

システム設計の頼れるヒーロー
教えて!GX3号



ホームページでGX Works3の便利機能を紹介!
これからも色々な便利機能を紹介していくよ
二次元コードをスキャンしてGX3号のページへ



システム設計の頼れるヒーロー 教えて!GX3号

第1回

NEW
Ver.Up
新機能

登場!GX3号



GX Works3ってどんな製品なんだろう?
誰か教えてくれないかなあ。



ヒーロー一見参!!

僕はGX Works3を皆に普及
させたいと願うGX3号!



僕がラダーエディタの便利機能を
教えてあげるよ!



キー入力機能の強化や分割表示、フロー
ティング表示機能により、作業効率が向
上し短時間でプログラミングができるん
だ!



便利そうな機能がたくさんあるね。
よろしく、GX3号!



次回の教えて!GX3号は、
プログラミングエディタでの『新しいキー入力』について!
君も今日からシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第2回



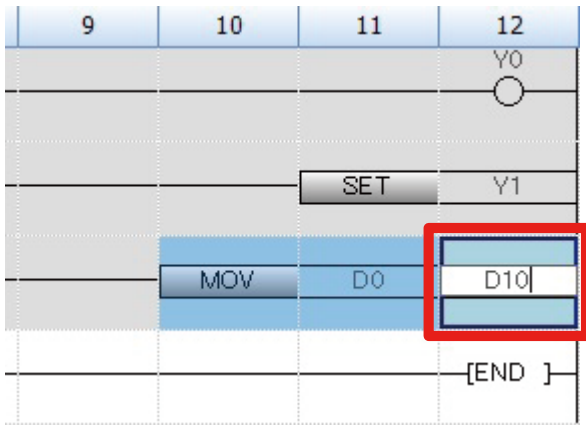
素早くデバイスを編集する!!



デバイスNo.やコメントを編集する時
いちいち右クリックでメニューを
選ぶのは手間だなあ…。

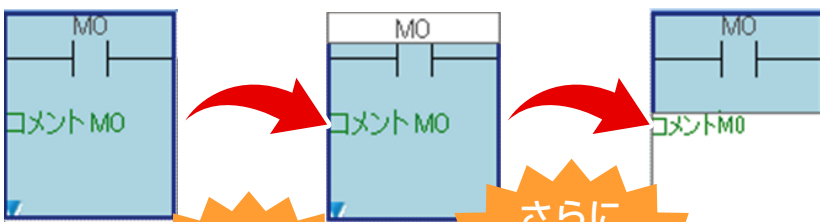
セルを選んで「F2」キーを
押すだけでデバイス入力欄を
編集状態にできるんだ!

そんなときは



F2
を押す!

コメント編集も
「F2」でできる!



F2
を押す!

さらに
F2
を押す!

ちなみにデバイス No.
はAlt+ ↑ ↓でも変更
できるんだ!



すごい!
これで簡単に編集できるね!

次回の教えて!GX3号は、

キー操作でカンタンに『回路記号反転』について!

君も今日からシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第3回



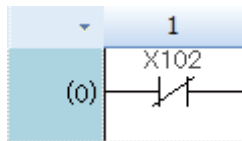
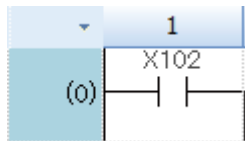
ワンタッチでの回路記号反転!



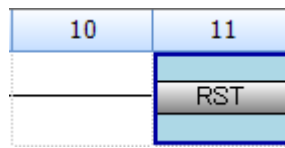
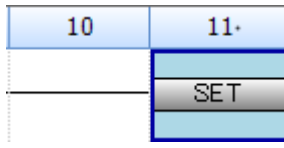
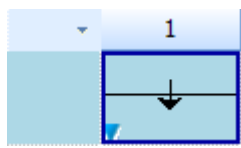
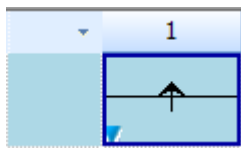
回路記号反転する時いちいち差し替えるのって手間だなあ…。



" / " キーを使えば回路記号 (a/b 接点) が反転できるよ!



さらに **Alt+ /** キーで回路記号 (立ち上がり / 下がり、SET/RST) が反転できるんだ!



おお! " / " キーを覚えるだけでサクサク編集ができるようになったよ!

次回の教えて!GX3号は、

プログラムの流れを一目で把握について!

君も今日からシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号 第4回



プログラムの流れを一目で把握

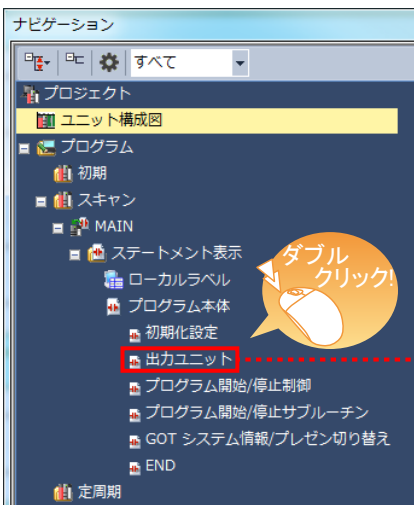


プログラムが長いとラダーのどの部分で
どんな制御をしているか把握しづらいなあ…。

行間ステートメントを設定し、ナビゲーションツリーに
表示することで処理の流れが見えるんだ！



おお！ラダー図を読まなくても
ナビゲーションツリーを見るだけで、
プログラムの流れがわかるね！



ナビゲーションツリーからダブルクリックで、
そのステートメント部分をラダーエディタに
表示することもできるよ！



書込み	1	2	3	4	5	6
+	[Title]出力ユニット					
2	Output unit 輸出单元					
3	(65)	SM402				
4	(102)	SM410	M1030			
5	(107)					



誰が作ったプログラム
でも把握しやすいね！

次回の教えて!GX3号は、
ラダーエディタの見やすい表示機能について！
君も今日からシーケンスアープ！





システム設計の頼れるヒーロー 教えて!GX3号

第5回



ラダーエディタを分割表示に!



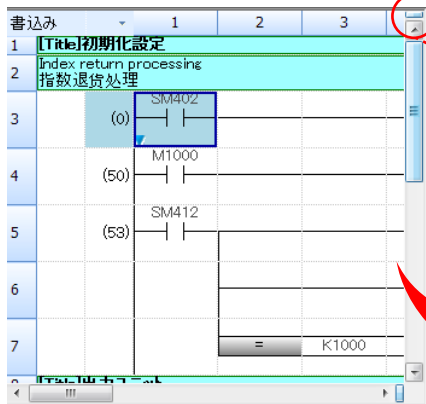
長いプログラムだと何度もスクロールしないと
見たい場所が見れないなあ…。

画面は分割できるんだ!
上下左右どちらも対応してるよ!



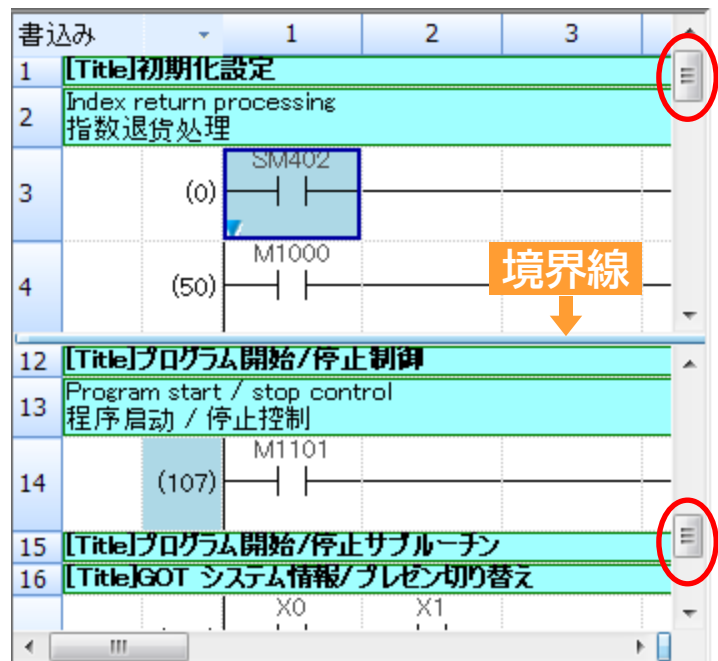
これなら見たい箇所を同時に見れて便利だね!

個別のスクロール



ドラック!!

上下分割



境界線

COMMENT [デバイスコメント]			
デバイス名(N)	D0	デバイス名(N)	X0
+	D0	データ0	X0
+	D1	データ17	X1
+	D2	データ34	X2
+	D3	データ51	X3
+	D4	データ68	X4

ちなみにデバイスコメントエディタ
も左右に分割できるよ! 別々のデバ
イスを同時に表示できるんだ!



次回の教えて!GX3号は、

プログラムの連続貼り付けについて!

君も今日からシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第6回



必要な部分の回路を連続貼付!



同じような回路の部分、
いちいち作るのめんどろなあ...

必要な部分をコピーして
連続貼付できる機能があるよ!



貼付個数、貼付方向、インクリメント数
(複製デバイスの増加数)を決めて連続して貼付!



似たような回路をたくさん作る
のに便利だね!部分的に使える
から使う機会も多そう!

書込み		1	2	3
		M10	M11	M12
1	(0)			
2		X1	X2	X3
3	(5)			



書込み		1	2	3
		M10	M11	M12
1	(0)			
2		X1	X2	X3
3		X2	X3	X4
4		X3	X4	X5
5		X4	X5	X6
6	(5)			

右クリックでも使えるし、
ショートカットは **Ctrl+Alt+V** だよ!
上手に活用しよう!



次回の教えて!GX3号は、

1クリックでの回路の一時変更について!

君も今日からシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第7回



1クリックで回路を一時変更する!



プログラムを一部変更して動作確認したいけど、デバッグ用に作成すると忘れてたりトラブルの原因になりそう…。

1クリックで一時変更できる機能があるよ!
変更してみたい部分をコピーして jmp 命令、
ステートメントも自動で作成してくれるよ!



回路一時変更
変更を元に戻す
変更した回路を適用



The screenshot shows a logic editor interface. A green-shaded area from line 4 to 8 is labeled '一時変更対象の回路ブロック' (Circuit block for temporary modification). This area contains an '自動ステートメント' (Automatic statement) block with a 'jmp' command. A red box highlights the 'jmp' command and the text 'jmp命令より綠色部分のラダー無効化' (Disable ladder in green area due to jmp command). The 'jmp' command is set to 'P0'. Below the green area, the original circuit logic is visible, including 'MOV D0 D10' and 'INC D11' instructions. The temporary modification is bounded by '[TemporaryStart]' at line 4 and '[TemporaryEnd]' at line 11. A red arrow points to the 'jmp' command in the temporary block.

ラダー図の色とステートメントで一時変更した箇所がすぐわかるし、一時回路をたくさん作っても行間ステートメント一覧で一時変更回路のみをフィルタできるよ!



変更箇所がわかりやすいし、
一時変更を忘れることなく管理できるね!
元に戻したいときも1クリックで簡単!

The dialog box '行間ステートメント一覧 [ProgPou]' has a search field and several filter options. The '回路一時変更時に自動挿入されたステートメント(C)' (Statements automatically inserted during temporary modification) option is selected. The list shows several temporary modification blocks with steps like '[TemporaryStart]', '[TemporaryModify]', and '[TemporaryEnd]'. A '開じる' (Open) button is at the bottom.

これでラダーの変更が多く
なっても混乱しない!
これで君も
シーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第8回



GX Works2形式のプロジェクトを開く!

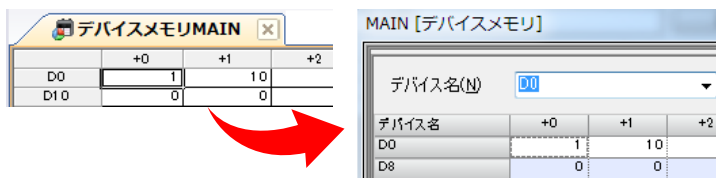


GX Works2 で作った昔のデータを
GX Works3 でも使いたいなあ…。

GX Works3 は GX Works2 で作ったプロジェクトも
開くことができるよ! デバイスメモリ、ラベル等の
設定も引き継ぐことができるんだ!



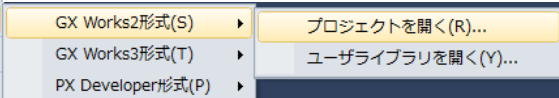
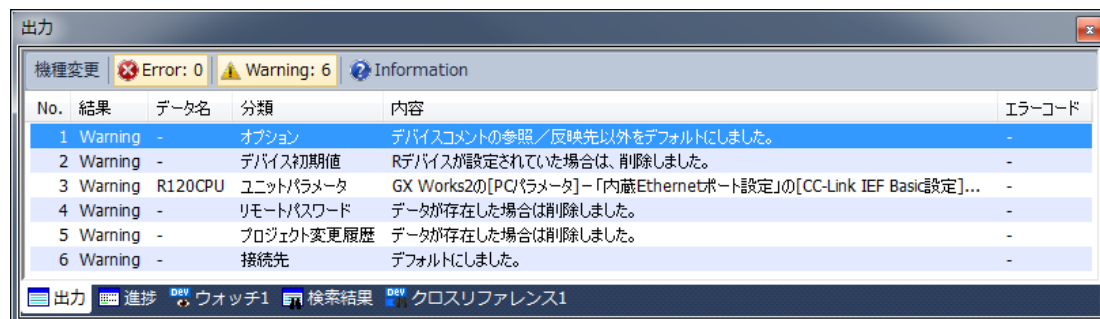
GX Works2からGX Works3へ引き継ぎ



開き方は "他形式ファイルを開く" を選択する
だけで簡単だし、引き継いだ際の変更点も開いた
時に出カウィンドウで教えてくれるよ!



今までのプロジェクトが使えるか心配
だったけど、これなら安心だね!



PX Developer 形式のデータも同じ方法で開くことが
できるよ! GX Works3 を使えば既存のデータもより
便利に扱える! これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第9回

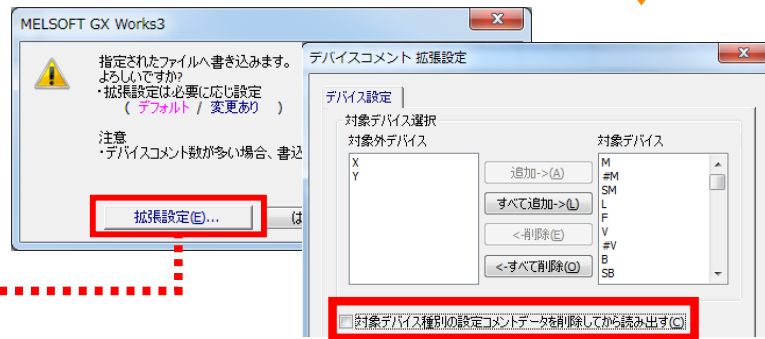
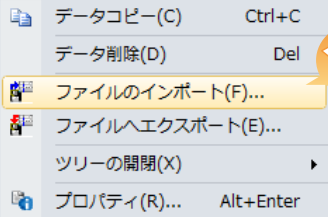
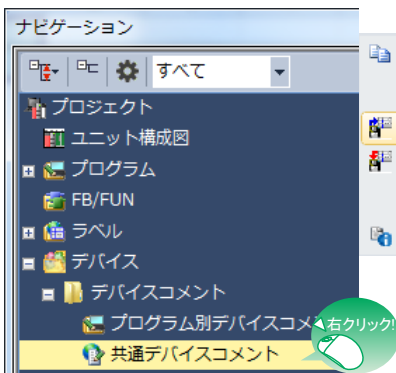


デバイスコメントのインポート・エクスポート

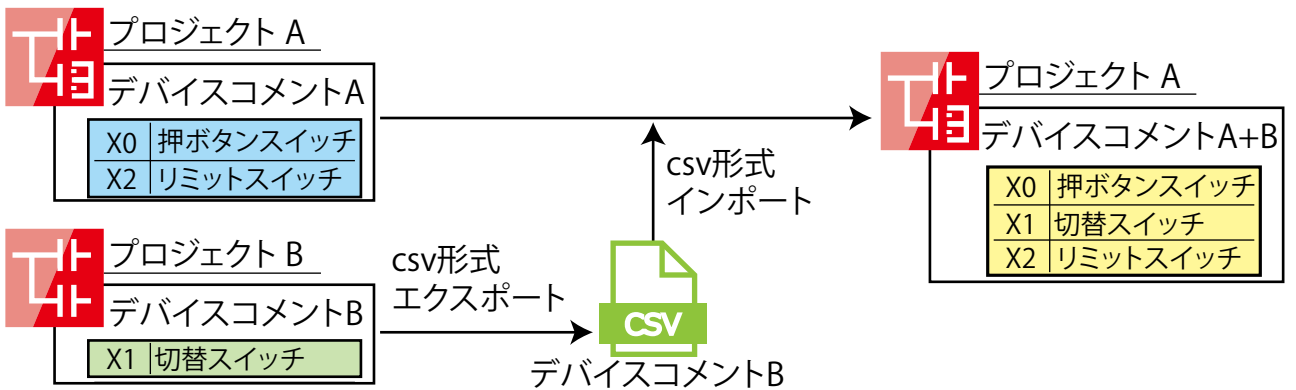


他のプロジェクトデータのコメントも
今のプロジェクトで使いたいなあ。

ナビゲーションウィンドウのデバイスコメントから
右クリックでデバイスコメントをインポート・エクス
ポートができるよ! (GX Works2 形式でも OK)



拡張設定にて細かな設定が可能だよ!
インポート時に“対象デバイス種別のコメント
データを削除してから読み出す”のチェックを
外せばデバイスコメントをマージできるんだ!



これで他のプロジェクトデータの
デバイスコメントを取り込めるね!

過去の資産も GX Works3 で上手に使っていきたいね!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

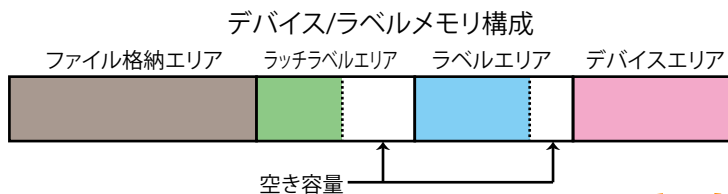
第10回



ラベル容量の使用状況確認!



デバイスエリアの容量をもっと増やしたいなあ。



ラベルエリアの使用状況は変換時に出カウィンドウにて確認できるよ!
空き容量をデバイスエリア、ファイル格納エリアに割り当てて容量を調整しよう!



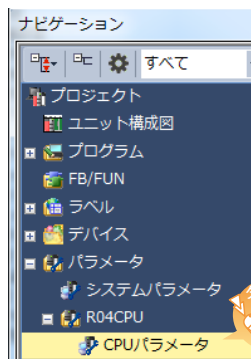
出力

変換 Error:0 Warning:0 Information

No.	結果	データ名	分類	内容	エラーコード
1	Information	ラベル	空き容量	96.41[%] (29616 [Word] = 30720 [Word] - (Global: 420 [Word] + Local: 684 [Word]))	-
2	Information	ラッチラベル	空き容量	83.01[%] (1700 [Word] = 2048 [Word] - (Global: 176 [Word] + Local: 172 [Word]))	-



ラベルエリアの空き容量が細かくわかるんだね!
これでラベルエリアに必要な設定値がわかったよ!



R04CPU CPUパラメータ

設定項目一覧	設定項目	項目	設定
メモリ/デバイス設定	デバイス/ラベルメモリエリア設定	拡張SRAMカセット設定	装着なし
	デバイス/ラベルメモリエリア容量設定	デバイスエリア	デバイスエリア容量
	デバイス/ラベルメモリエリア設定	ラベルエリア	ラベルエリア容量
	デバイス/ラベルメモリエリア設定	ラッチラベルエリア	ラッチラベルエリア容量
	デバイス/ラベルメモリエリア設定	ファイル格納エリア	ファイル格納エリア容量
	デバイス/ラベルメモリ構成の確認	デバイス/ラベルメモリ構成の確認	<確認>
	デバイス/ラベルメモリエリア詳細設定	デバイス/ラベルメモリエリア詳細設定	

ナビゲーションウィンドウのCPUパラメータからメモリ/デバイス設定にて容量の調節ができるよ!ラベルエリアやデバイスエリアを適切な設定値に設定しよう!



各エリア容量を適切な設定値に増減して上手に使おう!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第11回



未使用ラベルの削除機能



回路編集をした後、未使用のラベルがたまるとな。データも大きくなるし整理したいよ…。

検索/置換(F)		
クロスリファレンス(S)		Ctrl+E
デバイス使用一覧(I)	クリック!	Ctrl+D
未使用ラベル一覧(O)	クリック!	
デバイス/ラベル検索(D)		Ctrl+F

クロスリファレンス1					
(未使用ラベル)		(プロジェクト全体)			
デバイス/ラベル	デバイス	回路記号	位置	プログラムファイル	
フィルタ条件	フィルタ条件	フィルタ条件	フィルタ条件	フィルタ条件	
B udLabel5			3行1列		
B dLabel3			1行1列	MAIN	ProgPou
B eLabel4			2行1列	MAIN	ProgPou

検索場所は“プロジェクト全体”を選んでね!

メニューバーの“検索 / 置換”→“未使用ラベル一覧”にて未使用ラベルを一覧化できるよ!
クロスリファレンス (Ctrl+E) から未使用ラベルを選択でも可能だよ!



これで使っていないラベルが簡単にわかるね!

クロスリファレンス1						
(未使用ラベル)		(プロジェクト全体)			表示(V) オプション(O)...	
デバイス/ラベル	デバイス	回路記号	位置	プログラムファイル名	データ名	コメント
フィルタ条件	フィルタ条件	フィルタ条件	フィルタ条件	フィルタ条件	フィルタ条件	フィルタ条件
B udLabel5			3行1列	Global		
B dLabel3			1行1列	MAIN	ProgPou	
B eLabel4			2行1列	MAIN	ProgPou	

ラベルエディタにジャンプしなくてもこの画面から直接削除が可能だよ!
それに複数のラベルをまとめて削除できるよ!



グローバル / ローカルラベルもまとめて表示できるのも便利だね!

ラベルの整理が簡単にできるね!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号 第12回

NEW

Ver.Up
新機能

検索した場所へ簡単にジャンプ



デバイスを検索してプログラムをたどっていくと元いた場所がわからなくなるなあ…。

クロスリファレンス、検索した場所に簡単操作でジャンプできるようになったよ！元いた場所がもう一度見たい時に確認できるんだ！

そんなときは



検索①

検索②

検索③

ツールバー
プログラム共通
or
Alt + Shift + ←

Alt + Shift + →



再度検索した場所にジャンプできるのか！
検索の前後を簡単に移動できるね！

キー操作は "Alt+Shift+←,→" ! ぜひ覚えておこう!
この機能は GX Works3 の最新バージョンにて試してみてね!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第13回

NEW

Ver.Up
新機能

ラダーエディタを好みの色に



プログラムを書くときは背景黒で作成するのが好み
なんだけど GX Works3 でも同じように使えないかな？

表示メニューの色設定から自由に変更できるよ！



表示(V)

- ツールバー(T)
- ステータスバー(B)
- 色およびフォント(C)...

各項目色を変更

- FB背景
- エディタ背景
- 命令背景
- 選択セル

グリッドを非表示



黒背景の設定が気に入ったよ！

ツール(T)

- オプション(O)...

オプション

- 動作設定
 - ラベルエディタ編集時の動作 同期させる
- 照合
 - 照合設定 詳細結果照合選択 照合結果をラダー図で表示
- グリッド
 - グリッドを表示する いいえ

セル毎に区切っている
グリッドも非表示に
できるよ！



好みに合わせて設定できるよ！
この機能は GX Works3 の最新バージョンにて試してみてね！
これで君もシーケンスアープ！





システム設計の頼れるヒーロー 教えて!GX3号

第14回

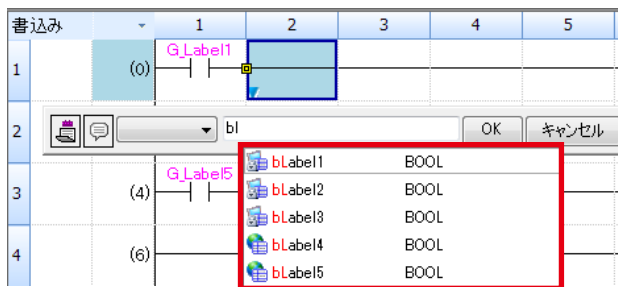
NEW

Ver.Up
新機能

ラベルを候補から簡単入力!



ラダーにラベルを使いたいけど全ての名前を覚えていないし、いちいち確認しないといけないのかなあ…。



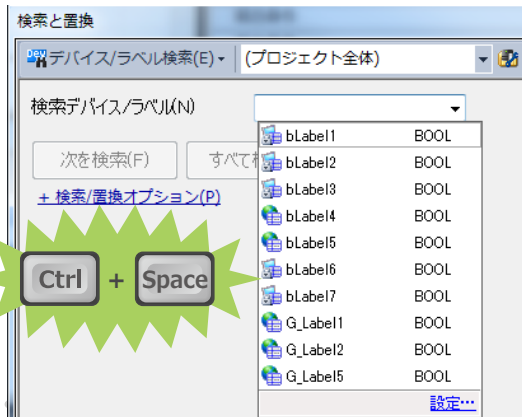
“先頭文字”を入力すると候補が一覧表示されるんだ!



他にも“**Ctrl+Space**”で全ラベルを候補表示できるよ!
どんなラベルがあったか把握できる!



どんなラベルがあったか覚えていなくても候補から選択できるね!



ウォッチ機能では“**プログラムブロック名 + 「/」**”でローカルラベルのみを表示できるから追跡したい時に使えるね!



ラベルの選択には候補表示をうまく使おう!
この機能はGX Works3の最新バージョンにて試してみてね!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第15回

NEW
Ver.Up
新機能

パラメータを引き継いでユニット変更

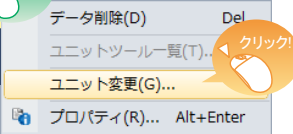


マルチ CPU のユニットを変更したいけど、ユニットを削除して新しいユニットを設定しないといけないの？

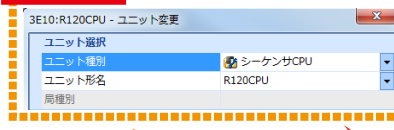
ユニット変更機能を使おう！
リフレッシュ設定を引き継いで変更できるよ！



右クリック!

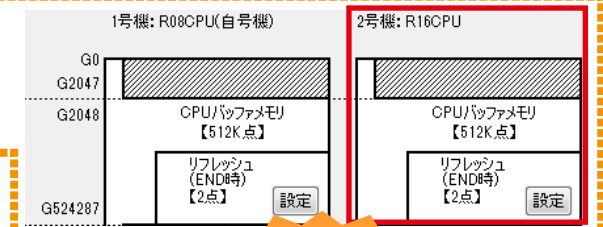
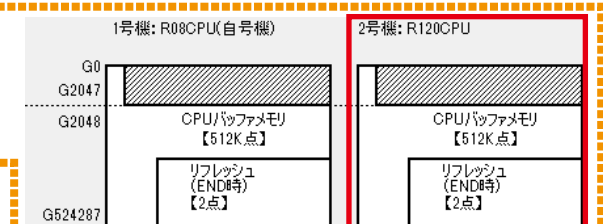
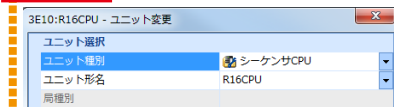


R120CPU



ユニット変更

R16CPU



リフレッシュ
設定引継ぎ

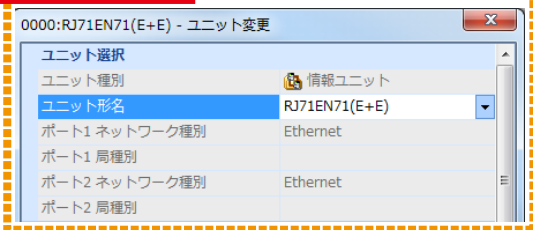


これでユニットを削除しなくても
変更できるね！

ユニット変更機能を使えば Ethernet ユニット (RJ71EN71) の
ポート 1 側 (Ethernet) のパラメータを保持したままポート 2 側
のネットワーク種別が変更できるよ！

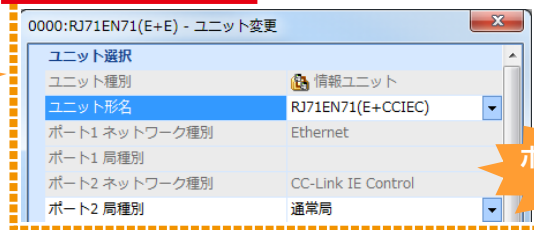


RJ71EN71(E+E)



ユニット変更

RJ71EN71(E+CCIEC)



ポート1設定
を引継ぎ

パラメータを保持できるからユニット変更しやすくなったね！
この機能は GX Works3 の最新バージョンにて試してみてね！
これで君もシーケンスアープ！





システム設計の頼れるヒーロー 教えて!GX3号

第16回

NEW

Ver.Up
新機能

指定した範囲内をまとめて置換



同じ内容のデバイスを流用したけどコメントを一つずつ直すのは面倒だなあ。



指定した範囲内をまとめて置換できるよ!

検索/置換(F)

デバイス/ラベル置換(R) Ctrl+H

命令置換(N)

文字列置換(P) Ctrl+Shift+H

クリック!

or

Ctrl + Shift + H

検索と置換

文字列置換(H) (プロジェクト全体)

検索文字列(N) A

置換文字列(V) B

次を検索(F) すべて検索(L) 置換(R) すべて置換(A)

クリック!

COMMENT [デバイスコメント]

デバイス名(N)	Japanese/日本語(表示対象)	English
X0	運転スイッチ_A	Operation switch_A
X1	コンパセンサ_A	Conveyor sensor_A
X2	モード切替_A	Mode switch_A
X3	コンパ逆転スイッチ_A	Conveyor reverse rotation switch_A
X4	運転スイッチ_A	Operation switch_A
X5	コンパセンサ_A	Conveyor sensor_A
X6	モード切替_A	Mode switch_A
X7	コンパ逆転スイッチ_A	Conveyor reverse rotation switch_A

ドラッグ!

置換

COMMENT [デバイスコメント]

デバイス名(N)	Japanese/日本語(表示対象)	English
X0	運転スイッチ_A	Operation switch_A
X1	コンパセンサ_A	Conveyor sensor_A
X2	モード切替_A	Mode switch_A
X3	コンパ逆転スイッチ_A	Conveyor reverse rotation switch_A
X4	運転スイッチ_B	Operation switch_B
X5	コンパセンサ_B	Conveyor sensor_B
X6	モード切替_B	Mode switch_B
X7	コンパ逆転スイッチ_B	Conveyor reverse rotation switch_B



これで流用した時の修正が簡単になったよ!

検索/置換(F)

デバイス/ラベル置換(R) Ctrl+H

命令置換(N)

文字列置換(P) Ctrl+Shift+H

クリック!

or

Ctrl + H

1	2	3	4
X102	X103		
X100			
X100	X101		
X100			

置換

1	2	3	4
X102	X103		
X100			
X105	X101		
X105			

検索と置換

デバイス/ラベル置換(C) (現在のウィンドウ)

検索デバイス/ラベル(N) X100

置換デバイス/ラベル(V) X105

次を検索(F) すべて検索(L) 置換(R) すべて置換(A)

クリック!

デバイスコメントだけでなくラダーエディタでも使えるんだ!



範囲指定しての置換を使って作業を効率化しよう!
この機能はGX Works3の最新バージョンにて試してみてね!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第17回

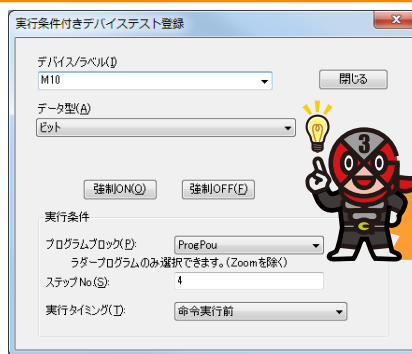
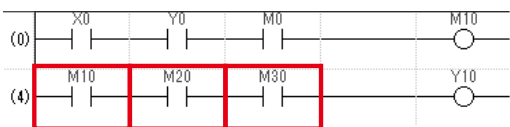
NEW
Ver.Up
新機能

指定したステップの動作を変更



装置でエラーが発生した時の動作を確認したいけど、他のステップの影響でうまく見れないなあ…。

実行条件付きデバイステスト機能を使おう！
これを登録しておけば自動的に条件に応じて値を変更してくれるよ！



他のステップに関係なくデバイス値を変更できる！

- デバッグ(B)
- シミュレーション(S)
- 現在値変更(M)... Shift+Enter
- 現在値変更履歴(L)...
- 強制入出力登録/解除(F)...
- 実行条件付きデバイステスト(E)**
- メモリダンプ(D)
- オフラインモニタ(O)
- プロセス制御拡張(P)

デバイスを
選択して登録

登録(R)... Ctrl+Enter
登録確認/解除(C)...

ショートカット
Ctrl + Enter

登録確認 / 解除画面とウォッチ画面でも確認できるんだ！

画面表示したまま
GX Works3操作可能

No.	選択	プログラムファイル	プログラムブロック	ステップNo.	デバイス/ラベル	割付デバイス	データ型	設定値	実行タイミング
1	<input type="checkbox"/>	MAIN	ProgPou	4	M10		ビット	ON	命令実行前
2	<input type="checkbox"/>	MAIN	ProgPou	5	M20		ビット	ON	命令実行前
3	<input type="checkbox"/>	MAIN	ProgPou	6	M30		ビット	ON	命令実行前

名称	現在値	データ型	実行条件付きデバイステスト
M10	TRUE	ビット	登録済
M20	TRUE	ビット	登録済
M30	TRUE	ビット	登録済

ウォッチ画面
から確認！

ウォッチに登録しておけば解除忘れ防止にもなるね！

プログラムの編集をしなくても動作を変更できるね！
これで君もシーケンスアープ！



システム設計の頼れるヒーロー 教えて!GX3号

第18回

NEW
Ver.Up
新機能

他形式ファイルの自動置き換え

GX Works3 新機能を紹介しますよ!
GX Works2 形式プロジェクトを開いた時に修正が
いらぬ命令が増えたんだ! (ST/FBD 言語)



たしか流用した後に命令や引数を変更しないと
変換エラーになることがあったよね?



GX Works2形式の
プロジェクトを開く!

```
MAIN [PRG] プログラム...
label_ENO := SIN(bLabel_EN, eLabel1, eLabel2);
```

GX Works3改善前

GX Works3改善後

```
MAIN [PRG] [ST] 2ステップ *
1 | bLabel_ENO := SIN bLabel_EN, eLabel1, eLabel2);

MAIN [PRG] [ST] 2ステップ *
1 | eLabel2 := SIN_E(bLabel_EN, bLabel_ENO, eLabel1);
```

手で修正が必要だったが...



自動で互換用
命令に!

プロジェクトを開くときに自動で互換用命令に変更
してくれるから手動での修正工数が減ったんだ!

プログラムの命令を変更する手間が減ったから
GX Works2 形式のデータが流用しやすくなったね!

プログラムを修正せずに引き継げるデータが増えたね!
これで君もシーケンスアープ!



システム設計の頼れるヒーロー 教えて!GX3号

第19回

NEW

Ver.Up
新機能

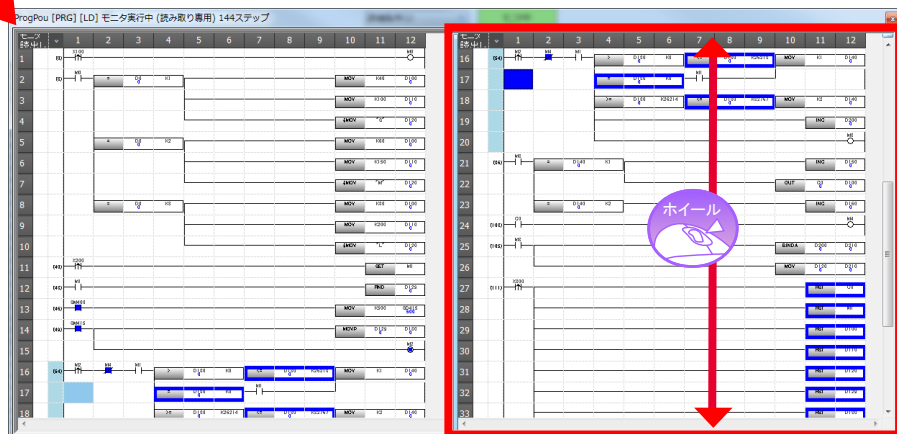
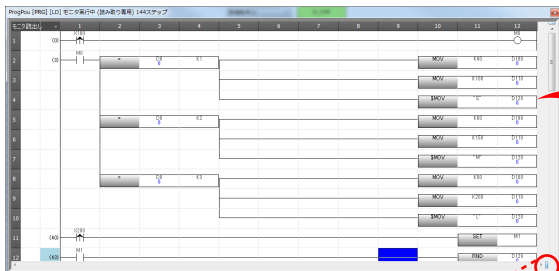
分割画面を別々にスクロール!



プログラムが長くて何度もスクロールしているよ。
見たい部分を同時に見れないかなあ…。



左右に画面を分割するといいわいよ!
分けた画面を別々にスクロール
できるようになったんだ!



ウィンドウ(W)

分割(P)

or

ドラッグ!



メニュー又は
右下バーを
ドラッグして
画面を分割

選択されたウィンドウのみスクロールすることができる!



見たい画面を同時に表示できるんだね!
離れた回路からコピー&ペーストする時や、
モニタしている時に重宝するね!

メニューの"ウィンドウ"から左右を同時にスクロール
することもできるから覚えておいてね!



ウィンドウ(W)

分割(P)

分割の解除(R)

分割ウィンドウ操作(S)

フローティング(F)

ドッキング(K)

次にフォーカス(N) Ctrl+Alt+J

前にフォーカス(P) Ctrl+Alt+K

左右を同時にスクロール(S)

クリック!

プログラムが長い時は分割して見たい回路を表示しよう!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第20回

NEW

Ver.Up
新機能

ウィンドウ幅にエディタを自動調整



ウィンドウ幅を変更したら回路図の倍率も変更しないと見にくいなあ…。

最新の GX Works3 では"ウィンドウ幅にエディタを合わせる"機能で自動で調整できるようになったよ!

そんなときは



何度も調節しなくて良くなったんだね!



ウィンドウ幅に合わせて倍率を自動調整



自動調整
モード切替

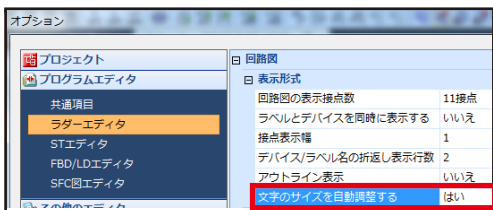
- 倍率を設定する(S)...
- 拡大(I) Ctrl+Num +
- 縮小(O) Ctrl+Num -
- ウィンドウ幅にエディタを合わせる(F)**

or

107%

- デバイスラベル名表示形式変更(L)
- アウトライン(U)
- デバイス表示(V) Ctrl+Alt+F6
- 文字のサイズ(X)
- 大きくする(B)
- 小さくする(M)
- 自動調整(A)

or



さらにメニューバーやオプションからウィンドウの倍率に合わせて文字サイズが自動調整できるようになったんだ!

みやすい



ウィンドウの倍率変更でレイアウトが変わらなくなるから回路図が見やすいままだね!

これでウィンドウ幅変更の使い勝手が良くなったね!
これで君もシーケンスアープ!





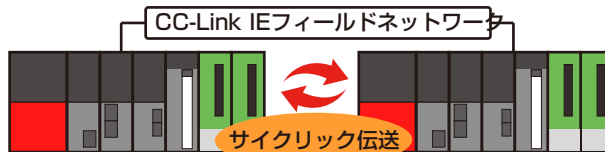
システム設計の頼れるヒーロー 教えて!GX3号 第21回



複数システム間のシミュレーション



ネットワークで繋いだプログラムを机上で確認したいなあ。



システムシミュレーション機能を使おう！
必要なプロジェクトを開いて"システム追加"からシステム構成を作成して"開始"しよう！

デバッグ(B) シミュレーション(S) → シミュレーション開始(S) → シミュレーション停止(P) → システムシミュレーション(E) → 起動(S) → 接続先設定(C) → 接続先解除(D)

手順① クリック!

現在開いているGX Works3プロジェクトからシステム構成を読み込んで、システムを追加します。

プロジェクト名(P): C:\Users\Desktop\通常局_ex3
システム番号(N): 2

手順② システム追加(A) クリック!

サイクリック伝送

RDY ERR RUN USR RESET STOP RUN 管理局_ex3
ERR PRM DLK ST: - NW

サイクリック伝送

RDY ERR RUN USR RESET STOP RUN 通常局_ex3
ERR PRM DLK ST: - NW

手順③ 開始(S) クリック!

システム毎にプロジェクトを接続！
接続が完了後プロジェクト名が表記される！

デバッグ(B) シミュレーション(S) → シミュレーション開始(S) → シミュレーション停止(P) → システムシミュレーション(E) → 起動(S) → 接続先設定(C) → 接続先解除(D)

手順④ クリック!

システム構成情報(S)

System1
Base0-R35B
R04CPU ☆ 白考機
R042C4
R042NT2P
RJ71GP21-SX
(空)

System2
Base0-R35B
R04CPU ☆ 白考機
R042C4
R042NT2P
RJ71GP21-SX
(空)

接続対象のシステムを表示する(D)
接続対象のユニットを表示する(M)

接続 キャンセル

システム構成ができれば
各プロジェクトを"接続先
設定"から"接続"しよう！



これでリフレッシュによる通信が確認できるね！

プロジェクトの書込みは従来と同じだよ！
複数のシステムでもシミュレーションができるんだ！
これで君もシーケンスアープ！





システム設計の頼れるヒーロー 教えて!GX3号

第22回

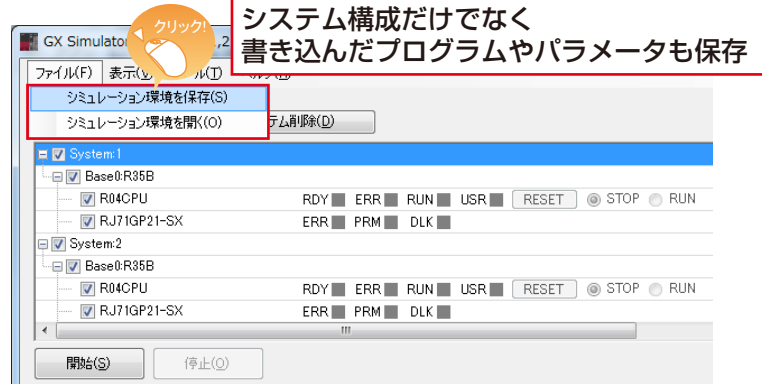
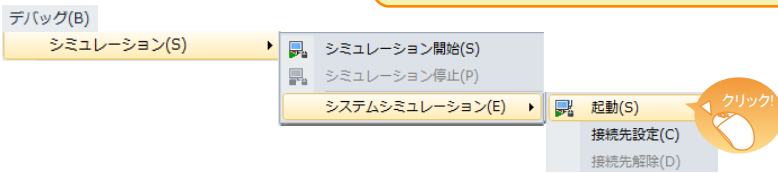


シミュレーション環境の保存



前回教えてもらった複数システム間のシミュレーションを使ってるんだけど毎回システム構成を設定するのは面倒だなあ…。

シミュレーション環境をファイルに保存しよう！
そうすれば次回使用する時にシミュレーション環境を手軽に復元できるよ！

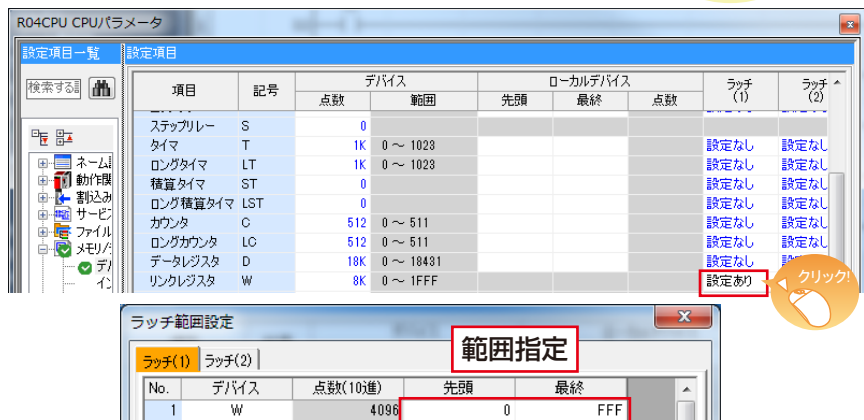


保存しておけば接続先を設定するだけで再開できるんだね！

シミュレーション時にラッチ機能も使えるよ！
CPU パラメータのメモリ / デバイス設定から設定できるよ！



ラッチしたデバイスは保存すれば再開しても使えるんだね！



シミュレーション環境の保存機能を上手に使おう！
これで君もシーケンスアープ！





システム設計の頼れるヒーロー 教えて!GX3号 第23回

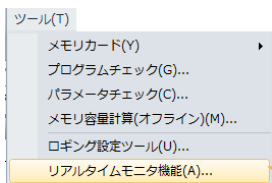


デバイスの変化をモニタで確認



装置のラダープログラムをメンテナンスしたから正しく動いているか確認したいなあ…。

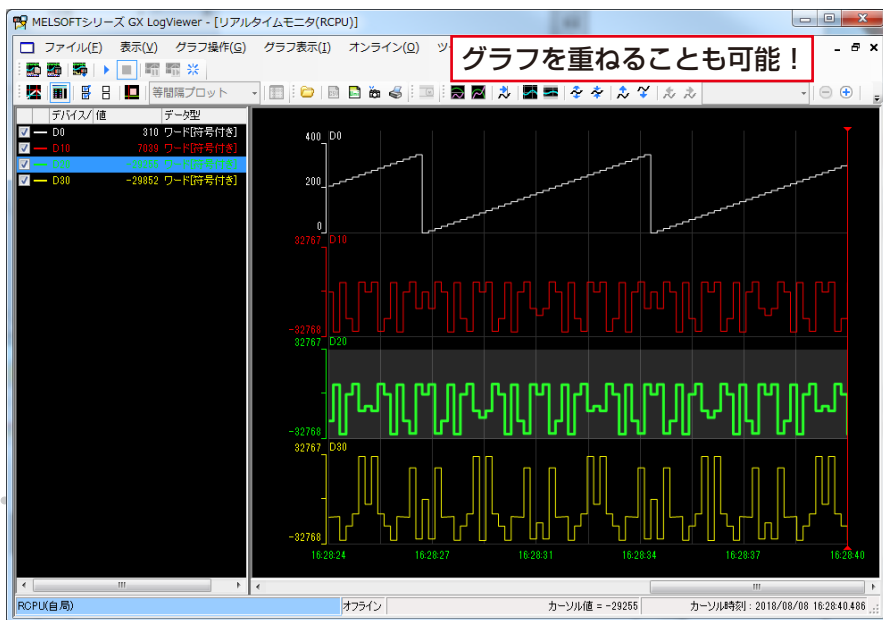
GX LogViewer を使おう！ GX Works3 の"ツール"→"**リアルタイムモニタ機能**"から簡単に起動できるよ！
確認したいデバイスを選択して変化をモニタできるんだ！



リアルタイムモニタ設定 (RCPU)

ファイルから読み込み... GX Works3プロジェクトの読み込み...
ラベルを指定する 参照先プロジェクト名

No.	ラベル	デバイス	データ型
1		D0	ワード(符号付き)
2		D10	ワード(符号付き)
3		D20	ワード(符号付き)
4		D30	ワード(符号付き)
5			



それぞれのデバイス毎の軌跡が見れるのか！
変化のタイミングがわかるのが便利だね！

確認したいデバイスが複数あっても同時に見れるから各デバイスの関係性もわかりやすいよ！



GX LogViewer は GX Works3 に同梱されているし、
FA サイトからもインストールできるよ！
これで君もシーケンスアープ！





システム設計の頼れるヒーロー 教えて!GX3号

第24回



リアルタイムモニタの簡単設定

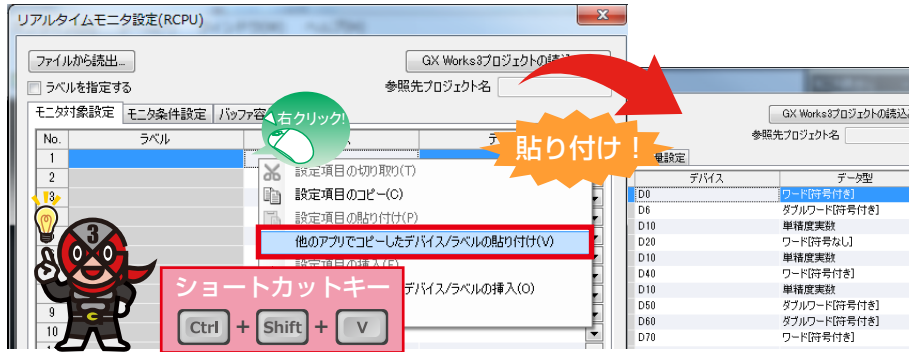


リアルタイムモニタに毎回いくつも
デバイスを設定するのは手間だなあ…。

設定するデバイスをウォッチウィンドウからコピーできる
よ! 右クリック"他のアプリでコピーしたデバイス/ラベル
の貼り付け"又はショートカットキーで貼り付けしてね!



ウォッチウィンドウ
からコピー



貼り付け!

ショートカットキー
Ctrl + Shift + V

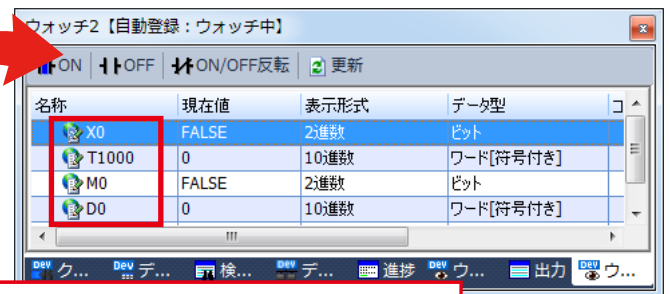
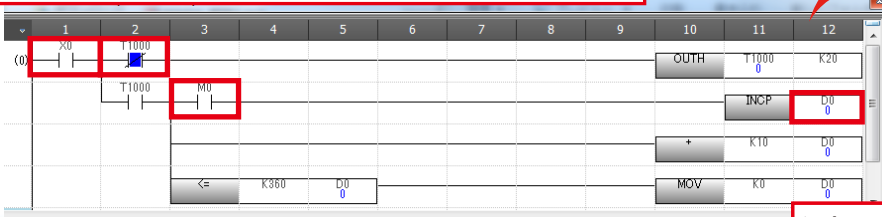


ウォッチウィンドウからコピーする
ことでまとめて設定できるんだね!

さらにラダーからウォッチウィンドウへの登録は自動登録機能を使えばより簡単に登録できるよ!



【ウォッチの自動登録機能の使用方法】
"ツール"の"オプション"から"モニタ" → "ラダーエディタ" → "ウォッチウィンドウへの自動登録の設定"より
指定した登録先のウォッチウィンドウに自動登録!



選択した回路にあるデバイスを全て自動登録!

これでリアルタイムモニタがより手軽に使えるね!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号 第25回



ロギングデータからデバイス状態を再現

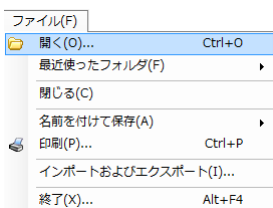


装置の動きがおかしいから何が原因か確認したいなあ…。

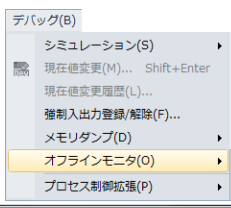
そんなときは



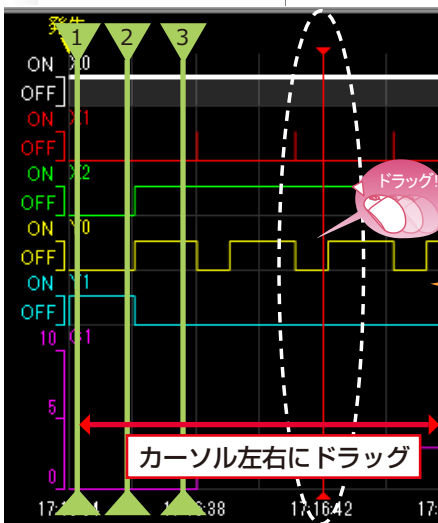
ロギングファイルを取っておけば、GX LogViewerのヒストリカルトレンドグラフとラダー図を連動してデバイス状態を再現できるよ!



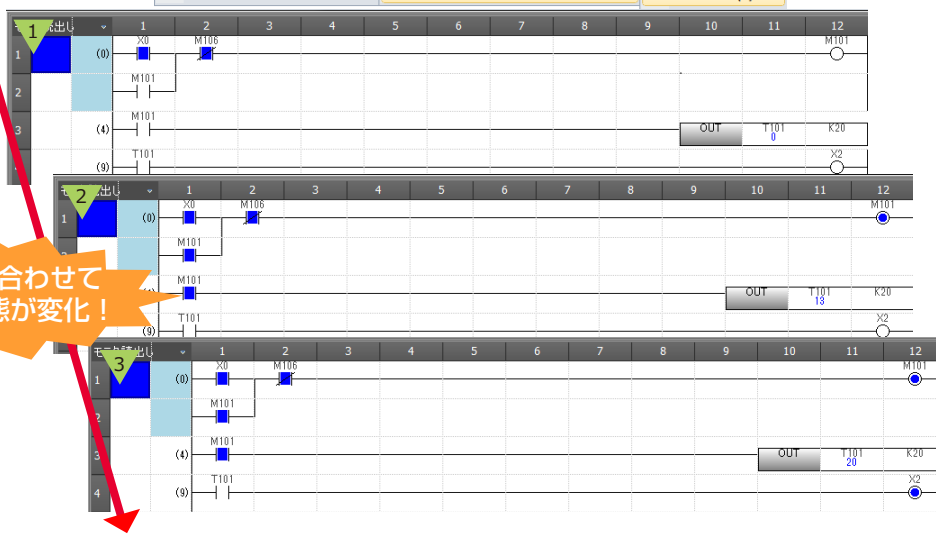
ロギングファイルはバイナリファイル形式で保存しておこう!



GX LogViewerでロギングファイルを開いたらGX Works3の"オフラインモニタ(ロギング)"の"接続先設定"から連携!



カーソルに合わせてデバイス状態が変化!



このグラフでは赤カーソルが当たっている時間のデバイス状態をラダー図を使って確認できるんだ!



その時の動きをラダー図で追って確認できるから、これを見れば何が原因かわかりそうだ!

この機能でデバイスの解析ができるんだね!これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号 第26回



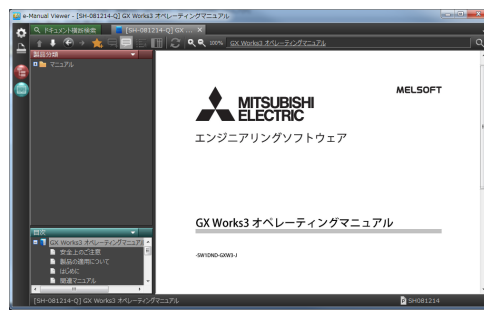
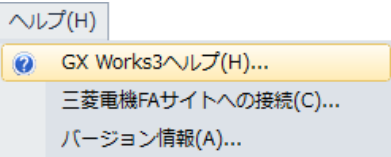
GX Works3ヘルプを上手に活用

GX Works3 で使うヘルプは便利な機能を
登載した電子マニュアルになってるんだ!

便利なヘルプ



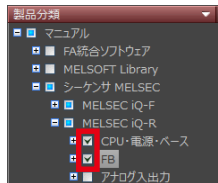
そうなんだ? 今までの
ヘルプとどう違うの?



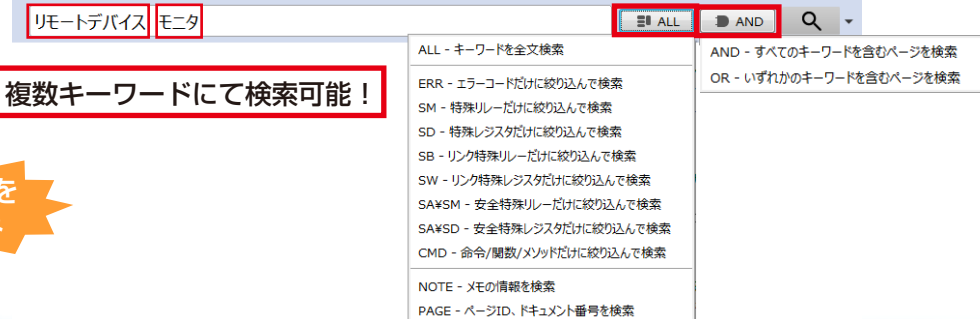
例えば『横断検索機能』! 複数のマニュアルから
検索したいキーワードを一気に検索できるんだ!

複数のマニュアルから検索してくれるのか!
一つ一つマニュアルを開かなくても良いんだね!

さらに、横断検索する時に検索したい
マニュアルを絞り込むこともできるよ!



ALL、ANDタブで検索方法も絞り込み



製品分類を
絞り込み

ヘルプを上手に使って欲しい情報を手早く見つけよう!
これで君もシーケンスアープ!



システム設計の頼れるヒーロー 教えて!GX3号 第27回



ヘルプを自分用にカスタマイズ

GX Works3 の関連マニュアルを必要に応じて一括でダウンロードできるよ!

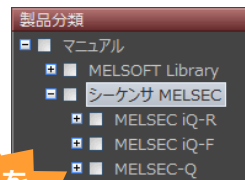
便利なヘルプ



ユニット毎とかジャンルでまとめてマニュアルをダウンロードできるんだ!?



ドキュメントタイトル	ドキュメント
<input type="checkbox"/> シーケンサ MELSEC	
<input checked="" type="checkbox"/> MELSEC iQ-F	
<input checked="" type="checkbox"/> MELSEC iQ-R	
<input type="checkbox"/> MELSEC-A	
<input type="checkbox"/> MELSEC-F	
<input type="checkbox"/> MELSEC-L	
<input checked="" type="checkbox"/> MELSEC-Q	
<input type="checkbox"/> エンジニアリングソフトウェア	
<input type="checkbox"/> ネットワーク関連製品	



マニュアルをまとめて管理

さらに、よく見るページをお気に入り登録したり、ノウハウ情報や覚えておきたいことをメモ機能を使って追加できるよ!



お気に入りに追加

メモを追加

マニュアルを使いやすいようにカスタマイズできるんだね!



メモを追加してより便利に!



そのカスタマイズしたお気に入りやメモはマニュアルが更新されても継承されるよ!



カスタマイズしたデータはエクスポートして別のPCにインポートすることでノウハウを継承できるよ!これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!Gx3号

第28回



「F1」キーでダイレクトに検索



他の人が作成した回路図に使われている命令の詳細が知りたいなあ…。

回路図上の知りたい命令を選択して「F1」キーを押すとヘルプが起動して詳細情報が確認できるよ!

便利なヘルプ



「F1」キーで
すぐに命令確認

ショートカット

F1

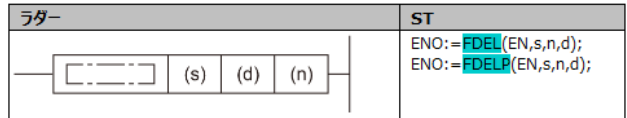
9	10	11	12
FDEL	D0	ZR0	Z0

データテーブルのデータ削除

FDEL(P)



データテーブルの(n)番目のデータを削除します。



探さずにその場で調べられるんだ!?! それは便利だね!

命令だけでなく汎用 FB や汎用 FUN でも「F1」キーを使って詳細が確認できるよ!

工数削減!



ショートカット

F1

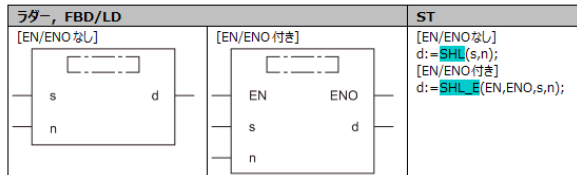
5	6	7	8
	SHL		
[D0]	IN	SHL	[D1]
[K3]	N		

33.1 nビット左シフト

SHL(E)



入力値を(n)ビット数分左へシフトして出力します。



これでどういう動作をする命令か確認できたから回路図を把握できたよ!

調べたい命令、FB/FUNがあれば「F1」キーを覚えておこう!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第29回

NEW
Ver.Up
新機能

ラベルの2重コイルチェック!

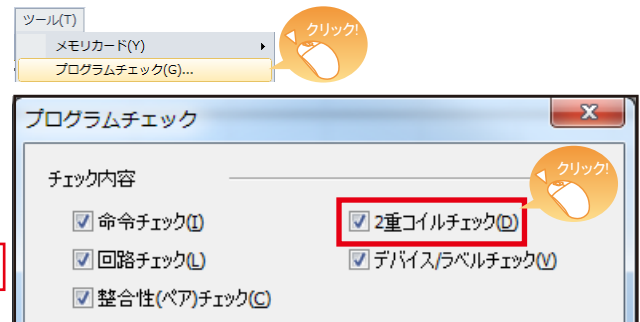
ラベルの便利機能を紹介するよ!
ラベルを使っているプログラムで、
2重コイルを検出できるようになったんだ!



2重コイルによる不整合の確認をクロスリファレンスで行う必要がなくなったんだね!?

10	11	12
		bLabel
		○
		bLabel
		○
SET		bLabel2
SET		bLabel2

同じコイルがあると...



No.	結果	データ名	分類	内容
1	CheckWarning	ProgPou	プログラムチェック	'bLabel(OUT)'は2重コイルとなっています。誤動作を起こす可能性があるためプログラムを見直してください。
	CheckWarning	ProgPou	プログラムチェック	'bLabel2(SET)'は2重コイルとなっています。誤動作を起こす可能性があるためプログラムを見直してください。

同じラベルを検出してくれる

2重コイルになっている部分を
動作確認前を見つけることができるよ!



ラベルを使ったプログラムが作りやすくなったよ!
覚えておくれ!

今回を機にラベルをどんどん活用していこう!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第30回

NEW
Ver.Up
新機能

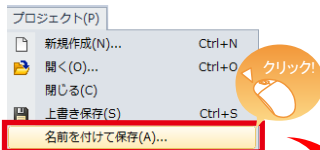
プロジェクトファイルの圧縮

プロジェクトファイルを圧縮して保存できるようになったから紹介するね!
これでファイルサイズを削減することができるよ!

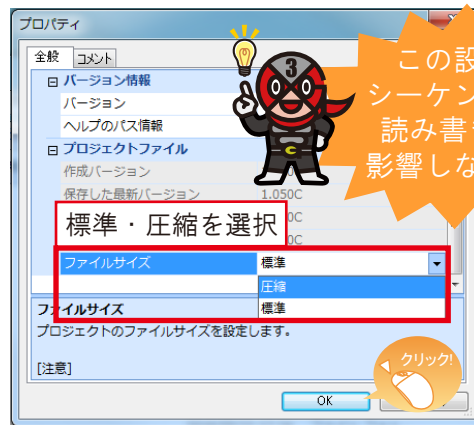
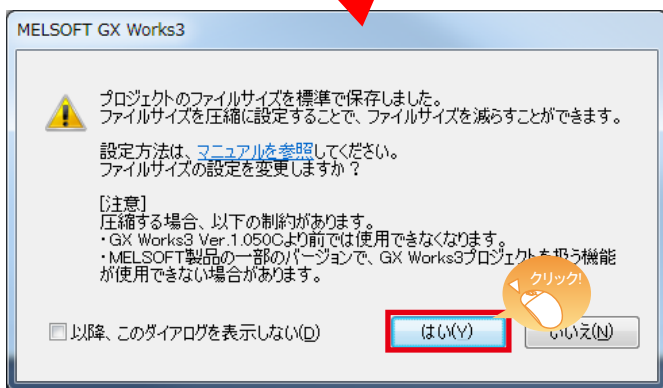
便利な新機能



本当だ! 圧縮するか選択する
ダイアログが出てきたよ!?



プロジェクトを開く時や名前を付けて
保存する時に確認されるよ!



この設定は
シーケンスとの
読み書きには
影響しないよ!

プロジェクトファイルの受け渡しに
かかる時間もコンパクトになるね!

GX Works3 の Ver1.050C より前で
開く場合は、標準形式で保存しよう!

圧縮形式のプロジェクトを使う場合は、
GX Works3 を最新にしよう!
これで君もシーケンスアープ!



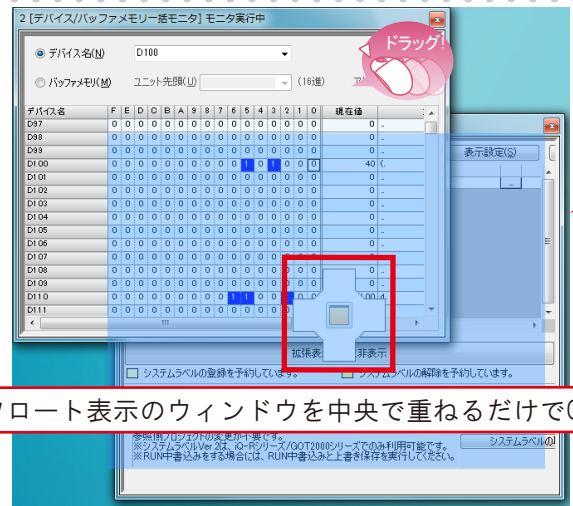
システム設計の頼れるヒーロー 教えて!GX3号 第31回

NEW Ver.Up 新機能 フロート表示ウィンドウの整頓



画面をたくさんフロート表示しちゃうと見たい画面がどこにいったかわからなくなるなあ…。

フロート表示のウィンドウをまとめられるようになったんだ!



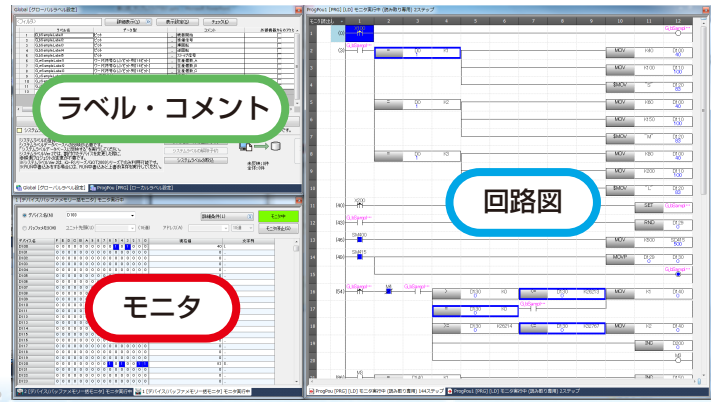
フロート表示のウィンドウを中央で重ねるだけでOK!



タブでウィンドウを切替!



整頓できるから目的のウィンドウもすぐ見つけれられるね!



まとめたフロートウィンドウは複数作ることもできるよ!



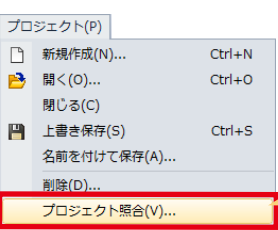
ウィンドウをまとめて自分が使いやすいように配置しよう!
これで君もシーケンスアープ!



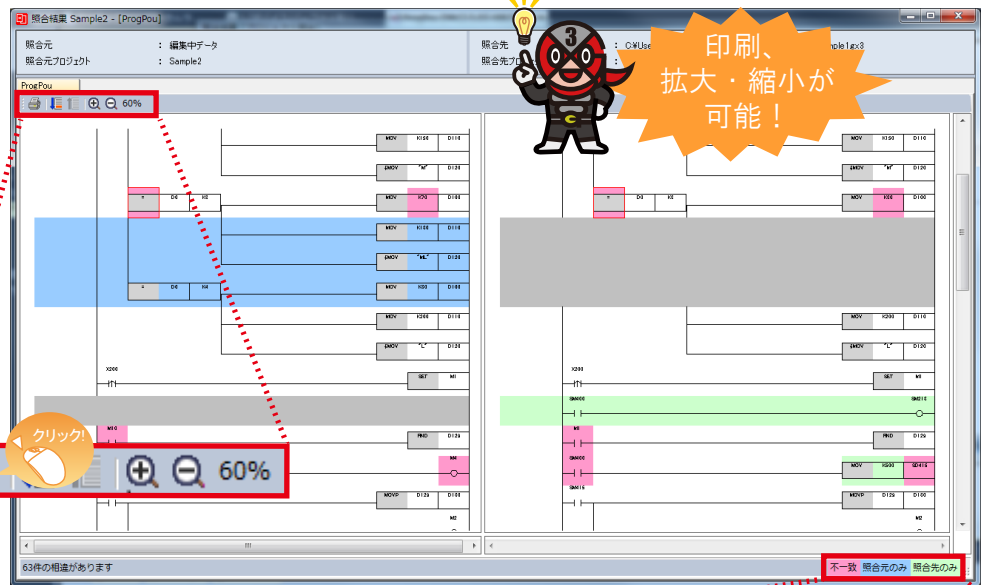
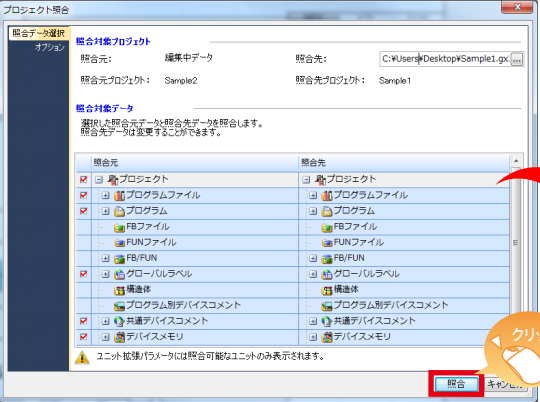


システム設計の頼れるヒーロー 教えて!GX3号 第32回

NEW Ver.Up 新機能 照合結果の新しい付加機能!

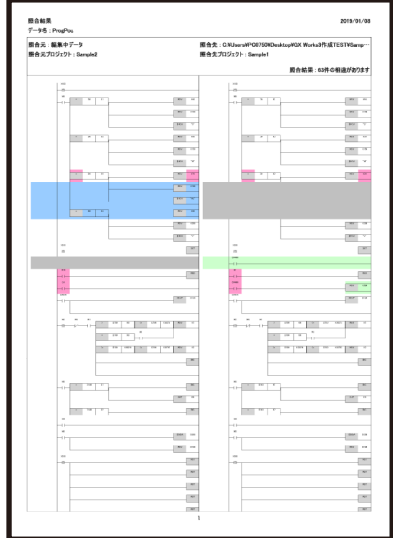


GX Works3 の照合機能は図形式で差分を色分けしてくれるんだ!
今回はさらに便利になってラダー図の照合結果の拡大縮小表示や印刷もできるようになったんだ!



印刷、拡大・縮小が可能!

不一致 照合元のみ 照合先のみ



変更状態によって色分けされているから一目瞭然だね!



GX Works3 のプロジェクトデータと過去のプロジェクトデータとの差分チェックがしやすいね!

上手に使って工数を減らそう!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第33回

NEW
Ver.Up
新機能

クロスリファレンスの強調/絞り込み



選択した
デバイスに連動して
強調表示して
くれるよ

クロスリファレンス機能がさらに
便利になったから紹介するよ!

カーソル移動

デバイス選択

デバイス/ラベル	プログラムファイル名	データ名	ステップ
<input checked="" type="checkbox"/> X0	X0	↑ト	0ステップ
<input type="checkbox"/> Y0	Y0	↓ト	1ステップ
<input type="checkbox"/> X1	X1	↑ト	2ステップ
<input type="checkbox"/> Y1	Y1	↓ト	3ステップ
<input type="checkbox"/> X2	X2	↑ト	4ステップ
<input type="checkbox"/> Y2	Y2	↓ト	5ステップ
<input type="checkbox"/> X3	X3	↑ト	6ステップ
<input type="checkbox"/> Y3	Y3	↓ト	7ステップ

選択したデバイスが強調表示

デバイス/ラベル	プログラムファイル名	データ名	ステップ
<input checked="" type="checkbox"/> X0	X0	↑ト	0ステップ
<input type="checkbox"/> Y0	Y0	↓ト	1ステップ
<input type="checkbox"/> X1	X1	↑ト	2ステップ
<input type="checkbox"/> Y1	Y1	↓ト	3ステップ
<input checked="" type="checkbox"/> X2	X2	↑ト	4ステップ
<input type="checkbox"/> Y2	Y2	↓ト	5ステップ
<input type="checkbox"/> X3	X3	↑ト	6ステップ
<input type="checkbox"/> Y3	Y3	↓ト	7ステップ

移動カーソルに追従して強調表示



クロスリファレンス画面で選択したデバイスを
すぐ見つけられるようになったよ!

他にも、ローカルラベルだけを表示したい時は、「プログラム
部品 (POU) 名 / ラベル名」で絞り込めるようになったよ!



クロスリファレンス1

bLabel1 (プロジェクト)

デバイス/ラベル	プログラムファイル名	データ名
フィルタ条件	フィルタ条件	フィルタ条件
bLabel1	MAIN	ProgPou1
bLabel1	MAIN	ProgPou2

クロスリファレンス1

ProgPou2/bLabel1 (プロジェクト)

デバイス/ラベル	プログラムファイル名	データ名
フィルタ条件	フィルタ条件	フィルタ条件
bLabel1	MAIN	ProgPou2

ProgPou2のローカルラベルを絞り込み!



プログラム部品 (POU) 毎にローカルラベルが絞り込めるんだね!

クロスリファレンスを上手に活用しよう!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第35回



インラインSTでの簡単ラベル登録



インライン ST ボックス内で新しくラベルを使いたいけど
わざわざラベルエディタで登録するのはめんどろなああ…。

登録したいラベル名を記述してカーソルを当てた
状態で「Alt」+「F2」キー！これで登録できるんだ！



```
1 wLabel1 := D0 - 100;
```



カーソルを
当てて

Alt + F2

OR

- 編集(E)
- インラインST(X)
 - インラインSTボックス挿入(B) Ctrl+B
 - テンプレート表示(E) Ctrl+Alt+F1
 - テンプレートの引数選択(左)(F) Alt+←
 - テンプレートの引数選択(右)(J) Alt+→
 - ラベルを登録する(V) Alt+F2

新たにラベル名「wLabel1」を登録

未定義ラベル登録

グローバルラベル、ローカルラベルに定義されていません。
新規に登録するラベル情報を設定してください。

ラベル名(L) wLabel1

ラベル設定情報

登録先(B) ローカルラベル(ProgPou)

クラス(S) VAR

データ型(Y) ワード[符号付き]

定数値(N)

コメント(M)

ラベル情報を登録後、ラベルエディタを開きラベルの詳細を設定する。(E)

クリック!

OK キャンセル

```
1 wLabel1 := D0 - 100;
```

登録完了しラベル用の表示色に



インライン ST ボックス内で直接ラベルを登録できるんだね !?

ラダーで作成中の時と同じように
インライン ST 機能でも登録できるよ !



ラベル登録は多用する機能だから是非覚えておこう !
これで君もシーケンスアープ !





システム設計の頼れるヒーロー 教えて!GX3号

第36回



インラインSTの構文テンプレート



IF文やCASE文を記述する時に
間違ふことがあるなあ…。

IF文などの制御構文の引数は
「Ctrl」+「Alt」+「F1」キーでテンプレート表示できるよ!



```
1 | if
```



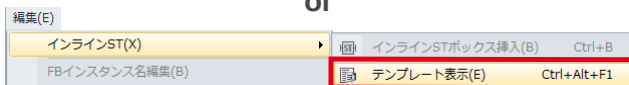
```
1 | IF ?条件式? THEN
2 |   ?実行文? ;
3 | ELSE
4 |   ?実行文? ;
5 | END_IF;
```

テンプレート表示!

ショートカット

Ctrl + Alt + F1

or



条件式などの
記述箇所も
わかりやすい



これなら制御構文を間違えずに記述できるね!

記述すべき引数にジャンプしたい時は、
「Alt」+「←」, 「Alt」+「→」キーでできるよ!



```
1 | IF ?条件式? THEN
2 |   ?実行文? ;
3 | ELSE
4 |   ?実行文? ;
5 | END_IF;
```



```
1 | IF ?条件式? THEN
2 |   ?実行文? ;
3 | ELSE
4 |   ?実行文? ;
5 | END_IF;
```

記述箇所にジャンプ!

ショートカット

Alt + ←, →



ジャンプ後は
選択状態になるから
そのまま入力
できるよ!

FUN(ファンクション)やFB(ファンクションブロック)でも
同様の操作でテンプレート表示ができるよ!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第37回



インラインSTでのデバイス型指定



インライン ST は便利だけどワードデバイスのデータ型はワード [符号付き] でしか使えないのかなあ…。

デバイス型指定子 (サフィックス) によってデータ型を指定できるよ!



デバイス型指定子

データ型	指定子	占有ワード数
ワード[符号付き]	なし	1ワード
ワード[符号なし]	:U	1ワード
ダブルワード[符号付き]	:D	2ワード
ダブルワード[符号なし]	:UD	2ワード
単精度実数	:E	2ワード
倍精度実数	:ED	4ワード



これでワード型以外のデータ型でもインライン ST 機能が使えるんだね!

デバイス型指定子が付加されたデバイスなら指定したデータ型の演算が正しくできるね!



実数を使いたい場合

```
SM400 1 D10:E := (D0:E + D2:E) * D4:E / E2; D10:E = 21.280; D0:E = 3.200; D2:E = 4.400; D4:E = 5.600;
```

整数を使いたい場合

```
SM400 1 D20:U := (D12:U + D14:U) * D16:U / K2; D20:U = 28; D12:U = 3; D14:U = 5; D16:U = 7;
```



演算式の指定子はそろえてね!

デバイス型指定子を覚えておこう!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号 第38回

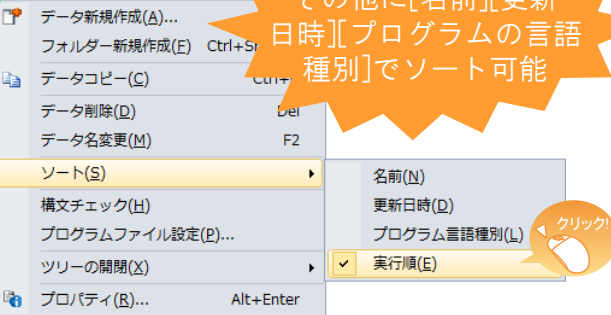


ナビゲーションを見やすく整頓

ナビゲーションウィンドウ内のソート機能を知っているかな? プログラムやプログラムブロック (POU) を [実行順] などに見やすく並び替えられるよ!



複雑なプログラムでも [実行順] に並び替えれば確認や変更の作業がしやすいね!

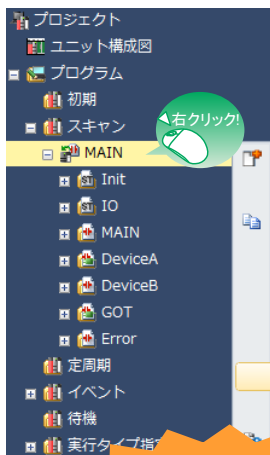
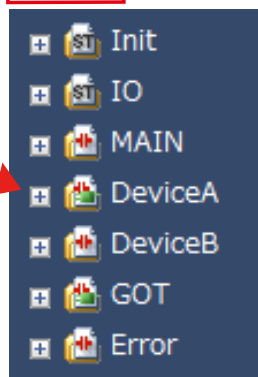


その他に[名前][更新日時][プログラムの言語種別]でソート可能

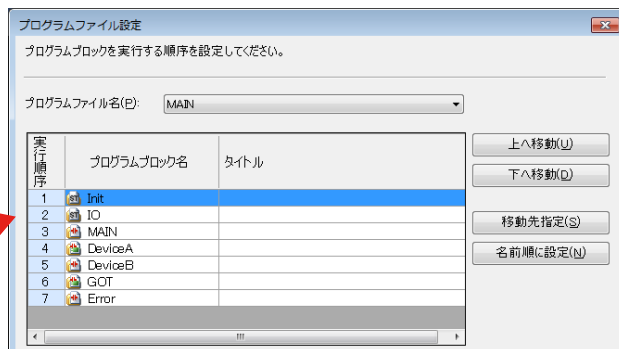
名前順



実行順



実行順はナビゲーションウィンドウの右クリックから変更できるよ!



右クリックメニューのプログラムファイル設定

ナビゲーションの右クリックを使いこなそう!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第39回

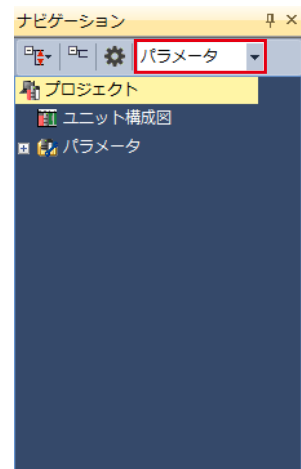
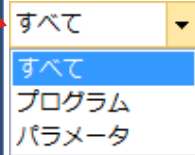


ナビゲーションを使いやすく表示

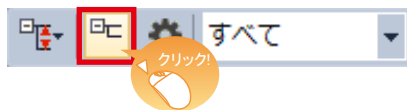
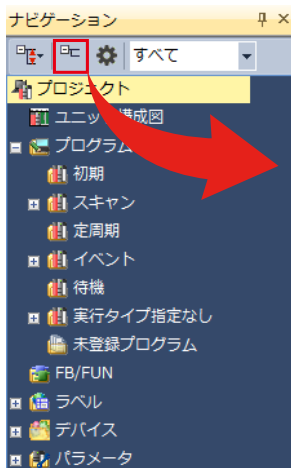
ナビゲーションウィンドウのフィルタ機能を知っているかな? 目的に応じて表示項目を [プログラム] のみ、 [パラメータ] のみ、 [すべて] 表示に切り替えられるよ!




作業に必要な情報だけを表示できるね!



さらにナビゲーションウィンドウのツールバーで [簡易表示] に切り替えると未使用のフォルダを非表示にできるよ!



ツールバーのオプション  でいろいろな設定を変更できるよ! これで君もシーケンスアープ!





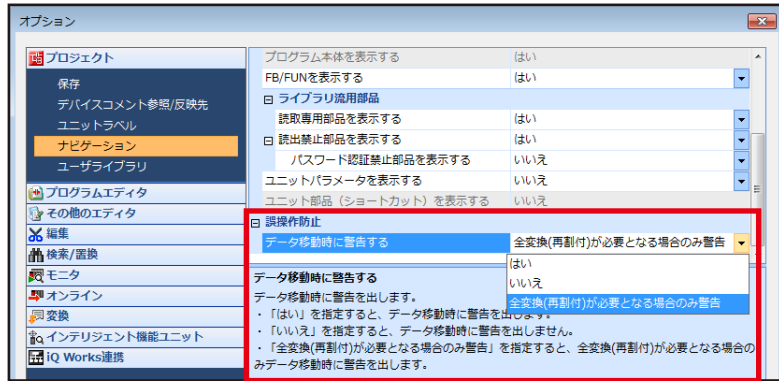
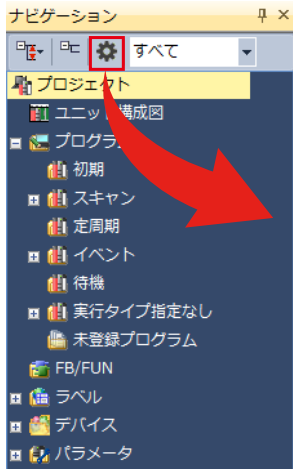
システム設計の頼れるヒーロー 教えて!GX3号

第40回

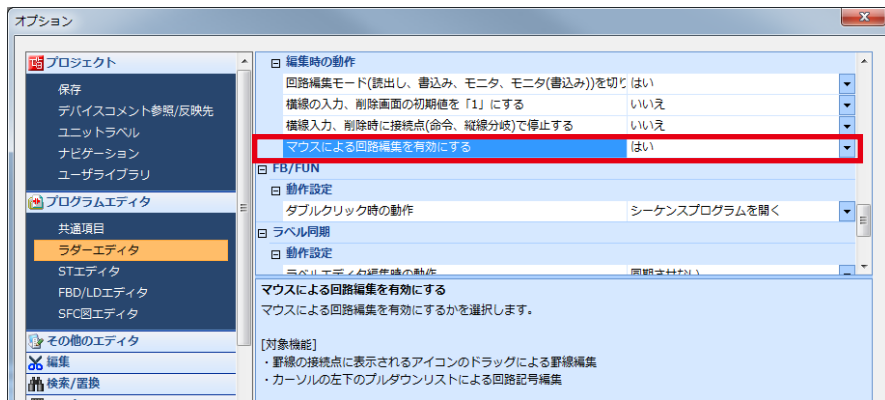


オプション設定で誤操作を防止

ナビゲーションウィンドウのオプションで設定できる、「誤操作防止」機能を知っているかな？
データを移動したときに警告を出してくれるよ！



ドラッグによる回路野線の誤操作が気になる人は
ラダーエディタのオプションから無効にできるよ！



慎重にデバッグしたい時は、データ移行時の警告は「はい」を
マウスによる回路編集は「いいえ」を選択するといいね！

設定を変更して自分なりにカスタマイズ！
これで君もシーケンスアープ！





システム設計の頼れるヒーロー 教えて!GX3号

第41回

NEW
Ver.Up
新機能

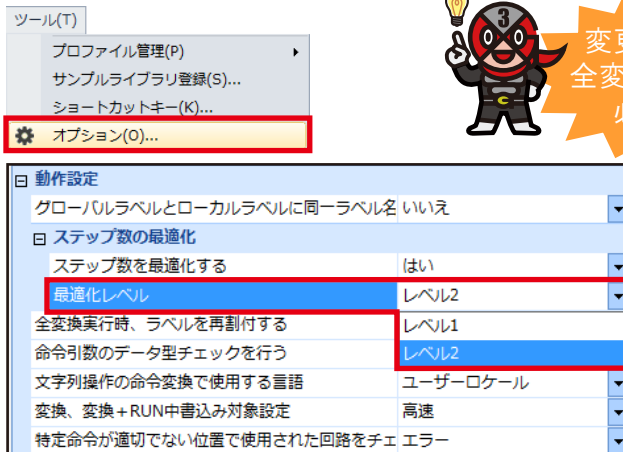
最適化でデータをコンパクトに!

ステップ数の最適化という機能を知っているかな?
プログラムサイズやスキャンタイムを削減することが
できるよ!



魅力的な機能だね!
どうやったら削減できるの!?

“ツール”→“オプション”から最適化レベルを
レベル2に変更しよう!
全変換する時に最適化できるよ!



変更した時は、
全変換・再割付が
必要だよ!

インライン ST 使用時は単項演算、二項演算、
IF, CASE 文で ELSE を使用した時に効果があるよ!



Ver.1.052E より前バージョンだとレベル1以下の
プロジェクトしか開けないから注意しよう!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第42回



ラベル不使用時のカスタマイズ

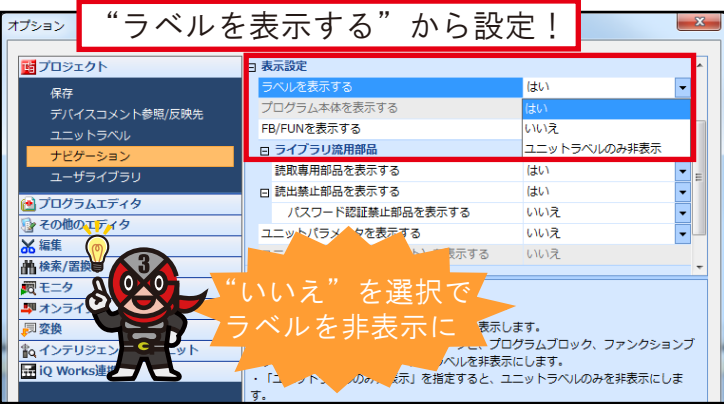


このプログラムにはラベルを使っていないから
ナビゲーションウィンドウをすっきりさせたいなあ…。

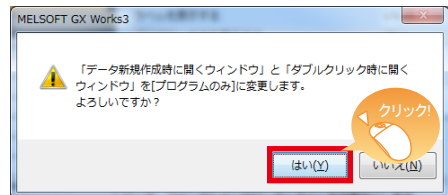


ラベルを使わない時は
オプションから非表示に
することもできるよ!

さらに“いいえ”を選択すると
表示されるダイアログから…



“いいえ”を選択で
ラベルを非表示に



ラベルを非表示にすると
左記の設定も合わせて
変更できるよ!



プログラム
ブロックから
プログラム本体が
開けるよ!



よく行う操作だから助かるね!

ラベルを使わない時は既定値に設定すると便利だね!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第43回

NEW
Ver.Up
新機能

一覧表示をハイライトでマーキング



クロスリファレンスはデバイスがたくさん表示されるから着目していたデバイスがどれかわからなくなるなあ…。

一覧表示した項目にハイライトをつけられるんだ! 着目しているデバイスをマークできるよ!



デバッグ中に怪しいデバイスを色分けで分類することもできるんだね!

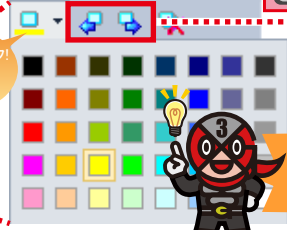
ショートカットキー
(前後のハイライトへ移動)

ハイライト設定

F2 , Shift + F2

クロスリファレンス1

デバイスラベル	デバイス	回路記号	位置	プログラムファイル名	データ名
フィルタ条件	フィルタ条件	フィルタ条件	フィルタ条件	フィルタ条件	フィルタ条件
M100	M100	- (-)	1ステップ	MAIN	ProgPou
M100	M100	- (-)	3ステップ	MAIN	ProgPou
M100	M100	- (-)	6ステップ	MAIN	ProgPou
M100	M100	- (-)	9ステップ	MAIN	ProgPou
M100	M100	- (-)	10ステップ	MAIN	ProgPou
M100	M100	- (-)	12ステップ	MAIN	ProgPou
M100	M100	- (-)	15ステップ	MAIN	ProgPou
M100	M100	- (-)	16ステップ	MAIN	ProgPou
M100	M100	- (-)	18ステップ	MAIN	ProgPou



ハイライト部分で次を検索してくれるよ!

	5	6	7	8	9	10	11	12
6	(14)	X5	M100				SET	M100
7	(17)	X6	M100				RST	M100
8	(20)	X7	M100					

クロスリファレンスだけでなく、検索結果や出力ウィンドウでも使えるよ!

画面上で明確化できるからメモを取ったりしなくてもスムーズに作業が行えるよ!



一覧表示の可読性がさらに向上したね! これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!Gx3号

第44回

NEW
Ver.Up
新機能

コメントでプログラミング

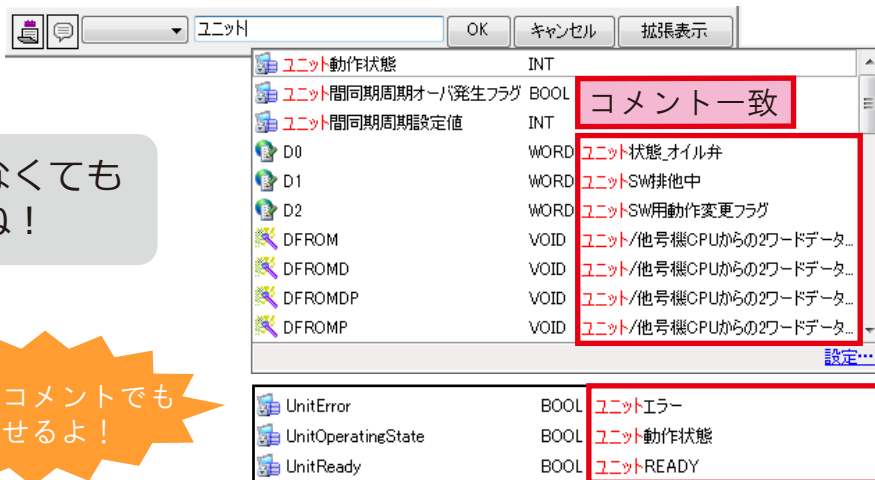


プログラミングするためにデバイス番号を覚えるのは大変だなあ…。

入力内容がデバイスコメントと一致したのも回路入力ダイアログで候補として表示するようになったよ!

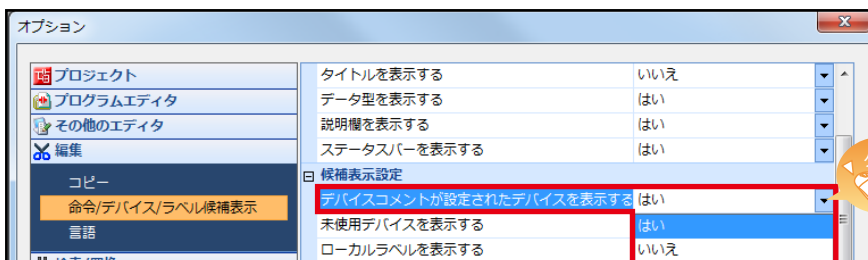


デバイス番号を覚えてなくてもコメントで探せるんだね!



ラベルコメントでも探せるよ!

デバイスコメントが設定されたデバイスを候補表示するにはオプションから設定してね!



コメントの記述を工夫してプログラミングしよう!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第45回

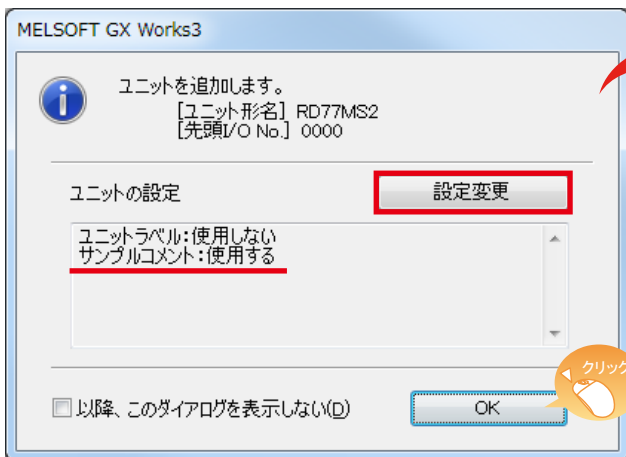
NEW
Ver.Up
新機能

サンプルコメントを自動で読み出す

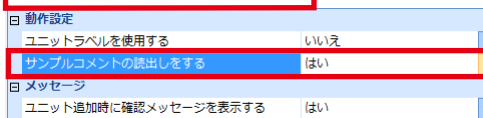


サンプルコメントをいつも使っているけど、いちいち読み出しするのは面倒だなあ…。

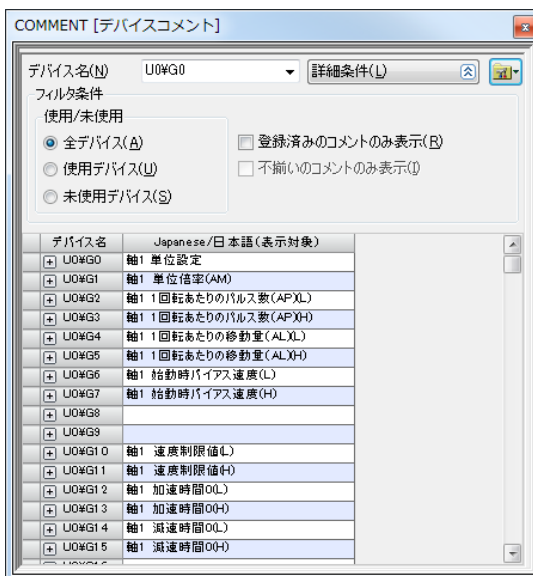
サンプルコメントの読み出しをする設定にしておけば、ユニットを追加した時に自動でコメントを読み出してくれるよ!



オプションにて設定



規定値にして
おくと便利だよ!



ユニット追加画面でサンプルコメントを使用するになっているか確認すればOKだね!

特殊リレーや特殊レジスタのデバイスコメントも自動で登録されるから便利だよ!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第46回

NEW
Ver.Up
新機能

選択したデバイスメモリの読み書き

選択したデバイスメモリ範囲をシーケンサに読み書きすることができるようになったよ!

便利な機能



部分的にデバイスに値を書込むことができるんだね!

画面上部のボタンから読み書きできるよ!



MAIN [デバイスメモリ]

デバイス名(N) D0 詳細条件(L)

デバイス名	+0	+1	+2	+3	+4	+5	+6	+7	+8
D0	1	2	3	4	5	5	5	5	5
D8	6	7	8	9	10	10	40	40	...
D16	11	12	13	14	0	0	0	0	...
D24	0	0	0	0	0	0	0	0	...
D32	0	0	0	0	0	0	0	0	...
D40	0	0	0	0	0	0	0	0	...
D48	0	0	0	0	0	0	0	0	...

オンライン(O) 選択範囲を読み出し(R)...

分割表示(G) 選択範囲を書込み(W)...

“Shift+カーソルキー”でも選択可!



書き込み対象を選んでね!

選択範囲を書込み

対象

- グローバルデバイス
- ローカルデバイス

プログラムファイル名
MAIN

※シーケンサに書き込まれているプログラムファイル名を指定してください。

OK キャンセル

レシピデータの一部の変更も手軽にできるね!

RUN 中書き込みでもできるからぜひ活用しよう!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第47回

NEW
Ver.Up
新機能

照合結果を不一致のみ表示!

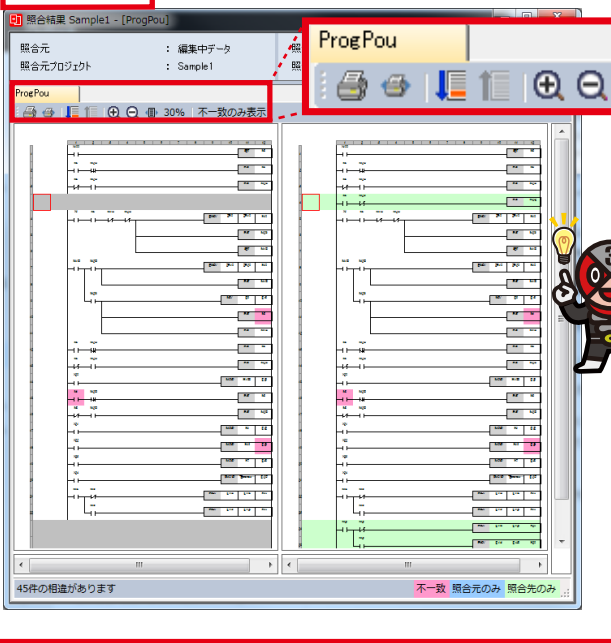


長いプログラムのおときは変更点だけを
まとめて見たいなあ…。

照合結果を見るときに不一致のみを表示
することができるようになったよ!



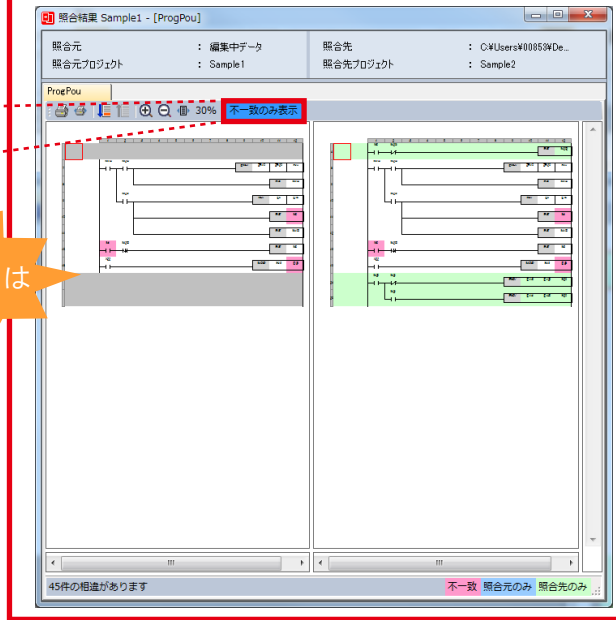
全体表示



切替ボタン

不一致のみ表示

不一致のみ表示



不一致のみ表示
時は切り替えボタンは
青色になるよ!



全体と不一致のみを簡単に切り替えられるし、修正
箇所だけを見ることができるから使いやすいね!

ラダープログラムだけでなく、FBD/LD や
STプログラムでも同様の機能が使えるよ!



不一致のみで印刷することもできるよ!
これで君もシーケンスアープ!





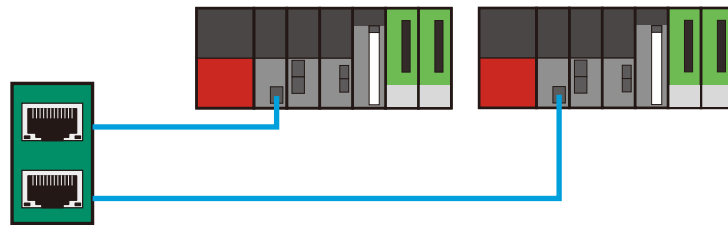
システム設計の頼れるヒーロー 教えて!GX3号

第48回

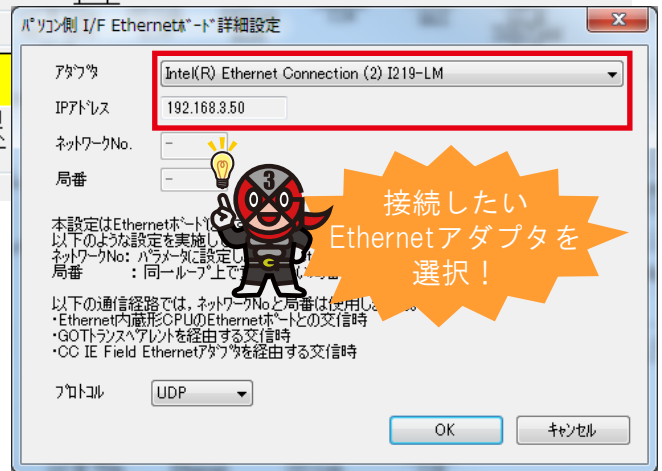
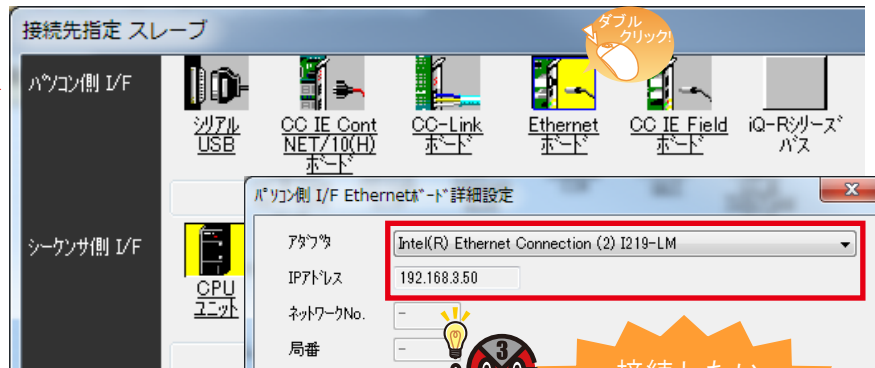
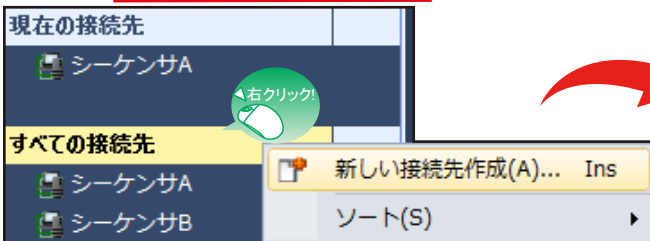
NEW
Ver.Up
新機能

Ethernetアダプタを切り替えて接続

Ethernet アダプタを直結接続のときにケーブルを差し替えなくてもシーケンサを切り替えられるようになったよ!



新しい接続先を追加



接続したい
Ethernetアダプタを
選択!



アダプタを変更して接続先を設定しておけば、接続先
ウィンドウで簡単に切り替えられるようになったんだね!

接続先の設定が GX Works3 で完結できるよ!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第49回

NEW

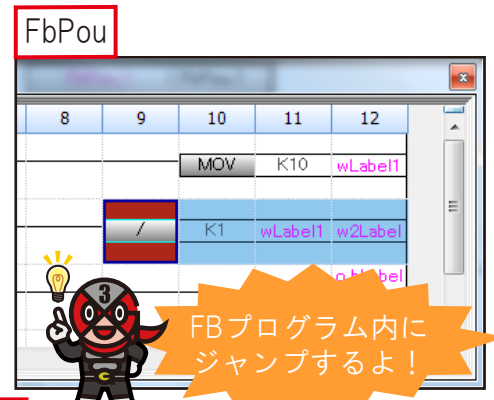
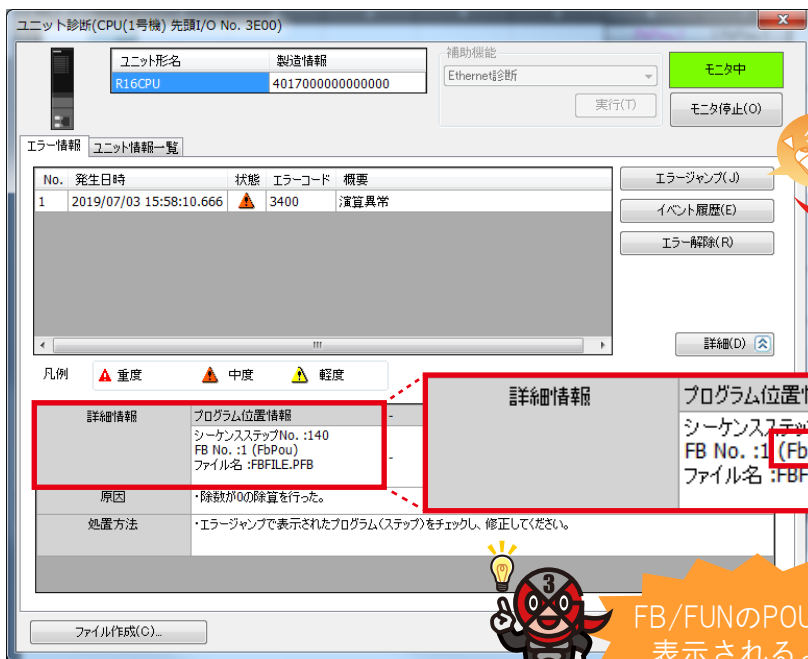
Ver.Up
新機能

エラー発生FBを診断で特定



FB ファイルで CPU エラーが発生したからエラー箇所を確認したいんだけど、どのFBか探すのが大変だなあ…。

ユニット診断機能からどのFB/FUN プログラムがエラーになっているか識別できるようになったよ!



FBプログラム内にジャンプするよ!



FB/FUNのPOU名が表示されるよ!



どのFBかすぐ探せるから助かるよ!

識別できたエラープログラムではエラー箇所は赤色になるからすぐ確認できるよ!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第50回

NEW
Ver.Up
新機能

ウォッチの表示形式を一括変更

右クリックから選択したデバイスの表示形式を2進数や16進数に一括で変えられるようになったよ!



ウォッチ1【ウォッチ中】

名称	現在値	表示形式	データ型
D0		10進数	ワード[符号付き]
D1		10進数	ワード[符号付き]
D2	45	10進数	ワード[符号付き]
D3	55	10進数	ワード[符号付き]

右クリック

- タイトルの変更(N)...
- ON/OFF表示(L)
- 選択したデバイス/ラベルの表示形式を変更
 - 2進数
 - 10進数
 - 16進数
- 選択したデバイスのデータ型を変更
- コメント編集(E)...

ドラッグ!

名称	現在値	表示形式	データ型
D0	H0019	16進数	ワード[符号付き]
D1	H001E	16進数	ワード[符号付き]
D2	H002D	16進数	ワード[符号付き]
D3	H0037	16進数	ワード[符号付き]



全デバイスの表示形式を変えたい時にすぐ変更できるから助かるよ!



デバイスだけでなくラベルでも変更できるよ!

データ型も一括で変更!

- タイトルの変更(N)...
- ON/OFF表示(L)
- 選択したデバイス/ラベルの表示形式を変更
- 選択したデバイスのデータ型を変更
 - ビット
 - ワード[符号なし]/ビット列[16ビット]
 - ダブルワード[符号なし]/ビット列[32ビット]
 - ワード[符号付き]
 - ダブルワード[符号付き]
 - 単精度実数
 - 倍精度実数
 - 文字列
 - 文字列[Unicode]
 - 時間
- コメント編集(E)...
- ファイルのインポート(E)...
- ファイルへエクスポート(Q)...

ウォッチ1【ウォッチ中】

名称	現在値	表示形式	データ型
D0	25	10進数	ワード[符号なし]/ビ...
D1	30	10進数	ワード[符号なし]/ビ...
D2	45	10進数	ワード[符号なし]/ビ...
D3	55	10進数	ワード[符号なし]/ビ...

リアルタイムモニタとの連携も捗るね!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!Gx3号

第51回

NEW

Ver.Up
新機能

連続貼り付けで配列要素もインクリメント

前に紹介した連続貼り付け機能がさらにパワーアップしたよ!
配列要素もインクリメントできるようになったんだ!

便利な機能



連続貼り付け

切り取り/コピーした回路内に存在するデバイスのデバイス番号をインクリメントしながら、現在のカーソル位置より連続で貼り付けます。

連続貼り付け数(1~99)(N): 個

ラベル名のインクリメント位置(L): 末尾の「数値」(b1Label1 >> b1Label1_1)

コメントを表示する。(C)

種別	コピー元	インクリメント後	インクリメント数
ローカルラベル	b10Label1[0]	>> b10Label1[1]	0
ローカルラベル	b10Label1	>> b10Label1	1
デバイス	X0Z0	>> X1Z0	1
デバイス	X0	>> X1	1
		>> Z0	0

多次元配列でもOK!

ローカルラベル	b20Label[0,0]	>> b20Label[1,2]	0
ローカルラベル	b20Label	>> b20Label	0
	0	>> 1	1
	0	>> 2	2

19999(10進数)です。
はインクリメントしません。

設定したラベルの付加情報を編集する。(G)

貼り付け方向(D):

配列要素[0]もインクリメント可能

「X0Z0」の「X0」もインクリメント可能

インクリメントの設定範囲は
-999999~999999に
拡大したよ!



修飾されるデバイスもインクリメント対象に
選べるようになったのか!

もちろん今までのインクリメント設定も選べるよ!

思ったとおりの回路が連続貼り付けで
作成できるようになったね!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!G×3号

第52回

NEW
Ver.Up
新機能

配列要素・ビット指定にコメント設定

ラベルコメントの設定で配列要素一つ一つにコメントが設定できるようになったよ!



ラベルエディタ

	ラベル名	データ型	Japanese/日本語(表示対象)
1	w10Error	ワード[符号付き](0..2)	エラー情報
2	wSample	ワード[符号付き]	
3	wTemp	ワード[符号付き]	ステータス

複数コメント表示設定(M)...

ラベルコメント設定を開く(G)

ショートカット

Alt + Enter

ラベルコメント

オプション(O)... <フィルタ>

ラベル名	Japanese/日本語(表示対象)
w10Error	エラー情報
└ w10Error[0]	駆動エラー
└ w10Error[1]	モニタエラー
└ w10Error[2]	その他エラー



右上に赤■のマークは要素ごとのコメントが設定されたことを表してるよ!

配列だけでなくビット指定ごとにコメントを設定することもできるんだ!



ラベルコメント

オプション(O)... <フィルタ>

ラベル名	Japanese/日本語(表示対象)
wTemp	ステータス
└ wTemp.0	起動状態
└ wTemp.1	停止状態
└ wTemp.2	エラー停止

要素ごとの用途がわかりやすくなったね!

ラベルコメントウィンドウを表示したままにすれば
選択したラベルとコメントを自動で表示してくれるよ!
これで君もシーケンスアープ!





NEW
Ver.Up
新機能

ラダーエディタでのコメント表示改善



ラベルの配列要素やビット指定のコメントを設定したけどラダー上で確認できるのかな？

もちろん表示できるよ！要素を定数で指定すると要素ごとのコメントが表示されるよ！



ラベルコメント	
オプション(O)... <フィルタ>	
w10Error	
ラベル名	Japanese/日本語(表示)
w10Error	エラー情報
+ w10Error[0]	駆動エラー
+ w10Error[1]	モータエラー
+ w10Error[2]	その他エラー

FMOV	K0	w10Error エラー情報	K3
MOV	K1	w10Error[D0] エラー情報	
MOV	K1	w10Error[2] その他エラー	

構造体（インスタンス）のコメント表示も変更できるようになったよ！使いやすいように変更してみてね！



<設定場所>

[その他のエディタ]→[ラベルエディタ共通]→
オプション [表示設定]→[構造体コメントの階層表示設定]

構造体コメントの階層表示設定	インスタンス、メンバ
区切り文字	末尾メンバのみ
拡張表示設定	先頭インスタンスのみ
階層表示エリアに表示する項目	インスタンス、メンバの順
動作設定	メンバ、インスタンスの順

ラダーエディタ

stMotor_1.wAccelerationTime
モータ-1.加速時間

要素やビット指定のコメントも表示できるからラダーがわかりやすくなったね！これで君もシーケンスアープ！





システム設計の頼れるヒーロー 教えて!GX3号

第54回

NEW
Ver.Up
新機能

クロスリファレンスのオプションが充実

クロスリファレンスのジャンプを実行した時、
カーソルの配置先が選べるようになったんだ!



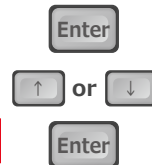
オプション	
プロジェクト	
プログラムエディタ	
その他のエディタ	
編集	
検索/置換	
共通項目	
クロスリファレンス	

動作設定	
検索結果の表示先	クロスリファレンス1
自動追従結果の表示先	クロスリファレンス2
自動追従したときの情報更新タイミング	常に更新する
ジャンプ実行時にカーソルをジャンプ先に配置する	いいえ
入力履歴からデバイス/ラベル名を表示	いいえ

「ジャンプ実行時にカーソルをジャンプ先に配置する」でいいえを選択



デバイス/ラベル	位置	回路記号
フィルタ条件	フィルタ条件	フィルタ条件
bLabel_1	0ステップ	↑↑
bLabel_11	1ステップ	-(-)
bLabel_2	2ステップ	↑↑
bLabel_12	3ステップ	-(-)
bLabel_3	4ステップ	↑↑



連続して
キー操作が
できるよ!

キー操作が続けられるから、回路の確認が素早くできるね

他にもオプション設定でラベルの
候補表示ができるようになったよ!

「入力履歴からデバイス/ラベル名を表示」でいいえを選択

クロスリファレンス1	
b	
b10Label1	BOOL(0..9)
bLabel_1	BOOL
bLabel_10	BOOL

候補一覧表示

用途に合わせてオプション設定で使い分けよう!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第55回

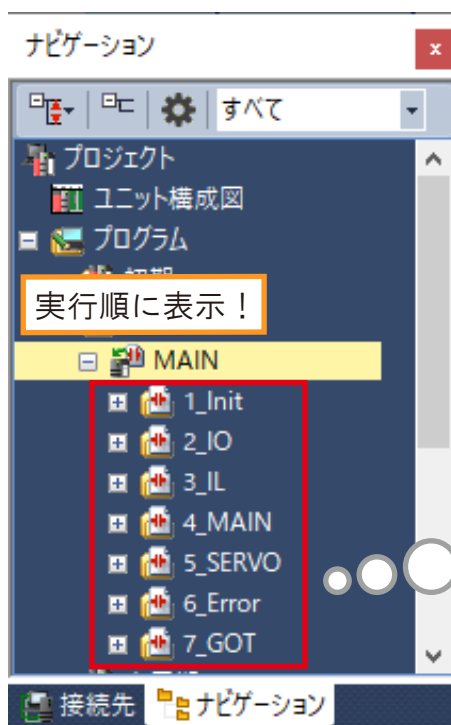
NEW
Ver.Up
新機能

プログラムをシーケンサから実行順で読出し

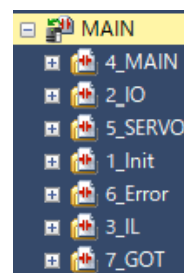
シーケンサから読み出した時、実行順に従ってプログラムを表示してくれるようになったよ!



読出し



以前は作成順に表示していたので手作業でソートを行う必要があった。



今までは作成順に表示していたのをデフォルトで実行順に表示してくれるようになったんだ!



実行順の方をよく使うから便利になったよ!

並び替える手間が無くなってより使いやすくなったね!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第56回

NEW
Ver.Up
新機能

エラー発生個所のFB階層情報



エラーが発生した原因を探したいけど、FBを使っているとどの順番でFBが使われているかわからないなあ…。

CPUエラー時のFB呼び出し順が分かるようになったよ！



ユニット診断(CPU(1号機) 先頭I/O No. 3E00)

ユニット形名	製造情報
R32CPU	33155409A0310151

補助機能
Ethernet診断

モニタ中
実行(T) モニタ停止(O)

エラー情報 ユニット情報一覧

No.	発生日時	状態	エラーコード	概要
1	2020/01/08 13:06:34.727	▲	H3400	演算異常

エラージャンプ(J)
イベント履歴(E)
エラー解除(R)

詳細(D)

凡例 ▲ 重度 ▲ 中度 ▲ 軽度

表示切替
 エラー詳細情報(A)
 FB階層情報(F)

ジャンプ先	プログラム/FB階層
223ステップ	ProgPou/FbPou_1/FbPou_1_1/FbPou2_1/FbPou3_1
FbPou3_1	ProgPou/FbPou_1/FbPou_1_1/FbPou2_1
FbPou2_1	ProgPou/FbPou_1/FbPou_1_1
FbPou1_1	ProgPou/FbPou_1
FbPou_1	ProgPou

FBの呼び出し順に該当回路にジャンプできる

ファイル作成(O)... 閉じる

FbPou3(MAIN/ProgPou/FbPou_1/FbPou_1_1/FbPou2_1/FbPou3_1) [FB] [LD] モニタ実行中 (読み取り専用) (24)ステップ

ステップ	1	2	3	4	5	6	7
6						MOV	W:Label2
7	U:Label1					SET	o:comp1
8							
9						v:count	w:count0

FbPou2(MAIN/ProgPou/FbPou_1/FbPou_1_1/FbPou2_1) [FB] [LD] モニタ実行中 (読み取り専用) (24)ステップ

ステップ	1	2	3	4	5	6	7
2	(o)					FbPou3_1	(FbPou3)
3						W:Label1	W:i_wLa...
4						W:Label2	W:i_wLa...



FBの呼び出し元にすぐたどり着けるから、原因調査が素早くできるね！

サブルーチン型のFBでこの機能が使えるよ！
FBのプロパティを確認してみよう！
これで君もシーケンスアープ！





システム設計の頼れるヒーロー 教えて!GX3号 第57回

NEW
Ver.Up
新機能

インラインSTの視認性向上!

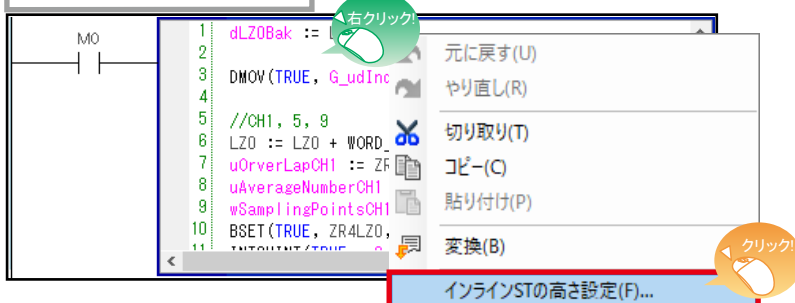


インライン ST を確認したいけど、ちょっと長くなるとスクロールが必要だから面倒だなあ。

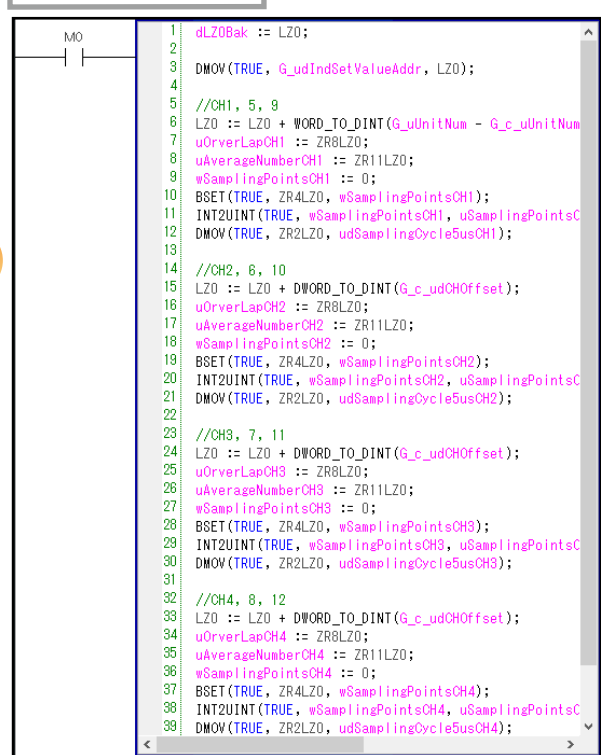
インライン ST の表示行数を変更できるようになったよ！
表示メニューもしくは右クリックメニューの「インライン ST の高さ設定」から変更してね！



インラインSTボックスを選択



表示行数を変更して表示



インラインSTの高さ設定

表示行数(B): 設定範囲(5~39)

設定対象(D):



「設定対象」を
変更して一括変更も
できるよ！



インライン ST にたくさん記述しても表示できる行数が増やせるから一目で確認できるね！

入力文字数も拡張しているので長いプログラムも
1つのインライン ST で作れるようになったよ！
これで君もシーケンスアープ！





システム設計の頼れるヒーロー 教えて!GX3号

第58回

NEW
Ver.Up
新機能

ST言語のテンプレートを見やすく表示

STプログラムで命令/FB/FUNのテンプレートを使うとき
引数を縦に並べられるようになったよ!



オプション

オプション		
プロジェクト	エラー箇所を表示する	はい
プログラムエディタ	アウトラインの有効化	はい
共通項目	対応する括弧を強調表示する	はい
ラダーエディタ	変更履歴を表示する	はい
STエディタ	編集時の動作	
FB/LDIエディタ	インデント機能を有効にする	はい
SFC図エディタ	タブ文字数	4
その他のエディタ	貼り付け時自動フォーマットを行う	はい
編集	テンプレート表示の引数を縦に並べる	はい
検索/置換	照合	
モニタ	照合設定	
オンライン	詳細表示(リスト形式)を最小化する	はい
変換	テンプレート表示の引数を縦に並べる	
	命令、FB/FUNのテンプレート表示を行ったときに、引数を縦に並べるかどうかを選択します。	



「編集時の動作」→
「テンプレート表示の
引数を縦に並べる」
にて変更できるよ!

インラインST

```
XO
1 ZP_REMTO
```

テンプレート表示

Ctrl + Alt + F1



```
XO
1 ZP_REMTO(
2 ?BOOL_EN?,
3 ?ANY16_OR_STRING_SINGLE_In?,
4 ?ANY16_s1?,
5 ?ANY16_s2?,
6 ?ANY16_s3?,
7 ?ANY16_s4?,
8 ?ANY16_s5?,
9 ?ANY16_n?,
10 ?ANYBIT_ARRAY_d?
11 );
```



僕はこっちのスタイルの方が好きだから「規定値に設定」して使ってみよう。

自分のプログラミングスタイルに合わせて変更しよう!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!Gx3号

第59回

NEW

Ver.Up
新機能

変更した現在値をイベント履歴で確認

イベント履歴で現在値変更の操作も履歴に残せるようになったよ!



装置の動きが変わったときに現在値変更があったかどうかを確認できるのか!

CPUパラメータ

RAS設定

- スキャンタイム監視時間(WDT)
- コンスタントスキャン設定
- 異常検出設定
- 異常検出時のCPUユニット動作
- LED表示設定
- イベント履歴設定

プログラム設定

SEC設定

イベント履歴設定

保存先	メモ리카ード
1ファイルあたりの保存容量設定	128 K/バイト
デバイス/ラベル操作の保存	保存する

発生するイベントが多くなるからSDカードに保存するのが便利だよ!

この機能を使うにはCPUパラメータで「デバイス/ラベル操作の保存」を「保存する」に設定しよう!

イベント履歴

No.	発生日時	イベント種別	状態	イベントコード	概要	発生元ユニット	先頭I/O No.
00001	2020/06/30 10:14:20.679	オペレーション	ⓘ	H24800	ワード単位デバイス書き込み(1点)	R16CPU	3E00
00002	2020/06/30 10:13:49.333	オペレーション	ⓘ	H24100	動作状態の変更(RUN)	R16CPU	3E00
00003	2020/06/30 10:13:48.253	システム	ⓘ	H00400	電源ON/RESET解除	R16CPU	3E00
00004	2020/06/30 10:13:44.332	オペレーション	ⓘ	H24101	動作状態の変更(STOP)	R16CPU	3E00
00005	2020/06/30 10:13:35.296	オペレーション	ⓘ	H24100	動作状態の変更(RUN)	R16CPU	3E00
00006	2020/06/30 10:13:35.296	オペレーション	ⓘ	H24001	リモート操作要求受付	R16CPU	3E00
00007	2020/06/30 10:13:13.555	システム	ⓘ	H00420	イベント履歴ファイル生成	R16CPU	3E00

凡例
 ▲ 重度
 ⚠ 警告
 ▲ 中度
 ⓘ 情報
 ▲ 軽度

詳細情報

操作元情報	デバイス名情報	書き込み値
接続ポート :USB	デバイス名:D1	0064

原因
ワード単位デバイスへの書き込みを行った。

処置方法
-

デバイス名や書き込み値を確認できるね!

これでトラブルが発生したときに早く状況把握ができるね!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!Gx3号

第60回

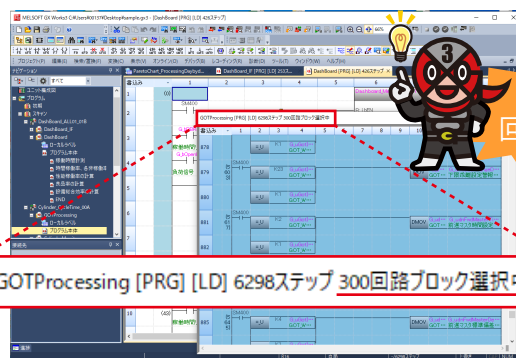
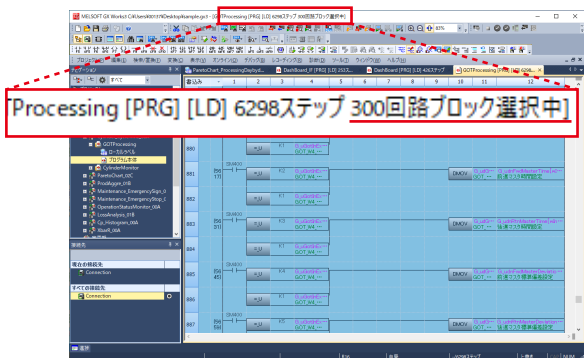
NEW

Ver.Up
新機能

ラダーのコピーと行挿入の操作性UP!

ラダーをコピーするときを選択している回路数が確認できるようになったんだ!

便利な機能



選択中の回路数を表示

100回路ブロック選択中]

200回路ブロック選択中]

300回路ブロック選択中]



タイトルバーでわかるから便利だね!

ラダーの行 / 列挿入、行 / 列削除、上書き / 挿入モード切替を任意のキーに割り付けられるようになったよ!



ショートカットキー設定

変更対象(D): ラダー

コマンド	キ-1	キ-2	キ-3	既定
連続貼り付け	Ctrl+Alt+V			
挿入して貼り付け				Alt+Ins
上書き/挿入モード切替	Shift+H			Ins
回路編集開始時の状態に戻す	Ctrl+Shift+Z			
行挿入	Shift+I			Shift+Ins
行削除	Shift+U			Shift+Del
列挿入	Shift+K			Ctrl+Ins
列削除	Shift+J			Ctrl+Del
NOP一括挿入				
NOP一括削除				

ツールメニューからショートカットキーを変更できるよ!



"Insert" キーがないキーボードでも行 / 列挿入がしやすくなるね!

よく使う操作が改善されて、さらにプログラムが作成しやすくなったよ!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第61回

NEW

Ver.Up
新機能

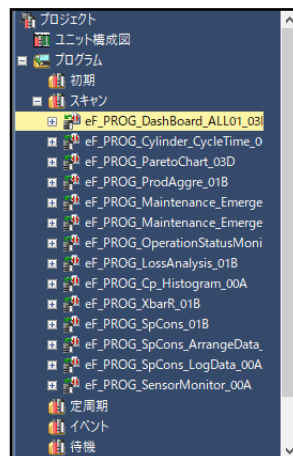
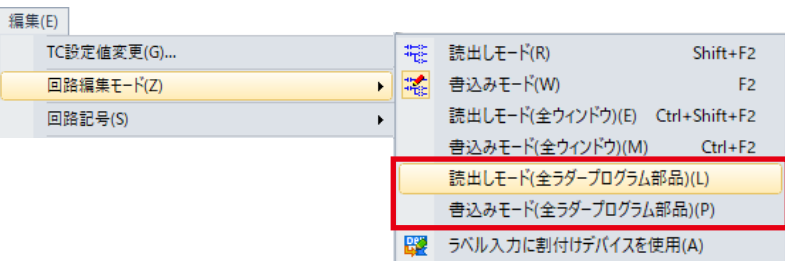
ラダープログラムの編集モードを一括切り替え!



誤操作防止で読出しモードにしておきたいけど
ひとつひとつ変更するのは面倒だなあ。

ラダープログラムの読出しモード / 書き込みモードを
一括で切り替えられるようになったよ!

そんなときは

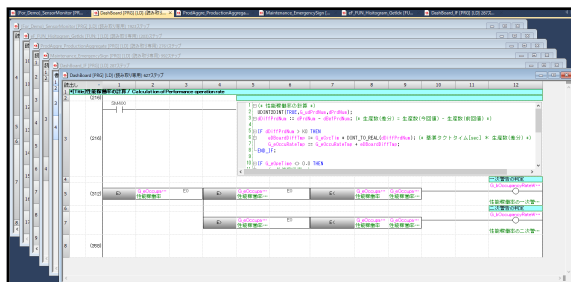
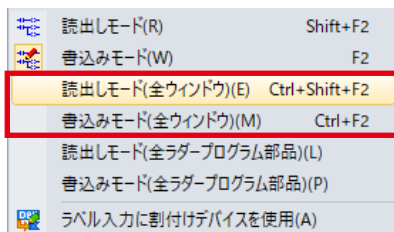


全ての
ラダープログラムを
一括で切り替え



よく使う機能だからショートカットキーを
割り当てて使ってみるね!

GX Works2 と同様の一括切り替え
操作もできるようになったよ!



開いている
ラダープログラムを
一括で切り替え

誤操作防止が簡単にできるようになったよ!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!Gx3号 第62回

NEW Ver.Up 新機能 ラベルの配列設定と行削除の操作性UP!

ラベルエディタで多次元配列の設定が簡単にできるようになったよ!



ラベルエディタ

ラベル名	データ型	
1 eF_FB_Vibra	eF_FB_Vibration	VAR
2 bENDInitFB	ビット	VAR
3 bOKInitFB	ビット	VAR
4 bErrInitFB	ビット	VAR
5 uErrIdInitFB	ワード符号なし	VAR



データ型選択

対象(L)

データ型(O)

- ワード符号なし/ビット列[16ビット]
- ダブルワード符号なし/ビット列[32ビット]
- ワード符号付き
- ダブルワード符号付き
- 単精度実数
- 倍精度実数
- 浮動小数
- 文字列[32]
- 文字列[Unicode][32]
- ビット
- タイム
- カウンタ
- ロングカウンタ
- 種別名
- ロング種別名
- ロングタイム

型分類

- 基本データ(S)
- 構造体(D)
- ファンクションブロック(E)

配列要素

配列(A)

要素数(1次元)(E)

要素数(2次元)(M)

要素数(3次元)(N)

OK キャンセル

1次元配列

要素数(1次元)(E)

データ型

2次元配列

要素数(1次元)(E)

要素数(2次元)(M)

データ型

3次元配列

要素数(1次元)(E)

要素数(2次元)(M)

要素数(3次元)(N)

データ型



要素数を設定するだけだから楽だね!

ラベルエディタの制約が緩和されて行削除がしやすくなったよ!



フィルタ条件

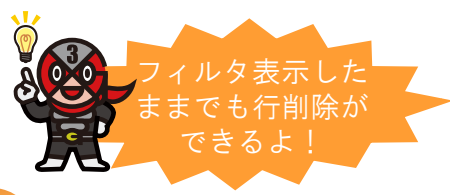
フィルタ条件

簡易表示(O) 表示設定(S) チェック(I)

ラベル名	データ型	クラス
1 eF_FB_Vibration Analysis_Init_R_00	eF_FB_Vibration Analysis_Init_R_00A	
2 bENDInitFB	ビット	
3 bOKInitFB	ビット	
4 bErrInitFB	ビット	
5 uErrIdInitFB	ワード符号なし/ビット列[16ビット]	
6 uIdnSetValueAddrInitFB	ダブルワード符号なし/ビット列[32ビット]	
12 bInitEN	ビット	

元に戻す(U) やり直し(R) 切り取り(T) コピー(C) 貼り付け(P) 削除(D) すべて選択(A) 複数コメント表示設定(M)...

行追加(1つ前)(N) 行追加(1つ後)(W) **行削除(E)**



フィルタ表示したままでも行削除ができるよ!



フィルタで見つけてすぐ削除できるのはいいね!

ラベルプログラムの作成がさらにしやすくなったよ!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第63回



MELSEC-A/QnAプロジェクトを簡単活用

「MELSEC-A/QnA→MELSEC iQ-R 変換サポートツール」を紹介するよ！1回の操作でMELSEC-A/QnAのプロジェクトをGX Works3で使えるようにできるんだ！



今までの変換イメージ



ツールによる変換イメージ



MELSEC-A/QnA→MELSEC iQ-R変換サポートツール

A/QnAシリーズ

iQ-Rシリーズ

見直し情報リスト



MELSEC-A/QnA→MELSEC iQ-R変換サポートツール			
変換結果		プロジェクト名	プロジェクト名
変換前	C:\Users\00137\Desktop\AQcnProject	ACRUPProject	
変換後	C:\Users\00137\Desktop\AQcnProject\CHANGE	test1	
変換結果表示		変換結果表示	
MADNプログラム	13		一括表示
変換されなかった特殊リレー特殊レジスタ	9	デバイス番号	ステップ番号
変換されなかった命令	4	命令名	ステップ番号
変換しが必要は特殊機能ユニット用プログラム	0	特殊機能ユニット名	ステップ番号
追加プログラム	22		一括表示
変換されなかった特殊リレー特殊レジスタ	15	デバイス番号	ステップ番号
変換されなかった命令	7	命令名	ステップ番号
変換しが必要は特殊機能ユニット用プログラム	0	特殊機能ユニット名	ステップ番号

見直し情報リストには置換えに必要な資料へのリンクとプログラムの該当箇所が記載されているよ！

リスト内にあるリンク先を見て作業を進めてみるよ！

三菱電機 FA サイトからダウンロードして使ってね！
これで君もシーケンスアープ！



システム設計の頼れるヒーロー 教えて!GX3号

第64回

NEW
Ver.Up
新機能

照合結果の差分をマージ!



別のプロジェクトで修正した部分を編集中のプログラムに反映したいけど良い方法はないかなあ…。

そんなときは



照合結果から不一致箇所を選択してマージ
することができるようになったよ!

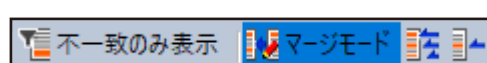
1 照合結果ウィンドウでマージモードに設定



or

ショートカット

Ctrl + E



[マージモード]
ボタンが青色のときに
マージできるよ!

2 不一致箇所を選択してマージ

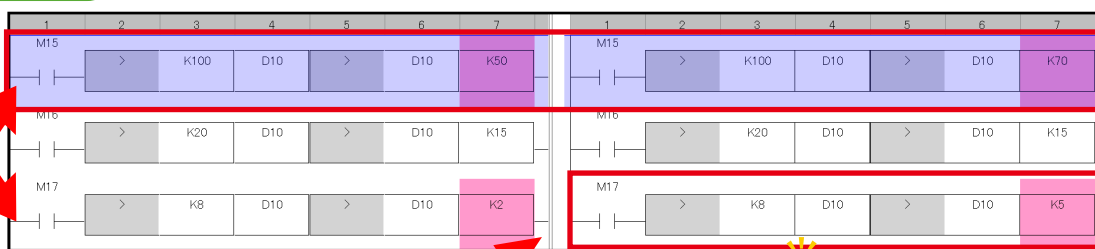
不一致箇所の選択



or

ショートカット

Alt + ↑ ↓



マージ

or

ショートカット

Alt + ←



照合先の
プログラムを反映

一括で差分をマージすることもできるんだ!



or

ショートカット

Alt + A



ラダーだけでなく ST プログラムでも
マージ機能は使用できるよ!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第65回

NEW
Ver.Up
新機能

イメージクリップでラダーをコピー



説明資料にラダー図を貼り付けて作成したいけど
必要な回路だけを貼り付けたいなあ…。

イメージクリップ機能により選択回路のみを
画像としてコピーすることができるんだ!



クリップボードに
選択回路をコピー!



欲しい回路のみをコピーできるから
加工の手間がいらないね!

縦に長い回路も一度にイメージクリップできるよ!



これでドキュメントを簡単に作成できるね!
イメージクリップで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!Gx3号

第66回

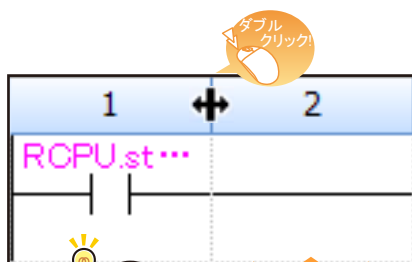
NEW
Ver.Up
新機能

ラダーエディタの列幅を簡単調整

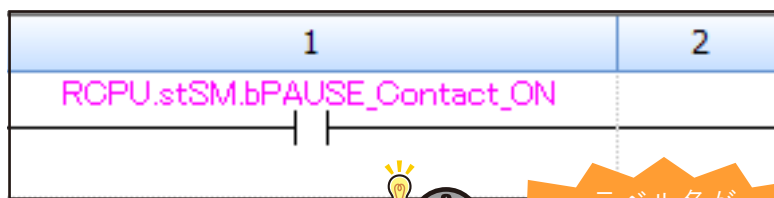


ラベル名が長いとラダーエディタの表示が省略されて見えないや。1つずつドラッグ操作で列幅を広げるのは煩わしいなあ。

簡単操作でラベル名に合わせて自動調整してくれるようになったよ!



列の境界線でダブルクリック!



ラベル名が省略されず表示されるように自動調整

ダブルクリックしたときの動作はオプションで変更できるよ!
以前と同じ動作にするには"固定幅に設定する"に設定しよう!



オプション	
プロジェクト	編集時の動作
プログラムエディタ	回路編集モード(読出し、書き込み、モニタ、モニタ(書き込み)
共通項目	横線の入力、削除画面の初期値を「1」にする
ラダーエディタ	横線入力、削除時に接続点(命令、縦線分岐)で停止
STIエディタ	マウスによる回路編集を有効にする
FBD/LDIエディタ	デバイス番号/ラベル名のインクリメント対象
SFC図エディタ	マウスによる列幅調整時の動作
その他のエディタ	FB/FUN
	動作設定

'Ver.1.070Y' からダブルクリックの動作が変わったよ!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第67回

NEW
Ver.Up
新機能

デバイスメモリをファイル名でフィルタ表示



シーケンサへの書込みのときデバイスメモリと
ファイルレジスタが見分けづらいなあ ...。

オプションでフィルタできるようになったから
見やすくなったよ!



オプション

モータ	<input type="checkbox"/> 指定した名前から始まるデバイスメモリのみ表示する	はい
オンライン	名前	MAIN
シーケンサへの書込み	<input type="checkbox"/> 指定した名前から始まるファイルレジスタのみ表示する	はい
シーケンサとの照合	名前	ZR
プログラム復元情報		



指定した名前から
始まるデータを表示

オンラインデータ操作

ユニット形名/データ名	<input type="checkbox"/>
<input type="checkbox"/> デバイスメモリ	<input type="checkbox"/>
MAIN	<input type="checkbox"/>
MAIN1	<input type="checkbox"/>
ZR	<input type="checkbox"/>
ZR1	<input type="checkbox"/>
<input type="checkbox"/> ファイルレジスタ	<input type="checkbox"/>
MAIN	<input type="checkbox"/>
MAIN1	<input type="checkbox"/>
ZR	<input type="checkbox"/>
ZR1	<input type="checkbox"/>



ユニット形名/データ名	<input type="checkbox"/>
<input type="checkbox"/> デバイスメモリ(フィルタ中)	<input type="checkbox"/>
MAIN	<input type="checkbox"/>
MAIN1	<input type="checkbox"/>
<input type="checkbox"/> ファイルレジスタ(フィルタ中)	<input type="checkbox"/>
ZR	<input type="checkbox"/>
ZR1	<input type="checkbox"/>



必要なデータだけを表示するから見やすくて、
間違えることもなくなるね!

デバイスメモリ、ファイルレジスタの
ファイル名を工夫して活用してね!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第68回

NEW
Ver.Up
新機能

シミュレーション機能を1クリックで起動

シミュレーション機能を1クリックで起動できるようになったよ!



シミュレーション開始(S)



書き込み
操作を省略!

~~オンラインデータ操作~~

~~表示(D) 設定(S) 関連機能(U)~~

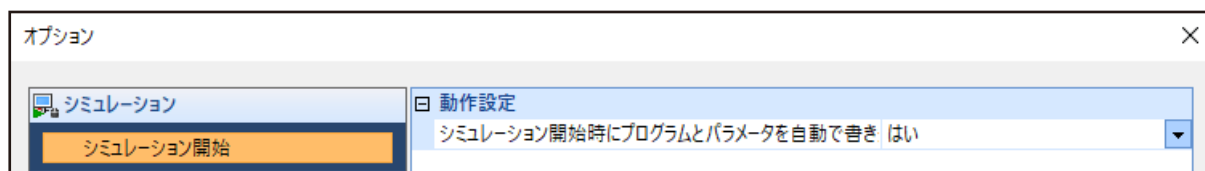
~~書き込み~~



書き込みの手間がなくなるのか、便利だね!
書き込まれるデータは決まってるのかな?

"パラメータ+プログラム"が書き込まれるよ!
オンライン操作のお気に入り設定から変更できるよ!

オプションで以前のようにファイルを選択することもできるよ!



いつも使う設定があればお気に入り設定しておこう!
これで君もシーケンスアープ!



システム設計の頼れるヒーロー 教えて!GX3号

第69回

NEW
Ver.Up
新機能

ウォッチ登録の自動判定を改善!



ウォッチウィンドウへ登録するのは便利だけど、あとで手直ししなくちゃいけない時があるんだよね。配列とか。

バージョンアップでウォッチ登録後の手直しが不要になったよ!今まで通り範囲選択して、ウォッチウィンドウへ登録してみよう!



1	2	3	右クリック!	ブックマーク(M) F8プロパティ(Y)	クリック!
D=	sample[D100]	D150		ウォッチウィンドウへ登録(H)	
				デバイス一括置換へ登録(A)	

ウォッチウィンドウ

名称	現在値	表示形式	データ型
D100	1	10進数	ワード[符号付き]
sample		-	ダブルワード[符号付き][2]
[0]	50	10進数	ダブルワード[符号付き]
[1]	100	10進数	ダブルワード[符号付き]
D150	100	10進数	ダブルワード[符号付き]



sample[D100]が、それぞれの要素で登録されるんだね。すぐにウォッチができるから楽だなあ。

デバイスのデータ型も、命令に合わせて登録されるようになったよ。



データ型に合わせて自動で登録してくれるんだ!ウォッチがさらにしやすくなったね。

便利になった登録で、効率よくウォッチをはじめよう!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号 第70回



特徴で検索してユニット追加



ナビゲーションウィンドウから新しいユニットを追加するけど、種別や形名を選ぶのが大変だなあ。



新規ユニット追加画面で、特徴や形名の一部を入力して検索できるようになったよ。関連するユニットが検索されるんだ。

新規ユニット追加

TSN

ユニット選択

ユニット種別 シーケンサCPU

ユニット形名 R00CPU

局種別

詳細設定



ユニット検索

TSN

ユニット種別	ユニット形名	特徴
モーションユニット	RD78G4	4軸 CC-Link IE TSN
モーションユニット	RD78G8	8軸 CC-Link IE TSN
モーションユニット	RD78G16	16軸 CC-Link IE TSN
モーションユニット	RD78G32	32軸 CC-Link IE TSN
モーションユニット	RD78G64	64軸 CC-Link IE TSN
モーションユニット	RD78GHV	128軸 CC-Link IE TSN
モーションユニット	RD78GHW	256軸 CC-Link IE TSN
ネットワークユニット	RJ71GN11-T2	CC-Link IE TSN



「TSN」など特徴で検索してみよう!

ユニットの形名はわかるけど、ユニット種別に迷ったときにも使えるよ!



これなら簡単にユニットを見つけられるね!

形名がわからないときは、いろんなキーワードで試してみてね。これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第71回



標準化でプログラムを読みやすくしよう!



これまで1本のラダープログラムで制御してきたけど、若い人たちにも引き継いでいかないといけないし……。見やすいプログラムも考えないとな。

よく使うプログラムを標準化してみるのはいかがでしょうか？標準化とは、社内やチーム内で使うプロジェクトの基本的なつくりを統一しておくことだよ。

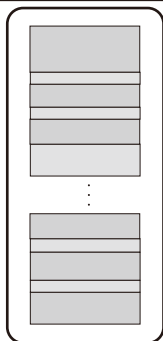


今使っているプログラムを標準化するには何から始めたらいいんだろう？

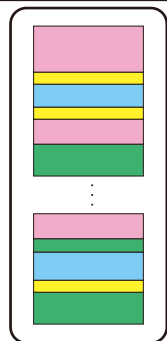
まずは構造化の概念を取り入れて、プログラムの分け方を統一してはどうか。他のメンバーが見てもどこにどんな処理があるかわかりやすくなるよ。



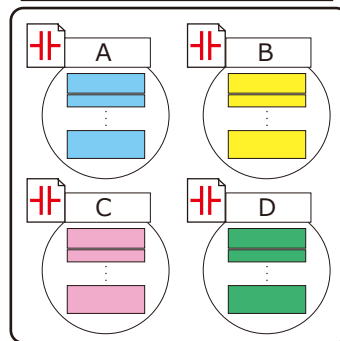
1本のラダープログラム



目的ごとに区切る



目的ごとにまとめる



構造化はこのように、目的（処理や機器、作業分担など）ごとにプログラムを区切り、まとめることだよ。



考え方はわかったけど、どうやればいいの？



次回から、GX Works3の機能と合わせて紹介していくよ！





システム設計の頼れるヒーロー 教えて!GX3号

第72回



ステートメント分割で標準化



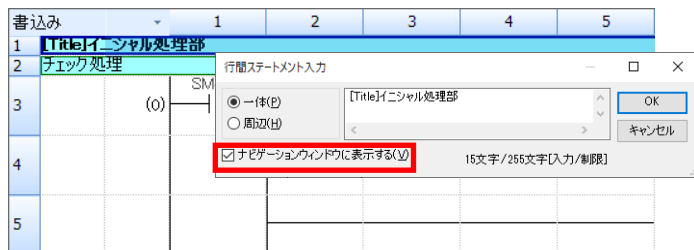
前に作ったプロジェクトがあるんだけど、次の案件ではこのプロジェクトをもっと見やすくしたいな。

まずは、ステートメントでプログラムを区切ることから始めてみよう。



前にステートメントの入れ方は教えてもらったね。

区切った中で、同じような処理がないか探してみよう。大きくグループ分けしたステートメントは、ナビゲーションウィンドウに表示させよう！



ステートメントを見れば、プログラム全体の流れがわかるんだね！

少しずつ、自分たちのスタイルに合ったステートメントの作り方を探していこう。
これで君もシーケンスアープ！





システム設計の頼れるヒーロー 教えて!GX3号 第73回



プログラムファイル・POU分割で標準化

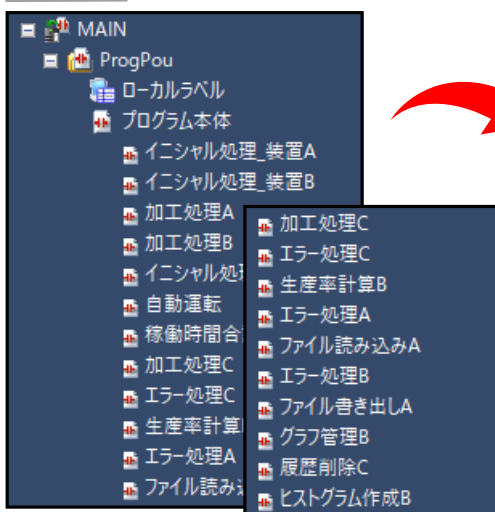


プログラムをステートメントで区切って整理したら、全体像が見えてきたよ。同じ装置の処理があちこちに見つかったんだ。これをもっと整理したいなあ。

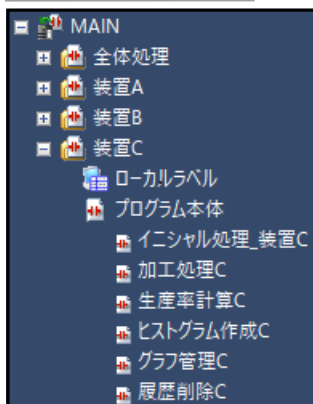
役割ごとに整理できるから、その装置の処理をプログラムファイルやPOUでまとめてみてはどうか？(ケース1)



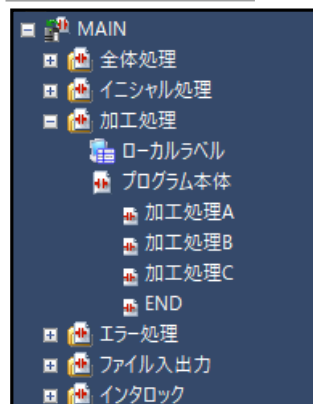
分割前



ケース1
装置ごとに分割



ケース2
工程ごとに分割



他にも、工程単位で分割することもできるよ！(ケース2)



どの分け方が一番良いのかなあ？

システムのよくある変更を意識して分割してみよう。装置単位で切り換わるなら装置単位で、工程単位での追加や削除がよくあるのなら工程単位で分割すると良いと思うよ。

プログラムを分けた後は、プログラム設定とプログラムファイル設定で実行順を整理しよう！これで君もシーケンスアープ！





システム設計の頼れるヒーロー 教えて!GX3号

第74回



ラベルを使って標準化!



プログラミングの話をしていてラベルや、変数って言葉を聞いたんだけど、それってどんなものなの?

ラベル名とデータ型を決めて使うものなんだ。例えばこんなメリットがあるよ。

そんなときは



ラベルを使うメリット

- ・名前から役割や用途がわかる。
- ・プログラムの流用時に、デバイスの時のようなアドレスの重複を気にしなくていい。
- ・装置の拡張時に、簡単にラベルの追加拡張ができる。
例えば、レシピデータ (D0~D199) を拡張するためには予約領域が必要…。
ラベルだと配列の要素数を増やすだけで拡張可能! *1 etc...

*1: 追加拡張の例

デバイス名	コメント(表示対象)
+ D193	レシピ194

デバイスの場合、拡張するにはあらかじめ予約領域が必要…。

+ D196	レシピ197
+ D197	レシピ198
+ D198	レシピ199
+ D199	レシピ200
+ D200	将来拡張用の予約領域
+ D201	将来拡張用の予約領域
+ D202	将来拡張用の予約領域
+ D203	将来拡張用の予約領域
+ D204	将来拡張用の予約領域
+ D205	将来拡張用の予約領域
+ D206	将来拡張用の予約領域

ラベルを使えば配列の要素数を変えるだけで拡張可能!

ラベル名	データ型	コメント(表示対象)
wRecipe	ワード[符号付き(0..199)]	レシピデータ

ワード[符号付き(0..299)]

ラベルを使ってみたいけど、デバイスを全部ラベルに置き換えるのは大変そうだなあ…。



前回までにプログラムを分けたよね。よく流用するところや、拡張する可能性のあるところからラベル化するのはどうかな!



次回も引き続きラベルについて紹介するよ! これで君もシーケンスアープ!





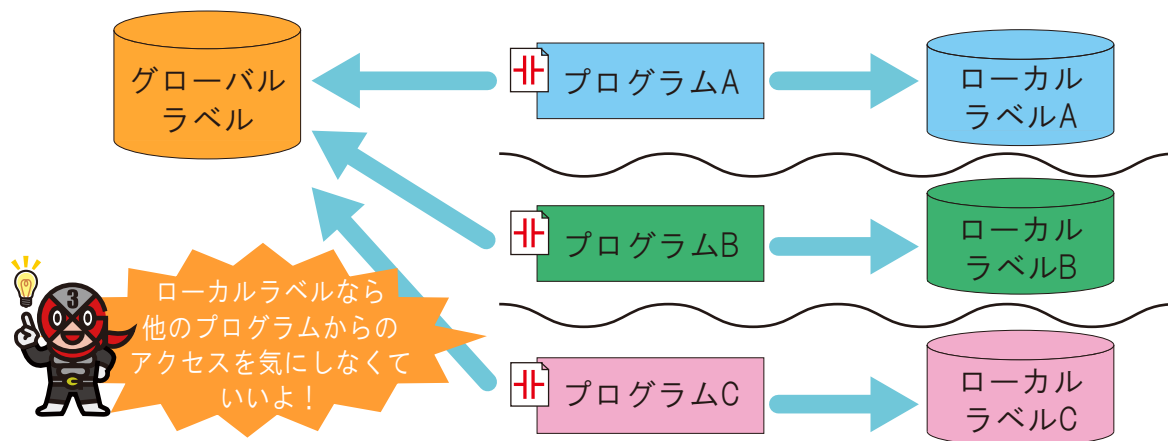
システム設計の頼れるヒーロー 教えて!GX3号

第75回



続・ラベルを使って標準化!

今回もラベルについて紹介するよ!ラベルにはグローバルラベルとローカルラベルの2種類があるんだ。ローカルラベルを使えば、流用しやすいプログラムが作れるよ!



他にもラベル名を工夫すると、プログラムがわかりやすくなるよ。データ型や使う場所がわかるような接頭語を付ける場合が多いね。装置に合ったラベルの命名規則を考えてみよう!



グローバルラベル			
ラベル名	データ型		クラス
G_bFlag	ビット	...	VAR_GLOBAL
G_wProduct	ワード[符号付き]	...	VAR_GLOBAL

ローカルラベル			
ラベル名	データ型		クラス
dAddition	ダブルワード[符号付き]	...	VAR
eVoltage	単精度実数	...	VAR

例えば	G: グローバルラベル
	b: ビット
	w: ワード[符号付き]
	d: ダブルワード[符号付き]
	e: 単精度実数
	etc...



ラベルは設計段階の準備が大事だね!

最初の設計をしっかりとっておけば、その後の流用やメンテナンスがしやすくなるよ!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第76回

NEW
Ver.Up
新機能

プログラム更新前のラベルメモリをバックアップ



操業の合間に全変換・再割付が必要な大きいプログラムの更新をすることになったんだけど、プログラムの更新後もラベルの値を維持して再開できないかな？

ラベルメモリを読み出して、再度書き込みができるようになったよ！更新前に、CPUメモリ操作で「ラベルメモリからの読出し」をしよう！



The screenshot shows the 'CPUメモリ操作' (CPU Memory Operation) window. In the '関連機能(L)' (Related Functions) menu, 'CPUメモリ操作(M)...' is selected. A red arrow points to the 'CPUメモリ操作' window, and another red arrow points to the 'ラベルメモリの読出し/書き込み' (Label Memory Read/Write) dialog box. In the dialog box, the '操作' (Operation) section has 'ラベルメモリからの読出し(A)' (Read from Label Memory) selected with a radio button. A red arrow points to this option. The '値のクリア(C)' (Clear Values) button is also highlighted with a red box and a callout 'ラベルメモリの読出し/書き込み(L)'. The '実行(E)' (Execute) button is visible at the bottom.

プロジェクトを書き込んでシーケンサをリセットした後「ラベルメモリへの書き込み」で値を書き込んで再開してね

再割付をしても、その前のラベルの値から再開できるんだ！

必要に応じて、デバイスメモリやファイルレジスタの読み書きも忘れないでね。
これで君もシーケンスアープ！



システム設計の頼れるヒーロー 教えて!GX3号

第77回

NEW
Ver.Up
新機能

データフロー解析から波形を表示!



オフラインモニタ中にデータフロー解析の画面でモニタができるから、デバッグがしやすいね!

さらに連携機能を強化したよ!
データフロー解析から GX LogViewer の波形データがすぐに見られるようになったんだ。



範囲選択して
「波形表示」で
GX LogViewer と連携!



モニタで値がおかしいとあたりを付けたラベル・デバイスの値の波形を確認しながらデバッグできるよ。



データフロー解析と合わせて、状況把握に役立ててね!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!Gx3号

第78回

NEW
Ver.Up
新機能

パラメータ設定操作をカスタマイズ!

パラメータ設定画面でダブルクリックしたときの動作を選べるようにしたよ!
選択項目の一覧を表示できるようになったよ。



設定項目	項目	設定
シンプルCPU通信設定	シンプルCPU通信使用有無	使用しない
	シンプルCPU通信設定	使用しない
時刻設定	時刻設定 (SNTPクライアント)	使用する
		使用しない



自分の好みに合わせてカスタマイズできていいね!

以前の操作性がいい場合は、オプションから変更できるよ。



オプション	
プロジェクト	動作設定
プログラムエディタ	プルダウンリストのダブルクリック時の動作
その他のエディタ	リストを表示する
編集	登録順に設定を切り替える
検索/置換	リストを表示する
パラメータ	プルダウンリストのダブルクリック時の動作
共通項目	ダブルクリック時に設定が変わるプルダウンリストについて、ダブルクリック時の動作を選択します。次回パラメータ設定画面を開いた場合に適用されます。
ネットワーク構成設定	・「登録順に設定を切り替える」を指定すると、プルダウンリストの登録順に設定を切り替えます。
モータ	・「リストを表示する」を指定すると、プルダウンリストを表示します。

好みの動作を規定値に設定してね。
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第79回



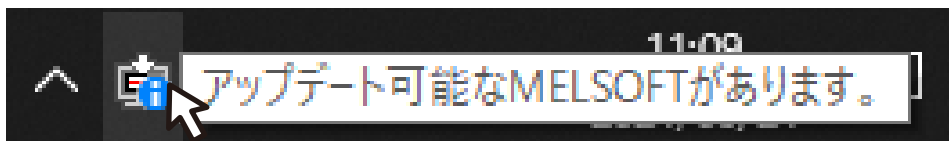
GX Works3を自動ダウンロード!

MELSOFT Update Manager を紹介するよ!
GX Works3 のアップデートがあると、
ポップアップで通知がくるんだ。



三菱電機 FA サイトにアクセスしてツールの
更新を毎回確認する手間がなくなるんだね!

いつでもタスクトレイの状態を確認できるよ。



更新があったときにインストーラを自動で
ダウンロードするかを設定で選ぶこともできるんだ。



まだインストールしていない人は、
三菱電機 FA サイトからダウンロードして使ってね!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第80回

NEW

Ver.Up
新機能

データフロー解析でモニタ値を表示!

データフロー解析結果でモニタ値を表示できるようになったよ! フロー図で値を確認してみよう。

便利な新機能



ツールバー

or



デバイスとラベルのモニタ値を表示できるんだ! フロー図内でビットの状態や数値も一緒に確認できるから便利だなあ。

スマートタグで表示形式を変更できるよ。



データ型を変更

進数表示を変更

モニタを活用してデバッグするとき役に立ってね!
これで君もシーケンスアープ!





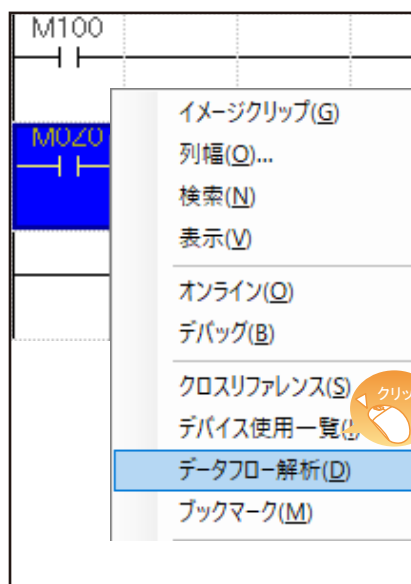
システム設計の頼れるヒーロー 教えて!Gx3号

第81回

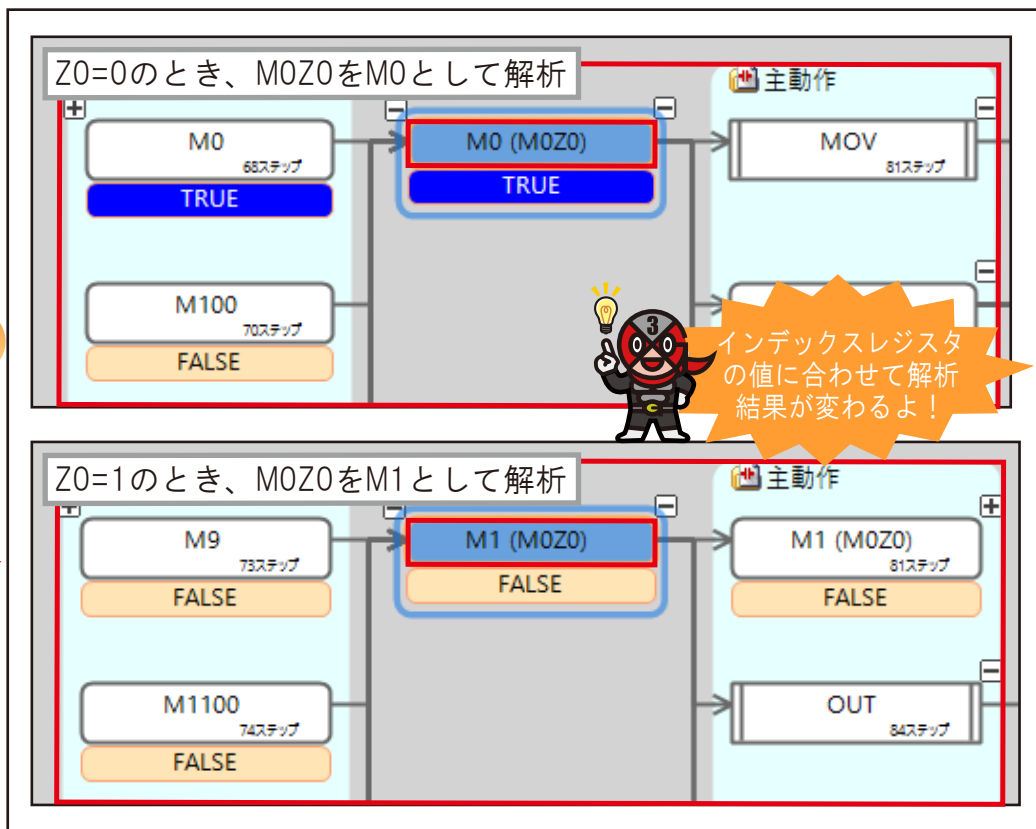
NEW
Ver.Up
新機能

オフラインモニタでインデックスレジスタ対応

オフラインモニタがもっと便利になったよ！
オフラインモニタ中のデータフロー解析で、インデックスレジスタの値に応じた解析ができるようになったんだ！



データフロー解析結果



今まではインデックスレジスタの部分だけ別に解析が必要だったけど、これなら一回で簡単に解析できるね。

新しくなったオフラインモニタとデータフロー解析で装置のトラブル解析をさらに効率化しよう！
これで君もシーケンスアープ！





システム設計の頼れるヒーロー 教えて!GX3号

第82回

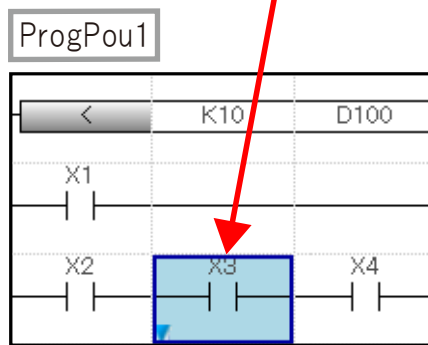
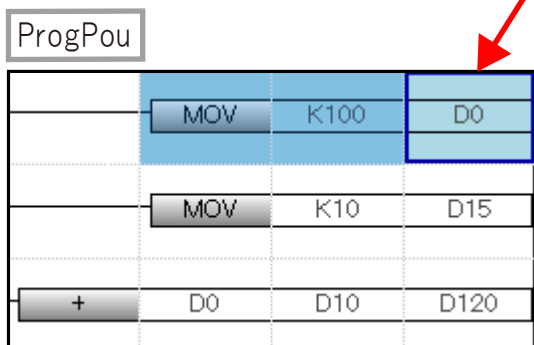
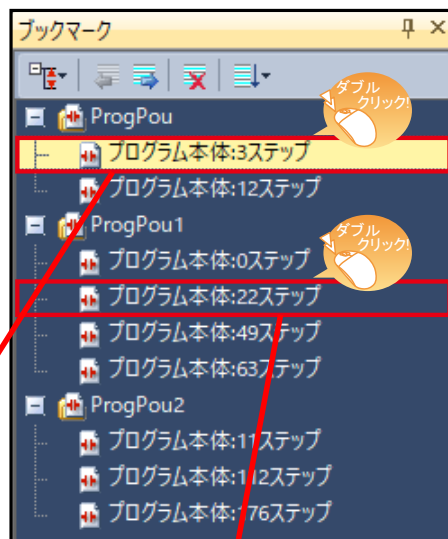
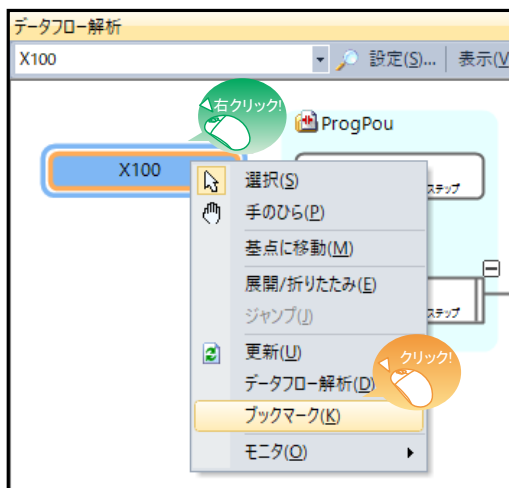
NEW

Ver.Up
新機能

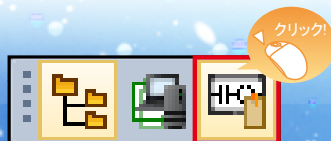
データフロー解析からブックマーク!

データフロー解析やプログラムエディタから
ブックマークが登録できるようになったよ!

便利な新機能



気になったところをあとで確認したいときも
見失わずにすぐに確認できていいね



ブックマークのウィンドウはツールバーから
表示 / 非表示が切り替えられるよ!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第83回

NEW

Ver.Up
新機能

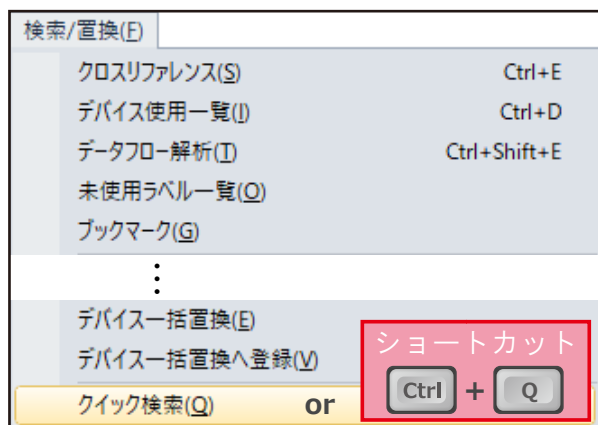
クイック検索からメニューを実行!



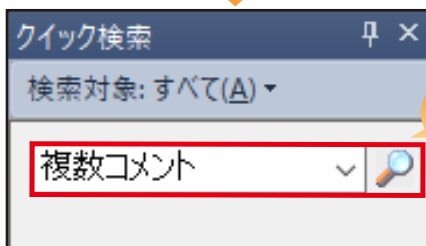
前に使った機能、名前はわかるんだけど
どこから開くのか忘れちゃったなあ…。

クイック検索を使ってみよう!
キーワードでナビゲーションウィンドウ内や
メニューの項目を検索できるよ。

便利な機能



ボタンで最近の
検索履歴からの
検索もできるんだ。



クイック検索結果					
検索対象	検索対象種別	表示/実行対象	表示対象種別	場所	ショートカットキー
フィルタ条件	フィルタ条件	フィルタ条件	フィルタ条件	フィルタ条件	フィルタ条件
複数コメント	メニュー	表示設定		表示#複数コメント#表示設定	
複数コメント	メニュー	コメント読出し		表示#複数コメント#コメント読出し	



ショートカットと Enter キーを使えば、
キー操作だけでも検索と実行ができるんだね!

GX Developer や GX Works2 のメニュー項目名でも
検索できるよ!これで君もシーケンスアープ!





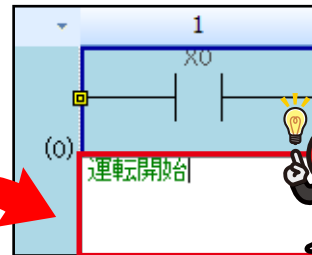
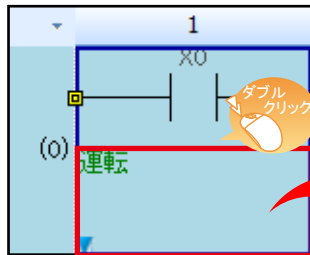
システム設計の頼れるヒーロー 教えて!GX3号

第84回

NEW
Ver.Up
新機能

ラダーエディタの入力改善!

ラダーエディタでのコメント編集がさらに便利になったよ!ダブルクリックするだけで編集できるようになったんだ!

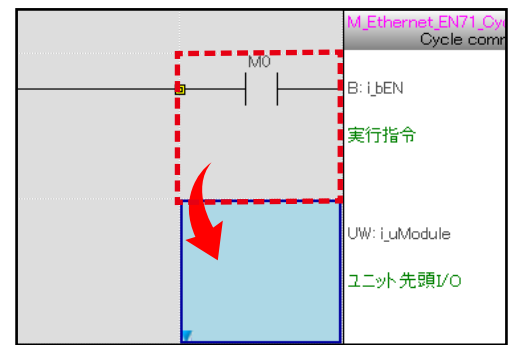


コメントのエリアをダブルクリック!



直感的にコメント編集ができるようになったね。

FBで最初に引数を入力するときの操作性も改善したよ。



次の引数にカーソルが移動するから引数入力が楽だね!

ラダーエディタがもっと使いやすくなったよ!
これで君もシーケンスアープ!





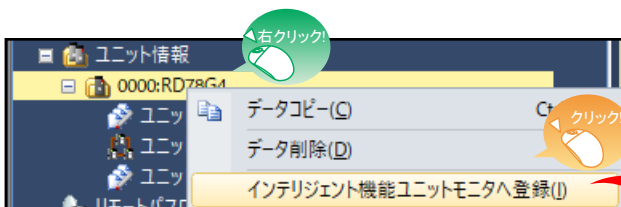
システム設計の頼れるヒーロー 教えて!GX3号

第85回

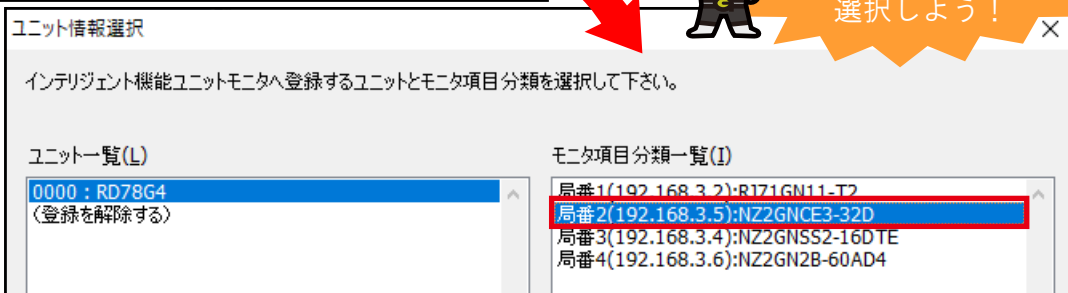
NEW
Ver.Up
新機能

デバイス局も一括でモニタ!

CC-Link IE TSN で接続したデバイス局をインテリジェント機能ユニットモニタに登録できるようになったよ!



デバイス局に
設定したユニットを
選択しよう!



デバイス局の状態をモニタしながら
プログラムを確認することができるんだ。



インテリジェント機能ユニットモニタ1(0000:RD78G4)-局番1(192.168.3.2):RJ71GN11-T2【ウォッチ中】

名称	現在値	割付(デバイス)	データ型
マスタ局のバッファメモリ			
自局CPU動作状態	STOP(中度・重度異常)	U0*G94539	ワード[符号なし]/ビット列[16ビット]
自局データリンク異常状態	正常	U0*G94212.9	ビット
各局データリンク異常状態	異常	U0*G94640.0	ビット
リモート入力(RX)			
リモート出力(RY)			



モニタする
デバイス局は10個まで
登録できるよ!

現在値変更もできるから、デバッグに便利だね!

効率的にエラーを解消するのに使ってね!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第86回

NEW
Ver.Up
新機能

FBマニュアルを簡単表示!



ユーザFBを使用する設計者が、該当するFBのマニュアルをすぐに見つけたいなあ。

e-Manual Createで作成したマニュアルを、プロパティの「ヘルプのパス情報」に設定すると呼び出せるよ!

便利な新機能

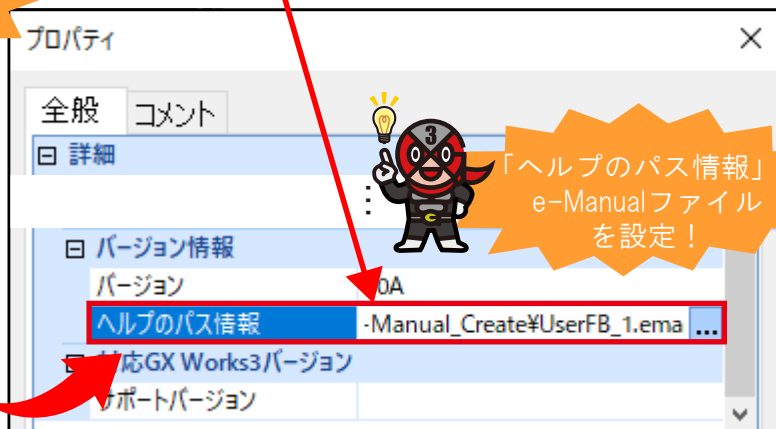
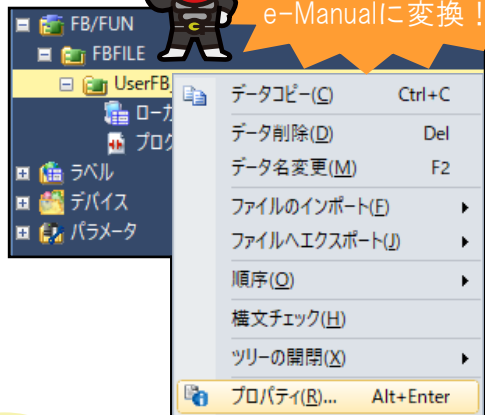


e-Manual Create

e-Manual



e-Manual Createで
マニュアルを
e-Manualに変換!



「ヘルプのパス情報」に
e-Manualファイル
を設定!



作ったマニュアルをすぐに表示できるんだ!

設定したマニュアルはナビゲーションウィンドウや部品選択ウィンドウ、プログラムエディタから「F1」キーで呼び出せるよ。



e-Manual Createは三菱電機FAサイトからダウンロードして使ってみてね!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第87回

NEW
Ver.Up
新機能

システムシミュレーションの機能拡張

CC-Link IE TSN/CC-Link IE Control/CC-Link IE Field に接続した時のシステムシミュレーションについて、以下を拡張したよ。

- ・ 応用設定の反映 (ネットワークのユニットパラメータ)
- ・ トランジェント伝送機能 (送受信命令)



ネットワークのユニットパラメータ

<input checked="" type="checkbox"/> 応用設定
<input checked="" type="checkbox"/> 通信速度設定
<input checked="" type="checkbox"/> サイクリック補助設定
<input checked="" type="checkbox"/> トランジェント伝送グループNo設定
<input type="checkbox"/> パラメータ名称
<input type="checkbox"/> 動的ルーチング設定
<input type="checkbox"/> 他局イベント取得設定
<input type="checkbox"/> ユニット動作モード設定
<input type="checkbox"/> セキュリティ
<input type="checkbox"/> リンク間転送設定



トランジェント伝送機能の命令をサポート!

送受信命令

JP.SEND	J1	D0	
JP.WRITE	J1	D100	D150



いろいろな設定でもシミュレーションできるのは嬉しいな!

シミュレーション中にネットワーク診断もできるんだ。

CC-Link IE TSN/ CC-Link IE Field診断

選択局通信状態モニタ (RJ71GN11-T2)

局番 0 異常発生 ネットワーク: CC IE TSN
CC-Link IE TSN Class: B
MACアドレス: 02-00-00-01-00-01 IPアドレス: [REDACTED]

ユニット異常...

RUN ERR
MST D LINK
P1SD/RD
P2SD/RD

PORT2 ケーブル断...



対応している命令は、GX Works3 オペレーティング
マニュアルの付録で確認してね。
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

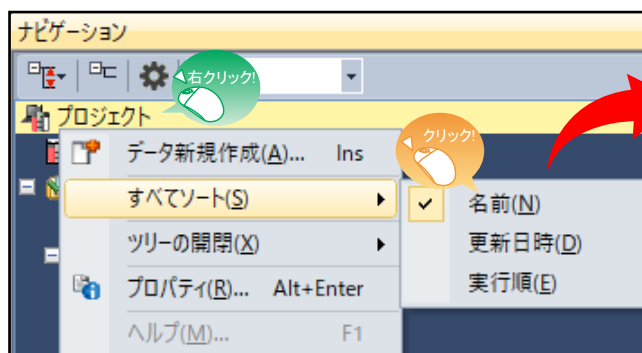
第88回

NEW

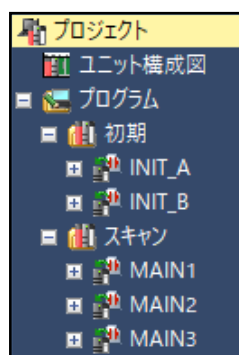
Ver.Up
新機能

ナビゲーションウィンドウの操作性改善

ナビゲーションウィンドウの機能がもっと便利になったよ!
「すべてソート」で、実行順の他に、名前順、更新日時順を
選択できるようになったんだ。



名前順

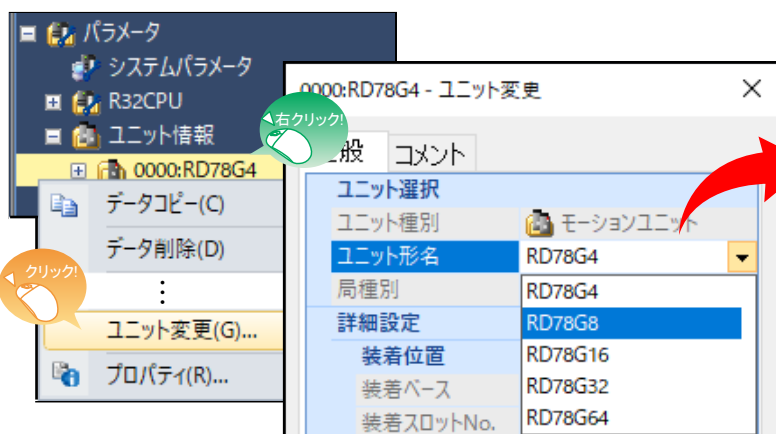


更新日時順



自分が見やすい順番にソートできて便利だね!

モーションユニットの軸数も、ナビゲーション
ウィンドウから変更できるようになったよ!



ナビゲーションウィンドウがさらに使いやすくなったよ!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

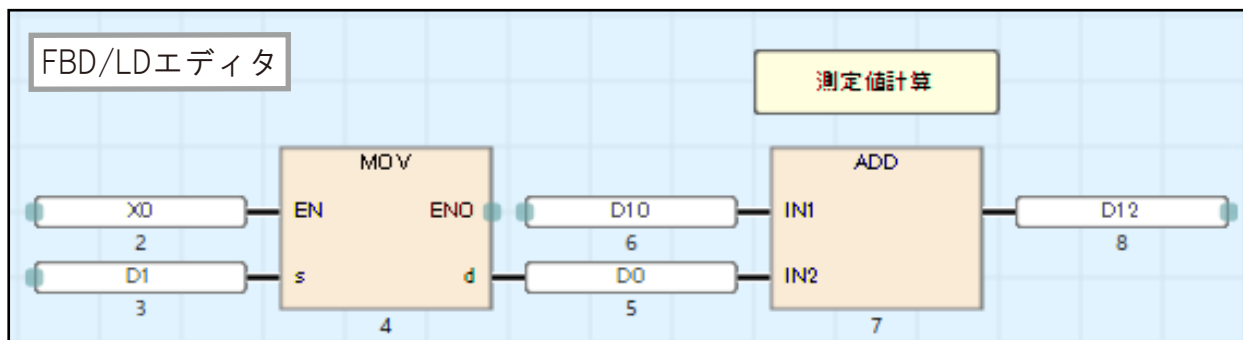
第89回

NEW

Ver.Up
新機能

FBD/LDでのインラインST対応!

GX Works3 では FBD 言語も使えるよ。各要素からの信号のフローがわかりやすい言語で、計装分野でよく使われるんだ。GX Works3 には FBD/LD エディタがあるよ。



これから FBD/LD エディタの新機能を紹介していくね!
まずは FBD/LD エディタのインライン ST について紹介するよ。



ツールバー

クリック!

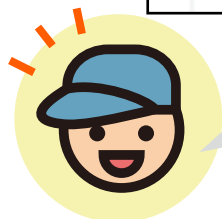
ダブルクリック!

開いたSTエディタでプログラムを編集!

```

STB_1: ProgPou [PRG] [FBD/LD: IST] 41Step
1 IF (dSample1-dSample2)>dSample3 THEN
2   dSample4:=dSample4*dSample5;
3 END_IF;
4

```



ちょっとした計算式の記述に便利だね!

用途によってエディタを使い分けてね!
これで君もシーケンスアープ!





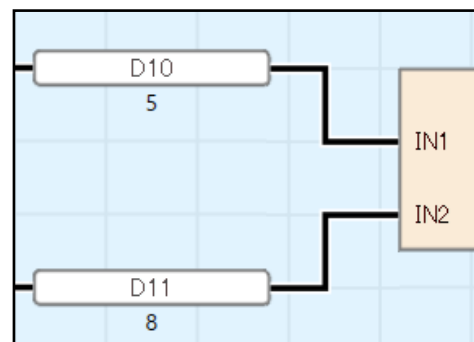
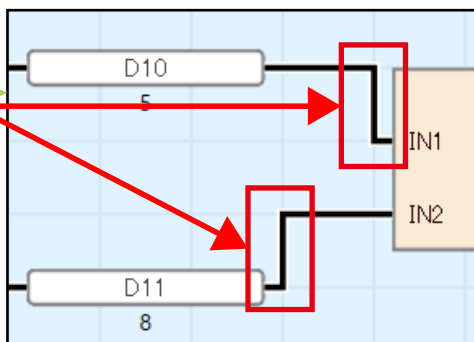
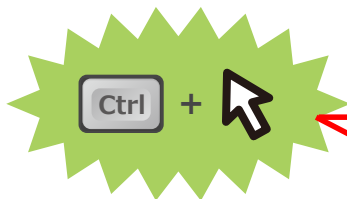
システム設計の頼れるヒーロー 教えて!GX3号

第90回

NEW
Ver.Up
新機能

FBD/LDで接続線を整形!

FBD/LD エディタで接続線が整形できるようになったよ!
Ctrl キーを押しながらマウスでドラッグして接続線を動かすと、折れ線の位置を自由に変えられるんだ。



見た目をきれいに整えられるね!

FBD/LD エディタの回路を一括で整形することもできるよ。整形する項目はオプションで設定してね。



編集(E)	
元に戻す(U)	Ctrl+Z
やり直し(R)	Ctrl+Y
切り取り(T)	Ctrl+X
コピー(C)	Ctrl+C
貼り付け(P)	Ctrl+V
削除(D)	
回路ブロック選択(K)	Ctrl+S
レイアウト(L)	



どの範囲を整形するか選んでね!

行挿入(I)	Shift+I
...	
回路ブロック内の整形(L)	Ctrl+Shift+I
回路ブロック内の一括整形(H)	
回路ブロック間の空白行の削除(R)	
回路ブロックの一括左挿入(A)	

オプション	
プロジェクト	
プログラムエディタ	
共通項目	
ラダーエディタ	
STエディタ	
FBD/LDIエディタ	
SFC図エディタ	
その他のエディタ	
回路ブロック	
回路ブロック番号を表示する	いいえ
回路ブロック一覧	
ジャンプ後、ダイアログを開じる	いいえ
回路ブロック内の整形	
接続線を整形する	はい
FB/FUNへ接続する変数部品位置を移動する	はい
空白列を削除する	はい
空白行を削除する	はい

整形機能で FBD/LD エディタの回路を整えてみよう!
これで君もシーケンスアープ!



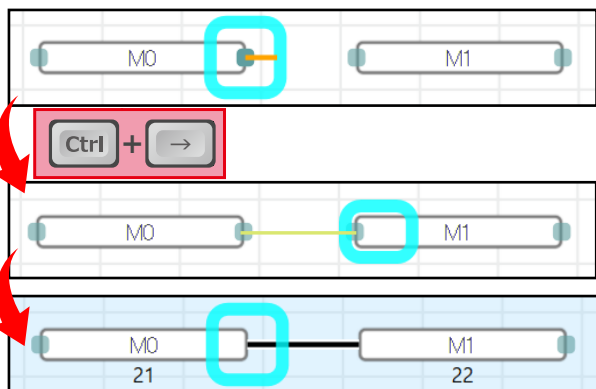


FBD/LDでもキーボードですばやく結線

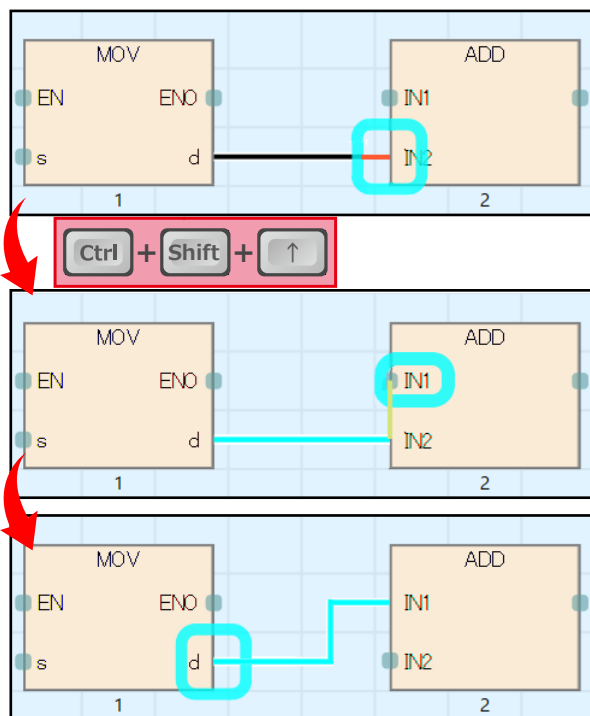
FBD/LD エディタも、キー操作で便利に使えるよ！
Ctrl キー、Shift キーと矢印キーの組み合わせで
結線してみよう。



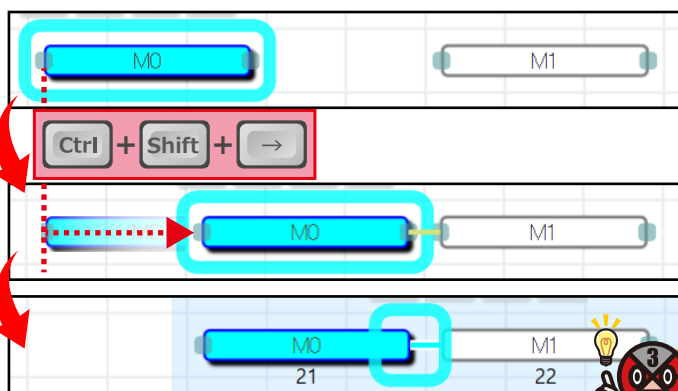
接続ポイントから、Ctrl+矢印キーで結線



Ctrl+Shift+矢印キーで、
接続ポイントを付けなおし



Ctrl+Shift+矢印キーで、部品を動かして自動結線



自動結線のショート
カットは、ツールバーで
有効にして使ってね！

ツールバー



キーボードでもマウスでも簡単に結線できるね。

FBD/LD エディタでも便利な
ショートカットキーを用意しているよ！
これで君もシーケンスアープ！





システム設計の頼れるヒーロー 教えて!GX3号

第92回

NEW
Ver.Up
新機能

FBD/LDのFBをコンパクトに配置!

FB/FUN をコンパクトに配置できるよ。
まずは、FB の高さを調整する方法を紹介するね!

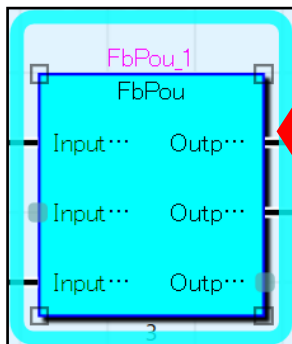


[編集]メニュー

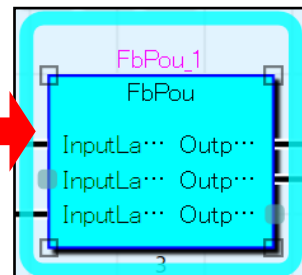


クリック

通常表示

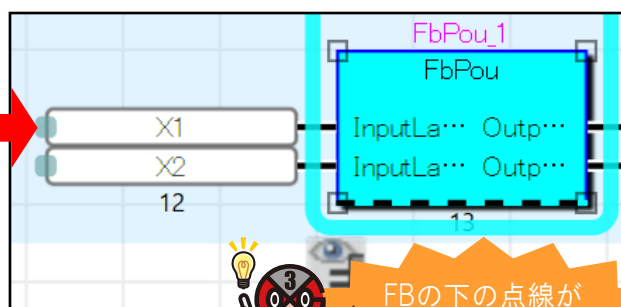
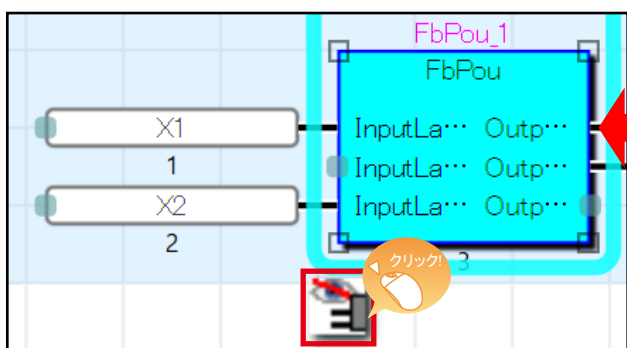


コンパクトに表示
(幅を維持)



[幅を維持]を
クリックするたびに
表示が切り換わるよ!

さらに、未接続引数を非表示にしよう。
FB の下のスマートタグで操作してね。



FB の下の点線が
非表示の引数がある
ことを示しているよ!



引数の多いFB もすっきり配置できるね!

新規追加時のFB/FUN の高さはオプションで選べるよ!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第93回



FBD/LDのモニタ表示を最適に

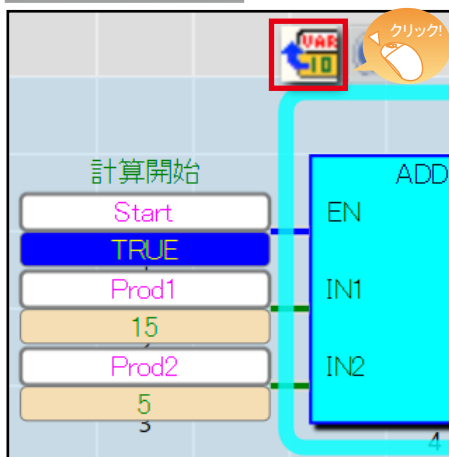


スマートタグ

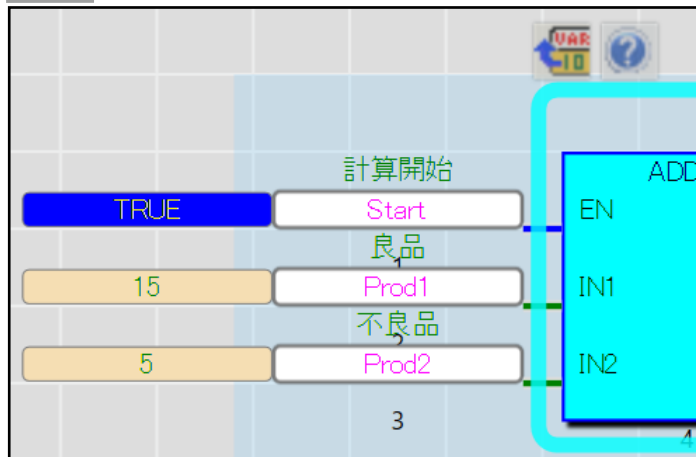


スマートタグで、モニタ表示位置を変更できるよ！
プログラム部品を移動しなくても、モニタでの
確認がしやすいんだ。

下(デフォルト)

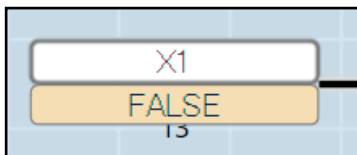


外側

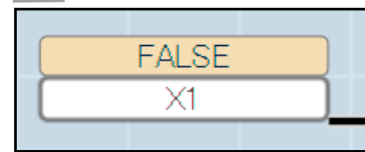


ラベル・デバイスを
選択すると
上下左右・変数部品上に
移動できるよ！

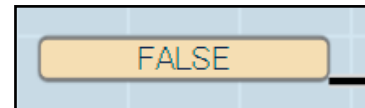
下(デフォルト)



上



変数部品上
(変数部品に重ねて表示)



部品の配置に合わせて変更できていいね！

変更した表示位置は、プロジェクトを閉じるまで有効だよ！
これで君もシーケンスアープ！





システム設計の頼れるヒーロー 教えて!GX3号

第94回

NEW

Ver.Up
新機能

ガイダンスフロー機能でiQ-Fを設定!

ガイダンスフロー機能を紹介するね!
やりたいことに対して、必要な機能の
準備 / 設定をサポートするよ!



CPUでできることを
一覧表示!やりたい
ことを選んでね!

ガイダンスフロー

CPU内蔵機能

- CPUユニットの稼働状況確認やエラー箇所特定のためデバイス値を保存したい
- エラー発生時に全デバイスの値を保存したい
- アナログ信号を入力したい
- アナログ信号を出力したい
- 485シリアルポートでプリンタやバーコードリーダなどとシ
- MODBUS_RTU機器と通信するマスタ局を設定したい
- MODBUS_RTU機器と通信するスレーブ局を設定したい
- MODBUS/TCP機器と通信するマスタ局を設定したい
- MODBUS/TCP機器と通信するスレーブ局を設定したい

データロギング機能

準備

SDメモ리카ードをCPUユニットへ装着する必要があります
CPUユニットロギング設定ツールをインストールしておく必要があります

データロギング機能の設定

データロギング機能を有効にする

必須 [内部バッファ使用機能]を[データロギング機能]に
オンサへ書き込む

CPUユニットロギング設定ツールでデータロギングを実行す

必須 データロギング設定を作成する

CPUパラメータ画面

項目	設定
グローバルポイント	2048 点
ポイント型ラベル	2048 点
内部バッファ容量設定	
合計容量	320 Kバイト
内部バッファ使用機能	データロギング機能
データロギング機能	
合計容量	320 Kバイト
設定No.1	80 Kバイト
設定No.2	80 Kバイト
設定No.3	80 Kバイト
設定No.4	80 Kバイト
メモリダンプ機能	0 Kバイト

フローに従って設定すると、迷わずにシーケンサの機能を使えるね。

あとで設定変更したいときも、この機能
から項目を選んで変更ができるよ。

iQ-F CPU を使う時に、クイック検索から
「ガイダンスフロー」で検索してみよう!
これで君もシーケンスアープ!



システム設計の頼れるヒーロー 教えて!GX3号

第95回



プログラム構成図でプロジェクトを解析

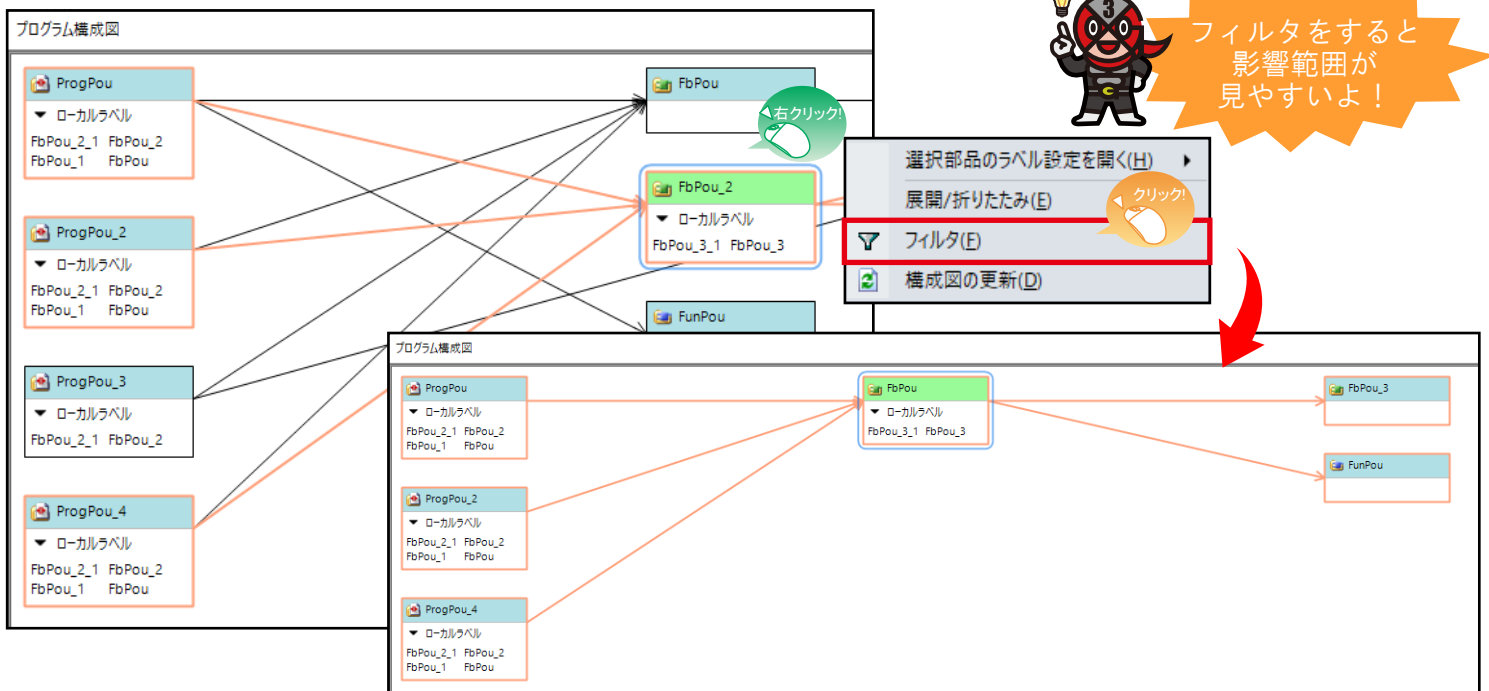


他の人が作ったプロジェクトのFBを直したいんだけど、影響する範囲をどうやって調べよう…。

プログラム構成図を使ってみよう！プログラムとFB、FUNの関係を解析して図示できるから、修正が影響する範囲を効率的に調べられるよ！

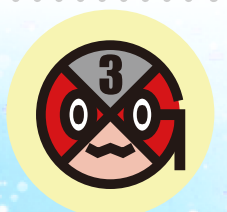


[ツール]メニュー→[プログラム構成図]



選択したブロックに関するブロックが強調表示されるんだ！影響範囲がわかりやすく見れていいね！

複雑なプログラムでも構造を把握しやすくなったね！
これで君もシーケンスアープ！





システム設計の頼れるヒーロー 教えて!GX3号

第96回

NEW
Ver.Up
新機能

ウォッチ機能でビット単位の値変更!

ウォッチ機能がさらに便利になったよ!
ビット以外のラベルもバイナリウォッチ画面に
登録するとビット単位で現在値変更できるんだ。



ウォッチ1【ウォッチ中】

ON OFF ON/OFF反転 更新

名称	現在値	表示形式	データ型	コメント
State[0]	4,067	10進数	ワード[符号なし]/ビット列[16ビット]	装置状態
State[1]	1,035	10進数	ワード[符号なし]/ビット列[16ビット]	エラー状況
ProgPou/Machine1	227,553,327	10進数	ダブルワード[符号なし]/ビット列[32ビット]	稼働状況(装置)

右クリック

- タイトルの変更(N)...
- ON/OFF表示(O)
- 選択したデバイス/ラベルの表示形式を変更
- 選択したデバイスのデータ型を変更
- 選択したラベルをバイナリウォッチ画面に追加(L)...
- コメント編集(E)...
- ファイルのインポート(I)...
- ファイルのエクスポート(O)...

クリック!

バイナリウォッチ【ウォッチ中】

表示形式を開く(O)...

ショートカット Shift + Enter

ダブルクリック

名称	データ型	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	現在値	文字列	コメント
State[0]	ワード[符号なし]/ビット列[16ビット]	0	0	0	0	1	1	1	1	0	0	0	0	0	0	1	1	4067	.	装置状態
State[1]	ワード[符号なし]/ビット列[16ビット]	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	1035	..	エラー状況
ProgPou/Machine1	ダブルワード[符号なし]/ビット列[32ビット]	0	0	1	1	0	0	0	0	0	0	1	0	1	1	1	1	12335	/0	稼働状況(装置1)
		0	0	0	0	1	1	0	1	0	0	1	0	0	0	0	0	3472	.	

ダブルクリックや [Shift]+[Enter] キーで
ビットを反転できるよ!



デバイス/バッファメモリー括モニタと同じように
使えるんだ!ビット単位での操作がしやすくなったね。

表示形式の変更もできるよ!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第97回

NEW
Ver.Up
新機能

ウォッチウィンドウ一括登録!

ウォッチ機能の新機能をさらに紹介するね!
選択したプログラムで使用している、すべての
ラベル・デバイスを一括登録できるようになったよ。



右クリック

- クロスリファレンス(S)
- デバイス使用一覧(I)
- データフロー解析(D)
- ブックマーク(M)
- FBプロパティ(Y)
- ウォッチウィンドウへ登録(H)
- ウォッチウィンドウ一括登録(Z)**
- デバイス一括置換へ登録(A)
- 波形表示(リアルタイムモニタ)(Z)...
- 波形表示(オフラインモニタ)(W)...
- 変化点検索(X)
- 命令ヘルプを開く(R)...
- ファイルのインポート(E)...
- ファイルへエクスポート(U)...

名称	現在値	表示形式	データ型	Japanese/日本語
SM400	TRUE	2進数	ビット	常時ON
X0	FALSE	2進数	ビット	
M0	TRUE	2進数	ビット	
D100	6,699	10進数	ワード[符号付き]	
bProgRun	TRUE	2進数	ビット	プログラム実行中
ProgPou/wData1	15,210	10進数	ワード[符号付き]	製品A
ProgPou/wData2	6,553	10進数	ワード[符号付き]	製品B
D200	21,763	10進数	ワード[符号付き]	
M100	TRUE	2進数	ビット	
M200	TRUE	2進数	ビット	
M1	FALSE	2進数	ビット	



上限の5120点まで
一括で登録できるよ!



一つずつ選択して登録する手間が省けていいね!

現在値の表示も、10進数表示は3桁ごとにカンマが入ってより見やすくなったよ!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第98回

NEW
Ver.Up
新機能

STエディタで複数箇所を同時編集!

STエディタで複数箇所の同時編集ができるようになったよ!
矩形選択をして編集してみよう!



文字を挿入してデバイスの範囲を変更

文字を上書きしてデバイス種別を変更

Alt + ドラッグ

1	D0	:= 100;
2	D1	:= 100;
3	D2	:= 100;
4	D3	:= 100;
5		

矩形選択状態

ProgPou [PRG] [ST] 13ステップ

1	D1000	:= 100;
2	D1001	:= 100;
3	D1002	:= 100;
4	D1003	:= 100;
5		

Alt + ドラッグ

1	D0	:= 100;
2	D1	:= 100;
3	D2	:= 100;
4	D3	:= 100;
5		

矩形選択状態

ProgPou [PRG] [ST] 13ステップ

1	ZR0	:= 100;
2	ZR1	:= 100;
3	ZR2	:= 100;
4	ZR3	:= 100;
5		

水色で矩形選択されたところに文字が置き換わるよ!



キーボード操作の場合は行と範囲を選択してね!

対象行を選択

Shift + Alt + or

範囲を選択

Shift + or



同じ内容の変更なら、まとめて編集することができるんだ!
デバイスの範囲が変わったときも編集しやすいね。

インライン ST でも使えるよ!
バージョンアップして使ってみてね。
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第99回

NEW
Ver.Up
新機能

インクリメンタル検索でハイライト表示!

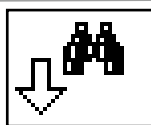
ST エディタでインクリメンタル検索ができるようになったよ! [Ctrl]+[I] キーでインクリメンタル検索モードにして、検索したい文字を直接入力しよう。



ショートカット

Ctrl + I

インクリメンタル
検索モードの時の
マウスカーソル



「D1」と入力

ProgPou [PRG] [ST] 21ステップ

```
1 D100 := D20 + D30;
2 IF D1< D100 THEN
3   M1 := TRUE;
4 ELSE
5   D100 := D102;
6 END_IF;
7
```



カーソルが変わったら
検索キーワードを入力
してね!この例では「D1」
と入力しているよ。

見つけたキーワードがハイライト表示されるね!

さらに [Ctrl]+[I] キーで
次のハイライトに移動しよう!

ショートカット
(下に検索)

Ctrl + I

ショートカット
(上に検索)

Shift + Ctrl + I

or

[検索/置換]メニュー

インクリメンタル検索(M)

下に検索(D) Ctrl+I

ジャンプ(J)... Ctrl+G

上に検索(U) Ctrl+Shift+I

次のハイライトにジャンプ

ProgPou [PRG] [ST] 21ステップ

```
1 D100 := D20 + D30;
2 IF D1< D100 THEN
3   M1 := TRUE;
4 ELSE
5   D100 := D102;
6 END_IF;
7
```

[Esc] キーでインクリメンタル検索モードを終了できるよ!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号 第100回



ラベルコメントウィンドウからドラッグ&ドロップ

ラベルエディタ

	ラベル名	データ型	Japanese/日本語
1	w10Error	ワード[符号付き](0.2)	エラー情報
2	wSample	ワード[符号付き]	
3	wTemp	ワード[符号付き]	ステータス

ラベルコメント

オプション(O)... <フィルタ>

ラベル名	Japanese/日本語(表示...)
w10Error	エラー情報
[-] w10Error[0]	駆動エラー
[-] w10Error[1]	モニタエラー
[-] w10Error[2]	その他エラー

ラベルの配列要素やビット指定ごとにコメントを設定できるラベルコメントウィンドウでもマウス操作でラベルが使えるようになったよ!



マウス操作だと直感的にラベルを扱えるね。

ラベルをドラッグ & ドロップでエディタに配置してね!



ラベルコメント

オプション(O)...

ラベル名	Japanese/日本語(表示...)
w10Error	エラー情報
[+] w10Error[0]	駆動エラー
[+] w10Error[1]	モニタエラー
[+] w10Error[2]	その他エラー

ラダー、ST、FBD/LDエディタで使えるよ!

MOV	K1	w10Error[D0] エラー情報
MOV	K1	w10Error[2] その他エラー
		w10Error[1] モニタエラー



ラベルの配列要素に設定したコメントを確認しながらプログラムできるから、間違いが減りそうだね!

ビット指定のラベルでも同じように使えるよ!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号 第101回



【特集】現在値変更をマスターしよう!

今回はウォッチウィンドウを使った現在値変更の使い方についてまとめるよ!まずは基本的な操作を紹介するね。現在値変更したいデバイスを、[Shift] キー + [Enter] キーで登録しよう!



プログラムエディタ

ショートカット
Shift + Enter

	MOV	D0	wLabel
		0	0
		SET	Y10
D+	D10	D20	dLabel
	123456789	234567890	358024679



登録後に数値を入力すると、マウス操作なしで現在値変更ができるよ!

ウォッチ1【ウォッチ中】

ON | OFF | ON/OFF反転 | 更新

名称	現在値	表示形式	データ型
D10	123,456,789	10進数	ダブルワード[符号付き]
D20	234,567,890	10進数	ダブルワード[符号付き]
dLabel	358,024,679	10進数	ダブルワード[符号付き]
D0	100	1進数	ビット



登録した項目が不要になった時は [Delete] キーで削除!

ウォッチウィンドウを常時表示する必要がなければ、[ESC] キーでウィンドウを閉じてプログラムエディタに戻ることができるよ!

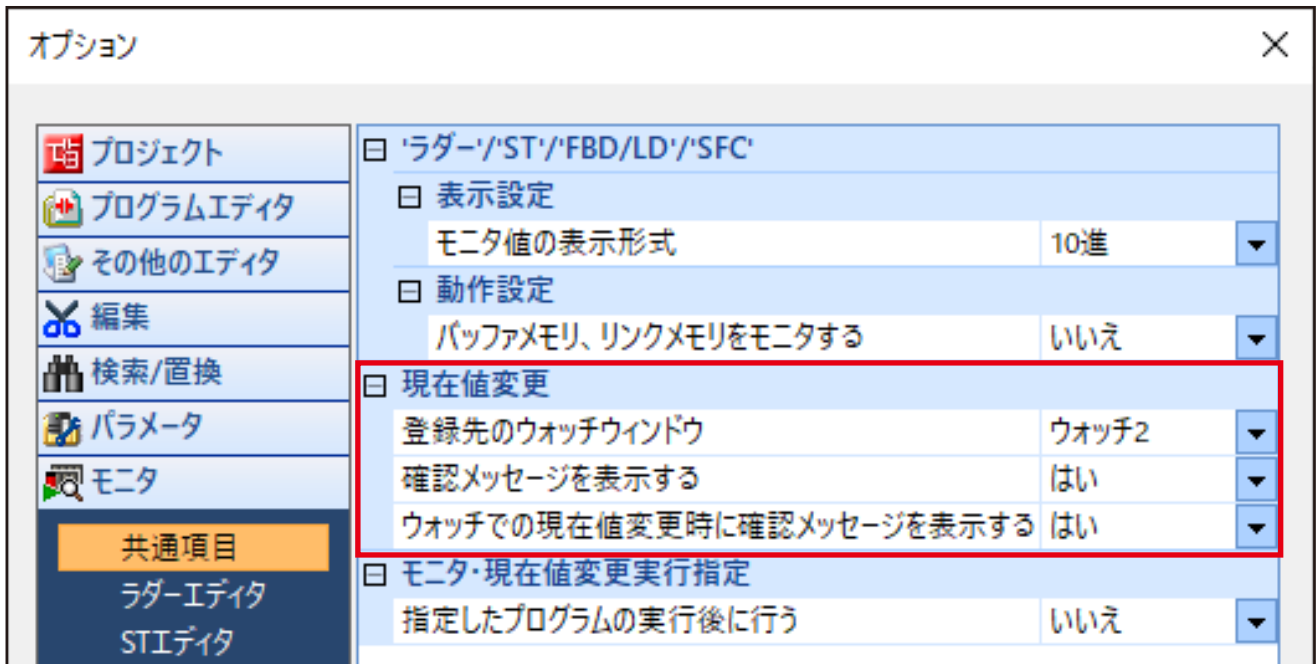


ショートカットキーを使うと便利だね!他にはどんな機能があるのかな?

次に紹介するオプションやバイナリウォッチウィンドウと組み合わせて、もっと効率的にデバッグしよう!



次はオプションの設定に関わる機能を2つ紹介するよ!
現在値変更のオプションから設定してね。



オプション: 登録先のウォッチウィンドウの使い分け!

最初に、ウォッチウィンドウをモニタ用と現在値変更用に
分ける使い方を紹介するよ。オプションの「登録先の
ウォッチウィンドウ」を「ウォッチ2」に設定してみよう。



モニタ用

名称	現在値	表示形式	データ型
D10	123,456,789	10進数	ダブルワード[符号付き]
D20	234,567,890	10進数	ダブルワード[符号付き]
dLabel	358,024,679	10進数	ダブルワード[符号付き]
D0	100	10進数	ワード[符号付き]
wLabel	100	10進数	ワード[符号付き]

現在値変更用

名称	現在値	表示形式	データ型
D0	100	10進数	ワード[符号付き]
wLabel	100	10進数	ワード[符号付き]



ウォッチ1をモニタ用、ウォッチ2を現在値変更用
として使えるんだね。モニタ用のデバイス/ラベルを
他のウォッチウィンドウに登録できるから見やすいね。

現在値変更の数値を入力した後で、確認してから反映させることもできるよ。オプションの「ウォッチでの現在値変更時に確認メッセージを表示する」から設定してね。



ウォッチ3【ウォッチ中】

名称	現在値
D10	123,456,789
D20	234,567,890
dLabel	358,024,679
D0	100
wLabel	0

確認メッセージ

MELSOFT GX Works3

現在値変更を実行してよろしいですか？

[名称] D0
[データ型] ワード[符号付き]
[変更前の値] 0
[変更後の値] 100

はい(Y) いいえ(N)

ウォッチ3【ウォッチ中】

名称	現在値
D10	123,456,789
D20	234,567,890
dLabel	358,024,679
D0	100
wLabel	0



第三者にチェックしてもらってから現在値を反映させるような使い方もできるね。



確認メッセージで「はい」を押すと現在値変更の数値が反映されるよ

【新機能】バイナリウォッチでビット指定の単位で値変更！

前に紹介したバイナリウォッチ画面は、ビット型以外のラベルでもビット指定の単位で現在値変更ができるよ。



バイナリウォッチ【ウォッチ中】

表示形式を開く(Q)...

名称	データ型	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	現在値	文字列	コメント
State	ワード[符号なし]/ビット列[16ビット]	0	0	0	1	0	1	0	0	1	0	0	1	0	1	0	1	5269	•	稼働状況
State2	ダブルワード[符号なし]/ビット列[32ビット]	0	0	1	1	1	0	0	0	0	1	1	0	0	1	0	0	14436	d8	エラー発生状態
		0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	258	..	



デバイス/バッファメモリー括モニタと同じように使えていいね！

現在値変更をマスターして使いこなそう！
これで君もシーケンスアープ！





システム設計の頼れるヒーロー 教えて!GX3号 第102回



デバッグ時にログを確認!

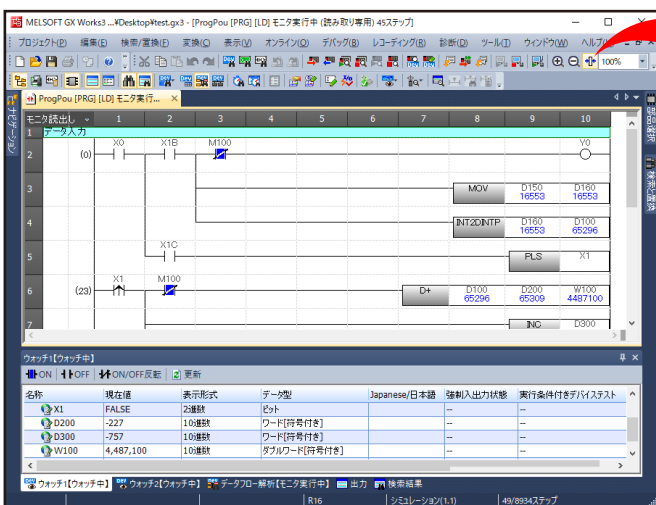


シミュレータでプログラムの動作を確認してたけれど、デバイス/ラベルの値のログが見れたらもっと簡単にデバッグできるんだけどなあ。

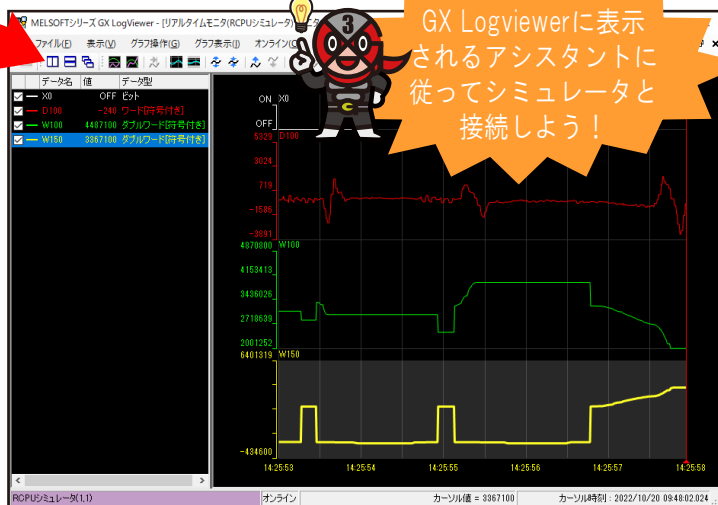
GX Works3 のシミュレータと GX LogViewer のリアルタイムモニタを組み合わせると、デバイス/ラベルの値のログを確認することができるよ!



GX Works3



GX LogViewer



GX Logviewerに表示されるアシスタントに従ってシミュレータと接続しよう!

GX LogViewer の「名前をつけて保存」メニューからログをファイルに出力して、後から内容を確認することもできるよ。



何回もシミュレータを起動して再現し直さなくても見たいところをすぐに確認できていいね!

リアルタイムモニタを活用してデバッグを効率化しよう!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号 第103回



ラダーエディタのコメントをデフォルトで表示

プロジェクトを作るときやシーケンサから読み出したときのコメント/ステートメント/ノートをはじめから表示するようになったよ!



オプション	
プロジェクト	コメント
プログラムエディタ	表示項目
共通項目	デバイス/ラベルコメント
ラダーエディタ	割付けデバイスのコメントを表示する
STエディタ	ステートメント
	ノート
	はい
	いいえ
	はい
	はい



既定値に設定(S)



既にGX Works3を使っている場合は設定がバージョンアップ前の非表示のままです。表示に変更したい場合は、オプションで表示する設定を既定値にしてください。

プログラム開始		データ設定
(0)	X0	MOV K100 D100
(0)	↑	処理数



コメントを表示させてラダーを見ていたから手間が減ったよ。でも、コメントを非表示にしたい場合は手間がかかりそう…。

[表示]メニューから、コメント/ステートメント/ノートの表示を一括で切り替えられるよ!



[表示]メニュー

<input checked="" type="checkbox"/>	コメント表示(D)	Ctrl+F5
<input checked="" type="checkbox"/>	ステートメント表示(S)	Ctrl+F7
<input checked="" type="checkbox"/>	ノート表示(Q)	
<input checked="" type="checkbox"/>	コメント/ステートメント/ノート一括表示(D)	クリック!

ショートカット

Ctrl + F11

GX Works3 を新規にインストールすると、最初からデフォルト表示になってるよ。これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!Gx3号 第104回

NEW
Ver.Up
新機能

ラベルエディタの表示をカスタマイズ!

ラベルエディタの3つのエリアの表示 / 非表示を
選べるようになったよ! オプションで設定してね。



オプション

プロジェクト	<input type="checkbox"/> 表示設定
プログラムエディタ	末尾の空行表示を1行にする はい
その他のエディタ	<input type="checkbox"/> ウィンドウの表示数を固定する いいえ
デバイスコメントエディタ	ウィンドウ表示数 5
ラベルエディタ共通	<input type="checkbox"/> 構造体コメントの階層表示設定 末尾メンバのみ
ユニット構成図	区切り文字 .
プログラム構成図	<input type="checkbox"/> 詳細表示設定 詳細表示設定
編集	<input type="checkbox"/> 拡張表示設定
検索/置換	階層表示エリアに表示する項目名 ラベル名
パラメータ	拡張表示エリア 常に表示する
	システムラベル設定表示エリア 表示



システムラベル設定
表示エリアのデフォルトを
非表示に変更しました。
ご使用の場合は表示に設定を
変更してください。

ラベルの詳細表示設定エリア

ラベル名	データ型	クラス	割付け(デバイス/ラベル)	初期値	定数	Japanese/日本語(表示対象)	備考	システムラベルの関連	システムラベル名	属性
stParameter	Parameter	VAR_GLOBAL	詳細設定							
bFlag	ビット	VAR_GLOBAL								
wProduction	ワード(符号付き)	VAR_GLOBAL								

拡張表示: 常に表示

stParameter

ラベル名	データ型	デバイス
ID	ワード(符号なし)/ビット列(16ビット)	
Name	文字列(32)	
Value	ワード(符号付き)	
Permission	ビット	

拡張表示エリア

システムラベルの設定表示エリア

システムラベルの登録/解除予約を確定するには、システムラベルデータベースへの反映が必要です。システムラベルデータベースに反映するを実行してください。システムラベル Ver. 2では、割付けデバイスを変更した際に、参照側プロジェクトの変更が不要です。※システムラベル Ver. 2は、iQ-Rシリーズ/GOT2000シリーズでのみ利用可能です。※RUN中書き込みをする場合には、RUN中書き込みと上書き保存を実行してください。

システムラベルの登録予約

システムラベルの解除予約

システムラベルの取込

システムラベルデータベースに反映する

未反映: 0件
全件: 0件



自分の好きな表示にカスタマイズできるから、手間が省けていいね。

既定値に設定すれば、新しく作成したプロジェクトの
ラベルエディタもオプションの設定に従って表示されるよ!
これで君もシーケンスアープ!



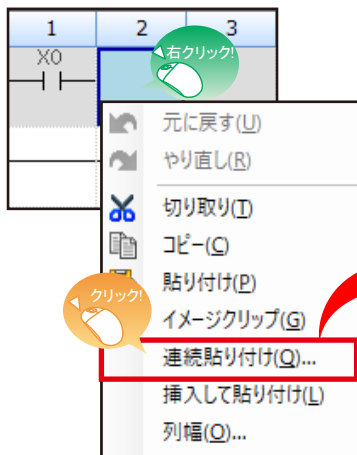


システム設計の頼れるヒーロー 教えて!GX3号 第105回

NEW
Ver.Up
新機能

ラダーエディタの連続貼り付け/表示を改善

ラダーエディタがさらに便利になったよ！接点回路を右方向に連続貼り付けしたときに、右母線で折り返すようになったんだ！



連続貼り付け

切り取り/コピーした回路内に存在するデバイスのデバイス番号をインクリメントしながら、現在のカーソル位置より連続で貼り付けます。

連続貼り付け数(1~99)(N): 個

ラベル名のインクリメント位置(L): 末尾の「数値」(b1Label1 >> b1Label1_1)

コメントを表示する。(C) インクリメント数一括設定(A)

種別	コピー元	インクリメント後	インクリメント数
デバイス	X0	>> X1	1

※インクリメント数の設定範囲は-9999999~9999999(10進数)です。
※実数定数、メンバ、インラインST内のデバイスはインクリメントしません。

ラベル情報を登録後、ラベルエディタで継続して登録したラベルの付加情報を編集する。(E)

ラベルを一番後ろに登録する。(R)

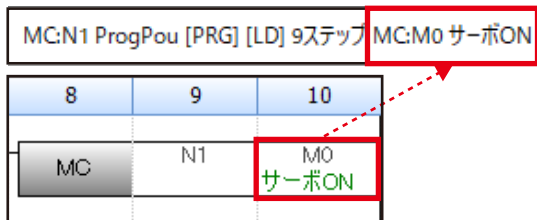
貼り付け方向(D):

1行を超える連続貼り付けの場合は自動で折り返すよ！

自動で折り返せるのは便利だね！

さらに、MC/MCR 命令を使うとウィンドウタイトルでデバイス/ラベルとコメントが表示できるよ！

デバイス/ラベルを編集する回路入力画面の表示も見やすくなったよ！これで君もシーケンスアープ！





システム設計の頼れるヒーロー 教えて!GX3号 第106回

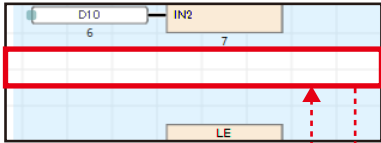
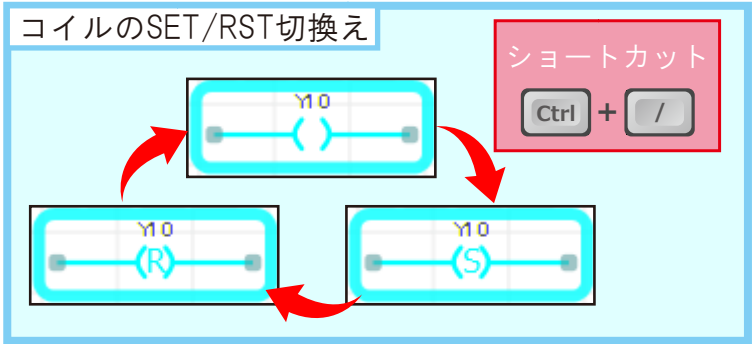
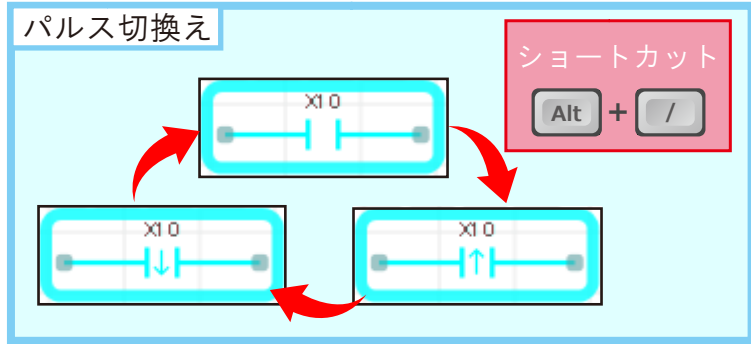
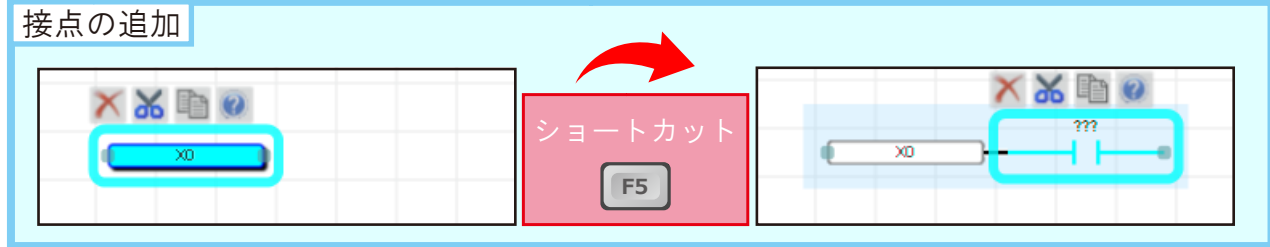


FBD/LDのショートカットキー!

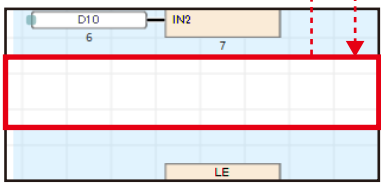
今回はFBD/LD エディタのショートカットキーを紹介するね! 効率的にキーボードでプログラミングしよう!



FBD/LD エディタにも、いろんなショートカットキーがあるんだね!



プログラムを途中で追加する場合は、行挿入で簡単にスペースを空けられるよ!



行挿入の場合
ショートカット
Shift + Ins

行削除の場合
ショートカット
Shift + Del

接点の追加とパルスの切換え、コイルのSET/RST 切換えは、ツールバーからも操作できるよ。これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号 第107回

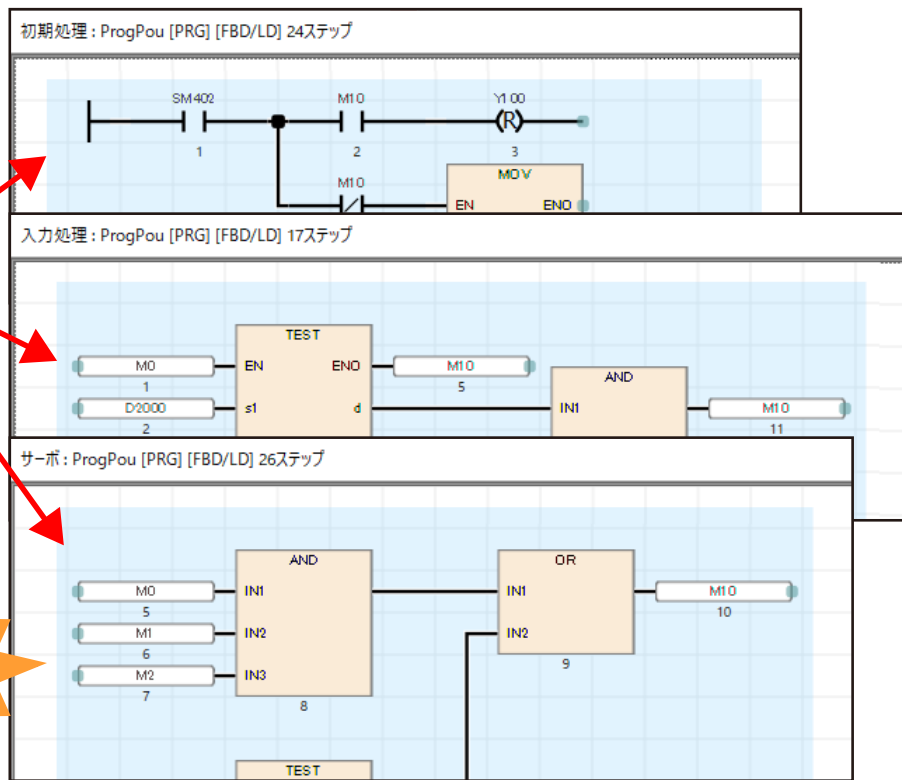
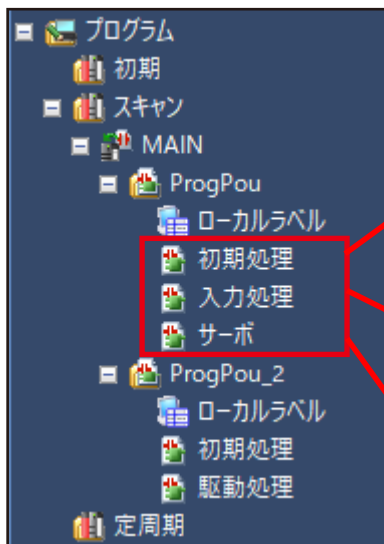


FBD/LDのワークシートを分割!

FBD/LD エディタはワークシートを分割して使えるよ!
分割したワークシートのローカルラベルは共通なんだ。



POU の中で工程や役割ごとに
分割するとわかりやすいね!



ワークシートは
ナビゲーションウィンドウ
のPOUを右クリックして
追加してね!

工程や役割ごとにワークシートで分割しておくよ、
簡単にプログラムの流用もできるよ。



わかりやすくワークシートを分割して
メンテナンス性を向上しよう!
これで君もシーケンスアープ!



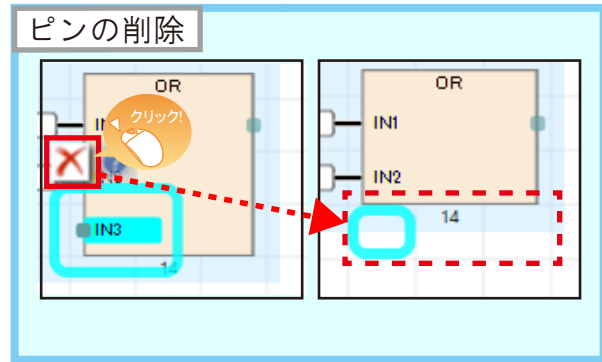
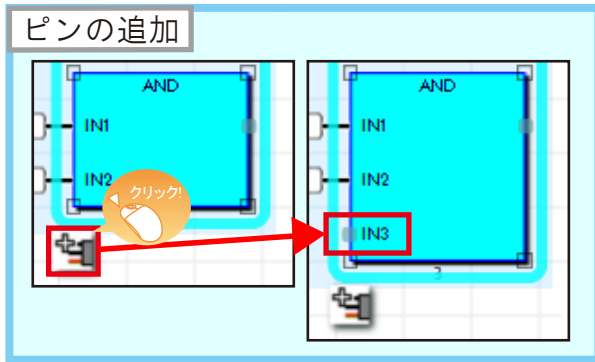


システム設計の頼れるヒーロー 教えて!GX3号 第108回

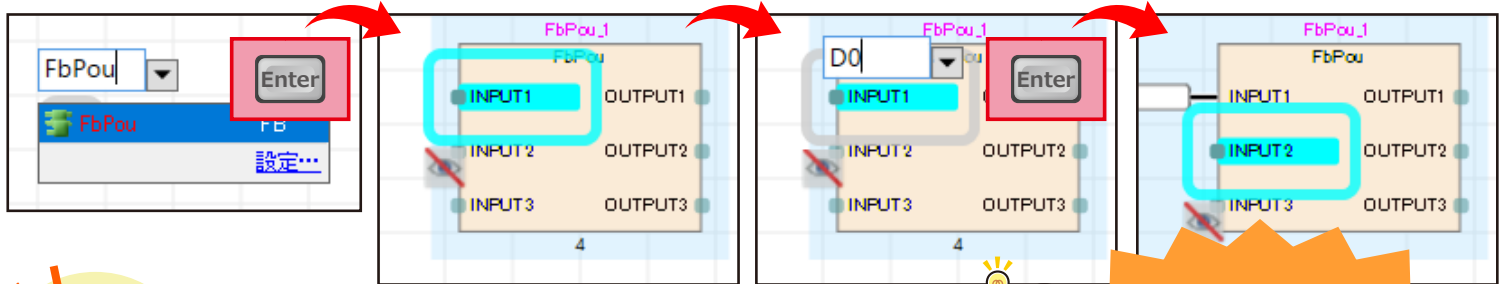


FBD/LDのFB/FUNを簡単操作!

FBD/LD エディタのFB/FUN を編集するときの便利な機能を紹介しますね!一部のFUNのスマートタグで、ピンの追加/削除ができるよ。



FB/FUN の追加や引数の編集がキーボードだけで簡単にできるんだ。[Enter] キーの押下で次の入力位置にカーソルが移動するよ。



プログラミングが楽になる機能がいっぱいあるね!



[Tab]/[Shift]+[Tab]キーで前後の引数にカーソルを移動することもできるよ

FB/FUN の便利な機能を使いこなそう!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!Gx3号 第109回

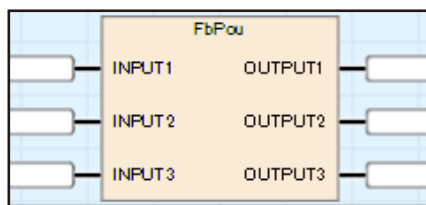
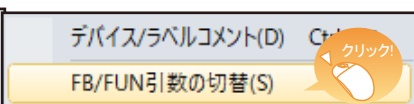
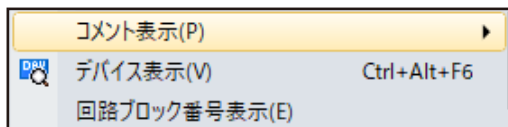


FBD/LDのFB/FUNのコメント確認

FBD/LD エディタの [表示] メニューから
FB/FUN の引数のラベル名をコメントに
切り替えて表示できるよ!



[表示]メニュー



プログラム全体の引数の表示を
一括で切り替えて確認できるんだ!

ラベルエディタの複数コメント表示設定から
英語や中国語のコメントにも切り替えられるよ。



ラベルエディタ

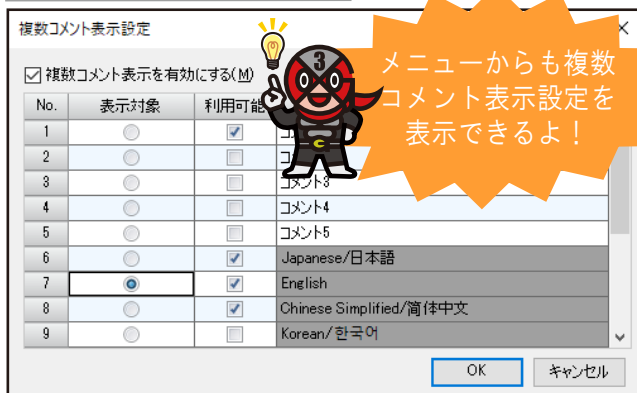


[表示設定]ウィンドウ



「詳細表示」に
してから「表示設定」
をクリック!

複数コメント表示設定



メニューからも複数
コメント表示設定を
表示できるよ!

コメント表示への切り替えはショートカット
キーに割り当ててることもできるんだ!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号 第110回



FBD/LDのコメントと部品をリンク

FBD/LD エディタのコメントは部品とリンクできるよ！
部品の移動に合わせてコメントも移動するんだ。



リンクしたい部品と
コメントを選択して
スマートタグを
クリック！



ドラッグすると
部品とコメントが
一緒に移動するよ！



コメントの位置を調整し直す手間が減るね！

[検索/置換]メニュー

回路ブロッカー覧(L)...

回路ブロッカー一覧から、コメントとリンク
した回路ブロックへジャンプもできるよ！



回路ブロッカー覧[プログラム本体: ProgPou]

No.	検索(E)
1	稼働状況判定
2	エラー判定
3	不良品カウント
4	生産数算出

回路ブロックへのジャンプ後、ダイアログを閉じる(N)

回路ブロックごとにわかりやすい
コメントを付けて使ってみてね！
これで君もシーケンスアープ！





システム設計の頼れるヒーロー 教えて!GX3号 第111回



FBD/LDのコメントをカスタマイズ!

今回は、FBD/LD エディタのコメントの便利機能を紹介しますよ!



コメントの表示順序変更

[編集]メニュー

簡易編集(Y)	
コメントの順序(M)	<ul style="list-style-type: none"> 最前面へ移動(E) 前面へ移動(B) 背面へ移動(S) 最背面へ移動(B)
プロセス制御拡張(O)	

初期化処理

クリック

バイスクリア

デバイスクリア

外部機器入力

コメントの色の変更

フォント色

背景色

色の設定

基本色(B):

作成した色(C):

色の作成(D) >>

OK

キャンセル

左揃えを使って
コメントを
揃えられるよ!



コメントのレイアウト変更

右揃えの場合

下揃えの場合

外部機器入力

外部機器入力

```

アラーム情報
ID       :Word
名称     :String
パラメータ :Word
発生日時:Time
    
```



メニューやスマートタグから、色々なカスタマイズができるんだね!用途に合わせて使い分けてみるよ。

コメント部品の見た目をカスタマイズしてプログラムを見やすくしよう!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号 第112回

NEW
Ver.Up
新機能

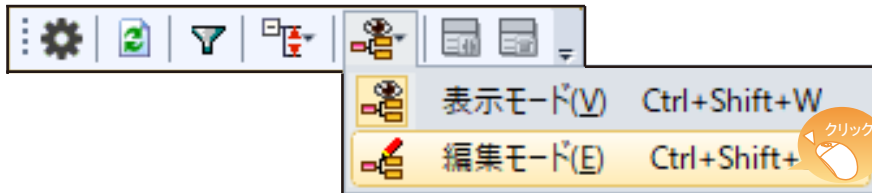
プログラム構成図を編集!

プログラム構成図でプログラム (POU) や FB/FUN の追加ができるようになったよ! ツールバーやショートカットキーから編集モードにしてみよう。



表示モードから編集モードに切替え

ツールバー

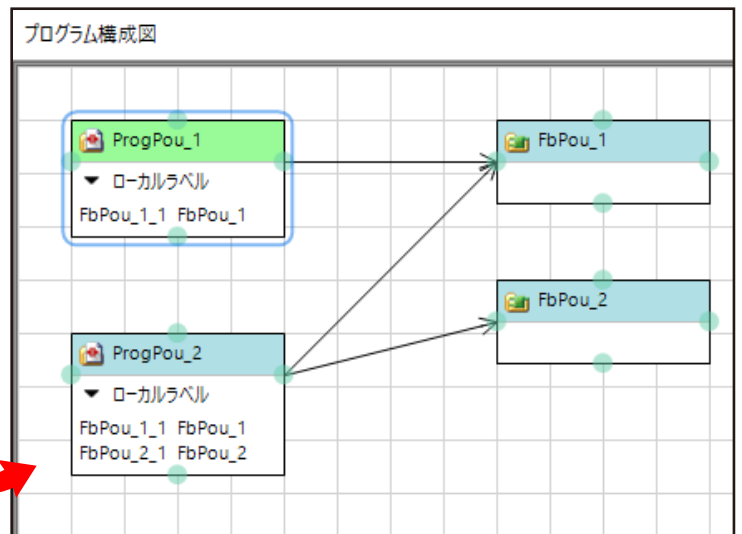
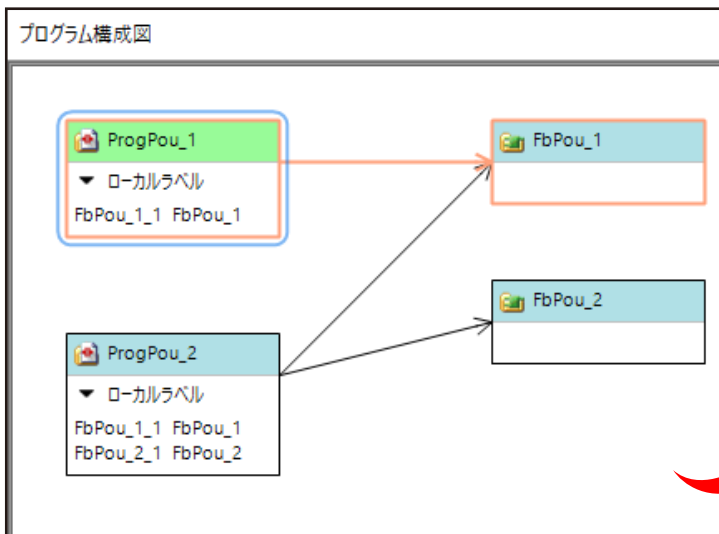


ショートカット

Ctrl + Shift + D

表示モード

編集モード

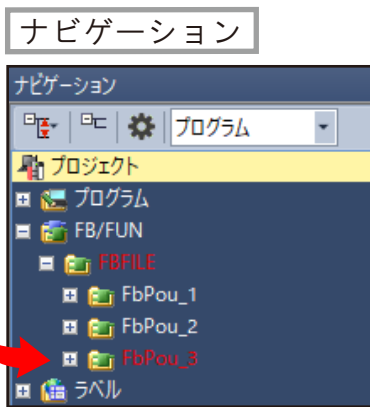
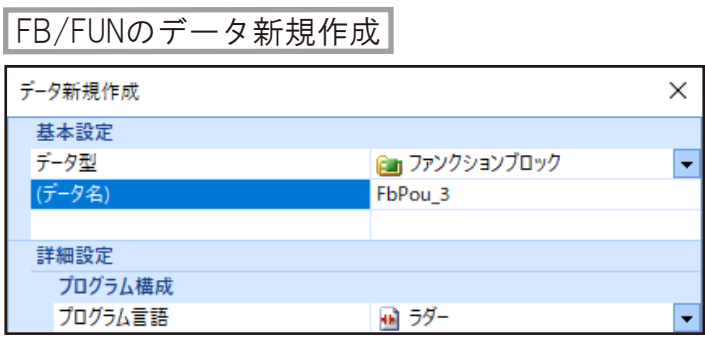


編集モードにすると、背景にグリッド線が表示されるんだね!

次に編集モードの使い方を紹介するね!



ツールバーのアイコンから POU や FB/FUN を追加してプログラム構成図に組み込んでみよう！



データを新規作成するとナビゲーションウィンドウにもデータが追加されるんだね！

追加した FB/FUN を他の POU や FB/FUN と繋いでみよう。繋いだ POU や FB/FUN から使えるようになるよ！



未定義ラベル登録

グローバルラベル、ローカルラベルに定義されていません。新規に登録するラベル情報を設定してください。

ラベル名(L) FbPou_3.1

ラベル設定情報

登録先(R) ローカルラベル(ProgPou_1)

クラス(S) VAR

データ型(Y) FbPou_3

定数値(N)

コメント(M)

ラベル情報を登録後、ラベルエディタを開きラベルの詳細を設定する。(E)

ラベルを一番後ろに登録する。(G)

OK キャンセル

使う側のPOUやFB/FUNからドラッグ！



繋ぐことでラベルの登録ができるんだね！

プロジェクト全体の構成を確認しながらFB/FUNの追加場所を考えてみてね。これで君もシーケンスアープ！





システム設計の頼れるヒーロー 教えて!Gx3号 第113回

NEW Ver.Up 新機能 デバイス使用一覧でコメントを編集!

デバイス使用一覧でコメントを編集できるようになったんだ!
ダブルクリックして編集してみよう!



コメントの編集

コメントのコピーや貼り付けもできるよ!



コメントのコピーと貼り付け



コメントのみのコピーや貼り付けは右クリックからだね!



[Ctrl]+[C]キーで行全体のコピーができるよ。他の資料に張り付けてね!

複数コメント表示設定

複数コメント表示を有効にする(M)

No.	表示対象	利用可能	
7	<input type="radio"/>	<input checked="" type="checkbox"/>	English

他の言語のコメントを編集したいときは表示対象の言語を切り替えてね!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号 第114回



ラダーでラベル名をより見やすく!

ラダーエディタでラベル名を見やすくカスタマイズできるようになったよ!
3種類から好きな表示設定を選んでね。



オプション

プロジェクト	回路図
プログラムエディタ	表示形式
共通項目	回路図の表示接点数 11接点
ラダーエディタ	ラベルとデバイスを同時に表示する いいえ
STエディタ	接点表示幅 1
FBD/LDIエディタ	デバイス/ラベル名の折返し表示行数 2
SFC図エディタ	アウトライン表示 いいえ
その他のエディタ	文字のサイズを自動調整する いいえ
編集	ラベル名の表示設定 メンバを優先して表示
検索/置換	編集時の動作
パラメータ	回路編集モード(読出し、書き込み、モニタリング)の表示形式
	先頭から表示
	末尾から表示
	メンバを優先して表示

先頭から表示

```
ProdNum.Go
odProdNu...
良品数
```

メンバを優先して表示

```
...GoodProd
Number
良品数
```

長いラベル名の場合の表示例を紹介するね。表示設定によって末尾の配列要素、構造体のメンバを確認しやすくなるよ。



ラベル名「AlarmData.AlarmOccurrenceDateTime[0]」の場合

先頭から

```
AlarmData.Alar
mOccurrenc...
発生日時
```

末尾から

```
...mOccurren
ceDateTime[0]
発生日時
```

メンバを優先

```
...AlarmOccur
renceDateTi...
発生日時
```



今作ってるプログラムでは配列の要素数が見れると嬉しいな。「末尾から表示」に設定してみるよ!

自分のプログラミングスタイルに合った表示設定を使ってね!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号 第115回



FBインスタンスのコメントを表示!

FB インスタンスのコメントをラダーエディタで表示できるようになったよ! ラベルエディタでFB インスタンスのコメント欄に入力してね。



ラベルエディタ

ラベル名	コメント(表示)
FbPou_A	装置A
FbPou_B	

ラベルのコメントが空欄の場合

FBのプロパティ

データ名	値
データ名	FbPou
タイトル	状態管理FB
更新日時	2023/05/17 15:31:04

インスタンスのコメントを表示

FbPou_A (FbPou)
装置A
B: bInLa... bOutL... :B



FBインスタンスのコメントは、タイトルよりも優先して表示するよ

FBのタイトルを表示

FbPou_B (FbPou)
状態管理FB
B: bInLa... bOutL... :B



今まで通りFBのタイトルを表示したいときは、インスタンスのコメントを空欄にしたらいんだね!

FB インスタンスのコメントを活用してラダーを見やすくしよう!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号 第116回



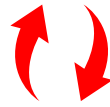
1クリックでウォッチを開始!

ウォッチウィンドウのウォッチ開始と停止が、ボタンから1クリックでできるようになったよ!



ウォッチ開始前(停止中)

名称	現在値	表示形式	データ型	割付(デバイス)	Japanese/日本語	強制入出力状態
X0	--	2進数	ビット	X0	ユニットREADY	--
D100	--	10進数	ワード[符号付き]	D100		--



ウォッチ中

名称	現在値	表示形式	データ型	割付(デバイス)	Japanese/日本語	強制入出力状態
X0	TRUE	2進数	ビット	X0	ユニットREADY	--
D100	100	10進数	ワード[符号付き]	D100		--



ウォッチが開始しているかどうかをボタンの状態からも確認できて便利だね!

名称	現在値	割付(デバイス)
入力信号(X):		
ユニットREADY	ON	X0

インテリジェント機能ユニットモニタにも同じようにボタンを追加しているよ!



今まで通り右クリックメニューからも操作できるから、スタイルに合わせて使ってね!これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号 第117回



インラインSTの高さを自動調整!

インライン ST の高さが自動で調整できるようになったよ! 編集中の行に合わせてボックスの高さが自動で変化するんだ。



```
1 IF bStart THEN
2     INC(TRUE,wProduct);
3 ELSE
4     bErr := TRUE;
5 END_IF;
```

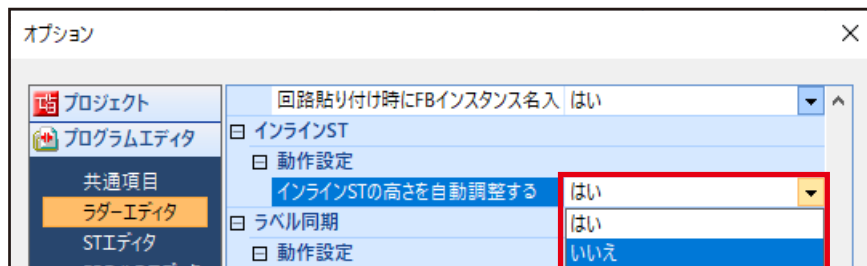
Enter × 4

```
1 IF bStart THEN
2     INC(TRUE,wProduct);
3 ELSE
4     bErr := TRUE;
5 END_IF;
6
7
8
9
```



[Enter] キーで改行する度に自動でボックスの高さが変わって、全体が見やすいね!

今まで通り高さを固定する場合はオプションから設定できるよ!



インライン ST の高さを、好みに合わせて設定しやすくなったね! これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号 第118回

NEW
Ver.Up
新機能

デバイスコメントの表示を一括開閉!

デバイスコメントエディタの改行したコメントを一括で開閉できるようになったよ!



COMMENT [デバイスコメント]		
デバイス名(N)	D0	
デバイス名	Japanese/日本語	
+ [+]	D0	動作状態
+ [+]	D1	エラー情報
+ [+]	D2	
+ [+]	D3	
+ [+]	D4	



改行したコメントをすべて非表示(N)
改行したコメントをすべて表示(O)



COMMENT [デバイスコメント]		
デバイス名(N)	D0	
デバイス名	Japanese/日本語	
- [+]	D0	動作状態 (装置 A)
- [+]	D1	エラー情報 (装置 A)
+ [+]	D2	



右クリックから一括で操作できるんだ!
コメントが確認しやすくっていいね。

ワードデバイスのビット指定の
コメントと同じように操作できるよ!



COMMENT [デバイスコメント]		
デバイス名(N)	D0	
デバイス名	Japanese/日本語	
- [+]	D0	動作状態 (装置 A)
- [+]	D1	エラー情報 (装置 A)
+ [+]	D2	
+ [+]	D3	



ビット指定情報をすべて非表示(B)
ビット指定情報をすべて表示(B)



COMMENT [デバイスコメント]		
デバイス名(N)	D0	
デバイス名	Japanese/日本語	
- [-]	D0	動作状態 (装置 A)
- [-]	D0.0	稼働中 (動作パターン1)
- [-]	D0.1	稼働中 (動作パターン2)

1行ずつ開閉するときは今まで通り左端の
[+]/[-] ボタンをダブルクリックしてね!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第119回



ラダーのショートカットキー紹介!

	ショートカットキー	動作内容
編集	F2	(書込みモード時)入力済みの回路記号を簡易編集
	F5 F7 F8	a接点/コイル/応用命令の入力
	/	回路記号(a接点/b接点)を切換え 演算結果反転の入力/削除
	Alt + /	立上り/立下りパルス、SET/RST命令を切換え
	Alt + ↑ Alt + ↓	デバイス番号の変更
	Ctrl + ← Ctrl + → Ctrl + ↑ Ctrl + ↓	罫線の入力/削除
	Ctrl + Shift + ← Ctrl + Shift + →	罫線の一括入力/削除
	Shift + Ins Shift + Del	行の挿入/削除
	Ctrl + Ins Ctrl + Del	列の挿入/削除
	Tab Shift + Tab	前/後の回路記号へジャンプ
	Alt + Ins	コピーした回路記号を挿入して貼り付け
モード切換え	Ins	挿入モード/上書きモードの切換え
	F2	(読出しモード時)書込みモードに切換え
	Shift + F2	読出しモードに切換え
	F3	モニタモードに切換え(共通)
	Shift + F3	モニタ(書込みモード)に切換え(共通)
その他	Ctrl + F11	コメント/ステートメント/ノートを一括で表示/非表示
	Ctrl + L	行間ステートメント一覧を表示
	F1	ヘルプを表示(言語共通)
	Ctrl + マウスのダブルクリック	アウトラインの展開/折りたたみ



[F2]を2回押すとコメントの編集!



[Alt + ↑]/[Alt + ↓]はデバイスのセルで使ってね!

今回はラダーエディタのショートカットキーだよ!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号 第120回



共通のショートカットキー紹介!

	ショートカットキー	動作内容
解析/ 検索	Ctrl + Shift + E	データフロー解析ウィンドウを表示
	Ctrl + E	クロスリファレンスウィンドウを表示
	Ctrl + D	デバイス使用一覧ウィンドウを表示
	Ctrl + F	デバイス/ラベル検索ウィンドウを表示
	Ctrl + Shift + F	文字列検索ウィンドウを表示
	Ctrl + H	デバイス/ラベル置換ウィンドウを表示
	Ctrl + Shift + H	文字列置換ウィンドウを表示
	Ctrl + Q	クイック検索ウィンドウを表示
変換	F4	変換
	Shift + F4	変換+RUN中書込み
	Shift + Alt + F4	全変換
モニタ	F3	モニタモードに切換え
	Shift + F3	モニタ(書込みモード)に切換え
	Alt + F3	モニタ停止
その他	Shift + Enter	現在値変更
	Ctrl + Enter	実行条件付きデバイステストの登録
	Ctrl + F12	次のブックマークにジャンプ
	Ctrl + Shift + F12	前のブックマークにジャンプ



[Shift+ Enter]を
押した後に、そのまま
数値を入力すると
現在値変更ができるよ



ブックマークは
右クリックメニューから
登録できるよ!

今回は共通のショートカットキーだよ!
これで君もシーケンスアープ!



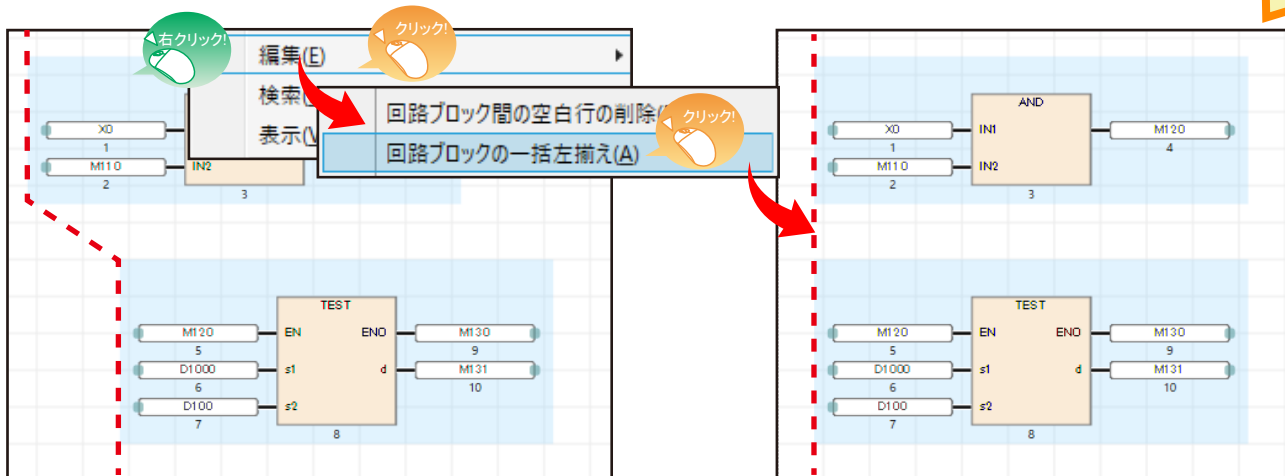


システム設計の頼れるヒーロー 教えて!GX3号 第121回



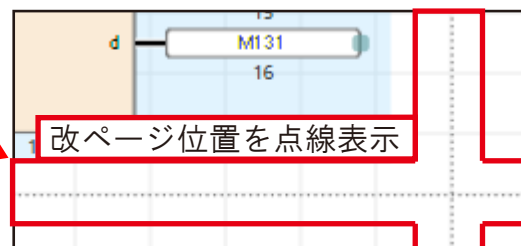
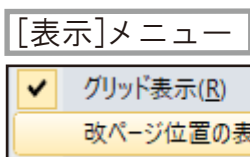
FBD/LDを整えて印刷!

今回は、FBD/LD エディタで見た目を整えるときや印刷するときに便利な機能を紹介するよ!
まずは回路ブロックを一括で左揃えしてみよう。



GX Works2 から流用したプロジェクトも簡単に整列できるね!

印刷するときはエディタ上の改ページ位置を点線で確認しよう! 改ページ位置の点線は、[表示]メニューから表示できるよ。



FBD/LD プログラムのドキュメント作成時に使ってみてね!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号 第122回



FBD/LDで割付けデバイスを活用!

FBD/LD エディタでも、グローバルラベルの割付けデバイスを使ってプログラムが作れるよ! [編集]メニューから [ラベル入力に割付けデバイスを使用] を選んでね。



グローバルラベルエディタ

ラベル名	データ型	割付け(デバイス/ラベル)
bTest	ビット	MO



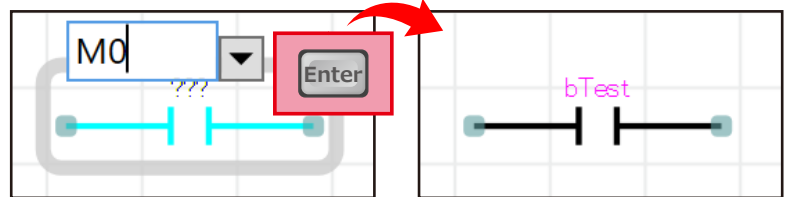
グローバルラベルエディタの割付けデバイスが使えるよ!

[編集]メニュー

編集モード(E) クリック!

自動結線(E) Ctrl+Alt クリック!

ラベル入力に割付けデバイスを使用(A) クリック!



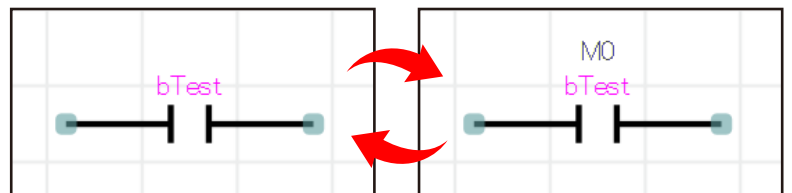
グローバルラベルと割付けデバイスのどちらでも入力できて便利だね!
エディタ上で割付けデバイスも確認できるのかな?

ラベルとデバイスを同時に表示することもできるんだ。表示メニューから設定しよう!



[表示]メニュー

デバイス表示(M) クリック! Ctrl+Alt+F6



今回紹介した機能は、オプションでも設定できるよ!
これで君もシーケンスアープ!



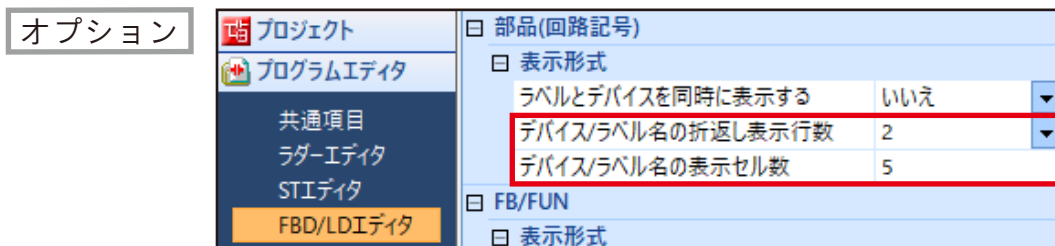


システム設計の頼れるヒーロー 教えて!GX3号 第123回

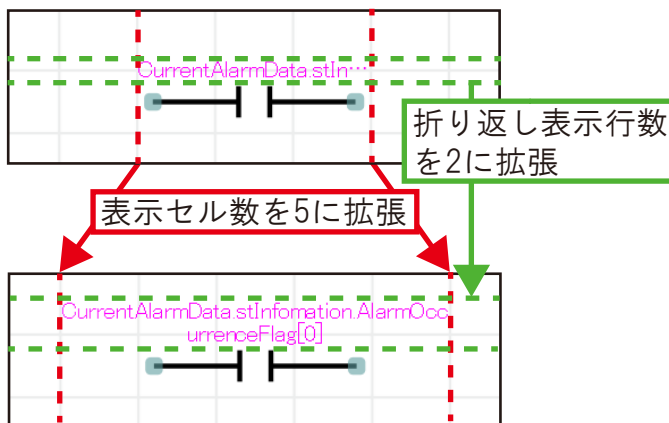


FBD/LDで長いラベル名を全体表示!

FBD/LD エディタで、ラベル名の表示を変更してみよう!
オプションで折返し表示行数と、表示セル数の2つを
変更できるよ。



<デフォルト>
折返し表示行数 : 1
表示セル数 : 3



<カスタマイズ後>
折返し表示行数 : 2
表示セル数 : 5



配列の要素数や構造体のメンバを記述してラベル名が
長くなっても、最後まで見れて便利だね!

オプション
[プログラムエディタ]
→[FBD/LDエディタ]



コメントの行数も、
同じようにオプション
から設定できるよ!



好きな表示形式にカスタマイズして使ってね!
これで君もシーケンスアープ!



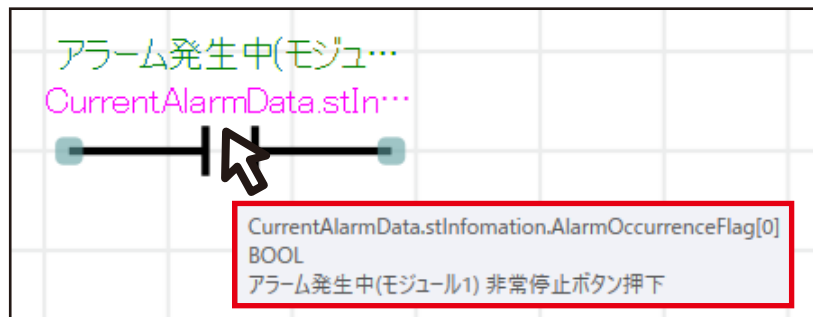


システム設計の頼れるヒーロー 教えて!GX3号 第124回



FBD/LDのツールヒントを表示!

FBD/LD エディタのツールヒントからも、
長いラベル名やコメントの全体を確認できるよ!
マウスカーソルをプログラム部品に合わせてみよう。



他の項目もツールヒントで確認できるのかな?

ツールヒントに表示する項目は
オプションでカスタマイズできるよ!



オプション	
プロジェクト	ツールヒント
プログラムエディタ	ツールヒントに表示する
共通項目	データ型
ラダーエディタ	クラス
STエディタ	定数値
FBD/LDエディタ	割付け(デバイス/ラベル)
SFC図エディタ	ラベルコメント
その他のエディタ	ラベルの備考
	デバイスコメント

自分好みにツールヒントをカスタマイズしてみてね!
これで君もシーケンスアープ!



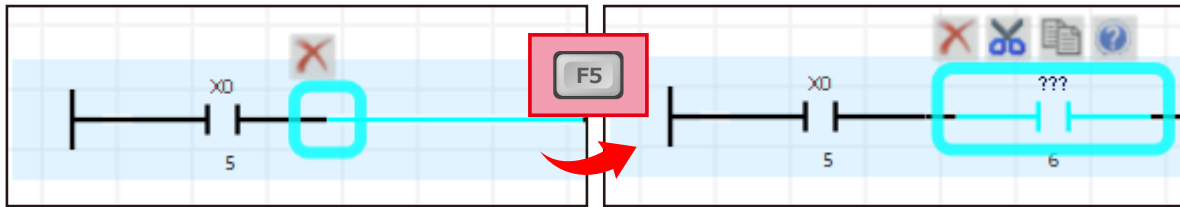


システム設計の頼れるヒーロー 教えて!GX3号 第125回



FBD/LDでネットワークの部品を操作!

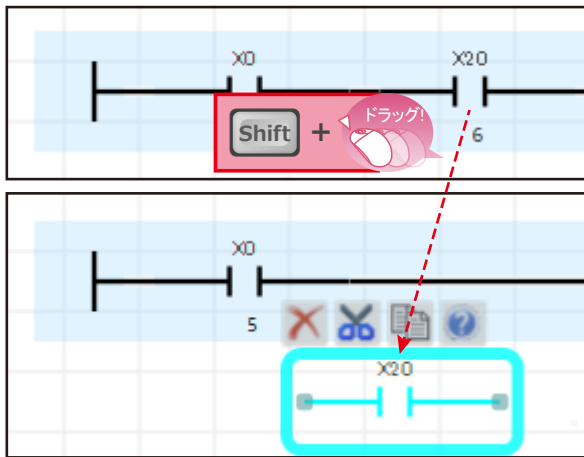
今回は FBD/LD エディタでの部品の操作について紹介するよ! まずは作成済みの回路に接点を挿入してみよう。



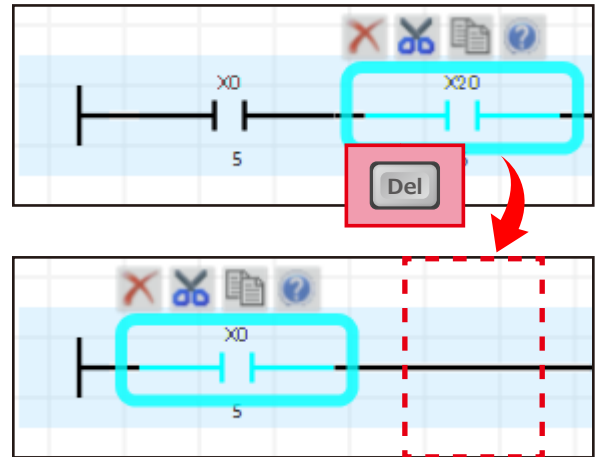
1 操作で選択した部品だけを抜き取ったり削除したりすることもできるよ!



回路中の部品を抜き取り



回路中の部品を削除



接続線を切断しなくても部品を操作できるんだ!

回路の修正も効率よく作業できるね!
これで君もシーケンスアープ!



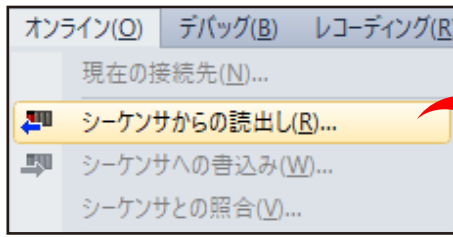


システム設計の頼れるヒーロー 教えて!GX3号 第126回



デバイスメモリの読出しを改善!

デバイスメモリの読出しを改善したよ! GX Works3 でプロジェクトを開いていなくても、シーケンサからデバイスメモリを読み出せるようになったんだ。



デバイスメモリのチェックボックスにチェックを入れて読み出そう。



シーケンサからデバイスメモリも含めて1回で読み出せるようになったんだね!

[全選択] ボタンやお気に入り設定を活用して、もっと便利に読み出そう! これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号 第127回



シーケンサの接続先を流用!

一部の異なる機種間で、接続先の設定を流用できるようになったよ!
容量のみが異なる機種間で試してみてね。



今プロセスCPUのR08PCPUとR16PCPUを使ってるから、設定をコピーしてみるよ!

同じ内容を設定する手間が削減できるね!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号 第128回



データフロー解析を印刷!

データフロー解析の結果を印刷できるようになったんだ!
印刷したい範囲のフローを表示してね。



The screenshot shows a window titled 'データフロー解析' (Data Flow Analysis) with a diagram of a 'd100' device. A red arrow points from the diagram to a '印刷' (Print) dialog box. The dialog has two main sections: '印刷項目(D)' (Print Items) and '項目別設定(データフロー解析)' (Item-specific Settings for Data Flow Analysis). In the '印刷項目' list, 'データフロー解析' (Data Flow Analysis) is checked and highlighted with a red box. The '項目別設定' section has '表示内容' (Display Content) set to 'データフロー解析ウィンドウと同じ設定で印刷する(T)' (Print with the same settings as the Data Flow Analysis window) and '表示(L)' (Display) set to 'デバイス/ラベルを表示' (Display device/label).



D100 の値がどこに影響するかを調べたいから、
D100 の右側だけを開いて印刷してみようかな。

デバイス/ラベルやコメントといった印刷時の表示内容は
カスタマイズできるよ!印刷のダイアログから選択してね。



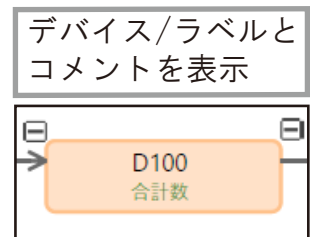
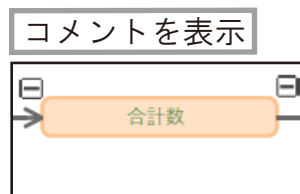
表示内容

- データフロー解析ウィンドウと同じ設定で印刷する(T)
- 指定した表示内容で印刷する(E)

表示(L) デバイス/ラベルを表示

表示行数(N) デバイス/ラベルを表示

位置情報の表示 デバイス/ラベルとコメントを表示



資料を作成する時に使ってみてね!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号 第129回



ユニット検索とウォッチへの登録を改善!

ユニットに関する機能が改善されたから紹介するね!
新規ユニット追加ウィンドウのユニット検索で、
「ユニット種別」も検索対象になったんだ。



新規ユニット追加

温度

ユニット選択

ユニット種別

ユニット形名

局種別

詳細設定

装着位置

ユニット検索

温度

ユニット種別	ユニット形名	特徴
温度入力	R60RD8-G	8チャンネル
温度入力	R60RD8-G(Q)	8チャンネル
温度入力	R60TD8-G	8チャンネル
温度入力	R60TD8-G(Q)	8チャンネル



「温度入力」などのユニット種別で探せるのは便利だね!

部品選択ウィンドウにあるユニットラベルの
操作性も改善したよ! ドラッグ & ドロップで
ウォッチウィンドウに登録してみてね。



部品選択

(部品検索)

表示対象: 0000:R60RD8-G

- ユニットラベル
 - 0000:R60RD8-G
 - R60RDG_1
 - R60RDG_1 (バージョン: 01A)
 - uIO (先頭I/O No.)
 - 入出力信号
 - 入力信号
 - bModuleREADY ユニットREADY
 - bOperatingConc 動作条件設定完了フラグ

ウォッチ1

ON OFF ON/OFF反転 更新 ウォッチ開始

名称	現在値	表示形式
R60RDG_1.bModuleREADY	--	--



目的のユニットラベルを簡単にモニタできるんだ!

ユニットの追加とウォッチが効率的にできるね!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号 第130回

NEW
Ver.Up
新機能

変化点検索一覧の列をソート!

以前に紹介したオフラインモニタの変化点検索が使いやすくなったよ!変化点検索一覧で列をソートできるようになったんだ。



変化点検索

条件
デバイス/ラベル(N)

検索種別(C)

データ型(D)

選択中のデータを指定した条件で検索します。

>

検索開始点(T)

指定した検索開始点から条件に合致するものを検索一覧に表示します。

変化点検索一覧を表示

変化点検索一覧

検索結果:8件

日時	インデックス	wLabel1	wLabel2
		検索対象値	比較対象値
2021/2/18 20:12:09.368	3334	1	5
2021/2/18 20:12:09.368	3335	2	5
2021/2/18 20:12:09.369	3336	3	5
2021/2/18 20:12:09.370	3337	4	5

「wLabel1」列を降順にソートして表示

変化点検索一覧

検索結果:8件

日時	インデックス	wLabel1	wLabel2
		検索対象値	比較対象値
2021/2/18 20:12:09.370	3337	4	5
2021/2/18 20:12:18.451	14817	4	5
2021/2/18 20:12:09.369	3336	3	5
2021/2/18 20:12:18.450	14816	3	5

「wLabel1」列を昇順にソートして表示

変化点検索一覧

検索結果:8件

日時	インデックス	wLabel1	wLabel2
		検索対象値	比較対象値
2021/2/18 20:12:09.368	3334	1	5
2021/2/18 20:12:18.448	14814	1	5
2021/2/18 20:12:09.368	3335	2	5
2021/2/18 20:12:18.449	14815	2	5

列のヘッダをクリックするたびに昇順と降順が切り替わってソートされるんだ!どの列でもソートできて便利だね。

ソートしている列が昇順なら「△」、降順なら「▽」が表示されるよ!

変化点検索の検索結果が確認しやすくなったね!
これで君もシーケンスアープ!





システム設計の頼れるヒーロー 教えて!GX3号

第1回

特別編

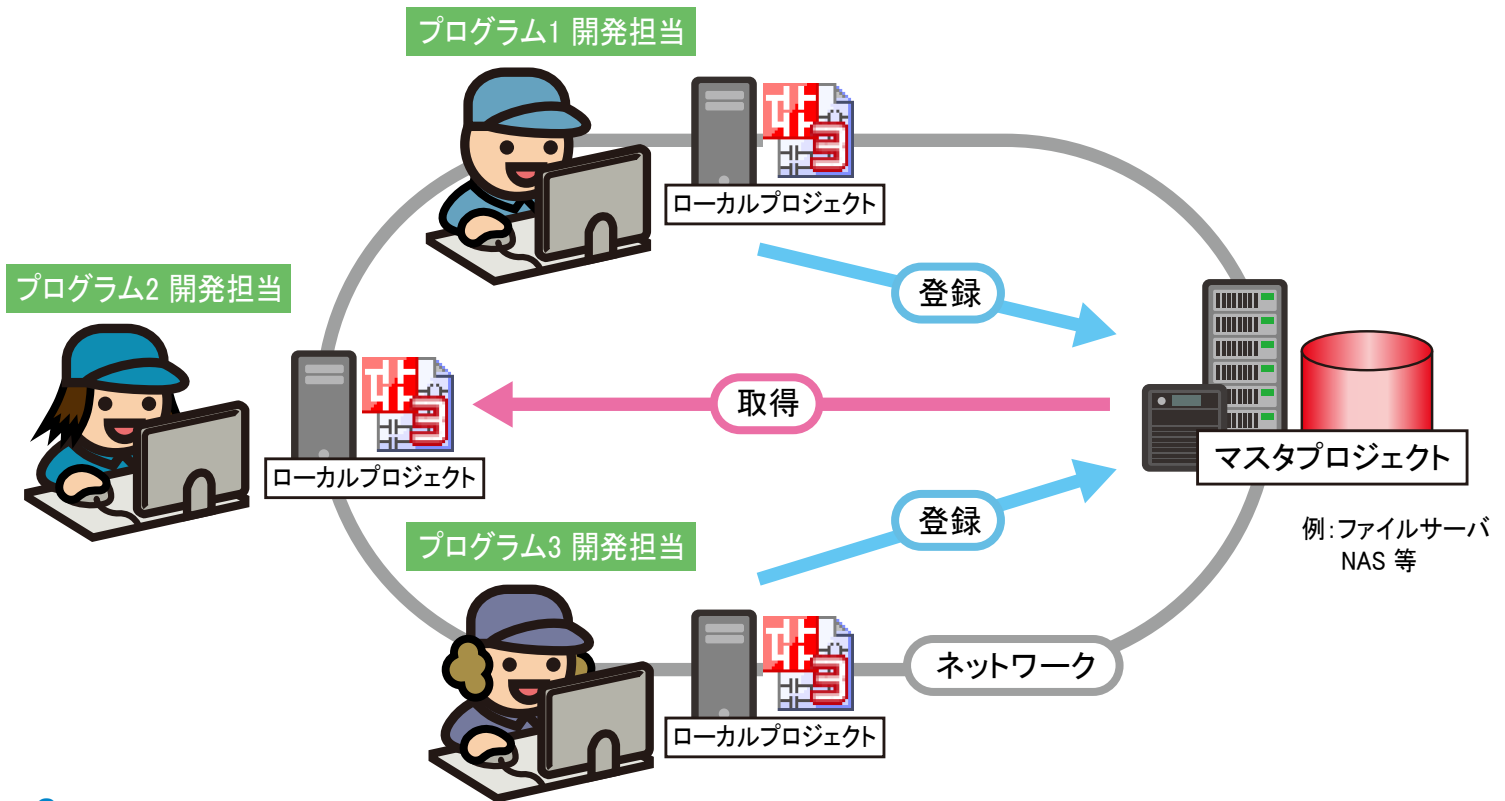
NEW
Ver.Up
新機能

1つのプロジェクトを複数人で開発!



プロジェクトの規模が大きいから
複数人でプログラムを開発したいなあ。

複数人で1つのプロジェクトを楽に編集できるようになったよ! 既存のネットワーク環境を利用してマスタプロジェクト共有することで開発できるようになったんだ!



複数人でマスタプロジェクトに登録していくと
せっかく編集したデータが上書られない?

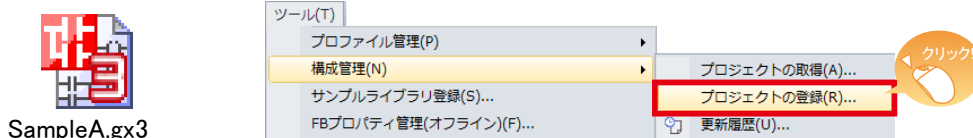
マスタプロジェクトからデータの取得・登録するときは
プログラム単位(POU)で行えるんだ!
プログラム単位で作業者を分ければ上書きも防げるよ!



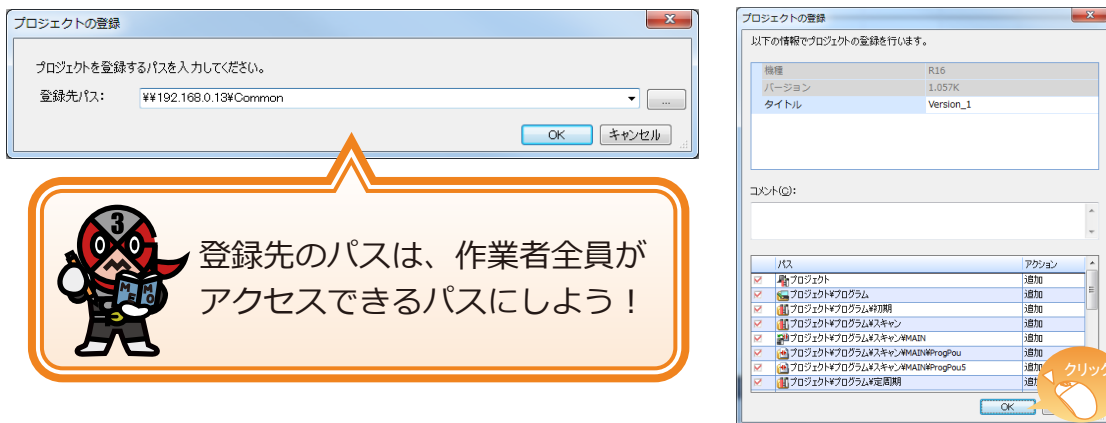
マスタプロジェクトとして登録するプロジェクトを準備しよう！
以前作成したプロジェクトを流用すると便利だよ！



プロジェクトを開いてマスタプロジェクトとして登録

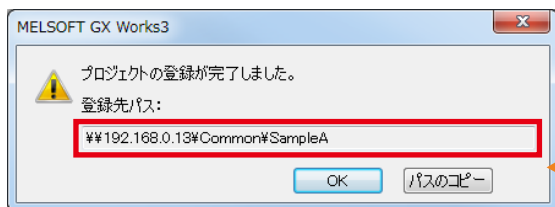


マスタプロジェクトを作成するファイルパスを指定！



登録先のパスは、作業者全員が
アクセスできるパスにしよう！

マスタプロジェクトの登録先パスが完成！



登録完了したときに登録先パス
をコピーできるよ！他の作業者
に展開するときに使おう！



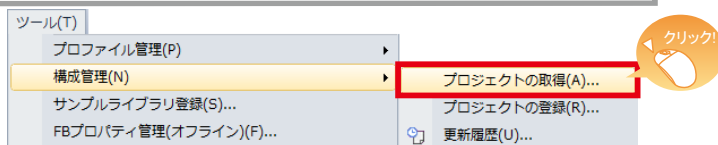
なるほど！これでマスタプロジェクトの登録が完了だね！

マスタプロジェクトからローカルプロジェクトの取得

ローカルプロジェクトとして作業用のプロジェクトを取得しよう！



GX Works3を起動してプロジェクトを取得しよう！



マスタプロジェクトを取得するパスを指定！

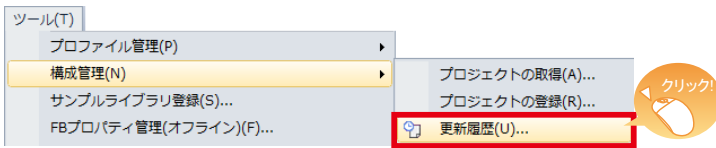


これで編集準備完了だね！ローカルプロジェクトを編集
したらマスタプロジェクトにどうやって登録するの？

マスタプロジェクトへの登録は更新履歴画面から行うんだ！



更新履歴

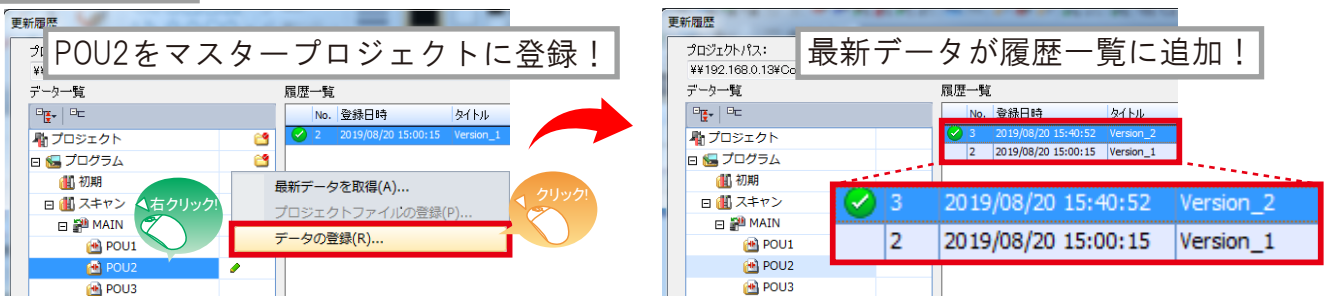


自分が担当したプログラムについて
マークは何だろう？

は自分で編集したってことだよ！
 忘れずに登録してね！



データ登録方法



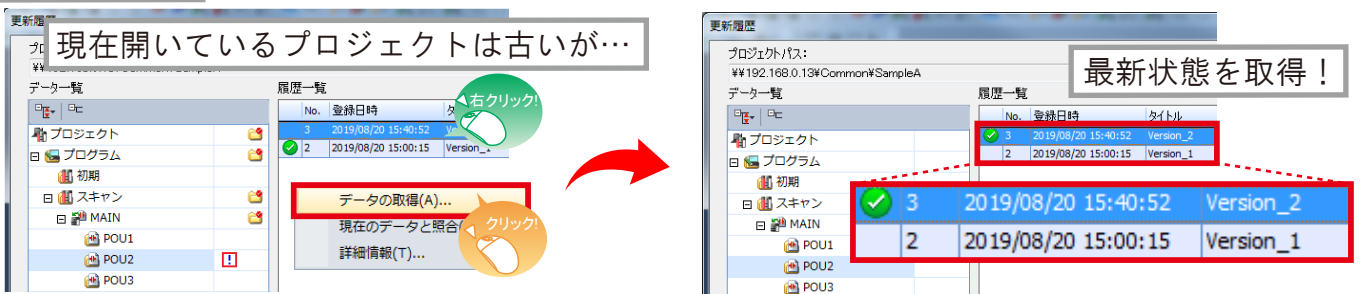
マークは他の作業者と編集データが競合しているから注意が必要だよ！
 このまま登録すると上書きされるので他の作業者に確認してね！
 現在のデータと照合して編集箇所を確認することもできるよ！

他の作業者の編集したプログラムの取得

他の作業者のプログラムの取得も更新履歴画面から行うんだ！



データ取得方法



マークはマスタプロジェクトに
 他の作業者の更新がある状態だよ！

最新のデータにするにはマークの
 データを取得したら良いんだね！



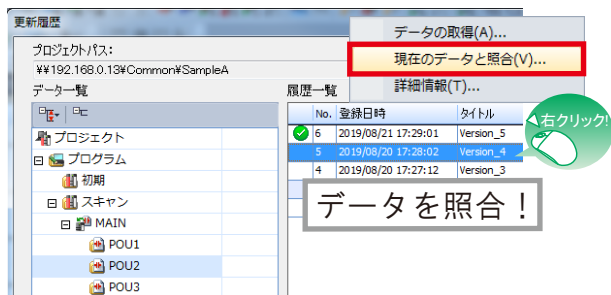


あれ？デバッグしたら動作がおかしくなったな…。昨日までは正しく動作してたのにどうしよう…。

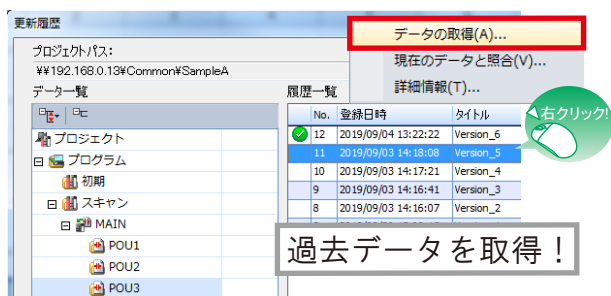
以前のデータと照合して確認してみても！
今まで登録したデータの履歴から必要なデータを選択して戻すこと(ロールバック)ができるよ！



照合方法



ロールバック方法

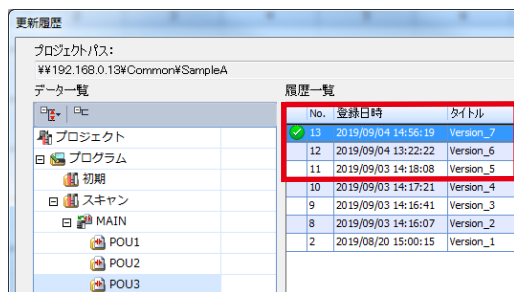


過去データでも取得方法は同じだよ！
登録した履歴は全て残せるから改定したら登録しよう！



他の作業者の担当プログラムをロールバックしたら上手くいったよ！POU3の担当者に連絡して修正してもらおうよ！

修正後のデータを登録



登録データの詳細

No.	登録日時	タイトル
13	2019/09/04 14:56:19	Version_7
12	2019/09/04 13:22:22	Version_6
11	2019/09/03 14:18:08	Version_5



確認してプログラムを修正しておくよ！

やり直したデータの登録を忘れないでね！



現地調整前の机上でのプログラム作成を効率化できるよ！これで君たちもシーケンスアープ！





システム設計の頼れるヒーロー 教えて!GX3号

第2回

特別編

NEW
Ver.Up
新機能

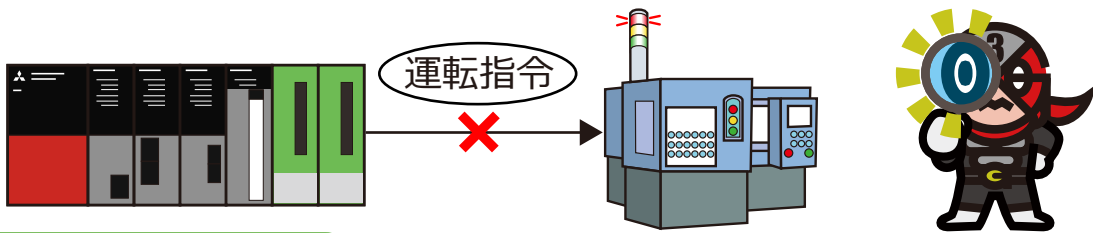
全機種対応! フロー表示で影響調査



あれ? 試運転してみたけど装置が動かないな…。
出力のデバイスを確認した方がいいかな?

データフロー解析を使ってみよう!
注目するデバイスを基点にして、関わる
デバイスのフロー表示ができるんだ!

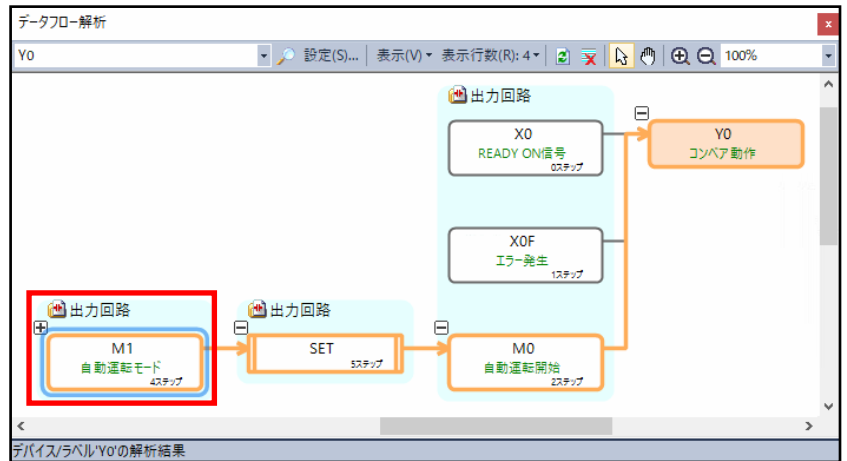
そんなときは



1 注目したいデバイスを選択

X0	X0F	M0	Y0
READY ON信号	エラー発生	自動運転開始	コンパア動作

2 データフロー解析で表示



3 確認したい回路にジャンプ

M1	M0
自動運転モード	SET 自動運転開始

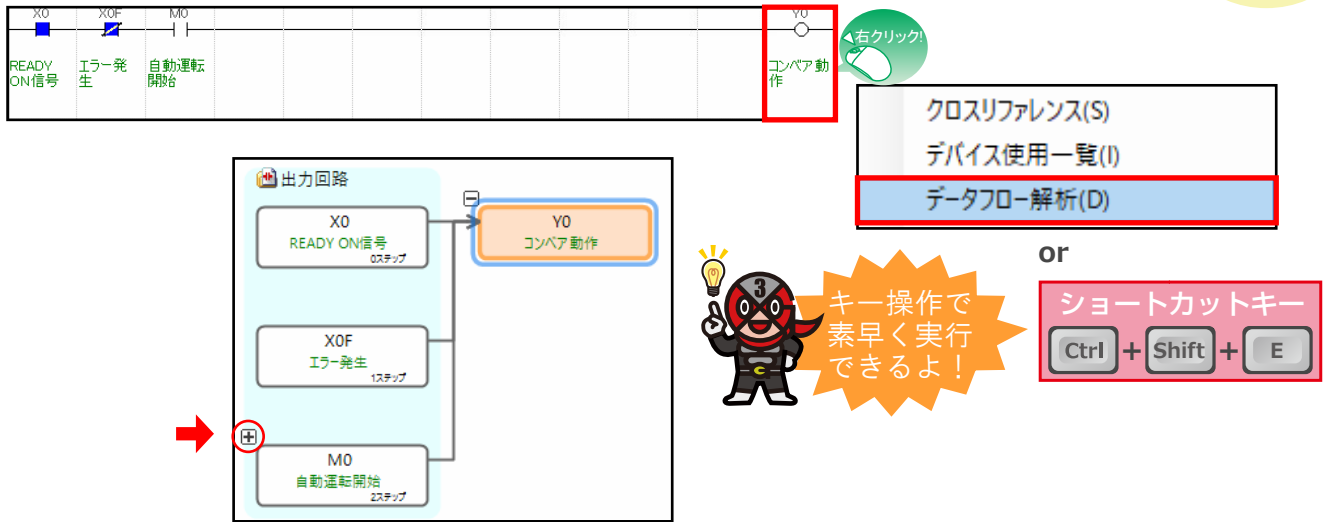
影響を与えるデバイスを一目で確認できるよ!



確認したい回路にもすぐジャンプできるんだね!
使い方を詳しく教えてよ!



基点となるデバイスからフロー表示しよう！
右クリックメニューからデータフロー解析を選択するんだ！



[Y0] の条件がフローに表示されるんだね！
この⊕マークは何だろう？

⊕をクリックすればさらに前の条件が見られるんだ！



命令名も併せて見られるから回路のイメージがしやすくなるね！

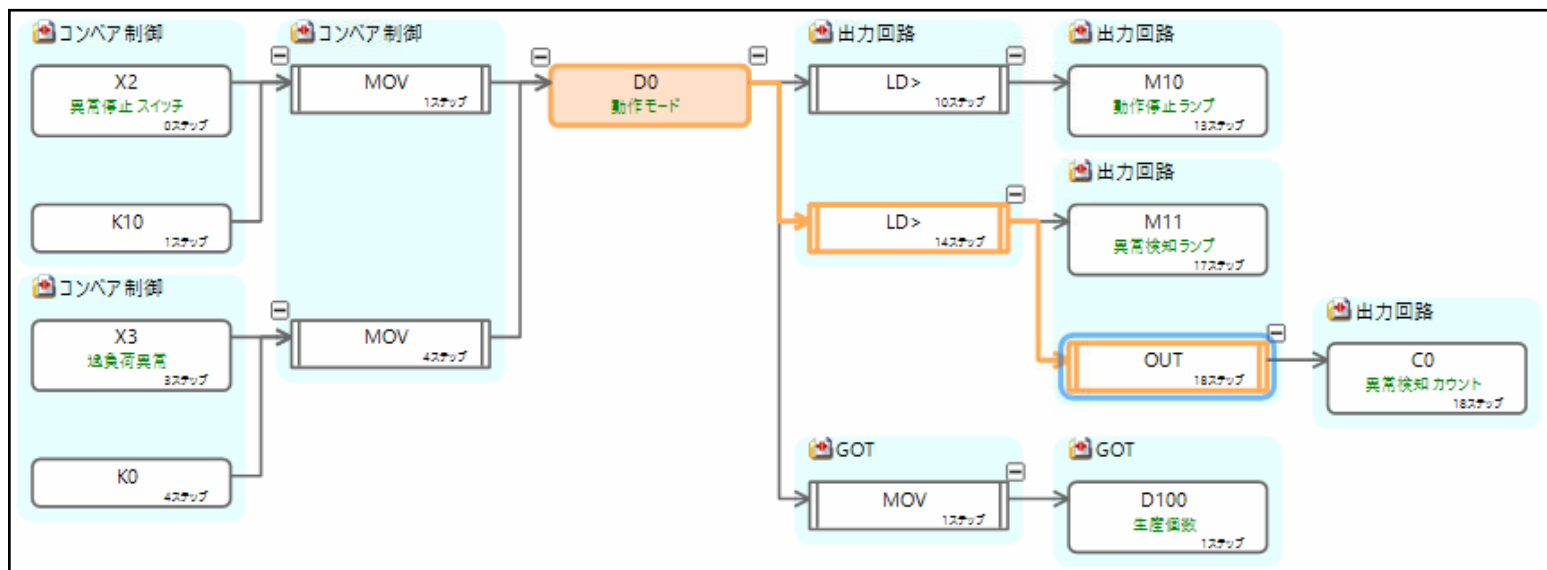


デバイスの流れが一目でわかるから原因特定がしやすくなるね！
トラブル原因の報告にも役立ちそうだ！



原因の個所はわかったけど、修正による影響範囲の確認が大変だなあ。

データフロー解析ならどの部分に影響を与えるかフロー表示で追えるんだ！



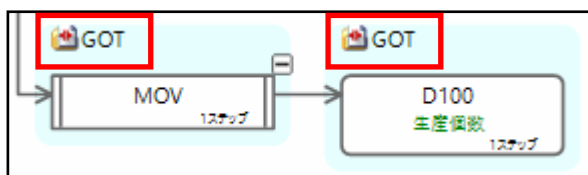
影響範囲

基点となるデバイスから右側が影響範囲だよ！



関係する部分がまとめて見られるから影響範囲の確認がしやすいね！

プログラム名も併せて確認できるんだ！



上にあるのがプログラム名だよ！



あれ!? プログラム [GOT] に影響を与えるみたいだ！

そのプログラムの担当者と相談して修正してね。



見落としがなくなると修正も安心だね！

データフロー解析はラダー言語だけでなく
ST 言語、FBD/LD 言語でも利用できるよ！

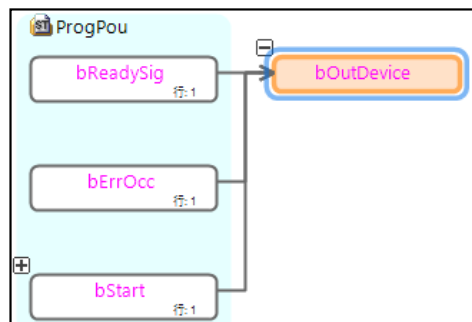


インライン ST でも使えるみたいだね！
デバイス / ラベルの関係性がわかりやすいし便利だね！

```

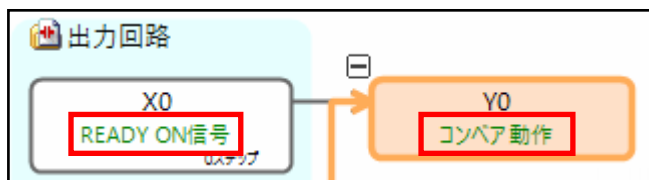
1 bOutDevice := bReadySig AND NOT bErrOcc AND bStart;
2 IF bTemp1
3   SET(TR
4 END_IF;
5 bTemp1 :=
6 IF G_wTemp
7   MOV(TRUE,K10,wTemp10);
8 END_IF;
9 IF G_wTemp1 > K10 THEN
10  MOV(TRUE,wTemp20,wTemp10);
11  OUT_C(TRUE,cdCount,K3);
12 END_IF;
    
```

- クロスリファレンス(S)
- デバイス使用一覧(I)
- データフロー解析(D)**



コメント
表示は切り替えが
できるよ！

- コメントを表示
- 表示(V) ▾
- デバイス/ラベルを表示(D)
 - コメントを表示(C)
 - デバイス/ラベルとコメントを表示(A)**
 - 位置情報の表示(P)



MELSEC iQ-R/iQ-Fの全機種に対応！

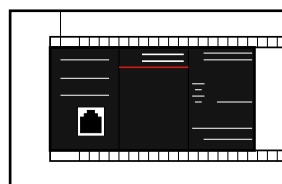


今まで作ってきた GX Works3
プロジェクトでも使えるのかな？

GX Works3 でサポートする全ての CPU で使えるんだ！
MELSEC iQ-R も iQ-F も対応しているよ！



MELSEC iQ-F でも使えるのはいいね！



データフロー解析を使ってエラーの原因追及や
デバッグを効率よく行おう！
これで君もシーケンスアープ！





システム設計の頼れるヒーロー 教えて!GX3号

第3回

特別編

NEW
Ver.Up
新機能

装置アラームをかんたん解析!



装置にアラームが出ていたんだけど、
原因の特定に時間がかかりそうだなあ。

レコーダユニットやカメラレコーダユニットで
記録したデータを、GX Works3 と GX LogViewer、
GX VideoViewer を連携して簡単に解析できるよ!



記録してあるデータを、連携して解析できるんだ!
もっと詳しく教えてよ!

今回は、装置の動作を解析するためにデータの
変化点から追っていく方法について紹介するよ。



まずは、レコーディングファイルからデータフロー解析でアラームの原因となったデバイス/ラベルを特定しよう！



データフロー解析なら前回教えてもらったね！

デバイス/ラベルがわかったら変化点検索を使ってみよう！



GX Works3

変化点検索

条件
 デバイス/ラベル(N) MO
 検索種別(C) 立下り
 データ型(D) ビット

検索開始点(I) 先頭のインデックス
 指定した検索開始点から条件に合致するものを検索

検索一覧(L)

表示(V)
 ツールバー(T)
 ✓ ステータスバー(B)
 色およびフォント(O)...
 ドッキングウィンドウ(K)
 ズーム(Z)

or

ツールバー
 ナビゲーション(N)
 接続先(I)
 イベント履歴(オフラインモニタ)(M)
 シークバー(オフラインモニタ)(K)
 変化点検索(オフラインモニタ)(X)
 変化点検索一覧(オフラインモニタ)(J)

変化点検索一覧

検索結果:4件

日時	インデックス	M0	検索対象値	比較対象値
2021/2/18 20:12:09.361	3327	FALSE		
2021/2/18 20:12:09.362	5428	FALSE		
2021/2/18 20:12:11.035	14808	FALSE		
2021/2/18 20:12:20.115	16907	FALSE		

ダブルクリックで変化点の日時にジャンプするよ。オフラインモニタで見てみよう。

変化点が見つかったら、シークバーを表示させよう！
 気になるところにログマーカーをつけてね。



シークバー

ログマーカー(L)

2021/2/18 20:12:09.362 インデックス 3327 / 22773] ファイル保存トリガ: 設定No.1, No.1

気になる箇所にログマーカーをつけたよ！

気になる箇所にログマーカーをつけたよ！





ログマーカーを GX LogViewer、GX VideoViewer と連携して何が起きていたのか確認してみよう！

GX Works3

プログラム上でのデバイス/ラベルの状態や関係を確認

- プログラムのオフラインモニタ (値の状態を確認)
- データフロー解析 (デバイス/ラベルの関係を確認)
- イベント履歴



GX LogViewer / GX VideoViewer

GX LogViewer

波形からデバイス/ラベルの値の変化を確認

- ユニットで収集したデータおよびイベントをビジュアルにグラフ表示 (データの変化を直感的に確認)
- グラフの重ね合わせ機能 (複数のグラフデータの差異を簡単に確認)

GX VideoViewer

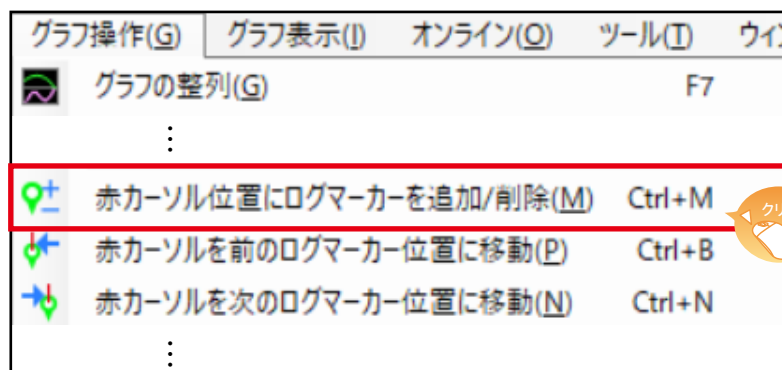
動画から装置の動作を確認

- 再生速度の調整及びコマ送り戻し機能 (詳細な装置の動作を分析可能)

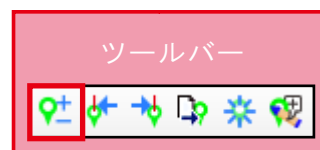
確認中に気になるところが見つかったら、さらにログマーカーを追加してね。



GX LogViewerの場合



or



GX VideoViewerの場合



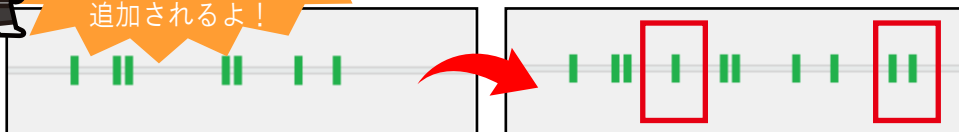
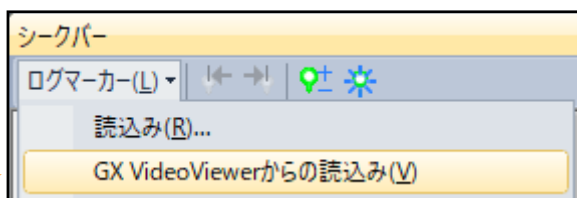
追加したログマーカーはツール間で共有しよう！
GX VideoViewer でつけたログマーカーを読み込もう。



GX Works3

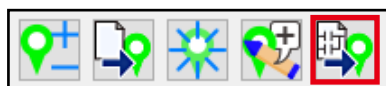


既存のマーカーはそのまま！読み込んだログマーカーが追加されるよ！



別のツールでつけたログマーカーでも読み込めるんだね！

GX VideoViewer



GX VideoViewer ではこのボタンで読み込めるよ！

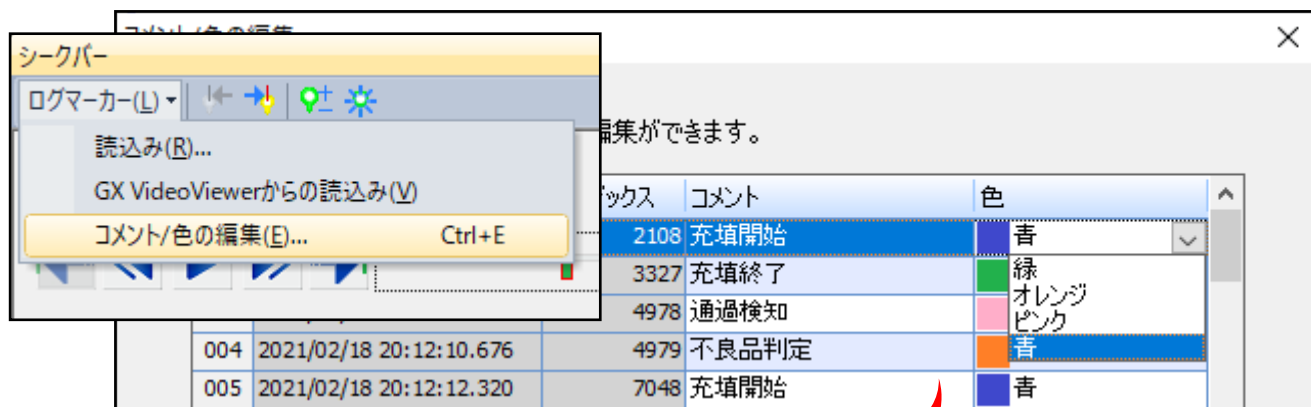


色とコメントを編集して使い分け！

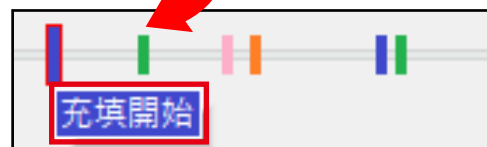
ログマーカーの判別をしやすいするためにコメントの記載と、色の変更ができるよ。



GX Works3



ほかの人に調査を引き継ぐときにも役に立ちそうだね



用途によって色分けするのもいいね。
マウスカーソルを合わせるとコメントを確認できるよ！



ログマーカーを基点に、3つのツールを使いこなそう！
これで君もシーケンスアープ！





システム設計の頼れるヒーロー 教えて!GX3号

第4回

特別編



プログラム外の世界をAI解析!



新しい装置の動作とプログラムの内容がよくわからないのに異常が起きたんだ…。シーケンサのプログラム部分はデータフロー解析で何とか理解したけど、装置の動作とプログラムの関係ががまだわかっていないんだよね。

そんな時はデータフロー解析で AI 解析!
AI がシーケンサの入出力の関係を解析するよ!



AI解析がログデータから装置動作を解析!



いままでのGXWorks 3

プログラム内の解析のみで装置動作が影響する部分については解析できない

装置のどの設計資料を見たらいいのかな?
たくさんある中から探さないといけなくて大変だ…



これからのGXWorks 3

AI解析でログデータから装置動作が影響を与えるプログラムの関係性を推測可能!

データフロー解析で装置動作をすぐに把握!
たくさん設計資料を見なくてもいいんだ!

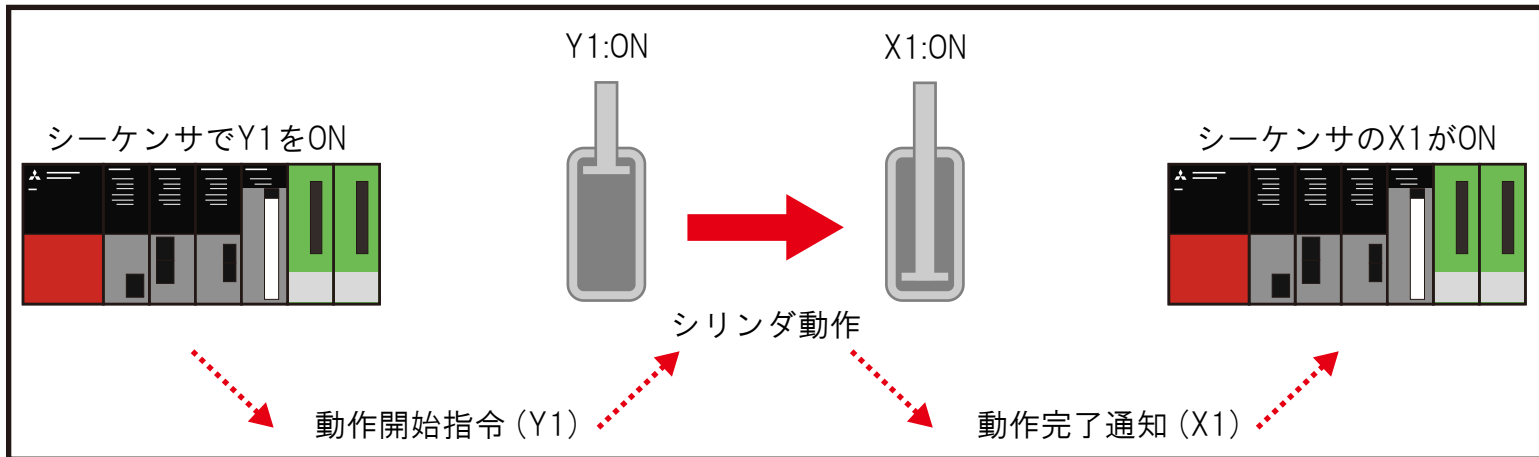


AIで解析なんてすごそうだなあ!

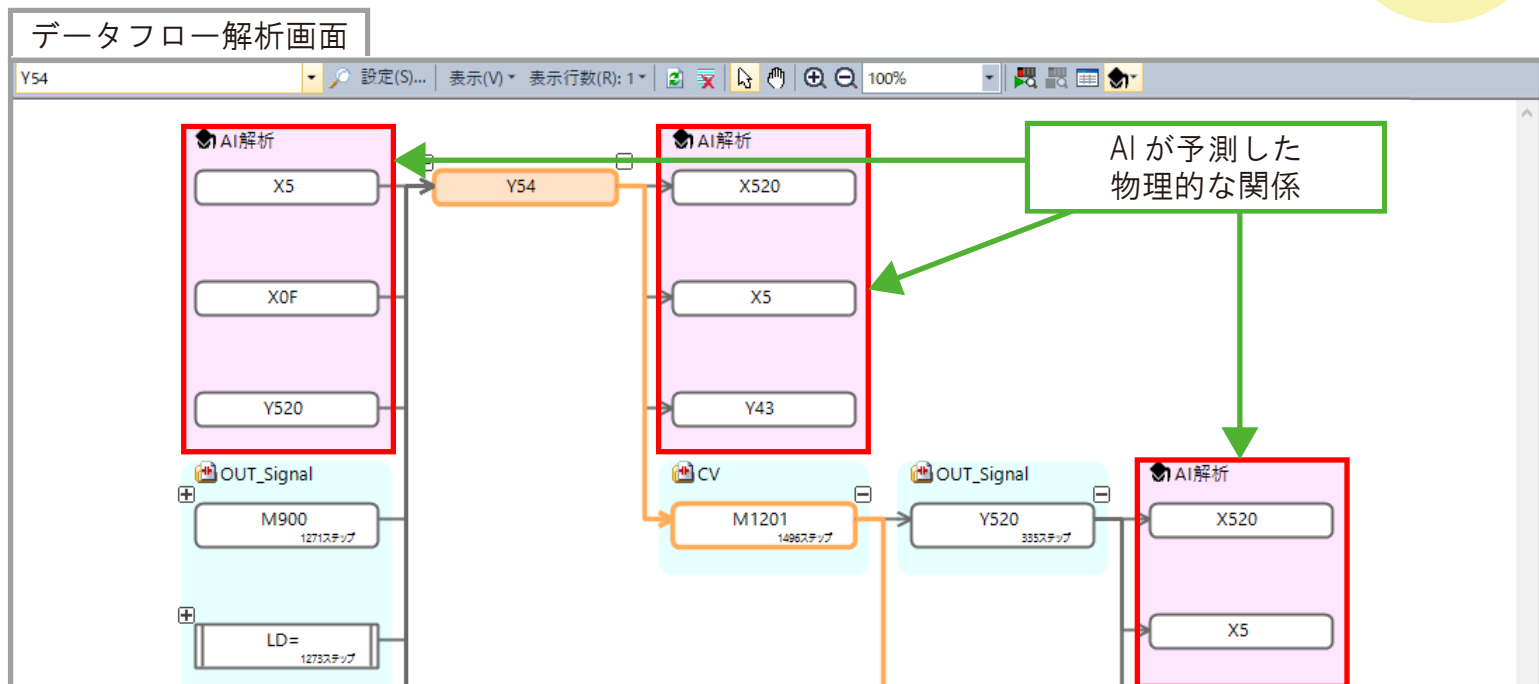


どんなことをAIが解析してくれるの？

装置の物理的な関係を見つけるんだ。例えば、シリンダの動作指令と動作完了の信号のようにプログラム上ではわからない物理的なデバイスの関連を見つけるよ。



見つけた物理的な関係をデータフロー解析上のピンク枠で表示するよ



どうやってAIが解析するの？

シーケンサが収集した装置のログデータを使って解析するんだ。





実際にAIを使ってみたいんだけど、
使い方についてもっと詳しく教えてほしいな。

まずはAIにログデータで学習させよう！
10～30サイクル分の正常動作時のログ
データを使うのがおすすめだよ。



AI解析で使用するログデータ

- ロギングファイル(.bin)
- レコーディングファイル (.melrc)



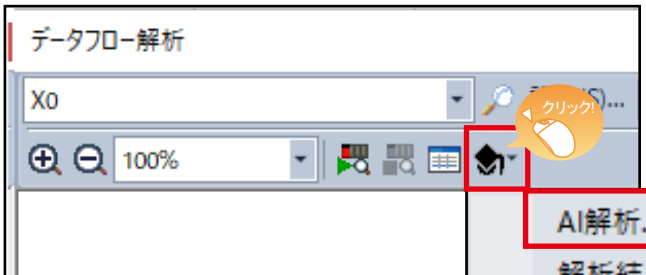
今までの装置稼働時のログデータで学習するんだね！

Let's AI解析！ : ログデータのファイルを選択

それじゃあ、AI解析の手順を説明するね。
データフロー解析からAIにログデータを渡そう！
事前に必要なログファイルを用意しておいてね。

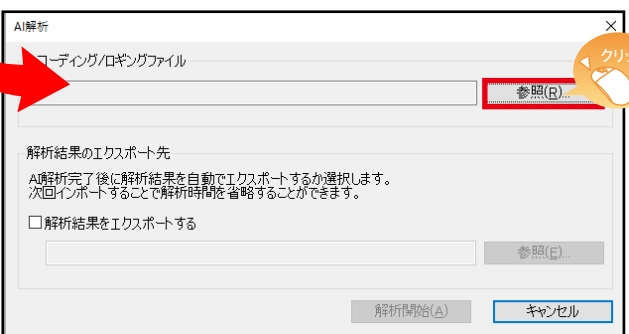


AI解析アイコンをクリックし、[AI解析]を選択



AI解析...
解析結果のインポート...
解析結果のエクスポート...

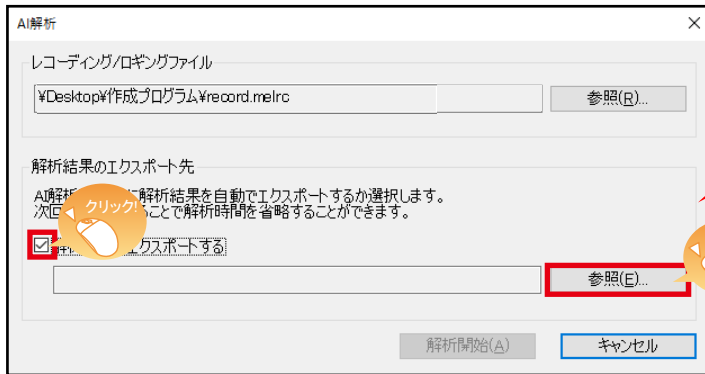
レコーディング / ロギングファイル [参照]
で解析ファイルを選択



[解析結果をエクスポートする] にチェックをつけてエクスポートしてね。

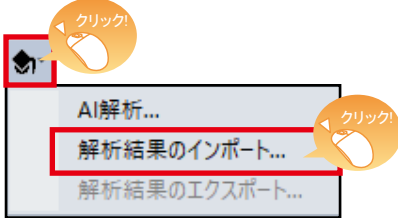


[解析結果をエクスポートする] をクリック



次回からエクスポートした解析結果をインポートして使ってね!

[解析結果のインポート] を選択



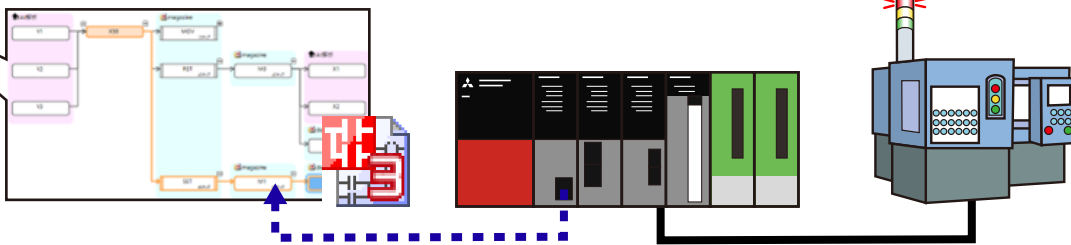
Let's AI解析!



準備が出来たら [解析開始] ボタンをクリックして、AI 解析しよう!



新しい世界が広がりそうでわくわくするね! 装置異常のデバッグで、早速試してみよう!



AI 解析でプログラムと装置動作の理解を深めてエラー原因の追及を効率よく行おう! これで君もシーケンスアープ!



MELSOFT GX Works3 教えて！GX3号

三菱電機株式会社

〒100-8310 東京都千代田区丸の内2-7-3(東京ビル)

お問合せは下記へどうぞ

本社機器営業部	〒100-8310	東京都千代田区丸の内2-7-3(三菱電機本社 東京ビル7F)	(03)3218-2606
関越機器営業部	〒330-6034	さいたま市中央区新都心11-2(明治安田生命さいたま新都心ビル)	(048)600-5835
新潟支店	〒950-8504	新潟市中央区東大通2-4-10(日本生命新潟ビル8F)	(025)241-7227
神奈川機器営業部	〒220-8118	横浜市西区みなとみらい2-2-1(横浜ランドマークタワー)	(045)224-2624
北海道支社	〒060-8693	札幌市中央区大通西3-11(北洋ビル)	(011)212-3792
東北支社	〒980-0013	仙台市青葉区花京院1-1-20(花京院スクエア)	(022)216-4546
北陸支社	〒920-0031	金沢市広岡3-1-1(金沢パークビル)	(076)233-5502
中部支社	〒450-6423	名古屋市中村区名駅3-28-12(大名古屋ビルヂング)	(052)565-3314
豊田支店	〒471-0034	豊田市小坂本町1-5-10(矢作豊田ビル)	(0565)34-4112
関西支社	〒530-8206	大阪市北区大深町4-20(グランフロント大阪 タワーA)	(06)6486-4122
中国支社	〒730-8657	広島市中区中町7-32(ニッセイ広島ビル)	(082)248-5348
四国支社	〒760-8654	高松市寿町1-1-8(日本生命高松駅前ビル)	(087)825-0055
九州支社	〒810-8686	福岡市中央区天神2-12-1(天神ビル)	(092)721-2247

三菱電機 FA www.MitsubishiElectric.co.jp/fa

インターネットによる情報サービス「三菱電機FAサイト」
 三菱電機FAサイトでは、製品や事例などの技術情報に加え、トレーニングスクール情報や各種お問い合わせ窓口をご提供しています。また、メンバー登録いただくマニュアルやCADデータ等のダウンロード、eラーニングなどの各種サービスをご利用いただけます。

技術相談チャットサービス

専門オペレータがチャットでお答えします！




FAWeb Shop <https://fa-webshop.MitsubishiElectric.co.jp/>

すぐ欲しい、今使いたいを、即注文！「三菱電機FAソリューションWeb Shop」
 お客様のものづくりをトータルでサポートする便利なウェブショップです。FA製品の小口・緊急でのご注文だけでなく、ものづくりや働き方の変化に対応したサービス・トレーニングスクールもご提供いたします。

電話技術相談窓口 受付時間※1 月-木 9:00~19:00、金 9:00~17:00 土・日・祝 9:00~17:00
 共通電話番号 052-712-2444

▶ 共通電話番号にお電話いただき、お客様相談内容に関する代理店、商社への「情報展開可否」をご承諾いただける場合は(1)、「ご承諾いただけない場合は(2)」のいずれかを入力後、「製品番号」を入力してください。製品番号は、ガイドスの途中でも入力いただけます。

対象機種	製品番号	受付時間、備考	対象機種	製品番号	受付時間、備考
産業用PC	産業用PC MELIPC	8	MELSERVOシリーズ		
ソリューションソフトウェア	Edgecross対応ソフトウェア (NC Machine Tool OptimizerなどのNC関連製品を除く) MELSOFT MailLab/MELSOFT VIXIO CC-Link IE TSN通信ソフトウェア	2 ⇒ 4	位置決めユニット (MELSEC IQ-R/Q/Lシリーズ) モーションユニット (MELSEC IQ-R/IQ-Fシリーズ) モーションソフトウェア モーションソフトウェア モーションコントローラ/ モーションユニット/ センシングユニット/ 組込み型サーボシステム コントローラ	1	
FA統合コントローラ	MELSEC MXコントローラ MX-Rモデル	2 ⇒ 2	モーションCPU (MELSEC IQ-R/Qシリーズ) センシングユニット (MR-MTシリーズ) シンプルモーションボード		
	MELSEC MXコントローラ MX-Fモデル	2 ⇒ 3	MELSOFT MTシリーズ/MRシリーズ/EMシリーズ		
シーケンサ	MELSEC IQ-R/Q/Lシリーズ (CPU内蔵Ethernet機能などネットワークを除く) MELSOFT GXシリーズ (MELSEC IQ-R/Q/L)	2 ⇒ 2	センサレスサーボ インバータ	3	
	MELSEC IQ-F/FXシリーズ (MELSEC IQ-F/FX) ネットワークユニット (CC-Linkファミリー/MELSECNET/Ethernet/シリアル通信)	2 ⇒ 1	表示器 GOT	4	
情報連携ユニット	MELSOFT通信支援ソフトウェアツール	2 ⇒ 3	産業用ロボット	5	☑を除く
	MELSEC/Q/ノコンボード Q80BDシリーズなど	2 ⇒ 4	FAセンサ MELSENSOR	6	☑、☒を除く
システムレコーダ	WinCPUユニット/C言語コントローラ/C言語インテリジェント機能ユニット	2 ⇒ 4	低圧遮断器	7 ⇒ 1	
	MESインタフェースユニット/高速データロガーユニット/高速データコミュニケーションユニット/OPC UAサーバユニット/GX LogViewer	2 ⇒ 5	低圧開閉器	7 ⇒ 2	☑を除く
MELSOFT統合エンジニアリング環境	レコーダユニット/カメラレコーダユニット/GX VideoViewer/GX VideoViewer Pro	2 ⇒ 6	電力管理用計器	7 ⇒ 3	
iQ Sensor Solution	MELSOFT Navigator/MELSOFT Update Manager	2 ⇒ 6	省エネ支援機器	7 ⇒ 4	☑、☒を除く
MELSEC計装/iQ-R/Q二重化	プロセスCPU/二重化機能/SIL2プロセスCPU (MELSEC IQ-Rシリーズ)	2 ⇒ 7	小容量UPS(5kVA以下)	7 ⇒ 5	☑、☒ 9:00~17:00 ☑、☒を除く
	プロセスCPU/二重化CPU (MELSEC-Qシリーズ) MELSOFT PXシリーズ	2 ⇒ 8	SCADA GENESIS64™	—	※2
MELSEC Safety	安全シーケンサ (MELSEC IQ-R/Qシリーズ) 安全コントローラ (MELSEC-WSシリーズ)	2 ⇒ 8	MELSOFT Gemini	—	※3
電力計測ユニット/絶縁監視ユニット	QEシリーズ/REシリーズ	2 ⇒ 9	MELSOFT Mirror	—	※4
			三相同モータ		[直通] 9:00~17:00 0536-25-0900 ☑、☒ 当社休日を除く
			電磁クラッチ・ブレーキ/テンションコントローラ		[直通] 9:00~17:00 052-712-5430 ☑、☒ 当社休日を除く

お問い合わせの際には、今一度電話番号をお確かめの上、お掛け間違いのないようお願いいたします。FA製品に関する最新情報は、「三菱電機FAサイト」< www.MitsubishiElectric.co.jp/fa >よりご確認ください。また、電話技術相談窓口の最新情報は、「三菱電機FAサイト」< www.MitsubishiElectric.co.jp/fa/contact-us >よりご確認ください。

e-Factory Alliance

三菱電機の e-F@ctory コンセプトはFA技術とIT技術を活用してe-F@ctory アライアンスパートナーとのコラボレーションにより、開発費用の削減、生産性の向上および保守の改善により「一歩先を行く」ものづくりを目指すことです。

**三菱電機FAサイト
仕様・機能に
関するお問合せ**



- ※1: 春季・夏季・年末年始の休日を除く
- ※2: SCADA GENESIS64™の技術相談は、「三菱電機FAサイト」< お問い合わせ >の「仕様・機能」>ウェブページからのお問い合わせ、またはGENESIS64™保守サービス (SupportWorX) (有償)の技術サポート窓口をご利用ください。詳細は、三菱電機FAサイトより、GENESIS64™保守サービス (SupportWorX)ガイド (BHP-F0005-0026)をご参照ください。
- ※3: MELSOFT Geminiの技術相談は、MELSOFT Gemini有償保守サービスをご利用ください。詳細は、三菱電機FAサイトより、3Dシミュレータ MELSOFT Gemini リーフレット (L(名)08815)をご参照ください。
- ※4: MELSOFT Mirrorの技術相談は、MELSOFT Mirrorの技術サポート窓口 (購入者向けサービス)をご利用ください。詳細は、三菱電機FAサイトより、MELSOFT Mirror オペレーティングマニュアル (SH-082663)をご参照ください。