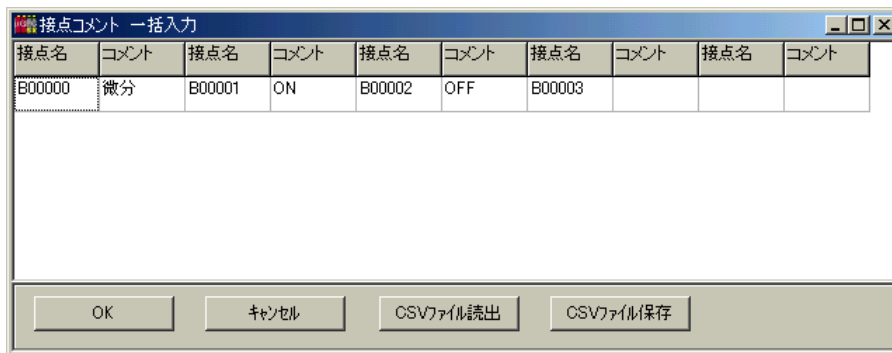
「接点コメント」

接点コメント設定ウィンドウを表示します。



リレー名 2 文字（例：G0、B0 など）を指定して下さい。

サブプログラム内で使用されている接点コメントを検索し、一覧表示します。



OK

表示されている内容を接点コメントに反映しウィンドウを閉じます。

キャンセル

表示されている内容を接点コメントにせずウィンドウを閉じます。

CSV ファイル読出

CSV ファイルに保存されている接点コメントを読み出します。

CSV ファイル保存

ウィンドウの内容を CSV ファイルに保存します。

4 - 5 - 3 「表示」

ページ変更

表示倍率

クロスリファレンス

全プログラムクロスリファレンス

ツールバー(T)

回路の表示倍率を変更します。

5%

17%

25%

40%

50%

70%

75%

80%

90%

100%

110%

120%

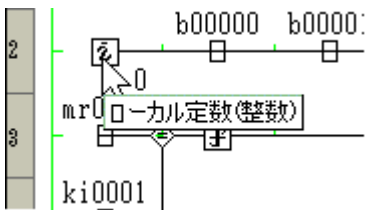
可能表示倍率
現在表示されて
いる倍率にチェ
ックマークが付
きます。

「クロスリファレンス」
回路内みのクロスリファレンス情報を検索します。

「全プログラムクロスリファレンス」
プロジェクト内の全サブプログラム、サブルーチンを対象に
クロスリファレンスを検索します。

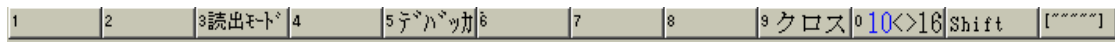
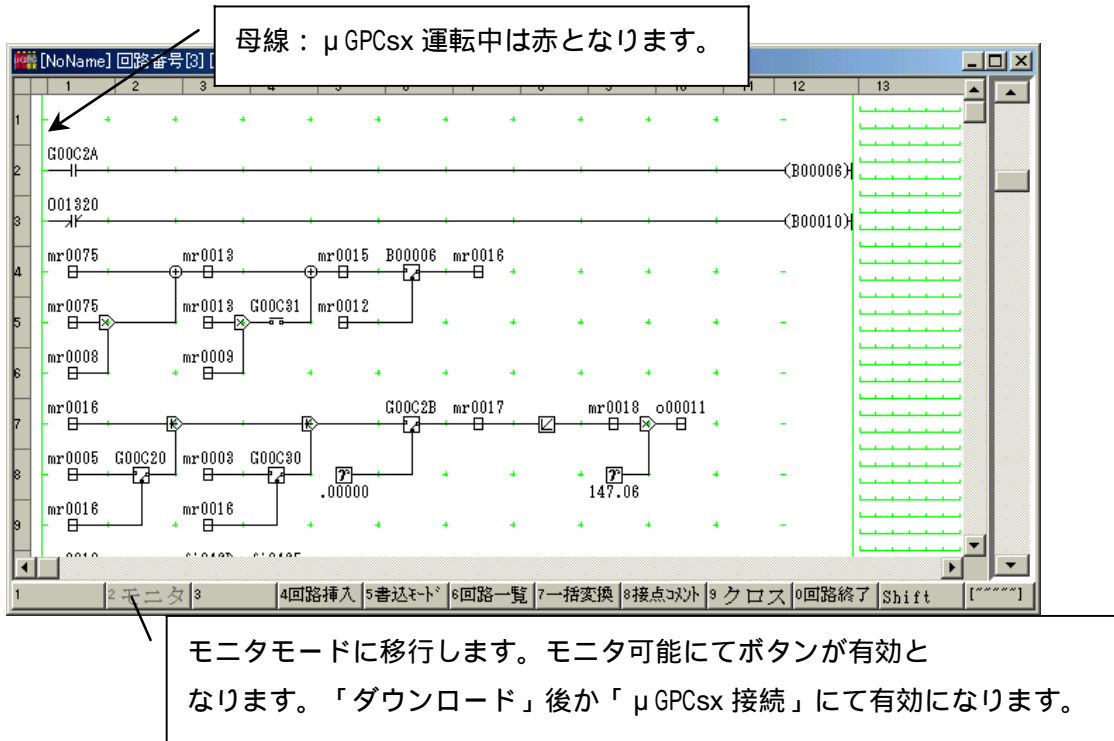
「ツールバー」
各ツールバーの表示非表示を選択します。

回路ウィンドウ内でシンボル、データ名が分からない場合は、マウスポインタをシンボルおよびデータ名の上に置くことにより、しばらくしてマウスポインタの下にシンボル、データ名の名称が表示されます。



4 - 6 オンラインでの回路

4 - 6 - 1 モニタ



- [読出モード] 読み出しモードへ戻ります。
- [デバッグ] デバッグモードへ移行します。
- [クロス] クロスリファレンスを検索します。
- [10<>16] 整数データの10進表示、16進表示を切り替えます。

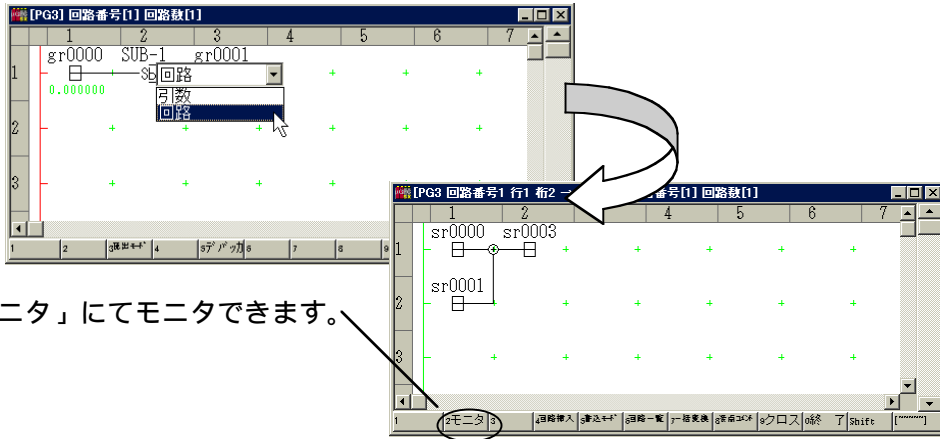
モニタでの回路の表示について

- A接点：コイルがONで赤となり、コイルがOFFで白（フォント色）となります。
- B接点：コイルがONで白（フォント色）となり、コイルがOFFで赤となります。
- NOT：論理演算入力の結果の反転を行ないます。（赤 NOT 白（フォント色） 白（フォント色） NOT 赤）
- コイル：左側の論理演算の結果に関わりなくコイルデータにより色を付けます。
- 罫線：左側の論理演算の結果を表します。結合線はOR条件により色を付けます。

4 - 6 - 2 サブルーチン内のモニタ

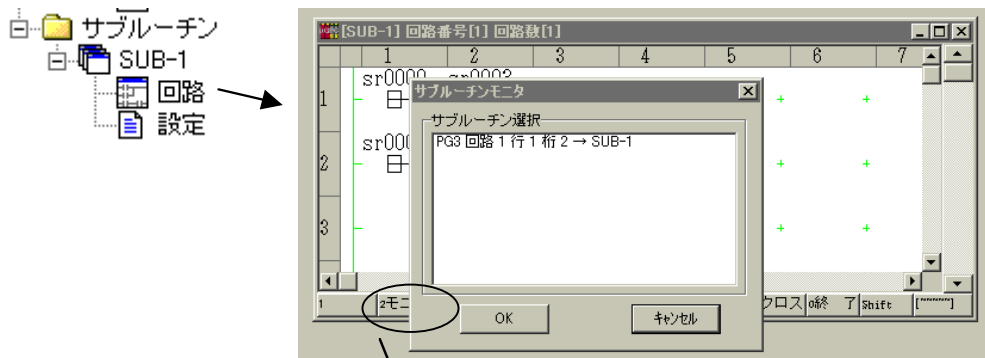
読み出し側からのモニタ

「読み出しモード」または「モニタ」から、サブルーチンシンボルをダブルクリックし、「回路」を選択します。するとサブルーチン回路が開きますので、



プロジェクトツリーからの選択

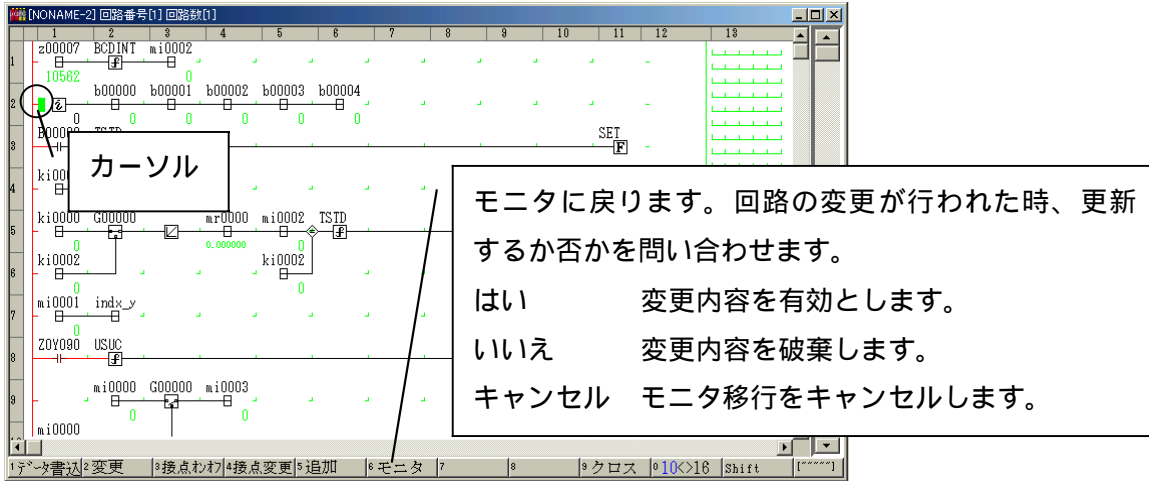
プロジェクトツリーのサブルーチンの「回路」を選択し、「モニタ」を選択すると、読み出し元の位置のリストが表示されますので、モニタしたいサブルーチンを選択して下さい。



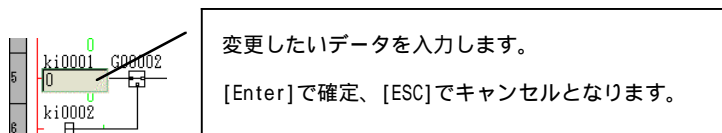
「モニタ」を選択すると、サブルーチンモニタダイアログボックスが表示されます。「OK」ボタンを左クリックするとモニタ可能となります。

4 - 6 - 3 デバッグ

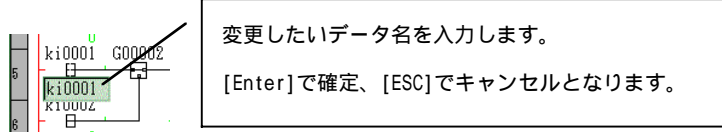
カーソル位置のシンボルに対してのデバッグ機能が行えます。デバッグ終了時、変更内容のキャンセルもできます。



データ書込 データを書き込みます。



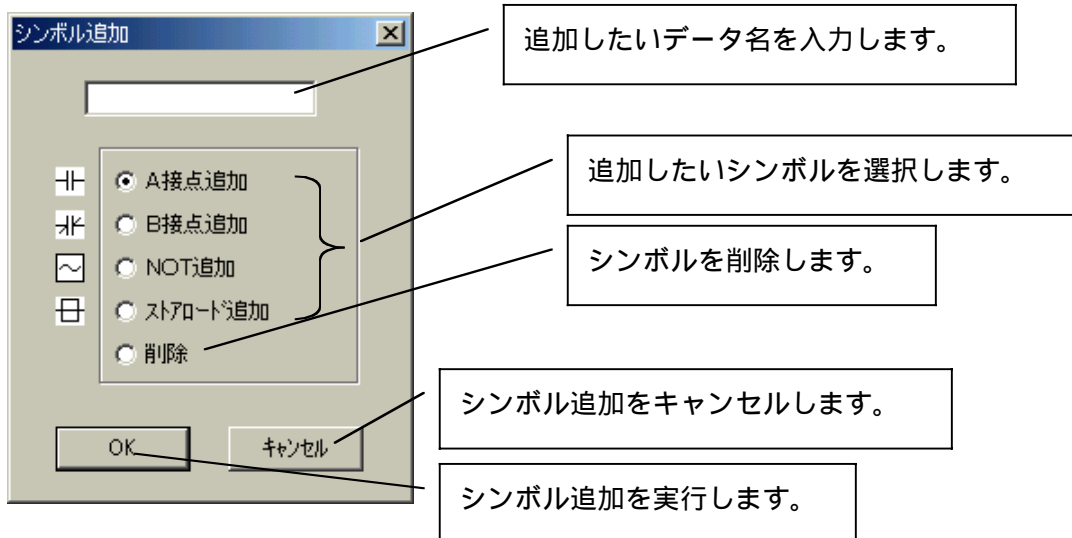
変更 データ名を変更します。



接点オンオフ 接点のリレーをオンオフ (ON、OFF) します。

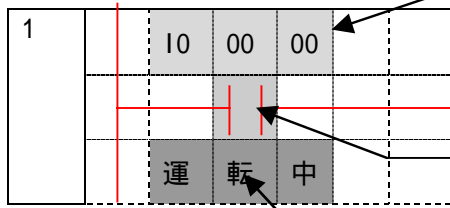
接点変更 接点を変更します。(A、a 接点 ←→ B、b 接点、c 接点)

追加 カーソル位置にシンボルを追加します。



ダブルクリックでのデバッグ機能

ラダー

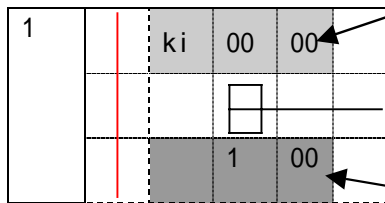


データ名変更 (テキストボックス表示)

接点変更 (A、B)

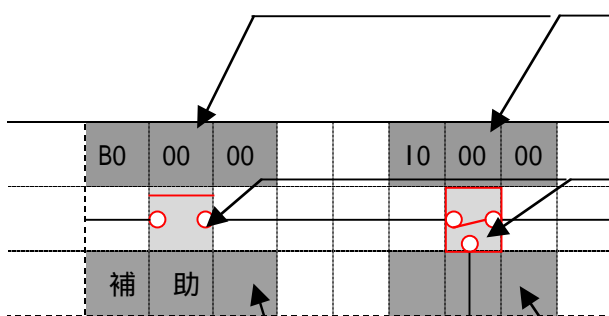
強制 ON/OFF (強制書き込み)

データフロー



データ名変更 (テキストボックス表示)

データ変更 (強制書き込み)



データ名変更 (テキストボックス表示)

接点変更 (a、b、c)

強制 ON/OFF (強制書き込み)



第5章 その他の項目の編集

5 - 1	リレー、レジスタ使用点数割付	5 - 1
5 - 2	定数データ	5 - 2
5 - 2 - 1	整数データ (ki)、実数データ (kr)	5 - 2
5 - 2 - 2	オンタイム (TS)、オフタイム (TR)	5 - 3
5 - 3	パターンデータ	5 - 4
5 - 4	メモリ転送定義	5 - 5
5 - 5	トレースバック	5 - 6
5 - 5 - 1	トレースバック	5 - 6
5 - 5 - 2	トレースバック設定ウインドウ	5 - 6
5 - 5 - 3	トレースバック設定ウインドウ リレー設定部...	5 - 7



第 5 章 その他の項目の編集

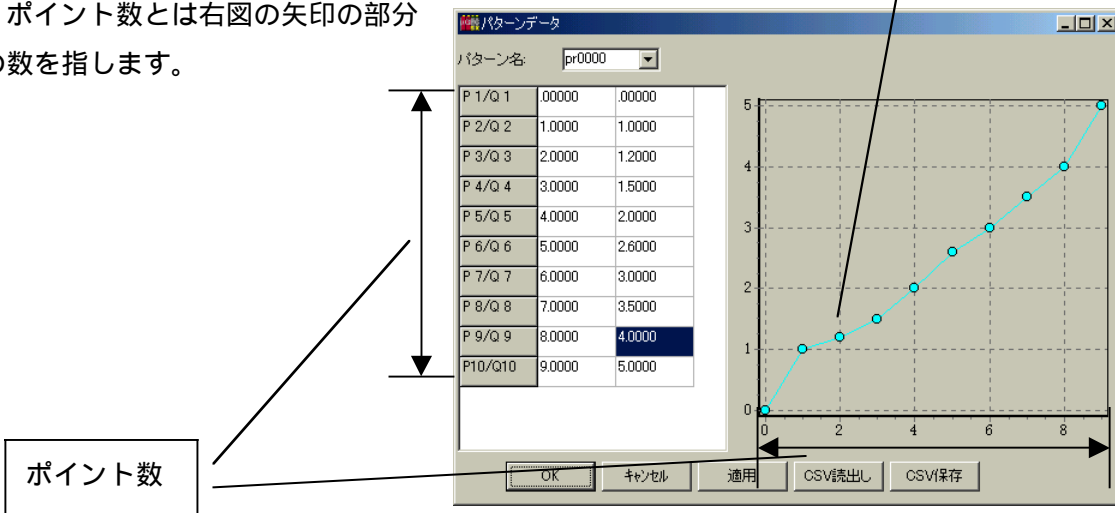
5 - 1 リレー、レジスタ使用点数割付

サブプログラム内で使用するローカルメモリの使用点数を設定します。
残りワード数を超えない数で設定して下さい。

整数パターン、実数パターン使用数を定義するとポイント数の定義が可能となります。

使用する場合は 2 ~ 100 の範囲で
設定して下さい。

ポイント数とは右図の矢印の部分
の数を指します。



パターン出力値

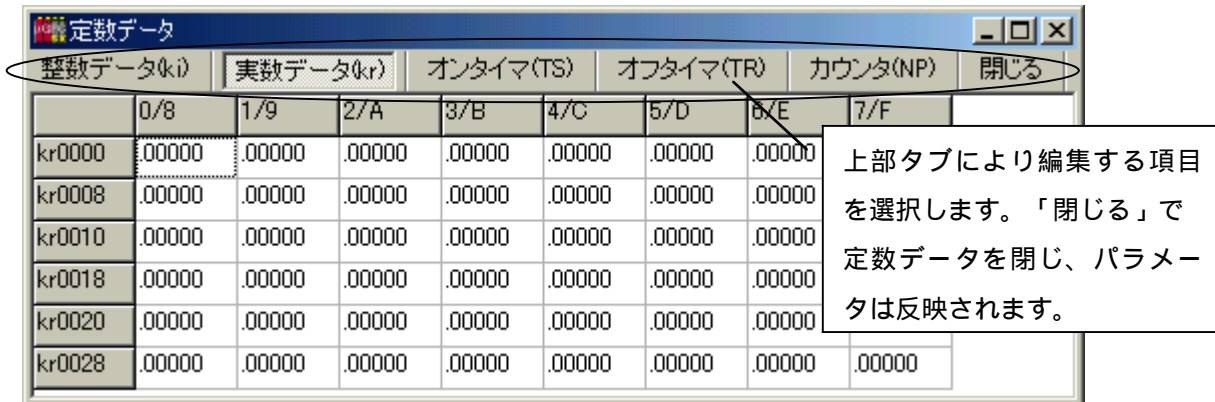
ポイント数

5 - 2 定数データ

回路内で使用する整数定数データ(ki)、実数定数データ(kr)、オンタイム値(TS)、オフタイム値(TR)、カウンタ値(NP)を定義します。

上部タブにより編集する項目を選択します。

5 - 2 - 1 整数データ(ki) 実数データ(kr)



定数値設定注意事項

定数値が0の時、

整数定数(kixxxx) では 000000

実数定数(krxxxx) では .00000 と表示します。

定数入力例

整数 123 (10進) 80H (16進)

 -123 (10進) 8005H (16進)

実数 123.4 .12345

 -123.4 -.2345

入力桁数を増やす為に 0.xxxx のゼロは省略とします。

5 - 2 - 2 オンタイム (TS)、オフタイム (TR)

定数データ										
整数データ(ki)	実数データ(kr)		オンタイム(TS)			オフタイム(TR)		カウンタ(NP)		閉じる
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F		
TS0000	00.00S	00.00S	00.00S	00.00S	00.00S	00.00S	00.00S	00.00S	00.00S	
TS0008	00.00S	00.00S	00.00S	00.00S	00.00S	00.00S	00.00S	00.00S	00.00S	

タイム値入力形式

00H00M . . . 60進 H:時

00M00S . . . 60進 M:分

00.00S . . . 10進 S:秒

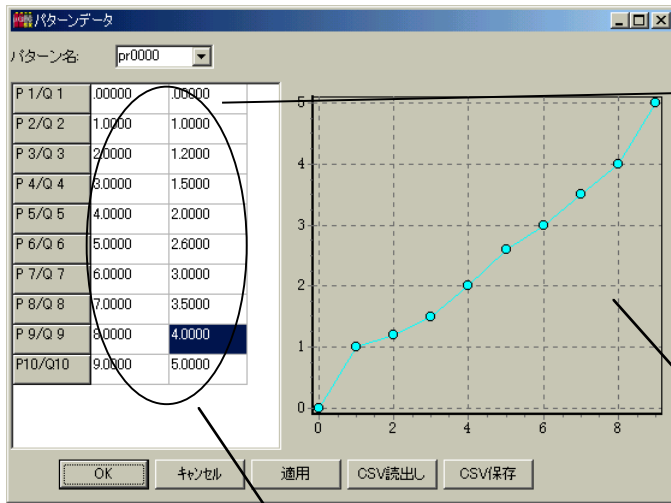
5 - 2 - 3 カウンタ (NP)

定数データ										
整数データ(ki)	実数データ(kr)		オンタイム(TS)			オフタイム(TR)		カウンタ(NP)		閉じる
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F		
NP0000	0	0	0	0	0	0	0	0	0	
NP0008	0	0	0	0	0	0	0	0	0	

カウンタ値設定範囲

0 ~ 65535

5 - 3 パターンデータ



パターン名
pi0000 ~ pi0004、
pr0000 ~ pr0004
のいずれかを選択します。

パターンプレビュー
パターンデータの P、Q に従
ってグラフ表示を行います。

パターンデータ P、Q の
表示、編集を行います。

OK

パターンデータを更新しウィンドウを閉じます。

キャンセル

変更内容を無効としウィンドウを閉じます。

CSV 読み出し

P、Q のパターンデータに CSV ファイルから読み込んだパターンデータを入力します。

“ファイルを開く”ダイアログボックスが表示されますのでファイル名を選択して下さい。

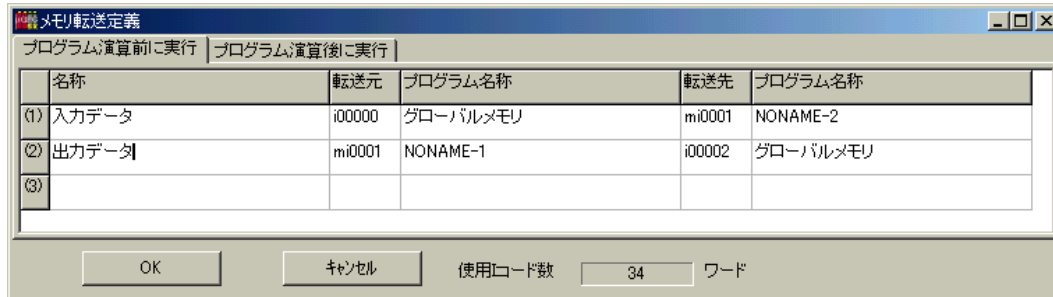
CSV 保存

P、Q のパターンデータを CSV ファイルに保存します。

“ファイルを保存する”ダイアログボックスが表示されますのでファイル名を選択して下さい。

5 - 4 メモリ転送定義

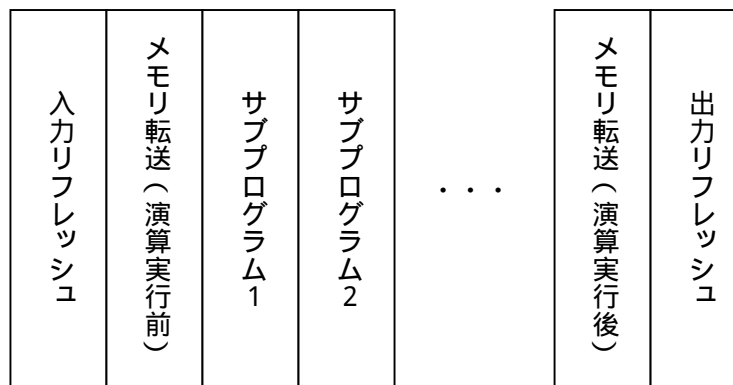
メモリ転送定義はタスク内のサブプログラムの演算実行前と実行後にデータの転送の定義を行うことができます。



グローバルメモリ、別々のサブプログラム間データやサブルーチンスタックレジスタ間での転送が行なえます。

メモリ転送はタスク内のサブプログラム演算の前後に処理が行われます。

下記の様な実行スケジュールとなります。



5 - 5 トレースバック

5 - 5 - 1 トレースバック

トレースバックとは連続したデータの一部分だけを詳しく調べたい時に、トリガーレジスタのいずれかのビットを ON させることにより、その ON したタイミングの前後 100 点のデータだけを保持することにより保持されたサンプルデータ値を後から読み出し、データの解析を可能とするものです。

まず最初に(CPU モジュール-パラメーターメモリ境界定義)においてトレースバック面数・サンプル数を設定してください。次にトリガーレジスタ、サンプルデータを設定します。設定はタスク 1, 2 それぞれに対して行えます。(サンプル点数は両方の合計となります。)

5 - 5 - 2 トレースバック設定 ウィンドウ レジスタ設定部

名称：任意の文字列を入力できます。

サンプルデータ：サンプルするレジスタ名を設定します。

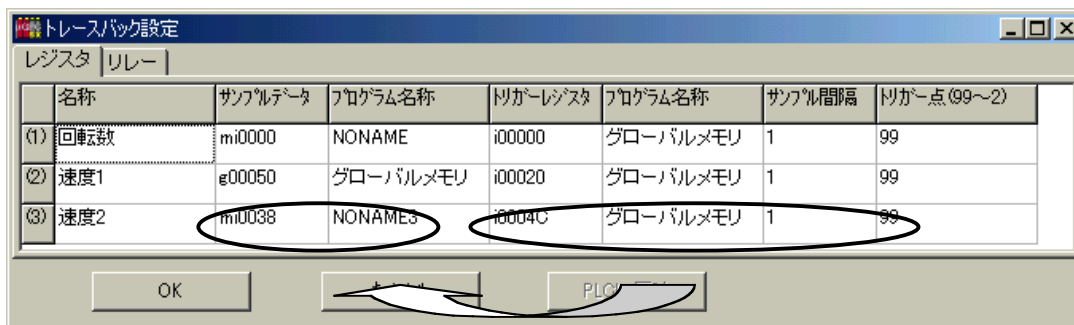
プログラム名称：サンプルするレジスタを使用しているプログラム名を設定します。

トリガーレジスタ：この値のどれかのビットの OFF ON でトリガーとなります。

サンプル間隔：スキャンによるスキップする値を定義します。

例) 1=1 スキャンに 1 サンプル 5=5 スキャンに 1 サンプル

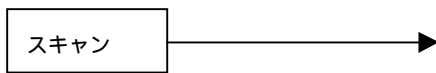
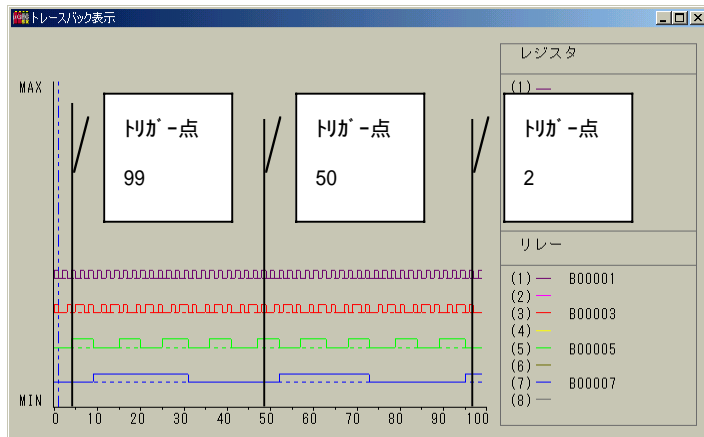
トリガー点：トリガーがかかった時刻に対して以後何点サンプルするか定義します。



レジスタの場合、サンプルデータおよびトリガーレジスタは最大 16 点まで設定可能です。

それは 1 つのトリガーレジスタに対して 1 つのサンプルデータしか設定することができないからです。よって上図の様に 3 番目のトリガーレジスタに対しても 3 番目のサンプルデータが設定できるのです。

トレースバックの例



5 - 5 - 3 トレースバック設定 ウィンドウ リレー設定部

“名称” ~ “トリガー点” まではレジスタと同様です。

名称	サンプルデータ	プログラム名称	トリガーレジスタ	プログラム名称	サンプル間隔	トリガー点(99~2)
(1)	B00000	NONAME-2	i00002	グローバルメモリ	1	90
(2)	B00022	NONAME-1				
(3)						
(4)						
(5)						
(6)						
(7)						

OK キャンセル PLCIに反映

リレーの場合、サンプルデータは 16 点まで設定可能ですが、トリガーレジスタは 1 点しか設定できません。それは 1 つのトリガーレジスタに対して 16 点までサンプルデータを設定することができるからです。よって上図のように 2 番目以降は“トリガーレジスタ” ~ “トリガー点” までは設定できません。

システム定義 - CPU - パラメーターメモリ境界定義 に設定により トレースバックメモリを設定しないとトレースバックの定義はできません。
 バッテリレス運転では電源を切断するとトレースバックデータは消えます。



第6章 システム定義情報

6 - 1	システム定義情報	6 - 1
6 - 2	システム定義情報の種類	6 - 1
6 - 3	システム構成定義の編集	6 - 2
6 - 3 - 1	ツールバーボタン	6 - 2
6 - 3 - 2	システム構成定義例	6 - 2
6 - 3 - 3	リモート I/O 使用時の定義例	6 - 2
6 - 3 - 4	モジュール挿入、プロパティ	6 - 3
6 - 3 - 5	I/O レジスタの割付	6 - 5
6 - 3 - 6	CPU モジュールパラメーターメモリ境界定義 ...	6 - 6
6 - 3 - 7	μGPCsx システムでのデータメモリの考え方	6 - 7
6 - 3 - 8	メモリ種別	6 - 8

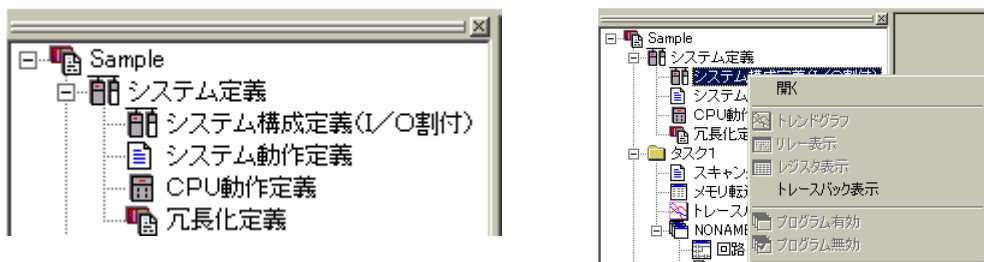


第 6 章 システム定義情報

6 - 1 システム定義情報

システム定義は、 μ GPCsx シリーズによりシステムを構築し、運用する上での全体システムの構成、動作設定および個別モジュールの動作設定を行ないます。

ツリーノードをダブルクリックか右クリックの「開く」により編集することができます。



6 - 2 システム定義情報の種類

システム構成定義

システムの構成、CPU メモリ定義、I/O パラメータを定義します。

システム動作定義

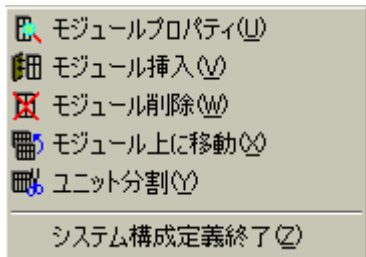
SX バスタクト時間、構成チェック待ち時間、CPU 初期化方法を定義します。

CPU 動作定義

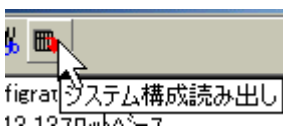
プロセッサ番号、WDT 時間、電源投入時の動作設定、バッテリーレス運転設定を定義します。

6 - 3 システム構成定義の編集

6 - 3 - 1 ツールバーボタン

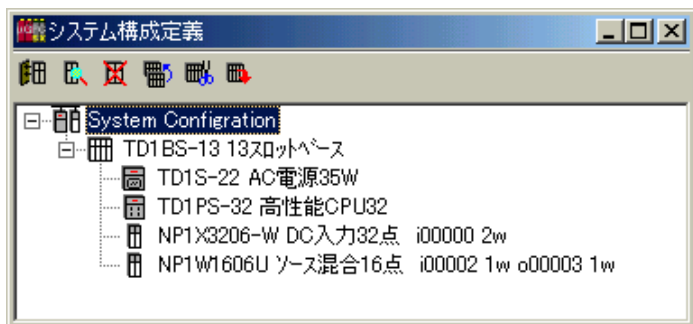


- モジュールのプロパティを表示します。
- 指定位置よりモジュールを挿入します。
- 指定位置のモジュールを削除します。
- 指定モジュールの位置を上のモジュールと交換します。
- 指定位置よりユニットを分割します。

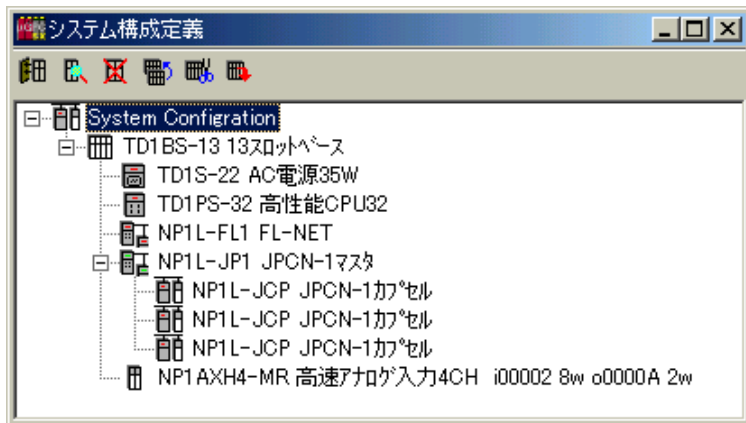


オンライン時、μ GPCsx よりシステム構成情報を読み出します。読み出せる条件はμ GPCsx が重故障でない時に限ります。

6 - 3 - 2 システム構成定義例



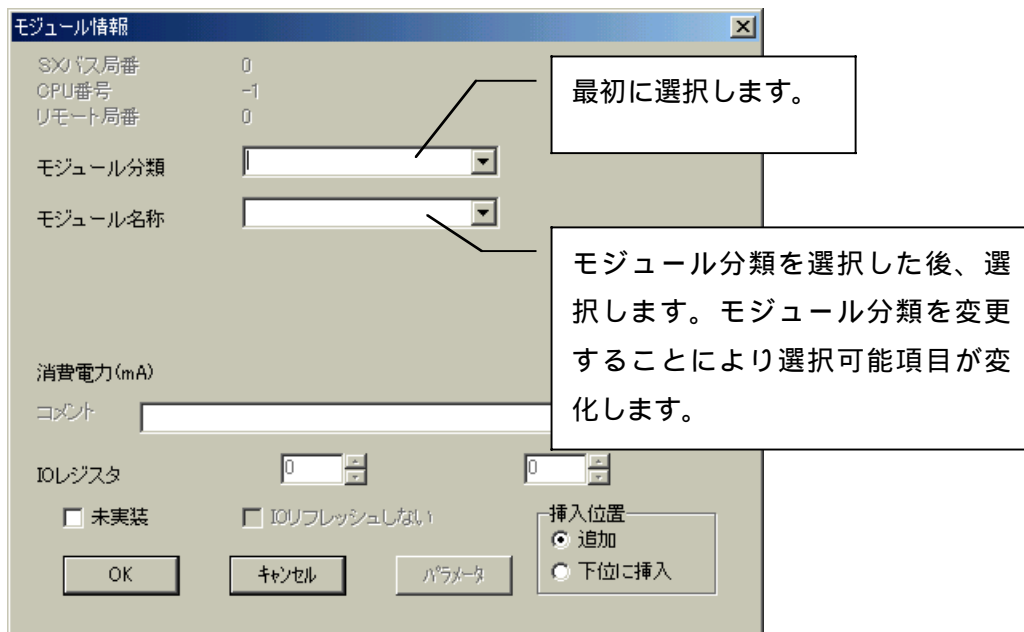
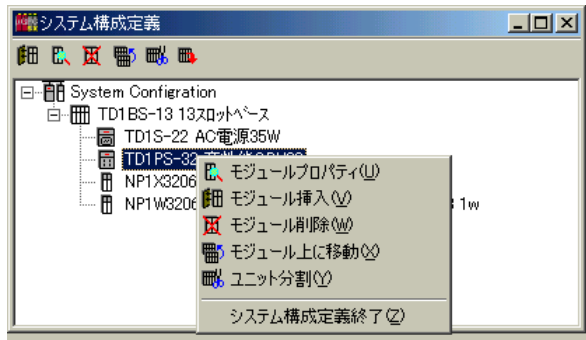
6 - 3 - 3 リモート I/O 使用時の定義例



リモート I/O (JPCN-1 「 OPCN-1 」, DeviceNet) ではマスターモジュールの下のスレーブモジュールを挿入します。

6 - 3 - 4 モジュール挿入、プロパティ

モジュールを挿入するとプロパティウィンドウが表示されますのでモジュール分類とモジュール名称を選択して下さい



CPU メモリ定義、IO パラメータを定義するにはモジュールのプロパティのパラメータボタンをクリックします。

パラメータの存在するモジュールのみパラメータボタンが有効になります。

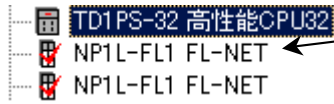
未実装とは

システム構成定義では定義しているが、実際の構成では無い場合に使用します。また構成定義の画面では、モジュールのマークに赤い印が表示されます。

CPU モジュールは未実装にすることができません。

リモート IO マスターモジュールを未実装にした場合には、全てのスレーブモジュールも未実装にしなければなりません。

追加



この位置に追加されます。

下位に挿入



この位置に挿入されます。

6 - 3 - 5 IOレジスタの割付

μGPCsx システムでの IO レジスタの割付は 16 進数表記で 32 ビットバウンダリとなります。リモート IO (OPCN-1 や DeviceNet) の入出力データも含まれます。

例)

System Configuration	アドレス	IO レジスタ割り当て
TD1BS-13 13 入力ポート		IO レジスタは割り当てません。
TD1S-22 AC 電源 35W		
TD1PS-74 高性能 CPU74		
NP1L-FL1 FL-NET		
NP1X1606-W DC 入力 16 点	0(h) 1(h)	i00000 1w 未使用領域になります。
NP1Y32T09P1 シック出力 32 点	2(h) 3(h)	o00002 2w
NP1Y64T09P1 シック出力 64 点	4(h) 5(h) 6(h) 7(h)	o00004 4w
NP1L-JP1 OPCN-1 マスタ	8(h)	リモート IO マスタは 1w 占有されます。
RSH64 VF64 用 OPCN-11/F	9(h) ~ 14(h) 15(h) ~ 1E(h)	i00009 12w o00015 10w
SDS64 同期制御用インタータ	1F(h) ~ 2A(h) 2B(h) ~ 34(h)	i0001F 12w o0002B 10w
KPB878 分散型マスターコントローラ	35(h) ~ 3E(h) 3F(h) ~ 4A(h)	i00035 10w o0003F 10w

6 - 3 - 6 CPUモジュールパラメータメモリ境界定義

メモリタイプ	設定値	ワード数	最終名
標準メモリ(M) 最大 131072W	8 KW	8192	G17FFF/e017FF/er17FE
グローバルメモリ(G0.g0.gr)			
ユーザーFBメモリ(FM) 最大 115712W	8 プログラム分	4096	17プログラムあたり 512
ローカルメモリ			
リテインメモリ(RM) 最大 129024W	0 KW	0	
リテインメモリ(保持メモリ)			
関数、サブルーチンインスタンスメモリ	3 KW	3072	現在 0
ワード使用			
トレースバックメモリ サンプル数 レジスタ	0 点	リレー16点	0 面分 使用数 0
ワード			
合計		3072	ワード
システムFBメモリ(SFM) 最大 129024W			
システムFBメモリ		114688	現在 0
ワード使用			

ここでは、1つのプロジェクト内で使用するメモリを設定します。

- ・グローバルメモリ

どのサブプログラム、サブルーチンからもアクセスできるメモリです。リレー、整数レジスタ、実数レジスタがあります。

- ・ローカルメモリ

サブプログラム内でのみ有効なメモリです。リレー、整数、実数に加えて微分、ラッチ、タイマ、カウンタなどがあります。

タスク 1・2 に存在するサブプログラムの数を設定してください。

1つのサブプログラム内で使用する点数を設定してください。

- ・リテインメモリ

電源を切っても内容を保持するメモリです。リレー、整数レジスタ、実数レジスタがあります。

- ・関数サブルーチンインスタンスメモリ

関数、サブルーチンを呼び出す毎に内部的に確保されるメモリです。現在使用ワード数はダウンロードしてから表示されます。関数、サブルーチンの使用数に依存するので設定値を超えた場合、後から増やすことができます。ユーザが使用することはできません。

- ・トレースバックメモリ

トレースバックで使用するメモリです。ユーザが使用することはできません。

- ・システムFBメモリ

システムFBで使用するメモリエリアです。ユーザが使用することはできません。

6 - 3 - 7 μ GPCsx システムでのデータメモリの考え方

μ GPCsx アプリケーションプログラムデータメモリはグローバルメモリ、ローカルメモリ、スタックレジスタという3つのグループに分けられます。

グローバルメモリ

すべてのサブプログラムからアクセスできるメモリです。

揮発性メモリ G0 (リレー)、g0 (レジスタ: 整数)、gr (レジスタ: 実数)

保持メモリ RI (リレー)、ri (レジスタ: 整数)、rr (レジスタ: 実数)

入力エリア I0 (リレー)、i0 (レジスタ: 整数)

出力エリア O0 (リレー)、o0 (レジスタ: 整数)

ローカルメモリ

サブプログラム内でのみ有効なデータメモリです。種類は揮発性メモリ、ラッチ、微分、タイマ、カウンタ、定数、パターン関数用メモリがあります。

B00000 ~、b00000 ~

LS0000 ~、LR0000 ~、LC0000 ~

US0000 ~、UC0000 ~、DS0000 ~、DC0000 ~、

TS0000 ~、TD0000 ~、tn0000 ~、TR0000 ~、TC0000 ~、

tf0000 ~、

NP

スタックレジスタ

サブルーチン内でのみ有効なローカルメモリです。

S10000 ~

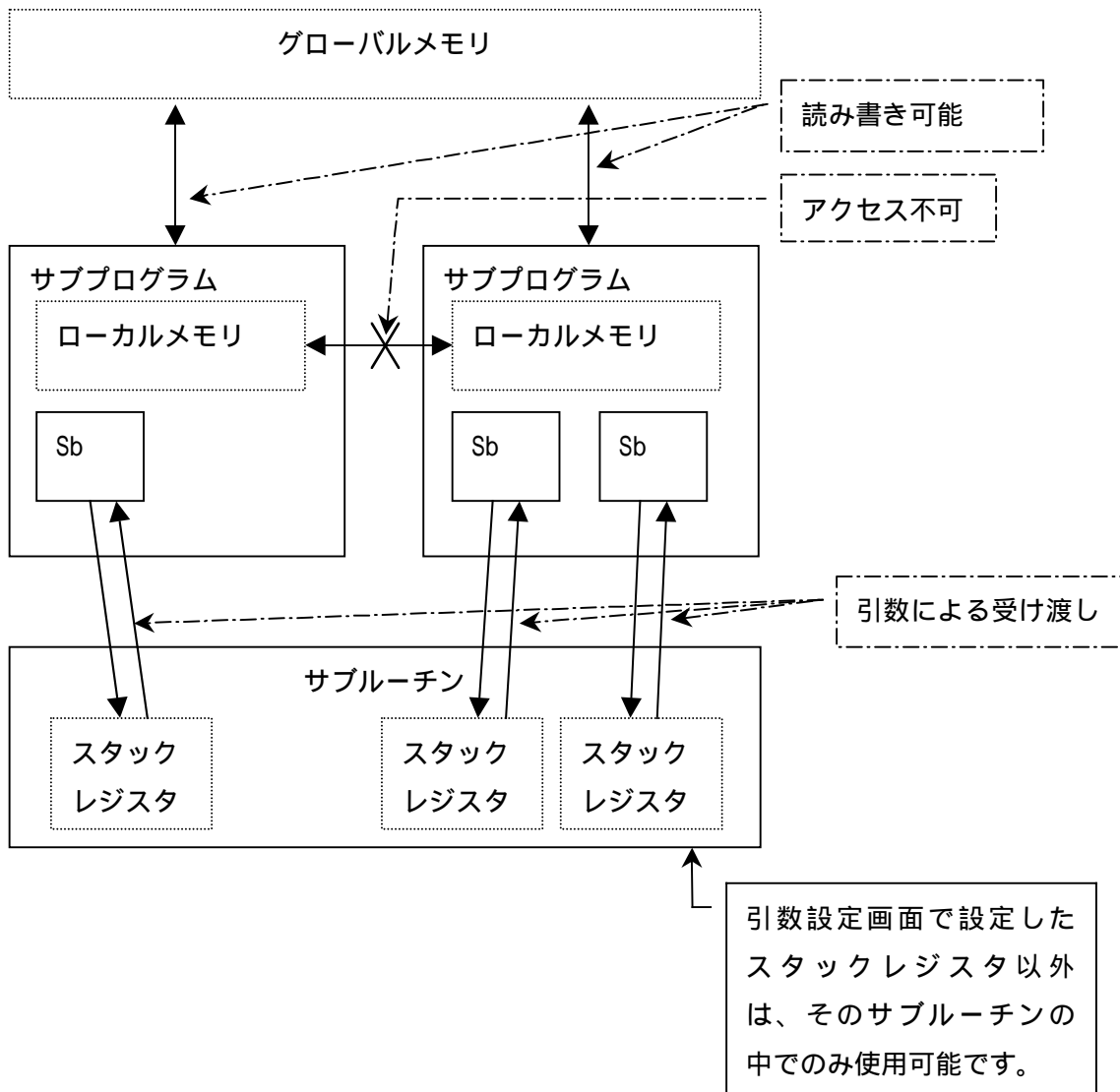
si0000 ~

sr0000 ~

6 - 3 - 8 メモリ種別

グローバルメモリ G00000 ~ g00000 ~ gr0000 ~ 等	ローカルメモリ B00000 ~、 b00000 ~ mi0000 ~、 mr0000 ~ ki0000 ~、 kr0000 ~ 等
スタックレジスタ S10000 ~ si0000 ~ sr0000 ~	

サブプログラム、サブルーチン、データメモリの関係



各 I/O モジュールのパラメータ

直結 I/O 動作モード定義

直結PIO動作モード定義	
項目	
デジタルフィルタ定数設定	設定しない
デジタルフィルタ定数設定値有効フラグ	無効

デジタルフィルタ定数設定

デジタル入力モジュールの入力遅延時間を設定します。

設定しない、100 μ s、1ms、3ms、3/10ms、10ms、30ms、100ms

デジタルフィルタ定数設定値有効フラグ

上記デジタルフィルタ定数設定を使用する場合、「有効」にして下さい。

下記モジュールが設定可能です。

NP1X1606-W、NP1X3206-W、NP1X6406-W、NP1X3202-W、NP1X3206-A

NP1W1606T、NP1W1606U、NP1W3206T、NP1W3206U

直結 I/O ホールド定義

項目	
直結HOLD定義	リセットモード
システムD0定義	しない

直結 HOLD 定義

「HOLD モード」

システムに異常があり、CPU モジュールが停止状態になったときに異常発生直前の出力状態を保持させたい場合や、CPU 停止直前の出力状態を CPU 停止中に保持させたい場合に使用する機能です。

「リセットモード」

出力を OFF にすると上記機能を無効にします。

下記モジュールが設定可能です。

NP1Y08T0902、NP1Y16T09P6、NP1Y32T09P1、NP1Y64T09P1、
 NP1Y08U0902、NP1Y16U09P6、NP1Y32U09P1、NP1Y64U09P1、
 NP1Y08S、NP1Y06S、NP1Y08R-04、NP1Y16R-08、NP1W1606T、
 NP1W1606U、NP1W3206T、NP1W3206U

システム D0 定義

1 コンフィグレーションに 1 つ、アプリケーションに関係なくシステムの運転状態を出力するビットを定義することができます。システム全体が正常に運転しているときは ON、システムに異常があるときは OFF します。設定可能なビットは出力モジュールの 0 ビット目のみです。

混合モジュール定義

混合モジュールに対しては直結 I/O 動作モード定義、直結 I/O ホールド定義を行いません。

項目	設定
デジタルフィルタ定数設定	設定しない
デジタルフィルタ定数設定値有効フラグ	無効
直結HOLD定義	リセットモード
システムDO定義	しない

下記モジュールが設定可能です。

NP1W1606T、NP1W1606U、NP1W3206T、NP1W3206U

JPCN-1 マスターモジュールパラメータ

JPCN-1 マスターモジュールパラメータは「NP1L-JP1」に対して設定します。

モジュール分類-I0 マスター



(1) 応答時間定義

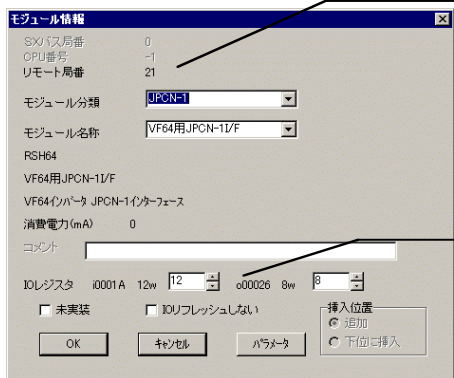
JPCN-1 通信においてマスタでのフレームの受信から送信開始までの応答時間（受信より送信までの切り換え時間）を設定します。

応答時間は以下の中で選択してください。

速度	備考	イニシャル設定
200 μs	JEM-F3008 準拠	○
150 μs		
100 μs		
最速	50 μs 程度です。	

注) 応答時間の値が特に指定されていないスレーブ機器が存在する場合は、200 μs を選択してください。

JPCN-1 スレーブ固有動作定義



JPCN-1 スレーブ局番となります。
IO 数や構成により自動的に割り振ります。
注) 設定はできません。

JPCN-1 スレーブの IO 数を設定します。
レジスタ、リレー番号は自動的に決まります。

JPCN-1 スレーブパラメータ

項目	
入力データ(DI)	なし
出力データ(DO)	なし
読み出しデータ(DR)	なし
書き込みデータ(DW)	なし
I/Oタイプ指定(ST)	なし
通信監視時間	1,500ms

(1) 通信監視時間定義

個々のスレーブ局に対して通信監視時間を設定します。

監視時間は以下の範囲で設定します。

- ・デフォルト：1500ms
- ・設定値：1 ~ 65534 (× 10ms)

注) 通信監視異常となったときのリセットは、JPCN-1 マスタ側電源の再投入を行いません。このとき、異常時の出力は“ホールド状態”となりますが、リセットを行なうことにより出力がOFF になりますので、注意してください。

(2) StypeM 定義

スレーブの仕様を設定します。

マスタは初期設定サービス時、各スレーブにこのデータをもとに構成のチェックをします。

記号	名称	内容
D I	入力データ	入力信号をもつスレーブ使用時 (DI モジュールなど)
D O	出力データ	出力信号をもつスレーブ使用時 (DO モジュールなど)
D R	読み出しデータ	データを出力するスレーブ使用時
D W	書き込みデータ	データを入力するスレーブ使用時
S T	I/O タイプ指定有無	この信号をチェックすることにより上記設定 (DI、DO、DR、DW) が有効となります。
S A	I/O 並び情報有無	1 モジュール内で入出力混合スレーブの場合設定します。本モジュールでは未使用です。

なお、StypeM はスレーブにより設定が変わりますので詳細は各スレーブで確認してください。

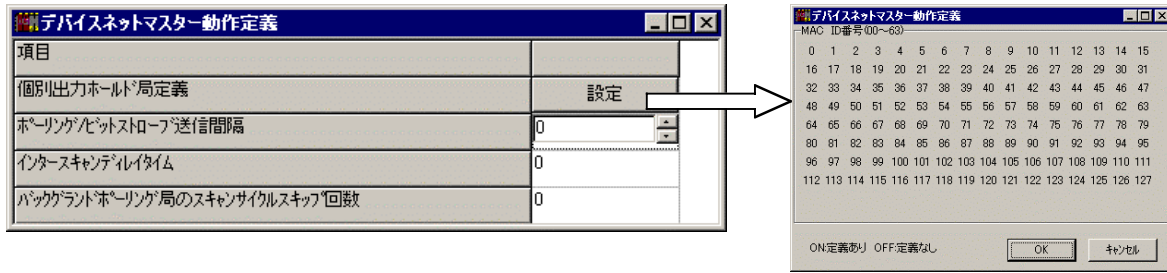
下記モジュールに対して設定します。

NP1L-RJ1、RSH64、SDS64、KPB878、TPC-193、

上記の型式以外のスレーブ機器を接続する場合は「NP1L-JCP (JPCN-1 カプセル)」を使用します。

デバイスネットマスター動作定義

NP1L-DN1 (モジュール分類-I/O マスター)



個別出力ホールド局定義

出力ノードのうちホールド設定が必要なノードの MAC ID を左クリックして選択します。

注) ハードウェアスイッチで出力ホールド設定するノードには効果がない場合があります。使用するスレーブノードの仕様を確認してください。

ポーリング/ビットストローブ送信間隔 (EPR)

スキャン方式がポーリングまたはビットストローブに設定されたスレーブノードにて使用される expected_packet_rate 属性値です。この属性値を 4 倍した値が、スレーブノードのタイムアウト値となり、タイムアウト値を超えてもスレーブノードがデータを受信しないと、そのノードは DeviceNet から脱落します。この値を 0 と設定したときは、接続されるスレーブノードの台数と伝送速度により下表のようになります。

インタースキャンディレイタイム (ISD)

マスターノードはスレーブノードに対して I/O データを送信した後、ここで設定した時間だけ、スレーブノードの応答を待ちます。

この値を 0 としたときは、接続されるスレーブノードの台数と伝送速度により下表のように設定されます。

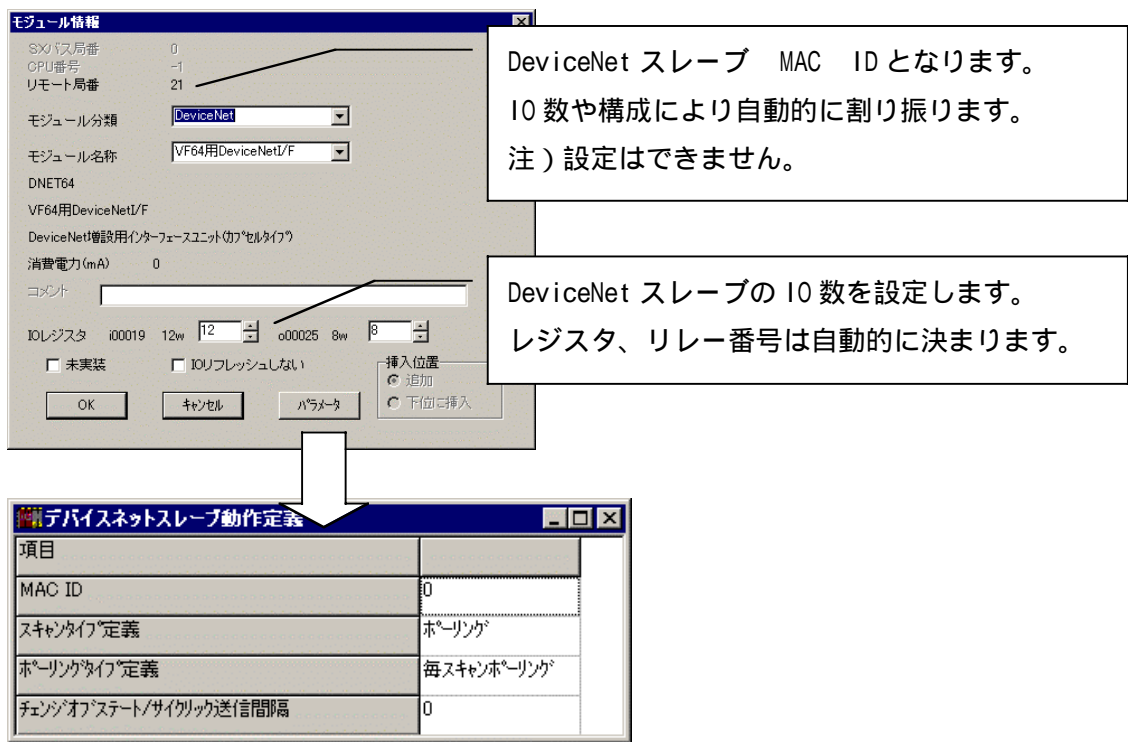
< 設定値 0 のときの EPR、ISD 値の値 >

接続するスレーブノード台数	伝送速度(bps)	EPR(ms)	ISD(ms)
1 - 21台	1 2 5	3 0	4 0
	2 5 0	2 6	3 6
	5 0 0	2 0	2 6
2 2 - 4 2台	1 2 5	6 4	8 6
	2 5 0	5 0	6 6
	5 0 0	3 8	5 0
4 3 - 6 3台	1 2 5	9 4	1 2 6
	2 5 0	7 6	1 0 0
	5 0 0	5 6	7 6

バックグラウンドポーリング局のスキャンサイクルスキップ回数

バックグラウンドポーリングを行なうように設定されたスレーブノードが、スキャンサイクルを何回スキップするかを設定します。最大 10000 回まで設定できます。0 を設定した場合は、毎スキャンポーリングと同じポーリング動作となります。

DeviceNet スレーブパラメータ



MAC ID

モジュール情報のリモート局番 (MAC ID) と同じ値を設定してください。

スキャンタイプ定義

DeviceNet スキャンタイプを設定します。スレーブノード仕様に合わせて下記より選んで設定してください。

ポーリング、ビットストローブ、チェンジオブステート、サイクリック

ポーリングタイプ定義

ポーリングタイプを下記より選択します。

毎スキャンポーリング、バググランドポーリング

チェンジオブステート/サイクリック送信間隔

チェンジオブステート/サイクリック送信間隔を設定します。スキャンタイプをチェンジオブステートまたはサイクリックに設定した場合、送信間隔を 0 に設定すると、そのノードからのデータ送信は行われません。

下記モジュールに対して設定します。

NP1L-DCP、DNET64

「NP1L-DCP (DeviceNet カプセル)」とは上記型式以外のモジュールを設定します。

FL - net モジュール動作定義

FL - net モジュール動作定義は「NP1L-FL1」のパラメータとして設定します。

項目	
機種コード定義 小分類	高性能CPU
ネットワークパラメータ書き込み(P):	許可
ブロックデータ書き込み(B):	許可
起動/停止指令(R):	許可
IPアドレス(I)HH:	192
IPアドレス(I)HL:	168
IPアドレス(I)LH:	250
トークン監視時間(T): (ms)	50
最小フレーム間隔(F): (×100 μs)	10
ノード名/設備名 有無	無効
ノード名/設備名	
各領域設定	無効
先頭アドレス(オフセット)自動設定	無効
領域1(A)先頭アドレス(0-511)	0
領域1(A)ワードサイズ(0-512)	0
領域2(R)先頭アドレス(0-8191)	0
領域2(R)ワードサイズ(0-8192)	0
領域1, 2 送信バンク切替操作CPU定義(S)	設定
全領域 受信バンク切替操作CPU定義(S)	設定
構成登録定義1	設定
構成登録定義2	設定

動作定義

機種コード定義 小分類

使用する CPU の種別を設定します。μGPCsx シリーズでは「高性能 CPU」のみです。

ネットワークパラメータ書き込み (P)

ブロックデータ書き込み (B)

起動/停止指令 (R)

FL-net ネットワーク (他ノード) からのデータ書き込み、起動/停止の許可/禁止を設定します。

IP アドレス (I) HH

IP アドレス (I) HL

IP アドレス (I) LH

IP アドレスを設定します。デフォルト値は 192 (HH), 168 (HL), 250 (LH), *** で、ホストアドレス部はモジュール前面のノード番号設定した値となります。

トークン監視時間 (T) 単位 ms

コモンメモリ領域を使用したサイクリック伝送の送信時間を監視するための監視時間を設定します。デフォルト値は 50ms で、設定範囲は 1~255ms です。μ-GPCH(TPC-196)と通信する場合は、「50」以上としてください。

最小フレーム間隔 (F) 単位 ×100 μs

他ノードからトークンを受けて自ノードがフレームを出すまでの時間を、フレーム間隔と呼びます。このとき、各ノードが最低限フレームを出すまで待たなければならない時間を最小フレーム間隔と呼びます。デフォルト値は 10×100 μs で、設定範囲は 0~50 です。単位は 100 μs です。「0」と設定した場合、待ち時間なしの最速で動作します。μ-GPCH(TPC-196)と通信する場合は、「50」としてください。

ノード名/設備名 有無

下記ノード名を指定するときは「有効」としてください。

ノード名/設備名

ノード名を指定します。最大半角 10 文字です。

領域設定

サイクリックデータ転送における自ノードの送信領域を設定します。

各領域設定

自ノード送信領域を使用する場合は「有効」としてください。

先頭アドレス (オフセット) 自動設定

ノード番号により先頭アドレスを計算します。

領域 1 (A) 先頭アドレス

領域 1 の自ノード送信領域の先頭アドレスを指定します。

領域 1 (A) ワードサイズ

領域 1 の自ノード送信ワード数を設定します。

領域 2 (R) 先頭アドレス

領域 2 の自ノード送信領域の先頭アドレスを指定します。

領域 2 (R) ワードサイズ

領域 2 の自ノード送信ワード数を設定します。

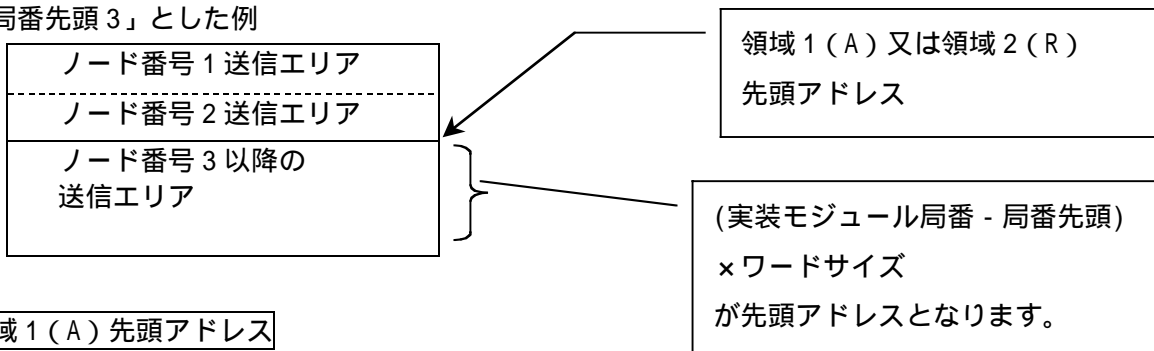
先頭アドレス(オフセット)自動設定 について

先頭局番を指定する事により、ダウンロード時に実装された FL-net モジュールのノード番号を読み出し「領域 1 (A) 先頭アドレス」、「領域 2 (R) 先頭アドレス」を「先頭局番」以降のノードの先頭アドレスを自動計算します。設定されると下記、領域 1 (A) 先頭アドレス、領域 2 (R) 先頭アドレスの設定は意味が変わります。

注) CPU モジュールが正常又はシステム初期化後の時に限ります。

FL-net サイクリックデータエリア

「局番先頭 3」とした例



領域 1 (A) 先頭アドレス

自動計算したい領域 1 の先頭アドレスを指定します。

領域 1 (A) ワードサイズ

「局番先頭 N」以降の領域 1 の共通な自ノード送信ワード数を指定します。

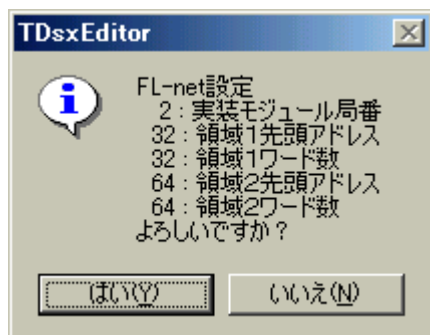
領域 2 (R) 先頭アドレス

自動計算したい領域 2 の先頭アドレスを指定します。

領域 2 (R) ワードサイズ

「局番先頭 N」以降の領域 2 の共通な自ノード送信ワード数を指定します。

(システム定義情報) ダウンロード時、下記メッセージボックスが表示されますので、この設定でよろしければ、「はい (Y)」でダウンロードを続けてください。「いいえ (N)」でダウンロードを中止します。



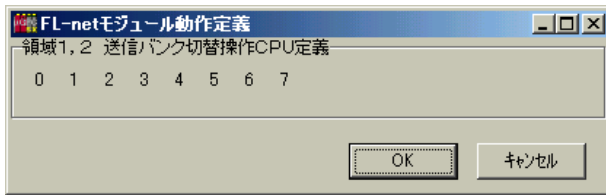
設定例) ダウンロード先ノード番号が 2 の場合

先頭アドレス(オフセット)自動設定	局番先頭1
領域1(A)先頭アドレス(0-511)	0
領域1(A)ワードサイズ(0-512)	32
領域2(R)先頭アドレス(0-8191)	0
領域2(R)ワードサイズ(0-8192)	64

領域 1 先頭アドレス : $0 + (2-1) \times 32 = 32$

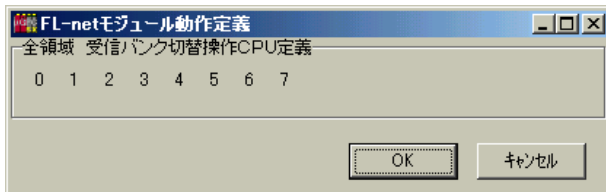
領域 2 先頭アドレス : $0 + (2-1) \times 64 = 64$

領域 1、2 送信バンク切替操作 CPU 定義 (S)



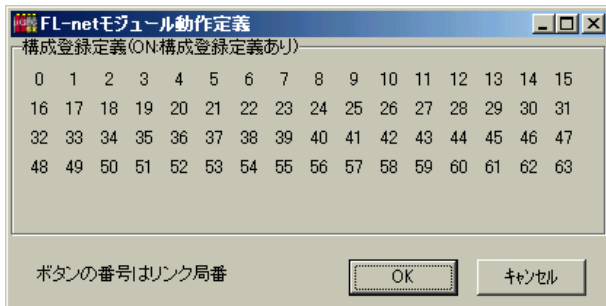
コモンメモリ内自ノード送信領域のリフレッシュ方式（同期 / 非同期）の設定および、同期方式時の切替指令を出す CPU を設定します。

全領域 受信バンク切替操作 CPU 定義 (S)



コモンメモリ内自ノード受信領域のリフレッシュ方式（同期 / 非同期）の設定および、同期方式時の切替指令を出す CPU を設定します。

構成登録定義 1、構成登録定義 2



自モジュールのノード番号を含め、FL-net 上に接続する全てのノードのノード番号を登録します。ただし0、255 は設定禁止です。

システム動作定義



(1) SX バスタクト時間

SX バスタクト時間は、SX バスに接続されたモジュール（入出力モジュールなど）とのデータ授受の周期です。

0.5ms から 10.0ms まで 0.5ms 刻みで設定することができます。

“ 0.5ms ”、“ 1.0ms ”、“ 2.0ms ”と 1.0ms より 1ms 刻みで設定します。

デフォルト値は 1.0ms です。

注 1) 0.5ms タクト周期は、高性能 CPU におけるシングル CPU、直結 I/O 256 点以下、リモート I/O および通信モジュール“ なし ”の条件で実行可能です。

注 2) CPU のファームウェアバージョンが V34 以前および V3A ~V3Z の製品では 1ms 刻みで最大 20ms まで設定できます。

(2) 構成チェック待ち時間

CPU モジュールはシステムの電源が投入されると、CPU モジュールの初期化を行ないます。初期化を完了すると、

1 つのコンフィグレーション上にあるモジュールの構成チェックを始めます。この構成チェックが終わるまでの時間を“ 構成チェック待ち時間 ”で設定します。

(3) 初期化方法

システムの電源が投入されると CPU モジュールの初期化が行われますが、その際、CPU モジュール内部のメモリ診断を実行するか、実行しないかを設定します。メモリ診断を実行する場合、CPU モジュールの初期化時間は約 4.5s、メモリ診断を実行しない場合は約 2.5s になります。

注 1) メモリ診断では、デバイスリード/ライトチェックを行なっています。

CPU のタクト周期設定について

SX バスでは、タクト周期に同期して CPU モジュールと I/O モジュールのデータ交換が行われます。以下にタクト周期毎によるスキャンの時間算出の概算式を示しますが、基本的に、SX バスのタクト周期は、システム構成に依存します。また、タクト周期でのタスク周期を必要とするシステムでは、アプリケーションプログラムの処理ステップ数に依存します。正確な実行時間は、実機での確認をする必要があります。

< タクト周期に依存するシステム構成 >

- ・ I/O 点数
- ・ CPU 台数
- ・ リモート I/O マスタ台数
- ・ 通信モジュール台数

(1) システム構成によるタクト周期の概略算出式

タクト周期 T (μ s)

CPU1 台 + 直結 I/O の場合 : T = Tb 【直結入出力点数によるベース時間 (Tb)】

直結 I/O 点数(点)	0	32	128	256	512	1024	2048	3072	4096	6144	8192
ベース時間 Tb(μ s)	418	504	507	510	556	695	1042	1388	1520	1711	1911

上記時間は、I/O の入力 : 出力比 = 1 : 1 の場合を示しており、出力が多くなると時間は増加し、出力が少なくなると減少しますが、その変動範囲は、約 ± 20 % です。

注 1) ただし、0.5ms タクト周期は、シングル CPU 、直結 I/O256 点以下、通信モジュール“なし”の条件で実行可能です。

注 2) タクト周期は、0.5、1、2、3、.....、18、19、20ms の中で選択可能であり、上記表の値を切り上げた値を選択します。

マルチ CPU の場合：【CPU 台数：n】

- $T = T_b + 210n$ (直結入出力点数：2048 点以下)
- $T = T_b + 200n + 190$ (直結入出力点数：2048 点を超える場合)

シングル CPU + リモート I/O の場合：【リモート I/O マスタ台数：m】

- $T = T_b + 250m + 430$ (直結入出力点数：2048 点以下)
- $T = T_b + 280m + 730$ (直結入出力点数：2048 点を超える場合)

マルチ CPU + リモート I/O の場合：【CPU 台数：n、リモート I/O マスタ台数：m】

- $T = T_b + 340n + 200m + 400$ (直結入出力点数：2048 点以下)
- $T = T_b + 405n + 260m + 340$ (直結入出力点数：2048 点を超える場合)

CPU1 台 + 直結 I/O () で通信モジュールを追加した場合：【通信モジュール台数：p】

- $T = T_b + 40p + 250$
- (~) で通信モジュールを追加した場合：【通信モジュール台数：p】
- $T = (\sim \text{で求めた時間}) + 85p$ 【リモートマスタモジュールがない場合】
- $T = (\sim \text{で求めた時間}) + 128p$ 【リモートマスタモジュールがある場合】

注3) リモート I/O は、2048 点/1 回線で計算しています。

付録2 - 2

(2) タクト周期でスキャン実行する場合の演算性能算出式

$$\text{演算時間} = \text{タクト周期時間} - [\text{SBM オーバヘッド時間 (200 }\mu\text{s)}] \\ - [\text{I/O リフレッシュ時間}] - [\text{POU 制御時間}]$$

$$\cdot \text{I/O リフレッシュ時間} = (2n + m + 60) \mu\text{s}$$

【n : I/O モジュール数、m : I/O 合計ワード数】

$$\cdot \text{POU 制御時間} = \text{PG 制御時間} + \text{ユーザ FB 制御時間} + \text{ユーザ FCT 制御時間} \\ = (4a + 6b + 7c) \mu\text{s}$$

【a : PG 本数、b : ユーザ FB コール回数、c : ユーザ FCT コール回数】

注1) POU とはプログラム構成単位の略です。

注2) PG とは各タスクのプログラムを称します。システムタスク×2、タスク1、タスク2で
a=4 となります。

注3) ユーザ FB とはサブプログラム、サブルーチンを称します。

注4) ユーザ FCT とは関数、システム関数を称します。

$$\text{演算時間} = [\text{タクト周期時間}] - (2n + m + 60) - (4a + 6b + 7c) \mu\text{s}$$

$$\text{プログラムステップ数} = [\text{演算時間} / 1 \text{ 命令実行時間} / 1024] k \text{ ステップ}$$

$$\text{プログラムステップ数} = [\text{演算時間} / 20.48] k \text{ ステップ} \text{【1 命令実行時間} = 20\text{ns の場合】} \\ = [\text{演算時間} / 61.44] k \text{ ステップ} \text{【1 命令実行時間} = 60\text{ns の場合】}$$

注5) 各種命令の実行時間は、プログラミングマニュアル 処理速度を参照願います。

注6) 命令実行時間は演算の対象となる変数が割り付けられるメモリのアクセス時間により変動します。したがって、命令がアクセスする変数の数によって下記のアクセス時間増加分を命令実行時間に加算する必要があります。メモリのアクセス時間は以下のようになります。

I/O メモリ : 基準時間 (20ns)

その他のメモリ : 40ns 加算

プロセッサバス経由の他の CPU メモリ : 3 μ s

(FI、fi、fr、EI、ei、er、w0~w7 がこれに該当します。)

(3) 時間計算例

シングル CPU の場合

スキャン時間 (タクト時間)	システム構成	POU 制御時間 (PG/FB/FCT 数)	プログラム実行可能時間 (20ns ステップ換算)
0.5ms	CPU...1 台 (通信モジュール使用不可) 直結 I/O : 256 点	68 μ s (4/4/4)	124 μ s (6k ステップ)
1ms	CPU...1 台 直結 I/O : 1024 点	136 μ s (8/8/8)	412 μ s (20k ステップ)
2ms	CPU...1 台 直結 I/O : 2048 点	480 μ s (16/32/32)	876 μ s (42k ステップ)
	CPU...1 台 リモート I/O : 2048 点	480 μ s (16/32/32)	876 μ s (42k ステップ)
	CPU...1 台 リモート I/O...2 台 : 4096 点 直結 I/O : 2048 点	480 μ s (16/32/32)	492 μ s (24k ステップ)

マルチ CPU の場合

スキャン時間 (タクト時間)	システム構成	POU 制御時間 (PG/FB/FCT 数)	プログラム実行可能時間 (20ns ステップ換算)
4ms	CPU...4 台 リモート I/O...2 台 : 4096 点 直結 I/O : 2048 点	480 μ s (16/32/32)	2492 μ s (484k ステップ=121k*4)
	CPU...4 台 リモート I/O...2 台 : 4096 点 直結 I/O : 1024 点 通信...2 台	480 μ s (16/32/32)	2620 μ s (508k ステップ=127k*4)

(4) 1:1 ウォームスタンバイ冗長化システムにおけるタクト周期の概略計算式

1:1 ウォームスタンバイ冗長化システムにおいては通常のマルチ CPU システムよりも稼動/待機 CPU 間の等値化データ転送が必要な分、タクト時間が増加します。下記に概略計算式を示します。下記において、冗長化タクト時間 1 (T_{R1}) と冗長化タクト時間 2 (T_{R2}) の大きい値の時間を採用します。

冗長化タクト時間 1 : T_{R1} [μs]

T_{R1} = 通常のタクト時間 + $596 \times N + 430$ (直結入出力点数 : 2048 点以下)

T_{R1} = 通常のタクト時間 + $626 \times N + 730$ (直結入出力点数 : 2048 点を越える場合)

N : 冗長化 CPU ペア数

注) 通常のタクト時間は (1) の概略計算式で求めたタクト時間です。このとき、CPU 台数は冗長化 CPU のペア数を使用して算出します。

冗長化タクト時間 2 : T_{R2} [μs]

T_{R2} = (I/O リフレッシュ時間) + $T_{DMA} + T_{CPY} + 200$ [μs]

・ I/O リフレッシュ時間 : $(2n + m + 60)$ [μs]

【n : I/O モジュール数、m : I/O 合計ワード数】

・ TDMA = [CPU 以外の SX バスモジュール数) + (CPU モジュール数) $\times 2$
+ (リモートマスタモジュール数 $\times 55$) + (直結/リモート全 I/O のワード数) + 512]
 $\times 0.5$ [μs]

・ TCPY = (標準メモリ、リテインメモリの等値化変数ワード数) $\times 0.3$
+ (ユーザ FB メモリ リテインメモリワード数) $\times 0.35$
+ 10 [μs]

CPU 動作定義

項目	
プロセッサ番号	0
WDT時間(ms)	4000
電源投入時の動作指定	RUN:運転/TERM:運転
バッテリーレス運転設定	しない

CPU 動作定義にはウォッチドグタイム、電源投入時の動作説明、バッテリーレス運転の 3 つの設定項目があります。

(1) ウォッチドグタイム

ウォッチドグタイムは 1ms から 4095ms まで設定できます。デフォルト値は 4095ms です。

(2) 電源投入時の動作指定

CPU モジュール前面のキースイッチが RUN または TERM 位置のときにシステム電源投入時の CPU モジュールの動作を指定します。

本設定と CPU モジュール前面のキースイッチ位置による CPU モジュールの動作は次のとおりです。

デフォルト値は“ RUN : 運転/TERM : 運転 ”です。

< キースイッチの位置による CPU の動作 >

システム定義の設定	動作	
	RUN	TERM
RUN: 運転/TERM: 運転	運転	運転
RUN: 運転/TERM: 前回状態	運転	前回状態 注)
RUN: 停止/TERM: 停止	停止	停止

注) 前回状態とは、システムの電源が切断される前の CPU の状態です。CPU が運転していたら運転となり、停止していたら停止となります。

バッテリーレス運転にした場合、CPU モジュールは CPU 運転開始時に常時メモリの初期化を行ない、運転を開始します。また、データのバックアップ異常は検出しません。

デフォルト値はバッテリーレス運転“ しない ”となっています。



第7章 オンライン機能

7 - 1	リレー表示	7 - 1
7 - 2	レジスタ表示	7 - 2
7 - 3	トレンドグラフ	7 - 3
7 - 3 - 1	トレンドグラフ	7 - 3
7 - 3 - 2	トレンドメニュー	7 - 3
7 - 3 - 3	レジスタ表示項目編集	7 - 5
7 - 3 - 4	リレー表示項目編集	7 - 6



第7章 オンライン機能


7 - 1 リレー表示


	FEDC	BA98	7654	3210	(HEX)	(DEC)
B00000	○ ○ ○ ○	○ ○ ○ ○	○ ○ ○ ○	○ ○ ○ ○	0H	0
B00010	○ ○ ○ ○	○ ○ ○ ○	● ○ ○ ○	● ○ ○ ○	44H	68

入力域（一番左のグリッド）にリレー名を入力して下さい。

入力は〔Enter〕キーで次の行に移行し、+16されたリレー名が自動設定されます。

16点単位で表示しますのでリレー名は最終桁の設定はできません。リレー名を変更したいときは〔F2〕キーを押してください。

（赤色）はリレーオン状態を示します。

（白色）はリレーオフ状態を示します。

注）表示するデータが増えると表示のリフレッシュが遅くなります。

メニューバー

「ファイル」

「リレー表示終了」

リレー表示を閉じます。

「編集」

「リレー表示1行削除」

リレー表示を1行削除します。カーソル行の設定を無効とし、1行下段以降を繰り上げます。最終行の場合は設定クリアのみです。

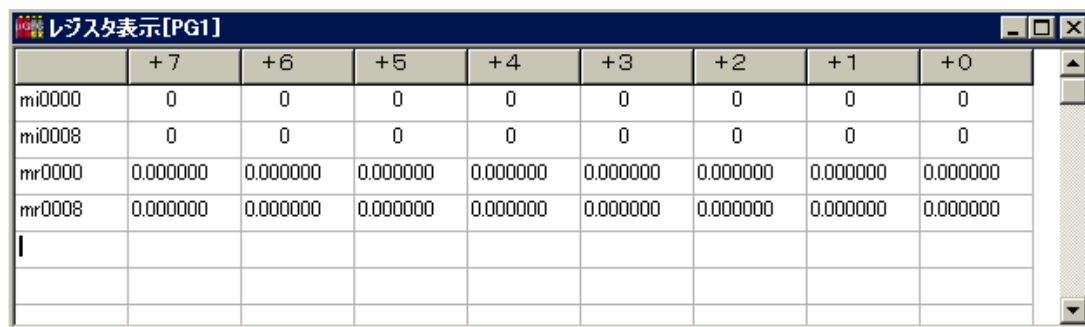
「接点 ON/OFF」

接点を ON/OFF します。チェックが付いている状態にて、該当データをクリックすることにより ON/OFF できます。

（青色）OFF から ON にするリレーを選択している事を示します。

（緑色）ON から OFF にするリレーを選択している事を示します。

7 - 2 レジスタ表示



	+7	+6	+5	+4	+3	+2	+1	+0
mi0000	0	0	0	0	0	0	0	0
mi0008	0	0	0	0	0	0	0	0
mr0000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
mr0008	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

入力域（一番左のグリッド）にレジスタ名を入力して下さい。入力は〔Enter〕キーで次の行に移行し、レジスタ名は+8 データされたレジスタ名が自動設定されますが、変更は自由に行なえます。データ表示は 1 行に 8 データを表示します。レジスタ名を変更したいときは〔F2〕キーを押してください。

注) 表示するデータが増えると表示のリフレッシュが遅くなります。

メニューバー

「ファイル」

「16 進表示」

整数データの 10 進表示、16 進表示の切り替えを行ないます。メニューがチェックされている状態が 16 進表示時です。

「レジスタ表示終了」

レジスタ表示を閉じます。

「編集」

「レジスタ表示 1 行削除」

レジスタ表示を 1 行削除します。カーソル行の設定を無効とし、1 行下段以降を繰り上げます。最終行の場合は設定クリアのみです。

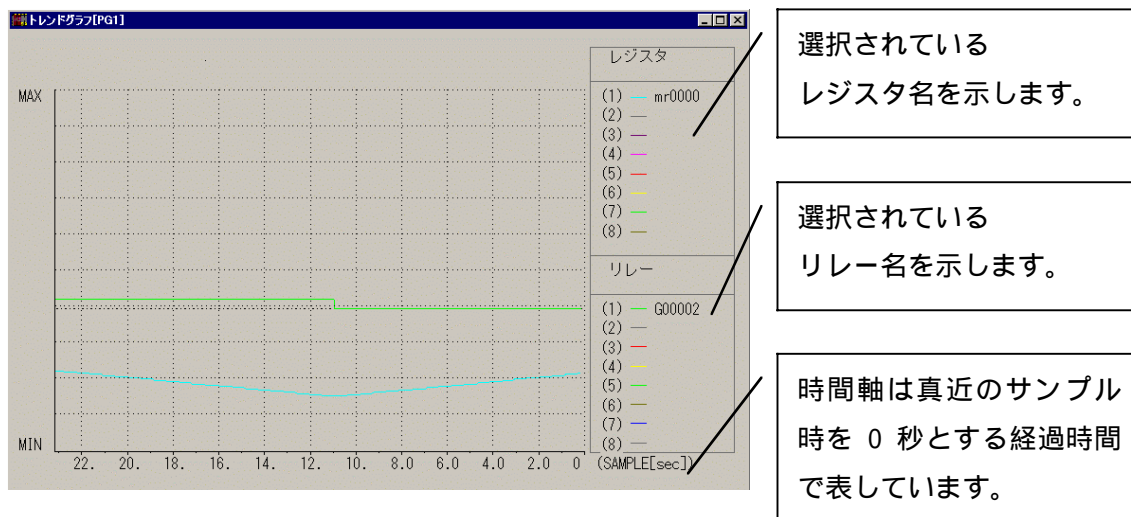
「データ変更」

チェックが付いた状態で、右クリックした位置のレジスタの値を変更し、 μ GPCsx に送信します。

7 - 3 トレンドグラフ

7 - 3 - 1 トレンドグラフ

ツールバーもしくはプロジェクトツリーの上で右クリックによりトレンドグラフを選択すると下図のトレンドウィンドウが開かれ、選択したレジスタやリレーのトレンドグラフがリアルタイムに表示されます。



トレンドウィンドウ上でマウスを右クリックすると下図のメニューが表示されます。

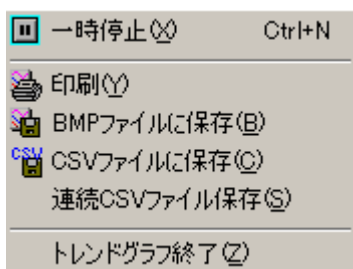
一時停止 (N)	Ctrl+N
罫線表示 (W)	
レジスタ表示項目の編集 (R)	
リレー表示項目の編集 (Y)	
トレンドグラフ終了 (Z)	

トレンドグラフを一時停止します。
 トレンドグラフに罫線を引きます。
 サンプルするレジスタを変更します。
 サンプルするリレーを変更します。
 トレンドグラフを終了します。

7 - 3 - 2 トレンドメニュー

トレンドメニュー（ツールバー）では下記機能が可能です。

「ファイル」メニュー



「一時停止」

トレンドグラフを一時停止します。

「印刷」

トレンドグラフを印刷します。

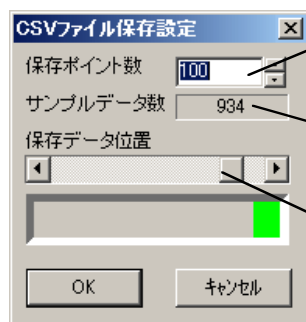
「BMP ファイルに保存」

トレンドグラフをビットマップファイルに保存します。

「CSV ファイルに保存」

トレンドグラフでサンプルしたデータを CSV ファイルに保存します。

ファイル名を指定した後下記ダイアログボックスが表示されますので保存パラメータを設定し保存してください。



保存するデータ数を指定します。

サンプルしたデータ数を示します。

サンプルしたデータの中から保存する箇所を指定します。サンプルした全体からのデータ位置がボ-グラフに示されます。

「連続 CSV ファイル保存」

トレンドグラフが動作している間、指定されたサンプル数毎に CSV ファイルを連続生成します。

ファイル名

(指定されたファイル名) 1.CSV、(指定されたファイル名) 2.CSV・・・

「トレンドグラフの終了」

トレンドグラフを終了します。

「編集」

- 「レジスタ表示項目の編集」
- 「リレー表示項目の編集」
- 「サンプリングタイムの編集」

「表示」



「罫線表示」

ツールバー説明

サンプリング間隔を設定します。設定範囲は 100ms ~ 10000ms (10s) までです。

Sampling Time [ms]

7 - 3 - 3 レジスタ表示項目編集

トレンドグラフのサンプルするレジスタの設定を変更します。

The screenshot shows a dialog box titled 'レジスタ表示項目編集' (Register Display Item Edit) with the following fields and controls:

レジスタ名	最大値	最小値	コメント	サンプリング無効
01	20	0		<input type="checkbox"/>
02	0	0		<input checked="" type="checkbox"/>
03	0	0		<input checked="" type="checkbox"/>
04	0	0		<input checked="" type="checkbox"/>
05	0	0		<input checked="" type="checkbox"/>
06	0	0		<input checked="" type="checkbox"/>
07	0	0		<input checked="" type="checkbox"/>
08	0	0		<input checked="" type="checkbox"/>

Buttons at the bottom: OK, キャンセル, 適用

Callout boxes and their descriptions:

- 「グラフの色を変更できます。」 (Graph color can be changed.) - Points to the color selection dropdowns.
- 「レジスタ名の属性 (2文字) を指定して下さい。」 (Specify the attribute (2 characters) of the register name.) - Points to the dropdown menu next to the register name.
- 「レジスタのオフセット部を指定して下さい。」 (Specify the offset part of the register.) - Points to the numerical input field for the register name.
- 「表示するグラフの最小値と最大値を指定して下さい。」 (Specify the minimum and maximum values of the graph to be displayed.) - Points to the '最大値' and '最小値' input fields.
- 「一時的にサンプリングを無効にできます。」 (Sampling can be temporarily disabled.) - Points to the 'サンプリング無効' checkboxes.
- 「設定を有効としダイアログを閉じます。」 (Enable settings and close the dialog.) - Points to the 'OK' button.
- 「設定を無効としダイアログを閉じます。」 (Disable settings and close the dialog.) - Points to the 'キャンセル' button.
- 「ダイアログを閉じずに設定を適用します。」 (Apply settings without closing the dialog.) - Points to the '適用' button.

7 - 3 - 4 リレー表示項目編集

トレンドグラフのサンプルするリレーの設定を変更します。

The screenshot shows a dialog box titled "リレー表示項目編集" (Relay Display Item Edit). It contains a table with columns for "リレー名" (Relay Name) and "コメント" (Comment). The first row shows a relay named "0002" with a green icon. To the right of the table are several checkboxes, all labeled "サンプリング無効" (Sampling Ineffective). At the bottom are three buttons: "OK", "キャンセル" (Cancel), and "適用" (Apply).

Callout boxes provide the following instructions:

- グラフの色を変更します。 (Change the graph color.)
- リレー名の属性 (2文字) を指定して下さい。 (Specify the relay name attribute (2 characters).)
- リレーのオフセット部を指定して下さい。 (Specify the relay offset part.)
- 設定を有効としダイアログを閉じます。 (Set the setting to effective and close the dialog.)
- 設定を無効としダイアログを閉じます。 (Set the setting to ineffective and close the dialog.)
- 一時的にサンプリングを無効にできます。 (Sampling can be temporarily ineffective.)
- ダイアログを閉じずに設定を適用します。 (Apply the setting without closing the dialog.)



第 8 章 印刷

8 - 1	印刷の概要	8 - 1
8 - 1 - 1	画面操作方法	8 - 1
8 - 1 - 2	プリンタ設定の確認と変更	8 - 2
8 - 1 - 3	図枠印字設定	8 - 2
8 - 2	個別印刷	8 - 3
8 - 2 - 1	[回路リスト]印刷	8 - 3
8 - 2 - 2	[システム定義]印刷	8 - 4
8 - 2 - 3	[パラメータ]印刷	8 - 4
8 - 2 - 4	[クロスリファレンス]印刷	8 - 5
8 - 2 - 5	[接点コメント]印刷	8 - 5
8 - 2 - 6	[その他]印刷	8 - 5



第 8 章 印刷

8 - 1 印刷の概要

TDsxEditor の印刷機能について以下に示します。

- ・ 回路リスト印刷
- ・ システム定義印刷
- ・ パラメータ印刷
- ・ クロスリファレンス印刷
- ・ 接点コメント印刷
- ・ その他印刷

8 - 1 - 1 画面操作方法

TDsxEditor の[ファイル(E)]メニューの[印刷(P)]を選択すると以下のような印刷詳細画面が表示されます。

[印字しない]を押すと、選択した項目を無効にし選択できません。また[印字実行[Enter]]ボタンが押せなくなります。

[印字する]を押すと、その中身が選択できます。また[印字実行[Enter]]ボタンが押せるようになります。

設定した各項目を有効にして印刷します。各項目のどれか 1 つを[印字する]にした時に押せるようになります。

設定した各項目を無効にして閉じます。

各項目を印刷する時に図枠をつけて印刷するか選択します。

「プリンタの設定」ダイアログを表示します。

設定した各項目を有効にして印刷します。各項目のどれか 1 つを[印字する]にした時に押せるようになります。

設定した各項目を無効にして閉じます。

設定した各項目を有効にして印刷します。各項目のどれか 1 つを[印字する]にした時に押せるようになります。

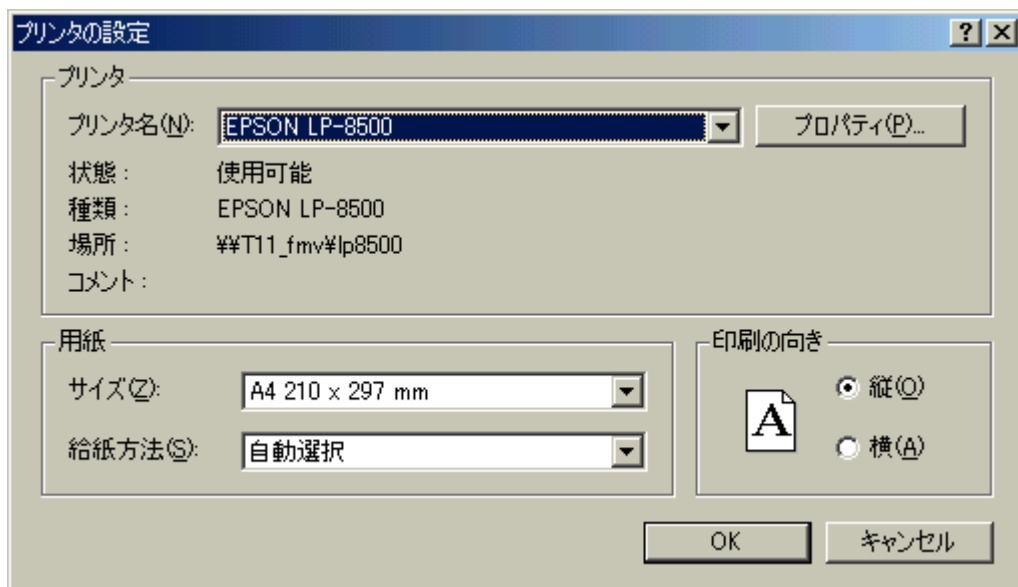
設定した各項目を無効にして閉じます。

8 - 1 - 2 プリンタ設定の確認と変更

使用する標準プリンタを選択します。

< プリンタを選択する手順 >

印刷設定画面上で[印刷設定]を押すと、[プリンタの設定]ダイアログが表示されます。



[プリンタ名(N):] リストボックスで使用するプリンタを選択します。

必要に応じて“用紙”、“印刷の向き”、“プロパティ”などを設定し、[OK]ボタンを左クリックします。

8 - 1 - 3 図枠印字設定

[印字する]を押して選択した各項目(回路リスト・システム定義・パラメータ・クロスリファレンス・接点コメント・その他(プロジェクトツリー・メモリ転送定義・トレースバック))を印刷するときに図枠をつけて印刷するか選択します。

[図枠印字]をクリックすると下のような項目が表示されます。



図枠印字しない 印刷するときに図枠を一緒に印刷しません。

デフォルト デフォルトで設定されている図枠ファイルをつけて印刷します。

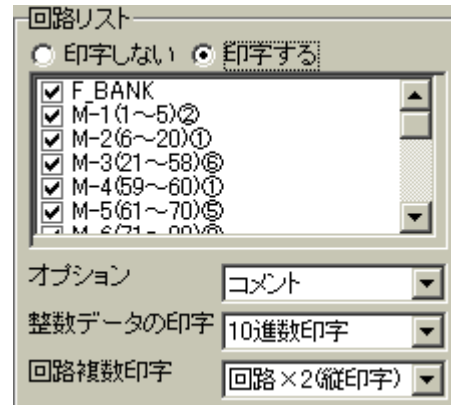
8 - 2 個別印刷

[回路リスト]や[システム定義]など印字したい項目を選択して印刷することができます。

8 - 2 - 1 [回路リスト]印刷

選択したプログラムの回路を印刷します。

プロジェクトで作成したタスク 1、
タスク 2、サブルーチンの各サブプログラムが表示されます。



回路を印刷する時に以下の機能を付加します。

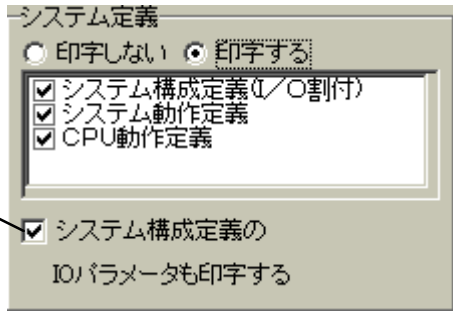
オプション	[コメント]	回路リストをそのまま印刷します。
	[クロスリファレンス]	接点の下にストアされたコイルのクロスリファレンスを印刷します。
	[スペース]	接点の下に何も印刷しません。
回路複数印字	[回路×1[横印字]]	印刷方向を横にして1回路分を印刷します。
	[回路×2[縦印字]]	印刷方向を縦にして2回路分を印刷します。
整数データの印字	[10進数印字]	回路内の整数データを10進数で印字します。
	[16進数印字]	回路内の整数データを16進数で印字します。

注) 図枠印字をする場合は横印字しかできません。

8 - 2 - 2 [システム定義]印刷

全体システムの構成、動作設定などを印刷します。

デフォルトではチェックされた状態になっています。そのままと[システム構成定義(I/O 割付)]を印刷するときに各モジュールのパラメータも一緒に印刷します。



システム構成定義(I/O 割付) システム構成定義で設定したシステム構成ツリーの内容を印刷します。

システム動作定義 システム動作定義で設定したパラメータを印刷します。

CPU 動作定義 CPU 動作定義で設定したパラメータを印刷します。

8 - 2 - 3 [パラメータ]印刷

選択したプログラムの使用点数およびパラメータの値を印刷します。

プロジェクトで作成したタスク 1、タスク 2、サブルーチンの各サブプログラムが表示されます。



パラメータ使用点数 プログラム内で使用しているパラメータの使用点数を印刷します。

パラメータの値 プログラム内で使用しているパラメータ(ki、kr、TS、TD、NP)の値を印刷します。

8 - 2 - 4 [クロスリファレンス]印刷

選択したプログラムのクロスリファレンスを印刷します。

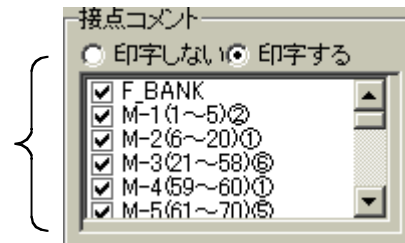
プロジェクトで作成したタスク 1、
タスク 2、サブルーチンの各サブ
プログラムが表示されます。



8 - 2 - 5 [接点コメント]印刷

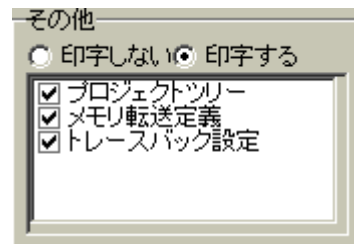
選択したプログラムの接点コメントを印刷します。

プロジェクトで作成したタスク 1、
タスク 2、サブルーチンの各サブ
プログラムが表示されます。



8 - 2 - 6 [その他]印刷

選択した項目を印刷します。



<プロジェクトツリー>

プロジェクトツリーを印刷します。

<メモリ転送定義>

タスク 1、2のメモリ転送定義を印刷します。

<トレースバック設定>

タスク 1、2のトレースバック設定の内容を印刷します。



第9章 環境設定、書き込みモード カスタマイズ

9 - 1	環境設定	9 - 1
9 - 1 - 1	色の設定	9 - 1
9 - 1 - 2	ツール設定	9 - 2
9 - 1 - 3	接続先通信設定	9 - 3
9 - 1 - 4	USB ドライバのセットアップ	9 - 4
9 - 2	書き込みモードカスタマイズ.....	9 - 6

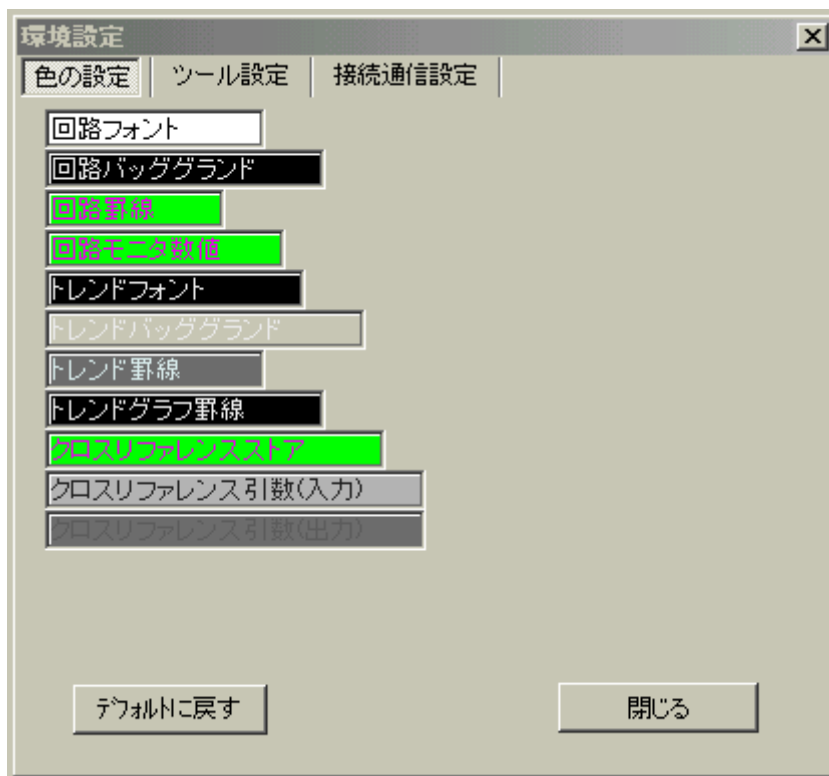


9章 環境設定、書き込みモードカスタマイズ

9 - 1 環境設定

9 - 1 - 1 色の設定

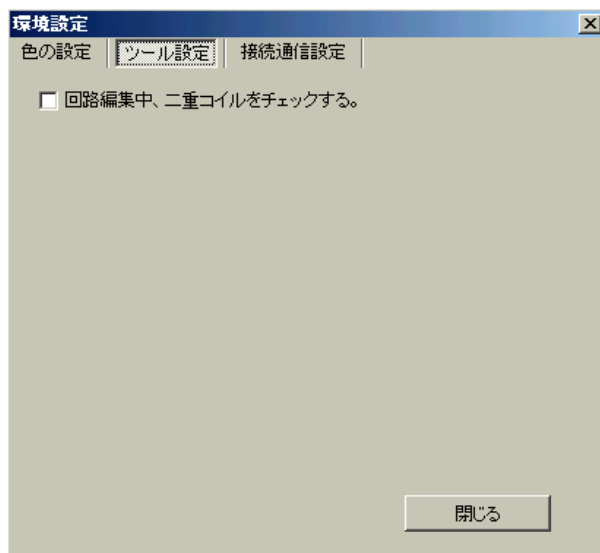
色の設定では画面の色が変更できます。変更したい項目を右クリックすることにより、色の設定ダイアログボックスが表示されますので変更したい色を選んでください。





9 - 1 - 2 ツール設定

ツール設定では TDsxEditor の各種設定を行います。



9 - 1 - 3 接続先通信設定

μGPCsx と TDsxEditor を接続する手段を決定します。

・COM ポート

PC に装備されている COM ポート (シリアルポート) を使用する場合に指定します。

・イーサネット

μGPCsx の PC カードモジュールにイーサネットカードを挿入し、イーサネット経由で TDsxEditor を使用できます。設定は μGPCsx (PC カードモジュール) の IP アドレス、ポートを指定して下さい。

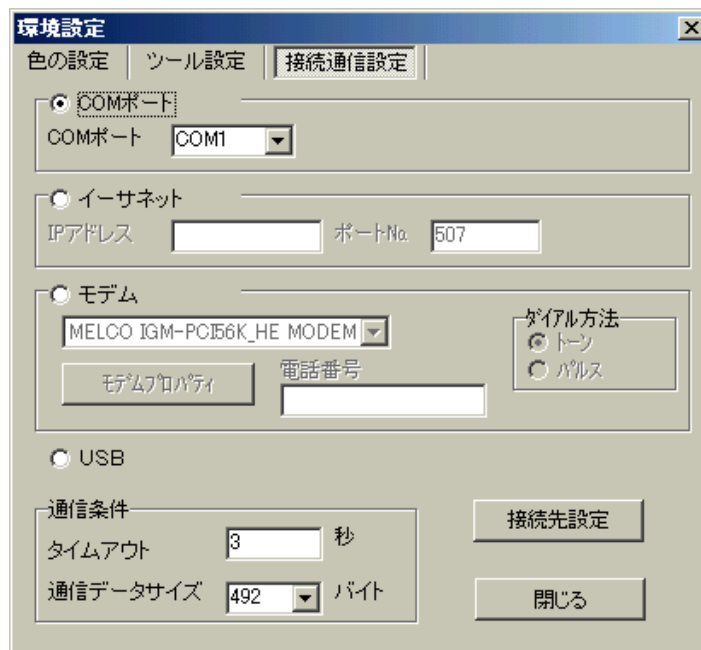
・モデム

μGPCsx の PC カードモジュールにモデムカードを挿入し、電話回線経由で TDsxEditor を使用できます。設定は μGPCsx (PC カードモジュール) の接続されている回線の電話番号、使用するモデムと PC (パソコン) に接続されている回線のダイヤル方法を指定します。

・USB

R タイプ CPU (TD1PS-117R) の USB コネクタと PC を USB にて接続します。

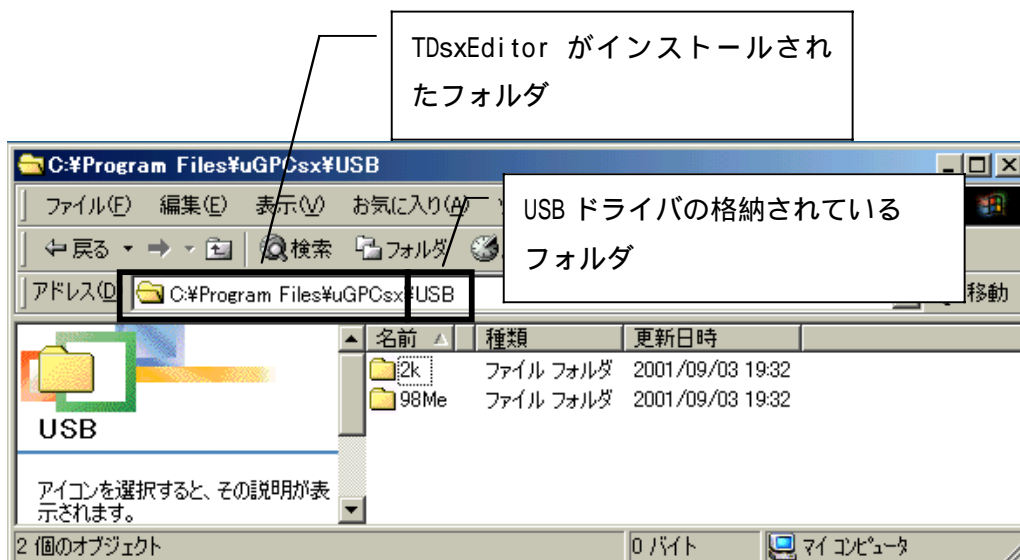
USB を使用するには USB ドライバをセットアップする必要があります。



9 - 1 - 4 USB ドライバのセットアップ

USB 接続にて TDsxEditor と μGPCsx を接続するには、USB ドライバをセットアップする必要があります。

μGPCsx と PC (パソコン) を USB ケーブルで接続すると、Windows のドライバインストール画面が表示されますので OS により下記フォルダを指定して下さい。



- 「2k」 Windows2000 用のドライバです。
- 「98Me」 Windows98/Me 用のドライバです。

9 - 2 書き込みモードカスタマイズ

書き込みモードカスタマイズでは回路ウィンドウの書き込みモードに於いてのショートカットキー（アクセラレータキー）が指定できます。

ショートカットキーを設定したい「書込モード機能」を選択した後、「ショートカットキー：設定」にてショートカットキーを選択します。

